Université d’Alger 1 Année Universitaire 2022-2023

Faculté des Sciences Licence SI

Département MI Module: Compilation

**Projet de TP: Création d’un Mini Compilateur**

Le but de ce projet est de réaliser un mini-compilateur d’un langage appelé « MiniAlgo », en effectuant les différentes phases de la compilation : lexical, syntaxique et sémantique.

1. **Description du Langage MiniAlgo**

**II.1 La structure générale**

**LANGAGE** MiniAlgo

**VAR**

Liste Declarations

**BEGIN**

Liste Instructions

**END**

1. **Liste des déclarations**

Nous pouvons déclarer dans ce langage des variables simples et des constantes ou des fonctions.

* Variable simple : NomVariable :Type ;
* Constante : NomConstante : CONST Type;

**Exemple**

bism=33 : **CONST INT**;

A83\_n, app , reste : **INT**; {*déclaration des entiers*}

Kms, LL6, Jn5 : **FLOAT**; {*déclaration des réels*}

Reponse : **BOOL**; {*déclaration des booleen*}

v

* Déclaration d’une fonction

Type **Function** NomFonction

**VAR**

ListeDec

**BEGIN**

Instructions

**Return** var ;

**END**

* Les types possibles sont : INT, FLOAT et BOOL
* La déclaration d’un ensemble de variables s’effectue en séparant les noms de variables par ‘,’.
* Chaque nomVariable est un identificateur. Un identificateur (idf) commence par une lettre alphabétique (a-z ou A-Z) suivie d’une suite de chiffres (0-9), lettres alphabétiques ( a-z ou A-Z), ou bien le symbole '\_' . Sa taille ne doit pas dépasser 12 caractères.

Les identificateurs doivent respecter les conditions suivantes :

* L’idf ne doit pas contenir de symboles '\_' successifs.
* Le le symbole '\_' ne doit pas apparaître à la fin de l’idf.
* Il y a deux types de constantes :

1. Les constantes entières : C’est une suite de chiffres, qui peut être signée ou pas
2. Les constantes réelles : C’est une suite de chiffres contenant le point décimal
3. **Liste des Instructions**

Dans notre langage, **SEULES** les instructions suivantes sont autorisées :

|  |  |  |
| --- | --- | --- |
| Instruction | Description | Exemple |
| Affectation | Idf =expression arithmétique ; | A =2 ;  A =C+D \*Z/E; |
| Boucle | **While ( condition)**  **BEGIN**  **Bloc Instructions**  **END** | While(i>2)  BEGIN  B=B+2;  END |
| condition | If (Condition)  **BEGIN**  Bloc instructions  **END** | If (a==2)  BEGIN  B=5 ;  C=B+1 ;  END |

* Les expressions arithmétiques utilisent les opérateurs arithmétiques : *+ - \* /*

Les conditions utilisent les opérateurs relationnels : < <= <> == > >=

1. **Les commentaires**

* Les commentaires sont une suite de caractères quelconques compris entre ‘{‘ et ‘}’, ne contenant pas de ‘}’.

1. **Travail à réaliser :**

Ci-dessous les différentes phases à effectuer afin de réaliser le compilateur demandé.

* **Analyse lexicale avec l’outil FLEX.**
* **Analyse syntaxique avec l'outil Bison**
* **Analyse sémantique pour les erreurs suivantes :**
* Idf non déclaré
* Idf double déclarée
* Non compatibilité de type
* Changement de valeur d’une constante
* **Gestion de la table de symboles** La table doit avoir au minimum les champs suivants :
* Nom : l’identificateur qui indique le nom de la variable ou constante.
* Type : le type de la variable ou la constante
* Const : indique si la variable est constante ou non.
* **Traitement des erreurs :**

Il est demandé d’afficher les messages d’erreurs adéquats à chaque étape du processus de compilation, en précisant le numéro de ligne et la colonne de l’erreur.

Bon courage