

Université de Rouen, UFR Sciences et  
Techniques  
Master 1 SSI – Compilation  
M2L, traducteur L<sup>A</sup>T<sub>E</sub>X en MathML

Boubekri, Abdelmalek `abdelmalek.boubekri@etu.univ-rouen.fr`  
Mezheri, Bilal `bilal.mezheri@etu.univ-rouen.fr`

2 janvier 2017

# 1 Introduction

Le projet consiste à réaliser un traducteur  $\text{\LaTeX}$  en MathML, le programme devra prendre les formules entourées du symbole \$ et produire le code MathML correspondant; et doit prendre en compte:

- les opérateurs;
- les identifiants;
- les constantes numériques;
- les fractions;
- les exposants et indices;
- les racines (sqrt et root);
- le groupement avec accolades;
- la somme et le produit itérés;

et tout ça sera intégré à un service web, idéalement la lecture et la sortie se feront sur l'entrée et la sortie standard.

# 2 Grammaire

AXIOME  $\rightarrow$  \$ FORMULE \$

FORMULE  $\rightarrow$  \sqrt RACINE FORMULE  
| \frac FRACTION FORMULE  
| UNDEROVER FORMULE  
| CARACTERE FORMULE  
| OPERATION FORMULE  
|  $\epsilon$

RACINE  $\rightarrow$  [FORMULE]{FORMULE}  
| {FORMULE}

FRACTION  $\rightarrow$  {FORMULE}{FORMULE}

OPERATION  $\rightarrow$   $\wedge$  FORMULE  
| \_ FORMULE

UNDEROVER  $\rightarrow$  \sum\_{FORMULE} {FORMULE}  $\wedge$  {FORMULE}  
| \prod\_{FORMULE} {FORMULE}  $\wedge$  {FORMULE}

CARACTERE  $\rightarrow$  IDENTIFIER | OPERATOR | NUMBER  
| \pm | \alpha | \beta | \infty | \mp  
| \cap | \cup | \subset | \supset | \subseteq | \supseteq

## 3 Fonctionnement

### 3.1 Analyse lexicale avec Flex

Le rôle de Flex est de découper la chaîne entrée en lexèmes (tokenizer) à reconnus grâce aux motifs définis, dans notre exemple les nombres, les identifiants ..., avant de procéder à l'analyse syntaxique.

**Exemple** petit fragment du fichier **l2m.lex**

```
/* La partie declaration Flex ou sont decrits les patrons */
DIGIT [0-9]
NUMBER {DIGIT}*|{DIGIT}*"."{DIGIT}*

%%

/* assigne la valeur du token lu a yyval, et retourne son
   type */
{NUMBER} { yy1val = strdup(yytext); return NUMBER;}
```

### 3.2 Analyse syntaxique avec Yacc

Yacc se charge de la vérification de la syntaxe de la chaîne de caractères à travers la grammaire décrite précédemment et implémentée dans le fichier **l2m.y**, puis d'afficher le résultat en MathML.

## 4 Utilisation

Récupérez l'archive depuis Github: <https://github.com/kabyliano/l2m.git>  
Afin de compiler le programme il suffit de lancer le **Makefile**

```
Make
```

Puis vous pourrez lancer le programme **l2m** avec comme argument le flux  $\text{\LaTeX}$ ; exemple:

```
./l2m "\$\frac{5}{9}\$"
```

## 5 Problèmes non résolus

- Erreur de segmentation (core dumped) dû à la mauvaise gestion de la mémoire.
- Récursivité mal implémentée.

- Conflit Décalage/Réduction (*Shift/reduce*) dû aux conflits entre  $\_$  et  $\wedge$  (POW / UND) et SUM / PROD.