Rapport Compilation

Herlin Loic, Yanovkyy Alexander, Brelot Julien, Metzger Benjamin

Janvier 2024

1 Introduction

Notre groupe se compose de Herlin Loic, Yanovkyy Alexander, Brelot Julien et Metzger Benjamin. Nous sommes quatre étudiants de TPS en spécialité RIO. Nous avons réussi à implémenter une grande partie d'un compilateur en C. Malheureusement par manque de temps, nous avons uniquement réalisé la partie MIPS pour les matrices.

2 Les capacités du compilateur

2.1 Arithmétique

Le compilateur peut traiter des opérations arithmétiques primaires (+,-,*et/), les enchainer et traite l'erreur de division par 0.

2.2 Boucle (for, while)

Les boucles while et for sont correctement implémenté, il est donc possible de les imbriqués.

2.3 Fonction

Il est possible de créer des fonctions autres que la fonction main, et de les appeler dans le main ou d'autre fonctions. Un rapport d'erreur a été ajouté (voir ci-dessus)

2.4 Conditions

Il est capable de compiler les conditions if et else ainsi que les conditions imbriquées. Mais il n'est pas possible d'avoir un if non accompagné d'un else.

2.5 Variables

La déclaration, l'assignation d'une valeur (entier ou réelle), expression ou fonction est supportée. Il est capable de gérer la portée des variables (voir ci-dessus) mais aussi la conversion int à float et de float à int.

2.6 Matrices

Par manque de temps nous n'avons pas pu implémenter les matrices côté frontend (yacc) mais côté backend (mips). Nous avons pu écrire le code assembleur nécessaire via des macros pour gérer toutes les opérations demandées, un fichier assembleur de tests a été apporté pour vérifier cela présent dans `src/matrix/matrix-test.asm`.

3 Points intéressants

3.1 Portée des variables

Lorsqu'une variable est déclarée, elle est initiée dans une portée, ce qui permet pouvoir suivre s'il y a une assignation ou utilisation d'une variable qui est hors de portée ou non.

```
1  int main()
2  {
3      int a;
4      j = 0;
5      for (int i = 0; i < 10; i++)
6      {
7          a = 1;
8          for (int j = 0; j < 10; j++)
9          {
10             a = 1;
11         }
12         j = 0;
13     }
14     i = 0;
15     j = 0;
16     return 0;
17 }
```

Figure 1: Code provenant de `tests/variables/errors/out_of_scope1.c`

Dans l'exemple ci-dessus, la variable j a été initialisé en ligne 8, mais en ligne 4,12,15 il y a une tentative d'assignation. Même chose pour i. Alors des messages d'erreurs apparaissent lors de la compilation de ce code comme montrer ci-dessous (l'exemple peu ce généraliser entre plusieurs fonctions).

```
1  ERROR: Variable not declared in line 4
2      j = 0;
3      ^
4  ERROR: Variable not declared in line 12
5          j = 0;
6          ^
7  ERROR: Variable not declared in line 14
8      i = 0;
9      ^
10 ERROR: Variable not declared in line 15
11     j = 0;
12     ^
13 Compilation failed with 4 errors
```

Figure 2: Messages d'erreur en compilant `tests/variables/errors/out_of_scope1.c`

3.2 Les rapports d'erreurs

Les erreurs, arithmétique ou de cohérence, sont traitées dans le yacc. Il peut détecter plusieurs erreurs si c'est le cas. Nous avons essayé de nous rapprocher le plus possible de ce que font les compilateurs actuels. Nous traitons les erreurs suivantes :

- Erreur de syntaxe
- Main non déclarée
- Fonction déclarée plus d'une fois
- Assignation d'une valeur à une constante
- Passage d'argument de type invalide à une fonction
- Division par 0
- Appel à une fonction non déclarée
- Appel à une fonction avec pas assez d'arguments
- Appel à une fonction qui ne prend aucun argument
- Appel à une fonction avec trop d'arguments
- Appel à une fonction avec des arguments de type invalide
- Variable utilisée non déclarée dans la portée courante
- Fonction utilisée non déclarée

Voici quelques exemples de message d'erreurs générées lors de la compilation d'un programme.

```
1 ERROR: syntax error, unexpected ID, expecting STR in line 3
2     printf(aazzaza);
3             ^~~~~~~
4 Compilation failed with 1 errors
```

Figure 3: Messages d'erreur lié à la syntax

```
1 ERROR: Division by zero in line 5
2     int c = a + b / 0;
3                     ^
4 Compilation failed with 1 errors
```

Figure 4: Messages d'erreur en compilant `tests/arithmetic/errors/div_0.c`

```
1 ERROR: Function already declared in line 2
2 int main() { return 0; }
3     ^~~~
4 Compilation failed with 1 errors
```

Figure 5: Messages d'erreur en compilant `tests/functions/errors/double_main.c`