

## Introduction :

Le but du projet d'analyse syntaxique est de produire un analyseur syntaxique en faisant usage des outils Flex et Bison. Le langage source tpc est une sorte de sous-ensemble du langage C. Lorsque le fichier d'entrée ne contient aucune erreur lexicale et/ou syntaxique, il est convenu que l'analyseur fournisse un arbre abstrait, traduit du fichier d'entrée. Il est également convenu que cet arbre abstrait soit affichable sur la sortie standard.

## Première implémentation :

En premier lieu, j'ai produit un analyseur qui fonctionnait, mais qui ne respectait pas parfaitement la grammaire imposée, des erreurs cachées dans des fichiers étaient considérées comme correcte. De plus, des règles étaient manquantes, notamment pour les tableaux.

## Rendu final :

Note: 88,34 / 100,00

### Rapport d'évaluation 📄 [-]

#### ▼ Failed tests

Test 14: Core - Good - 14  
Test 15: Core - Good - 15  
Test 31: Core - Good - 31  
Test 121: Ext-Array - Good - 02  
Test 124: Ext-Array - Good - 05  
Test 126: Ext-Array - Good - 07  
Test 127: Ext-Array - Good - 08  
Test 128: Ext-Array - Good - 09  
Test 129: Ext-Array - Good - 10  
Test 130: Ext-Array - Good - 11  
Test 131: Ext-Array - Good - 12  
Test 132: Ext-Array - Good - 13  
Test 133: Ext-Array - Good - 14  
Test 134: Ext-Array - Good - 15  
Test 135: Ext-Array - Good - 16  
Test 136: Ext-Array - Good - 17  
Test 137: Ext-Array - Good - 18  
Test 138: Ext-Array - Good - 19  
Test 139: Ext-Array - Good - 20

Ma version finale du projet affiche 88,34% de réussite au bac à sable fournis, en étant bloquée à 86% pendant un bon moment, à cause des tableaux, blocage qui subsiste.

## Difficultés :

La première difficulté rencontrée fut dans l'établissement du makefile correspondant. En effet, j'ai eu du mal à comprendre quelle règle devait être compilée avant l'autre. J'ai par ailleurs fini par obtenir l'aide de mon chargé de TD, Monsieur, qui a fini par me fournir son makefile (ne trouvant pas d'où venait le problème dans le mien). J'ai également du faire appel par mail à Monsieur Laporte pour des problèmes liés au bac à sable.

J'ai par la suite rencontré des problèmes pour lancer le bac à sable sur mon projet, à cause d'une extension ".tab.h" qui n'avait en fait pas lieu d'exister aux yeux du banc de test. En le remplaçant par un simple ".h", l'erreur a disparu. J'ai donc par la suite, pu commencer les évaluations.

J'avais un problème avec les strepy, problème réglé en implantant une union ayant pour champs les byte, comp, ident, num.

## Compilation et exécution :

Je compile mon projet de la manière suivante : make

./bin/tpcas [OPTIONS] < FILE.tpc

où [OPTIONS] correspond à une option à choisir ;

- -h ou -help pour afficher une description de l'interface utilisateur
- -t ou -tree pour afficher l'arbre abstrait sur la sortie standard.
-

et FILE.tpc correspond au fichier à traiter se situant dans le répertoire test, qui contient un panache de tests bons (good) comme mauvais (syn-error).

Le fichier test.sh est un fichier écrit en bash permettant de tester le panache de tests.

Exécutez la commande suivante pour exécuter le fichier test.sh :

- Pour obtenir les droits d'exécution : `chmod +x test.sh`
- Pour lancer le script : `./test.sh` (produit un rapport.txt dans le dossier rep)

Pour nettoyer le contenu des fichiers binaires (bin) et objets (obj) : `make clean`

Sources :

- <https://totodu.net/Compilation/Bison>
- Les cours et informations de Monsieur Laporte.