

## **Projet d'analyse syntaxique**

Analyseur syntaxique du langage TPC, incluant les tableaux  
à une dimension constante entière

# Choix lors de la réalisation du projet

## Gestion du conflit Bison

Pour résoudre le conflit du ELSE, nous avons utilisé comme mentionné en cours/TD, l'instruction `%expect 1` pour indiquer qu'on attend uniquement un seul conflit.

On peut indiquer ça et ne pas changer la grammaire (ambiguë) car la façon dont Bison gère le conflit est satisfaisante.

## Arbre Abstrait

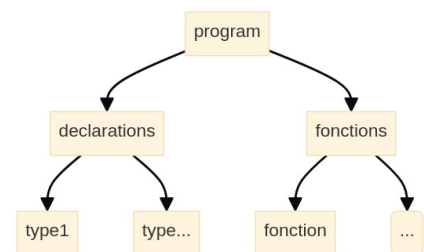
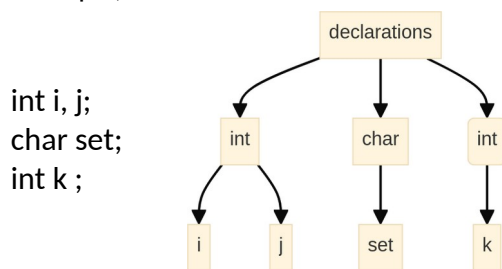
`declarations`

Ce nœud est présent au début du programme ainsi qu'au début du corps des fonctions.

Il répertorie les variables par type, par instruction.

S'il n'y a pas de déclaration de variable, on a pas de fils de `declarations`

Par exemple,



`fonctions`

C'est une liste de `fonction`. Chaque nœud fonction est frère d'une autre fonction.

S'il n'y a pas de fonction, on a pas de fils de `fonctions`.

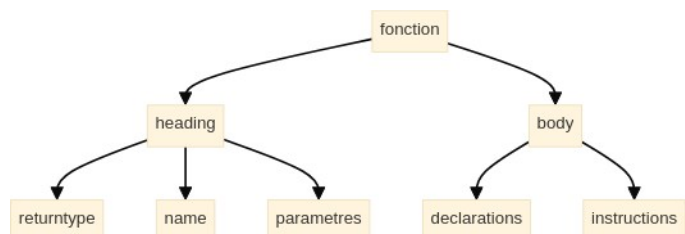
`fonction`

C'est une fonction, ce nœud a deux fils.

`heading`

C'est l'entête de la fonction. Elle est composée :

- du type de retour de la fonction
- du nom de la fonction
- des paramètres de la fonction, soit `void` ou une liste d'arguments.



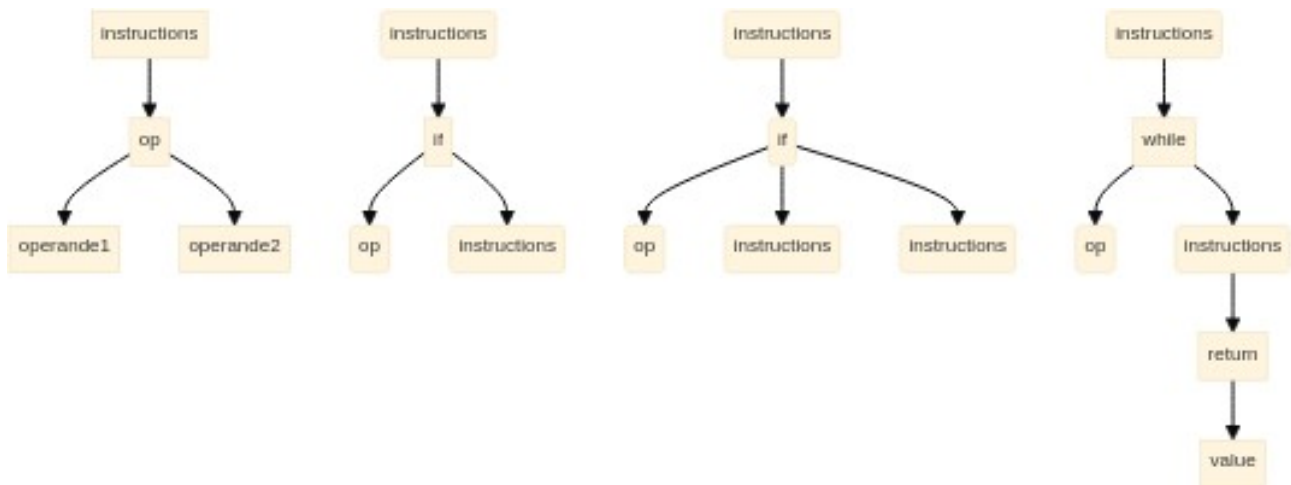
`body`

C'est le corps de la fonction. Il est composé du nœud `declarations` ainsi que d'instructions.

## instructions

Ce nœud contient les instructions d'une fonction ou d'une structure de contrôle (if ou if else ou while) ou d'un appel de fonction.

S'il n'y a pas d'instructions, alors ce nœud n'a pas de fils.



Pour le if else, on a deux fils instructions de if. Le premier fils est le then, et le deuxième est le else.

Pour le return, il contient une valeur qui est son fils, sinon il n'a pas de fils.

Pour les appels de fonctions, leur nom est la racine de l'arbre et ses fils sont les arguments (ils sont frères entre eux). S'ils n'y a aucun argument, alors la racine n'a pas de fils.

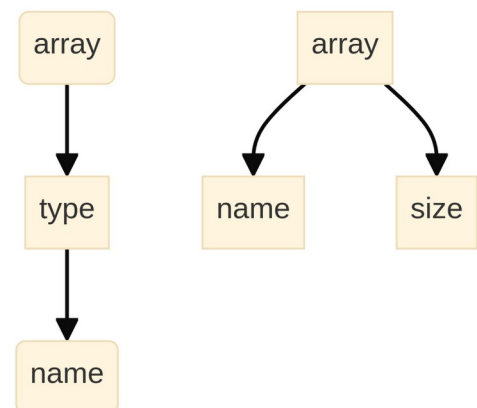
## Grammaire des tableaux et arbre abstrait

La grammaire des tableaux a été plutôt rapide à appliquer. Néanmoins il a fallu faire un choix sur l'affichage dans l'arbre abstrait.

Dans les expressions, un tableau s'affiche avec son nom et sa taille/son index en fils de `array`.

Pour les arguments de fonction, le nom du tableau est fils du type des éléments du tableau, qui est fils d'`array`.

Pour les déclarations, c'est la même chose que les arguments de fonctions avec la taille



## Difficultés rencontrées

Nous avons fait face à quelques difficultés lors de nos premiers exercices sur Bison, qui se sont dissipées au fil du temps.

Nous avons également beaucoup cherché la manière d'implémenter l'arbre abstrait et la création d'un fils intermédiaire (dans une variable locale).

Nous avons aussi eu un peu de difficulté sur la gestion des options avec `getopt`. A noter que les options `--t` (et `-h`) fonctionnent alors que nous avons un doute pour déterminer si elles devraient fonctionner.

Aussi, il a été difficile de déceler les 3 tests Core qui bloquaient sur le bac à sable. Finalement, c'était juste une incompréhension au niveau du lexème `CHARACTER`.