

# APS projet

Liuyi CHEN 21113643, Hongyu YAN 21113644

Mai 2022

## 1 Etat d'avancement

Nous avons réussi à implémenter les fonctionnalités d'APS jusqu'à APS2. L'APS2 a réussi à passer des dizaines tests de toutes les versions (APS0, APS1, APS1a et APS2 inclut). Nous avons fait l'évaluateur, lexer, parser et génération du terme prolog en OCaml. La partie typage est faite avec prolog.

- APS0 : 100%
- APS1 : 100%
- APS1a : 100%
- APS2 : 100%

Dans le rendu, nous avons mis deux version APS2 et APS1a. Étant donnée qu'ils ont une différence de *SET* pour la partie typage. Dans APS1a, nous ne pouvons que *SET* une variable définit par *VAR*. Alors que d'après la formulaire d'APS2, il n'y a pas de contrainte de type pour *SET*.

## 2 Commande d'instruction

Chaque version d'APS comporte trois modules :

- un module d'analyse syntaxique et de traduction en term prolog.
- un typeur (écrit en prolog) exécutable avec swipl.
- un module fournissant l'évaluateur.

Chacun de ces modules peut être testé avec les commandes suivantes :

- génération du terme prolog : `./prologTerm progXXX.aps`
- typage : `./typrog progXXX.aps`
- évaluation : `./exeprog progXXX.aps`