

INF431 : *Devoir Maison 1*

Wilson JALLET

13 février 2018

Question 1

L'algorithme brutal parallèle consisterait en :

- lancer un *thread* pour chaque position de début i envisagée, soit $|Y| - |X| + 1$ *threads*
- pour chaque thread, comparer $Y_{i:i+|X|-1}$ à X , ce qui se fait en temps $O(|X|)$

Le pseudo-code est comme suit :

```
boolean[] result;  
  
for i=0 to len(Y)-len(X)+1:  
    Processor i does {  
        result[i] = X.isSubstringAt(Y,i)  
    }  
  
return result;
```

Le temps d'exécution est $O(|X|)$, le nombre d'opérations est $O(|X|(|Y| - |X|))$ et l'efficacité sur p processeurs est

$$e_p = O\left(\frac{|X| + |Y|}{p|X|}\right).$$

Question 2

CREW : Comme on écrit dans un tableau dans l'algorithme précédent, il n'y a jamais d'écritures concurrentes dans la même case mémoire. L'algorithme reste identique, avec un temps d'exécution $O(|X|)$.

EREW : Il y a plusieurs lectures concurrentes des entrées de la chaîne X . On peut paralléliser en faisant en sorte que chaque *thread* effectue sa propre copie de X , ce qui se fait en temps total $O(\log |X|(|Y| - |X|))$. Ensuite, chaque *thread* i fait les $O(|X|)$ comparaisons nécessaires pour savoir si i est une occurrence de X dans Y . Le temps d'exécution est $O(|X|(|Y| - |X|))$.

Question 3

La JVM est proche d'une PRAM CREW, donc on peut programmer l'algorithme CRCW/CREW précédent.

On peut créer une classe `Task` étendant l'interface `Runnable` correspondant à la comparaison à partir de la position i sur la chaîne Y , contenant un champ `int i`. Ensuite, on crée les *threads* nécessaires via `new Thread(new Task())` et on les démarre via la méthode `start`, suivi des opérations de synchronisation avec `join`. Chaque *thread* écrit sur sa propre case du tableau des résultats, il n'y a pas de mécanisme de verrou supplémentaire à considérer.

Question 4

On crée une variable `res` qui sera le résultat, et on lance un *thread* par entrée du tableau A : le *thread* i calculera `res = min(res, i)`. Cela garantit un temps d'exécution $O(1)$.

Question 5

On lance un *thread* pour chaque $i \in \llbracket 2, \pi(X) \rrbracket$, et on lance sur chacun d'eux l'algorithme précédent en temps constant sur le tableau $A = (X_k \neq X_{i+k-1})_k$.

Question 6