

# PR-Analyse-Syntaxique

---

Fait par Amal Abdallah et Lucie Souiou

## Sommaire

---

[Manuel d'utilisateur](#)

[Etat d'avancement du projet](#)

[Organisation du travail](#)

---

## Manuel d'Utilisateur

---

### Installation

Pour créer l'exécutable, il suffit de lancer la commande `make` dans le terminal en étant dans le répertoire du projet. Cela créera tous les fichiers requis pour compiler le programme et créera également l'exécutable `tpcas` dans le dossier `bin`.

### Lancement du programme

Pour exécuter le programme, il faut lancer l'exécutable dans le terminal à l'aide de la commande `./bin/tpcas`, suivi du chemin du fichier à analyser, puis du paramètre optionnel `-t/--tree` qui permet d'afficher l'arbre syntaxique avec les valeurs de tokens, ou `-l/--label` qui permet d'afficher l'arbre syntaxique avec les labels des tokens. Il est également possible de saisir le fichier dans l'entrée standard directement dans le terminal ou de la manière suivante :

```
./tpcas [OPTIONS] < /test/good/progClassique.tpc
```

Il est également possible d'obtenir de l'aide en utilisant l'option `-h/--help`. Attention, cet option ne fait pas exécuter l'analyseur syntaxique et envoie simplement les messages d'aides à l'emploi.

Afin d'automatiser l'analyse des tests, il suffit d'utiliser le script `ScriptTest.sh`, qui permet d'attribuer une note à l'analyseur syntaxique, et de vérifier tous les tests rapidement.

---

## Etat d'avancement du projet

---

Nous avons pu finir le projet dans son intégralité en répondant bien à toutes les consignes du sujet.

L'analyseur syntaxique fonctionne bien comme demandé, en indiquant si il y a erreur syntaxique (si oui, avec sa ligne et le caractère proche de l'erreur).

Nous avons aussi bien implementé l'affichage de l'arbre syntaxique avec deux options (valeurs ou label des

tokens).

Nous avons donc bien complété toutes les fonctionnalités.

---

## Organisation du travail

---

Nous avons tout d'abord commencé par la modification de la grammaire en s'appuyant sur les tests du bac à sable et ce qui était demandé dans le sujet.

Nous avons eu à plusieurs reprises quelques difficultés afin de compléter tous les tests sans savoir quelle était leur nature, en particulier avec les tests de tableaux.

Ensuite, Amal a géré la création et l'affichage des arbres, et Lucie les affichages d'erreurs syntaxiques.

Nous avons également rencontré des difficultés en générant l'arbre, qui provoquait une erreur de segmentation dont nous n'avons pas trouvé la source.

Amal a donc repris cette partie de zéro pour régler ce problème.

Enfin, nous avons créé un ensemble de tests valides et un ensemble de tests d'erreur syntaxique afin de pouvoir tester notre analyseur syntaxique plus facilement et régler certains problèmes qui n'étaient pas détectés auparavant.

Enfin, Lucie a implémenté un parseur d'argument à l'aide de `getopt`, et Amal un script permettant de lancer tous les tests automatiquement et donnant une note à l'analyseur syntaxique.