

# Rapport Livrable 2

## Introduction:

Le but de cette période était de corriger les problèmes du livrable 1 et d'implémenter le plus de choses possibles pour le livrable 2, à savoir les fonctions de parsing et la mise en place des objets permettant de garder en mémoire les éléments parsés.

## I- Répartition du travail

Nous avons d'abord poursuivi le travail du livrable 1. Cela comprenait le fait de renommer les fichiers avec des noms compréhensibles et proches de leur fonction. Siryne s'est donc occupée de renommer les noms des structures, des variables et des fichiers. Ensuite, il a fallu effectuer la restructuration de la fonction `re_read`. Siryne s'est donc occupée de la refaire. Siryne a également corrigé des erreurs et warnings attendant à ces fichiers. Il a fallu rédiger de nouveau les fonctions `re_match` et celles qui en dérivent qui avaient été faites par Jalal. Siryne les a donc revues et Yassine, chef de projet, a corrigé les erreurs non résolues par Siryne et Jalal. Yassine a poursuivi l'écriture du main et débogage du livrable 1.

Ensuite, le livrable 2 a été lancé. Siryne a, avec l'aide de Yassine et Jalal, rédigé les fonctions proposées par le sujet pour le parsing, mises dans le fichier `parsing_tools`. Siryne a rédigé les fonctions de parsing des non terminaux en recopiant les formes des non terminaux données dans le sujet, page 74.

Yassine a utilisé `gdb` et `valgrind` afin de corriger les erreurs de segmentation plus facilement ainsi que les erreurs d'algorithme qui font que l'ensemble ne fasse pas la fonction désirée. Une grosse partie du travail a été de chercher les erreurs et les corriger, on a constaté en fin de compte que la non réalisation de tests individuels pour chaque fonction ont une répercussion énorme sur l'efficacité du code global.

## II- Poursuite du travail

Faire les fichiers tests que l'on n'a pas eu le temps de faire pour les fonctions. Siryne doit ajouter les objets python aux fonctions de parsing. Siryne doit aussi modifier le code de `re_read` pour le cas de "`^`", car le sujet donnait une méthode plus concise que celle utilisée. Dans ce même fichier, utiliser des `switch case` à la place des multiples "`if`".

Jalal a entamé la création de la fonction qui permet de copier les chaînes de caractères dans l'ordre et les stocker dans tableau (champ `value`) et de taille fixe (champ `size`) pour créer l'objet `pyobj` de type `list`. On a pas mis ces fichiers parce qu'ils ne sont pas encore déboguer.