

SORBONNE UNIVERSITÉ

RAPPORT FINAL

16 Décembre 2022

Projet de APS

Auteurs :

Shuhan DUAN

Yukai LUO

Encadrants :

Romain DEMANGEON

Loïc SYLVESTRE

Hector SUZANNE



Table des matières

1	Présentation	2
1.1	Présentation	2
1.2	Language Choisi	2
1.3	Test	2
2	Progression	4
3	Conclusion	5

Chapitre 1

Présentation

1.1 Présentation

Dans ce projet, on a développé un nouveau langage de programmation, nommé "APS". Contrairement à beaucoup des langages de programmation, le compilateur aps se compose de trois parties : la partie syntaxique, la partie typage et la partie sémantique. La partie syntaxique se compose de lexer, parser et AST ; la partie typage se compose d'un typeur, et nous avons écrit une imprimante qui montre les nœuds d'AST du programme pour passer au typeur ; la partie sémantique se compose d'un évaluateur, qui traduit ocaml, le langage que nous utilisons pour écrire APS, en APS.

1.2 Language Choisi

Nous avons choisi d'utiliser OCaml pour développer APS, car OCaml est un langage à typage statique qui peut détecter de nombreuses erreurs de programmation courantes lors de la compilation, ce qui améliore la fiabilité et la robustesse du code. De plus, OCaml est un langage de programmation fonctionnel qui prend en charge des fonctionnalités telles que les fonctions d'ordre supérieur, les fermetures et la récursivité, ce qui facilite l'écriture de programmes hautement abstraits.

Nous avons choisi d'utiliser Prolog pour développer notre typeur, car Prolog offre un mécanisme de correspondance de règles qui facilite la mise en correspondance de modèles et de types, ce qui est très utile pour développer un typeur. De plus, la syntaxe de Prolog est très flexible, ce qui permet une extension facile à des systèmes de types plus complexes. De plus, le code Prolog est généralement plus concis que celui des autres langages de programmation car ses règles et requêtes peuvent être exprimées de manière compacte.

1.3 Test

On a mis notre projet APS sur le gitlab de STL ([lien](#)).

Commande pour entrer la répertoire : **"cd <repertoire>"** (aps0,aps1,aps2...)

Commande pour vérifier le programme : **"dune exec main.exe test/<programme>"** (pro0.aps,prog1.aps etc..)

Après taper cette commande, le résultat de évaluateur et printeur va être affiché sur terminal. Veuillez copier le résultat de priter, et ouvrir prolog avec la commande : **“swipl -p typeur.pl”**, puis le coller pour vérifier le résultat.

Chapitre 2

Progression

On a bien fini la partie APS0, APS1 et APS1a. Cependant, pour APS2, nous avons rencontré quelques problèmes, par exemple, l'id de SET n'était pas reconnu dans le PROC.ET il connais pas quelque mots-clefs (alloc ,nth)

Chapitre 3

Conclusion

Pendant ce projet, nous avons rencontré des difficultés. Nous n'étions pas familiers avec prolog, nous avons donc passé beaucoup de temps sur le typeur. Et plus, pendant le développement d'APS0, comme il s'agit de notre premier contact avec un projet similaire, notre expérience est relativement faible, et nous avons fait beaucoup d'erreurs. Par exemple, dans le lexer, nous avons écrit `ident` devant de nombreux tokens, ce qui a entraîné la reconnaissance de ces tokens comme `ident`, ce qui nous a coûté beaucoup de temps pour corriger cette erreur.

Mais après avoir terminé ce projet, nous avons une compréhension plus complète de la développement des langages de programmation. Cela nous sera d'une grande aide pour notre future étude. En même temps, les trois professeurs, Romain, Loïc et Hector, nous ont été d'une grande aide pour mener à bien ce projet, ils ont toujours répondu patiemment à nos questions, même si certaines d'entre elles étaient très stupides. Sans l'aide des professeurs, nous n'aurions pas pu mener à bien ce projet. Merci beaucoup!