**Exercice1**

Soit l'automate fini déterministe suivant :

- Q = {q0, q1, q2} (ensemble des états),

- Σ= {0, 1} alphabet,

-δ(fonction de transition) définie par :

* δ(q0, 0) = q1,
* δ (q0, 1) = q0,
* δ (q1, 0) = q1,
* δ (q1, 1) = q2,
* δ (q2, 0) = q2,
* δ (q2, 1) = q2,

- q0  est l'état initial, et

- F ={q2} est l'ensemble des états acceptants.

1. Tracer cette automate

2. Donne des exemples de chaînes acceptées par cet automate.

3. Quelle est la propriété que satisfait une chaîne pour être acceptée ?

**Exercice2 :**

**Problème** : Créer un automate fini qui reconnaît toutes les chaînes binaires contenant le sous-mot 101 au moins une fois.

* **Alphabet** : Σ={0,1}
* **Objectif** : Concevoir un NFA pour ce langage, puis comparer avec un DFA.

**Exemple de Fonctionnement du NFA**

Pour la chaîne 1101010 :

**Exercice3 :**

Conçois un automate fini déterministe (DFA) qui reconnaît les chaînes de l’ensemble L = {a, b} contenant un nombre pair de `a`.

**Exercice4 :**

Concevons un automate fini déterministe (DFA) pour vérifier si une chaîne est un nom de variable valide en langage C.

**Règles de Nommage des Variables en C**

En C, un nom de variable valide doit respecter les règles suivantes :

1. Il doit commencer par une lettre (majuscule ou minuscule) ou un underscore (\_).
2. Les caractères suivants peuvent être des lettres, des chiffres ou des underscores (\_).
3. Un nom de variable ne peut pas être un mot réservé du langage (ici, nous ne traiterons pas cette contrainte pour simplifier l’exercice).

**Exemples**

* Chaînes valides : var1, \_temp, data\_123
* Chaînes invalides : 1var, #temp, var-2

**Exercice 5:**

Soit l'ensemble des entiers multiples de '10'

1- Donner une expression régulière qui valide ces mots.

2- Donner l'automate à états finis déterministe correspondant.

3- Donner une grammaire régulière équivalente.

**Exercice 6 :**

Soit l'ensemble des entiers multiples de '5'

1- Donner une expression régulière qui valide ces mots.

2- Donner l'automate à états finis déterministe correspondant.

3- Donner une grammaire régulière équivalente.

**Exercice7 :**

* + Donner un AFD qui accepte les mots spécifiés par l'expression régulière : (01|10)+

**Exercice8 :**

* + Donner un AFD qui accepte les mots spécifiés par l'expression régulière : (01|10)+

(0|1)+ 10 (1|0)+

**Exercice 9 :**

* + Donner un AFD qui accepte les mots sur l'alphabet {0,1} qui ne contiennent pas la chaîne : "011"