



Sujet de mini-projet de compilation – M1 Informatique

Un traducteur LaTeX vers MathML

{nicolas.bedon,ludovic.mignot,florent.nicart}@univ-rouen.fr

5 décembre 2016

1 Présentation générale de

Le mini-projet de compilation est mutualisé avec le mini-projet de langages Web et représente une fonctionnalité de ce dernier. Il consiste en l'écriture d'un traducteur prenant en entrée une chaîne de caractères représentant une formule mathématique en \LaTeX et produit en sortie un flux MathML représentant cette formule. Par exemple la formule \LaTeX `$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$` produira :

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<math xmlns='http://www.w3.org/1998/Math/MathML'>
```

```
  <mrow>
    <mrow>
      <mi>x</mi>
      <mo>=</mo>
      <mfrac>
        <mrow>
          <mo>±</mo>
          <mi>b</mi>
          <mo>&#x00b1</mo>
          <msqrt>
            <mrow>
              <msup>
                <mrow>
                  <mi>b</mi>
                </mrow>
                <mrow>
                  <mn>2</mn>
                </mrow>
              </msup>
              <mo>-</mo>
              <mn>4</mn>
              <mi>a</mi>
              <mi>c</mi>
            </mrow>
          </msqrt>
        </mrow>
        <mrow>
          <mn>2</mn>
          <mi>a</mi>
        </mrow>
      </mfrac>
    </mrow>
  </math>
```

Le rendu en \LaTeX et
en MathML doit
ressembler à :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

2 Spécifications

Le langage reconnu et traduit ne sera qu'un sous-ensemble du langage de l'environnement mathématique de \LaTeX . C'est-à-dire du code que l'on peut écrire entre les symboles $\$$. Puisque nous ne traiterons que les formules mathématiques et que nous ignorerons tout le reste du langage \LaTeX , nous pouvons même exclure les symboles $\$$ (qui servent à délimiter les formules mathématiques) de nos entrées. L'environnement mathématique étant lui-même assez vaste, nous restreindrons la traduction aux éléments suivants :

- Les symboles, traditionnellement noté $\$a\$$ en \LaTeX , qui devront produire un "math identifier" : $\langle \text{mi} \rangle a \langle \text{mi} \rangle$;
- Les constantes numériques ($\$2\$$), qui devront produire un "numerical" : $\langle \text{mn} \rangle 2 \langle \text{mn} \rangle$;
- les opérateurs mathématiques simples qui devront produire un "math operator" : $\langle \text{mo} \rangle + \langle \text{mo} \rangle$, auxquels on ajoutera : \pm ($\$ \backslash \text{pm} \$ \rightarrow \&\text{plusmn} ;$), \mp ($\$ \backslash \text{mp} \$ \rightarrow \&\#x2213 ;$), \cap ($\$ \backslash \text{cap} \$ \rightarrow \&\text{cap} ;$), \cup ($\$ \backslash \text{cup} \$ \rightarrow \&\text{cup} ;$), \subset ($\$ \backslash \text{subset} \$ \rightarrow \&\text{sub} ;$), \supset ($\$ \backslash \text{supset} \$ \rightarrow \&\text{sup} ;$), \subseteq ($\$ \backslash \text{subse} \text{teq} \$ \rightarrow \&\text{sube} ;$) et \supseteq ($\$ \backslash \text{supsete} \text{teq} \$ \rightarrow \&\text{supe} ;$) ;
- les fractions de deux expressions ($\$ \backslash \text{frac } e \text{ } f \$ \rightarrow \langle \text{mfrac} \rangle \langle \text{mfrac} \rangle$) :

$$\frac{e}{f}$$

- les expressions f en exposant d'une expression e ($\$e^f\$ \rightarrow \langle \text{msup} \rangle \langle \text{msup} \rangle$) : e^f
- les expressions f en indice d'une expression e ($\$e_f\$ \rightarrow \langle \text{msub} \rangle \langle \text{msub} \rangle$) : e_f
- le groupement avec des accolades pour des expressions complexes ($\$\{e\}\$$), exemple avec $\$ \backslash \text{frac} \{x * \sqrt{y}\} \{2a\} \$$:

$$\frac{x * \sqrt{y} + z}{2a}$$

- l'opérateur de racine ($\$ \backslash \text{sqrt}[n] \text{ } x \$ \rightarrow \langle \text{mroot} \rangle \langle \text{mroot} \rangle$ ou $\langle \text{msqrt} \rangle \langle \text{msqrt} \rangle$) : $\sqrt[n]{x}$
- la somme itérée ($\$ \backslash \text{sum}_{i=1}^n u_i \$ \rightarrow \langle \text{munderover} \rangle \langle \text{munderover} \rangle$) :

$$\sum_{i=1}^n u_i$$

- et le produit itéré ($\$ \backslash \text{prod}_{i=1}^n u_i \$ \rightarrow \langle \text{munderover} \rangle \langle \text{munderover} \rangle$) :

$$\prod_{i=1}^n u_i$$

On fera remarquer que le groupement utilisant des accolades concerne la syntaxe \LaTeX uniquement et non celle de la formule : $\$\{a+b\} * 2\$$ donne $a + b * 2$ au lieu de $\$(a+b) * 2\$$ qui donne $(a + b) * 2$

3 Ce qu'il vous est demandé

Vous devez écrire en C, et en utilisant *flex* et *bison* pour les parties concernant les analyses lexicales et syntaxiques, un programme permettant de réaliser chacune des fonctions présentées dans la section précédente.

Celui-ci devra pouvoir être exploité facilement par un service Web (écrit en PHP, Java ou le langage que vous aurez choisi pour votre projet web coté serveur) qui lui fournira la chaîne \LaTeX à traduire et récupérera le résultat, le plus simple étant de fournir l'entrée comme argument de l'exécutable et de récupérer le résultat sur la sortie standard (mais ce n'est qu'une des façons possible de faire). Vous prendrez en compte la gestion de la détection des erreurs afin que le service Web puisse indiquer si la traduction a été effectuée avec succès ou non. L'analyse des erreurs n'est pas demandée. La date de rendu du projet web étant antérieure, on pourra utiliser un bouchon dans ce dernier, c'est à dire un exécutable qui produit un flux MathML constant quelque soit l'entrée en attendant que le traducteur soit prêt.

Bien entendu, votre projet devra être écrit le plus proprement possible : algorithmique adaptée, code clair et commenté.

Vous veillerez à vérifier la conformité de vos sorties à l'aide d'un navigateur Web. (N'oubliez pas que le rendu peut varier en MathML si la formule est en mode **inline** ou **bloc**.)

Vous pouvez étendre le projet si vous le souhaitez, en rajoutant des fonctionnalités par exemple. Cependant, ne le faites que si la base qui vous est demandée est implantée et fonctionne correctement : il est préférable d'avoir un projet qui fait correctement le minimum plutôt que d'avoir un projet étendu dont le minimum demandé ne fonctionne pas.

Votre projet devra être rendu avec un jeu d'exemples illustrant le mieux possible ses fonctionnalités, ainsi qu'un rapport contenant un rapport de développement et un manuel d'utilisation.

Le travail est à réaliser individuellement ou en binôme. La date limite de rendu du projet est fixée au **3 janvier 2017**. Il devra être livré dans une archive au format tar gzippé de nom `FrancoisDupontJacquesDurant.tar.gz` pour un binôme (si Francois Dupont et Jacques Durant sont vos noms), envoyée en pièce jointe à un courriel de sujet “Projet L2M” à l’adresse de courriel de votre chargé de TP (Nicolas.Bedon@univ-rouen.fr, Ludovic.Mignot@univ-rouen.fr ou Florent.Nicart@univ-rouen.fr). Si vous avez fait le projet en binôme à cheval sur deux groupes de TP, votre projet est à envoyer aux deux chargés de TP. Vous vous mettrez en copie du courriel pour vérifier que vous n’oubliez pas la pièce jointe. L’extraction du fichier d’archive devra produire un répertoire de nom `FrancoisDupontJacquesDurant` contenant le code source de votre projet, un `makefile`, des jeux d’exemples et un rapport de projet.

Votre projet fera l’objet d’une soutenance sur machine.

4 Références

- Une référence du langage MathML est disponible à cette adresse : <https://developer.mozilla.org/fr/docs/Web/MathML/Element>
- Pour ceux qui ne connaissent pas \LaTeX , un petit guide pour débutant est disponible à cette adresse : <https://www.ljll.math.upmc.fr/privat/documents/manuelLatex.pdf>