Chapitre 1 : Définitions

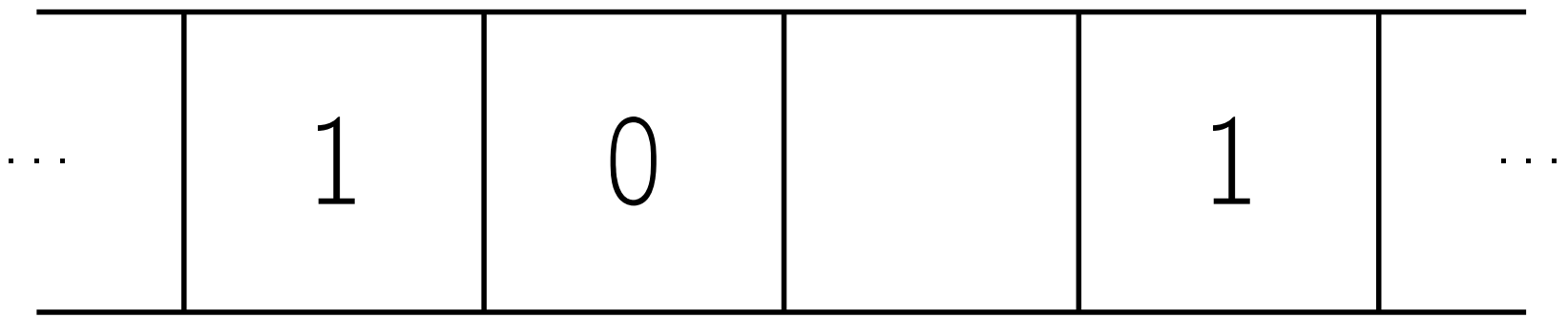
ENSIIE - Modèles de calculs - Machines de Turing

Dimitri Watel (dimitri.watel@ensiie.fr)

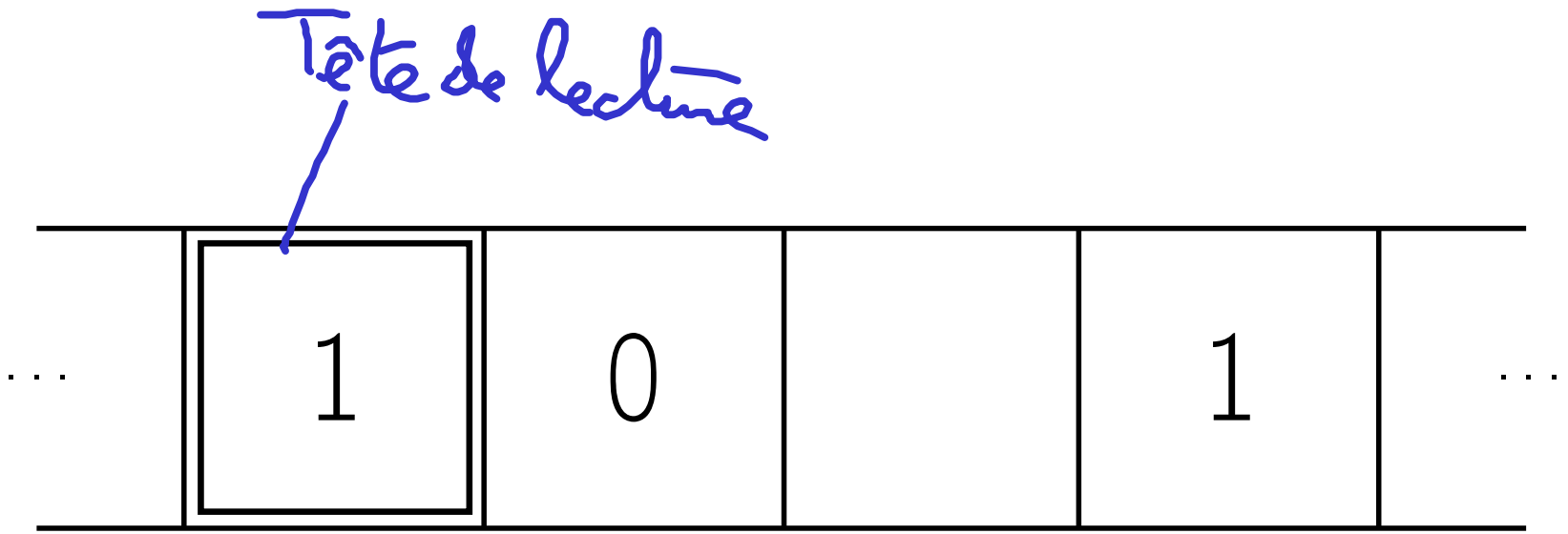
2018

Machine de Turing

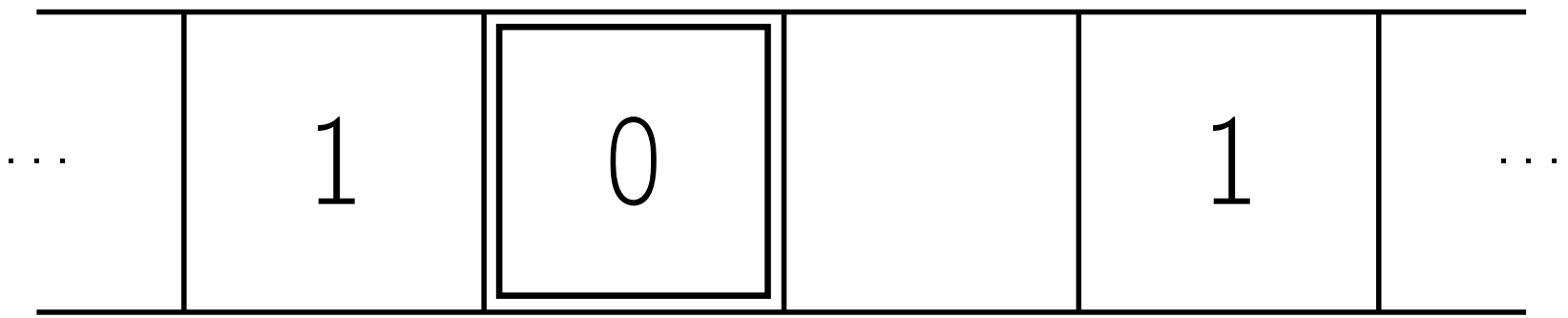
Machine de Turing - la bande



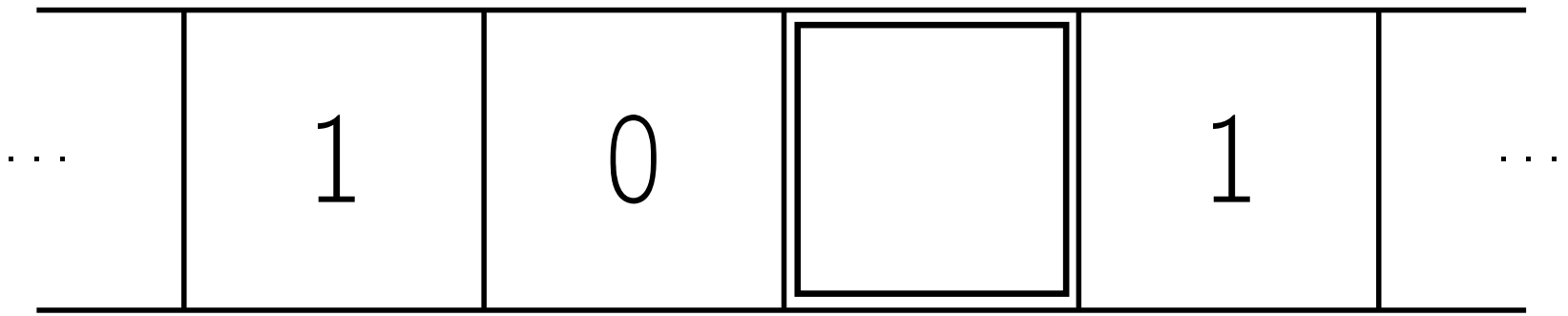
Machine de Turing - la tête



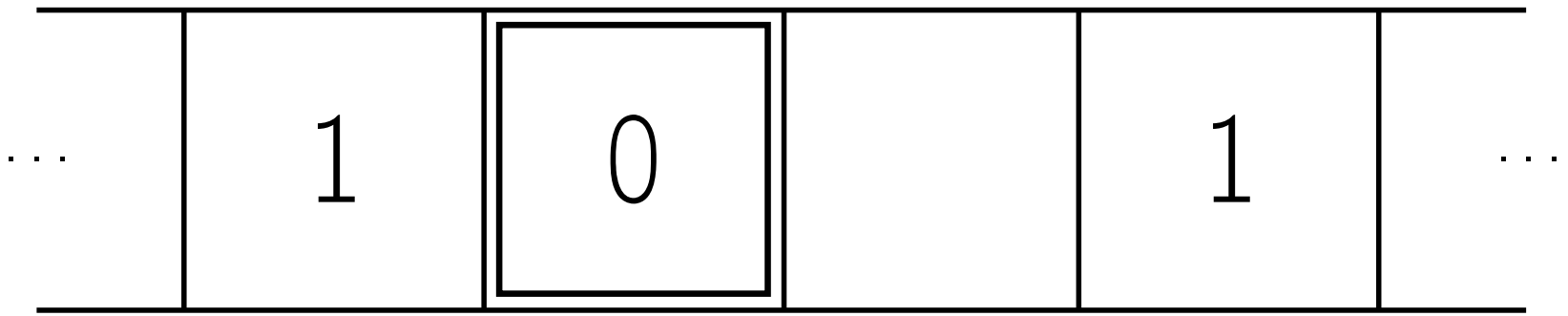
Machine de Turing - la tête



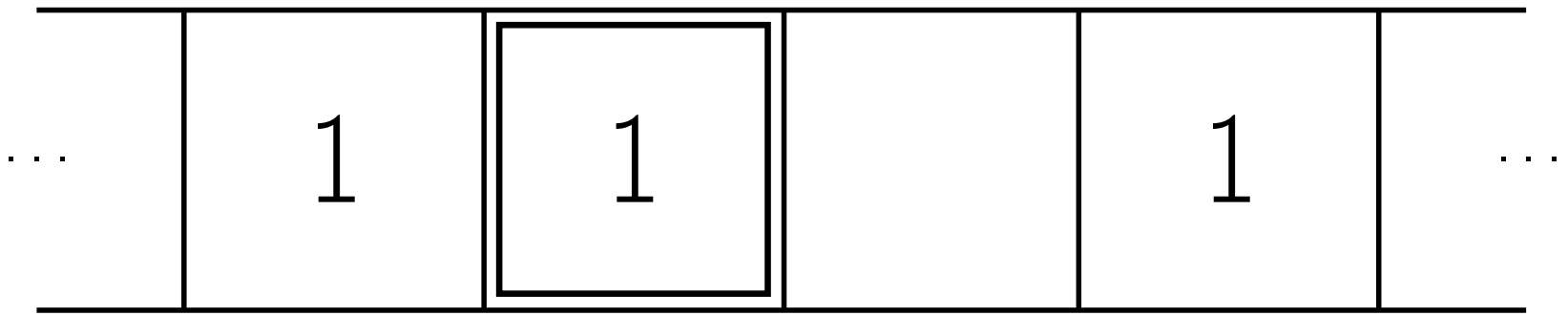
Machine de Turing - la tête



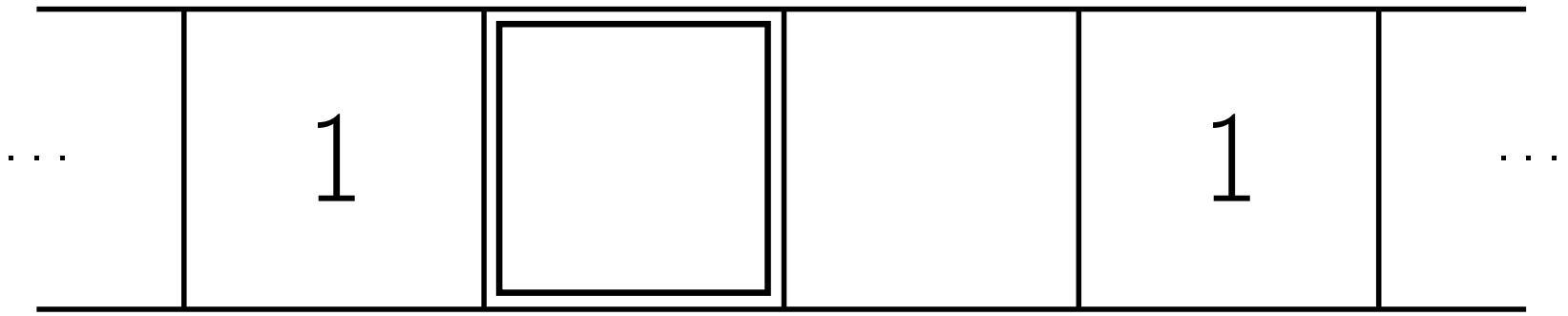
Machine de Turing - la tête



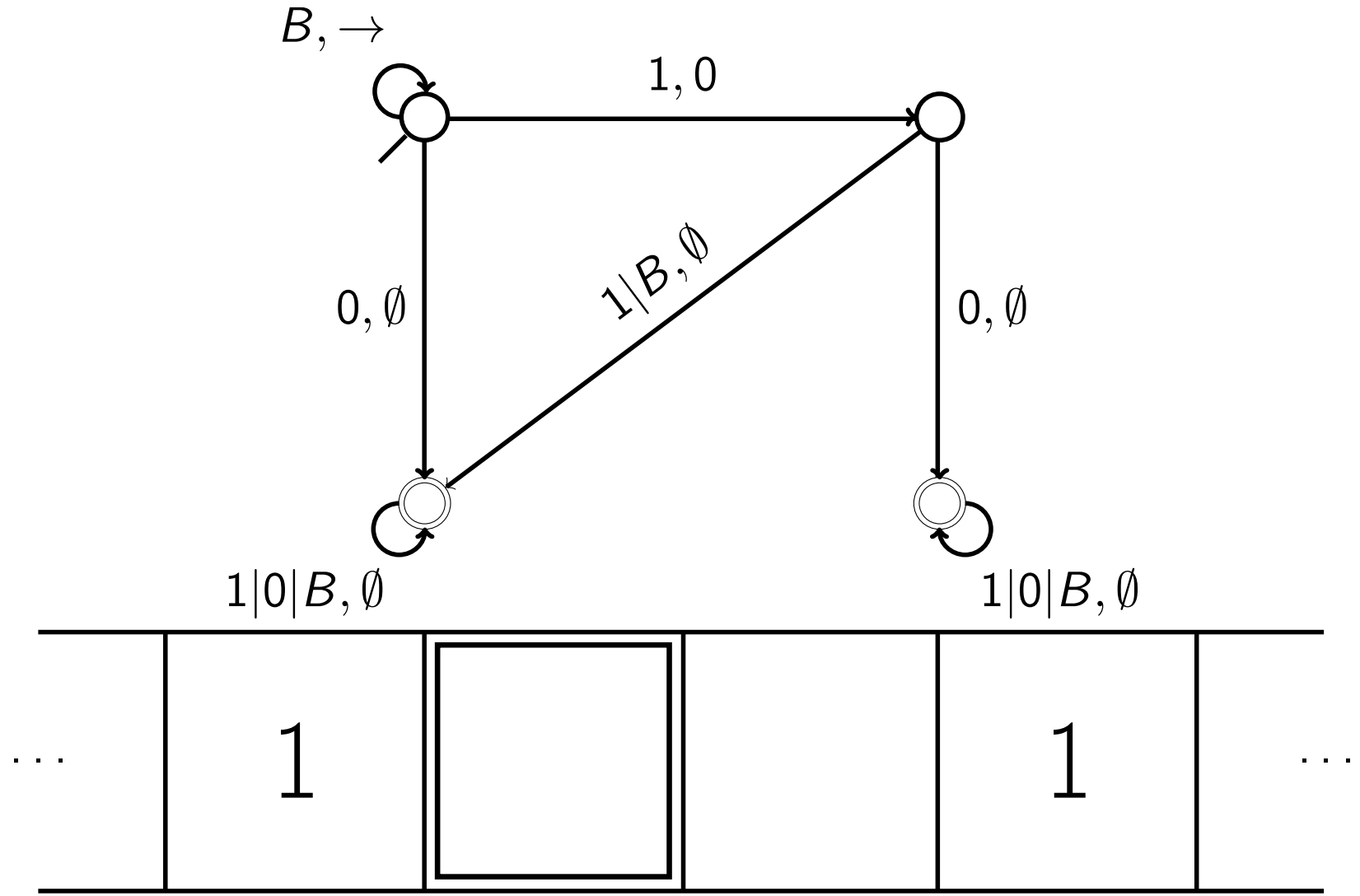
Machine de Turing - la tête



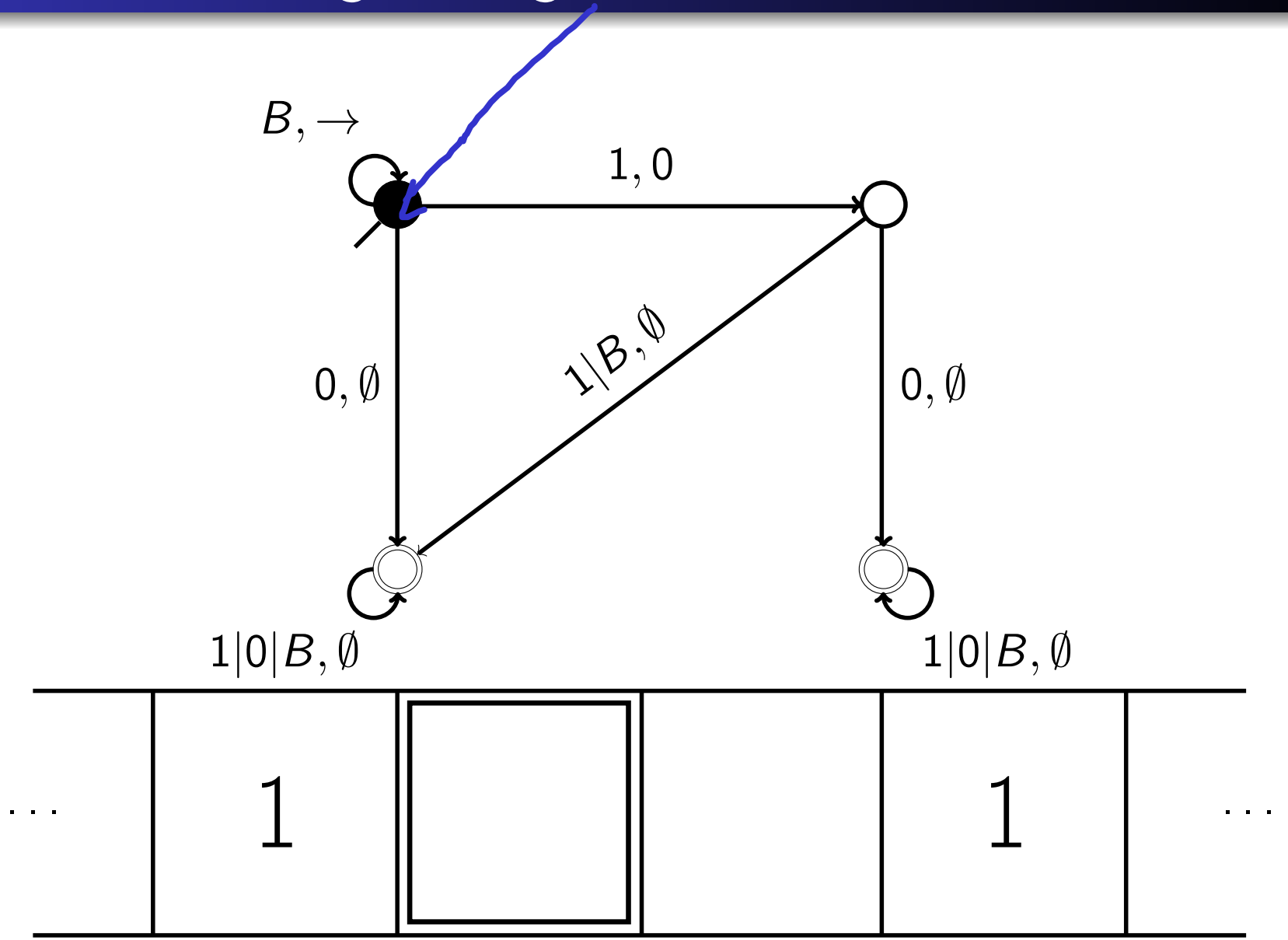
Machine de Turing - la tête



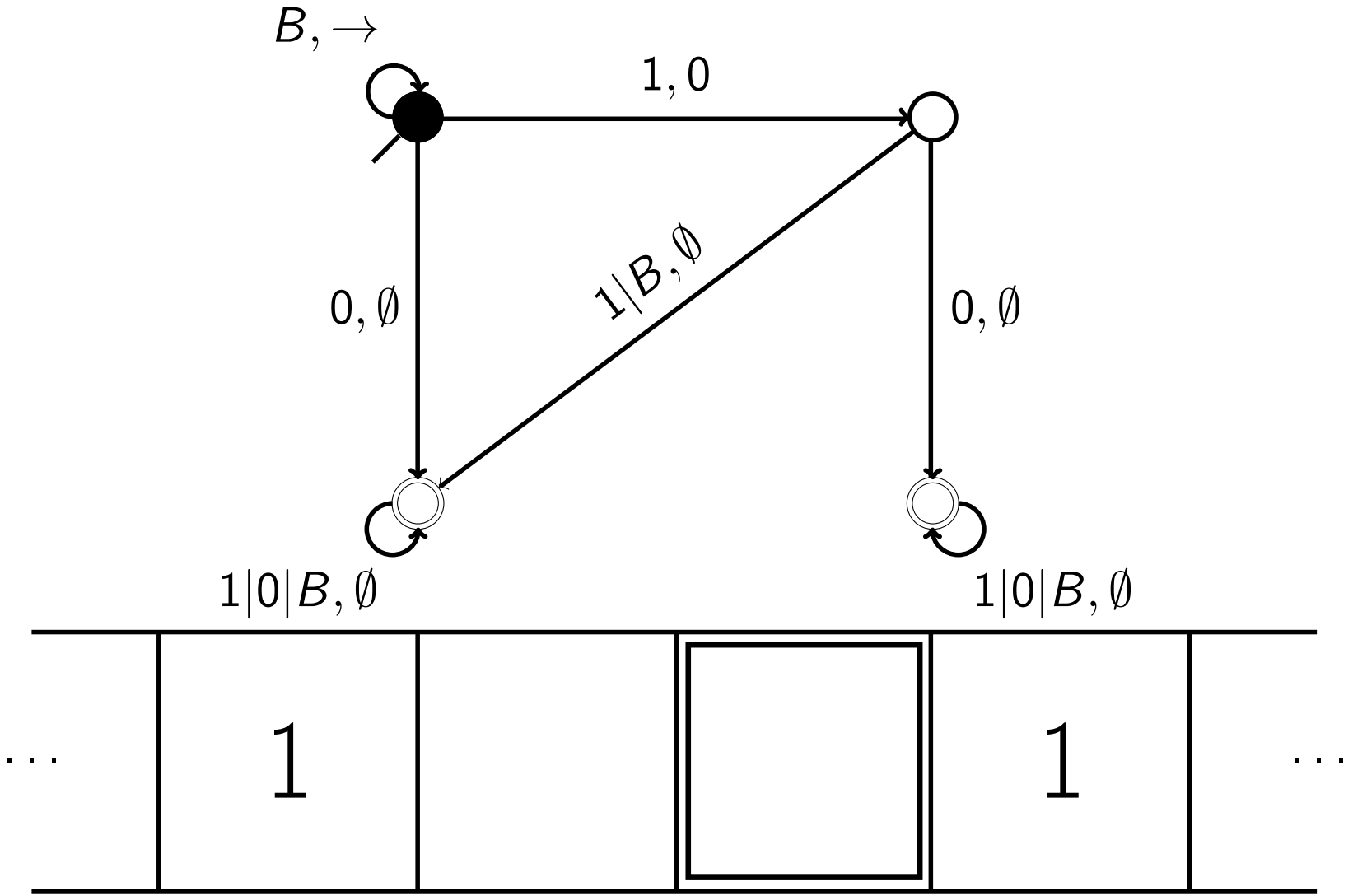
Machine de Turing - le graphe d'états



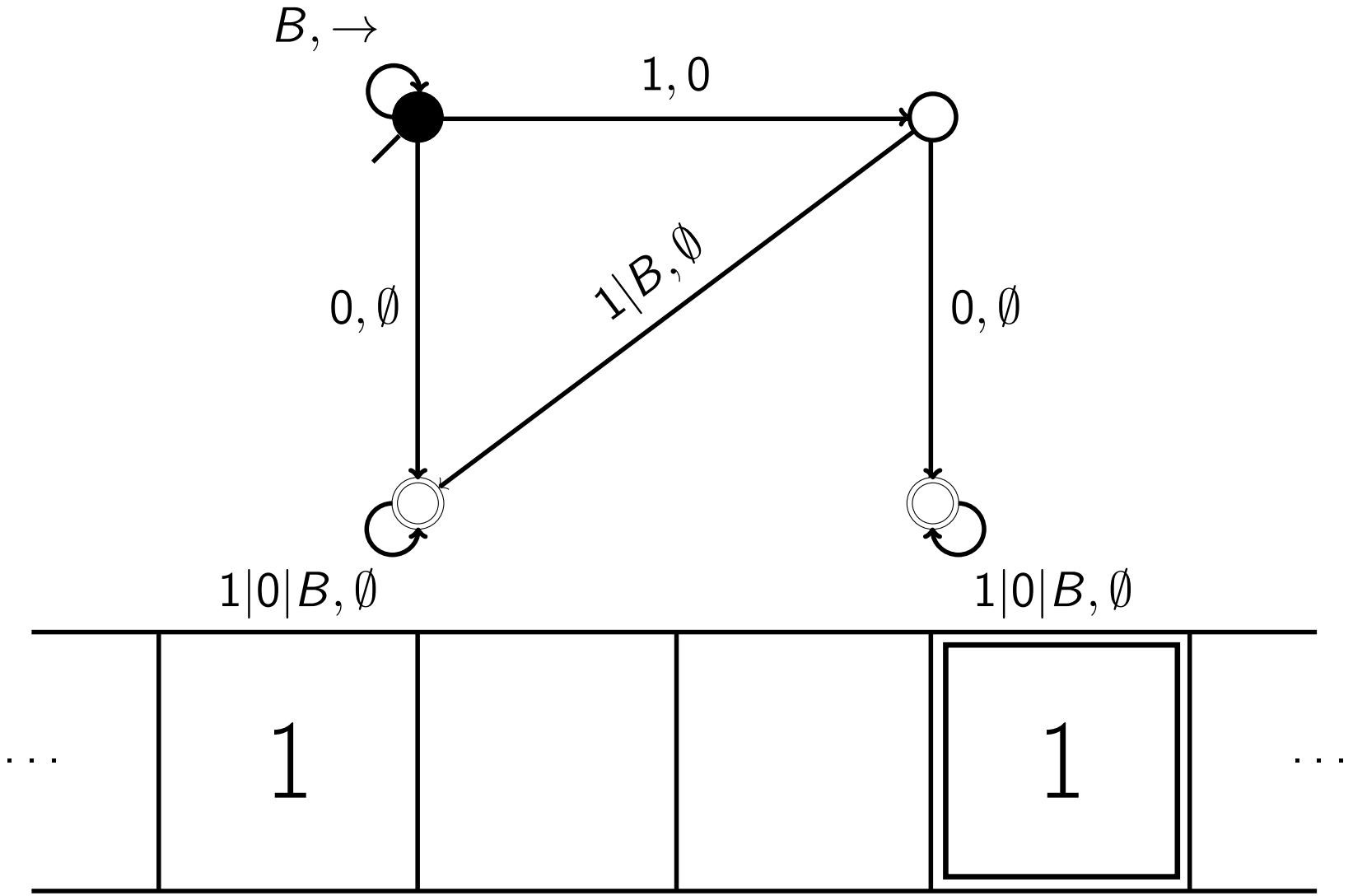
Machine de Turing - le registre d'état



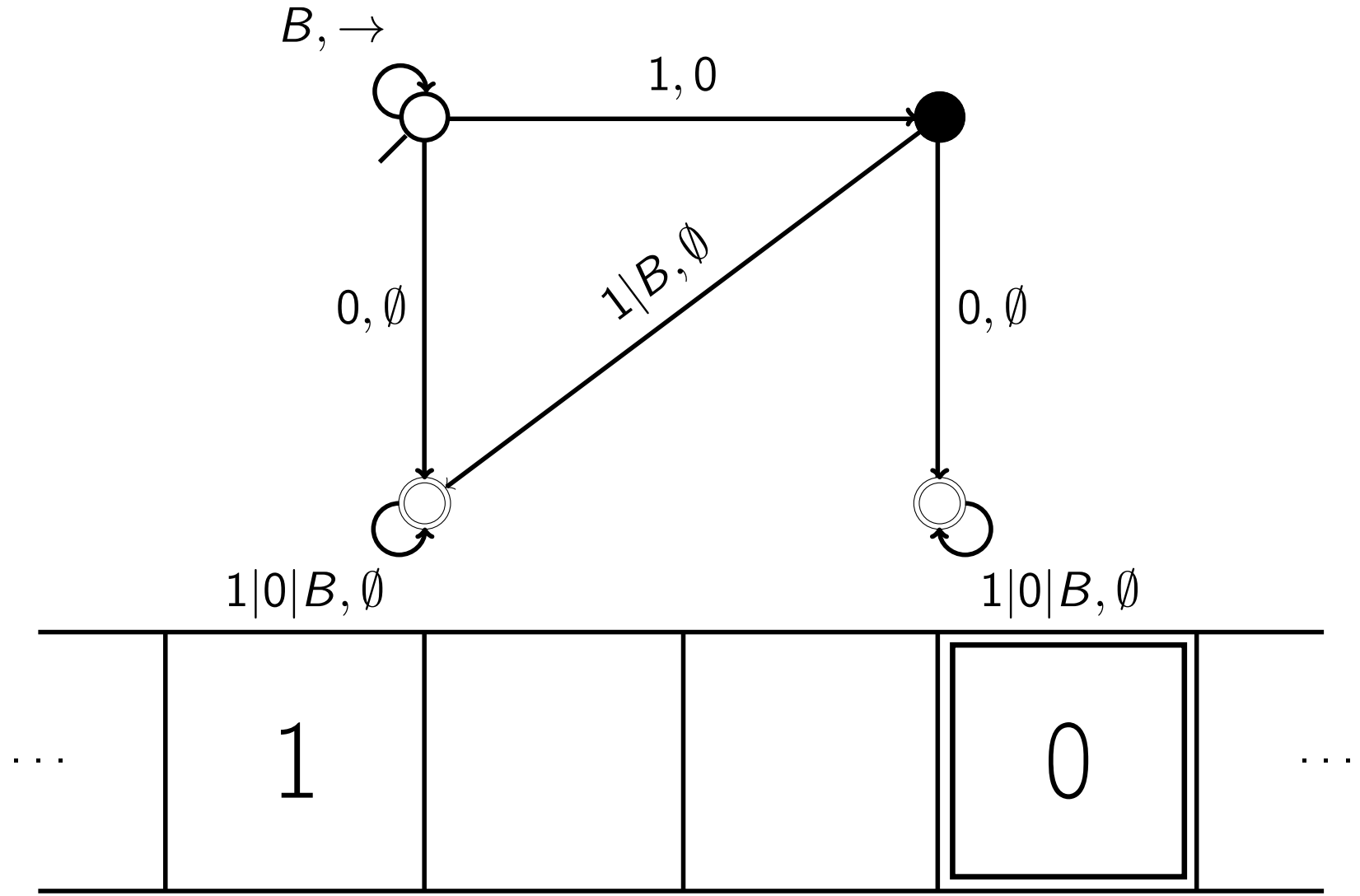
Machine de Turing - le registre d'état



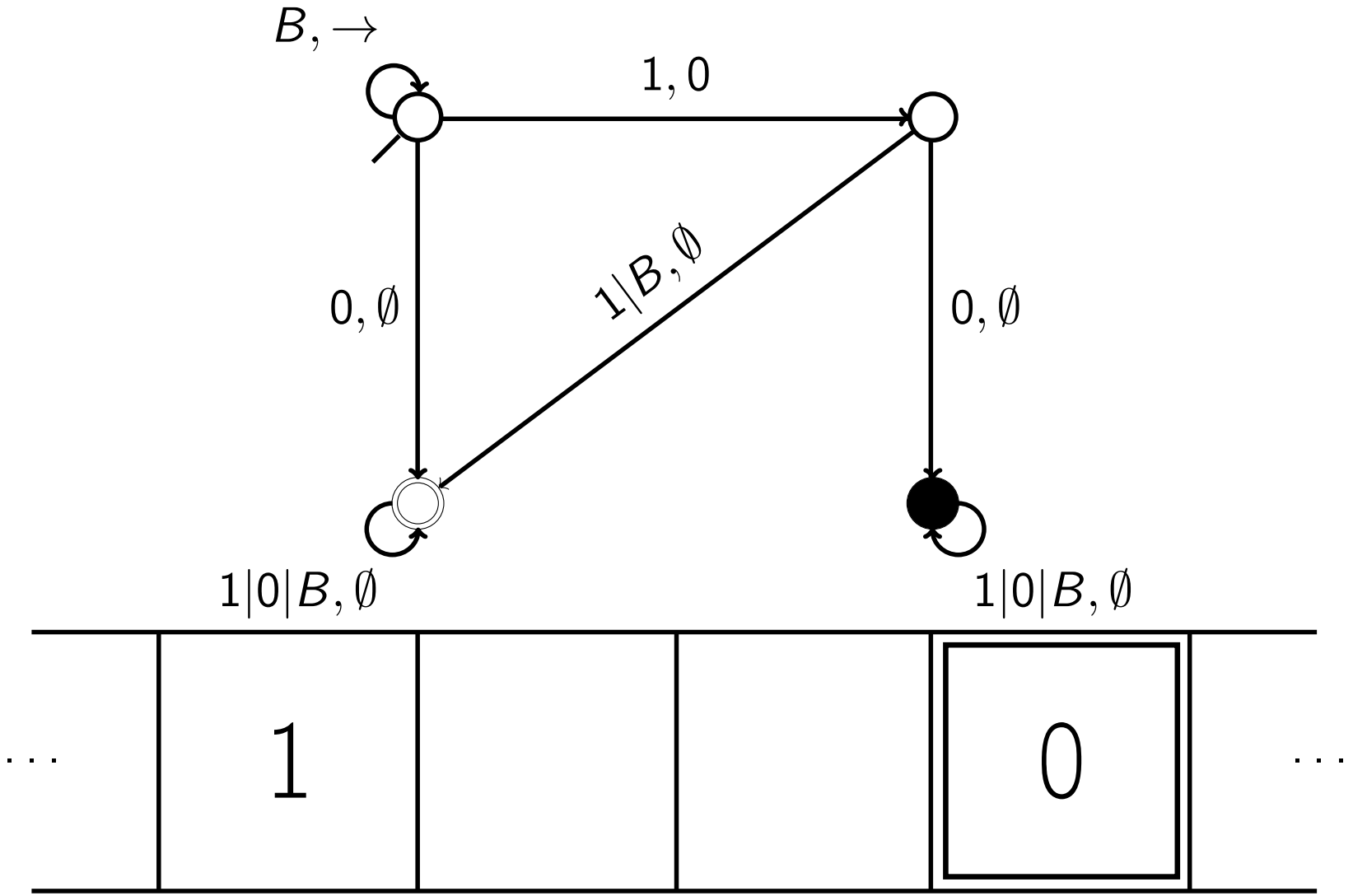
Machine de Turing - le registre d'état



Machine de Turing - le registre d'état



Machine de Turing - le registre d'état



Machine de Turing

Définition

Une machine de Turing contient

- une *bande* $\mathcal{B}$ contenant des cellules numérotées de $-\infty$ à $+\infty$
- un *alphabet* $\Gamma = \{0, 1, B\}$ pouvant être écrit sur la bande
- une *tête* pointant sur la case 0
- un *graphe d'états* fini orienté $G = (S, A)$ où A est étiqueté avec $\Gamma \times (\Gamma \cup \{\leftarrow, \rightarrow, \emptyset\})$, un *symbole de lecture* et une *action*. Deux arcs avec même origine ont des symboles de lecture distincts.
- un *registre d'état* pointant sur l'*état initial* q_0 de S
- un sous-ensemble $F \subset S$ d'*états terminaux* de S

B est un symbole spécial, le *blanc*. Par défaut, la bande est intégralement remplie de B.

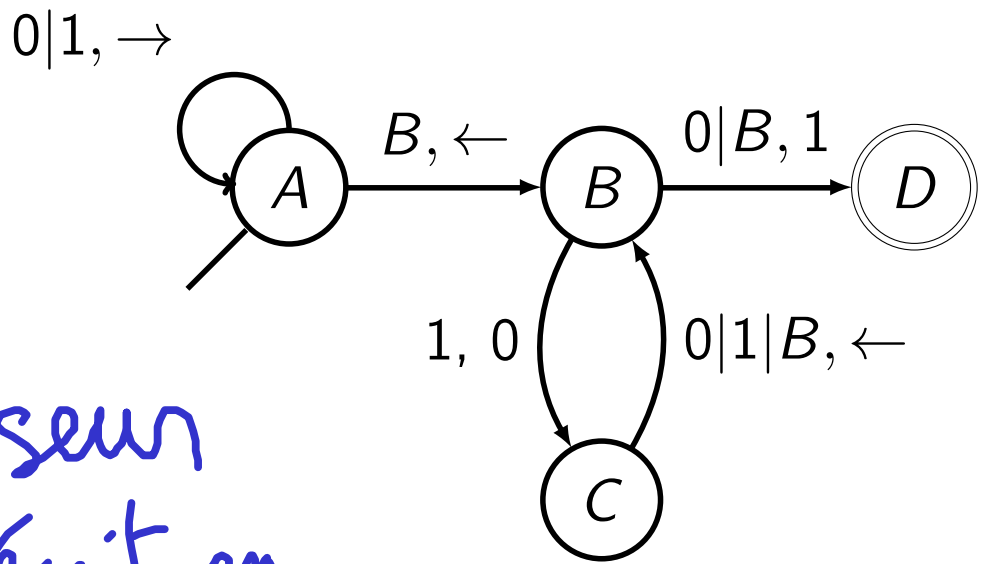
Calcul d'une machine de Turing

On commence par écrire un mot fini x (composé de 1 et de 0, mais pas de B) sur la bande. Le premier caractère est sur la case 0.
À chaque itération,

1. la tête lit le symbole s sur la case c_i où elle pointe ;
2. le registre d'état, pointant sur l'état q, choisit un arc (q, q'), dont le symbole de lecture est s et l'action est a ;
3. on exécute l'action a :
   - si a est 0, 1 ou B, on remplace le symbole sur la case c_i par celui-ci,
   - si a est respectivement $\leftarrow$ ou $\rightarrow$, la tête se déplace respectivement de c_i vers c_{i-1} ou c_{i+1},
   - si a est $\emptyset$, on ne fait rien ;
4. le registre d'état se déplace sur q' ;
5. si $q' \in F$, la machine s'arrête, sinon on recommence.

Exemple

Exécutons la machine suivante avec le mot $x = \{001100\}$ en entrée.



calcule le successeur du mot écrit en entrée

Configuration d'une machine de Turing

Définition

La configuration d'une machine est définie par

- l'état q sur lequel est le registre d'état
- le mot w qui est écrit sur la bande
- la case c_w où est le premier caractère de w
- sur quelle case c_h est la tête de lecture

On note (q, w, c_w, c_h) une telle configuration.

Transitions

Une transition permet de changer la configuration de la machine. On note $(q, w, c_w, c_h) \triangleright (q', w', c'_w, c'_h)$ le changement d'une transition.

On note $(q, w, c_w, c_h) \triangleright^* (q', w', c'_w, c'_h)$ si, en partant de la configuration (q, w, c_w, c_h), une suite de transitions mène à la configuration (q', w', c'_w, c'_h).

Arrêt de la machine de Turing

Une machine de Turing s'arrête pour un mot w avec w' sur la bande s'il existe une configuration (q', w', c'_w, c'_w) où $q' \in F$ et telle que

$$(q_0, w, c_0, c_0) \triangleright^* (q', w', c'_w, c'_w)$$

.

Fonction calculable par une machine de Turing

On note $\overline{n}^2$ le nombre n écrit en binaire.

Définition

Soit $f : \mathbb{N} \to \mathbb{N}$ une fonction. f est dites calculable s'il existe une machine de Turing de $\mathcal{M}$ qui s'arrête pour $\overline{n}^2$ avec $\overline{f(n)}^2$ sur la bande.

L'exemple précédent montre que $f(n) = n + 1$ est calculable.

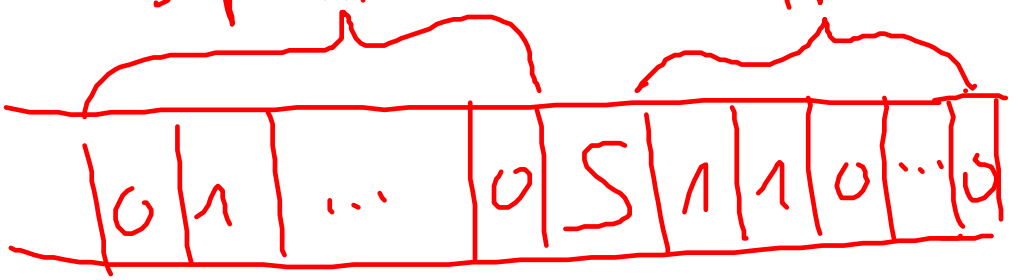
Exercices

Ecrire des machines de Turing pour calculer les fonctions suivantes :

- $prec(n) = n - 1$
- $add(n, m) = n + m$
- $log(2^n) = n$
- $duppl(n) = \overline{n}^2\overline{n}^2$
- $rev(n) = $ le mot $\overline{n}^2$ retourné

définie sur $\mathbb{N}^*$

⚠ Besoin d'un symbole supplémentaire pour séparer n et m en entrée



Normalement on utilise jamais le symbole B, il sert uniquement à remplir la bande infiniement à droite et à gauche.
Le problème est le cas suivant : 000 BBBBB 11
Une machine qui lit cette entrée et qui voit 000 puis B peut se demander si elle a fini de lire l'entrée ou non. Pour éviter cette situation, on peut par exemple:
- utiliser un autre symbole (S ou B') pour faire des blancs différents de B.
- encadrer l'entrée avec deux symboles (par exemple L000BBBBB11L)

Langage accepté

Un langage L est un sous-ensemble (fini ou non) de mots binaires : $L \subset \{0,1\}^*$.

Définition

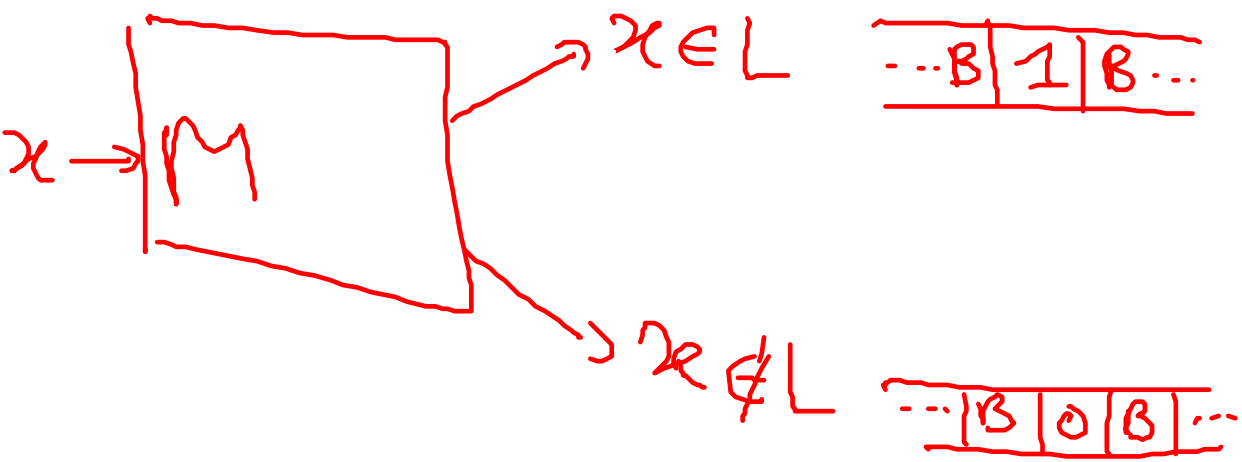
Soit $\mathcal{M}$ une machine de Turing, le langage $L(\mathcal{M})$ accepté par $\mathcal{M}$ est l'ensemble des mots pour lesquels $\mathcal{M}$ s'arrête.

- Soit $\mathcal{M}$ la machine qui calcule $prec(n)$, $L(\mathcal{M})$ est l'ensemble des mots binaires : $L(\mathcal{M}) = \{0,1\}^*$. $\mathbb{N} / \{0\} = \{0,1\}^* - 0^*$ sauf 0
- Soit $\mathcal{M}$ la machine qui calcule $log(2^n)$, $L(\mathcal{M})$ est l'ensemble des puissances de 2 : $L(\mathcal{M}) = 1(0^*)$.

Langage décidable

Définition

Un langage L est *décidable* si et seulement s'il existe une machine $\mathcal{M}$ qui calcule la fonction $f(x) = 1$ si $x \in L$ et 0 sinon.



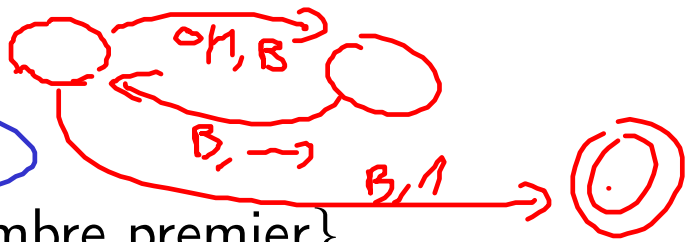
M s'arrête pour tous les mots

$L(M) = \{0,1\}^*$

Exercices

Ecrire des machines de Turing pour décider les langages suivants :

- $L = \{0,1\}^*$
- $\{1^n 0^n, n \in \mathbb{N}\}$
- $L = \{\overline{n}^2, n \text{ nombre premier}\}$



↳ utiliser des machines "boîte noire" ($f(n,m) = n \times m$; $f(n) = n+1$; $f(n,m) = (n == m)$)

Machine équivalente

Partition des états terminaux

On peut partitionner les états terminaux en deux ensembles : les états d'*acceptation* F_Y et de *refus* F_N.

Définition

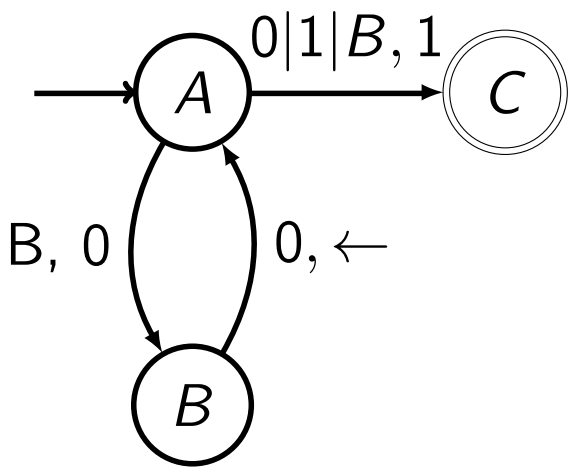
Une machine de Turing *accepte* un mot x, si et seulement si, lorsque l'on exécute la machine avec ce mot écrit sur la bande, la machine s'arrête sur un état acceptant. Elle le *refuse* si elle s'arrête sur un état de refus.

Machine équivalente à une machine décidant un langage

Soit un langage décidable L, il existe une machine $\mathcal{M}$ et une partition des états terminaux F de $\mathcal{M}$ en deux ensembles, les états d'*acceptation* F_Y et de *refus* F_N, telle que pour tout $x \in L$, $\mathcal{M}$ accepte x et pour tout $x \notin L$, $\mathcal{M}$ refuse x.

Machine déterministe, non déterministe

Les machines décrites jusqu'à présent sont *déterministes* : une configuration détermine une unique transition qui mène à une autre configuration.
Dans une machine non déterministe, une configuration peut mener à plus qu'une autre configuration. La machine suivante accepte tous les mots. Si elle reçoit le mot vide, alors elle peut écrire n'importe quelle puissance de 2 sur la bande.



Définition

Une machine de Turing *non déterministe* contient

- une *bande* $\mathcal{B}$ contenant des cellules numérotées de $-\infty$ à $+\infty$
- un *alphabet* $\Gamma = \{0, 1, B\}$ pouvant être écrit sur la bande
- une *tête* pointant sur la case 0
- un *graphe d'états* fini orienté $G = (S, A)$ où A est étiqueté avec $\Gamma \times (\Gamma \cup \{\leftarrow, \rightarrow, \emptyset\})$, un *symbole de lecture* et une *action*.
- un *registre d'état* pointant sur l'*état initial* q_0 de S
- un sous-ensemble $F \subset S$ d'*états terminaux* de S

On n'interdit plus deux arcs avec même origine d'avoir des symboles distincts.

Non déterminisme $\neq$ aléatoire

Il ne faut pas confondre une machine non déterministe avec une machine aléatoire.

Choix d'une machine aléatoire

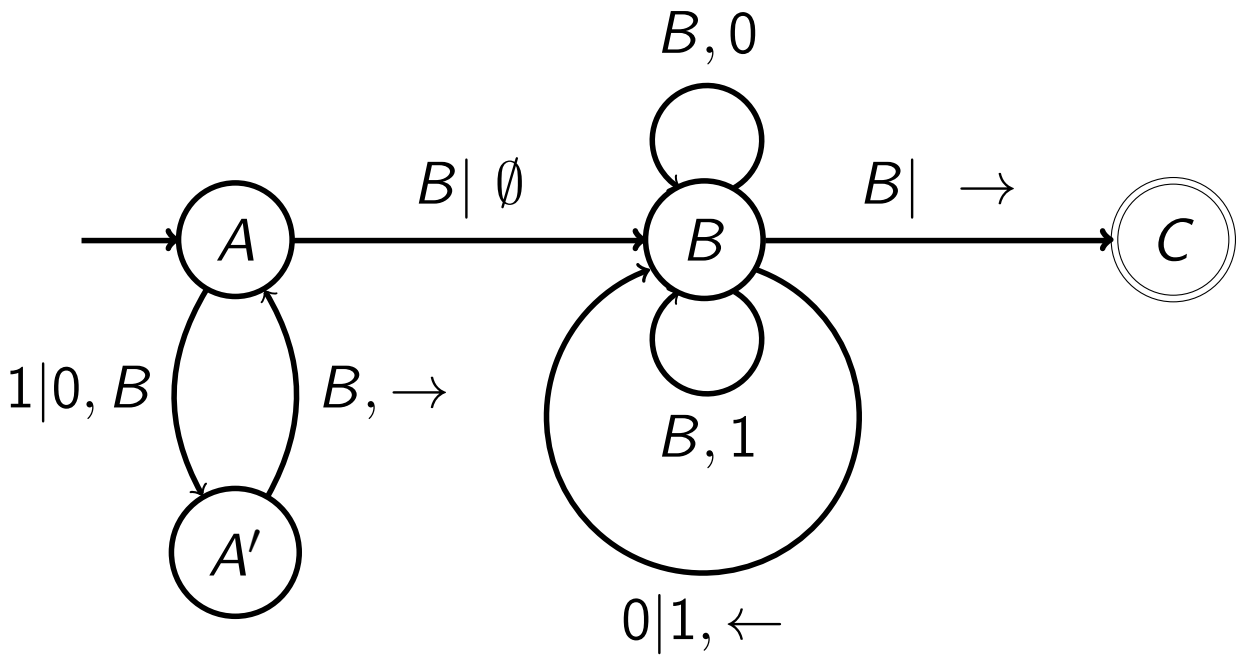
Une machine aléatoire effectue des choix en fonction d'un générateur de bits aléatoire **qu'elle ne maîtrise pas**.

Choix d'une machine non-déterministe

Une machine non-déterministe est capable de visualiser tous les choix possibles, et de toujours prendre le *meilleur* choix, si un tel choix existe.

Calcul avec une machine de Turing non déterministe

Une machine non déterministe ne peut être utilisée (telle quelle) pour calculer une fonction ! La machine suivante peut calculer n'importe quelle fonction.



Acceptance/Refus d'un mot

Définition

Les états terminaux F sont partitionnés en deux ensembles des états d'*acceptation* F_Y et de *refus* F_N

Définition

Une machine de Turing non déterministe accepte un mot x, si et seulement si, lorsque l'on exécute la machine avec ce mot écrit sur la bande, **il existe une suite de choix** telle que la machine s'arrête sur un état acceptant. Elle le refuse si elle peut s'arrêter sur un état de refus.

Acceptance/Refus fort d'un mot

Définition

Les états terminaux F sont partitionnés en deux ensembles des états d'*acceptation* F_Y et de *refus* F_N

Définition

Une machine de Turing non déterministe accepte un mot x *fortement*, si et seulement si, lorsque l'on exécute la machine avec ce mot écrit sur la bande, **quelle que soit la suite de choix**, la machine s'arrête sur un état acceptant. Elle le refuse fortement si elle doit s'arrêter sur un état de refus.

Cette dernière définition n'est pas conventionnelle mais bien pratique

Langage accepté, refusé

Définition

Soit $\mathcal{M}$ une machine de Turing non déterministe, le langage $L_Y(\mathcal{M})$ est l'ensemble des mots acceptés par $\mathcal{M}$.

Définition

Soit $\mathcal{M}$ une machine de Turing non déterministe, le langage $L_Y^f(\mathcal{M})$ est l'ensemble des mots acceptés fortement par $\mathcal{M}$.

Définition

Soit $\mathcal{M}$ une machine de Turing non déterministe, le langage $L_N(\mathcal{M})$ est l'ensemble des mots refusés par $\mathcal{M}$.

Définition

Soit $\mathcal{M}$ une machine de Turing non déterministe, le langage $L_N^f(\mathcal{M})$ est l'ensemble des mots refusés fortement par $\mathcal{M}$.

Exercices

Ecrire des machines de Turing non déterministes $\mathcal{M}_1$,$\mathcal{M}_2$, $\mathcal{M}_3$ et $\mathcal{M}_4$ telles que :

- $L_Y(\mathcal{M}_1) = L_N(\mathcal{M}_1) = \{0,1\}^*$
- $L_N(\mathcal{M}_2) = \{1^n 0^n, n \in \mathbb{N}\}$ (et $L^f_N(\mathcal{M}_2) \neq \{1^m 0^m, m \in \mathbb{N}\}$)
- $L^f_Y(\mathcal{M}_3) = \{\overline{n}^2, n$ nombre premier$\}$ et $L_N(\mathcal{M}_3) = \{\overline{n}^2, n$ nombre non premier$\}$
- $L^f_Y(\mathcal{M}_4) \neq \{\overline{n}^2, n$ nombre premier$\}$ et $L_N(\mathcal{M}_4) = \{\overline{n}^2, n$ nombre non premier$\}$

Impossible car $\overline{L^f_Y(\mathcal{M}_4)} = L_N(\mathcal{M}_4)$ (si la machine s'arrête pour tous les mots)

Si la machine peut ne pas s'arrêter, ok.

Langage accepté

Définition

Soit $\mathcal{M}$ une machine de Turing non déterministe,
$L(\mathcal{M}) = L_Y(\mathcal{M}) \cup L_N(\mathcal{M})$.

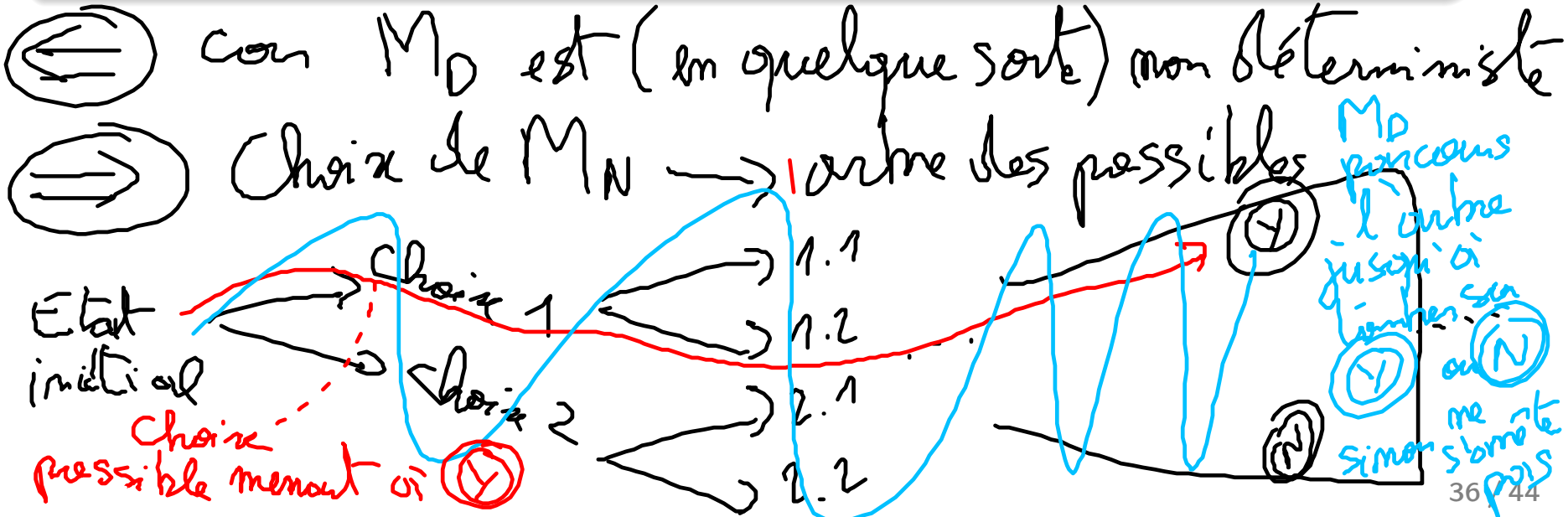
C'est l'ensemble des mots pours lesquels $\mathcal{M}$ peut s'arrêter.

L'impuissance du non déterminisme.

la machine non déterministe doit faire des choix. Ces choix, on peut les représenter sous forme d'un arbre des possibles, chaque choix crée deux branches (les admirateurs de la SF des mondes parallèles connaissent).
Les noeuds de l'arbre sont des configurations de la machine (état + bande + tête).
La machine déterministe peut simuler la machine non déterministe en testant tous les choix possibles ; c'est à dire en explorant toutes les configurations de l'arbre des possibles.
On explore en largeur (c'est le trait bleu). D'abord la configuration initiale, puis les successeurs, puis les successeurs des successeurs.
Si Mn s'arrête, il existe une configuration dans l'arbre avec un état final, que Md découvrira nécessairement. Dans ce cas Md s'arrête. Sinon, aucune configuration de l'arbre n'a d'état final, donc Md explorera infiniement l'arbre des possibles et ne s'arrêtera pas.

Théorème

Soit L un langage, il existe une machine $\mathcal{M}_N$ non déterministe telle que $L(\mathcal{M}_N) = L$ si et seulement s'il existe une machine $\mathcal{M}_D$ déterministe telle que $L(\mathcal{M}_D) = L$.



Décidabilité non déterministe

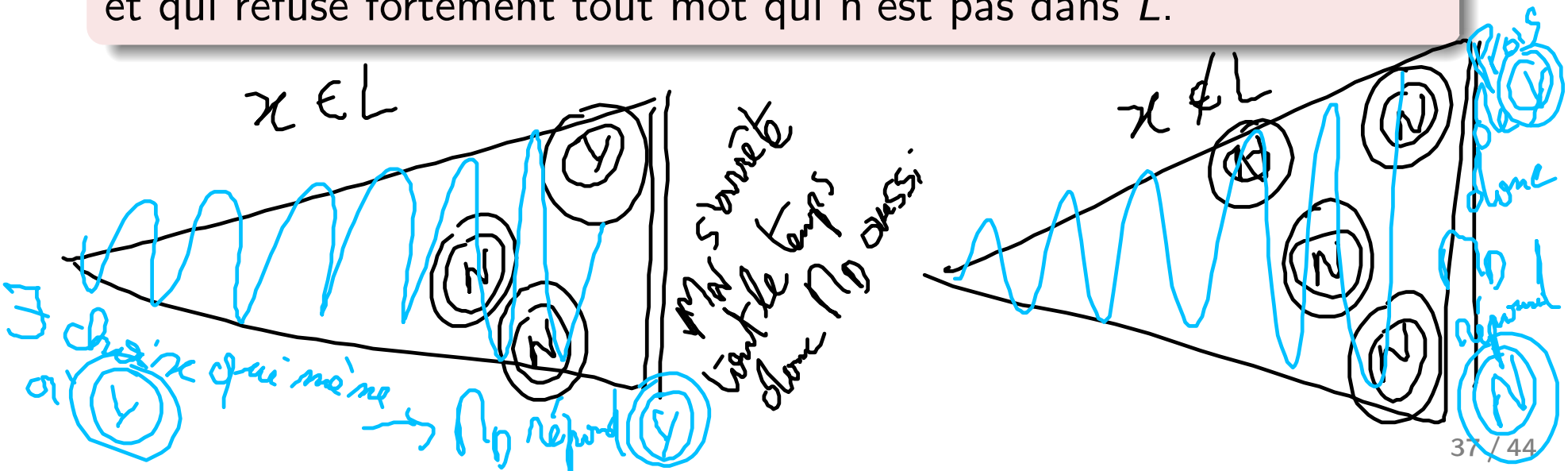
Même idée que précedemment, sauf que, puis que Mn s'arrête tout le temps :
- l'arbre n'est pas infini
- toutes les branches mènent à un état final (acceptant ou refusant)

Si x est dans L, un de ces états finaux est acceptant. Sinon ils sont tous refusant.
Si Md explore l'arbre en largeur, il lui suffit de s'accepter si il tombe sur un état acceptant et de refuser s'il n'y en a aucun.

Difficile à définir, mais il existe des moyens :

Théorème

Un langage L est *décidable* par une machine **déterministe** si et seulement s'il existe une machine $\mathcal{M}$ *non déterministe* qui s'arrête pour tout mot, quel que soient ses choix, qui accepte tout mot de L et qui refuse fortement tout mot qui n'est pas dans L.



Autres machines de Turing équivalentes

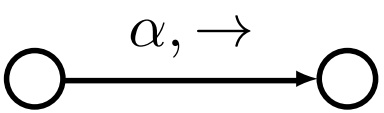
On peut, de manière équivalente,

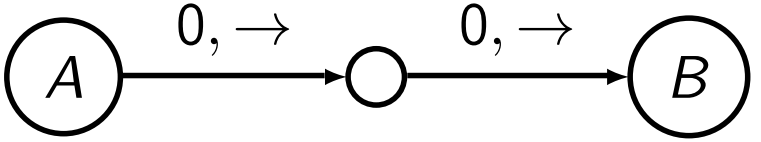
- ajouter des bandes,
- changer l'alphabet
- séparer lecture et écriture,
- fusionner des actions un nombre fini de fois,
- ...

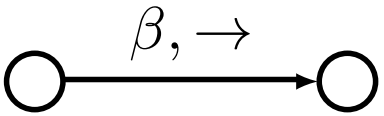
Comme dit en cours, vous avez tout à fait le droit d'utiliser ces propriétés pour construire vos machines.

Changer l'alphabet

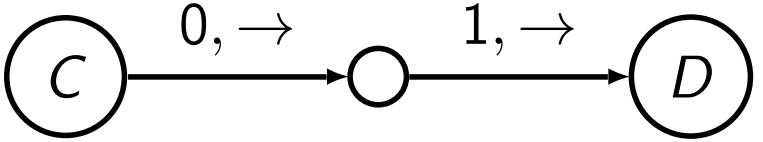
$\Sigma = \{\alpha, \beta, \gamma\} + B$ $\Sigma = \{0, 1\} + B$

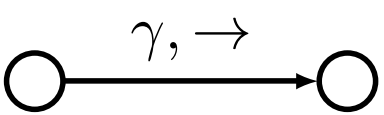


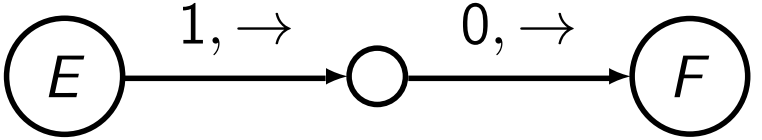




$\Leftrightarrow$







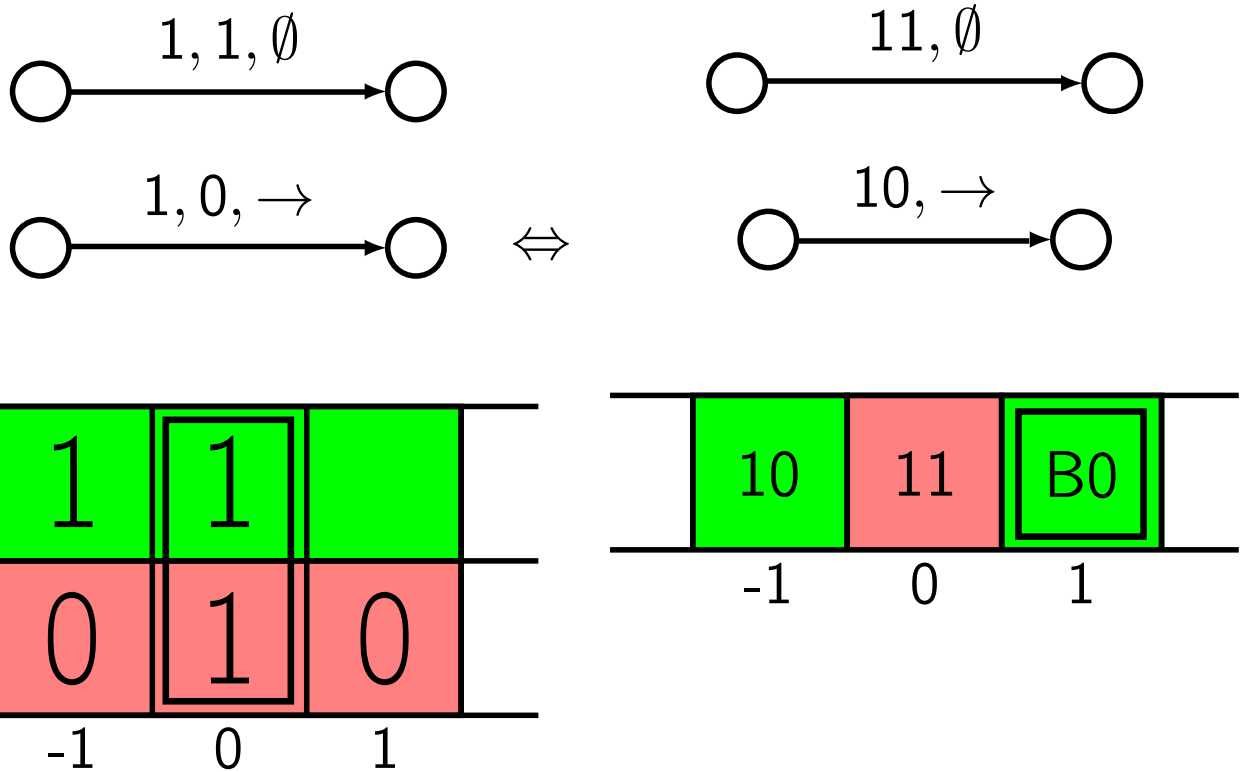
Remarque, on peut, par exemple se passer du blanc B avec la correspondance suivante :
0 -> 10
1 -> 11
B -> 00
Ainsi, une machine n'a besoin en réalité que de 2 symboles.
(avec 1 symbole ça ne marche pas, inutile d'essayer).

Changer l'alphabet

Théorème

Soit $\mathcal{M}$ une machine avec un alphabet Σ à $n \geq 2$ symboles et un symbole B, il est possible de simuler la machine $\mathcal{M}$ avec un alphabet à 2 symboles. (ou 2 symboles et un B)

Ajouter une bande, tête double

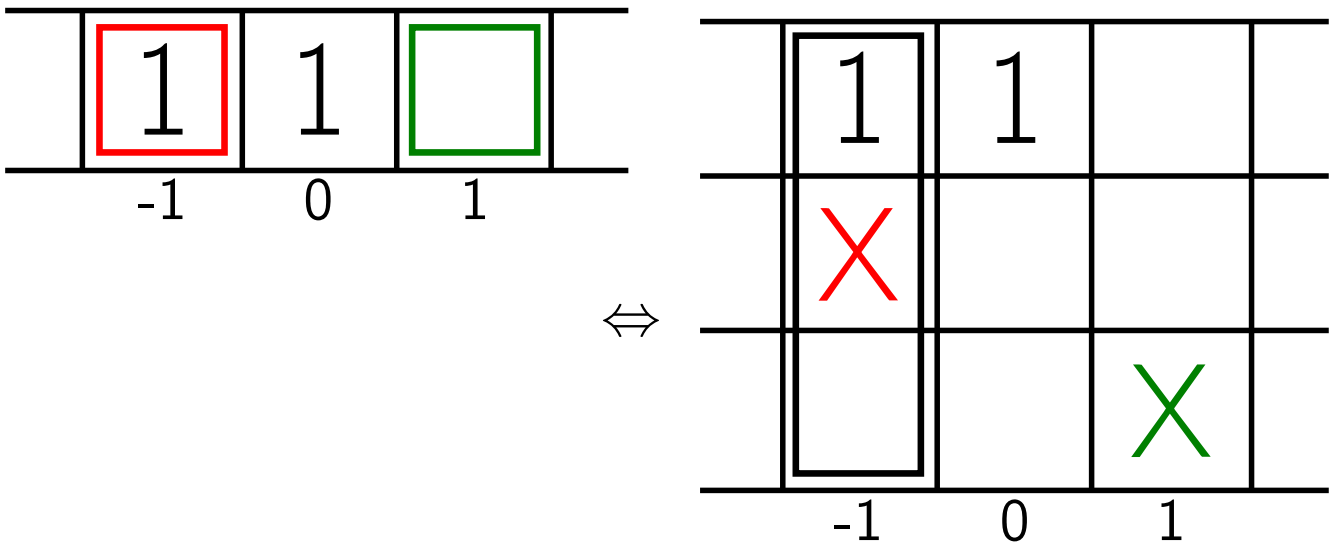


Ajouter une bande, tête double

Théorème

Soit $\mathcal{M}$ une machine avec k bandes et un alphabet Σ, il est possible de simuler la machine $\mathcal{M}$ avec une bande et un alphabet à $(|\Sigma| + 1)^k$ symboles.

Ajouter une tête



Ajouter une tête

Théorème

Soit $\mathcal{M}$ une machine avec une bande et k têtes indépendantes et un alphabet Σ, il est possible de simuler la machine $\mathcal{M}$ avec $k+1$ bandes et un alphabet à $|\Sigma| + k$ symboles.