

---

# Projet de Compilation

*Math2Tex*

---

AZOU Raphaël

Master **ASSEL**

`raphael.azou@ensta-bretagne.org`

ENSTA BRETAGNE

May 6, 2024

## Description du projet

L'écriture d'un document scientifique Latex implique souvent d'écrire et de mettre en forme formules, démonstrations, équations, et d'autres formes d'expressions mathématiques plus diverses les unes que les autres. Ce processus de rédaction peut alors être complexe si vous n'êtes pas familier avec la syntaxe mathématique Latex. Ce projet à donc pour objectif de résoudre ce problème en proposant une grammaire simplifiée, rapide à saisir et facilement compréhensible et mémorisable pour un rédacteur occasionnel. Regardons un exemple concret pour comprendre le besoin

```
\begin{equation}
  P^h = \sum_{i=0}^{n-1} (a^i + \alpha)
\end{equation}
```

Ce code demande de bien se rappeler de la syntaxe, notamment pour la somme, et est un peu complexe à saisir avec beaucoup de caractères différents. Il faudrait ainsi normaliser ces caractères en introduisant la notion de fonctions au nom évocateur. Par exemple, pour réaliser un exposant on pourrait introduire `pow(x, 3y+1)` au lieu de `x^{3y+1}`. Certes, plus de caractères sont requis, mais une approche par fonctions dispose de plusieurs avantages comme par exemple de l'auto-complétion, ce qui n'existe pas pour la rédaction d'expressions mathématiques avec des éditeurs Latex. Voici donc le code précédent écrit avec la future grammaire du compilateur

```
math{
  equation{
    pow(P, h) = sum(i:0, n-1, pow(a,i) + ALPHA)
  }
}
```

Dans un premier temps, la grammaire se voudra simpliste est non exhaustive, mais des éléments de grammaire seront rajoutés une fois les premier tests du compilateur concluants.

# 1 Grammaire

math	<b>math</b> { <i>equation</i> *   <i>expression</i> * }
equation	<b>equation</b> { <i>expression compop expression</i> }
expression	<i>function</i>   <i>statement</i> ( <i>op statement</i> )*
function	<i>literal</i> <i>parenth</i>
statement	( <i>function</i> )? ( <i>parenth</i> )? <i>term</i>
parenth	( ( <i>expression</i> )* (,)* )
term	<b>LIT_INT</b>   <b>LIT_FLOAT</b>   <b>IDENTIFIER</b>   <b>LIT_CHAR</b>   <i>greek_letter</i>
equop	=   ≠
op	+   -   *   /
greek_letter	<b>ALPHA</b>   <b>BETA</b>   <b>GAMMA</b>   <b>DELTA</b>   <b>EPSILON</b>   <b>ZETA</b>   <b>SIGMA</b>   <b>THETA</b>   <b>OMEGA</b>   <b>PHI</b>   <b>TAU</b>   <b>MU</b>   <b>CHI</b>   <b>KAPPA</b>

Deux types d'expressions distinctes peuvent être rédigées. Les équations, identifiables par le symbole =, et les expressions qui représentent les expressions mathématiques.

## 2 Example AST

Voici un graphe qui représente notre arbre de syntaxe abstraite, résultat de ce texte d'entrée :

$$\int (a + b \cdot x)^{2+\alpha} dx = \frac{(a + b \cdot x)^{3+\alpha}}{b(3+\alpha)}$$
