ENSIMAG

PROJET GL

GL 59

Bilan de gestion d'équipe et de projet

WALID BAKIR

WALID ABIDI

ABDELOUAHAB MESTASSI

EL MEHDI SALAH-EDDINE

YASSIN EL KHALDI AHANNACH

1 Organisation adoptée

Pour ce projet, on avait quatre parties à faire et à répartir entre les cinq membres de l'équipe. La partie A du compilateur (lexeur et parseur), la partie B de vérifications contextuelles, la partie C de génération de code et la partie extension qui est dans notre cas l'extension Trigo. Nous avons décidé de répartir le travail de ces parties en prenant en compte les compétences et les points forts de chaque membre de l'équipe. Ainsi, un étudiant était chargé d'écrire le lexeur, deux étudiants ont écrit le parseur et ont implémenté l'extension, un étudiant a implémenté la partie B et deux autres étudiants ont écrit la partie de génération de code.

Au début du projet nous avons commencé par évaluer la quantité de travail nécessaire pour les différentes étapes. On a ensuite mis en place un planning qui doit être respecté et qu'on a essayé de mettre à jour chaque semaine en fonction de notre avancement et des problèmes rencontrés, ce qui a été très utile et nous a permis de respecter toujours les dates limites des rendus .

L'organisation est l'élément clé pour réussir un projet, et afin d'optimiser notre travail le mieux possible, chaque membre d'équipe doit assurer la complétion de sa partie à la fin de chaque délai posé, tout en s'assurant de la qualité de sa tâche (en faisant des tests extensives, d'écrire du code propre et lisible.).

Dans l'ensemble, cette répartition de travail nous a permis d'implémenter les différentes parties du projet et les finir avant la date limite. Il restait cependant des points non ou mal implémentés dans notre compilateur dans l'étape de génération de code.

2 Historique du projet

2.1 Étape A

2.1.1 Lexeur : Abdelouahab Mestassi

Cette partie consistait à traduire la lexicographie de Deca en utilisant l'outil de génération d'analyseur lexical ANTLR4. Cette partie a été correctement implémenté les trois premiers jours du projet après la prise en main et la compéhension de l'outil ANTLR4.

2.1.2 Parseur : Walid Bakir - Walid Abidi

Au début, l'ensemble du projet était un peu flou. Nous ne savions pas exactement ce que nous devions faire, y compris cette partie (Syntaxe). Le code a été écrit en Antlr4, qui à ce moment-là était quelque chose de complètement nouveau pour nous tous. Nous ne savions pas ce que c'était et à quoi cela servait et, surtout, nous n'avions aucune expérience de l'écriture de code en Antlr. Nous avons pris un temps considérable pour essayer de comprendre la tâche dans son ensemble. Petit à petit, nous avons commencé à coder et à tester notre code à chaque étape du processus. Après la réalisation de l'étape B, nous avons eu quelques erreurs dans le parseur et nous avons travaillé à les corriger en même temps. La deuxième partie (Objet) a été beaucoup plus facile que la première car nous avons alors compris exactement ce que nous devions faire et comment le faire.

2.2 Étape B : Abdelouahab Mestassi

L'étape B est l'étape de vérifications contextuelles. On vérifie la syntaxe contextuelle du programme et en cas d'erreur, on génère un message indiquant l'origine de l'erreur. On peut diviser son implantation en deux parties. La première partie qu'il fallait faire rapidement est la compréhension de cette étape, des règles de grammaire du poly et les différentes erreurs qu'il faut générer. J'ai donc lu plusieurs fois la partie syntaxe contextuelle du poly et j'ai écrit un fichier rassemblant l'ensemble des erreurs possibles, avec pour chaque erreur, la règle correspondante et le message d'erreur qu'il faut afficher. Ce premier travail a été crucial car j'ai ensuite commencé la deuxième partie de l'implantation : le développement du code gérant les vérifications et c'était plus facile à faire car je savais alors exactement ce qu'il fallait faire. J'ai réussi a implémenté correctement l'étape B, en corrigeant les différents bugs et erreurs que j'ai pu relevé après le développement de l'étape C.

2.3 Étape C : Yassin El Khaldi Ahannach - Salah-Eddine El Mehdi

La partie la plus exigeante du code était l'étape de génération du code assembleur. Nous avons commencé cette partie depuis le premier suivi en essayant de comprendre comment utiliser de manière efficace le résultat des parties A et B qui est l'arbre abstrait décoré. Ensuite, on était amené à comprendre comment fonctionne le code assembleur et la gestion des registres en se basant sur le poly du projet et les vidéos disponibles. Cependant, le début effectif du codage de cette partie a commencé juste avant le deuxième suivi avec un retard et une moyenne assimilation des notions, et la partie génération de code du rendu intermédiaire n'était pas assez fonctionnelle .Pour cela on a essayé de rattraper et combler les lacunes de notre partie en essayant d'approfondir nos connaissances sur le langage assembleur, étant donné la complexité de la partie avec objet. Parmi les ressources clés de notre étude, on retrouve l'exemple du code généré dans le poly (partie IV) qui a joué un rôle primordial dans la compréhension. Les sections réalisées de cette partie ont été optimisées, testées et sont satisfaisantes. Néanmoins, quelques fonctionnalités du langage objet n'ont pas pu être complétées à temps. Une meilleure gestion du personnel aurait peut-être pu nous ai-

der à atteindre nos objectifs fixés à l'entame du projet .

Une grande partie de notre travail était consacrée pour la validation du compilateur et s'assurer que la génération du code était correcte, ce qui se faisait de manière progressive en construisant des bases de tests à mesure que de nouvelles fonctionnalités étaient ajoutées .

2.4 Extension : Walid Bakir - Walid Abidi

Le travail sur l'extension a commencé par une série de recherches, les idées d'implantation qui revenaient souvent était les algorithmes basés sur des approximations polynomiales ou l'algorithme Cordic. Nous avions choisi une implémentation utilisant les polynômes de Chebyshev et les développements en série entière. Nous avions eu recours à quelques ouvrages.

Au début, l'objectif n'était pas totalement clair, nous avons rencontré des difficultés concernant la compréhension des calculs sur les flottants et la notion d'ulp que nous avions pu surmonter par la suite grâce à un travail quotidien organisé.

Pour la réduction d'intervalle, nous avons utilisé la méthode de Cody Waite, la méthode de Payne Hanek était l'une des choix possibles qui offrait plus de précision mais que nous avions choisi de ne pas implémenter au final pour respecter les contraintes horaires que nous avions mis en place .

Dans l'ensemble, nous avons essayé de fournir un travail cohérent et assez précis, le travail nécessitait de solides bases mathématiques et de bonnes connaissances sur les flottants.