

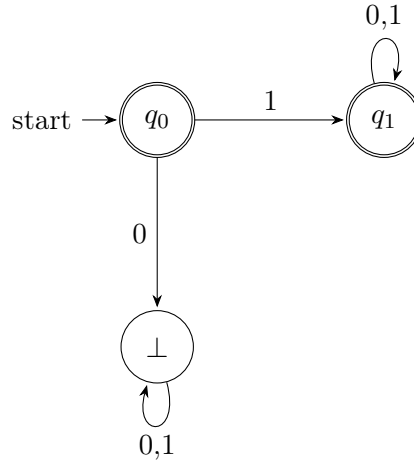
On note $A = \{0, 1\}$ et ε le mot vide. En OCaml, 0 est représenté par `false` et 1 par `true`.

1. $L_1 = \{\varepsilon\} \cup 1A^*$: il s'agit des représentations binaires sans zéro initial, le nombre nul étant représenté par ε .

(a) $41 = 32 + 8 + 1 = (101001)_2$.

(b) $(10101010)_2 = 128 + 32 + 8 + 2 = 170$.

(c) Un automate déterministe complet reconnaissant L_1 est le suivant. L'état initial q_0 est final pour accepter ε ; q_1 est final; $\perp$ est un état puits.



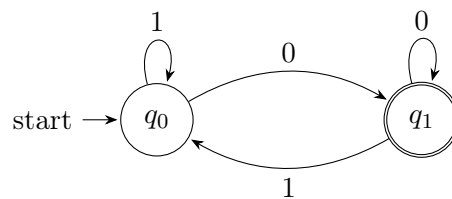
(d) Il suffit de tester si le mot est vide ou commence par `true`.

```

let langage_1 m =
  match m with
  | [] -> true
  | true :: _ -> true
  | false :: _ -> false
  
```

2. $L_2 = A^*0$ est l'ensemble des mots non vides se terminant par 0.

(a) On utilise deux états : q_0 (initial, non final : mot vide ou dernier symbole 1) et q_1 (final : dernier symbole 0).



(b) On peut tester la dernière lettre en parcourant la liste.

```

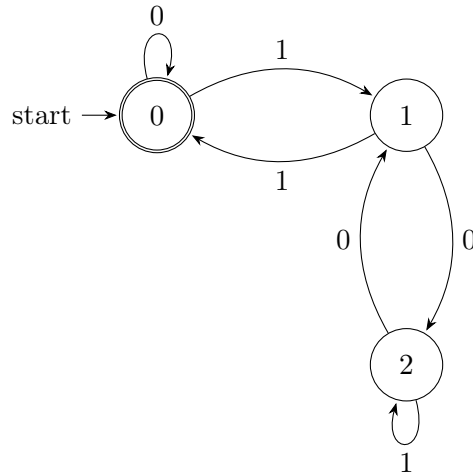
let langage_2 m =
  let rec finit_par_zero = function
    | [] -> false
    | [false] -> true
    | [_] -> false
    | _ :: suite -> finit_par_zero suite
  in
  finit_par_zero m
  
```

3. L'automate $\mathcal{A}_3$ a pour états $\{0, 1, 2\}$, état initial et final 0, et transition $\delta(q, a) = (2q + a) \bmod 3$.

(a) La table des transitions est

q	$\delta(q, 0)$	$\delta(q, 1)$
0	0	1
1	2	0
2	1	2

D'où l'automate :



(b) On effectue la lecture dans l'automate en lisant le mot et on vérifie qu'on termine bien dans l'état 0.

```

let langage_3 m =
  let rec aux e m = match m with
    | [] -> e=0
    | a::n -> aux (2*e+(if a then 1 else 0) mod 3) n
  in aux 0 m

```

(c) Montrons par récurrence sur $n \geq 1$ que

$$\widehat{\delta}(0, \omega_1 \cdots \omega_n) = \left(\sum_{k=1}^n \omega_k 2^{n-k} \right) \bmod 3.$$

Initialisation : pour $n = 1$, $\widehat{\delta}(0, \omega_1) = \delta(0, \omega_1) = \omega_1 \bmod 3$, ce qui est bien la somme annoncée.

Hérédité : supposons l'égalité vraie au rang n . Pour $\omega_{n+1} \in A$, par définition de la transition étendue,

$$\begin{aligned}
 \widehat{\delta}(0, \omega_1 \cdots \omega_n \omega_{n+1}) &= \delta(\widehat{\delta}(0, \omega_1 \cdots \omega_n), \omega_{n+1}) \\
 &= \left(2\widehat{\delta}(0, \omega_1 \cdots \omega_n) + \omega_{n+1} \right) \bmod 3 \\
 &= \left(2 \sum_{k=1}^n \omega_k 2^{n-k} + \omega_{n+1} \right) \bmod 3 \\
 &= \left(\sum_{k=1}^{n+1} \omega_k 2^{n+1-k} \right) \bmod 3.
 \end{aligned}$$

La propriété est donc vraie pour tout $n \geq 1$.

Interprétation : L_3 contient exactement les mots dont la valeur binaire est divisible par 3, ainsi que ε (accepté par l'état initial final).

4. (a) L_4 est l'ensemble des écritures binaires canoniques d'entiers strictement positifs, pairs et multiples de 3, c'est-à-dire des multiples strictement positifs de 6 :

$$L_4 = \{\text{bin}(6k) \mid k \in \mathbb{N}^*\}.$$

Ici $\text{bin}(m)$ désigne l'écriture binaire usuelle de m sans zéro initial. Par exemple 110 (valeur 6) et 1100 (valeur 12) appartiennent à L_4 .

- (b) **Oui.** Les langages L_1 , L_2 et L_3 sont reconnaissables par des automates finis. La classe des langages reconnaissables est stable par intersection. On peut construire l'automate produit de $\mathcal{A}_1$, $\mathcal{A}_2$ et $\mathcal{A}_3$, dont les états sont des triplets et dont les états finaux sont les triplets composés d'états finaux des trois automates. Il reconnaît exactement L_4 .