

**Práctica. AFD y AFND**

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE  
HIDALGO  
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

**LIC. CIENCIAS COMPUTACIONALES**

**AUTOMATAS Y COMPILADORES**

ALUMNO:

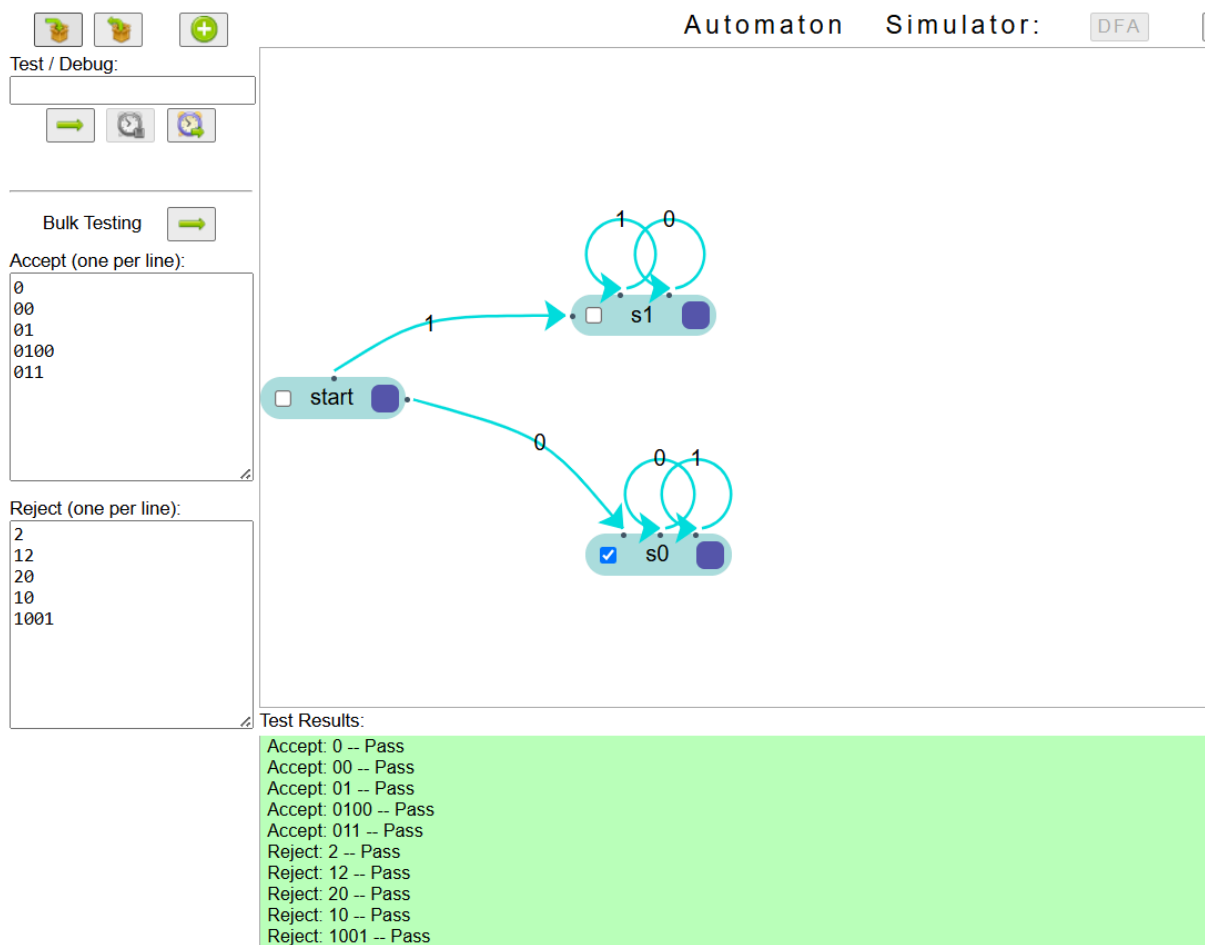
MARCO YAHIR BALTAZAR GARCIA

PROFESOR:

EDUARDO CORNEJO VELAZQUEZ

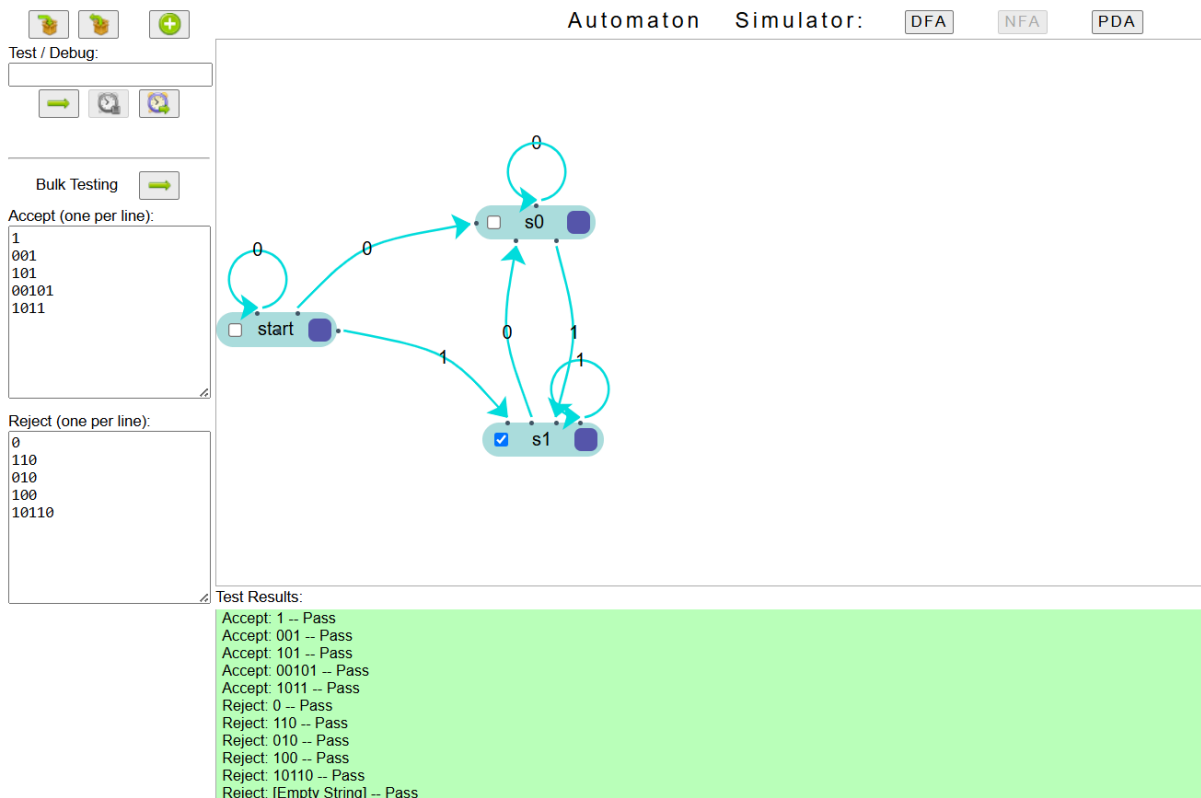
Ejercicio 1. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{0, 1\}$ , que acepte el conjunto de palabras que inician en "0".

$F = \{s0\}$   
 $\Sigma = \{0, 1\}$   
 $Q = \{\text{Start}, s0, s1\}$   
 $q_0 = \text{start}$   
 $f(\text{start}, 0) = s0$   
 $f(s0, 0) = s0$   
 $f(s0, 1) = s1$   
 $f(\text{start}, 1) = s1$   
 $f(s1, 0) = s1$   
 $f(s1, 1) = s1$



Ejercicio2. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{0, 1\}$ , que acepte el conjunto de palabras que terminan en "1".

$F=\{s1\}$   
 $\Sigma= \{0, 1\}$   
 $Q=\{start, s0, s1\}$   
 $q0= start$   
 $f(start,0)=s0$   
 $f(start,1)=s1$   
 $f(s0,0)$   
 $f(s0,1)=s1$   
 $f(s1,0)=s0$   
 $f(s1,1)=s1$



Ejercicio 3. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma= \{0, 1\}$ , que acepte el conjunto de palabras que contienen la subcadena "01".

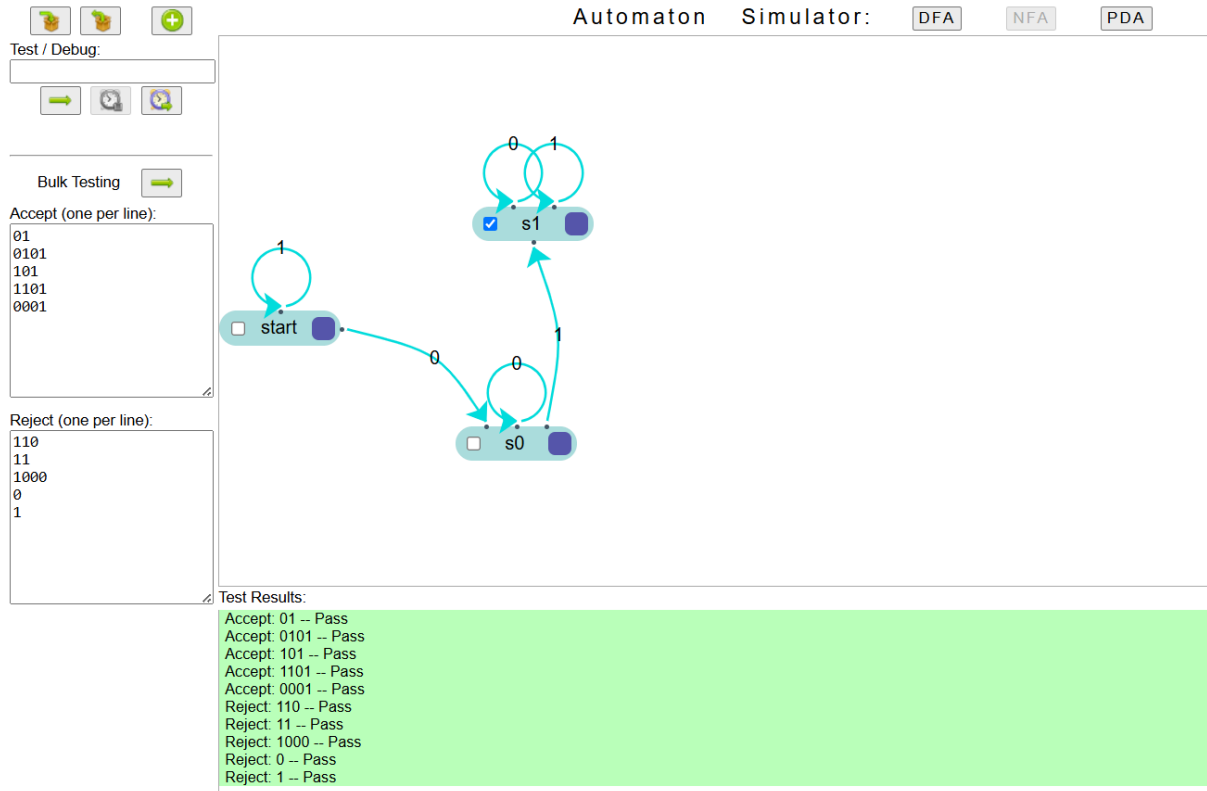
$F=\{s1\}$   
 $\Sigma= \{0, 1\}$   
 $Q=\{Start, s0,s1\}$   
 $q0= start$   
 $f(start,0)=s0$   
 $f(start,1)= start$

$f(s0,0)=s0$

$f(s0,1)=s1$

$f(s1,0)=s1$

$f(s1,1)=s1$



Ejercicio 4. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{0, 1\}$ , que acepte el conjunto de palabras que no contienen la subcadena "01".

$F = \{\text{start}, s0\}$

$\Sigma = \{0, 1\}$

$Q = \{\text{Start}, s0, s1\}$

$q_0 = \text{start}$

$f(\text{start}, 1) = \text{start}$

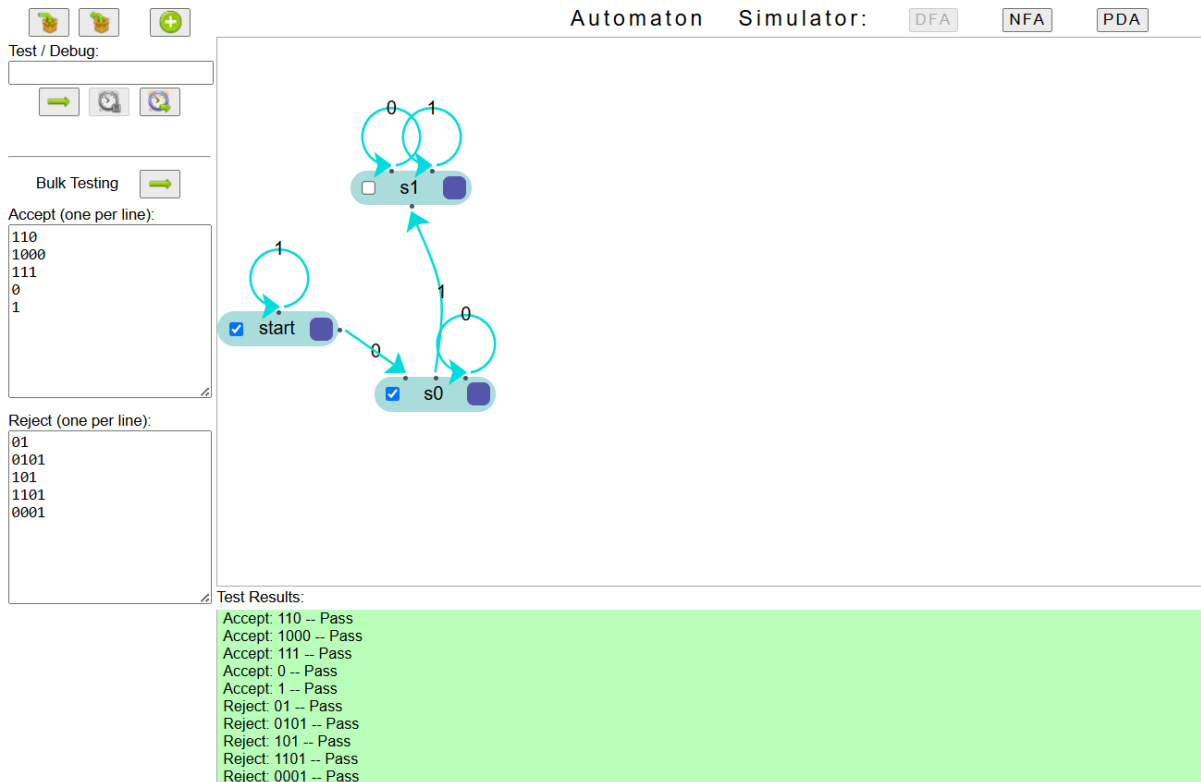
$f(\text{start}, 0) = s0$

$f(s0, 0) = s0$

$f(s0, 1) = s1$

$f(s1, 0) = s1$

$f(s1, 1) = s1$



Ejercicio 5. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ , que acepte el conjunto de palabras que inician con la subcadena “ac” o terminan con la subcadena “ab”.

$F = \{s2, s4\}$

$\Sigma = \{a, b, c\}$

$Q = \{\text{Start}, s1, s2, s3, s4, s5\}$

$q0 = \text{start}$

$f(\text{start}, a) = s1$

$f(\text{start}, b) = s5$

$f(\text{start}, c) = s5$

$f(s1, a) = s3$

$f(s1, b) = s4$

$f(s1, c) = s2$

$f(s2, a) = s2$

$f(s2, b) = s2$

$f(s2, c) = s2$

$f(s3, a) = s3$

$f(s3, b) = s4$

$f(s3, c) = s5$

$f(s4,a)=s3$

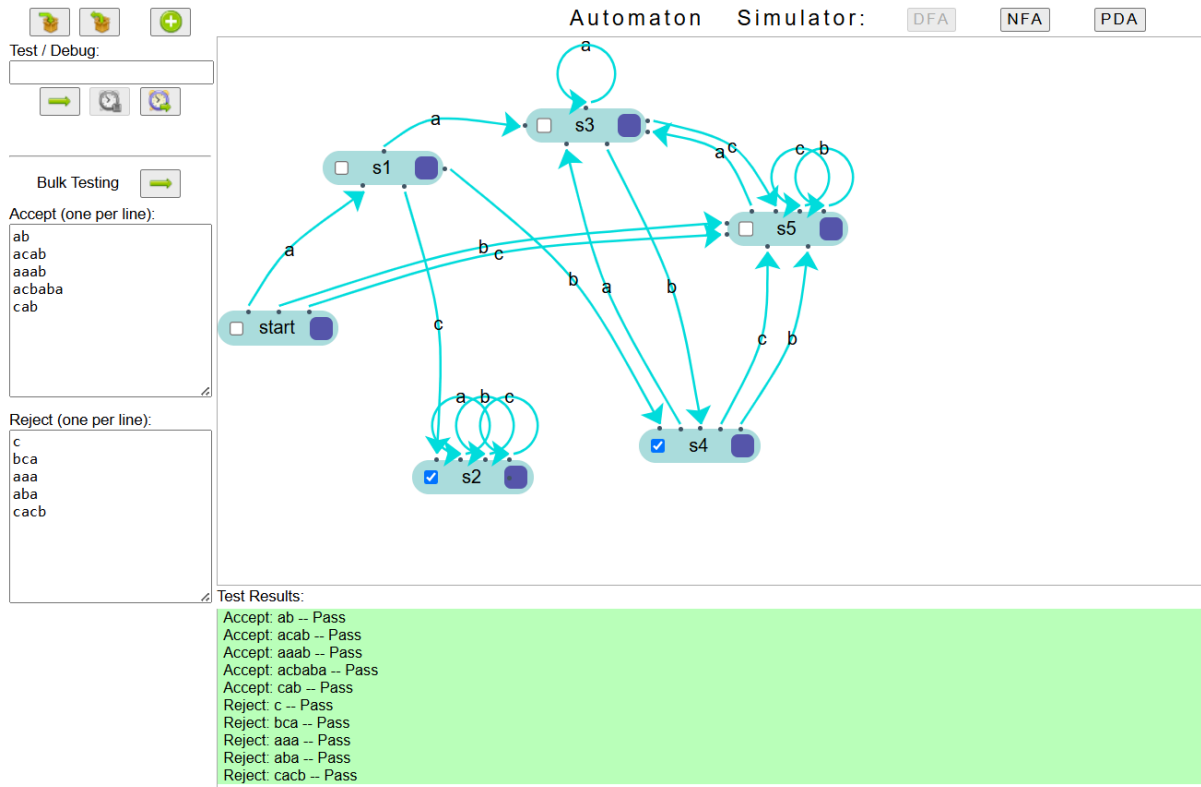
$f(s4,b)=s5$

$f(s4,c)=s5$

$f(s5,a)=s3$

$f(s5,b)=s5$

$f(s5,c)=s5$



Ejercicio 6. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ , que acepte el conjunto de palabras que inician con la subcadena “ac” y no terminan con la subcadena “ab”.

$F = \{s1, s2, s3\}$

$\Sigma = \{a, b, c\}$

$Q = \{\text{Start}, s0, s1, s2, s3, s4, s5\}$

$q0 = \text{start}$

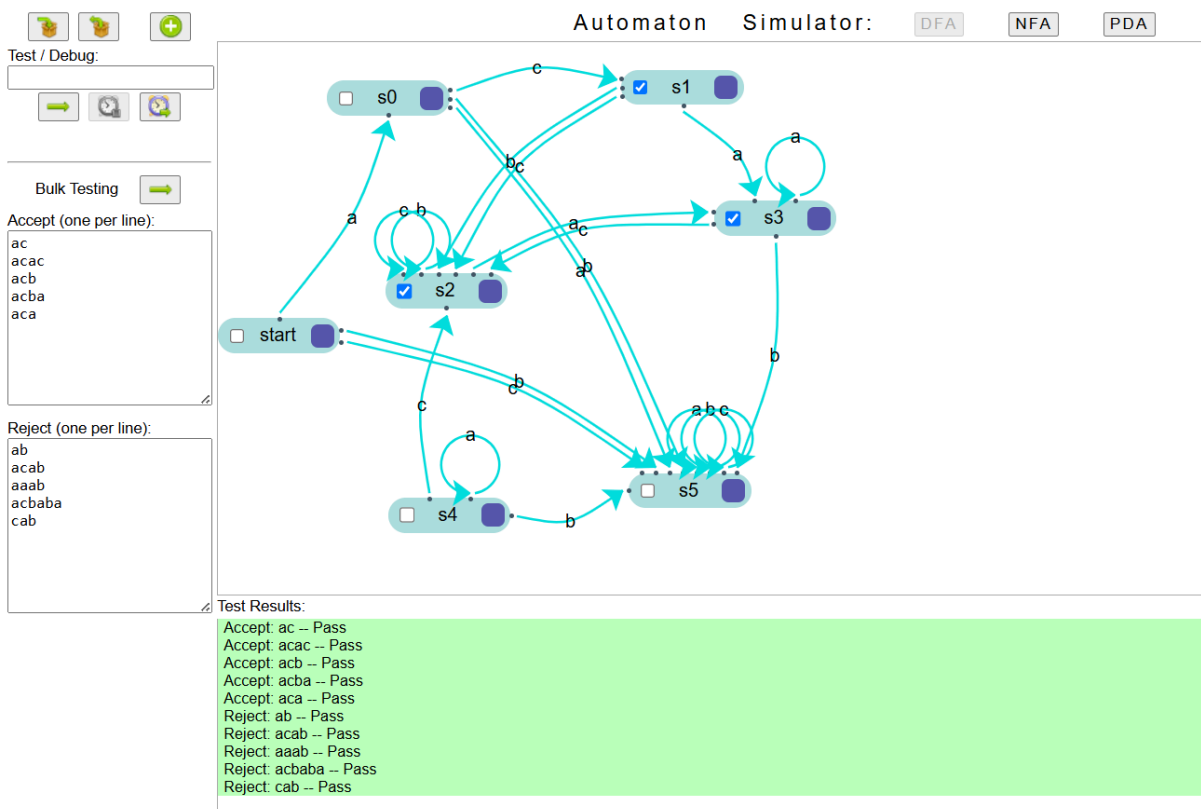
$f(\text{start}, a) = s0$

$f(\text{start}, b) = s5$

$f(\text{start}, c) = s5$

$f(s0, a) = s5$

$f(s0, b) = s5$   
 $f(s0, c) = s1$   
 $f(s1, a) = s3$   
 $f(s1, b) = s2$   
 $f(s1, c) = s2$   
 $f(s2, a) = s3$   
 $f(s2, b) = s2$   
 $f(s2, c) = s2$   
 $f(s3, a) = s3$   
 $f(s3, b) = s5$   
 $f(s3, c) = s2$   
 $f(s4, a) = s4$   
 $f(s4, b) = s5$   
 $f(s4, c) = s2$   
 $f(s5, a) = s5$   
 $f(s5, b) = s5$   
 $f(s5, c) = s5$



Ejercicio 7. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ , que acepte el conjunto de

palabras que inician con la subcadena "ac" o no terminan con la subcadena "ab".

$F = \{\text{start}, s_0, s_2, s_3, s_4\}$

$\Sigma = \{a, b, c\}$

$Q = \{\text{Start}, s_0, s_1, s_2, s_3, s_4\}$

$q_0 = \text{start}$

$f(\text{start}, a) = s_0$

$f(\text{start}, b) = \text{start}$

$f(\text{start}, c) = \text{start}$

$f(s_0, a) = s_0$

$f(s_0, b) = s_1$

$f(s_0, c) = s_2$

$f(s_1, a) = s_0$

$f(s_1, b) = \text{start}$

$f(s_1, c) = \text{start}$

$f(s_2, a) = s_3$

$f(s_2, b) = s_2$

$f(s_2, c) = s_2$

$f(s_3, a) = s_3$

$f(s_3, b) = s_4$

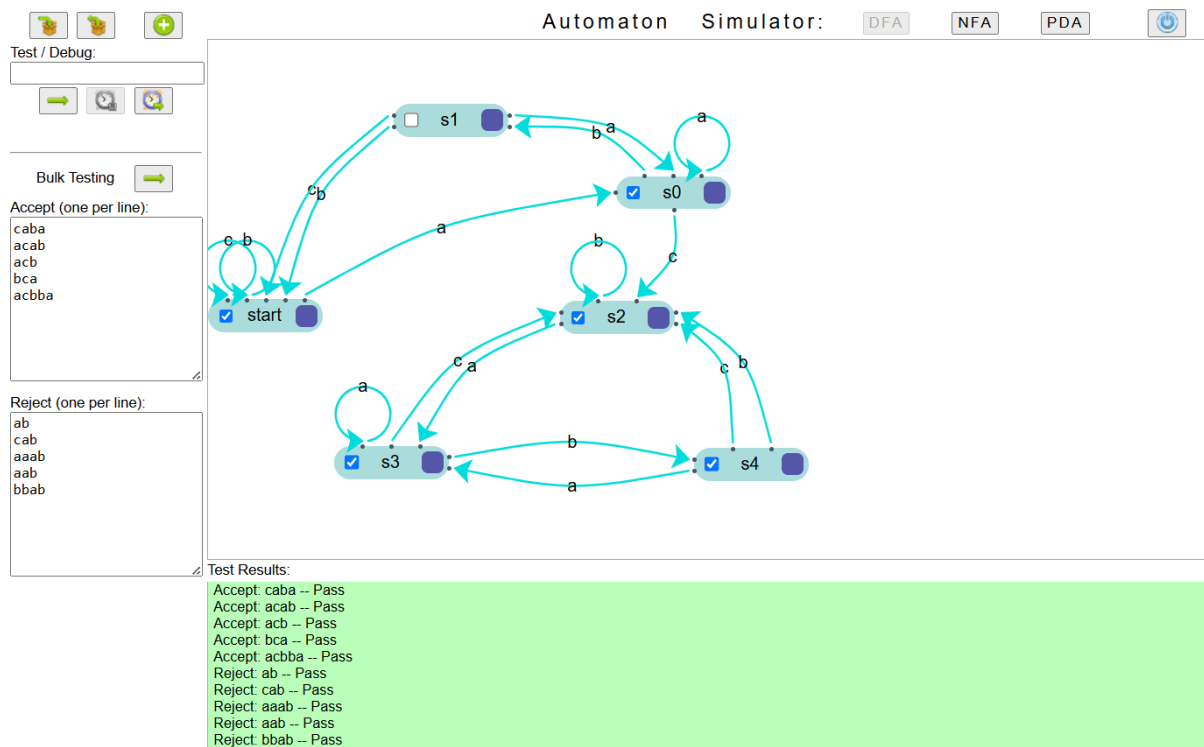
$f(s_3, c) = s_2$

$f(s_4, a) = s_3$

$f(s_4, b) = s_2$

$f(s_4, c) = s_2$





Ejercicio 8. Obtenga un Autómata Finito Determinista (AFD) dado el lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ , que acepte el conjunto de palabras que no inician con la subcadena “ac” y no terminan con la subcadena “ab”.

$F = \{\text{start}, s0, s1, s2\}$

$\Sigma = \{a, b, c\}$

$Q = \{\text{start}, s0, s1, s2, s3, s4\}$

$q_0 = \text{start}$

$f(\text{start}, a) = s0$

$f(\text{start}, b) = s1$

$f(\text{start}, c) = s1$

$f(s0, a) = s2$

$f(s0, b) = s3$

$f(s0, c) = s4$

$f(s1, a) = s2$

$f(s1, b) = s1$

$f(s1, c) = s1$

$f(s2, a) = s2$

$f(s2, b) = s3$

$f(s2, c) = s1$

$f(s3, a) = s2$

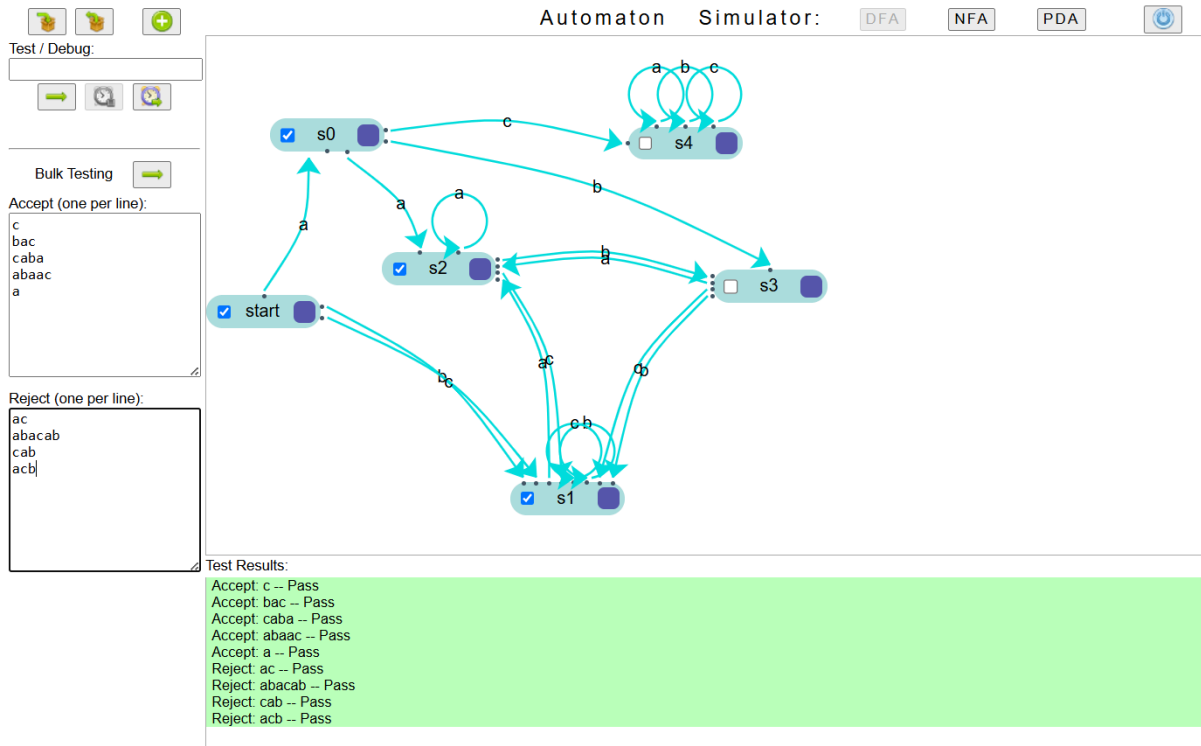
$f(s3, b) = s1$

$f(s3, c) = s1$

$f(s4, a) = s4$

$f(s4, b) = s4$

$f(s4, c) = s4$



Ejercicio 9. Obtenga un Autómata Finito No Determinista (AFND) dado el lenguaje definido en el alfabeto  $\Sigma = \{0, 1\}$ , que acepte el conjunto de palabras que no contienen a la subcadena "01".

$F = \{\text{start}, s0\}$

$\Sigma = \{0, 1\}$

$Q = \{\text{start}, s0, s1\}$

$q0 = \text{start}$

$f(\text{start}, 1) = \text{start}$

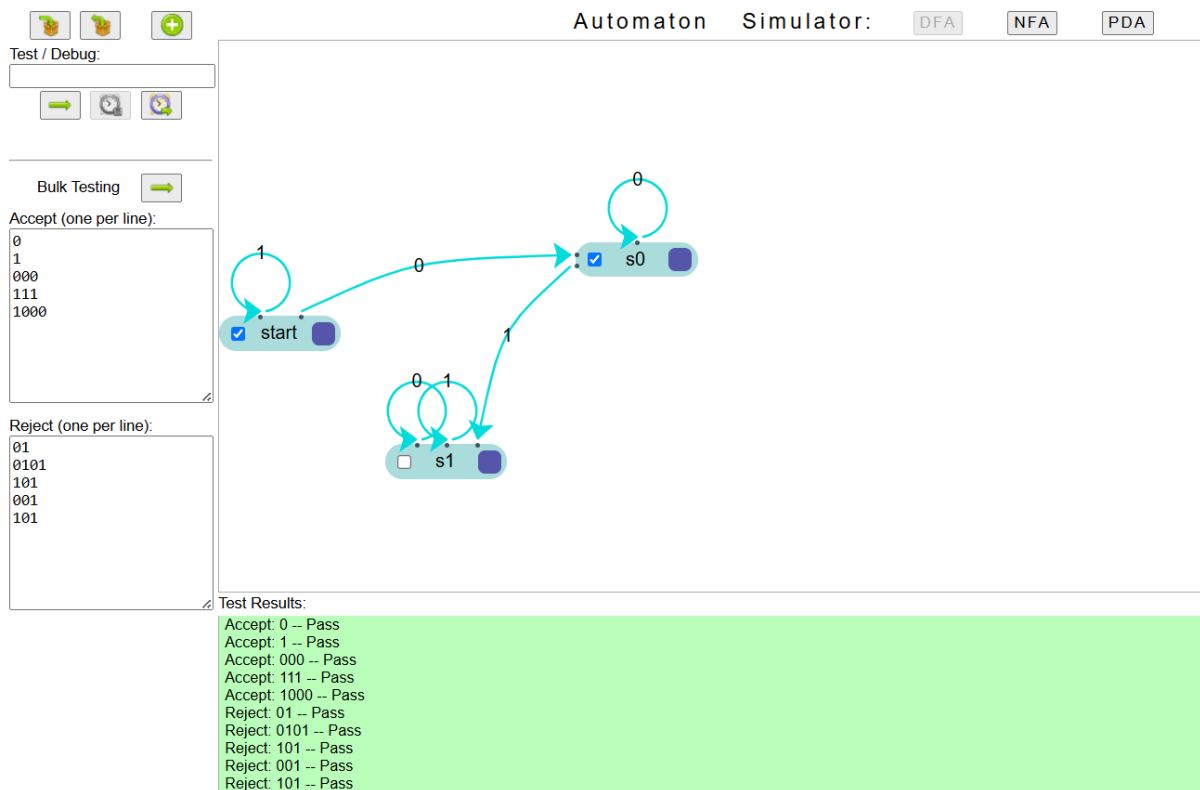
$f(\text{start}, 0) = s0$

$f(s0, 0) = s0$

$f(s0, 1) = s1$

$f(s1, 0) = s1$

$f(s1, 1) = s1$



Ejercicio 10. Obtenga un Autómata Finito No Determinista (AFND) dado el lenguaje definido en el alfabeto  $\Sigma = \{a, b, c\}$ , que acepte el conjunto de palabras que inician en la subcadena “ac” y terminan en la subcadena “ab”.

$F = \{s3\}$

$\Sigma = \{a, b, c\}$

$Q = \{start, s0, s1, s2, s3\}$

$q0 = start$

$f(start, a) = s0$

$f(s0, c) = s1$

$f(s1, a) = s1$

$f(s1, a) = s2$

$f(s1, b) = s1$

$f(s1, c) = s1$

$f(s2, b) = s3$

**Automaton Simulator:**
DFA
NFA
PDA

**Test / Debug:**

---

**Bulk Testing**

**Accept (one per line):**

```

acab
acabab
acacacab
acbcbab
          
```

**Reject (one per line):**

```

ab
ac
acb
cab
acaba
          
```

```

graph LR
    start((start)) -- a --> s0((s0))
    s0 -- c --> s1((s1))
    s1 -- c --> s0
    s1 -- b --> s1
    s1 -- a --> s2((s2))
    s2 -- b --> s3(((s3)))
          
```

**Test Results:**

```

Accept: acab -- Pass
Accept: acabab -- Pass
Accept: acacacab -- Pass
Accept: acbcbab -- Pass
Reject: ab -- Pass
Reject: ac -- Pass
Reject: acb -- Pass
Reject: cab -- Pass
Reject: acaba -- Pass
          
```