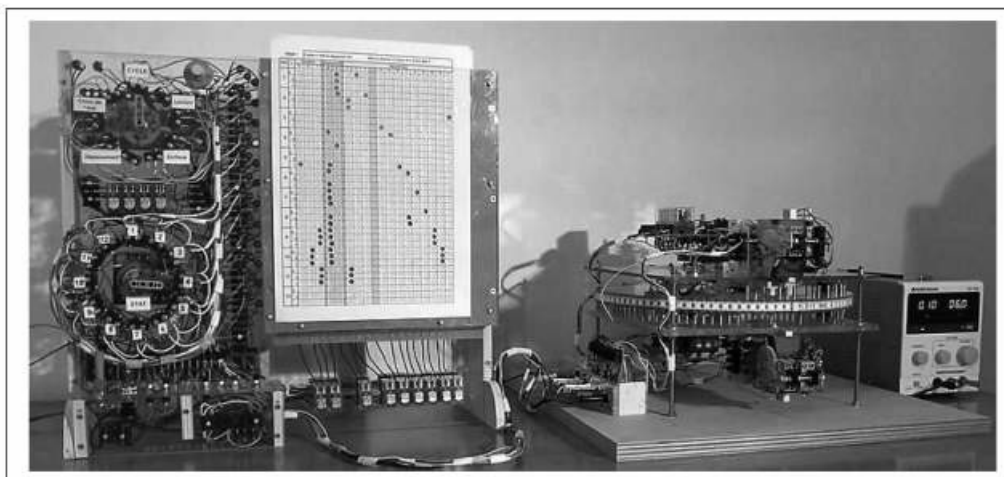
	<b>Algorithmique</b>	<b>NSI T<sup>ale</sup></b>
	Programmer avec une machine de Turing	TP

\*

## 1. Introduction



Descriptifs techniques, diagrammes et vidéos sur le site : [www.machinedeturing.org](http://www.machinedeturing.org)

Dans ce TP est présenté un prototype de la machine imaginée par Alan Turing pour modéliser le concept d'algorithme et donner une réponse négative au problème de la décision en logique du premier ordre posé quelques années auparavant par Hilbert et Ackermann.

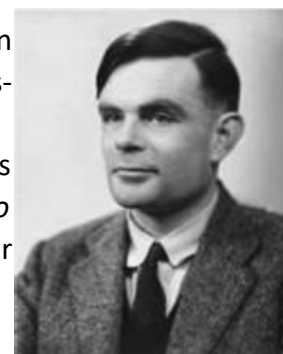
Pour simplifier, est-il possible de trouver une méthode « effectivement calculable » pour décider si une proposition est démontrable ou non ? Si ce problème mathématique est difficile, le fonctionnement de sa machine est abordable par un large public et vous trouverez ci-dessous de nombreux exemples.

Alan Turing (1912-1954) est un mathématicien britannique. Il obtient sa licence en 1934 au King's Collège de Cambridge. Il passe ensuite sa thèse à Princeton aux États-Unis.

Le prototype présenté ici est directement issu de la description qu'il a donnée dans son article publié en 1936 et intitulé : *On Computable Numbers, with an Application to the Entscheidungsproblem* (problème de la décision). Vous trouverez le texte initial sur le site :

[www.machinedeturing.org](http://www.machinedeturing.org) onglet *machine de Turing*.

Alan Turing n'a jamais construit cette machine.



Dans le prototype, on a ajouté un lecteur de feuilles perforées qui permet de le programmer facilement. L'intérêt de ce prototype est qu'il permet de mettre en œuvre un très grand nombre d'algorithmes dont on peut immédiatement visualiser le fonctionnement.

C'est aussi une belle illustration du génie de ce grand mathématicien.

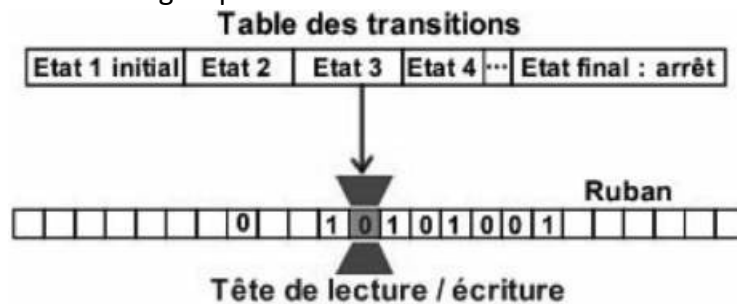
## 2. Description théorique de la machine

Un ruban infini comporte des cases qui peuvent contenir des symboles ; une tête d'écriture et de lecture peut se déplacer sur ce ruban.

La tête peut lire le symbole déjà écrit, en écrire un autre et se déplacer d'une case à gauche ou d'une case à droite.

Les **actions à exécuter** (écriture, déplacement et choix d'un état) sont inscrites dans ce qu'on appelle les **états**.

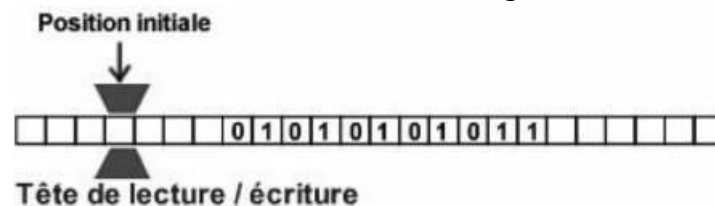
L'ensemble des états possibles est regroupé dans la **table des transitions**.



## 3. Exemple détaillé

Trouver une séquence et en remplacer les 0 par des 1

Une séquence de 0 et de 1 est écrite sur le ruban et la tête est à gauche de cette séquence.



Pour cet algorithme, on utilise deux états :

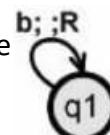
- À l'état 1
  - la tête devra se déplacer à droite tant qu'elle lira un blanc, dès qu'elle trouve un 0 ou un 1 elle passe à l'état 2.
- À l'état 2
  - si elle lit un 0, elle écrit 1 et se déplace à droite ;
  - si elle lit un 1, elle n'écrit rien et se déplace à droite ;
  - enfin, si elle lit un blanc, c'est qu'elle a fini de parcourir la séquence, elle passe à l'état final et s'arrête.

On construit le diagramme de la machine de la façon suivante :

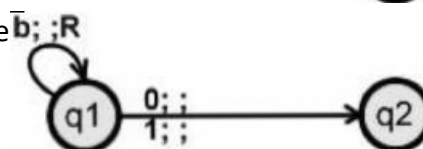
- Les états sont écrits sous la forme **q1**, **q2**, etc. et **qF** représente l'état final.
- Les déplacements à gauche et à droite sont représentés par les lettres **L** et **R** (Left et Right)
- Les triplets indiqués représentent (**lecture** ; **écriture** ; **déplacement**) et la flèche indique le prochain état.

### Dans l'état 1 (q1) :

Le triplet (**b ; ; R**) avec la flèche signifie que si la tête lit un blanc, elle n'écrit rien, elle se déplace d'une case à droite et elle reste à l'état 1.



Les triplets (**0 ; ;**) et (**1 ; ;**) signifient que si la tête lit un 0 ou un 1, elle n'écrit rien, elle ne se déplace pas et elle passe à l'état 2.

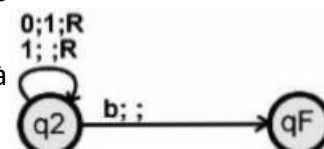


### Dans l'état 2 (q2) :

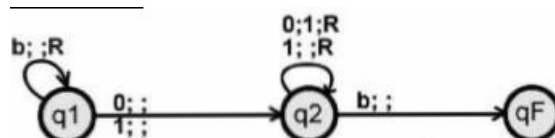
Le triplet (**0 ; 1 ; R**) signifie que si la tête lit un 0, elle écrit 1, elle se déplace à droite et elle reste à l'état 2.

Le triplet (**1 ; ; R**) signifie que si la tête lit un 1, elle n'écrit rien, elle se déplace à droite et elle reste à l'état 2.

Le triplet (**b ; ;**) signifie que si la tête lit un blanc, elle n'écrit rien, elle ne se déplace pas et elle passe à l'état final **qF**.



Le diagramme complet est le suivant :



Et voila le résultat :

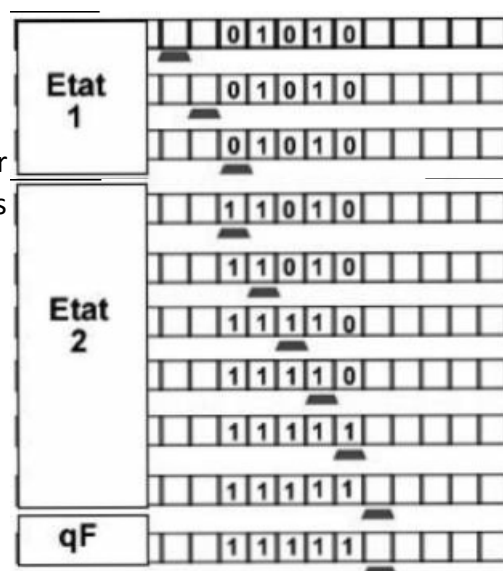


### Représentation de la machine en fonctionnement.

Le trapèze représente la tête de lecture/écriture.

On voit clairement ici que la tête de lecture ne fait que se déplacer de la gauche vers la droite (ce n'est pas le cas général) et que dans l'état 2 elle remplace les 0 par des 1.

L'état final **qF** provoque l'arrêt de la machine.



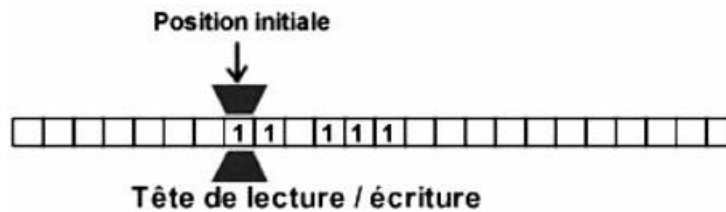
### Exercices :

- Ajouter un 1 à droite d'une séquence, la tête étant sous le chiffre de gauche.
- Remplacer, dans une séquence, tous les 1 par des 0 et tous les 0 par des 1.
- Écrire la suite infinie : 10101010... (il n'y a rien sur le ruban au départ).

## 4. Quelques diagrammes pour débiter

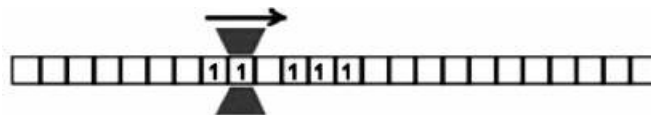
### 4.1. Faire une addition en unaire

Écrire un nombre en unaire revient à écrire autant de 1 que la valeur du nombre, exemple 5 s'écrit **11111**. Il faut additionner deux nombres représentés par des **1** (dans cet exemple 2 et 3), et séparés par un espace blanc.

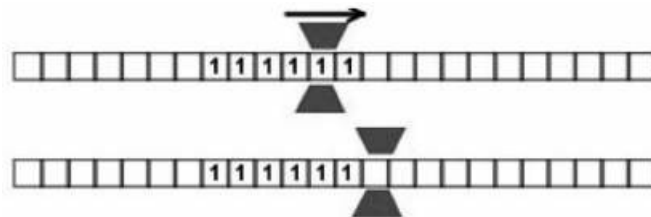


Pour cela la tête de lecture va parcourir le premier nombre, arrivée sur le blanc, elle va écrire un **1**, puis elle va parcourir le deuxième nombre, arrivée au blanc elle va reculer d'une case et va terminer en mettant à blanc le **1** le plus à droite.

**État 1** : la tête se déplace vers la droite jusqu'à trouver un blanc



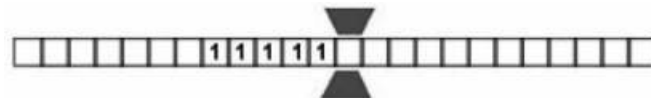
**État 2** : la tête se déplace de nouveau vers la droite jusqu'à trouver un blanc



Quand elle a trouvé un blanc, elle recule d'une case pour se mettre sur le **1** le plus à droite.



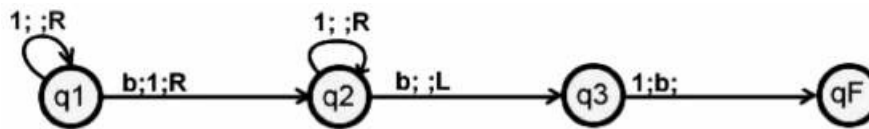
**État 3** : Elle écrit un blanc à la place du dernier 1 et passe à l'état final pour s'arrêter.



La table des transitions est la suivante :

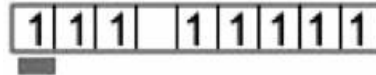
Etat	Lecture	Ecriture	Déplacement	Nouvel état
<b>q1</b>	b	1	R	<b>q2</b>
	0		R	
	1		R	
<b>q2</b>	b		L	<b>q3</b>
	0			
	1		R	
<b>q3</b>	b			Etat final qF
	0			
	1	b		

Et le diagramme correspondant



#### Exercice :

- Faire une soustraction en unaire : deux nombres X et Y sont écrits en unaire et séparés par un blanc, X est inférieur à Y et la tête est sous le chiffre de gauche de X. On pourra, successivement, remplacer un 1 de X par un 0 et supprimer un 1 de Y.

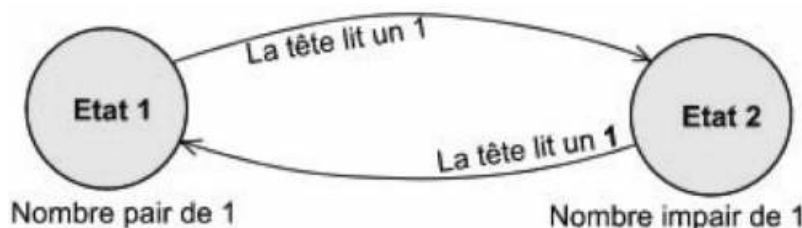


### 4.2. Nombre pair de 1

Une séquence de 0 et de 1 est écrite sur le ruban, la question posée à la machine est de savoir s'il y a un nombre pair de 1. Pour nous répondre oui, elle écrira un 1 à droite de la séquence sinon elle écrira un 0.



Pour cela la tête va parcourir la séquence en passant alternativement de l'état 1 à l'état 2 et réciproquement à chaque fois qu'elle rencontre un 1



Ainsi l'état 1 correspond à l'état dans lequel se trouve la machine si elle a lu un **nombre pair de 1**.

Dans les états 1 et 2, si la machine rencontre un 0, elle se déplace à droite et ne fait rien d'autre.

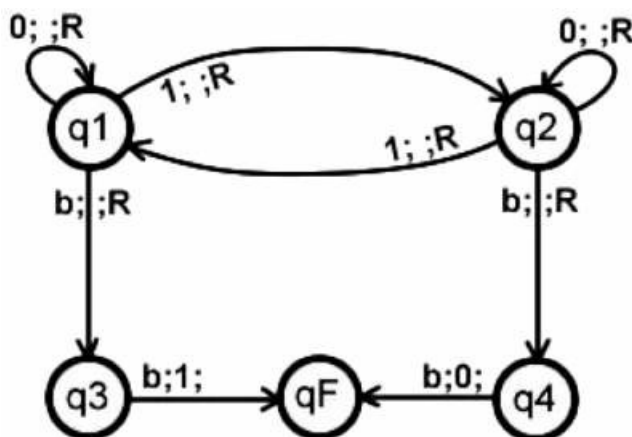
La machine « comprend » qu'elle a fini la lecture de la séquence quand elle rencontre le premier blanc.

- À l'état 3 (réponse OUI), elle écrit 1 sur un blanc puis s'arrête.
- À l'état 4 (réponse NON) elle écrit 0 sur un blanc puis s'arrête.

La table des transitions est la suivante :

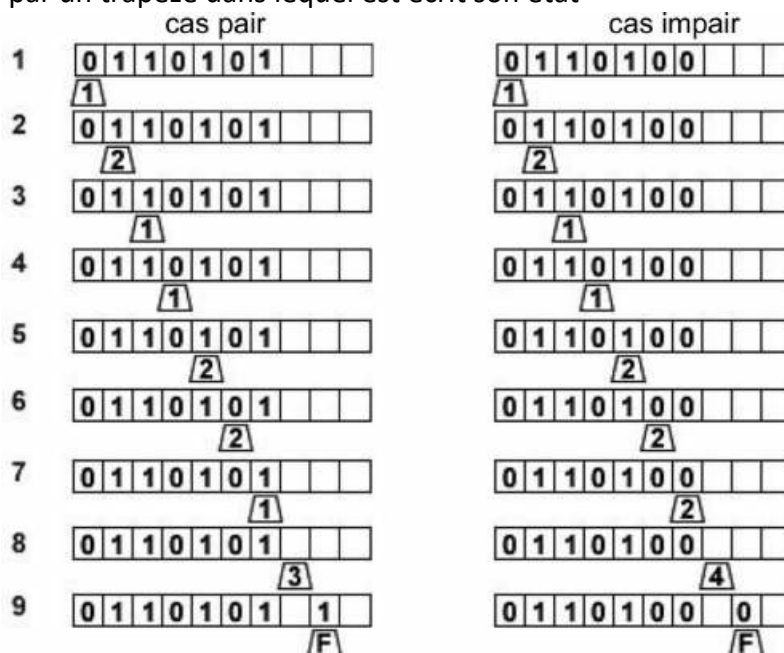
Etat	Lecture	Ecriture	Déplacement	Nouvel état
q1	b		R	q3
	0		R	
	1		R	q2
q2	b		R	q4
	0		R	
	1		R	q1
q3	b	1		Etat final qF
	0			
	1			
q4	b	0		Etat final qF
	0			
	1			

Et le diagramme correspondant :



### Représentation de la machine en fonctionnement

La tête est représentée par un trapèze dans lequel est écrit son état



Commentaires pour le cas pair :

- Ligne 2 : la tête lit un 1, elle passe à l'état 2
- Ligne 3 : la tête lit un autre 1, elle repasse à l'état 1
- Ligne 8 : la tête a terminé dans l'état 1, elle passe à l'état 3.
- Ligne 9 : il y a un nombre pair de 1, la tête écrit un 1 pour la réponse.

### Exercices :

- Écrire un 1 si une séquence contient un nombre pair de 0.
- Écrire un 1 si une séquence contient un nombre pair de 0 et un nombre pair de 1.

### 4.3. Doubler le nombre d'éléments

Une suite de 1 est écrite sur le ruban, la tête de lecture/écriture est située sous le 1 le plus à droite. La machine doit doubler le nombre de 1 déjà écrits.



## Comment faire ?

Quand la machine va lire un **1** à gauche, elle va le remplacer par un **0** et elle va se déplacer à droite pour écrire un autre **0**. Elle va continuer ainsi jusqu'à ce qu'il n'y ait plus de **1** à gauche. A ce moment, il y aura sur le ruban un nombre de **0** qui sera le double du nombre initial de **1**.

La dernière étape consiste simplement à remplacer tous ces **0** par des **1**.

## Fonctionnement :

Elle remplace le 1 de droite par un 0

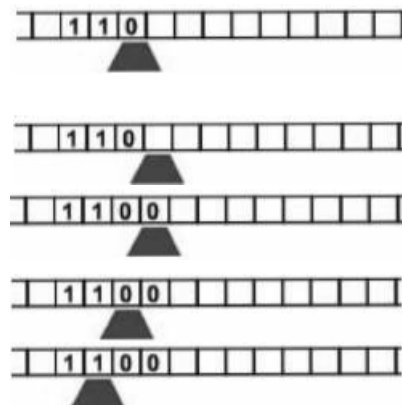
Elle se déplace à droite et elle passe à l'état 2.

Elle lit un blanc, elle écrit un nouveau 0

Elle se déplace à gauche et elle repasse à l'état 1

Elle se déplace sur les 0 vers la gauche jusqu'à trouver un 1.

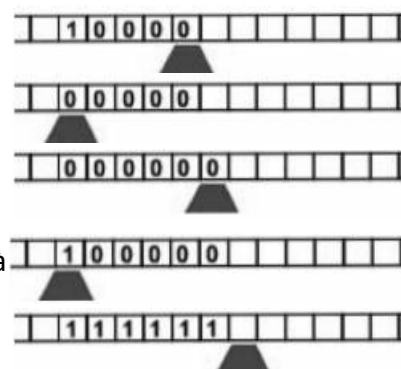
On recommence alors le cycle précédent.



Les principales étapes suivantes sont alors :

Ici, les trois 1 sont devenus six 0

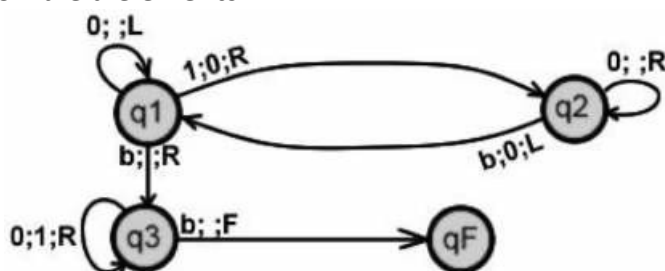
À l'état 3, elle remplace tous les 0 par des 1 et elle termine en passant à l'état final.



## Table des transitions :

Etat	Lecture	Ecriture	Déplacement	Nouvel état	Commentaire
q1	b		R	q3	- Il n'y a plus de 1
	0		L		- se déplace vers la gauche
	1	0	R	q2	- Remplace un 1 par un 0
q2	b	0	L	q1	Se déplacer sur les 0, trouver un blanc, écrire un nouveau 0 et passer à l'état 1
	0		R		
	1				
q3	b			Final	Remplacer tous les 0 par des 1 et terminer.
	0	1	R		
	1				

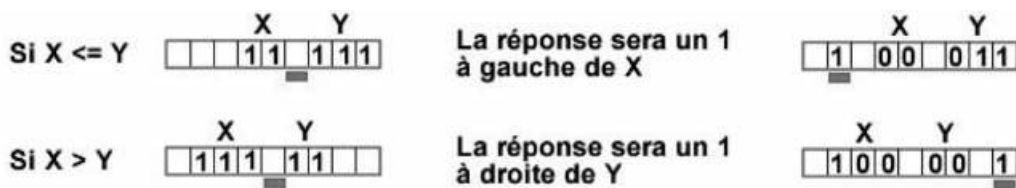
## Diagramme : « doubler le nombre d'éléments »



## Exercice :

- Tripler le nombre de 1.

#### 4.4. Comparaison de deux nombres écrits en unaire



La tête de lecture va remplacer les 1 par des 0 alternativement dans X et Y jusqu'à trouver un blanc, soit à gauche de X et dans ce cas on aura  $X \leq Y$ , soit à droite de Y et dans ce cas on aura  $X > Y$ .

##### État 1 :

- Se déplacer vers la gauche jusqu'à trouver un 1, le remplacer par un 0 et se déplacer à gauche.

##### État 2 :

- Si on lit un 1 se déplacer à droite et passer à l'état 3.
- Si on lit un blanc,  $X \leq Y$ , se déplacer à gauche et passer à l'état 5.

##### État 3 :

- Se déplacer vers la droite jusqu'à trouver un 1, le remplacer par un 0 et se déplacer à droite.

##### État 4 :

- Si on lit un 1, se déplacer à gauche et passer à l'état 1.
- Si on lit un blanc,  $X > Y$ , se déplacer à droite et passer à l'état 6.

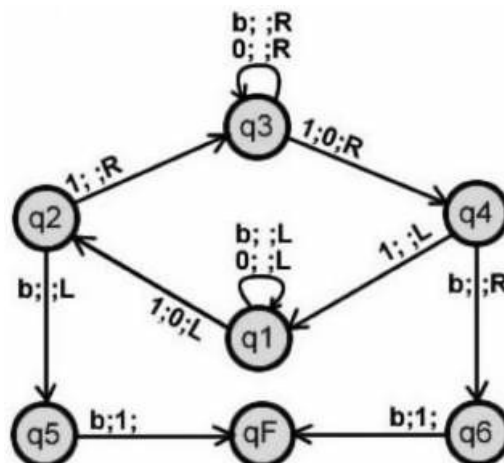
##### État 5 :

- $X \leq Y$ , écrire 1 et passer à l'état final.

##### État 6 :

- $X > Y$ , écrire 1 et passer à l'état final.

Le diagramme est le suivant



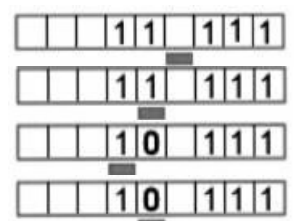
Les étapes dans le cas où  $X < Y$

##### État 1 :

- Lit blanc, va à gauche.
- Lit 1, écrit 0, va à gauche pour voir si il reste des 1 dans X et passe à l'état 2.

##### État 2 :

- Lit 1 (il reste des 1 dans X, va à droite, passe à l'état 3.





### État 3 :

- Lit 0, va à droite.
- Lit blanc, va à droite.
- Lit 1, écrit 0, va à droite pour voir s'il reste des 1 dans Y et passe à l'état 4.

### État 4 :

- Lit 1 (il reste des 1 dans Y), va à gauche et passe à l'état 1.

**État 1 :**

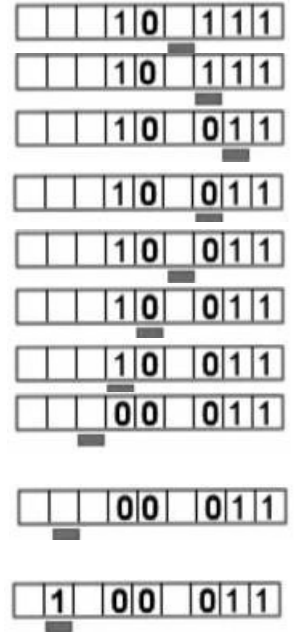
- Lit 0, va à gauche.
- Lit blanc, va à gauche.
- Lit 0, va à gauche.
- Lit 1, écrit 0, va à gauche pour voir s'il reste des 1 dans X et passe à l'état 2.

**État 2 :**

- Lit blanc : il ne reste pas de 1 dans X, donc  $X \leq Y$ .
- Va à gauche et passe à l'état 5.

### État 5 :

- Lit blanc, écrit 1 et passe à l'état final.



### Exercise :

- Écrire la table de transition pour ce diagramme