

Rapport TP1

Thomas Bille - 1754864
Jean-Marie Robin - 1753255

Pour la compétition nous choisissons le code de recherche locale : code2.

Pour tester l'exemple donner lancer le script parametre.py .

Question 1 :

Expliquez ce que fait le code suivant :

```
def fct(f, g, p, l):  
    return map(lambda x: f(g(x)),  
               [x for x in l if p(x)])
```

Fonction qui prend en parametre :

f : fonction qui retourne un objet 'a qui prend un en paramètre un objet de type 'c
g : fonction qui retourne un objet 'b qui prend un en paramètre un objet de type 'a
p : fonction qui retourne un boolean
l : liste d'objet 'c

$fct : (fun 'b \rightarrow 'a) \rightarrow (fun 'c \rightarrow 'b) \rightarrow (fun 'a \rightarrow boolean) \rightarrow 'c\ list \rightarrow 'a\ list$

Fonction qui retourne :

Une liste des éléments de la liste l qui vérifie la propriété p. Si l'élément vérifie la fonction p on applique f(g(element)) .

Dans ce cas :

$fct : (fun\ string \rightarrow int) \rightarrow (fun\ int \rightarrow string) \rightarrow (fun\ int \rightarrow boolean) \rightarrow int\ list \rightarrow int\ list$

```
def fct(f, g, p, l):  
    return map(lambda x: f(g(x)),  
               [x for x in l if p(x)])
```

```
def f(x) :  
    return int(x)+1
```

```
def g(x) :  
    return str(x)
```

```
def p(x) :  
    if x>2 :  
        return True  
    else :  
        return False
```

```
l = [1,2,4,5]  
solution = fct(f,g,p,l)  
print solution  
#  
# solution = [5,6]  
#
```

Question 2 :

Quelles sont les points fort et les faiblesses de vos implémentations ?

Recherche arborescente :

- calcul du coût et de l'heuristique à revoir
- ceci implique un programme plus long à trouver la solution

Recherche locale :

- problème pour un type de pièce le programme plante. Pas de réponses au problème... (tester fichier parametre2.py)