Solution série 1 du TD logique mathématique

Exercice 1 En déroulant la machine de Turing sur la séquence donnée, on aura les configurations suivantes :
On commence par mettre l'état q_0 au début de la séquence comme suit :
$\#q_0s_0s_0s_2s_1s_2s_1s_2s_0s_2\#$
On applique l'instruction $q_0s_0Dq_1$ et on aura :
$\#s_0q_1s_0s_2s_1s_2s_1s_2s_0s_2\#$
Maintenant et comme la machine est à l'état q_1 et elle lit le symbole s_0, donc il faut exécuter l'instruction $q_1s_0Dq_1$ qui est une boucle sur les symboles s_0. Après l'exécution, on aura la configuration suivante :
$\#s_0s_0q_1s_2s_1s_2s_1s_2s_0s_2\#$
Le nouvel état de la machine est toujours le q_1 et elle lit le s_2 donc il faut exécuter l'instruction $q_1s_2Dq_1$ et on aura :
$\#s_0s_0s_2q_1s_1s_2s_1s_2s_0s_2\#$
On continu de la même manière et on aura :
$\#s_0s_0s_2s_1q_2s_2s_1s_2s_0s_2\#$
$\#s_0s_0s_2q_3s_1s_2s_1s_2s_0s_2\#$
$\#s_0s_0q_3s_2s_1s_2s_1s_2s_0s_2\#$
$\#s_0q_3s_0s_2s_1s_2s_1s_2s_0s_2\#$
$\#q_3s_0s_0s_2s_1s_2s_1s_2s_0s_2\#$
$q_3\#s_0s_0s_2s_1s_2s_1s_2s_0s_2\#$
On est à la configuration $q_3\#$ qui est la condition et on remarque qu'il n' y a pas d'action donc la machine se bloque et on dit qu'on est à l'état d'arrêt.

Exercice 2 Je vais donner deux solutions.
Solution 1 :
$q_0(a/b)Dq_1$
$q_1(a/b)Dq_1$ ces deux instructions sont utilisées pour parcourir tout le mot c'est à dire boucler jusqu'à la fin du mot.
$q_1\#Gq_2$ La fin du mot est trouvée, donc il faut revenir en arrière d'une case.
q_2aDq_3 Si le dernier symbole est un "a", on passe à droite avec l'action Dq_3.
$q_3\#Fq_4$ On va forcement tomber sur le symbole blanc qui est le $\#$ et dans ce cas on l'écrase avec le symbole "F" qui représente "Faux" c'est à dire on écrit un "F" à la place du $\#$.
q_4FArrêt La nouvelle configuration de la machine est sans action pour que la machine s'arrête.
q_2bDq_5 Si le dernier symbole est un "b" c'est à dire avec q_2 on va lire un dernier symbole qui est un "b", il faut aller à droite et changer l'état.
$q_5\#Tq_6$ En allant à droit, on trouvera un $\#$, donc il faut le remplacer par un "T" pour dire True ou un "V" pour dire Vrai.
q_6TArrêt.
Solution 2 :
q_0aDq_1 Si on lit le symbole "a", on passe à l'état q_1 c'est à dire si la machine est à l'état q_1 cela veut dire qu'elle vient de lire un "a".
q_0bDq_2 Si on lit le symbole "b", on passe à l'état q_2.

q_1aDq_1 Elle lit encore un "a", donc elle reste à l'état q_1.
q_1bDq_2 Si elle lit un "b", elle passe à l'état q_2.
q_2bDq_2 Elle lit un "b", donc elle garde l'état q_2.
q_2aDq_1 Elle lit un "a", elle passe à q_1.
$q_2\#Tq_2$ Si elle lit un # ça veut dire que la fin du mot est trouvée et l'état q_2 nous informe que le dernier symbole lu est un "b", donc elle écrit un "T" ou un "V".
q_2TArrêt.
$q_1\#Fq_1$ Le dernier symbole est un "a", donc elle écrit un "F".
q_1FArrêt.

Exercice 3 Solution :
q_0aDq_1 Le premier symbole est un "a".
q_0bDq_2 Le premier symbole est un "b".
$q_1(a/b)Dq_1$ Boucler jusqu'à la fin du mot sachant que le premier symbole du mot est un "a".
$q_1\#Gq_3$ La fin du mot est trouvée, donc il faut revenir à gauche d'une case et changer l'état.
q_3aDq_4 Le dernier symbole est un "a" sachant que le premier est un "a" aussi.
$q_4\#Tq_4$ Écrire un "T" sur le #.
q_4TArrêt.
$q_2(a/b)Dq_2$ Boucler jusqu'à la fin du mot sachant que le premier symbole du mot est un "b".
$q_2\#Gq_5$ La fin du mot est trouvée, donc il faut revenir à gauche d'une case et changer l'état.
q_5bDq_4 Le dernier symbole est un "b" sachant que le premier est un "b" aussi, il faut faire droite et aller à l'état q_4, car si la machine est à l'état q_4 et elle lit le # elle va directement le remplacer avec "T" en exécutant l'instruction $q_4\#Tq_4$.
q_3bDq_6 Si le dernier symbole est un "b" sachant que le premier est un "a", elle passe à droite et à l'état q_6 (un nouvel état).
$q_6\#Fq_6$ Écrire un "F".
q_6FArrêt.
q_5aDq_6 Le dernier symbole est un "a" sachant que le premier est un "b", passer à droite et à l'état q_6, car après la tête de la machine va trouver un # donc elle l'écrase avec un "F" en exécutant l'instruction $q_6\#Fq_6$.

Exercice 4 Je vais donner trois solutions.
Solution 1 :
q_00Dq_1
q_11Dq_2 On va lire directement les deux premiers symboles "01" sans modification car les mots commencent toujours par "01".
q_210q_3 On arrive au 3eme symbole qui est toujours un "1", on le remplace par un "0".
$q_3(0/1)Dq_3$ parcourir le reste du mot jusqu'à #.
$q_3\#$Arrêt. On arrête la machine en exécutant aucune action.
Dans cette solution qui est juste, on a utilisé 4 états "q_0, q_1, q_2, q_3" et 5 instructions.
Solution 2 :

$q_0 0 D q_1$
$q_1 1 D q_2$ On va lire directement les deux premiers symboles "01" sans modification car les mots commencent toujours par "01".
$q_2 1 0 q_3$ On arrive au 3eme symbole qui est toujours un "1", on le remplace par un "0".
$q_3$0Arrêt. Comme on ne va rien changer après, donc on bloque la machine en exécutant aucune action.
Avec la solution 2, on a gardé le même nombre d'états mais on a réduit le nombre d'instructions à 4.

Solution 3 :
$q_0 0 D q_1$
$q_1 1 D q_2$ On va lire directement les deux premiers symboles "01" sans modification car les mots commencent toujours par "01".
$q_2 1 0 q_2$ On arrive au 3eme symbole qui est toujours un "1", on le remplace par un "0" mais on garde le même état q_2 car on sait qu'on ne tombera jamais sur une configuration $q_2 0$ car le 3eme symbole est toujours un "1".
$q_2$0Arrêt.
Avec la solution 3, on a utilisé uniquement 3 états q_0, q_1, q_2 et 4 instructions. C'est la solution la plus optimale en termes du temps d'exécution et d'espace mémoire.

Exercice 5 Solution :
$q_0 0 D q_1$ Il faut toujours commencer par lire le mot le plus simple, dans notre cas le mot 0001#, donc on commence par lire le premier "0".
$q_1 0 D q_2$ lire le 2 eme "0".
$q_2 0 D q_3$ lire le 3 eme "0".
$q_3 1 D q_4$ lire le "1".
$q_4 \# T q_5$ Si après la lecture du "1", on trouve un #, ça veut dire qu'on vient de lire la séquence "0001" qui représente le premier mot dans le ruban. Donc, on remplace le # par le "T" pour dire que le mot contient la séquence "0001".
$q_4 (0/1) D q_4$ Si après la lecture de la séquence "0001" on ne trouve pas le # directement, ça veut dire que le mot n'est pas terminé mais il contient la séquence. Dans ce cas, on boucle avec l'état q_4 jusqu'à la fin du mot.
$q_5 T D q_5$ Après l'écriture du "T" on passe à droite avec le même état q_5.
q_5#Arrêt. Si on trouve un 2 eme #, ça veut dire que c'est la fin du ruban qui contient un seul mot. Donc jusqu'à présent on a traité les cas #0001#, #00010....#, #00011.......#, #0001##, #00010.....## et #00011........##.
$q_5 0 D q_1$ Le mot suivant commence par un "0", on retourne à q_1 car on vient de lire un "0".
$q_5 1 D q_6$ Si le mot suivant commence par un "1", il faut changer l'état.
$q_6 1 D q_6$ Boucler sur les "1".
$q_6 \# F q_5$ On trouve un # avec l'état q_6 ça veut dire q'on n'a pas trouvé la séquence, il faut écrire un "F".
$q_5 F D q_5$ Tester sur l'autre mot.
$q_6 0 D q_1$ On a trouvé le premier "0" après une séquence de "1".
$q_1 \# F q_5$ On n'a pas trouvé la séquence et on a trouvé un #.
$q_2 \# F q_5$ On n'a pas trouvé la séquence et on a trouvé un #.
$q_3 \# F q_5$ On n'a pas trouvé la séquence et on a trouvé un #.

$q_1 1 D q_6$ Après un "0", on a trouvé un "1", on passe à droite et à l'état q_6 pour tester sur les "1". Si on trouve , on boucle avec l'instruction $q_6 1 D q_6$, sinon si on trouve un "0", on le considère comme étant le premier "0" et on exécute $q_6 0 D q_1$ sinon si on tombe sur le #, ça veut dire que c'est la fin du mot et qu'on ne trouve pas la séquence, on exécute $q_6 \# F q_5$.
$q_2 1 D q_6$ La même chose avec q_2.
$q_0 1 D q_6$ i le mot commence par un "1", on teste sur les "1" en allant vers l'état q_6.
$q_3 0 D q_3$ Si on trouve plus de "000" c'est à dire "0000" ou "00000" ou plus, on continu à chercher le "1" car on considère par exemple dans le mot "1000000010" il y a la séquence 0001 qui est en rouge "1000000010".

Maintenant, si dans le mot "1000000010" on considère que la séquence "0001" n'existe pas c'est à dire on prend en considération uniquement 0001, alors on supprime l'instruction $q_3 0 D q_3$ et on la remplace par ces 4 instructions.
$q_3 0 D q_7$ On passe à droite et on change complètement l'état.
$q_7 0 D q_7$ On boucle sur les "0" avec le nouvel état.
$q_7 1 D q_6$ Si on trouve un "1" après les "0", on passe à l'état q_6.
$q_7 \# F q_5$ Si on trouve un #, on écrit un "F" et on passe à q_5.

Exercice 6 Solution :
$q_0 a D q_1$
$q_1 a D q_2$
$q_2 b D q_3$
$q_3 (a/b) D q_3$
$q_3 \# T q_4$
$q_4 T D q_4$
q_4#Arrêt.
$q_4 a D q_1$
$q_4 b D q_5$
$q_5 b D q_5$
$q_5 a D q_1$
$q_5 \# F q_5$
$q_5 F D q_4$
$q_0 b D q_5$
$q_1 b D q_5$
$q_1 \# F q_5$
$q_2 \# F q_5$
$q_2 a D q_6$
$q_6 a D q_6$
$q_6 b D q_5$
$q_6 \# F q_5$

Exercice 7 Solution :
$q_0 1 D q_1$ Lire le premier "1".
$q_1 1 D q_2$ Lire le 2eme "1".
$q_2 0 1 q_3$ Remplacer le "0" qui vient après deux "1" uniquement par un "1".

$q_3 1 D q_4$
q_4#Arrêt.
$q_4 0 D q_4$ Boucler sur les "0" avec l'état q_4.
$q_4 1 D q_1$ On vient de lire un premier "1", donc il faut aller à droite et passer à l'état q_1 .
$q_0 0 D q_4$ Le mot commence par un "0" il faut parcourir tous les "0".
$q_1 0 D q_4$ Après un "1", on a un "0".
q_1#Arrêt.
$q_2 1 D q_5$ On vient de lire un 3eme "1", donc il faut boucler sur les "1".
$q_5 1 D q_5$ Boucle sur les "1".
$q_5 0 D q_4$ On trouve un "0" après trois "1" ou plus.
q_5#Arrêt.
q_2#Arrêt.

Exercice 8 Solution :
La procédure à appliquer : c'est qu'il faut écrire une machine qui parcourt les "a" puis les "b" et si elle trouve un "a" entre les "b", elle le transforme en "b" puis elle revient en arrière et elle transforme le premier "b" de la liste en "a".

$q_0 a D q_1$ On suppose que le mot commence par un "a".
$q_1 a D q_1$ Dans le cas où on a plusieurs "a" il faut boucler sur tous les "a".
$q_1 b D q_2$ Le premier "b" après le ou les "a".
$q_2 b D q_2$ Dans le cas où on a plusieurs "b" il faut boucler sur tous les "b", jusqu'à présent le mot est trié.
$q_2 a b q_3$ Si on trouve un "a" après les "b", on le remplace par un "b" et on change d'état.
$q_3 b G q_3$ On revient en arrière en parcourant tous les "b".
$q_3 a D q_4$ On trouve le premier "a" qui se trouve avant les "b".
$q_4 b a q_1$ On remplace le premier "b" par un "a".
$q_0 b D q_2$ Le mot commence par un "b".
q_1#Arrêt.
q_2#Arrêt.
$q_3 \# D q_4$ Le cas où le mot commence par des "b" après on trouve un "a".

Exercice 9 Solution :

q_0 0 # q_1	q_4 (0/1) D q_4
q_0 1 # q_2	q_4 # G q_9
q_1 # D q_3	q_9 1 # q_6
q_2 # D q_4	q_5 # T q_5
q_3 (0/1) D q_3	$q_5 T$ Arrêt
q_3 # G q_5	q_5 1 F q_5
q_5 0 # q_6	q_5 F Arrêt
q_6 # G q_7	q_9 0 F q_5
q_7 (0/1) G q_7	q_9 # T q_5
q_7 # D q_8	q_8 1 # q_2
q_8 0 # q_1	q_8 # T q_5

Cette solution utilise un nombre minimal d'état, mais elle n'est pas rapide car à chaque fois on revient à gauche jusqu'au début puis on refait le même traitement (pour éviter ça, on peut faire le traitement des deux cotés). Avec la deuxième solution, on utilisera un nombre d'états qui peut aller jusqu'à q_{17}.
Remarque : Vous pouvez donner cet exercice (pour la deuxième solution)comme devoir.

Exercice 10 Solution :
$q_0(0/1)Dq_1$
$q_1(0/1)Dq_1$
$q_1\#Gq_2$
$q_2 0 1 q_3$
$q_2 1 0 q_4$
$q_4 0 G q_2$
$q_2\#1q_3$
$q_3(0/1)Dq_3$
$q_3\#Dq_5$
$q_5\#Dq_6$
$q_6\#$Arrêt.
$q_6(0/1)Dq_1$

Exercice 11 Solution :
1- Pour montrer que la fonction plus = x + y est une fonction primitive récursive, on utilise la règle de récursion suivante :
- plus (x,0) = x + 0 = x = $\pi_1^1(x)$.
- plus (x,y+1) = plus(x,y)+1 = succ (plus(x,y))= succ($\pi_2^3(x, plus(x,y), y)$)= succ$\circ\pi_2^3$(x,plus(x,y),y).

Conclusion :
La fonction h = $\pi_1^1(x)$ est une fonction primitive récursive de base car c'est une projection, la fonction g = succ$\circ\pi_2^3$ est primitive récursive car elle est construite par composition de deux fonctions primitives récursives de base succ et π. Nous avons ainsi réussi à construire la fonction "plus" par la règle de récursion à partir des deux fonctions primitives récursives h et g. La fonction "plus" est par conséquent primitive récursive.

2- La fonction Sigma = $\Sigma_{i=0}^{x} i$
Nous allons procéder à une construction en utilisant la règle de récursion.

- Sigma(0) = 0 qui est une constante.
- Sigma(x+1) = (0+.....+x)+ (x+1) = Sigma(x)+ (x+1) = plus(Sigma(x),x+1)= plus(Sigma(x),x)+1 = succ(plus(Sigma(x),x)) = plus ($\pi_1^2(Sigma(x), x)$,succ$\circ\pi_2^2$(Sigma(x),x))

Conclusion :
La fonction constante h=0 est une fonction primitive récursive de base (la fonction zéro()), la fonction plus est primitive récursive, les fonctions π_1^2, succ et π_2^2 sont primitives récursives. Donc Sigma est primitive récursive.

3- La fonction prédécesseur (pred(x)).
pred(x)= $\{ {}^{x-1}_{0} \quad {}^{si \quad x \neq 0}_{si \quad x=0}$
En utilisant la règle de récursion on aura :

- pred(0) = 0, elle est primitive récursive.
- pred(x+1) = x = g(x,pred(x)) = $\pi_1^2(x, pred(x))$
- Ou bien pred(x+1) = x = g(x,pred(x)) = $\sigma(\pi_2^2(x, pred(x)))$, elle est primitive récursive.

Elle est primitive récursive car toutes les fonctions qui la composent sont primitives récursives. Dans cet exemple, j'ai utilisé σ à la place de succ (c'est la même chose).

4-La fonction différence tronquée (Diff(x,y) ou $\dot{-}(x, y)$ ou x$\dot{-}$y) sachant que :
x$\dot{-}$y $=\{ {}^{x-y}_{0} \quad {}^{si \quad x \geq}_{si \quad x<y}$.
Par la règle de récursion on aura :

- x$\dot{-}$0 = x = $\pi_1^1(x)$(fonction identité), elle est primitive récursive car c'est une fonction de base.
- x$\dot{-}$(y+1) = pred(x$\dot{-}$y) = pred($\pi_2^3(x, x \dot{-} y, y)$), elle est primitive récursive.

Elle est primitive récursive car elle est composée de plusieurs fonctions primitives récursives.
Exemple :
2$\dot{-}$1 = 2 $\dot{-}$ (0+1) = pred (2 $\dot{-}$ 0) = pred (2) = 1.
1$\dot{-}$2 = 1 $\dot{-}$ (1+1) = pred (1 $\dot{-}$ 1) = pred (1 $\dot{-}$(0+1))= pred(pred(1 $\dot{-}$ 0)) = pred(pred(1))= pred(0)= 0.

5- La fonction différence absolue
$|x-y| = \{ {}^{x-y}_{y-x} \quad {}^{si \quad x \geq y}_{si \quad six<y}$
On peut écrire $|x-y|$ = (x$\dot{-}$y)+(y$\dot{-}$x), donc elle est primitive récursive car elle est composée de la fonction "plus" qui est primitive récursive et la fonction $\dot{-}$ qui est primitive récursive aussi.

6- La fonction alpha tel que $\alpha(x)$= $\{ {}^{1}_{0} \quad {}^{si \quad x=0}_{si \quad x \neq 0}$.
Par la règle de récursion on a :

- $\alpha(0) = 1 = succ \circ zro()$, elle est primitive récursive.
- $\alpha(x+1) = 0$ = zéro(), elle est primitive récursive.

Donc la fonction alpha est une fonction primitive récursive car elle est composée de plusieurs fonctions primitives récursives.
On peut aussi écrire $\alpha(x)$ sous la forme suivante :
$\alpha(x)$ = 1 $\dot{-}$ x c'est à dire :

- 1 $\dot{-}$ 0 = 1
- $\alpha(x+1)$ = 1 $\dot{-}$ x $\dot{-}$ 1 = pred (1 $\dot{-}$ x)= pred(π_2^2(x,1 $\dot{-}$ x)).

Exemple :
$\alpha(3)$ = 1 $\dot{-}$ 3 = 1 $\dot{-}$ (2+1)= pred(1 $\dot{-}$ 2) = pred (1 $\dot{-}$ (1+1))= pred(pred(1 $\dot{-}$ 1))= pred(pred(1 $\dot{-}$(1 $\dot{-}$ 0)))= pred(pred(pred(1 $\dot{-}$ 0)))=pred(pred(1))=pred(0)=0.

7- La fonction multiplication mult = x * y.
Par la règle de récursion

- mult(x,0) = x * 0 = 0 = zéro().
- mult(x,y+1) = mult(x,y)+x

Si on suit la formalisation donnée en cours, on aura :

- mult(x,0) = x * 0 = 0 = zéro()=h(x), elle est primitive récursive.
- mult(x,y+1) = mult(x,y)+x = g(y,mult(x,y),x) = π_2^3(y,mult(x,y),x) + π_3^3(y,mult(x,y),x)

On peut aussi écrire g = succ$\circ\pi_3^3$(mult(x,y),x-1,plus(mult(x,y),x-1)).
La fonction mult est primitive récursive car elle est constituée de fonctions primitives récursives.

8- La fonction factorielle Fact(x) = $x!$
En utilisant la règle de récursion, on aura :

- 0! = 1 = succ$\circ$zéro() c'est une composition de deux fonctions primitives récursives.
- $(x+1)!$ = g(x,$x!$)= (x+1)* $x!$ = succ(x) * $x!$ = succ($\pi_1^2(x, x!)$) * $\pi_2^2(x, x!)$ = mult(succ($\pi_1^2(x, x!)$),$\pi_2^2(x, x!)$), elle est primitive récursive (composition de fonctions primitives récursives)

Donc la fonction Fact(x) est primitive récursive.
Exemple :
Fact(3) = fact(2+1) = succ(2)* Fact(2)= succ(2)* Fact(1+1) = succ(2)*succ(1)*Fact(1)= succ(2)*succ(1)*Fact(0+1) = succ(2)*succ(1)*succ(0)* Fact(0)= 3*2*1*1=6.