Compilation - TD6

Grammaires à attributs

1 Grammaire du langage Alpha

```
System -> ’system’ id ’(’ InputDeclList ’)’
                      ’returns’ ’(’ OutputDeclList ’)’
                      ’var’ LocalDeclList ;
                      EquationBlock;
        | ’system’ id ’(’ InputDeclList ’)’
                      ’returns’ ’(’ OutputDeclList ’)’
                      EquationBlock;

InputDeclList -> VarDeclList
OutputDeclList -> VarDeclList
LocalDeclList -> VarDeclList

VarDeclList -> VarDeclList ’;’ VarDecl
             | VarDecl

VarDecl -> id ’:’ Domain ’of’ Type;

Type -> ’real’ | ’boolean’ | ’integer’    // peut être un token

Domain -> ’{’ IdList ’|’ IneqList ’}’

IdList -> NonEmptyIdList |

NonEmptyIdList -> NonEmptyIdList ’,’ id
                | id

IneqList -> NonEmptyIneqList |

NonEmptyIneqListt -> NonEmptyIneqList’;’ Ineq
                   | Ineq

Ineq -> AffineExp Comparator AffineExp

Comparator -> ’<’ | ’>’ | ’=’ | ’<=’ | ’>=’  // peut être un token

AffineExp -> TermSum | ’-’ TermSum

TermSum  ->  TermSum ’+’ Term
           | TermSum ’-’ Term
           | Term

Term -> integer id | id | integer
```

```
EquationBlock -> ’let’ EquationList ’tel’

EquationList -> EquationList ’;’ Equation
              | Equation

Equation -> id ’=’ Expression

Expression -> Expression ’+’ Term
            | Expression ’-’ Term
            | Term
            | ’-’ Term

Term -> Term ’*’ Factor
      | Term ’/’ Factor
      | Factor

Factor -> ’case’ ExpressionList ’esac’
        | Domain ’:’ Expression          // Quel parenthésage
        | Expression ’.’ AffineFunction  // entre ces deux là ?
        | (Expression)

ExpressionList -> ExpressionList ’;’ Expression
                | Expression

AffineFunction -> ’(’ IdList ’->’ AffineExpList ’)’

AffineExpList -> NonEmptyAffineExpList |

NonEmptyAffineExpList -> NonEmptyAffineExpList ’,’ AffineExp
                       | AffineExp
```