Travail pratique 2 - Concept des langages

Annie-Pier Coulombe 1067670
Jad Yammine 1067212

December 4, 2017

Voici le rapport du Travail pratique numéro 2 dans le cadre du cours IFT-2035

1 Introduction

Pour commencer, nous avons dû, comme le point numéro 1.1 nous le suggèrait, apprendre d'avantage sur Prolog. Cet apprentissage s'est continué tout au long du travail, lorsque de nouveaux obstacles se présentaient. Celà pris un moment pour bien comprendre comment ce que nous devions faire et comment le faire. Après avoir lu la donnée et essayé de la lié avec le code fournis, nous avons pu commencer a coder.

2 L'initialisation de l'environnement

2.1 Les règles

Premièrement, nous avons dû comprendre que l'environnement n'était pas déjà prêt a être utilisé. Nous avons donc localisé ce qu'il manquait et comment nous devions l'implanter en cherchant dans le code fournis les endroits critiques

2.2 Le codage

Une fois que nous avons trouvé que la fonction infer s'occupait des règles du langage, nous avons pu commencer a l'écrire en Prolog. Il était facile de commencer et écrire les règles pour le int et le type mais les suivantes nous ont pris plus de temps et de discussion entre collègues de classe. Cependant avec beaucoup d'effort, nous sommes parvenus a compléter le tout sans trop d'erreures.

2.3 L'initialisation

Lorsque l'environnement a été initialisé, nous devions passer à la prochaine étape, qui était d'essayer de "vérifier" les équations du sample.

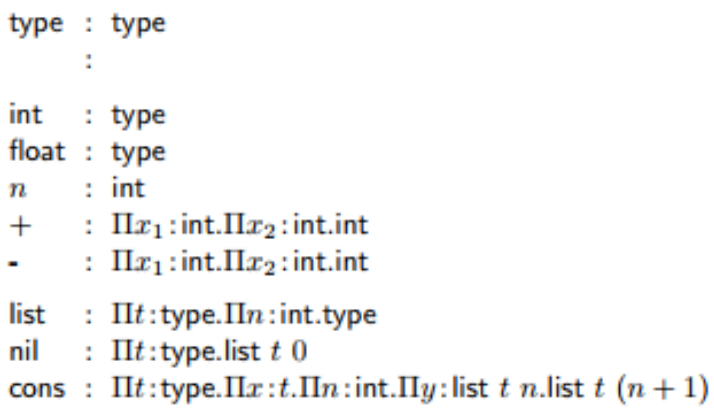

Figure 1: Les règles nécessaire afin d'initialiser l'environnement.

3 Vérification des tests

3.1 Compréhension du sample

Initialement, nous avons essayé de comprendre la fonction sample afin de cibler les fonctions nécessaires à la résolution des expression algébrique.Pour celà nous avons ajouté des "write" a quelques endroits dans le code afin de voir quelles fonctions étaient utilisées et qu'est-ce qu'elle recevait comme paramètre. Nous avons remarqué que nous avions besoin, pour chaque itération du programme, d'initialiser l'environnement, par conséquent, nous avons ajouté à la fin du code une fonction "calls" qui initialise automatiquement l'environnement pour nous. Exemple: "calls(1+2)."

3.2 Expend

Afin de pouvoir commencer à résoudre une équation, nous devions utiliser la fonction expend afin d'éliminer le sucre syntaxique des expressions.

3.3 + - * vs /

Pour les expressions +, -, ou *, nous avons utilisé le expend pour construire le app, éliminant le sucre syntaxique pour ensuite la passer à une fonction infer qui gère chacune d'entres elles. Cependant pour la division nous avons dû construire un infer pour pouvoir la traiter indépendament, puisque nous devions lui suggérer le type float plutôt que int afin que la division puisse nous fournir une donnée avec des décimales.

3.4 Cons

Malgré un manque de temps, de connaissances et ressources, nous n'avons pas réussi a résoudre le cons. Nous étions sur la bonne voie mais un manque de temps dû au travail et d'autres travaux nous a causé un contre-temps.