

Université Cadi Ayyad
École Supérieure De Technologie-Safi
Département : Informatique
Filière : DUT ou GI

Rapport De projet De Fin D'étude
DUT ou GI

Application graphique de traduction de formules infixes en code
Postscript

Réalisé par :

JADIR Mohammed
ROUDANI Zainab

Encadré par :

BAYAR Abdorahman

ANNÉE UNIVERSITAIRE : 2022/2023

Dédicaces

Je dédie ce projet :

A nos chers parents pour leurs sacrifices
pour leur patience, leur amour et leur confiance en nous.

Ils ont tout fait pour notre bonheur et notre réussite.

nous vous rends hommage par ce modeste ouvrage en
témoignage de notre gratitude éternelle et de notre infinie
amour envers eux, témoignage de notre profonde affection et
de notre attachement indéfectible,

A nos chers amis en gage de notre sincère appréciation Et
gratitude pour leurs efforts, leur soutien continu leur aide
précieuse et leur amour durant nos études.

Que Dieu nous gardons toujours unis,

A notre cher superviseur BAYAR Abdelouahad qui nous lui
accompagnés durant ce projet. Qu'il reçoive nos sincères
nous lui nous sincèrement reconnaissant pour ses efforts, son
soutien et son aide dans la réalisation de ce projet.

Remerciements

Nous tenons à exprimer nos sincères remerciements et notre profonde gratitude envers M. BAYAR Abdorahman, de nous avoir donné l'opportunité de réaliser cette mission ce projet. Nous sommes reconnaissants envers lui pour sa disponibilité et ses conseils qui nous ont permis d'accomplir toutes nos tâches avec succès.

Enfin, nous voudrions exprimer nos remerciements pour cette opportunité de travailler ensemble sur ce projet passionnant.

Notre collaboration avec vous a été remarquable, et nous avons apprécié votre professionnalisme et votre enthousiasme tout au long du processus. Merci de nous avoir permis de contribuer à cette réussite..

Table des matières

Résumé	8
Chapitre I Introduction Générale	9
Chapitre II Présentation du projet	10
1 Introduction	10
2 Cahier de charge	10
Analyseur syntaxique pour la conversion d’infixe en postfixe :	11
Convertisseur postfixe en PostScript :	11
Ajouter une fonction d’infixe et sa valeur en PostScript dans un fichier : . . .	12
Modification de fichier contenant les fonctions d’infixe et leurs valeurs en Post-Script :	12
3 Réalisation d’application	12
Outils et environnements techniques :	14
.	18
.	18
Bibographie	19
Annexe 120	

Table des figures

II.1	Schéma explicatif	13
II.2	Schéma de conception1	13
II.3	Schéma de conception2	14
II.4	Logo HTML5	14
II.5	Logo css"	15
II.6	Logo JS	15
II.7	Logo BOOTSTRAP	15
II.8	Logo BOOTSTRAP	16
II.9	Logo vs-code	16
II.10	Logo git	16
II.11	Logo github	17

Résumé

Le présent rapport détaille la réalisation d'un projet de Développement d'une application graphique de traduction de formules infixes en postfixe et en code Postscript.

Le but principal du projet est de créer un programme qui puisse convertir une expression arithmétique, écrite en notation infixée, en une notation postfixée et postscriptée. La conversion s'effectuera en utilisant l'algorithme de notation polonaise inversée. Cette conversion est importante pour faciliter les calculs mathématiques et les opérations sur les expressions arithmétiques.

Le programme devra permettre à l'utilisateur d'entrer l'expression arithmétique en notation infixée et d'obtenir en sortie la même expression en notation postfixée et postscriptée. Pour cela, le programme devra appliquer les différentes étapes de l'algorithme de notation polonaise inversée, notamment la gestion des opérateurs et des parenthèses.

Le projet nécessitera donc la conception d'un algorithme efficace pour la conversion et la mise en place d'une interface utilisateur conviviale. Le programme devra également être testé pour s'assurer de son bon fonctionnement dans différents scénarios et situations.

Chapitre I

Introduction Générale

L'écriture des formules mathématiques en notation infixe est une méthode courante et largement utilisée dans les manuels scolaires, les livres et les documents scientifiques. Cependant, cette notation peut être difficile à manipuler et à évaluer automatiquement pour les ordinateurs. Pour résoudre ce problème, les notations postfixe et postscript ont été développées.

Le but de ce projet de fin d'études est de créer une application desktop qui peut convertir des expressions mathématiques écrites en notation infixe en notation postfixe et postscript. Cette application permettra aux utilisateurs de saisir des expressions mathématiques complexes, qui seront ensuite converties en notation postfixe ou postscript. Les avantages de cette conversion incluent une évaluation plus rapide et une réduction de la complexité du code.

Ce projet couvrira plusieurs aspects, notamment l'analyse syntaxique et sémantique des expressions mathématiques, la conception d'algorithmes de conversion efficaces et l'implémentation de l'application en utilisant python en générale.

L'application résultante sera utile pour les enseignants de mathématiques, les étudiants, les ingénieurs et les scientifiques qui travaillent sur des projets impliquant des calculs mathématiques complexes. Elle permettra de faciliter les calculs, d'augmenter la productivité et de réduire les erreurs de saisie.

Chapitre II

Présentation du projet

1 Introduction

Nous exposons, dans ce chapitre, la solution que nous avons proposée et cette conception du système à réaliser qui a pour but de rendre flexible la tâche de la gestion. En d'autres termes, ce chapitre devrait répondre à la question : comment faire ?

2 Cahier de charge

Le problématique

La conversion d'expressions mathématiques de la notation infixée à la notation postfixée ou postscript est courante en programmation. Cela permet une manipulation plus aisée de ces expressions dans les programmes informatiques.

Cependant, le temps d'analyse de l'expression mathématique nécessaire pour effectuer cette conversion peut varier en fonction de la complexité de l'expression, et il est important de prendre en compte ce temps d'analyse pour garantir l'efficacité et la performance de l'application de conversion.

Dans ce projet de fin d'études, l'objectif est d'analyser les différentes méthodes de conversion existantes et de proposer des améliorations pour réduire le temps d'analyse des expressions mathématiques. De plus, il sera étudié comment implémenter efficacement un algorithme de conversion d'expression arithmétique en notation infixée vers la notation postfixée et postScript, en prenant en compte les différentes règles de priorité et d'associativité des opérateurs.

Notations mathématiques

Dans la plupart des langages de programmation, les formules mathématiques sont écrites en notation infixée

- **Notation infixée** C'est la notation habituelle utilisée en mathématiques, où l'opérateur est placé entre les deux opérandes. Par exemple :

$(5 + 3) * 2$

- **Notation postfixée** : aussi appelée notation polonaise inverse, l'opérateur est placé après les deux opérandes. Par exemple :

'5 3 + 2 *'

- **Notation postscript** : c'est une variante de la notation postfixée utilisée dans le langage de programmation PostScript. Les opérandes sont empilés sur une pile, et les opérateurs sont appliqués aux éléments du sommet de la pile. Par exemple :

'5 3 add 2 mul'

Les fonctionnalités d'application

Analyseur syntaxique pour la conversion d'infixe en postfixe :

Cette fonctionnalité devrait prendre en entrée une expression mathématique en notation infixée et la convertir en notation postfixée en utilisant un algorithme de pile (Stack).

L'analyseur syntaxique doit être en mesure de gérer les opérateurs mathématiques de base tels que +, -, *, /, les parenthèses et les fonctions avec ses paramètres comme cos sin et d'autres fonctions avec personnalisation de chaque fonction.

Convertisseur postfixe en PostScript :

Cette fonctionnalité devrait prendre en entrée une expression mathématique en notation postfixée et la convertir en notation PostScript en utilisant une syntaxe appropriée.

Il doit également gérer les opérateurs mathématiques de base et les fonctions mathématiques telles que sin, cos et avec des fonctions personnalisées comme 'somme' en infixe peut être 'sum' en postscript.

Ajouter une fonction d'infixe et sa valeur en PostScript dans un fichier :

Cette fonctionnalité doit permettre à l'utilisateur d'ajouter une nouvelle fonction d'infixe dans l'application avec sa valeur correspondante en PostScript.

Vous pouvez stocker ces fonctions et leurs valeurs dans un fichier texte ou une base de données.

Modification de fichier contenant les fonctions d'infixe et leurs valeurs en PostScript :

Cette fonctionnalité doit permettre à l'utilisateur de modifier le fichier contenant les fonctions d'infixe et leurs valeurs en PostScript.

Vous pouvez également ajouter une fonctionnalité de recherche pour faciliter la modification de fichiers volumineux.

Conclusion générale

En utilisant cette structure, vous pouvez créer une application qui offre une fonctionnalité complète pour la conversion d'expression mathématique en notation postfixée et en notation PostScript, ainsi que la gestion de fonctions d'infixe et leurs valeurs en PostScript.

3 Réalisation d'application

Introduction :

Dans ce chapitre, on va s'intéresser à la réalisation du projet de fin d'étude. On évoquera ainsi ma conception avant d'entamer la réalisation et puis la mise en place soutenue de figure commentées.

Analyse et conception :

Après avoir détaillé le cahier des charges et bien assimilé les objectifs de notre projet, une conception avant l'initialisation de la réalisation de notre projet était illustrée comme suit :

Notre Input, il s'agit d'une formule mathématique. A l'aide d'un code python on va traiter la formule pour l'adaptation de besoin de client tels que :



FIGURE II.1 – Schéma explicatif

- **Input :**
Un utilisateur crie une formule soit infixe, postfixe ...
- **Output :**
La forme du résultat dépend des besoins de l'utilisateur (infixe, postfixe, PostScript).
- **Fonction besoin a ajouté (infixe=>key) :**
La fonctionnalité à ajouter consiste à traduire des formules en notation infixe vers la notation PostScript. Cette fonctionnalité sera activée par une clé spécifique.
 - Fonction a saisi par l'utilisateur comme =>'somme()'
- **Fonction besoin a ajouté (postscript=>value) :**
La fonction de conversion doit également prendre en compte l'affichage de la formule si l'utilisateur souhaite spécifier une fonction, telle que "sum()". Ainsi, si l'utilisateur entre "Somme()", la fonction doit être convertie en "sum()". La valeur de clé en PostScript est également un aspect important à considérer lors de la conversion de la formule en PostScript.

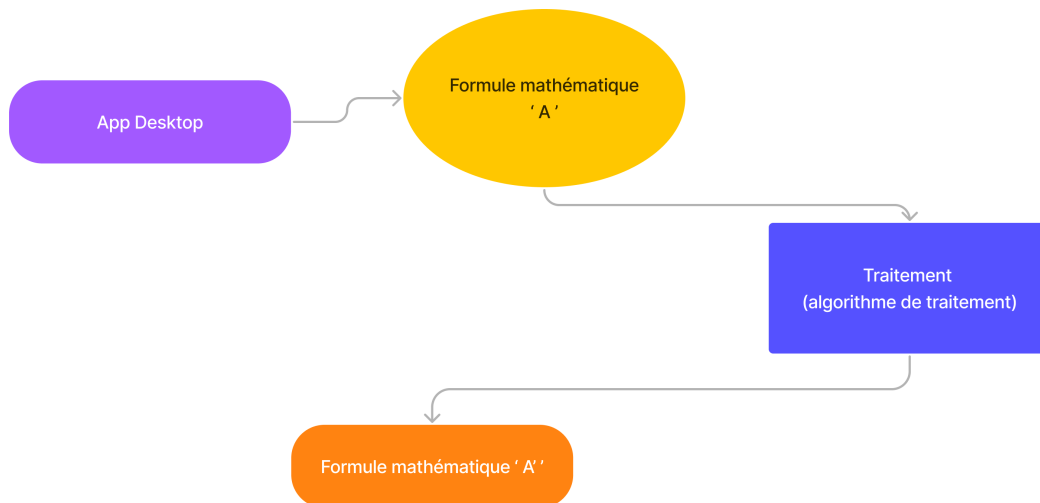


FIGURE II.2 – Schéma de conception1

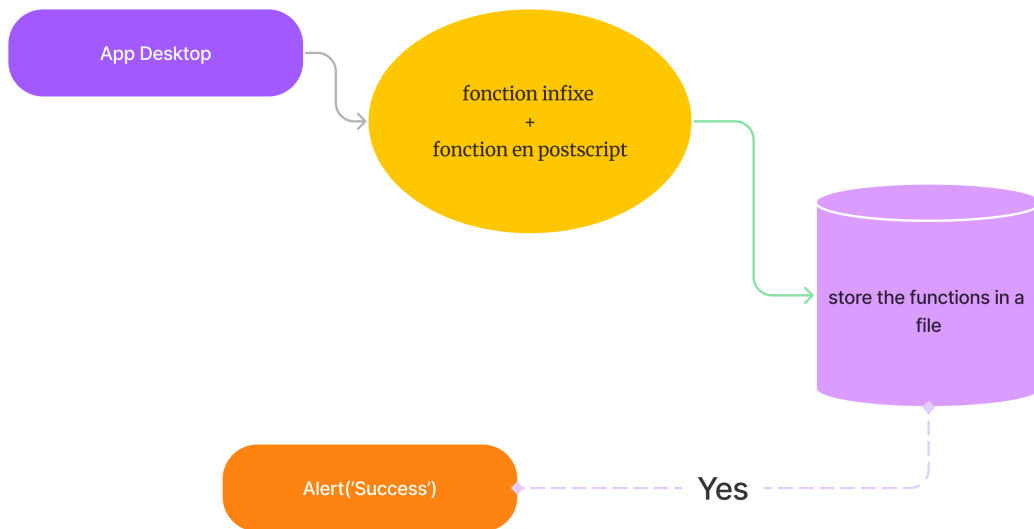


FIGURE II.3 – Schéma de conception2

Développement :

Outils et environnements techniques :

1. HTML5 :



FIGURE II.4 – Logo HTML5

Le HyperText Markup Language, généralement abrégé HTML ou, dans sa dernière version, HTML5, est le langage de balisage conçu pour représenter les pages web. tem **CSS3** :



FIGURE II.5 – Logo css"

Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web.

2. Java Script :



FIGURE II.6 – Logo JS

Il s'agit d'un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d'animer des images, et tout ce à quoi on peut penser. Bon, peut-être pas tout, mais vous pouvez faire bien des choses avec quelques lignes de JavaScript.

3. BOOTSTRAP :



FIGURE II.7 – Logo BOOTSTRAP

Python est un langage de programmation interprété, orienté objet et de haut niveau. Il est conçu pour être facile à lire, à écrire et à maintenir, et est souvent utilisé pour des applications telles que le développement web, la science des données, l'analyse de données, l'automatisation de tâches et la création de scripts. Il est également populaire pour l'enseignement de la programmation en raison de sa syntaxe claire et de sa grande communauté de développeurs et de contributeurs. Python est un langage multi-plateforme, ce qui signifie qu'il peut être exécuté sur différents systèmes d'exploitation tels que Windows, Mac OS et Linux. Il dispose également d'une vaste bibliothèque standard, qui contient de nombreux modules et packages pour faciliter le développement de logiciels.



FIGURE II.8 – Logo BOOTSTRAP

4. **BOOTSTRAP :**

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

5. **Visual-studio-code :**



FIGURE II.9 – Logo vs-code

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique, le débogage, et les commandes git.

6. **GIT :**



FIGURE II.10 – Logo git

Git est un système de contrôle de version distribué gratuit et open source conçu pour tout gérer, des petits aux très grands projets, avec rapidité et efficacité.

7. GITHUB :



FIGURE II.11 – Logo github

GitHub est une plateforme de développement qui fournit un service cloud aux développeurs pour stocker et gérer leur code. De l'open source à l'entreprise, vous pouvez héberger et revoir du code, apporter des modifications, gérer des projets et construire des logiciels aux côtés de millions d'autres développeurs .

Conclusion :

kihvugjyhfgvujyhg

Bibographie

- [https ://www.marsamaroc.co.ma/](https://www.marsamaroc.co.ma/) (consulté le ...)
- [https ://www.openclassrooms.com](https://www.openclassrooms.com)

Annexe 1