

Rapport du projet de compilation

Maazouz Mehdi, Lecocq Alexis

May 9, 2017

Annee Academique 2016-2017

Contents

1	Introduction	2
2	Grammaire	2
2.1	Le “For”	3
2.2	Le “If”	3
3	Problèmes rencontrés	3
4	Conclusion	3

1 Introduction

Dans le cadre du cours de Compilation donné en 3^{ème} Bachelier, nous avons dû réaliser un programme capable de générer un fichier contenant du texte en fonction de certaines données reçues en paramètre. Pour ce faire, le langage de programmation utilisé est le Python.

2 Grammaire

Nous avons essayé de respecter au mieux la grammaire donnée, initialement dans l'énoncé du projet. Cependant, nous avons procédé à quelques ajouts et modifications afin d'atteindre les objectifs demandés dans l'énoncé.

Pour les opérateurs arithmétiques, nous avons pris la peine de définir dans la grammaire les variables “term” ainsi que “factor” afin d'éviter le cas où la grammaire serait ambiguë.

```
def p_intop_plus_minus(p):
    '''intop : intop PLUS term
              | intop MINUS term'''
    p[0] = (p[2], infos(p), p[1], p[3])

def p_intop_term(p):
    '''intop : term'''
    p[0] = p[1]

def p_term_times_divide(p):
    '''term : term TIMES factor
             | term DIVIDE factor'''
    p[0] = (p[2], infos(p), p[1], p[3])

def p_term_factor(p):
    '''term : factor'''
    p[0] = p[1]

def p_factor_num(p):
    '''factor : INTEGER'''
    p[0] = p[1]

def p_factor_variable(p):
    '''factor : variable'''
    p[0] = p[1]

def p_variable(p):
    '''variable : VARIABLE'''
    p[0] = ('variable', infos(p), p[1])
```

Concernant l'ajout des booléens, nous nous sommes inspirés du cours. Nous commençons avec la variable `boolop` qui dérive sur une autre variable `boolop`, un opérateur "AND" ou "OR" ainsi que la variable `bool`. Cette dernière correspond au lexème boolean ou à la variable "comparison". Qui est le résultat de deux variables "intop" muni d'un lexème comparator.

```
def p_boolop_boolop_bool(p) :
    '''boolop : boolop AND bool
    | boolop OR bool'''
    p[0] = (p[2], infos(p), p[1], p[3])

def p_boolop_bool(p) :
    '''boolop : bool'''
    p[0] = p[1]

def p_bool_boolean(p) :
    '''bool : BOOLEAN'''
    p[0] = p[1]

def p_bool_comparison(p) :
    '''bool : comparison'''
    p[0] = p[1]

def p_comparison(p) :
    '''comparison : intop COMPARATOR intop'''
    p[0] = (p[2], infos(p), p[1], p[3])
```

2.1 Le "For"

2.2 Le "If"

3 Problèmes rencontrés

4 Conclusion