Rapport d'Activité n°2 - Projet Compilation

Lucie BOUCHER, Rémi BOURDAIS, Lola MONTIGNIER

05 Décembre 2022

1 Choix LL(1) et Transformation de la grammaire

Nous avons modifié la manière de prioriser les calculs dans notre grammaire afin que l'arbre syntaxique ne penche plus vers la gauche.

L'arbre syntaxique associé à cette grammaire est droit et plus lisible.

Nous avions une grammaire partiellement LL(1) au premier jalon à cause de la règle 'rulelse'. Désormais, après modification, notre grammaire n'est pas LL(1) à cause des règles 'rulelse' et 'expr'.

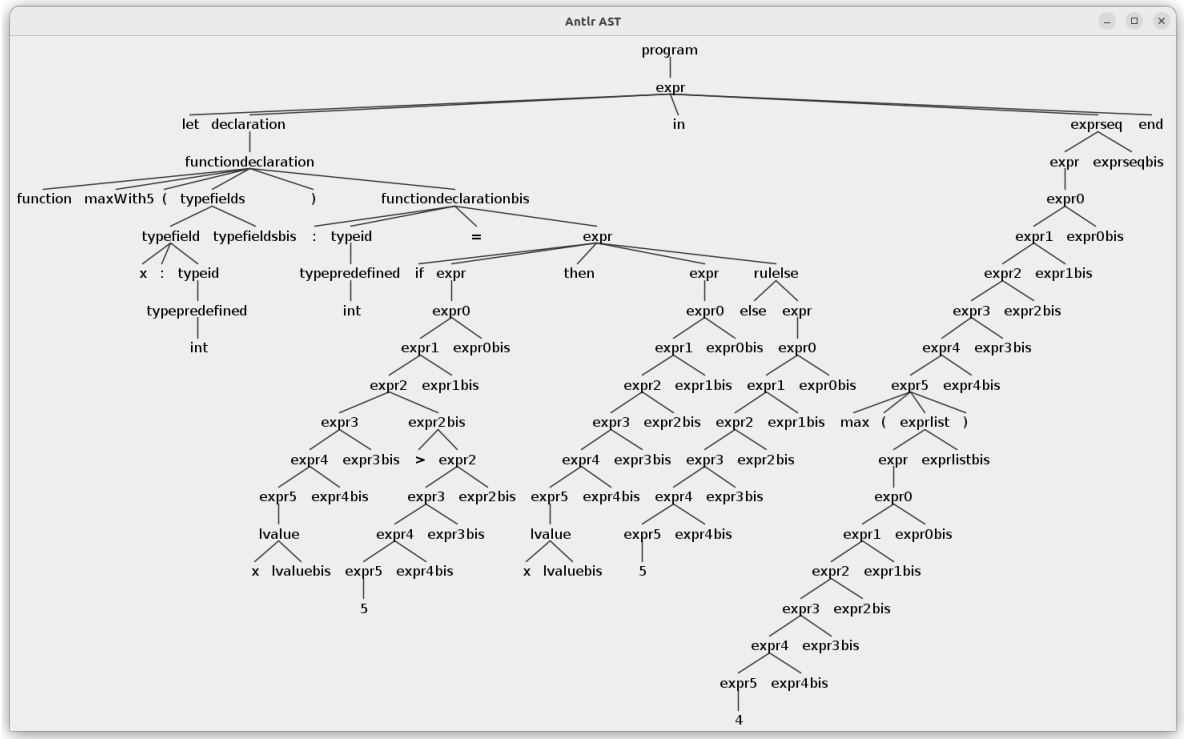


Figure 1: Ancien arbre syntaxique de la grammaire partiellement LL(1) penchant à gauche

On obtient donc l'arbre suivant:

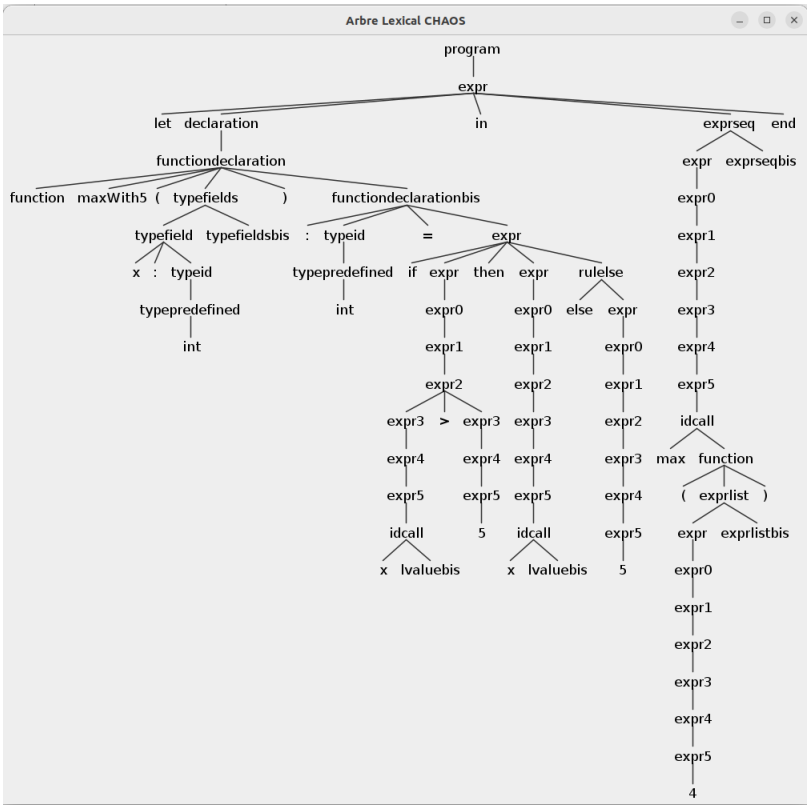
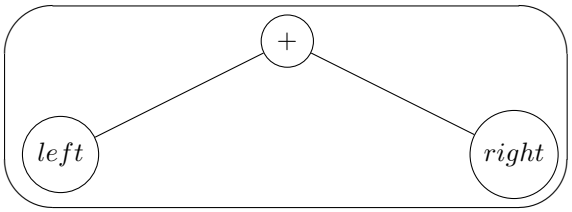


Figure 2: Arbre syntaxique actuel de la grammaire partiellement LL(1)

2 Construction de l'AST

Nous somme ensuite passer à la construction de l'arbre syntaxique abstrait (AST).

Nous avons, dans un premier temps, réfléchis, règle par règle, aux éléments qui allaient apparaître ou non dans l'AST. Voici par exemple un mini-AST avec la règle 'plus'.



Ensuite nous avons suivi la procédure pour construire un AST comme dans le TP n°2. Nous avons associé des labels à nos règles et créé des classes java associés à ces labels. De plus, la classe AstCréator permet de sélectionner ou non les éléments pour consrtuire l'AST. Enfin grâce à GraphViz, un logiciel de visualisation graphique open source, nous visualisons notre AST. Ci-dessous, une première version de notre AST sur le programme des huit reines.

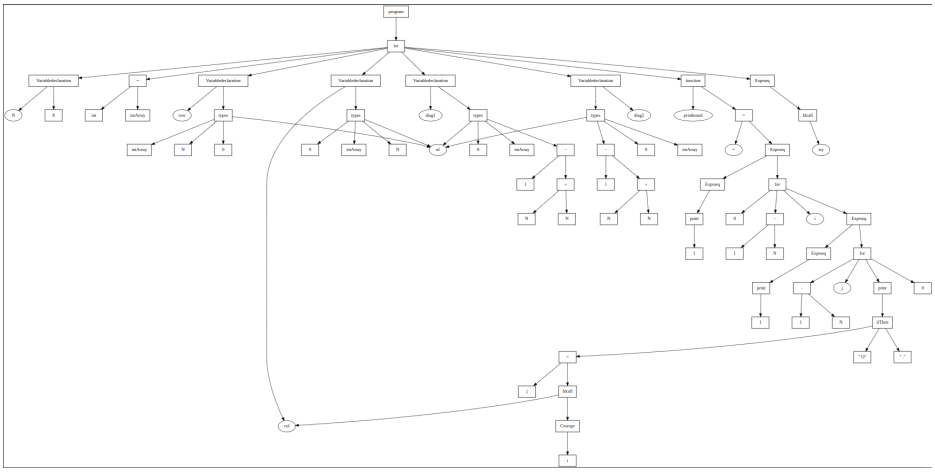


Figure 3: Premier AST obtenu contenant trop de noeuds unaires

Ce premier AST nous a permis de voir ce qui n'allait pas dans notre code. Trop de noeuds unaires étaient encore présents. De plus, certaines parties des programmes test n'étaient pas présentes dans l'arbre donc nous avons amélioré notre arbre afin qu'il soit le plus lisible possible avec la totalité des informations représentées.Ci-dessous, la version finale de notre AST sur le programme des huit reines.

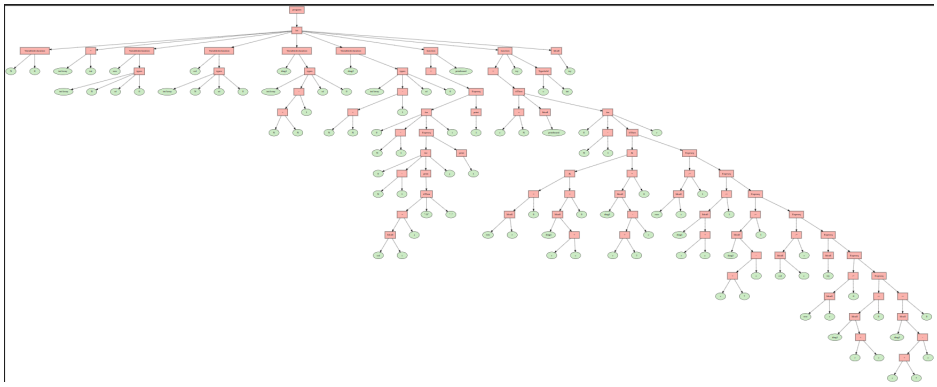


Figure 4: AST final obtenu après modifications

3 Construction des Tables des symboles

Pour finir nous avons construit les tables des symboles. Chaque table des symboles est associée à un bloc du programme analysé. Nous avons construit la classe 'Table' qui contient une liste de 'VarType' et une liste de 'ProFonc'. Nous avons une classe TDSCreator qui parcours l'AST et ajoute les éléments dans la table lorsqu'ils sont déclarés. Ci-dessous les tables des symboles pour le programme Declagogo. Si une table ou une partie de la table est vide. Nous avons décider de ne pas l'afficher par soucis de lisibilité.

```
| TABLE ID=0
| Nom de la fonction=programme
| ID du pere=-1
| Types:
| Nature | Name          | Type      | Deplacement  | Dimensions   | Type des éléments
| Type   | intArray      | Array of  | 2            | 1            | int,
| Type   | intcro        | TypeList  | 6            | 3            | int,string,int,
| Type   | int2          | Type      | 7            | 1            | int,
| Variables:
| Nature | Name          | Type      | Deplacement  | Dimensions
| Var    | N             | Int       | 1            | 1
| Var    | intcro.c      | int       | 3            | 1
| Var    | intcro.d      | string    | 4            | 1
| Var    | intcro.e      | int       | 5            | 1
| Var    | row           | intArray  | 8            | N
| Var    | col           | intArray  | 9            | N
| Var    | diag1         | intArray  | 10           | N+N-1
| Var    | diag2         | intArray  | 11           | N+N-1
| Procédures:
| Nature     | Name          | Type      | Nb Args
| procedure  | try2          | null      | 0
| Fonctions:
| Nature     | Name          | Type      | Nb Args | Types Arguments
| fonction   | try           | null      | 1       | int,
| fonction   | try3          | int       | 1       | int,

| TABLE ID=1
| Nom de la fonction=try
| ID du pere=0
| Variables:
| Nature | Name          | Type      | Deplacement  | Dimensions
| Var    | c             | int       | 1            | 1

| TABLE ID=2
| Nom de la fonction=try2
| ID du pere=0

| TABLE ID=3
| Nom de la fonction=try3
| ID du pere=0
| Variables:
| Nature | Name          | Type      | Deplacement  | Dimensions
| Var    | c             | int       | 1            | 1
```

Figure 5: Les TDS obtenus après modifications