

Rapport - Travail pratique #1

IFT-2035

Zouhair El Yani

18 octobre 2024

1 Problèmes rencontrés

Tout d'abord, le principal défi que j'ai rencontré concernait l'énoncé, qui était non seulement très long, mais exigeait également une grande attention pour être bien compris et assimilé. Personnellement, j'ai commencé le travail pratique dès le jour de sa publication et j'ai passé toute la journée du dimanche à lire et relire le sujet. Cela a été particulièrement difficile, car la majorité des concepts étaient nouveaux pour moi, notamment la syntaxe de Lisp, qui m'a pris beaucoup de temps à comprendre, surtout pour bien différencier les expressions des déclarations. J'avais également des difficultés à savoir dans quelle mesure nous étions censés comprendre le code déjà fourni, car il utilisait de nombreuses notions que nous n'avions pas encore abordées en cours.

Lors de l'écriture du code manquant, j'ai rencontré plusieurs problèmes qui m'ont coûté beaucoup de temps, notamment la possibilité de tester le code que j'avais implémenté. Je ne savais même pas comment utiliser ghci pour exécuter mes propres tests, mais lors de la disponibilité du 3 octobre, j'ai pu résoudre ce problème et clarifier plusieurs ambiguïtés avec l'aide du démonstrateur.

Une fois cette capacité de test débloquée, j'ai pu utiliser "readSexp" pour tester différentes expressions et observer la structure exacte d'une S-expression, ce qui m'a énormément aidé. Après avoir implémenté la majorité des cas pour s2l, j'ai rencontré beaucoup de difficultés avec "fix". En observant les différentes S-expressions, j'ai remarqué qu'il y avait de nombreux cas à gérer pour distinguer correctement les déclarations de variables des déclarations de fonctions et ainsi éliminer le sucre syntaxique. Cela m'a posé beaucoup de difficultés, et je dirais que c'était la partie la plus compliquée. J'étais tellement fatigué par l'analyse des S-expressions que j'avais du mal à regarder l'écran. Après avoir surmonté cette étape grâce à la fonction auxiliaire snodeSep, je me suis rendu compte qu'il me manquait un cas pour les définitions mutuellement récursives dans fix, ce qui m'a conduit à réimplémenter ma fonction.

Après plusieurs tentatives, j'ai réussi à gérer les définitions mutuellement récursives, mais cela causait des problèmes pour les autres cas. Je me suis donc retrouvé avec deux implémentations complémentaires que j'ai dû fusionner pour obtenir une fonction plus générale capable de traiter tous les cas.

2 Surprises

Durant ce travail pratique, j'ai rencontré plusieurs surprises, tant positives que négatives. Pour commencer, j'ai été étonné de constater à quel point ma compréhension de l'énoncé évoluait au fur et à mesure que j'avancais dans le travail : ce qui me semblait obscur au début devenait progressivement plus clair. Travailler "en largeur" sur les deux fonctions m'a aidé à mieux appréhender la vision d'ensemble, et j'ai pu terminer l'implémentation en 20h.

Une autre surprise a été de découvrir que la fonction `s2l` qu'on devait implémenter, devait non seulement traduire les `Sexp` en `Lexp`, mais également éliminer le sucre syntaxique des déclarations. J'étais initialement confus quant à la manière d'aborder cette tâche, ce qui m'a causé un certain stress.

Cependant, la plus grande surprise a été de réaliser que les définitions dans `fix` pouvaient être mutuellement récursives. Je ne m'y attendais pas du tout et, au premier abord, cela m'a semblé être une tâche presque impossible à accomplir.

3 Choix que j'ai dû faire

Dans les jours qui ont suivi la publication du TP, mon coéquipier a dû abandonner le cours pour des raisons personnelles. J'ai alors été confronté à un choix : soit tenter de réaliser le projet seul, soit chercher un nouveau coéquipier. Ce qui m'a poussé à le faire seul, c'est le fait que ce travail nécessite une compréhension assez approfondie, mais n'exige pas une quantité excessive de code. J'ai donc décidé de poursuivre seul, afin de me donner l'opportunité de surmonter tous les obstacles que ce travail présente.

4 Options que j'ai sciemment rejetées

J'ai rejeté une seule option : celle de modifier le code source. Premièrement, étant donné que je n'étais pas très familier avec le code, il y avait un risque élevé de le casser en effectuant des modifications. En effet, le code source contient de nombreux éléments interdépendants, ce qui augmente considérablement la probabilité d'introduire des erreurs. Deuxièmement, je savais qu'en procédant ainsi, ma solution s'écarterait de celle attendue, ce qui entraînerait davantage de complications : demander de l'aide serait devenu plus difficile, car il aurait été nécessaire de détailler les modifications apportées. De plus, il y aurait eu un risque que ma solution soit moins efficace que celle prévue.