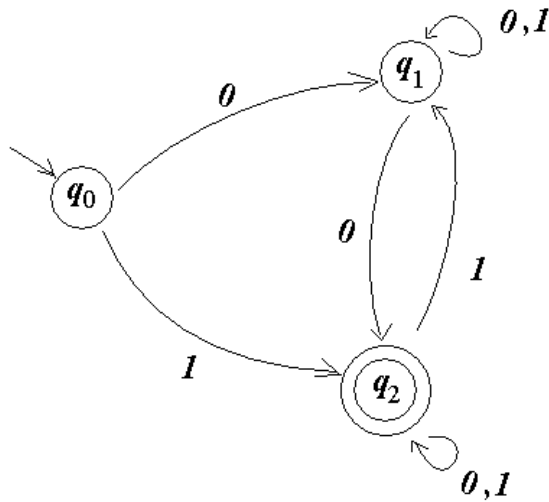
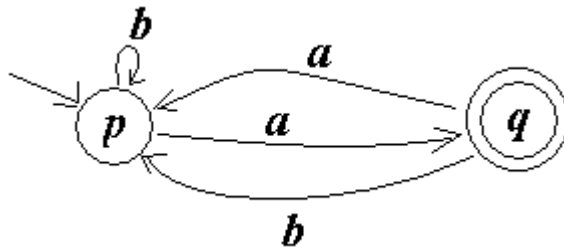


Automate finite

1.1 Probleme cu AF

1. Sa se reprezinte tabelar urmatoarele automate finite.
Sint ele AFD sau AFN?



2. Sa se reprezinte sub forma de graf automatul finit: $M=(Q,\Sigma, \delta,q_0,F)$

	0	1	
q0	q2	q1	1
q1	q3	q0	0
q2	q0	q3	0
q3	q1	q2	0

Verificati apoi, bazandu-va pe graful obtinut, ca:

- a) secventele 1010, 1100 sunt acceptate de automat
- b) secventa 1011 nu este acceptata de automat

3. Sa se construiasca un AF care accepta

a) $L = \{ a a a \}$, $\Sigma = \{ a \}$

b) $L = \{ w_1 a a w_2 \mid w_1, w_2 \in \{ a, b \}^* \}$

c) cuvinte peste alfabetul $\{0,1\}$ cu proprietatea ca:

orice cuvânt al limbajului conține cel puțin 2 zerouri consecutive

d) cuvinte peste alfabetul $\{a, b, c\}$ cu proprietatea ca:

primul simbol al cuvântului este același cu cel cu care se termină cuvântul

e) cuvinte peste alfabetul $\{a, b, c\}$ cu proprietatea ca:

există un simbol în cuvânt care mai apare cel puțin o dată în cuvânt

f) $L = \{ c^{3n}, n \in \mathbb{N}^* \}$

g) limbajul ce conține secvențe peste $\Sigma = \{a, b\}$ cu nr. par de simb. **a** și nr. par de simb **b**

h) limbajul ce conține secv. peste $\Sigma = \{a, b\}$ cu nr. impar de simb. **a** și impar de simb **b**

i) $L = \{ 1^n 0^m 1 u \mid n \geq 0, m \geq 1, u \in \{0,1\}^* \}$

j) $L = \{ 0 (10)^n 01^m \mid n \geq 0, m \geq 0 \} \cup \{ (10)^n 01^m \mid n \geq 1, m \geq 0 \}$ cu cel mult 4 stări

k) $L = \{ 0^m 1^n \mid m, n \in \mathbb{N} \} \cup \{ 1^p 0^q \mid p, q \in \mathbb{N} \}$

l) $L = \{ w_1 a a w_2 \mid w_1 \in \{b, ab\}^*, w_2 \in \{a, b\}^* \}$

1.2 Structuri de date pentru automate finite

Descriți o modalitate de reprezentare pentru AF

(gândiți câteva modalități)

ex:

- AF care are ca alfabet mulțimea caracterelor reprezentabile în calculator (alfabet fixat)

- structura de date care are în vedere ca operația cea mai frecventă cu AF

este verificare acceptare secvență

```
StateMachine
  description: String          /** A description of the state machine */
  startState :State
  states: Set<State>           /** Can be List, Map, ... */
                               /** What about redundancy? */
```

```
State
  description: String          /** The name of the state */
  isAcceptState: Boolean
  transitions: MultiMap<Symbol, State> /** Can be List, Map ... ? */
```