

Rapport de projet Sémantique et Traduction des langages

Damien Hostettler, Simon Maurel, Qi Chen et Vicky Dincher

17 Juin 2016

Table des matières

1	Construction de la table des symboles	2
1.1	Contenu et hiérarchie	2
1.2	Gestion des variables globales	2
1.3	Ajouts pour le $\mu C\#$	2
2	Préconditions et gestions des erreurs de type	2
2.1	Opérateurs et compatibilité de types	2
2.2	Types particuliers	2
	Gestion des structs	2
	Gestion des pointeurs	2
3	Fonctions et leurs surcharges	2
4	Génération de code	2

Introduction

Le but de ce projet a été de réaliser un compilateur pour les langages μC et $\mu C\#$. Ce compilateur doit vérifier les erreurs détectables lors de la compilation (erreurs de types, variable non définies ...) et doit générer la traduction du programme compilé en langage **TAM**.

La réalisation de ce compilateur passe par la gestion de la table des symboles, des erreurs de type, ainsi que la génération de code

1 Construction de la table des symboles

La table des symboles doit contenir toutes les informations sur ce qui est déclaré dans le programme (variables, types, fonction) sauf leur valeur en temps réel.

Une table des symboles est une liste d'élément de type *INFO* que l'on peut repérer par leur nom (nom de variable par exemple).

1.1 Contenu et hiérarchie

Nous avons donc modélisé notre table comme un *HashMap* $\langle \text{String}, \text{INFO} \rangle$. Ce sont les différents couples (Nom des variables (fonctions ...), informations liées).

On trouve ainsi plusieurs type d'informations (toutes héritées de la classe *INFO*) :

- Les *INFOVAR* liées aux variables. Elles contiennent simplement le type de la variable, et son emplacement dans la pile.
- Les *INFOTYPE* liées aux types créés avec *typedef*. Elles contiennent un type (celui créé).
- Les *INFOFONC*, liées aux fonctions. Elles contiennent le type de retour de la fonction, la liste des différentes possibilités de paramètres pouvant être utilisés avec cette fonction (surcharges), ainsi qu'une TDS fille de la TDS courante, contenant les informations sur les variables (ou types) locales à la fonction.

On crée donc une TDS fille à chaque nouvelle fonction, mais également lorsque l'on rentre dans un nouveau bloc. On obtient ainsi la hiérarchie suivante :

1.2 Gestion des variables globales

1.3 Ajouts pour le $\mu C\#$

2 Préconditions et gestions des erreurs de type

2.1 Opérateurs et compatibilité de types

2.2 Types particuliers

Gestion des structs

Gestion des pointeurs

3 Fonctions et leurs surcharges

4 Génération de code