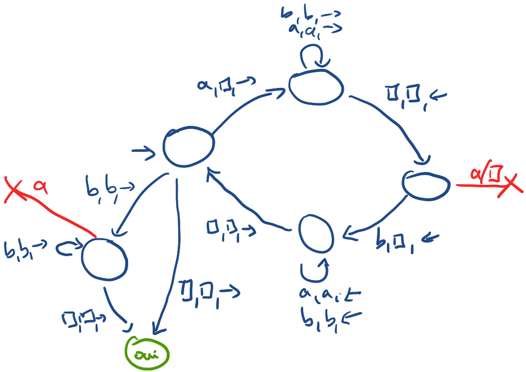
1. Soit $L_1 = \{a^m b^n : m \leq n\}$ l'ensemble des mots constitués de m symboles a puis n symboles b, avec $n \geq m$.

   i. Quel(s) mot(s) parmi les suivants appartien(nen)t à L_1 : b, aab, abb, $ababb$, $aabb$?

   ii. Décrire une machine de Turing qui reconnaît L_1.

i. $b = a^0 b^1$ ✓ $\cancel{aab = a^2 b^1}$ $abb = a^1 b^2$ ✓ $\cancel{ababb}$ $aabb = a^2 b^2$ ✓

ii.



2. Soit $\mathcal{M}$ la machine de Turing décrite par la table de transition ci-dessous. *Chaque ligne représente un quintuplet (état courant, symbole lu, symbole écrit, déplacement, nouvel état). L'état initial est q_0.*

q_0	0	0	$\rightarrow$	q_0
	1	1	$\rightarrow$	q_0
	$\square$	$\square$	$\leftarrow$	r_0
r_0	0	0	$\leftarrow$	r_0
	1	1	$\leftarrow$	r_1
	$\square$	$\square$	$\rightarrow$	s
r_1	0	1	$\leftarrow$	r_0
	1	0	$\leftarrow$	r_2
	$\square$	1	$\leftarrow$	s
r_2	0	0	$\leftarrow$	r_1
	1	1	$\leftarrow$	r_2
	$\square$	0	$\leftarrow$	r_1

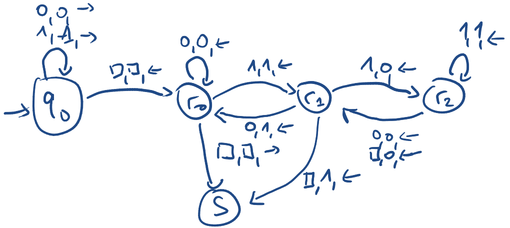
i. Représenter $\mathcal{M}$ sous forme d'automate.

ii. Appliquer $\mathcal{M}$ sur les entrées suivantes : 0, 1, 10, 11 et 110. *On demande uniquement l'état du ruban à la fin du calcul.*

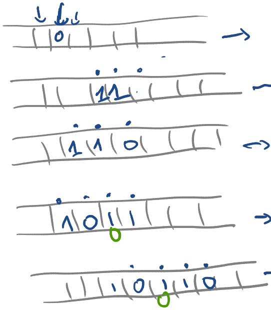
iii. Justifier que $\mathcal{M}$ s'arrête sur toute entrée, et *préciser dans quel état.*

iv. En interprétant le contenu du ruban comme un entier en binaire, décrire le calcul effectué par $\mathcal{M}$.

i.



ii.



→ 0

→ 11

→ 110

→ 1001

→ 10010

iii. - L'entrée est finie donc on ne reste pas bloqué en q_0
- Dans les états r_0, r_1, r_2, on va toujours vers la gauche donc on finit par atteindre un □. Dans ce cas on va en S et on est bloqué.

iv. $0 \rightarrow 0$

$1 \rightarrow 11_2 = 3$

$10_2 = 2 \rightarrow 110_2 = 6$

$11_2 = 3 \rightarrow 1001_2 = 9$

$110_2 = 6 \rightarrow 10010_2 = 18$

$\Rightarrow$ Multiplication par 3

Exercice 2 (sur 5 pts). *Variantes des machines de Turing*

On considère des variantes des machines de Turing possédant plusieurs rubans ayant chacun des *droits* en lecture et/ou écriture : un ruban peut être en *lecture seule* (on peut lire le contenu du ruban mais rien écrire dessus), en *écriture seule* (on peut écrire sur le ruban, mais on ne peut pas relire ce qu'on y a écrit), ou en *lecture/écriture* (on peut à la fois lire et écrire).

1. On suppose que tous les rubans sont en lecture seule. Cette variante permet-elle de reconnaître tout langage reconnaissable ?
2. On suppose que la machine a trois rubans : un ruban en lecture seule contenant l'entrée, un ruban en écriture seule sur lequel on écrit la sortie, et un 3ème ruban en lecture/écriture. Cette variante permet-elle de calculer toute fonction calculable ?
3. On suppose que les rubans de la machine sont soit en lecture seule, soit en écriture seule (mais aucun en lecture/écriture). Cette variante permet-elle de calculer toute fonction calculable ?

1. Non car la machine n'a qu'une mémoire finie.

2. Oui : on copie l'entrée sur le 3ème ruban, on effectue le calcul sur le 3ème ruban, puis on recopie la sortie sur le ruban de sortie.

3. Non car on n'a pas de mémoire exploitable

- On peut supposer qu'il n'y a qu'un ruban en lecture seule → entrée
  écriture seule → sortie
-

Exercice 3 (sur 7 pts). *(In)calculabilité*

1. Soit L un langage et $\overline{L}$ son complémentaire.

i. Montrer que si L est fini (c'est-à-dire contient un nombre fini de mots), il est décidable.

ii. Montrer que si $\overline{L}$ est fini, alors L est décidable.

i. Si $L = \{w_1, \dots, w_k\}$, on peut faire l'algo suivant :

$A(w)$:

Si $w = w_1$: renvoyer VRAI

Si $w = w_2$: renvoyer VRAI

⋮

Si $w = w_k$: renvoyer VRAI

Renvoyer FAUX

ii. - Si $\overline{L}$ est fini, il est décidable donc reconnaissable et co-reconnaissable. Donc L est co-reco. et reco. donc décidable.

- Autre preuve : idem i. en inversant VRAI et FAUX

2. Soit $K = \{\langle A, w\rangle : A(w)\downarrow\}$ le problème de l'arrêt, et $L_2 = \{\langle A, w\rangle : w \notin L(A)\}$.

   i. Montrer que si on sait décider L_2, alors on sait décider K.

   ii. Le langage L_2 est-il décidable ? reconnaissable ? co-reconnaissable ?

i. - On suppose qu'on a un algo A_2 tq $A_2(\langle A, w\rangle) = \begin{cases} 1 \text{ si } w \notin L(A) \\ 0 \text{ sinon} \end{cases}$

- On veut utiliser A_2 pour décider le problème de l'arrêt.

  ↳ On a en entrée $\langle A, w\rangle$ et veut savoir si $A(w)\downarrow$.

  A partir de A, on construit A' : sur l'entrée w, A' simule A. Si A s'arrête, A' passe dans l'état "oui".

  Alors : $A(w)\downarrow \iff w \in L(A')$

  Il suffit de passer $\langle A', w\rangle$ en entrée à A_2 et de répondre l'inverse de A_2.

ii. - L_2 n'est pas décidable car sinon K serait décidable

- L_2 est co-reconnaissable car on peut exécuter $A(w)$ et si $w \in L(A)$, alors $A(w)$ s'arrête dans l'état oui. (On sait reconnaître si $\langle A, w \rangle \notin L_2$)

- L_2 n'est pas reconnaissable car il serait décidable sinon.

<u>Via Rice</u> $P_{\overline{w}} = \{ f : f(w) \neq 1 \}$ $\rightarrow$ Montre que L_2 indécidable