

APS

ISHIWATA Tomohiro, RETAIL Tanguy.

12 avril 2016

# Contents

## Version sans AST

Dans le dossier "*Version*", il suffit de lancer le makefile, puis d'exécuter *aps\_launcher* en précisant le nom du fichier à ouvrir en premier argument. Le résultat est affiché dans le flux de sortie standard.

Par exemple `./aps_launcher sample.aps >> res.pl` prend en entrée le programme *aps sample*, et fournit en sortie un fichier prolog *res*.

Pour tester le typeur / évaluateur:

lancer la commande `swipl`

puis compiler le typeur / évaluateur

`[typeur].`

`is_well_typed(res.pl).`

## Version AST (non fonctionnelle)

Dans le dossier "*VersionAST*" se trouvent les fichiers composant l'AST, les analyseurs lexicaux et syntaxiques. L'AST est écrit en C++ et nous avons un problème pour générer le prolog. En effet, un *ASTBlock* contient un champ de classe "`list < ASTCmds > cmds`" auquel nous ajoutons des objets héritant de *ASTStat* et *ASTDec* (eux même héritant de *ASTCmds*). Le problème est que lors de l'ajout dans une liste (Plus généralement un Container), le C++ ajoute par copie et nous perdons l'information sur le type réel de l'objet que nous ajoutons ; et sommes incapables d'appeler les méthodes de cet objet, mais seulement celles de *ASTCmds*.

N'étant pas experts en C++, nous n'avons pas trouvé le temps de régler ce problème, qui nous empêche de travailler avec des AST pour le moment.

Pour essayer, il est néanmoins possible de compiler avec le makefile, et de lancer avec le fichier *aps\_launcher* en précisant le nom du fichier à ouvrir en premier argument.

## Extension

Nous disposons d'une extension qui permet d'afficher les erreurs lexicales et syntaxiques dans certains cas, en spécifiant le numéro de ligne et le caractère.