

Rapport de Projet : ARC - Algo/RAM Compiler

Joseph REGNAULT

January 2, 2025

Contents

1	Introduction	2
2	Structure et Fonctionnement du Programme	2
2.1	Architecture Générale	2
2.2	Fonctionnement	2
3	Structure de la Mémoire	2
3.1	Modèle RAM	2
3.2	Organisation Mémoire	3
4	Caractéristiques du Compilateur et du Langage Reconnu	3
4.1	Langage Reconnu	3
4.2	Limitations	3
5	Choix de Conception	3
5.1	Choix de la Grammaire	3
5.2	Modularité du Programme	4
6	Limitations et Améliorations Futures	4
6.1	Limitations Identifiées	4
7	Informations supplémentaires	4
8	Conclusion	4

1 Introduction

Ce rapport présente le travail effectué sur le projet ARC (Algo RAM Compiler). Il abordera les points importants de la réalisation du projet, des informations sur les choix effectués pendant l'écriture du compilateur, ainsi que les limitations du logiciel.

2 Structure et Fonctionnement du Programme

2.1 Architecture Générale

Le compilateur fonctionne en plusieurs étapes :

- **Analyse Lexicale** : Transformation du code source en une suite de tokens.
- **Analyse Syntaxique** : Construction d'un arbre syntaxique abstrait (AST) à partir des tokens et de la grammaire.
- **Analyse Sémantique** : Vérification des types et du respect des règles du langage.
- **Génération de Code** : Production de code RAM à partir de l'AST.

2.2 Fonctionnement

L'analyse lexicale est gérée par flex, l'analyse syntaxique par Bison, et les deux autres étapes sont implémentées en C.

3 Structure de la Mémoire

3.1 Modèle RAM

La machine RAM n'étant composée que de l'accumulateur, d'une mémoire programme, et d'une mémoire de travail linéaire ne pouvant stocker que des entiers, il est nécessaire de s'organiser afin d'implémenter des structures plus complexes.

3.2 Organisation Mémoire

Notre compilateur utilisera une organisation mémoire simple composée de l'accumulateur, d'une zone de registres temporaires nécessaires pour certaines opérations, d'une zone mémoire dédiée au stockage des variables, et d'une pile.

4 Caractéristiques du Compilateur et du Langage Reconnu

4.1 Langage Reconnu

Le compilateur reconnaît les éléments suivants du langage Algo :

- Affectation de variables (entières).
- Structures de contrôle (`if`, `while`, `for`).
- Fonctions (déclaration uniquement).
- Opérations arithmétiques.
- Opérations logiques.

4.2 Limitations

Les limitations actuelles incluent :

- Pas de gestion des pointeurs/tableaux.
- Pas de différence de contexte pour les variables locales/globales
- Pas de génération de code pour les fonctions autres que MAIN.

5 Choix de Conception

5.1 Choix de la Grammaire

La grammaire utilisée est une grammaire simple décrivant le langage algorithmique. Elle ne permet pas de déclarer de variables globales, donc toutes les variables sont globales par défaut.

5.2 Modularité du Programme

Chaque étape du compilateur est encapsulée dans un module distinct.

- Le module d'analyse syntaxique utilise Bison et C pour générer les AST correspondants au code.
- Le module traitant la table de symboles utilise une structure de données appelée `tabsymb` afin de gérer les variables et leurs adresses.
- Le module de génération de code est organisé en une série de fonctions traitant les différents types de nœuds de l'AST.

6 Limitations et Améliorations Futures

6.1 Limitations Identifiées

- La gestion des fonctions est partielle : bien que la grammaire les reconnaisse et que leurs variables soient bien répertoriées dans la table des symboles, le compilateur ne génère pas de code pour leur exécution.
- Les structures de données d'adressage comme les pointeurs ou les tableaux ne sont pas implémentées.

7 Informations supplémentaires

Plusieurs fichiers de test se trouvent dans le répertoire `test/` présentant les fonctionnalités du compilateur. Chaque élément est introduit dans un fichier éponyme, et d'autres fichiers contiennent divers tests non spécifiques.

8 Conclusion

Ce projet a permis de développer un compilateur semi-fonctionnel pour une version simplifiée du langage Algo. Bien qu'il soit limité, le programme produit du code fonctionnel pour la machine RAM.