

2023-RO-02 Notation postfixe

Body

Une expression mathématique est constituée :

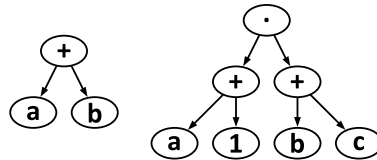
- d'un *opérateur* : $+$, $-$, \cdot ou $:$
- et d'*opérandes* : des chiffres comme 1, 2, ..., des lettres comme a, b, ..., ou d'autres expressions comme $(1 + 2)$.

La structure d'une expression mathématique peut être représentée par une *arborescence*. Ce diagramme composé d'opérateurs et d'opérandes est dessiné comme cela : un cercle contenant un opérateur est relié à l'arborescence de ses opérandes. Dans le cas le plus simple, il s'agit de cercles contenant des chiffres ou des lettres.

On peut dériver la *notation postfixe* d'une expression mathématique d'une telle arborescence. Dans cette notation, on écrit chaque expression en commençant par les opérandes suivis des opérateurs.

Expression mathématique : $a + b$ $(a + 1) \cdot (b + c)$

Arborescence :



Notation postfixe : $a b +$ $a 1 + b c + \cdot$

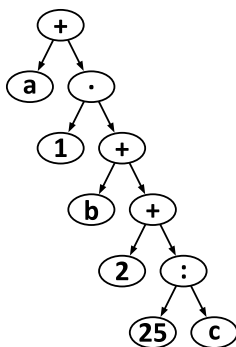
Voici la notation postfixe d'une autre expression :

$a 1 + b 2 + \cdot 25 c : +$

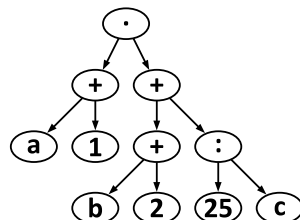
Question/Challenge - for the brochures

Quelle arborescence correspond à cette expression ?

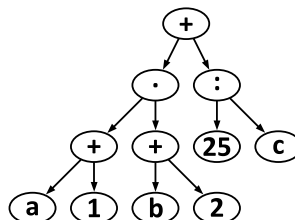
Answer Options/Interactivity Description



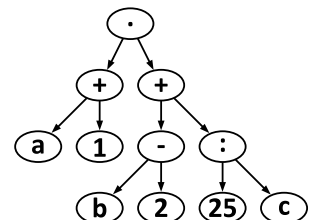
A)



B)

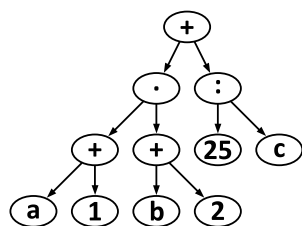


C)



D)

Answer Explanation



La bonne réponse est C :

Comme décrit dans la donnée de l'exercice, l'opérateur central d'une expression mathématique se trouve toujours tout en haut de l'arborescence (il forme sa *racine*) et tout à la fin de la notation postfixe. Si l'on veut trouver l'arborescence d'une expression en notation postfixe, il faut chercher le dernier signe de la notation postfixe tout en haut de l'arborescence, dans ce cas le $+$. Seules les arborescences des réponses A et C ont un $+$ à leur racine.

L'opérateur $+$ a deux opérandes, un à gauche et un à droite. En notation postfixe, on voit directement (à l'avant-dernier signe) que l'opérande à droite de l'expression est également une expression avec l'opérateur $:$. L'arborescence doit donc avoir le signe $:$ à droite sous la racine. Ce n'est le cas que pour l'arborescence de la réponse C, qui doit donc être la bonne réponse.

On peut le vérifier en transformant toute l'arborescence de la réponse C en notation postfixe :

- les trois plus « petits » arbres, constitués de 3 éléments chacun, deviennent $a\ 1\ +$, $b\ 2\ +$ et $25\ c\ :$.
- Les deux petits arbres de gauche deviennent les opérandes du $+$ d'en haut ; l'arbre partiel de gauche devient donc $a\ 1\ +\ b\ 2\ +\ \cdot$. Le troisième petit arbre est l'opérande de droite.
- L'arborescence de la réponse C s'écrit donc $a\ 1\ +\ b\ 2\ +\ \cdot\ 25\ c\ :$ en notation postfixe, ce qui est exactement l'expression de la donnée de l'exercice.

This is Informatics

La *notation postfixe*, aussi appelée *notation polonaise inverse*, est souvent utilisée pour formuler des expressions, mathématiques ou autres (par exemple en programmation) de manière compacte et univoque. Si l'on écrivait l'expression de l'arborescence de la réponse C en notation normale (c'est à dire avec les opérateurs entre les opérandes, appelée aussi *notation infixe*), il faudrait utiliser des parenthèses : $(a + 1) \cdot (b + 2) + 25 : c$, dont on n'a pas besoin avec la notation postfixe. La notation postfixe a été introduite sous la forme de notation préfixe, avec les opérateurs devant les opérandes, par Jan Łukasiewicz (1878–1956). On utilise cette notation entre autres pour les fonctions mathématiques $f(x, y)$ et en programmation `fonctionname(argument1, argument2, argument3)`. En informatique, elle est utilisée entre autres pour l'*analyse syntaxique* (*parsing* en anglais) des expressions d'un langage de programmation.

Dans la passé récent, beaucoup de gens ont découvert la notation postfixe en utilisant les premières calculatrices scientifiques : elle permettait d'entrer et de calculer des expressions mathématiques rapidement, de manière fiable et sans parenthèses. Il existe encore aujourd'hui une communauté de gens qui utilisent les calculatrices programmables (comme l'HP-35s) avec la notation postfixe.

This is Computational Thinking

Um die Aufgabe zu lösen, muss ein komplexer mathematischer Ausdruck durch *Zerlegen* oder *Dekomposition* in kleinere mathematische Ausdrücke verwandelt werden, die für sich im Kontext verständlich sind. Zudem muss ein Vorgang dahingehend *analysiert* werden, dass seine Funktionsweise auch in rückwärtiger Reihenfolge verständlich wird. Im Grunde wird dabei dieselbe Information zwischen unterschiedlichen Darstellungsweisen konvertiert.

Informatics Keywords and Websites

- Notation polonaise inverse: https://fr.wikipedia.org/wiki/Notation_polonaise_inverse
- Arbre syntaxique: https://fr.wikipedia.org/wiki/Arbre_syntaxique
- Jan Łukasiewicz: https://fr.wikipedia.org/wiki/Jan_Łukasiewicz
- HP 35s: <https://fr.wikipedia.org/wiki/HP-35s>

Computational Thinking Keywords and Websites

- Zerlegen, Dekomposition: <https://de.wikipedia.org/wiki/Modell#Modellbildung>