

Parcours des arbres en profondeur - Autour du compte est bon

Suite à l'exécution du script « compteEstBonEleve.py », la variable expressions contient la liste des 2 764 800 combinaisons d'opérations utilisant les 6 nombres de la liste cards.

Certaines opérations ne sont pas valides. Nous verrons plus tard comment améliorer ce code mais dans un premier temps, nous allons l'utiliser en l'état comme outil de travail.

Ces expressions sont données sous forme d'arbre, comme par exemple :

```
      +
     / \
    +  9
   / \
  *  5
 / \
+  -
/\  /\
10 3 25 8
```

La valeur de cette expression est $((10 + 3) \times (25 - 8) + 5) + 9 = 235$.

Les nœuds de ces arbres sont des instances de la classe Number ou de la classe Operator.

Nœud du type Number

Les nœuds de la classe Number sont les feuilles de l'arbre. On utilise uniquement la méthode `get_value()` qui renvoie la valeur du nœud (un des 6 nombres de la liste cards) et la méthode `__repr__` qui renvoie cette même valeur sous forme de chaîne de caractères.

Nœud du type Operator

Tous les autres nœuds sont du type Operator. Ils disposent des méthodes :

- `left()` renvoie le fils gauche
- `right()` renvoie le fils droit
- `get_op()` renvoie l'opérateur : $\lambda x, y: x + y$ (ou une autre des quatre opérations)
Dans le cas de l'addition, `expression.get_op()(4, 5)` renverra 9.
- `get_symbol` renvoie le symbole correspondant à l'opérateur : +, -, * ou /

`isinstance(expression, Operator)` renvoie un booléen indiquant si le nœud expression est du type Operator (ou Number).

1. Travail sur une des 2 764 800 expressions, par exemple expressions[0]

- Exécuter le script « compteEstBonEleve.py ».
- En utilisant les méthodes `get_symbol`, `get_value`, `left` et `right` des classes Number et Operator (ou de leur classe mère Node), reconstituer sur papier l'arbre stocké dans `expressions[0]`.
- Évaluer la valeur de cet arbre (toujours sur papier).

2. Représentation de l'arbre

L'application graphviz permet de représenter graphiquement des arbres : <http://www.graphviz.org/>

On y accède en Python avec la bibliothèque graphviz : <https://pypi.org/project/graphviz/>

a. Tester la fonction `showtree` sur `expressions[0]`.

b. De quel type de parcours s'agit-il ?

3. Écriture de fonctions

a. Écrire une fonction `eval_expression(expression)` qui renvoie la valeur d'une expression. De quel type de parcours s'agit-il ?

b. Écrire une fonction `retourne_expression` qui renvoie l'expression sous forme de chaîne de caractères :

$$((10 + 3) \times (25 - 8) + 5)$$

De quel type de parcours s'agit-il ?

c. Écrire une fonction `affiche_calculs(expression)` qui détaille les calculs d'une expression, comme le font les participants lors du jeu :

$$10 + 3 = 13$$

$$25 - 8 = 17$$

$$13 * 17 = 221$$

$$221 + 5 = 226$$

d. Écrire une fonction `resoudre(expressions, goal)` qui va afficher sous forme de chaîne toutes les expressions égales à la valeur `goal`.

e. Tester la fonction `resoudre`. Est-ce que toutes les solutions affichées conviennent ? Proposer, sans les programmer, des pistes pour améliorer le code.

4. Travail sur les classes

a. Remplacer la fonction `eval_expression` par une méthode `eval(self)` de la classe `Operator`. On effectuera tous les ajouts nécessaires (méthodes, propriétés) à la classe `Operator`.

Les erreurs de divisions par 0 vont être réglées dans la question suivante.

b. Modifier la fonction `me_rec` afin de :

- permuter les termes d'une soustraction ou d'une division si nécessaire pour que le nombre le plus grand soit toujours en premier.
- éviter les divisions incorrectes, c'est-à-dire les divisions par 0 et celles à reste non nul.

c. Définir une méthode `__repr__(self)` de la classe `Operator` permettant un affichage sous forme de chaîne de caractères. La méthode `__repr__` est celle appelée lorsqu'on affiche l'objet (par appel dans la console ou fonction `print` en l'absence de méthode `__str__`).

5. Travail sur le solveur du Compte est bon

a. Modifier la fonction `me_rec` pour enregistrer uniquement dans la liste `expressions` les expressions donnant le résultat attendu.

b. La fonction `me_rec` teste-t-elle toutes les expressions possibles ou seulement celles utilisant toutes les cartes ? Modifier cette fonction si besoin pour tester toutes les expressions possibles.