

IFT 2105 A2016 - Devoir 1

Philippe Caron

18 octobre 2016

1 Languages RÉPÉTER et TANTQUE

Le programme suivant retourne le premier 1 de la suite de syracuse (à partir de quel moment, le cycle devient trivial 1, 4, 2, 1, ...). Afin de faciliter la lecture des macros sont utilisées. Le développement se fait en remplaçant l'appel par le texte contenu dans la macro, et l'argument de l'appel par l'argument spécifié :

```
C(rx) =  
  r0 ← 0           // pour les appels subséquents de C(rx)  
  r1 ← 1           // r1 sera la constante 1  
  tant que rx ≠ r1 faire [  
    r2 ← rx  
    MOD2(r2)  
    tant que r2 ≠ r3 faire [      // r3 sera la constante 0  
      MUL3(rx)  
      inc(rx)  
      inc(r0)  
    ]  
    r2 ← rx  
    MOD2(r2)  
    tant que r2 ≠ r1 faire [  
      DIV2(rx)  
      inc(r0)  
    ]  
  ]  
  rx ← r0
```

Un tel programme en langage RÉPÉTER serait impossible à concevoir puisqu'il faudrait connaître une borne supérieure à C(x), or on n'a jamais pu réussir à prouver que la suite de la conjecture de syracuse se terminait à tout coup, il est donc impossible de savoir si le programme va s'arrêter, et bien entendu impossible de trouver une borne supérieure.

*Macros à la page suivante...

```

MOD2( $r_x$ ) =
   $r_4 \leftarrow 0$ 
   $r_5 \leftarrow 0$ 
   $r_6 \leftarrow 0$ 
  tant que  $r_4 \neq r_x$  faire [
    inc( $r_4$ )
    inc( $r_5$ )
     $r_6 \leftarrow 0$ 
     $r_7 \leftarrow r_4$ 
    tant que  $r_7 \neq r_x$  faire [
      inc( $r_6$ )
      inc( $r_7$ )
    ]
    tant que  $r_6 \neq r_3$  faire [ $//r_3 = 0$ 
      inc( $r_4$ )
       $r_6 \leftarrow 0$ 
       $r_5 \leftarrow 0$ 
    ]
  ]
   $r_x \leftarrow r_5$ 
DIV2( $r_x$ ) =
   $r_8 \leftarrow 0$ 
   $r_{10} \leftarrow 0$ 
   $// r_9 \leftarrow r_x //$  Pour division sécuritaire (pas nécessaire)
   $// MOD2(r_9)$ 
   $//$  tant que  $r_9 \neq r_3$  faire[
   $//$    inc( $r_8$ )
   $//$    inc( $r_9$ )
   $//$  ]
  tant que  $r_8 \neq r_9$  faire [
    inc( $r_8$ )
    inc( $r_8$ )
    inc( $r_{10}$ )
  ]
   $r_x \leftarrow r_{10}$ 

MUL3( $r_x$ ) =
   $r_{11} \leftarrow 0$ 
   $r_{12} \leftarrow 0$ 
  tant que  $r_{11} \neq r_x$  faire [
    inc( $r_{11}$ )
    inc( $r_{12}$ )
    inc( $r_{12}$ )
    inc( $r_{12}$ )
  ]
   $r_x \leftarrow r_{12}$ 

```

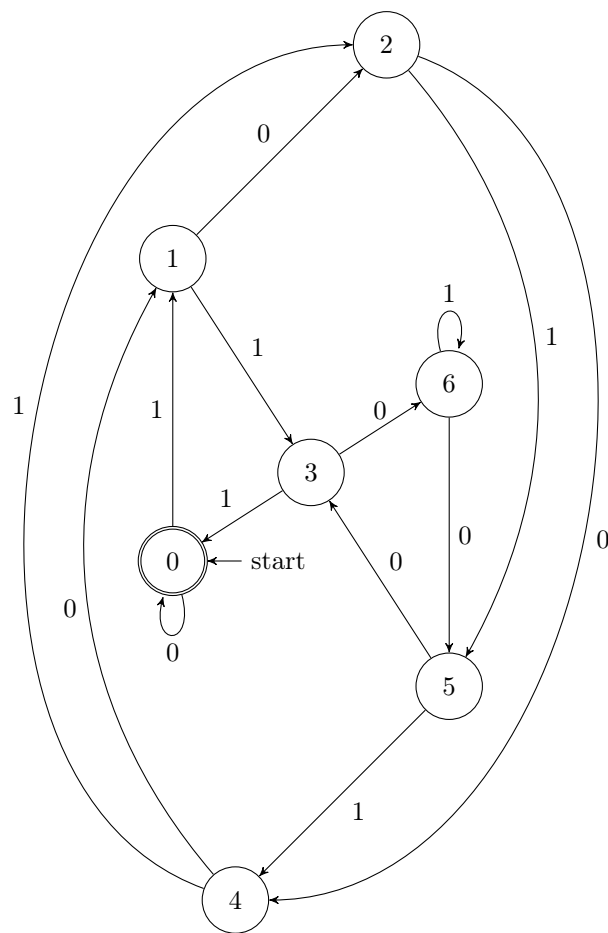
2 Langage régulier

a)

Le langage $L = \{x \mid \text{le nombre } x \text{ en binaire est divisible par } 7\}$ est régulier :

Preuve Pour représenter le langage, on imagine que chaque chiffre binaire rencontré est en fait une opération sur une valeur initiale de 0. Ainsi le chiffre 0 signifie $\times 2$ et le chiffre 1 signifie $\times 2 + 1$. Puis on évalue chacune de ces opérations avec un module de 7. Le résultat est toujours parmi les 7 états représenté ci-dessous, et l'état 0 est l'état acceptant.

Le langage est donc régulier puisqu'il peut-être représenté par un automate :



b)

Le langage $L = \{\omega \mid |\omega|_a = 2 \times |\omega|_b + 10\}$ n'est pas régulier.

Preuve Soit $R = \{\omega \in \{a, b\}^* \mid \omega = a^n \cdot b^{2n+10}\} \subset L$, si R n'est pas régulier et qu'il est compris dans L , alors L n'est pas régulier.

Supposons l'inverse, si R est régulier, alors il existe un $p > 0$ t.q. :

— $\forall i \geq 0, xy^iz \in R$, si $|\omega| \geq p$

— $|y| > 0$

— $|xy| \leq p$

Prenons le mot $\omega = a^{2p+10}b^p \in R$, avec $|\omega| = 3p + 10 > p$. Alors il s'en suit que y ne contient forcément que des a . Or cela signifie que $xy^2z \notin R$. Nous arrivons à une contradiction. R n'est pas régulier, ce qui implique de L n'est pas régulier.

3 Langage hors contexte

a)

Soit le langage suivant : $L = \{\omega \mid |\omega|_a = 3|\omega|_b\}$

$$G = (V, \Sigma, R, S)$$

où :

$$V = \{S\}$$

$$\Sigma = \{a, b\}$$

$$R = \{S \rightarrow SaSaSaSbS \mid SaSaSbSaS \mid SaSbSaSaS \mid SbSaSaSaS \mid \epsilon\}$$

b)

Le langage $L = \{\omega \mid \omega = x \cdot x \cdot x\}$ n'est pas hors contexte.

Preuve Supposons l'inverse, alors il existe un p t.q. :

— $\forall i \geq 0, uv^ixy^iz \in L$, si $|\omega| \geq p$

— $|vy| > 0$

— $|vxy| \leq p$

Choisissons le mot $\omega = x^p \cdot x^p \cdot x^p \in L$, avec $|\omega| = 3p > p$. Pour que mot après pompage soit aussi dans L , il y doit avoir un mot de la forme $(x' \cdot \delta) \cdot (x' \cdot \delta) \cdot (x' \cdot \delta)$ où $x' \cdot \delta = x$. Or il y a 3 δ , mais seulement deux fragments de pompage, il faut donc qu'un des fragment soit au moins de la taille de $|x| + |\delta|$ et qu'un autre fragment soit de la taille de δ pour que cela fonctionne. Mais c'est impossible puisque δ est non nul ($|y| > 0$), et $|x| + 2|\delta| > p \geq |vxy|$. On arrive donc à une contradiction et on prouve que L n'est pas hors contexte.

4 Marchine de Turing

Pour décider si un nombre est dans la suite de Fibonacci, on utilise un algorithme permettant de calculer les termes de la suite sous la forme suivante :

```
abab
aabab
aaabaab
aaaaabaaab
aaaaaaaaabaaab
...
```

Les "b" ajoutent cependant 2 à la longueur de la chaîne c'est pourquoi on ajoute 2 au nombre de départ. La machine effectue le calcul de la suite de Fibonacci sur la suite préexistante de 1 et regarde si celle-ci la couvre complètement et exactement, si c'est le cas, la machine accepte (A) sinon elle refuse (X).

Cette machine de Turing utilise l'alphabet de ruban $\{0, 1, a, b\}$

Le tableau des état de la machine (21) est à la page suivante :

Tableau des états

State	Read	Write	Next	Direction	State	Read	Write	Next	Direction
I_1	1	a	I_2	R	W_1	1	a	D_3	L
	0	-	X	-		0	-	X	-
	a	-	X	-		a	a	W_1	R
	b	-	X	-		b	-	X	-
I_2	1	b	I_3	R	D_3	1	1	M_1	R
	0	-	A	-		0	-	X	-
	a	-	X	-		a	a	D_3	L
	b	-	X	-		b	b	D_3	L
I_3	1	a	I_4	R	F_1	1	a	F_1	L
	0	-	A	-		0	0	M_4	R
	a	-	X	-		a	-	X	-
	b	-	X	-		b	-	X	-
I_4	1	a	P_1	R	M_4	1	1	K_1	L
	0	-	A	-		0	-	A	-
	a	-	X	-		a	a	M_4	R
	b	-	X	-		b	b	M_4	R
P_1	1	1	P_1	R	K_1	1	-	X	-
	0	1	P_2	R		0	-	X	-
	a	-	X	-		a	b	K_2	L
	b	-	X	-		b	-	X	-
P_2	1	1	P_2	L	K_2	1	-	X	-
	0	1	P_2	L		0	-	X	-
	a	-	X	-		a	a	K_2	L
	b	b	D_1	L		b	a	K_2	L
D_1	1	-	X	-	K_3	1	-	X	-
	0	-	X	-		0	-	X	-
	a	a	D_1	L		a	b	K_4	L
	b	b	D_2	L		b	-	X	-
D_2	1	-	X	-	K_4	1	-	X	-
	0	0	M_1	R		0	-	X	-
	a	a	D_2	L		a	a	K_4	L
	b	-	X	-		b	a	K_5	R
M_1	1	-	X	-	K_5	1	1	I_5	L
	0	-	X	-		0	-	A	-
	a	1	M_2	R		a	a	K_5	R
	b	b	F_1	L		b	b	K_5	R
M_2	1	-	X	-	I_5	1	-	X	-
	0	-	X	-		0	-	X	-
	a	a	M_2	R		a	-	X	-
	b	b	M_3	R		b	b	D_1	L
M_3	1	-	X	-					
	0	-	X	-					
	a	a	M_3	R					
	b	b	W_1	R					