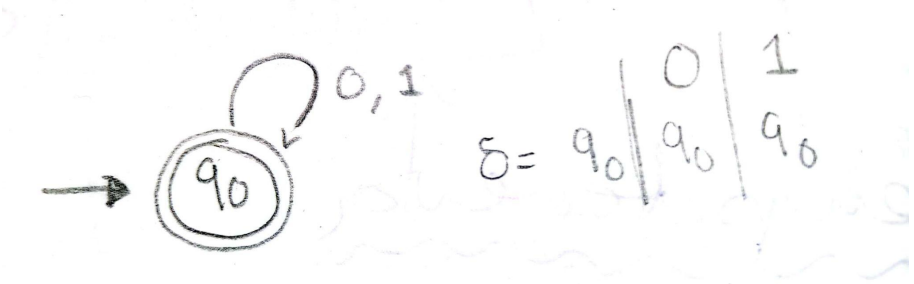
Tecnológico de Monterrey

Actividad: Actividad 3.2. Ejercicios sobre autómatas.
Materia: Implementación de Métodos Computacionales.
Profesor: Raul García Jacas
Alumno: Héctor Miranda Garía
Matrícula: A01658845

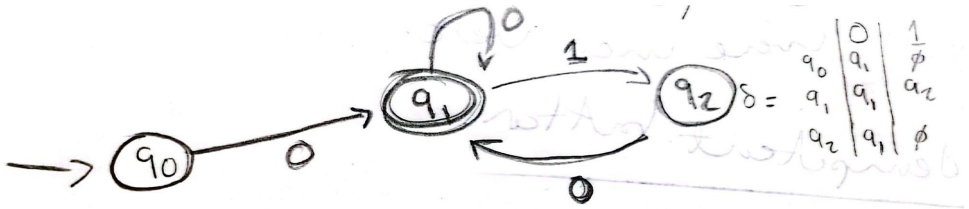
Actividad 3.2 Ejercicios sobre autómatas

II.VII. Construya los autómatas finitos deterministas que reconozcan los siguientes lenguajes.

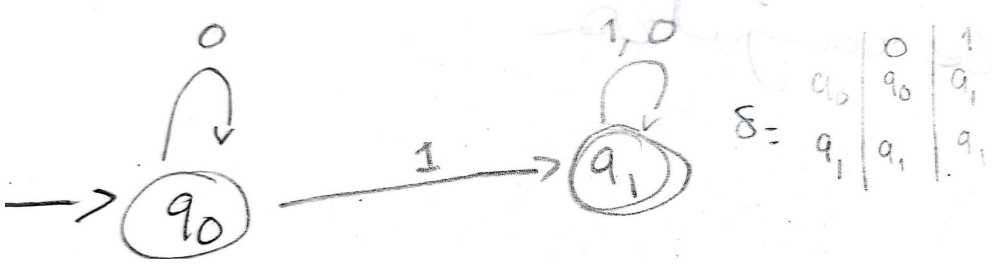
I. Todas las cadenas sobre el alfabeto {0, 1}.



II. Las cadenas sobre el alfabeto {0, 1} que comienzan y terminan con 0.

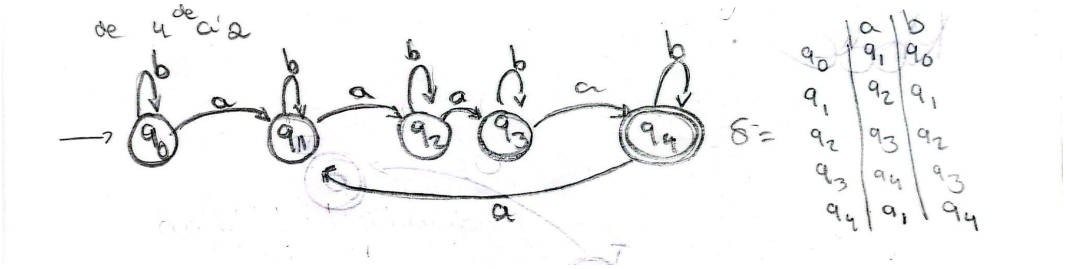


III. Las cadenas sobre el alfabeto {0, 1} con un 1 al menos.

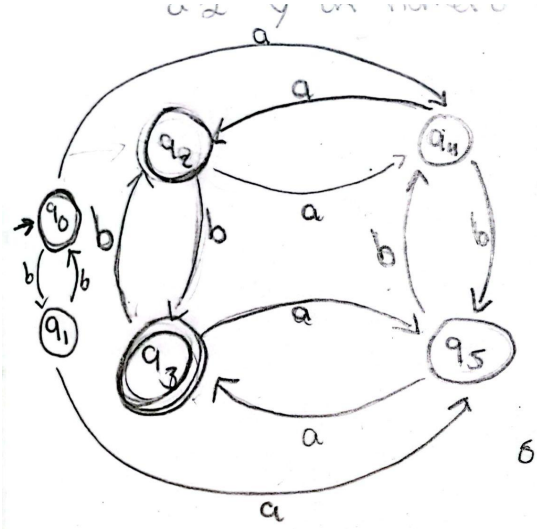


II.VIII. Diseñe un AFD que reconozca los siguientes lenguajes sobre {a, b}.

I. Todas las palabras tienen un número múltiplo de 4 a 's.



II. Todas las palabras tienen un número par de a 's y un número impar de b' s.



abbababaaaab

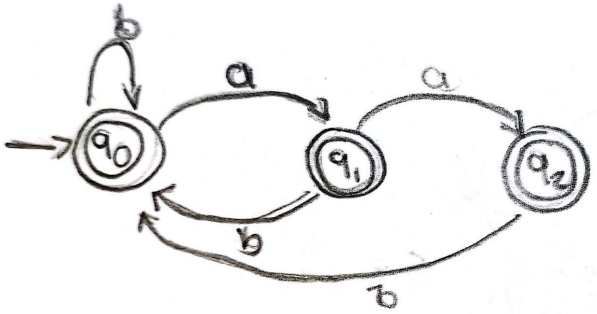
aaab

baabbaaaaa

$\delta =$

	a	b
q_0	q_4	q_1
q_1	q_5	q_0
q_2	q_4	q_3
q_3	q_5	q_2
q_4	q_2	q_5
q_5	q_3	q_4

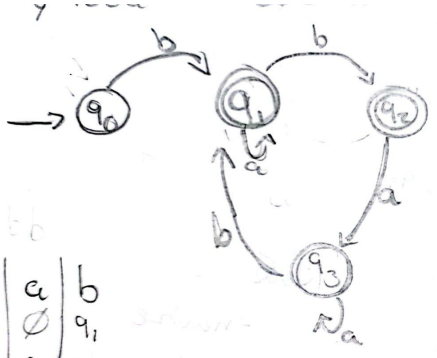
III. Todas las palabras no tienen tres a 's consecutivas.



$\delta =$

	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	$\emptyset$	q_0

IV. Todas las palabras no tienen tres b 's consecutivas y toda a está entre dos b' s.



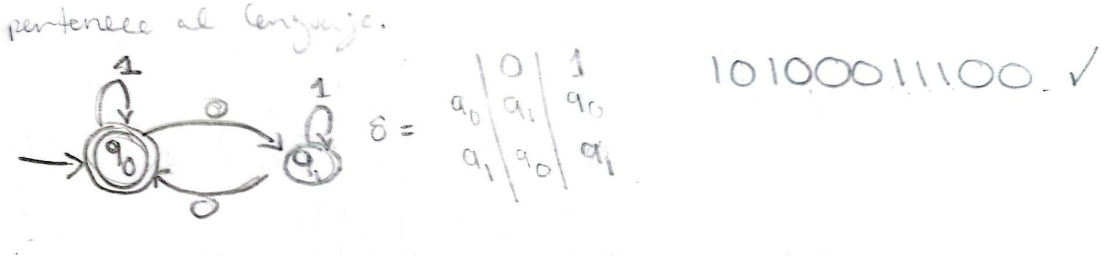
bbaaaaabbbab ✓

bbabbbaaab ×

babaaaabbaaab ✓

	a	b
q_0	$\emptyset$	q_1
q_1	q_1	q_2
q_2	q_3	$\emptyset$
q_3	q_3	q_1

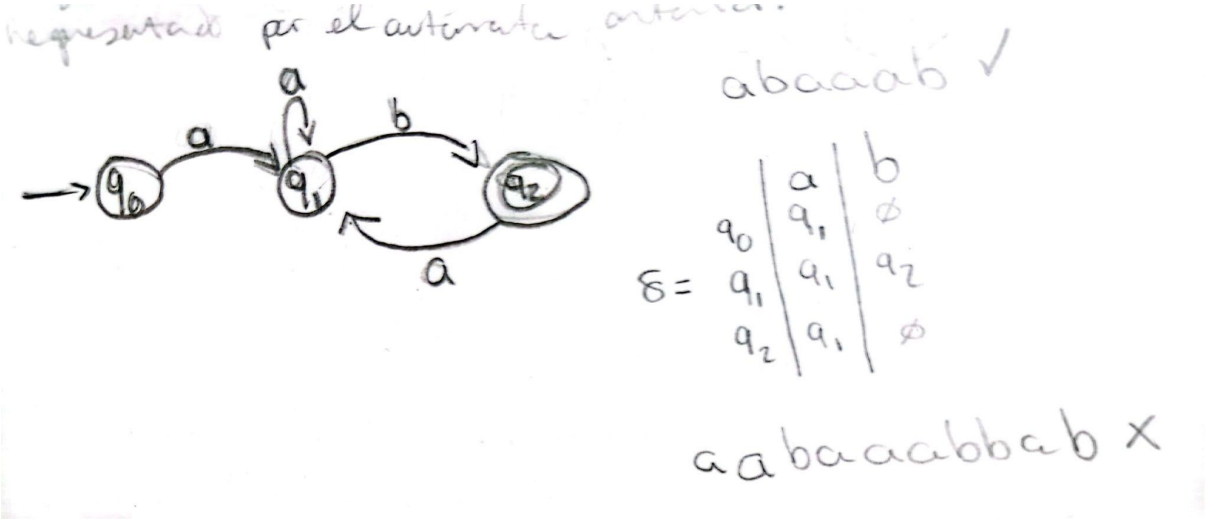
II.IX. Construya un autómata que reconozca el lenguaje formado por las cadenas de 0 y 1 con un número par de ceros. Implemente una función en pseudocódigo que reciba una cadena como parámetro y, usando el mecanismo de estados del autómata, determine si pertenece al lenguaje.




```
/*
==================================================
Name:   Héctor Miranda García
ID:     A01658845
Date:   03/03/2022
==================================================
This code is intended to represent the functioning
of a DFA. In this case, the problem proposes a DFA
that recognizes all strings composed of 0's and 1's
, and so that there is an even number of 0's.
==================================================
*/

#include <iostream>
#include <string>

#define Q0  0
#define Q1  1

bool check_dfa(std::string cadena, int cadena_length){
        int state = Q0;
        for(int i = 0; i < cadena_length; i++){
            switch(state){
                case Q0:
                    if(cadena[i] == '1') state = Q0;
                    else if(cadena[i] == '0') state = Q1;
                break;

                case Q1:
                    if(cadena[i] == '1') state = Q1;
                    else if(cadena[i] == '0') state = Q0;
                break;
            }
```

```
        }
        if(state == Q0) return true;
        return false;
};

int main(){

        int state = Q0;
        std::string cadena = "10100011100";
        int cadena_length = cadena.length();

        bool result = check_dfa(cadena, cadena_length);

        if(result == true) std::cout << "Compatible string: " << cadena
<< "\n";
        else std::cout << "Incompatible string:" << cadena << std::endl;

    return 0;
}
```

II.X. Construya un autómata finito determinista que reconozca todas las cadenas que comiencen con una a, terminen en una b y no contengan nunca dos b consecutivas. Escriba un programa que realice el proceso de reconocimiento representado por el autómata anterior.

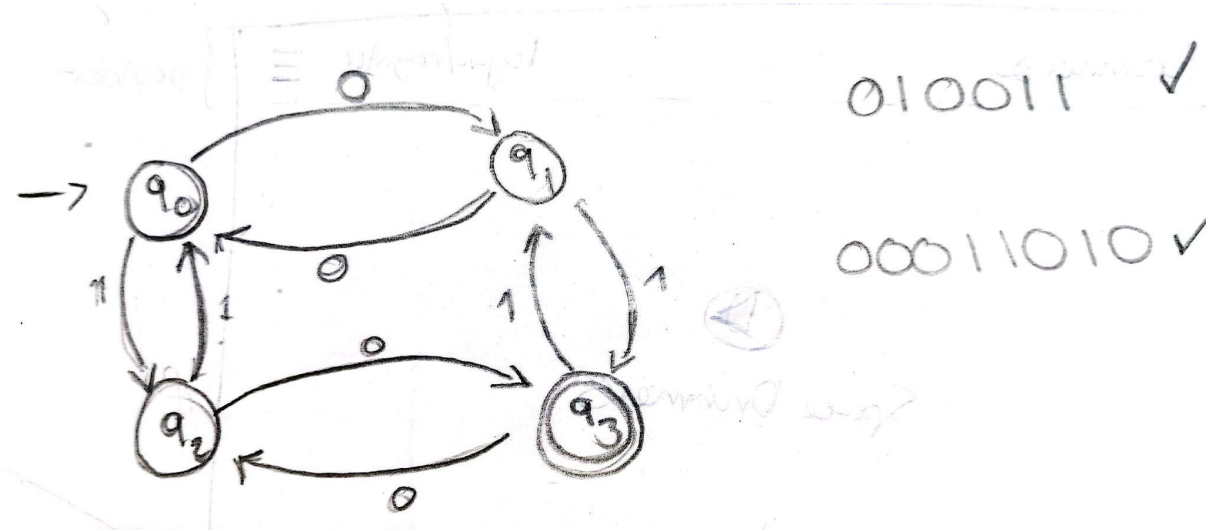



```
/*
==============================================
Name:    Héctor Miranda García
```

```
ID:     A01658845
Date:   03/03/2022
================================================
This code is intended to represent the functioning
of a DFA. In this case, the problem proposes a DFA
that recognizes strings that start widh an 'a' and
ending with a 'b'; also it should not contain two
consecutive b's.
================================================
*/
#include <iostream>
#include <string>


#define Q0      0
#define Q1      1
#define Q2      2
#define ERROR   3


bool check_dfa(std::string cadena, int cadena_length){
    int the_state = Q0;
    for(int i = 0; i < cadena_length; i++){
        switch(the_state){
            case Q0:
                if(cadena[i] == 'a') the_state = Q1;
                else the_state = ERROR;
            break;

            case Q1:
                if(cadena[i] == 'a') the_state = Q1;
                else if(cadena[i] == 'b') the_state = Q2;
            break;

            case Q2:
                if(cadena[i] == 'a') the_state = Q1;
              else the_state = ERROR;
            break;

        }
    }
    if(the_state==Q2) return true;
    return false;
};
```

```
int main(){

    std::string cadena = "abaabab";
    int cadena_length = cadena.length();

    bool result = check_dfa(cadena, cadena_length);

    if(result == true) std::cout << "Compatible string: " << cadena << "\n";
    else std::cout << "Incompatible string: " << cadena << "\n";
    return 0;

};
```

II.XI. Construya un autómata finito determinista que reconozca todas las cadenas que contengan una cantidad impar de 0 's y una cantidad impar de 1'w. Escriba un programa que realice el proceso de reconocimiento representado por el autómata.




```
/*
==================================================
Name:   Héctor Miranda García
ID:     A01658845
Date:   03/03/2022
==================================================
This code is intended to represent the functioning
of a DFA. In this case, the problem proposes a DFA
that recognizes strings that contain an odd quantity
of 0's and 1's
==================================================
*/
```

```cpp
#include <iostream>
#include <string>

#define Q0      0
#define Q1      1
#define Q2      2
#define Q3      3

bool check_dfa(std::string cadena, int cadena_length){
    int the_state = Q0;
    for(int i = 0; i < cadena_length; i++){
        switch(the_state){

            case Q0:
                if(cadena[i]=='0') the_state=Q1;
                else if(cadena[i] == '1')the_state=Q2;
            break;

            case Q1:
                if(cadena[i] == '0') the_state = Q0;
                else if(cadena[i] == '1') the_state = Q3;
            break;

            case Q2:
                if(cadena[i] == '0') the_state = Q3;
                else if(cadena[i] == '1') the_state = Q0;
            break;

            case Q3:
                if(cadena[i] == '0') the_state = Q2;
                else if(cadena[i] == '1') the_state = Q1;
            break;

        }

    }
    if(the_state == Q3) return true;
    return false;
};

int main(){
    std::string cadena = "010011";
    int cadena_length = cadena.length();
    bool result = check_dfa(cadena, cadena_length);
```

```
    if(result == true) std::cout << "Compatible string: " << cadena <<
"\n";
    else std::cout << "Incompatible string: " << cadena << "\n";

    return 0;
}
```