

Projet de ADS4

Interpréteur de texte HTML

Lydia ITOUCHENE
Ibrahim BAMBA

Table des matières

1.INTRODUCION	1
2.PRESENTATION DU PROJET	1
3.CONCEPTION	1
4.COMPILATION ET EXECUTION	2
❖Compilation :	2
❖Exécution :	2
5.CONCLUSION	4

1. INTRODUCTION

Le projet qui nous a été confié est de réaliser un analyseur lexical et syntaxique afin de construire l'arbre de syntaxe abstraite d'un document décrit par une grammaire donnée. Pour bien répondre au problème, nous allons présenter les différentes étapes du projet et les solutions que nous avons utilisées.

2. PRESENTATION DU PROJET

Le but du projet est de permettre la traduction d'un texte écrit dans le langage SimpleText (inspiré du LATEX) en HTML.

HTML : L'HyperText Markup Language, généralement abrégé **HTML**, est le format de données conçues pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom.



3. CONCEPTION

Nous avons fait un analyseur lexical et syntaxique afin de construire l'arbre de syntaxe abstraite d'un document décrit par la grammaire ci-dessous.

- * DOCUMENT -> DECLARATIONS CORPS EOF
- * DECLARATIONS -> \set{ ID } { CONSTANCE_COULEUR } DECLARATIONS
 | \abb{VALABB}{SUITE_ELEMENTS} DECLARATIONS
 | E
- * CORPS -> \begindoc SUITE_ELEMENTS \enddoc

```

* SUITE_ELEMENTS -> ELEMENT SUITE_ELEMENTS | E
* ELEMENT -> MOT
    | VALABB
    | \linebreak
    | \bf{ SUITE_ELEMENTS }
    | \it{ SUITE_ELEMENTS }
    | \couleur{ VAL_COL } { SUITE_ELEMENTS }
    | \for{ INT } { SUITE_ELEMENTS }
    | ENUMERATION
* VAL_COL -> CONSTANTE_COULEUR | ID
* ENUMERATION -> \beginenum SUITE_ITEMS \endenum
* SUITE_ITEMS -> ITEMS SUITE_ITEMS | E
* ITEM -> \item SUITE_ELEMENTS

```

Cette grammaire correspond aux trois étapes du projet demandées à savoir l'interpréteur de texte (la suite mot) et tout ce qu'on peut faire avec les mots(suite de caractères sauf les (\t,\r,\n)) c'est à dire de mettre quelques-uns en italique ,en gras, en couleur ou en mettant une phrase en abréviation , et en fin la partie bonus qui consiste à ajouter des boucles qui permettront de répéter plusieurs fois la même partie de texte.

Ensuite, Nous avons fait un interpréteur pour l'arbre construit qui va produire la traduction du document lu vers HTML.

4. COMPILATION ET EXECUTION

❖ Compilation :

Pour compiler notre projet il suffit de compiler le fichier « makefile » qui contient toutes les commandes pour la compilation de tous les fichiers en écrivant sur le terminal : make.

❖ Exécution :

Pour l'exécution : on teste d'abord tous les fichiers BAD pour s'assurer que le programme rejette bien tous les fichiers BADs:

«java Main testBad/badCompil»

«java Main testBad/badCompilBis»

«java Main testBad/badParser»

«java Main testBad/badParserBis»

```
mouloud: Screenshot
Symbol : VALABB | Value : \Mr
Symbol : RACC
Symbol : LACC
Symbol : MOT | Value : Monsieur
Symbol : RACC
Symbol : BEGINDOC
Symbol : MOT | Value : Salut
Symbol : VALABB | Value : \Mr
Symbol : MOT | Value : you. How are you doing ? Are you
Symbol : VALABB | Value : \nr
Symbol : MOT | Value : E.M ! ?
Symbol : MOT | Value : Il y a erreur dans l'utilisation de l'abréviation.
Symbol : VALABB | Value : \nr
Symbol : MOT | Value : n'a pas été déclaré.
Symbol : MOT | Value : Cette erreur ne sera détectée que lors de la compilation.
Symbol : ENDDOC
Symbol : EOF
The code is correct.

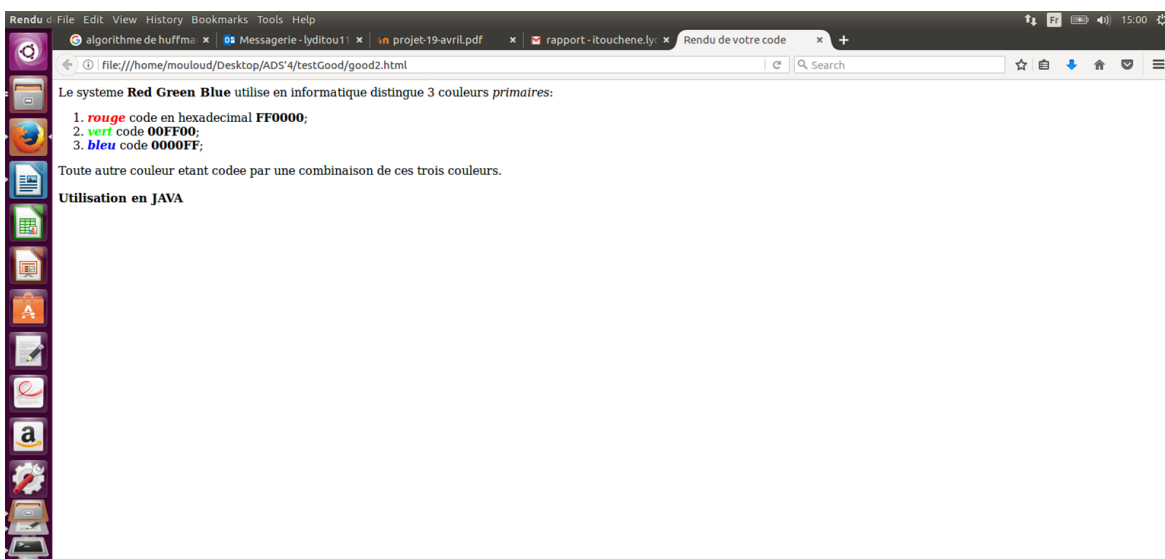
Compilation error
java.lang.Exception: Abbreviation '\nr' is not declared
mouloud@lydia-virtualbox:~/Desktop/ADS'4$ java Main testBad/badParser
Symbol : BEGINDOC
Symbol : BEGINENUM
Symbol : ENDENUM
Symbol : COLOR
Symbol : LACC
Symbol : ID | Value : FF0073
The code is not correct.
ParserException: Parser exception : 'RACC' expected but 'Symbol : MOT | Value : 37' found at line 8 and column 17
mouloud@lydia-virtualbox:~/Desktop/ADS'4$ java Main testBad/badParserBis
Symbol : SETTER
Symbol : LACC
Symbol : ID | Value : FFFFFFFF
Symbol : RACC
Symbol : LACC
Symbol : MOT | Value : blanc
Symbol : RACC
Symbol : ABB
Symbol : LACC
The code is not correct.
ParserException: Parser exception : 'VALABB' expected but 'Symbol : MOT | Value : ppp' found at line 2 and column 6
mouloud@lydia-virtualbox:~/Desktop/ADS'4$
```

Ensuite on exécute les fichiers good :

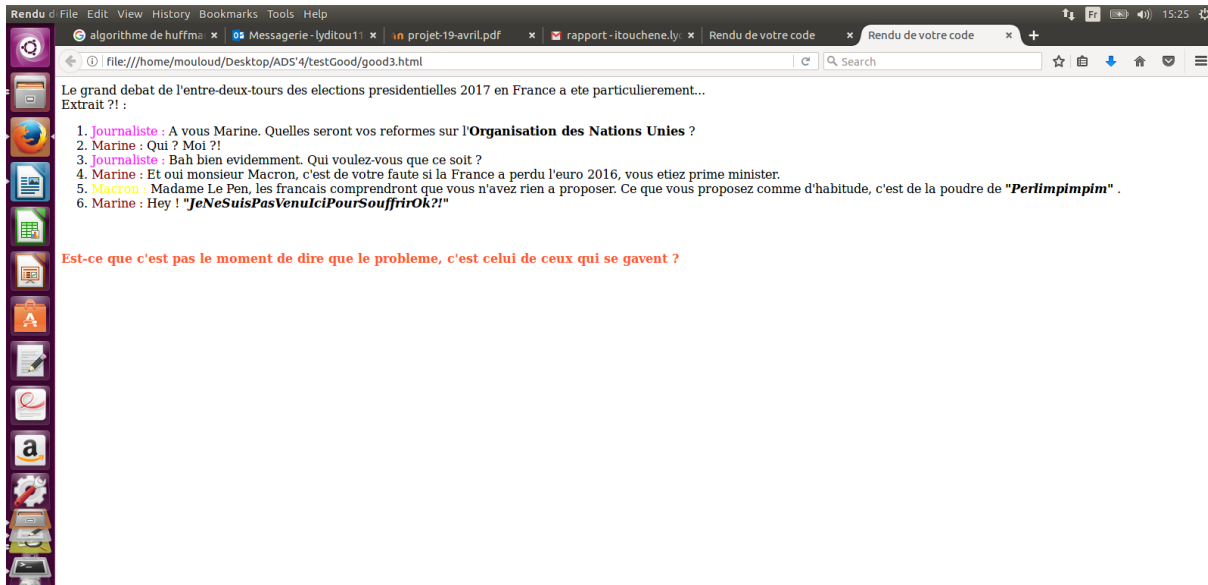
«java Main testGood/good1»



«java Main testGood/good2»



«*java Main testGood/good3*»



5. CONCLUSION

Ce projet fut très intéressant à plusieurs points de vue. D'une part, la mise en place d'une Organisation entre les deux membres du binôme fut enrichissante et nous a permis de Tester nos connaissances. D'autre part, Il nous a permis de bien comprendre les principes de l'analyse lexicale et syntaxique.