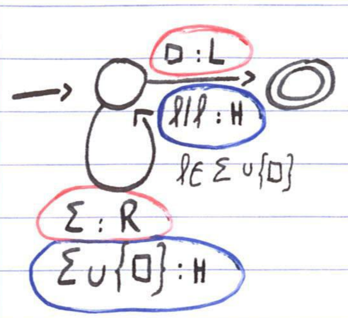
08/02/2018

TURING

① TD

Séance 3 - suite fiche TD1 -

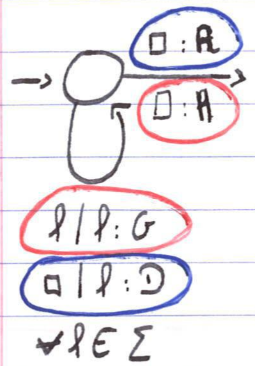
question 15 -



On place correctement les têtes de lecture.

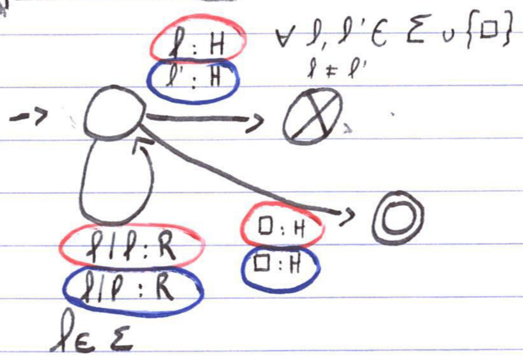
rouge : bande 1.
bleu : bande 2.

=> MR



recopie.
// manque la remise en place des têtes

question 16 -



=> Meq

question 17 -

Mpal = [MR ; Meq]

question 18 · À faire pour le DS de MCAL.

Fiche TD 2.

Exercice 1.

q1. $\forall i,j \in [0..N],\ i<j \Rightarrow t[i] \leqslant t[j]$

q2.

i	0	5	2	4	3	1
$t[i]$	2	2	3	4	42	521

NON → tableau et pas multiensemble
→ les indices changent pas.

$M = \cancel{\{T(0,2), T(5,2), T(2,3), T(4,4), T(3,42), T(1,521)\}}$
$= \{T(0,2), T(1,2), T(2,3), T(3,4), T(4,42), T(5,521)\}$

question 3.

$$T(i, v_i),\ T(j, v_j) \xrightarrow{i<j \wedge v_i > v_j} T(i, v_j),\ T(j, v_i)$$

question 5.

a) $M = \{T(0,2), T(1,521), T(2,3), T(3,42), T(4,4), T(5,2)\}$

$M = \{T(0,2), T(1,2), T(2,3), T(3,4), T(\cancel{3}4,42), T(5,521)\}$

b) 2

c) ~~4~~ 1 // execution en parallele

question 4. $\forall T(i,k), T(j,l) \in M,\ i<j \Rightarrow k \leqslant l.$

Exercice 3.

q_{14}) a) $q \xrightarrow[0/1:L]{1/0:R} q'$: pas d'effet

b) $q \xrightarrow[1/0:L]{0/1:R} q'$:

q15. a) T_1 lit 0, T_2 lit 1, les tetes sont alignés.
b) les tetes sont pas là (mais elles sont où ?)
c) T_1 lit 0 et la deuxieme est pas là.

08/02/2018

TURING

2 TD

question 16 .

$^{\infty}\square$	$\$\square\$\square$	$1\square 0\square$	$0\uparrow 1\square$	$1\square 0\square$	$1\square 1\square$	$\square\square 1\uparrow$	$\square^{\infty}$

fiche TD2

MCAL/MT 2018 - série 2

Exercice 1 : Tableaux et tri en Gamma

En Gamma un tableau

i	0	1	2	3	4	5
t$[i]$	2	521	3	42	4	2

est représenté par le multi-ensemble

$$\mathscr{M} = \{\,\mathrm{T}(0,2), \mathrm{T}(1,521), \mathrm{T}(2,3), \mathrm{T}(3,42), \mathrm{T}(4,4), \mathrm{T}(5,2)\,\}$$

où les éléments T*(indice,valeur)* flottent dans la solution chimique.

Q1. Donnez la propriété logique qui caractérise le fait que le tableau t$[0..N]$ est trié dans l'ordre croissant sous la forme d'une formule commençant par $\forall$.

Q2. Donnez le multi-ensemble qui correspond au tableau T trié dans l'ordre croissant.

Q3. Donnez la/les règle(s) Gamma qui permettent d'obtenir un multi-ensemble d'éléments T*(indice,valeur)* trié dans l'ordre croissant.

Q4. Donnez la propriété logique qui caractérise le fait que le multi-ensemble $\mathscr{M}$ d'élements T*(indice,valeur)* est trié dans l'ordre croissant sous la forme d'une formule commençant par $\forall$.

Q5.

(a) Donnez l'exécution la plus efficace de cet algorithme sur le multi-ensemble $\mathscr{M}$ de l'exemple.

(b) Combien d'applications des règles sont nécessaires pour obtenir un multi-ensemble trié ;

(c) Combien d'étapes sont nécessaires pour obtenir le multi-ensemble trié.

Exercice 2 : Terminaison de programmes GAMMA

On considère divers programmes GAMMA dont on va étudier la terminaison. Pour chacun d'eux donnez

1. un ensemble de départ telle que l'exécution qui termine
2. un ensemble de départ telle que l'exécution qui ne termine pas

Lorsque l'un est impossible, expliquez pourquoi.

Donnée manipulées Les programmes suivantes opèrent sur des constructeurs à un arguments A(.) et B(.) et un constructeur sans argument O qui correspondent au type CAML :

```
type data = O | A of data | B of data
```

Q6. Donnez 5 éléments de type data.

Q7. Étudiez la terminaison du programme $\Gamma_1 \stackrel{\text{def}}{=} \mathrm{A}(\mathrm{O}) \xrightarrow[r_1]{} \mathrm{B}(\mathrm{O}) \;||\; \mathrm{B}(x) \xrightarrow[r_2]{} \mathrm{A}(x)$

Q8. Étudiez la terminaison du programme $\Gamma_3 \stackrel{\text{def}}{=} \mathrm{A}(x) \xrightarrow[r_1]{} \mathrm{A}(\mathrm{B}(x)) \;||\; \mathrm{B}(\mathrm{A}(x)) \xrightarrow[r_2]{} \mathrm{A}(x)$

Q9. Étudiez la terminaison du programme $\Gamma_4 \stackrel{\text{def}}{=} \mathrm{A}(\mathrm{B}(x)) \xrightarrow[r_1]{} \mathrm{B}(\mathrm{B}(x)) \;||\; \mathrm{B}(\mathrm{B}(x)) \xrightarrow[r_2]{} \mathrm{A}(x)$

Q10. Donnez un programme constitué d'une seule règle qui ne termine jamais quel que soit le multi-ensemble de départ ...

(a) Γ_{5a} fait grandir le multi-ensemble

(b) Γ_{5b} fait grandir l'élément mais pas le multi-ensemble

Q11. Étudiez la terminaison du programme $\Gamma_6 \stackrel{\text{def}}{=} \mathrm{A}(x) \xrightarrow[r_1]{} \mathrm{A}(x), \mathrm{B}(x)$

Q12. Ajoutez une règle au programme Γ_6 afin qu'il termine pour tout multi-ensemble de départ

Q13. Étudiez la terminaison du programme $\Gamma_7 \stackrel{\text{def}}{=} \mathrm{A}(x) \xrightarrow[r_1]{} \mathrm{A}(x), \mathrm{A}(x) \;||\; \mathrm{A}(x) \xrightarrow[r_2]{} -$

Exercice 3 : Machine multi-bande → Machine de Turing classique

On considère le cas d'une MT M à k bandes $B_1, \ldots, B_k$ avec chacune sa propre tête de lecture/écriture, notées $T_1, \ldots, T_k$.

— Une transition de M à k bandes est de la forme

$$\delta(\mathbf{q},\ (\ell_1, \cdots, \ell_k)\ ) = (\mathbf{q}',\ (e_1, \cdots, e_k),\ (d_1, \cdots, d_k)\ )$$

où les ℓ_i sont des symboles lus, les e_i les symboles écrits et les d_i sont les déplacements.
Elle s'effectue simultanément sur les k bandes, à condition que le vecteur des symboles lus par les k tête de lecture/écriture sur les k bande correspondent aux symboles $(\ell_1, \ldots, \ell_k)$ attendus par la transition. Si c'est le cas, la machine change d'état, elle écrit chaque symbole e_i sur la bande B_i et effectue le déplacement d_i de la tête de lecture/écriture T_i ; ceci sur chacune des bandes d'après la fonction de transition δ de M :

— Pour simplifier, au démarrage, seule la bande B_1 contient l'entrée $\$.\omega$ de la MT M, les autres bandes sont de la forme

$^\infty\square$	\$	$\square^\infty$

Le but du TD est de montrer le résultat suivant pour $k = 2$ qu'on peut ensuite généraliser à $k \in \mathbb{N}$.

Proposition 1 *Une machine de Turing à k bandes utilisant un alphabet Σ peut être simulée par une machine de Turing à* **une seule bande** *utilisant un alphabet plus riche.*

Principe de la simulation pour $k = 2$

On considère une machine de Turing M à 2 bandes opérant sur Σ.

— les transitions de M sont de la forme $\textcircled{q} \xrightarrow[\ell_2/e_2:d_2]{\ell_1/e_1:d_1} \textcircled{q'}$.
La partie $\ell_1/e_1 : d_1$ concernent la bande B_1 et la partie $\ell_2/e_2 : d_2$ concerne la bande B_2.

— Supposons qu'**à un instant de l'exécution de M**, la bande B_1 contient le mot \$.1.0.1.1 avec T_1 positionnée sur 0 et la bande B_2 contient le mot \$.0.1.0.1.0.1 avec T_2 positionnée sur le dernier caractère.
On peut superposer les bandes B_1 et B_2 en alignant les \$ et ajouter deux bandes fictives T_1 et T_2 comportant un unique symoble ↑ à la position de la tête de lecture/écriture. L'empilement des bandes forment un tableau infini qui modélise le contenu des bandes B_1, B_2 et la position des têtes T_1, T_2 :

B_1 :	$^\infty\square$	\$	1	0	1	1	$\square$	$\square$	$\square^\infty$
T_1 :	$^\infty\square$	$\square$	$\square$	↑	$\square$	$\square$	$\square$	$\square$	$\square^\infty$
B_2 :	$^\infty\square$	\$	0	1	0	1	0	1	$\square^\infty$
T_2 :	$^\infty\square$	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$	↑	$\square^\infty$

Q14. On suppose que M est dans l'état $\mathbf{q}$, donnez l'effet de chacune des transitions sur le tableau précédent (a) $\mathbf{q} \xrightarrow[0/1:L]{1/0:R} \mathbf{q'}$ (b) $\mathbf{q} \xrightarrow[1/0:L]{0/1:R} \mathbf{q'}$ (c) $\mathbf{q} \xrightarrow[0/1:R]{0/1:R} \mathbf{q'}$ (d) $\mathbf{q} \xrightarrow[1/1:H]{0/0:H} \mathbf{q'}$

Pour remplacer les 2 bandes par une seule, on considère chaque colonne du tableau comme un symbole de la machine M' à **une seule bande** qui doit simuler le comportement de M.
Un symbole de M' est un vecteur de symboles de la forme (s_1, t_1, s_2, t_2) où $t_1 = \uparrow$ indique la présence de la tête de lecture/écriture sur le symbole s_1 et $t_2 = \square$ indique que la tête de lecture/écriture de B_2 n'est pas sur s_2.

Q15. Interprétez les vecteurs de symboles suivants correspondant à une colonne du tableau.
Exemple : (a) $(0, \uparrow, 1, \uparrow)$; (b) $(0, \square, 1, \square)$; (c) $(0, \uparrow, 1, \square)$;

Q16. Donnez le ruban R de la MT M' correspondant au tableau.

Q17. Soit $\Sigma = \{0, 1, \$\}$ l'alphabet de la machine M. Donnez l'alphabet Σ' sur lequel travaillera la machine M' et indiquez sa taille. Généralisez au cas Σ_k où la MT m' simule une MT M à k bandes.

Construction de la MT à une bande équivalente à la MT à deux bandes

Considérons la MT à deux bandes $M = (\Sigma, \mathscr{Q}, \mathbf{q}, \delta, \mathscr{A}cc, \mathscr{E}rr)$. **Notre objectif** est de contruite une MT $M' = (\Sigma', \mathscr{Q}', \mathbf{q}', \delta', \mathscr{A}cc', \mathscr{E}rr')$ à une bande qui simule la machine M à 2 bandes. La MT M' travaillera sur des symboles (s_1, t_1, s_2, t_2) qui encodent les contenus des rubans de M et la position des têtes.
Considérons une transition $\mathbf{q} \xrightarrow[\ell_2/e_2:d_2]{\ell_1/e_1:d_1} \mathbf{q'}$ de M.

Pour simuler une transition de la machine M à 2 bandes, il faut connaître le symbole courant sur lesquel pointe chaque tête puis simuler les actions de la transitions. Pour cela,

1. dans un premier parcours de gauche à droite du ruban, on détermine la position de chaque tête de lecture/écriture et les symboles ℓ_1, ℓ_2 lus afin de déterminer la transition de la machine d'origine à exécuter.
2. dans un deuxième parcours de droite à gauche du ruban, la machine effectue les écritures e_1, e_2 et les déplacements d_1, d_2 correspondant aux actions de la machine d'origine.

On distingue deux phases dans le fonctionnement de la MT M' :

1. la recherche du symbole courant (ℓ_1, ℓ_2) de M : c'est la tâche de la MT M_{lect}
2. la réalisation des actions de la transition τ de M associée à cette lecture : c'est la tâche de la MT M_τ

Q18. Complétez. Les états de M_{lect} sont de la forme $\mathbf{q}_{(s_1,s_2)}$ où $\mathbf{q} \in \mathscr{Q}$ et $s_1, s_2 \in \Sigma \cup \{?\}$. L'ensemble $\mathscr{Q}_{lect}$ des états de la MT M_{lect} est donc $\ldots\ldots \times (\Sigma \cup \ldots\ldots\ldots) \times (\Sigma \cup \ldots\ldots\ldots)$ et $|\mathscr{Q}_{lect}| = |\mathscr{Q}| \times (\ldots\ldots + 1)^2$.

> **Indication :** On note $\mathbf{q}_{(?,?)}$ l'état initial de M_{lect} qui évolue en $\mathbf{q}_{(\ell_1,?)}$ lorsque qu'elle a trouvé le symbole ℓ_1 pointé par la tête de lecture sur la bande B_1 puis évolue en $q_{(\ell_1,\ell_2)}$ lorsqu'elle a trouvé le symbole ℓ_2 pointé par la tête de lecture sur la bande B_2

Q19. Donnez une MT M_{lect} qui effectue un parcours de gauche à droite du ruban et trouve le symbole lu par chaque tête de M. Commencez par considérer les transitions de l'état initial $\mathbf{q}_{(?,?)}$ puis généralisez à un état quelconque de la forme $\mathbf{q}_{(s_1,s_2)} \in \mathscr{Q}_{lect}$.

Remarque Si on adopte la convention que la MT M démarre dans l'état initial $\mathbf{q}$ sur le $(\$, \$)$ alors la première phase – et donc la machine M_{lect} – sont inutiles. C'était juste pour s'échauffer. En réalité on peut se passer de M_{lect} et faire démarrer M' dans l'état $\mathbf{q}_{(\$,\$)}$. Il est alors inutile de rechercher les caractères pointés par les têtes, il suffit de les mémoriser au moment du déplacement des têtes. Il faudra malgré tout chercher les têtes pour effectuer les écritures et les déplacements.

Simultation des transitions de M

Considérons la transition suivante de M : $\tau \stackrel{\text{def}}{=} \mathbf{q} \xrightarrow[\ell_2/e_2\ :\ d_2]{\ell_1/e_1\ :\ d_1} \mathbf{q'}$ qui lit *simultanément* ℓ_1 sur la bande 1 et ℓ_2 sur la bande 2, écrit e_1 sur la bande 1 et e_2 sur la bande 2, puis effectue le déplacement d_1 (resp. d_2) de la tête de la bande 1 (resp. 2).

Un état de M_τ est de la forme $\mathbf{q}^{e_1:d_1,e_2:d_2}_{(\ell_1,\ell_2)}$ où (ℓ_1, ℓ_2) indique les symboles lus par les deux têtes de M et $e1 : d_1, e_2 : d_2$ indique les actions qu'ils restent à effectuer. Le symbole $\bullet$ sera utilisé pour indiquer qu'une action a été réalisée.

> **Indication :** On passe dans un état $\mathbf{q}^{\bullet:\uparrow,e_2:d_2}_{(e_1,\ell_2)}$ lorsqu'on a effectué l'écriture e_1 et le déplacement d_1 et qu'il reste à inscrire le symbole $\uparrow$ indiquant la position de la tête 1. On passe dans un état $\mathbf{q}^{\bullet:\bullet,e_2:d_2}_{(e_1,\ell_2)}$ lorsqu'on a effectué les actions qui concerne la bande 1. On passera dans un état $\mathbf{q}'_{(s_1,s_2)}$ quand on aura effectué toutes les actions de la transition τ.

Q20. Complétez les pointillés Complétez les transitions suivantes qui effectuent les actions $e_1 : d_1, e_2 : d_2$ en découvrant tout d'abord la tête 1 puis la tête 2.

1. on effectue l'écriture e_1 et on efface la flêche $\uparrow$ qui indique la position de la tête T_1

$$\mathbf{q}^{e_1:d_1,e_2:d_2}_{(\ell_1,\ell_2)} \xrightarrow[\forall s_1,s_2 \in \Sigma,\ \ \forall t_2 \in \{\square,\uparrow\},]{(s_1,\uparrow,s_2,t_2)/(e_1,\ldots\ldots,s_2,t_2)\ :\ d_1} \mathbf{q}^{\bullet:\uparrow,e_2:d_2}_{(\ldots,\ell_2)}$$

2. on inscrit la flêche $\uparrow$ qui indique la position de la tête T_1

$$\mathbf{q}^{\bullet:\uparrow,e_2:d_2}_{(?,\ell_2)} \xrightarrow[\forall s_1,s_2 \in \Sigma,\ \ \forall t_2 \in \{\square,\uparrow\},]{(s_1,\square,s_2,t_2)/(s_1,\uparrow,s_2,t_2)\ :\ H} \ldots\ldots\ldots\ldots\ldots\ldots$$

3. on effectue l'écriture e_2 et on efface la flêche $\uparrow$ qui indique la position de la tête T_2

$$\mathbf{q}^{\bullet:\bullet,e_2:d_2}_{(s_1,\ell_2)} \xrightarrow[\forall c_1,c_2 \in \Sigma,\ \ \forall t_1 \in \{\square,\uparrow\},]{(c_1,t_1,c_2,\uparrow)/\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ :\ d_2} \ldots\ldots\ldots\ldots\ldots$$

4. on inscrit la flêche $\uparrow$ qui indique la position de la tête T_2 et on termine la transition $q \to q'$ de M

$$\ldots\ldots\ldots\ldots\ldots \xrightarrow[\forall c_1,c_2 \in \Sigma,\ \ \forall t_1 \in \{\square,\uparrow\},]{(c_1,t_1,c_2,\square)/\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ :\ H} \mathbf{q}^{\bullet:\bullet,\bullet:\bullet}_{(s_1,\ldots\ldots)} \equiv \ldots\ldots_{(s_1,\ldots\ldots)}$$

En supposant que la machine M comporte la transition $\mathbf{q'} \xrightarrow[s_2/e'_2\ :\ d'_2]{s_1/e'_1\ :\ d'_1} \mathbf{q''}$, complétez l'état $\mathbf{q'}$ en indiquant en indice les symboles en face des têtes de M et en exposant les prochaines actions à effectuer.

Q21. Complétez les transitions Les transitions précédentes ne sont pas complètes : il manque les transitions qui font avancer M_τ à la recherche d'une tête de lecture.

Q22. Complétez les transitions Les transitions précédentes ne sont pas complètes : il manque les cas où l'on trouve d'abord la tête 2 avant la tête 1.

Conclusion Vous comprenez que la traduction d'une MT à 2 bandes en une MT à 1 bande est fastidieuse. Plutôt que de l'effectuer à la main il est préférable de la programmer. Ce sera l'objectif du projet dans les années à venir.

fiche TD1

MCAL/MT 2018 - série 1 - Machine de Turing (2 TD)

Exercice 1 : Machine de Turing de base

On considère l'alphabet $\Sigma = \{0, 1, \$\}$. On rappelle que $\square$ ne fait pas partie de l'alphabet.

Q1. Que fait la MT $\longrightarrow (1) \xrightarrow{\square:L} ((2))$, avec sur l'état 1 la boucle $\Sigma/\square:L$? Quel langage reconnaît-elle ?

Q2. Dessinez une machine $M_{\overrightarrow{?\square}}$ qui déplace la tête de lecture jusqu'au dernier symbole à droite différent de $\square$.

Q3. Dessinez une machine M_{eff} qui efface le ruban et termine, à condition de démarrer sur une partie non vierge du ruban. Pour cela elle remplace tous les symboles par $\square$, tout en respectant la contrainte qui interdit à tout moment d'avoir un ruban de la forme $\boxed{^\infty\square \mid \omega_1 \mid \square \mid \omega_2 \mid \square^\infty}$ Comment se comporte votre machine si on l'appelle sur le ruban vierge.

Q4. Dessinez une machine $M_{\overleftarrow{\$}}$ qui déplace la tête de lecture vers la gauche jusqu'à rencontrer le marqueur \$ ou un blanc $\square$. Elle termine dans l'état accepteur $\circledcirc$ si elle trouve le \$ et dans l'état erreur $\otimes$ sinon.

Q5. Dessinez une transition de MT qui ne fait rien. Par la suite on notera $\circ \xrightarrow{\epsilon} \circ$ ce type de transition.

Q6. Pour la MT de la question **Q2** donnez sa description sous la forme d'un sextuplet $(\Sigma, \mathscr{Q}, q_I, \delta, \mathscr{A}cc, \mathscr{E}rr)$ où Σ est l'alphabet de M, $\mathscr{Q}$ est l'ensemble des états de M, q_i son état initial, sa fonction de transition $\delta : \mathscr{Q} \times \Sigma \cup \{\square\} \to \Sigma \cup \{\square\} \times \{L, H, R\} \times \mathscr{Q}$, $\mathscr{A}cc \subseteq \mathscr{Q}$ est l'ensemble des états accepteurs, $\mathscr{E}rr \subseteq \mathscr{Q}$ est l'ensemble des états d'erreurs.

Exercice 2 : Zoom sur les macro-transitions

On considère l'alphabet $\Sigma = \{0, 1, \$\}$. On note $[n]_2$ l'écriture binaire, de gauche à droite (*ie.* avec les unités à gauche) de l'entier $n \in \mathbb{N}$. Par exemple, $[4]_2 = 001$, $[5]_2 = 101$.

Q7. Donnez une telle MT M_{inc} qui incrémente de 1 un entier n écrit sur le ruban en binaire. Vous utiliserez un état q_r pour mémoriser la retenue r à propager. On autorise uniquement des transitions de la forme $q \xrightarrow{l/e:d} q'$ qui effectue à la fois une lecture, une écriture, un déplacement.

Q8. On considére un alphabet $\Sigma = \{s_1, s_2, s_3, s_4\}$. Expliquez comment réaliser les transitions suivantes à l'aide de transitions classiques : $q \xrightarrow{\{s_1,s_2\}/s_3} q'$ $\quad q \xrightarrow{\Sigma:d} q'$ $\quad q \xrightarrow[\ell \in \Sigma - s_1]{\ell/\square:d} q'$

Q9. (DS 2014) Comment traduire les transitions $\xrightarrow{s}$ d'un automate (à nombre) d'états fini A en transition de machines de Turing pour obtenir une MT M équivalente à A au sens où $\mathscr{L}(A) = \mathscr{L}(M)$.

Q10. (Projet 2014) Montrez qu'on peut traduire une MT en une MT avec uniquement les deux formes de transitions écriture ou déplacement.

Q11. (Projet 2014) Appliquez votre transformation sur la MT M_{inc}.

Exercice 3 : L'alphabet minimal Σ_2

Montrez qu'on peut transformer une MT M opérant sur un alphabet $\Sigma = \{0, 1, \$\}$ en une MT M' équivalente opérant sur l'alphabet $\Sigma_2 = \{\boxed{0}, \boxed{1}\}$.

> **Indication :** On représente les 4 symboles de $\Sigma \cup \{\square\}$ par des couples de symboles de Σ_2 *ie.*
> $0 = (\boxed{0}, \boxed{0})$, $1 = (\boxed{1}, \boxed{1})$, $\$ = (\boxed{0}, \boxed{1})$, $\square = (\boxed{1}, \boxed{0})$
> Quand la machine M fait une transition sur un symbole de Σ la machine M' fait deux transitions.

Q12. Transformez la machine $M_{\overrightarrow{?\square}}$ de la question **Q2** en une machine équivalente $M'_{\overrightarrow{?\square}}$ opérant sur Σ_2.

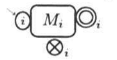
Q13. Transformez la MT M_{effG} de la question **Q1** opérant sur Σ_4 en une machine équivalente M'_{effG} opérant sur Σ_2.

Q14. (à chercher) Donnez une version optimisée de la machine M'_{effG}.

PROJET 2017 L'un des objectifs du projet est d'implanter cette transformation de Σ_4 vers Σ_2 puis de généraliser cette transformation à des MT opérant sur un alphabet Σ_{2^N} contenant 2^N symboles $s_1, \ldots, s_{2^N}$.

Exercice 4 : Exécution séquentielle de deux MT

Étant données deux MT $M_1 = (\Sigma_1, \mathscr{Q}_1, (1), \delta_1, \mathscr{A}cc_1, \mathscr{E}rr_1)$ et $M_2 = (\Sigma_2, \mathscr{Q}_2, (2), \delta_2, \mathscr{A}cc_2, \mathscr{E}rr_2)$, construire la MT notée $[M_1; M_2]$ qui exécute M_1 puis exécute de M_2 à partir de la position où s'est arrêtée M_1.

> **Indication :** Pour une instruction seuls deux états terminaux sont possibles $\circledcirc$ ou $\otimes$ en cas d'erreur.
>
> On suppose que toutes les MT M_i sont de la forme 

Exercice 5 : Reconnaissance de langages classiques

Soit l'alphabet $\Sigma = \{a, b, c\}$. Pour chacun des langages suivants, donnez une MT qui le reconnaît $L_1 = \Sigma^*$, $L_2 = \emptyset$, $L_3 = \{\epsilon\}$, $L_4 = \{a^n b^n \mid n \in \mathbb{N}\}$, $L_5 = \{\omega.R(\omega) \mid \omega \in \Sigma^*\}$ où R est l'opération qui renverse un mot et donc L_5 est l'ensemble des palindromes de longueur paire sur Σ, $L_6 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Exercice 6 : Renversement et Palindrome avec des MT à deux bandes

Les transitions d'une MT à deux bandes sont de la forme $q \xrightarrow[\ell_2/e_2:d_2]{\ell_1/e_1:d_1} q'$. La partie $\ell_1/e_1 : d_1$ concernent la bande B_1 et la partie $\ell_2/e_2 : d_2$ concerne la bande B_2.

Q15. Donnez un MT M_R qui réalise la fonction $R : \Sigma^* \to \Sigma^*$ qui renverse un mot fourni.

B_1 :	$^\infty\square$	s_1	…	s_n	$\square^\infty$
B_2 :	$^\infty\square$	$\square$	…	$\square$	$\square^\infty$

$\xrightarrow[M_R]{}{}^*$

B_1 :	$^\infty\square$	s_1	…	s_n	$\square^\infty$
B_2 :	$^\infty\square$	s_n	…	s_1	$\square^\infty$

Q16. Donnez une MT à deux bandes M_{eq} qui *décide* si les mots inscrits sur les bandes sont identiques.

> **Indication :** M_{eq} « décide » signifie que M_{eq} termine toujours et atteint ◎ si c'est vrai et un état ⊗ sinon.

Q17. À l'aides des MT précédents, donnez une MT à deux bandes M_{pal} qui accepte uniquement les mots de la forme $\omega.R(\omega)$ ou $\omega.s.R(\omega)$

Q18. Donnez une MT à *une seule bande* M'_{pal} qui accepte les mots de la forme $\omega.R(\omega)$ ou $\omega.s.R(\omega)$

Q19. Complétez $\mathscr{L}(M_{pal}) =$ ....

Exercice 7 : Des MT avec états labelisés pour mémoriser des données temporaires

On considére l'alphabet $\Sigma = \{0, 1, \$\}$. Vous avez le droit d'utiliser les MT déjà construites, l'opérateur de séquence et les transitions génériques.

Définition 1 *Un état* ⓠ ***labelisé par le symbole*** ℓ *est noté* (q_ℓ). *On garde en mémoire le symbole* ℓ *grâce au nom de l'état.*

Q20. Donnez une MT $M_{0?}$ qui teste si le ruban contient l'entier 0 à droite du symbole \$. On adopte la convention que l'entier 0 est représenté par un unique 0. La MT $M_{0?}$ accepte le mot 0 et rejette tout autre mot.

Q21. Donnez une MT M_{dec} qui décremente l'entier n écrit sur le ruban en binaire au sens suivant : $M_{dec}([n]_2) = \mathbb{V}([n-1]_2)$ si $n > 0$ et $M_{dec}(0) = \mathbb{E}\mathrm{rr}$ *ie.* qu'elle s'arrête dans l'état ⊗ pour indiquer que la fonction $M_{dec}(0)$ n'est pas définie.

> **Exemples :**
>
> - $^{\infty}\square \mid 0 \mid 0 \mid 1 \mid \square^{\infty} \xrightarrow{M_{dec}} {}^{\infty}\square \mid 1 \mid 1 \mid \square \mid \square^{\infty}$ et non $^{\infty}\square \mid 1 \mid 1 \mid 0 \mid \square^{\infty}$
> - $^{\infty}\square \mid 0 \mid 0 \mid 1 \mid 1 \mid \square^{\infty} \xrightarrow{M_{dec}} {}^{\infty}\square \mid 1 \mid 1 \mid 0 \mid 1 \mid \square^{\infty}$

Q22. Donnez une MT $M_{\overrightarrow{dec}}$ qui, à partir de la position courante, décale le contenu du ruban d'une case vers la droite et introduit un \$ à sa position de départ et place la tête sur le \$.

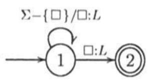
$$^{\infty}\square \mid \omega_1 \mid \underset{\uparrow}{\ell} \mid \omega_2 \mid \square \mid \square^{\infty} \quad \xrightarrow[M_{\overrightarrow{dec}}]{}{}^{*} \quad {}^{\infty}\square \mid \omega_1 \mid \underset{\uparrow}{\$} \mid \ell \mid \omega_2 \mid \square^{\infty}$$

Q23. On considère un alphabet quelconque Σ contenant au moins $\square$ et \$. Donnez une MT qui recopie la donnée du ruban (partie comprise entre les $\square$) au delà du premier $\square$ de droite et insère un séparateur \$ entre les deux copies de la donnée. Pour simplifier, on suppose que la donnée ne contient pas de symboles \$.

MCAL/MT 2018 - série 1 - Machine de Turing (2 TD)

Exercice 1 : Machine de Turing de base

On considére l'alphabet $\Sigma = \{0, 1, \square, \$\}$

Q1. Que fait la MT 

Q2. Dessinez une machine $M_{\overrightarrow{?\square}}$ qui déplace la tête de lecture vers la droite jusqu'au dernier symbole différent de $\square$.

Q3. Dessinez une machine M_{eff} qui efface le ruban et termine, à condition de démarrer sur une partie non vierge du ruban. Pour cela elle remplace tous les symboles par $\square$, tout en respectant la contrainte qui interdit à tout moment d'avoir un ruban de la forme $\boxed{{}^{\infty}\square} \boxed{\omega_1} \boxed{\square} \boxed{\omega_2} \boxed{\square^{\infty}}$ Comment se comporte votre machine si on l'appelle sur le ruban vierge.

Q4. Dessinez une machine $M_{\overleftarrow{\$}}$ qui déplace la tête de lecture vers la gauche jusqu'à rencontrer le marqueur \$ ou un blanc $\square$. Elle termine dans l'état accepteur ◎ si elle trouve le \$ et dans l'état erreur $\otimes$ sinon.

Q5. Dessinez une transition de MT qui ne fait rien. Par la suite on notera $\circ \xrightarrow{\epsilon} \circ$ ce type de transition.

Q6. Pour une des MT précédentes donnez sa description sous la forme d'un sextuplet $(\Sigma, \mathscr{Q}, q_I, \delta, \mathscr{A}cc, \mathscr{E}rr)$ où Σ est l'alphabet de M, $\mathscr{Q}$ est l'ensemble des états de M, q_i son état initial, $\delta : \mathscr{Q} \times \Sigma \to \Sigma \times \{L, H, R\} \times \mathscr{Q}$, $\mathscr{A}cc \subseteq \mathscr{Q}$ est l'ensemble des états accepteurs, $\mathscr{E}rr \subseteq \mathscr{Q}$ est l'ensemble des états erreur.

Exercice 2 : Zoom sur les macro-transitions

On considère l'alphabet $\Sigma_4 = \{\$, \square, 0, 1\}$. On note $[n]_2$ l'écriture binaire, de gauche à droite (*ie.* avec les unités à gauche) de l'entier $n \in \mathbb{N}$. Par exemple, $[4]_2 = 001$, $[5]_2 = 101$.

Q7. Donnez une telle MT M_{inc} qui incrémente de 1 un entier n écrit sur le ruban en binaire. Vous utiliserez un état q_r pour mémoriser la retenue r à propager. On autorise uniquement des transitions de la forme $q \xrightarrow{l/e:d} q'$ qui effectue à la fois une lecture, une écriture, un déplacement.

Q8. On considére un alphabet $\Sigma = \{s_1, \ldots, s_4\}$. Expliquez comment réaliser les transitions suivantes à l'aide de transitions classiques : $q \xrightarrow{\{s_1,s_2\}/s_3} q'$ $\quad q \xrightarrow{\Sigma:d} q'$ $\quad q \xrightarrow[\ell \in \Sigma - \$]{\ell/\square:d} q'$

Q9. (DS 2014) Comment traduire les transitions $\xrightarrow{s}$ d'un automate (à nombre) d'états fini A en transition de machines de Turing pour obtenir une MT M équivalente à A au sens où $\mathscr{L}(A) = \mathscr{L}(M)$.

Q10. (Projet 2014) Montrez qu'on peut traduire une MT en une MT avec uniquement les deux formes de transitions écriture ou déplacement.

Q11. (Projet 2014) Appliquez votre transformation sur la MT M_{inc}.

Exercice 3 : L'alphabet minimal Σ_2

Montrez qu'on peut transformer une MT M opérant sur un alphabet $\Sigma_4 = \{\$, \square, 1, 0\}$ en une MT M' équivalente opérant sur l'alphabet $\Sigma_2 = \{\boxed{0}, \boxed{1}\}$.

> **Indication :** On représente les 4 symboles de Σ_4 par des couples de symboles de Σ_2 *ie.* $0 = (\boxed{0}, \boxed{0})$, $1 = (\boxed{1}, \boxed{1})$, $\$ = (\boxed{0}, \boxed{1})$, $\square = (\boxed{1}, \boxed{0})$
> Quand la machine M fait une transition sur un symbole de Σ_4 la machine M' fait deux transitions.

Q12. Transformez la machine $M_{\overrightarrow{?\square}}$ de la question **Q2** en une machine équivalente $M'_{\overrightarrow{?\square}}$ opérant sur Σ_2.

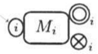
Q13. Transformez la MT M_{effG} de la question **Q1** opérant sur Σ_4 en une machine équivalente M'_{effG} opérant sur Σ_2.

Q14. (à chercher) Donnez une version optimisée de la machine M'_{effG}.

PROJET 2017 L'un des objectifs du projet est d'implanter cette transformation de Σ_4 vers Σ_2 puis de généraliser cette transformation à des MT opérant sur un alphabet Σ_{2^N} contenant 2^N symboles $s_1, \ldots, s_{2^N}$.

Exercice 4 : Exécution séquentielle de deux MT

Étant données deux MT $M_1 = (\Sigma_1, \mathscr{Q}_1, \searrow\!\!(1), \delta_1, \mathscr{A}cc_1, \mathscr{E}rr_1)$ et $M_2 = (\Sigma_2, \mathscr{Q}_2, \searrow\!\!(2), \delta_2, \mathscr{A}cc_2, \mathscr{E}rr_2)$, construire la MT notée $[M_1; M_2]$ qui exécute M_1 puis exécute de M_2 à partir de la position où s'est arrêtée M_1.

> **Indication :** Pour une instruction seuls deux états terminaux sont possibles ◎ ou $\otimes$ en cas d'erreur.
>
> On suppose que toutes les MT M_i sont de la forme 

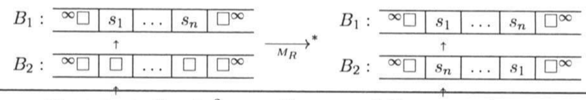
Exercice 5 : Reconnaissance de langages classiques

Soit l'alphabet $\Sigma = \{a, b, c\}$. Pour chacun des langages suivants, donnez une MT qui le reconnaît $L_1 = \Sigma^*$, $L_2 = \emptyset$, $L_3 = \{\epsilon\}$, $L_4 = \{a^n b^n \mid n \in \mathbb{N}\}$, $L_5 = \{\omega.R(\omega) \mid \omega \in \Sigma^*\}$ où R est l'opération qui renverse un mot et donc L_5 est l'ensemble des palindromes de longueur paire sur Σ, $L_6 = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Exercice 6 : Renversement et Palindrome avec des MT à deux bandes

Les transitions d'une MT à deux bandes sont de la forme $q \xrightarrow[\ell_2/e_2:d_2]{\ell_1/e_1:d_1} q'$. La partie $\ell_1/e_1 : d_1$ concernent la bande B_1 et la partie $\ell_2/e_2 : d_2$ concerne la bande B_2.

Q15. Donnez un MT M_R qui réalise la fonction $R : \Sigma^* \to \Sigma^*$ qui renverse un mot fourni.



Q16. Donnez une MT à deux bandes M_{eq} qui *décide* si les mots inscrits sur les bandes sont identiques.

Indication : M_{eq} « décide » signifie que M_{eq} termine toujours et atteint ◎ si c'est vrai et un état ⊗ sinon.

Q17. À l'aides des MT précédents, donnez une MT à deux bandes M_{pal} qui accepte uniquement les mots de la forme $\omega.R(\omega)$ ou $\omega.s.R(\omega)$

Q18. Donnez une MT à *une seule bande* M'_{pal} qui accepte les mots de la forme $\omega.R(\omega)$ ou $\omega.s.R(\omega)$

Q19. Complétez $\mathscr{L}(M_{pal}) =$ ....

Exercice 7 : Des MT avec états labelisés pour mémoriser des données temporaires

On considère l'alphabet $\Sigma = \{0, 1, \$, \square\}$. Vous avez le droit d'utiliser les MT déjà construites, l'opérateur de séquence et les transitions génériques.

Définition 1 *Un état* ⓠ ***labelisé par le symbole*** ℓ *est noté* (q_ℓ). *On garde en mémoire le symbole* ℓ *grâce au nom de l'état.*

Q20. Donnez une MT $M_{0?}$ qui teste si le ruban contient l'entier 0 à droite du symbole \$. On adopte la convention que l'entier 0 est représenté par un unique 0. La MT $M_{0?}$ accepte le mot 0 et rejette tout autre mot.

Q21. Donnez une MT M_{dec} qui décremente l'entier n écrit sur le ruban en binaire au sens suivant : $M_{dec}([n]_2) = \mathbb{V}([n-1]_2)$ si $n > 0$ et $M_{dec}(0) = \mathbb{E}\mathrm{rr}$ *ie.* qu'elle s'arrête dans l'état ⊗ pour indiquer que la fonction $M_{dec}(0)$ n'est pas définie.

Exemples :

- $\begin{array}{|c|c|c|c|c|}\hline {}^\infty\square & 0 & 0 & 1 & \square^\infty \\ \hline\end{array} \xrightarrow{M_{dec}} \begin{array}{|c|c|c|c|c|}\hline {}^\infty\square & 1 & 1 & \square & \square^\infty \\ \hline\end{array}$ et non $\begin{array}{|c|c|c|c|c|}\hline {}^\infty\square & 1 & 1 & 0 & \square^\infty \\ \hline\end{array}$
- $\begin{array}{|c|c|c|c|c|c|}\hline {}^\infty\square & 0 & 0 & 1 & 1 & \square^\infty \\ \hline\end{array} \xrightarrow{M_{dec}} \begin{array}{|c|c|c|c|c|c|}\hline {}^\infty\square & 1 & 1 & 0 & 1 & \square^\infty \\ \hline\end{array}$

Q22. Donnez une MT $M_{\overrightarrow{dec}}$ qui, à partir de la position courante, décale le contenu du ruban d'une case vers la droite et introduit un \$ à sa position de départ et place la tête sur le \$.

$$\begin{array}{|c|c|c|c|c|c|}\hline {}^\infty\square & \omega_1 & \ell & \omega_2 & \square & \square^\infty \\ \hline \multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{\uparrow} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \end{array} \xrightarrow[M_{\overrightarrow{dec}}]{}{}^{*} \begin{array}{|c|c|c|c|c|c|}\hline {}^\infty\square & \omega_1 & \$ & \ell & \omega_2 & \square^\infty \\ \hline \multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{\uparrow} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \end{array}$$

Q23. On considère un alphabet quelconque Σ contenant au moins $\square$ et \$. Donnez une MT qui recopie la donnée du ruban (partie comprise entre les $\square$) au delà du premier $\square$ de droite et insère un séparateur \$ entre les deux copies de la donnée. Pour simplifier, on suppose que la donnée ne contient pas de symboles \$.

05/05/2018

TURING

1

CM

Les busy BEAVERS (sujet DS 2018)

On considère des MT qui operent sur l'alphabet

$\Sigma = \{1\}$

<u>Definition</u> : La longueur de l'execution d'une MT est le nombre de transitions qu'elle effectue avant de s'arreter.

$M(w) \xrightarrow[\ell]{} ◎$

La longueur de l'execution de M sur le mot w est ℓ.

<u>Definition</u> : (BB_h) Parmi les MT à h-etat $q_1, \dots, q_h$

BB_h est la MT qui a la plus longue execution sur le ruban vide.

$BB_h(\varepsilon) \xrightarrow{\ell} ◎$

// execution la plus longue.

Pour toutes autre MT M_h à h états.

$M_h(\varepsilon) \xrightarrow{\ell'} ◎$

<u>Exemple</u> : $BB_4(\varepsilon) \xrightarrow{107} ◎$

$BB_6(\varepsilon) \xrightarrow{3.15.10^{10387}} ◎$

On en deduis que la fonction LEBB : $\mathbb{N} \longrightarrow \mathbb{N}$

longueur execution busy beaver $h \longrightarrow |BB_h(\varepsilon)| = \ell$

si $BB_h(\varepsilon) \xrightarrow{\ell} ◎$

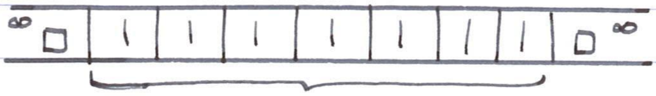
<u>Proposition</u> : La fonction LEBB n'est pas calculable. Autrement dit, il n'existe pas de MT M_{LEBB} qui réalise la fonction LEBB.

Preuve par contradiction : Supposons que MLEBB existe et montrons qu'on obtient une contradiction.

On peut choisir de concevoir M_{LEBB} operant sur $\Sigma = \{1\}$, on peut réaliser la fonction :

$LEBB : [N] \longrightarrow [N]_1$

Si M_{LEBB} existe elle a un certain nombre d'états n.
Que donne $M_{LEBB}(\underbrace{|||||}_{[k]_1})$, elle inscrit sur le ruban $|BB_k(\varepsilon)|$ en base 1.



$$[\ell]_1 = |BB_k(\varepsilon)|$$

il faut au minimum ℓ étapes pour inscrire le résultat sur le ruban
Appelons M_{LEBB} avec $\underbrace{|||||\ldots|||}_{n+1}$, M_{LEBB} inscrit sur le ruban

$|BB_{n+1}(\varepsilon)|$ sous forme de bâtons. Pour cela, M_{LEBB} fait au minimum $|BB_{n+1}(\varepsilon)|$ étapes pour inscrire les bâtons.

Contradiction.

$$BB_{n+1}(\varepsilon) \xrightarrow{\ell} \circledcirc$$

$$\underbrace{M_{LEBB}}_{n}(1^{n+1}) \xrightarrow{\ell \leq} \circledcirc$$

l'idée c'est que M_{LEBB} a n états aurait une exécution plus longue que BB_n sur le ruban vide.
ce qui contredit la définition de BB_n.
Conclusion : On a supposé que M_{LEBB} existait (e), on aboutit à une contradiction, donc M_{LEBB} n'existe pas et la fonction LEBB n'est pas calculable.

Ensemble dénombrable ($\mathbb{N}^*$) / non dénombrable ($\mathbb{N} \to \mathbb{N}$)

$\mathbb{N} \simeq E$ $\qquad\qquad$ $\mathbb{N} \simeq \mathbb{N} \times \mathbb{N}$

$\qquad\qquad\qquad\qquad n \longleftrightarrow (i, j)$

Le principe de numérotation de cantor donne explicitement les bijections de $\mathbb{N} \to \mathbb{N}^p$. Cas simples de la bijection de $\mathbb{N} \longleftrightarrow \mathbb{N} \times \mathbb{N}$

05/05/2010
TURING

(1) (CM) On construit le tableau infini des couples d'entiers. $[0 \dots \mathbb{N}[\times [0 \dots \mathbb{N}[$

	0	1	2	3	4	5	… $\mathbb{N}$
0	(0,0) 0	(0,1) 2	(0,2) 5	(0,3) 9	(0,4) 14	(0,5) 20	
1	(1,0) 1	(1,1) 4	(1,2) 8	(1,3) 13	(1,4) 19	(1,5) 26	
2	(2,0) 3	(2,1) 7	(2,2) 12	(2,3) 18	(2,4) 25	(2,5) 32	
3	(3,0) 6	(3,1) 11	(3,2) 17	(3,3) 24	(3,4) 31	(3,5) 38	
4	(4,0) 10	(4,1) 16	(4,2) 23	(4,3) 30	(4,4) 37	(4,5) 44	
5	(5,0) 15	(5,1) 22	(5,2) 29	(5,3) 36	(5,4) 43	(5,5) 50	
⋮							
$\mathbb{N}$							

On les numerote (rouge)

Généralisation : $\mathbb{N} \simeq \mathbb{N} \times \underbrace{(\mathbb{N} \times \mathbb{N})}_{\simeq \mathbb{N}}$

$$\underbrace{\mathbb{N} \times \mathbb{N}}_{\simeq \mathbb{N}}$$

$$n_3 \overset{?}{=} (i, j, k) \equiv (i, \underbrace{(j, k)}_{\simeq n_2})$$

$$\equiv (i, n_2) \simeq n_3$$

$$n_3 \simeq_2 (i, n_2)$$
$$\simeq (j, k)$$
$$\Downarrow_3 \; (i, j, k)$$

Pour les vecteurs $\mathbb{N}^p$ on procede de la meme façon en remarquant que $(i_1, i_2, \dots, (i_{p-2}, (i_{p-1}, i_p)))$

conclusion : $\mathbb{N} \simeq \mathbb{N}^1$
$\mathbb{N} \simeq \mathbb{N}^2$
$\mathbb{N} \simeq \mathbb{N}^3$
$\mathbb{N} \simeq \mathbb{N}^4$

$$\mathbb{N} \simeq \mathbb{N}^* \supseteq \{ a, \;\; , z, \sqcup, \{, \}, (,), := \}$$
$$\supseteq \cup$$

Proposition : $\mathbb{N}^*$ = les listes d'entier, les mots, les textes, les programmes, les MT écrites sur un alphabet fini sont denombrables en bijection avec $\mathbb{N}$.

La prochaine fois on voit que les fonctions $f : \mathbb{N} \to \mathbb{N}$ sont non - denombrable.

Le modele de la machine chimique.
Les algorithmes : réactions chimiques

chimie : $\underbrace{H, O, H}_{\text{réactif.}} \underbrace{\xrightarrow[C^\circ < 10^6]{}}_{\text{réactif.}} \underbrace{H_2O}_{\text{produit.}}$

algo : $x, y \xrightarrow[x < y]{} y - x$

<u>Données</u> : $\{1, 2, 0, 1, 1, 0, 3\}$

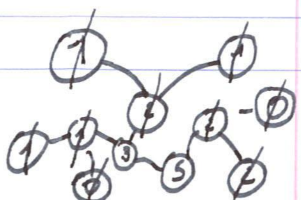
<u>Atomes</u> $\mathbb{N}$, $\mathbb{B}$ool, $\mathbb{C}$har

<u>Molecules</u> : sont des termes formés à l'aide de constructeurs écrits en majuscule. exemple : $C(\ldots, \ldots)$
$A(\ldots, \ldots)$

permet de faire : | des couples $C(1, 2)$
| des triplets $C(1, C(3, 4))$

<u>Les conditions</u> : de la réaction $\xrightarrow{\text{cond}}$:
- predicats usuels : $x = y \wedge x < z \wedge x \bmod 3 = u$
- pattern matching : $x \xrightarrow{x \neq C(\ldots,\ldots)} x + 1$

<u>Exercice</u> : $[\![1, 1, 0, 2, 0, 1]\!]$ $\quad x, y \longrightarrow x + y$: $[\![5]\!]$
en 3 instants



05/05/2018

TURING

3 CM $[\![\, (C(1,0), C(2,2), C(C(3,1), C(3,1)) \,]\!]$

$$C(x, y) \xrightarrow[1]{x \neq y} x, y$$

$$C(x, y) \xrightarrow[2]{x = y} x$$

01/02/2018

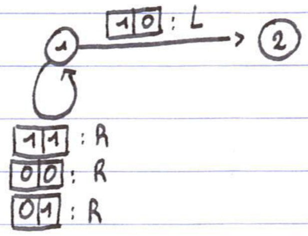
TURING

TD

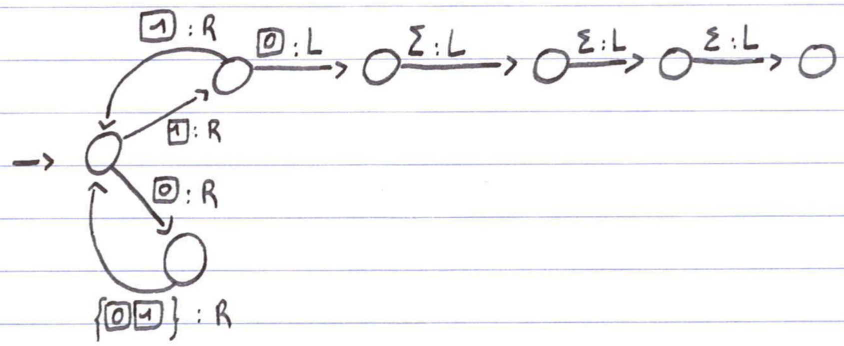
Séance 2 - Suite de la fiche de la séance 1 -

Exercice 3 - question 12

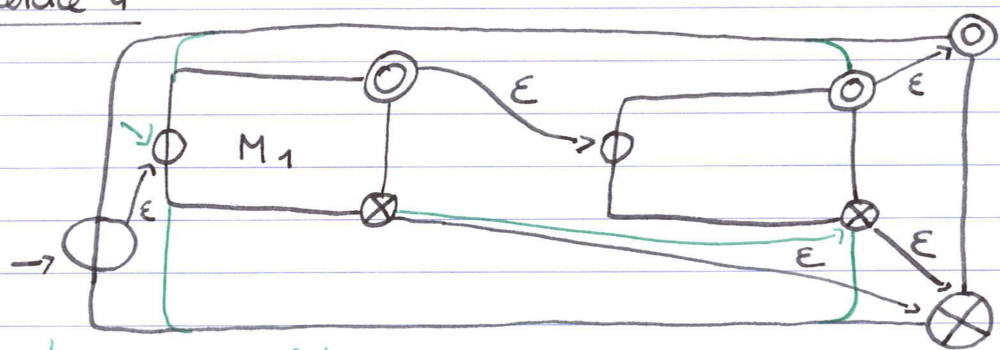
idée :



traduction de l'idée en MT fonctionelle :



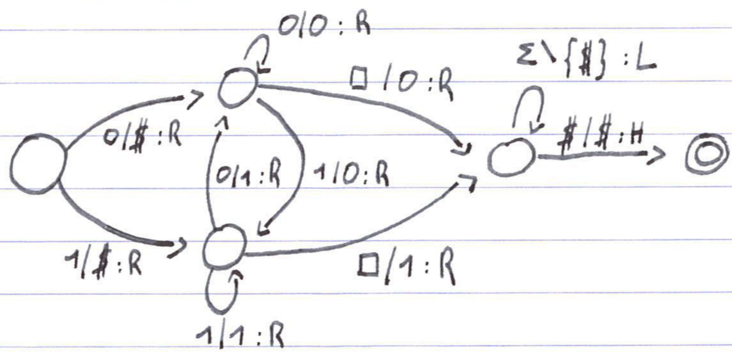
Exercice 4 -



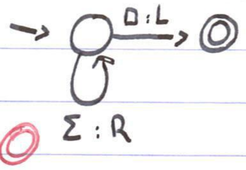
-> deuxieme solution.

-> on efface les etats accepteurs et erreur des machines intermediaires apres les avoir reliés.

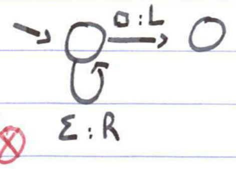
Exercice 7 - question 21 -



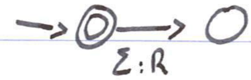
Excercice 5 -

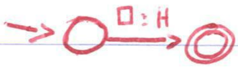
L_1 = 

L_1 = 

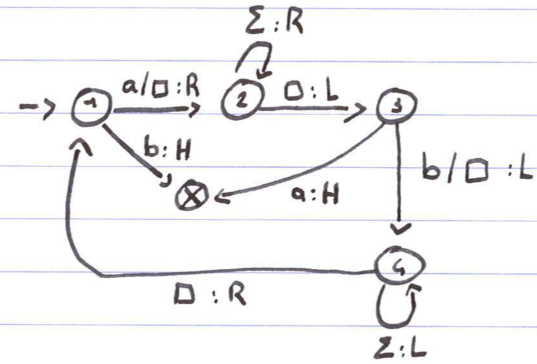
L_2 = 

L_2 = 

L_3 = 

L_3 = 

L_4 : Principe : pour chaque a lu, on va effacer un b à l'autre extrémité du mot.



Excercice 6 -

29/01/2018
TURING

1 CM

Séance ≃ 3 - CM machine de Turing -

DS le vendredi avant les vacances.

Calculabilité.

Les fonctions $F : \mathbb{N}^p \longrightarrow \mathbb{N}^*$

$$f(x_1, \dots, x_p) = () \quad (x_1) \quad \dots$$

<u>Définition</u> - On note $\mathbb{N}^* = \bigcup_{i \in \mathbb{N}} \mathbb{N}^i$

$\mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$ = couple d'entiers $= \{(m_1, m_2) \mid m_1, m_2 \in \mathbb{N}\}$

$\mathbb{N}^3 = \mathbb{N} \times \mathbb{N} \times \mathbb{N}$: triplet d'entiers

$\mathbb{N} = \mathbb{N} \times \dots \times \mathbb{N}$ = i-uplet d'un entiers

$\mathbb{N}^0 = \{()\}$ = contient le vecteur à 0 elements, dit vecteur vide.

<u>Exemple</u> : $\mathbb{N}^1 = \{(0), (1), (2), \dots\} \neq \mathbb{N}$

<u>Exemple de fonction</u> :

inc : $\mathbb{N}^1 \longrightarrow \mathbb{N}^1$
$\quad m \longrightarrow (m+1)$

euclide : $\mathbb{N}^2 \longrightarrow \mathbb{N}^2$
$\quad (a,b) \longrightarrow (q, r)$ tel que $a = b \times q + r$ et $r < b$

sub : $\mathbb{N}^2 \longrightarrow \mathbb{N}^*$
$\quad (a,b) \longrightarrow (a-b) \in \mathbb{N}^1$ si $a > b$
$\quad\quad\quad\quad ()$ sinon

dfp : $\mathbb{N}^1 \longrightarrow \mathbb{N}^*$
$\quad (m) \longrightarrow (p_1, p_2, \dots, p_k)$ tel que $m = p_1 \times \dots \times p_k$ et p_i premier

Codage en binaire des vecteurs de $\mathbb{N}^*$

but : codar un vecteur (1, 2, 5) en binaire sur un ruban.

On utilise l'alphabet $\{ (,), ,, 0, 1 \}$: 5 symboles.

on a $2^2 \leq 5 \leq 2^3$ donc on peut rajouter des symboles : □, \$, §

… □	(	1	,	2	,	5	)	□ …

binaire $\hookrightarrow \Sigma_8 = \Sigma_2 \times \Sigma_2 \times \Sigma_2$ où $\Sigma_8 = \{ (,), ,, 0, 1, □, \$, § \}$

$\Sigma_2 = \{\boxed{0}, \boxed{1}\}$

0 $\longrightarrow$	0 0 0	
1 $\longrightarrow$	1 1 1	
($\longrightarrow$	1 1 0	
) $\longrightarrow$	0 1 1	
; $\longrightarrow$	0 1 0	
# $\longrightarrow$		
§ $\longrightarrow$		
□ $\longrightarrow$	1 0 0	

En binaire on concatene tout les codages.

<u>Conclusion</u>: On peut voir toutes les fonctions
$f : \mathbb{N}^p \longrightarrow \mathbb{N}^*$
comme des fonctions travaillant en binaire
$f : \Sigma_2^* \longrightarrow \Sigma_2^*$ où $\Sigma_2 = \{0, 1\}$

au lieu d'ecrire $f(\vec{d}) = \vec{r}$ où $w_d = [\vec{d}]_2$
on écrira $f(w_d) = w_r$ $\quad w_r = [\vec{r}]_2$

Fonctions calculables -

<u>Definition</u> : Les fonctions calculables sont les Machines de Turing qui s'arrentent pour toute entrée binaire $w_d \in \{0, 1\}^*$

<u>Definition</u> : Une MT M_f ~~att~~ réalise / implemente une fonction $f : \Sigma_2^* \longrightarrow \Sigma_2^*$ si
1. l'execution de M_f s'arrête pour toute entrée $w_d \in \Sigma_2^*$
2. $M_f(w_d) = V(w_r)$ pour toutes entrées w_d tel que $f(w_d) = w_r$
3. $M_f(w_d) =$ Error pour toutes entrées pour laquelle f n'est pas définie.

<u>Exemple</u> : $decr : \mathbb{N}^1 \longrightarrow \mathbb{N}^1$
$(0) \longrightarrow$ pas definie
$M_{decr}(0) \longrightarrow \otimes$

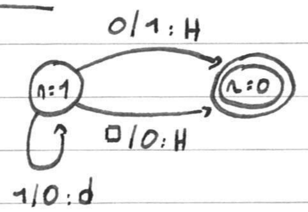
<u>Exercice</u> : Donner une MT M_{next} qui réalise la fonction next
$next : \Sigma^* \longrightarrow \Sigma^*$ qui enumere tout les mots binaires possibles.

23/01/2018

TURING

② EM

$\varepsilon \to 0$	$M_{next} = M_{inc}$
$0 \to 1$	sinon on manquerait les mots avec 0 non significatif
$1 \to 00$	00, 000, ..
$00 \to 10$	<u>M next</u> :
$10 \to 01$	
$01 \to 11$	
$11 \to 000$	



Ensemble dénombrable / non dénombrable

but : nombre de MT possible $\infty \ll$ nombre de fonction $f : \mathbb{N}^p \to \mathbb{N}^k$

<u>Definition</u> : Un ensemble E est dénombrable si il est en bijection avec $\mathbb{N}$.
On le note $E \simeq \mathbb{N}$. Autrement dit, à chaque entier i on peut associer un element e_i de E.
c'est la fonction get : $\mathbb{N} \to E$
$i \to E_i$

<u>Théoreme de cantor - bernstein</u>

il existe une bijection $E \simeq \mathbb{N}$ si et seulement si il existe deux injections
$h : E \to \mathbb{N}$
$g : \mathbb{N} \to E$

⚠ Le théoreme ne dit pas que h et g forment une bijection
$h(g(i)) \neq i$ (pas forcement) $\quad h^{-1} \neq h$
$g(h(e)) \neq e$ (pas forcement) $\quad g^{-1} \neq h$

<u>Application</u> : Montrons que $\mathbb{N} \simeq \mathbb{N} \times \mathbb{N}$
Donnons deux injections

$g : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$
$n \to (n, n)$

$h : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$
$(n, p) \to 2^n \times 3^p$

Rappel -

f est injective si $\forall a_1, a_2$. $a_1 \neq a_2 \Rightarrow f(a_1) \neq f(a_2)$

$$f : A \to B$$

$$\begin{array}{ccc} a_1 & \to & f(a_1) \\ \neq & & \neq \\ a_2 & \to & f(a_2) \end{array}$$

→ pas de collision lors de la convertion

f est surjective si $\forall b \in B$, $\exists a \in A$ tel que $f(a) = b$

convertion $f : A \to B$
qui garantie que toutes données B provient de la convertion d'une donnée $a \in A$

f est bijective si elle est injective et surjective
: conversion inversible.

25/01/2018

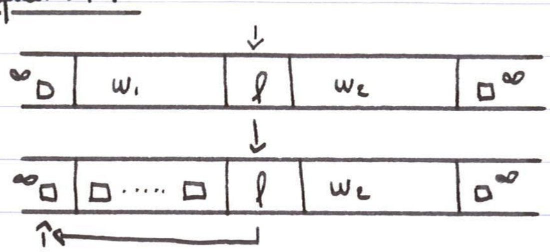
TURING

Machine de Turing TD : séance 1.

(1) TD

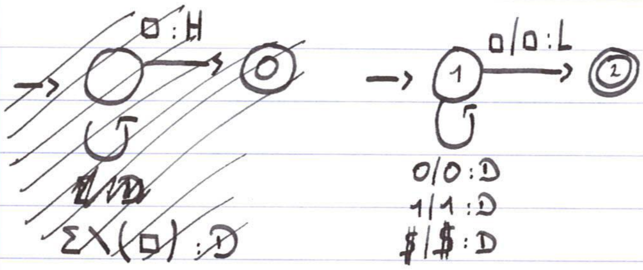
Exercice 1 - Machine de Turing de base

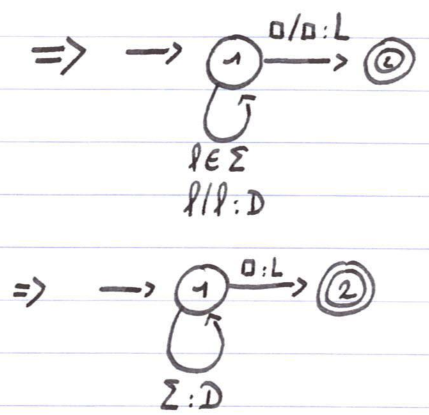
question 1 -



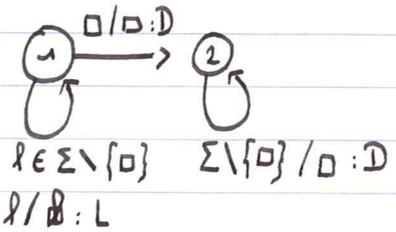
La MT efface le ruban à gauche de la position initiale et lis le mot. quel que soit le ruban au départ, la MT M_1 s'arrête dans un état accepteur donc $\mathcal{L}(M_1) = \Sigma^*$

question 2 -





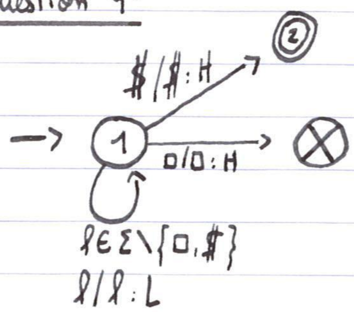
question 3 -



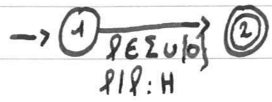
lis à gauche jusqu'à la fin du mot, puis efface tout le mot de gauche à droite.

Si elle demarre sur un blanc, elle effacera à droite jusqu'au prochain blanc.

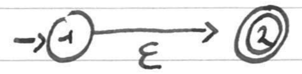
question 4 -



question 5 -



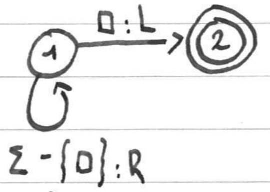
⟺



question 6

Σ alphabet
$\mathcal{L}$ ensemble des etats
q_i etat initial
δ sa fonction de transition
$Acc \in L$ ensemble etat accepteu
Err ensemble des etat d'errem $\in L$

pour



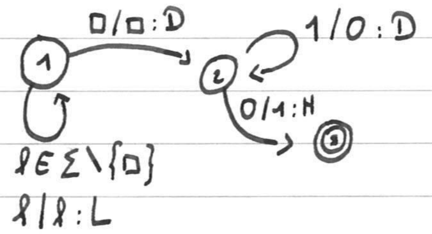
$(\{1, 0, \square\}, \{q_1, q_2\}, q_1, \delta, \{2\}, \{\})$

avec δ : $\delta(q_1, \ell) = (\ell, R, q_1), \ \ell \in \Sigma - \{\square\}$

$\delta(q_1, \square) = (\square, L, q_2)$

Exercice 2 - Zoom sur les macro-transitions

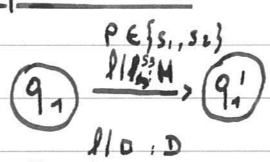
question 7 :

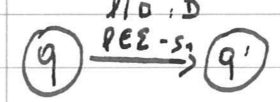


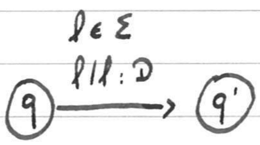
On va à gauche de l'entier à lire, puis on lis l'entier de gauche à droite : si on ~~met~~ lit un 1, on ecrit un 0 et on lit la suite, si c'est un 1 on recommence et sinon on écrit un 1.

-> ne traite pas le cas ou le ruban est à rempli de 1.

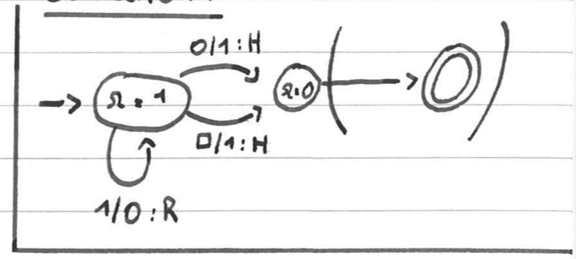
question 8 -



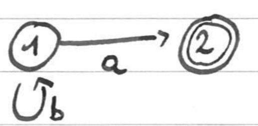




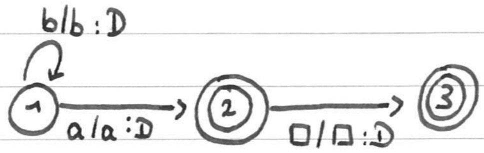
correction :



question 9 - sur $\Sigma = \{a, b\}$



⟺



La MT ne sait pas si on est à la fin du mot

$\mathcal{L}(A) = b^*.a = \{ b^n . a \mid n \in \mathbb{N} \}$

en MT :

$^{\infty}\square$	b	b	b	b	b	a	$\square^{\infty}$

22/01/2018

TURING

① CM

Rappel :

$^{\infty}\square$	w_1	l	w_2	$\square^{\infty}$

$\equiv (w_1, q, l.w_2)$

Definition : Configuration initiale.

Pour executer une MT M sur un mot w.

① À partir d'un ruban vierge
- l'utilisateur inscrit le mot w sur le ruban.
- il place la TDL (tete de lecture) sur le debut du mot et la MT (machine de turing) dans son état initial.

$^{\infty}\square$	w	$\square^{\infty}$

↳ q initial

$\equiv (\varepsilon, q_{initial}, w)$

② Le mot w a été inscrit sur le ruban par une precedente MT
- La MT precedente doit placer la TDL au bon endroit.

$^{\infty}\square$	w_1	#	w	$\square^{\infty}$

↑

Il est parfois utile de mettre un marqueur # pour proteger les données écrites avant #.
- La MT M démarre dans son état initial.

Definition : Configuration terminale : Une configuration est terminale (w_1, q, w_2) si q est un état terminal $q \not\to$: il n'y a pas de transitions possible à partir de q

Definition : Une execution d'une MT M est une suite de configurations $(w_0, q_0, w_0') \xrightarrow[M]{} (w_1, q_1, w_1') \dots \to (w_m, q_m, w_m') \to (\xrightarrow[m]{} \infty)$ qui commence avec une configuration initiale et passe d'une configuration à une autre en effectuant une transition de M. L'execution d'une MT peut être infinie.

Definition : Pour qu'une execution M(w) rende un resultat, il faut atteindre une configuration terminale

3 cas possible :

① $M(w) = V(w_2) \equiv (\varepsilon, q, w) \xrightarrow[M]{*} (w_1', q \not\to, w_2')$

si la MT M termine dans un etat accepteur (V) : execution normale et le resultat est w_2'.

② $M(w) = \mathbb{F}(w_2') \equiv (\varepsilon, \rightarrow q, w) \xrightarrow[M]{}^{*} (w_1', q_t \not\to, w_2')$
si la MT termine dans un état non-accepteur $\mathbb{F}$. exécution normale et le résultat est w_2'.

③ $M(w) = \text{Error} \equiv (\varepsilon, \rightarrow q, w) \xrightarrow[M]{}^{*} (w_1', \otimes \not\to, w_2')$
Le constructeur Error signale une erreur et le mot w_2' ne doit pas être considéré comme un résultat valide.

Une exécution infinie :
$(\varepsilon, \rightarrow q, w) \longrightarrow^{*\infty}$ sera noté $M(w) = ?$

<u>Définition</u> : Langage associé à une MT.
C'est l'ensemble des mots w pour lesquels l'exécution $M(w)$ s'arrête dans un état accepteur

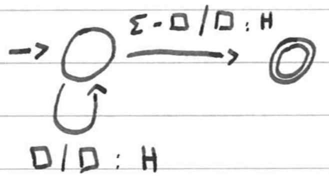
$$\mathcal{L}(M) = \{ w \in \Sigma^* \mid M(w)\ \mathbb{V} \}$$

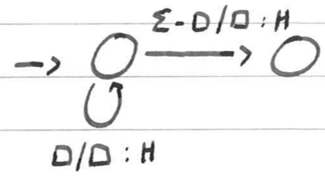
$$= \{ w \in \Sigma^* \mid (\varepsilon, \rightarrow \bigcirc, w) \xrightarrow[M]{}^{*} (w_1', \circledcirc, w_2') \}$$

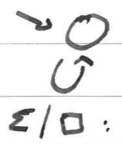
Rappel
$\mathcal{L}(\rightarrow \circledcirc \circlearrowleft a) = \{ a^n \mid n \in \mathbb{N} \}$

⚠ $w \notin \mathcal{L}(M)$ si $M(w) = \mathbb{F}$
ou bien $M(w) = \text{Error}$
ou bien $M(w) = ?$
} ce qu'on ne résume pas. $M(w) \neq \mathbb{V}$

<u>Application</u> : Donner une MT M qui ne termine pas pour ε
ie. $M(\varepsilon) = ?$ où $\varepsilon \equiv {}^{\infty}\square \mid \square \mid \square^{\infty}$
mais qui termine pour tout autre mot.

M_1 :


M_2 :


MT qui ne termine jamais :


, langage : $\{\}$

MT équivalente qui termine :


22/01/2018

- TURING

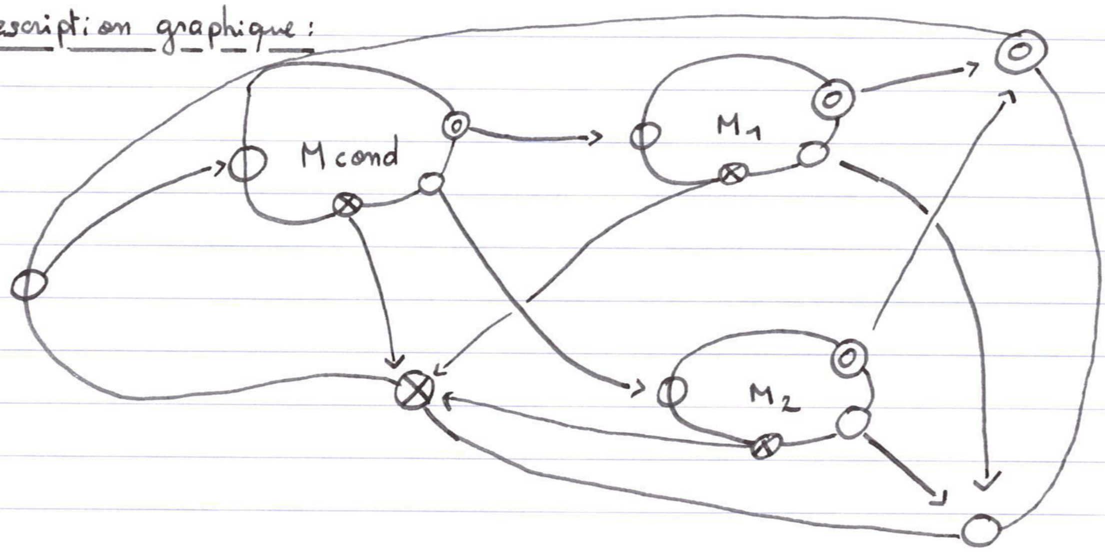
(2) (CM) Programmer avec une MT.

- Les fonctions et les instructions sont des MT basiques.
  M_{inc} , M_{dec} , $M_{\overleftarrow{\#}}$, $M_{\overrightarrow{\square}}$, $M_{?\overrightarrow{\square}}$

- Les predicats sont des MT qui rendent :
  $\mathbb{V}$ si $M(w) \to^* ◎$
  $\mathbb{F}$ si $M(w) \to ○$
  Error si $M(w) \to ⊗$

- La conditionelle $M \stackrel{def}{=} [$if (M_{cond}) then M_1 else $M_2]$

Description graphique :



Les transitions qu'on a ajoutées $\longrightarrow$ correspondent à $\square/\square:H$, $1/1:H$, $0/0:H$

ce que l'on note $\dfrac{\ell/\ell:H}{\ell \in \Sigma}$

- La sequence $M \stackrel{def}{=} [M_1 ; M_2 , M_3]$ en $\neg$ ~~B~~ $\square$

- L'itérateur $M \stackrel{def}{=} [$while $(m_{cond}$ { M_{body} } $]$ cf. 2015

- Les variables i, j, ... : chaques variable est representé par un ruban sur lequel est inscrit la valeur.

B_i = | $\infty\square$ | w_i | $\square\infty$ |

B_j = | $\infty\square$ | w_j | $\square\infty$ |

On peut transformer une MT à plusieurs bandes en une MT classique à un seul ruban.
- l'affectation $i := w$ consiste à recopier le mot w sur la bande B_i

15/04/2018

TURING

04 CM

Rappels sur les Automates à états finis

- langage : ensemble de mots écrits sur un alphabet Σ.
- Σ^* est l'ensemble de tout les mots possible
- Tout langage L est un sous ensemble de Σ^* : $L \subseteq \Sigma^*$

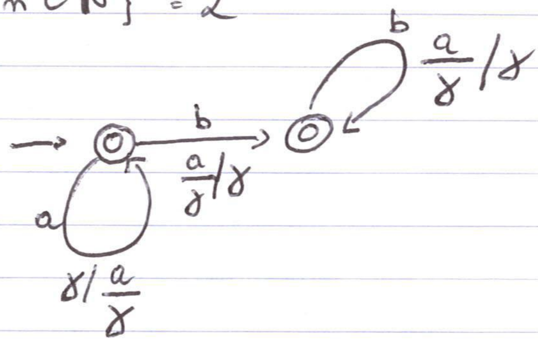
Les automates à nombre d'états finis (AEF) reconnaissent les langages réguliers.
Les AEF ne reconnaissent pas tous les langages, en particulier sur $\Sigma = \{a, b\}$.
Le langage $\{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas reconnaissable par AEF

Les automates à une pile (AUP) reconnaissent les langages dits algebriques, en particulier $\{a^n b^n \mid n \in \mathbb{N}\}$ mais ils ne reconnaissent pas tout les langages. En particulier $\{a^n b^n c^n \mid n \in \mathbb{N}\}$

Principe sur un exemple :
AUP reconnaissant $\{a^n b^n \mid n \in \mathbb{N}\} = \mathcal{L}$

pile : γ 

γ / γ
pile avant / pile après



Machine de Turing - (MT)

Extension des AUP

① Les MT possedent une tete de lecture / ecriture qui se deplace sur un ruban.
② Le mot à reconnaitre est ~~ec~~ écrit sur le ruban.
Ces deux extensions mineures donnent aux MTs la "puissance du calculable"

plus etonnant : Si on ajoute une deuxieme pile aux AUP, on atteind la "puissance du calculable"
(en TD)

Exercice à chercher (**) : Donner un AUP qui reconnait $\{w \in \{a,b\}^* \mid |w|_a = |w|_b\}$

Representation d'une MT

- On considère l'alphabet $\Sigma = \{0, 1, \square, \$\}$
- Un ruban vierge est noté $^{\infty}\square \,|\, \square^{\infty}$
- Les transitions sont de la forme :

$q \xrightarrow{l/e} q'$: dans l'état q, si on lit un l sur le ruban, on ecrit un e et on passe en q' $^{\infty}\square \,|\, w \,|\, e \,|\, w' \,|\, \square^{\infty}$

$q \xrightarrow{l:d} q'$: " " , on effectue le deplacement d de la tete et on passe en q'

$d \in \{L, H, R\}$

left here rigth

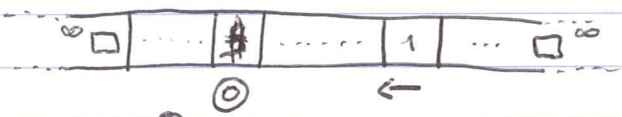
$q \xrightarrow{l/e:d} q'$: lire l, ecrit e, effectue le deplacement d.

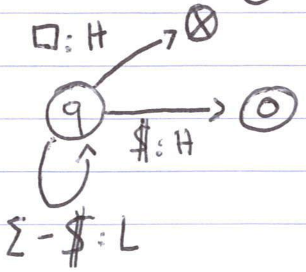
Les états d'une MT (comme AEF)

- etat initial noté $\rightarrow q$
- un états accepteur ◎
- des états non-accepteurs ○
- eventuellment un etat erreur / rejet ⊗

un état est dit terminal s'il n'a pas de transitions sortantes, noté $q \not\rightarrow$

Exercice : Donnez une MT qui cherche le marqueur $ vers la gauche, on suppose qu'on demarre sur un symbole $\neq \square$



: 

15/07/2018
TURING

1 Convention utilisés dans ce cours

(C1) On considère uniquement des MT déterministes, càd pas deux transitions différentes sur un même symbole lu.

(C2) au delà d'un blanc vers la droite ou la gauche, il n'y a que des blancs.
ruban ... $^{\infty}\square | w_1 | \square | w | \square^{\infty}$ ...

(C3) les MT ont - un état initial
- au plus un état accepteur
- au plus un état erreur/rejet

Notation mathématique d'une MT (cf. livre)
Pour définir précisément une MT, il faut indiquer :
- Σ, l'alphabet sur lequel on travaille
- Q, l'ensemble des états utilisés dans les transitions $Q = \{q_1, q_2, \dots\}$
- $\rightarrow q$, l'état initial
- $A \subseteq Q$: soit $A = \{\}$, soit $A = \{q_{acc}\}$ l'état accepteur éventuel.
- $ER \subseteq Q$: soit $ER = \{\}$, soit $ER = \{\otimes\}$ l'état erreur/rejet optionnel.

La fonction de transition:
$$\delta : Q \times \Sigma \longmapsto \Sigma \times \{L, H, R\} \times Q$$
$$\delta(q, \ell) = (e, d, q') = q \xrightarrow{\ell | e : d} q'$$
$$= (q, \ell | e : d, q')$$

Exécution d'une MT
Considérons une MT $M = (\Sigma, Q, \rightarrow q, A, ER, \delta)$

Définition : Une configuration $(w_1, q, \ell . w_2)$ contient toute l'information nécessaire pour la prochaine transition de la MT :
- Le ruban actuel est ... $^{\infty}\square | w_1 | \ell | w_2 | \square^{\infty}$ ...
- La tête est sur le symbole ℓ.
- La MT est actuellement dans l'état q.
- une configuration est de type $\Sigma^* \times Q \times \Sigma \times \Sigma^*$

Effet des transitions sur les configurations.

exemple : $(w_1, q, l.w_2)$ $q \xrightarrow{l/e} q'$ $(w_1, q', e.w_2)$

RUBAN : $w_1 | q | w_2$ $\qquad$ $w_1 | e | w_2$

exemple 2 : $(w_1, q, l.w_2)$ $q \xrightarrow{l:H} q'$ $(w_1, q', l.w_2)$

exemple 3 : $(w_1, q, l_1.l_2.w_2)$ $q \xrightarrow{l_1:R} q'$ $(w_1.l, q', l_2.w_2)$

$w_1 | l_1 | l_2 | w_2$ $\qquad$ $w_1 | l_1 | l_2 | w_2$

exemple 4 : $\cancel{(w_1, q', e.w_2)}$ $(w_1.l_1, q, l_2.w_2)$ $q \xrightarrow{l_2:L} q'$ $(w_1.q', l_1.l_2.w_2)$

A faire : Donnez une MT qui efface le ruban. on suppose qu'elle demarre sur un symbole $\neq$ D

http://www-verimag.imag.fr/~perin