

Documentation utilisateur

Grenoble-INP ENSIMAG

Groupe 19

Salaheddine ASLOUNE

Wen LUO

Anass BOUBAKRI

El Mehdi KHADFY

Amine BELFKIRA

Première partie : Description du compilateur du point de vue de l'utilisateur et mode d'emploi

Deuxième partie : Commandes et options

Troisième partie : Messages d'erreurs

Quatrième partie : Limitations

Introduction

Cette documentation constitue un guide utilisateur pour le compilateur Deca. Elle met en évidence les différentes options de compilation, les limites techniques du compilateur et aussi la liste des messages d'erreur et leurs significations.

Ce document présente aussi le mode d'utilisation de l'extension, c'est-à-dire le processus de génération des fichiers .class .

1. Description du compilateur du point de vue de l'utilisateur et mode d'emploi

Notre compilateur deca est un compilateur deca complet. Il peut être utilisé par le programme principal `decac`. La syntaxe d'utilisation de l'exécutable decac est la suivante:

```
decac [[-p | -v] [-n] [-r X] [-d]* [-P] [-w][-j] <fichier deca>...] | [-b]
```

Une fois la compilation effectuée, on peut faire appel au compilateur deca par la commande précédente. On désignera le fichier d'entrée à partir de la racine du projet, de la forme <chemin/nom_du_fichier.deca>. Notez bien que le suffixe .deca est obligatoire. Sauf erreur d'utilisation, le fichier assembleur résultant doit être dans le même répertoire que le fichier source. Pour plus de détails sur la commande decac veuillez vous référer à la section 2. Commande et options.

Une fois le fichier assembleur généré, il suffit de faire appel à la commande ima afin de l'exécuter.

2. Commandes et Options

Pour afficher toutes les commandes possibles ,vous pouvez taper la commande decac sans arguments .

p	decac s'arrête à l'étape de la construction de l'arbre. La sortie est un programme deca syntaxiquement correct.
v	decac s'arrête à l'étape de vérification.
n	Supprime les tests à l'exécution spécifiés dans les points 11.1 et 11.3 de la sémantique de Deca du cahier de charge.
r X	Limite les registres banalisés disponibles à X registres. ($4 \leq X \leq 16$)
d	Active les traces de debug.
P	Lance la compilation des fichiers sources en parallèle, s'il y a plusieurs fichiers sources.
b	Affiche une bannière indiquant le nom de l'équipe. Cette option ne peut être utilisée que de la manière suivante: decac -b.
j	génère le fichier class pour la JVM

Les options '-p' et '-v' sont incompatibles. L'options '-b' comme spécifiée précédemment ne peut être utilisée que seule et sans autres arguments. Si cette option est utilisée avec d'autres arguments une erreur indiquant que la commande est incorrect sera levée .

3. Messages d'erreurs:

3.1 Les erreurs de syntaxe

3.1.1 Token recognition error

Erreur levée lorsque la syntaxe du code est incorrect, cela permet en particulier de vérifier l'orthographe des mots-clés ,le bon parenthésage et les guillemets.

Exemple :

```
includemissing.deca:12:0: token recognition error at: '#include c'
```

3.1.2 Value Error

Cette erreur est levée lorsqu'il y a utilisation d'un entier ou un float de taille dépassant les limitations imposées par les méthodes parseInt() et parseFloat() des classes Integer et Float.

Exemple:

```
Floatlong.deca:2:12: Value is too large
```

3.1.3 Invalide Lvalue Error

Erreur levée quand le côté gauche d'une 'affectation n'est pas une expression affectable (paramètre, variable ou attribut)

3.1.4 Include File Not Found

Le fichier spécifié est introuvable. Le nom du fichier ou son chemin est incorrect.

3.1.5 Circular Include Error

Cette erreur est levée lorsqu'il y a une inclusion circulaire.

3.2 Les erreurs de context

3.2.1 Double Définition

Cette erreur est levée lorsqu'un Symbol est définie dans le même scope plusieurs fois

3.2.2 Type Not Match

Le type de données requis est incohérent avec le type de données fourni .

3.2.3 Operation Error

Les types de données des deux côtés de l'opérateur sont incompatibles

3.2.4 Cast Error

Le type de données existant ne peut pas être converti au type de données spécifié.

Exemple :

```
int x = 1;
```

```
type_class(_) a = (type_class(_)) (x);
```

3.2.5 Class Not Defined

La classe n'est pas définie ou n'est pas importée.

3.2.6 Incorrect Value

La valeur requise ne correspond pas à la valeur spécifiée.

3.2.7 Method Call Failed

Lors de l'appel d'une méthode, le type de paramètre ou le nombre de paramètres n'est pas conforme aux exigences.

3.2.8 Error Arithmetic Operation

Les deux côtés de l'opérateur doivent être entiers ou flottants.

3.3 Les erreurs de codegen

3.3.1. Arithmetic Overflow

Cette erreur regroupe toutes les erreurs de débordement lors d'une opération arithmétique. Dans ce cas l'erreur levée est:

Error: Overflow during arithmetic operation

Exemple :

```
int x = 3 / 0
```

3.3.2. StackOverflow

Lorsqu'il y a un débordement de pile, l'erreur suivante est levée :

Error: Stack Overflow

Exemple :

On rencontre fréquemment cette erreur lors d'appels récursifs trop profonds.

3.3.3. Heap Overflow

Lorsqu'il y a un débordement de tas, l'erreur suivante est levée:

Error: Heap Overflow

3.3.4. NULL Dereference

Cette erreur est levée lors de l'utilisation d'un objet qui référence à null. L'erreur correspondante est :

Erreur: dereferencement de null

3.3.5. Error Dynamic_Cast

Cette erreur est levée lorsque le type statique demandé à l'opérande n'est pas une classe père de son type dynamique

Erreur : Cast can't be done

3.3.6. Return Error

Cette méthode est levée lorsqu'une méthode ne renvoie pas un type correspondant à son type de retour .

L'erreur correspondante est :

Erreur : sortie de la methode <class>.<method> sans return

4. Limitations

Voici une liste des limitations de notre compilateur deca :

- ★ Les nombres entiers doivent être compris entre -2^{31} et $2^{31} - 1$. cette limitation est imposé par la fonction `parseInt()`;
- ★ Les nombres flottants doivent être compris entre 2^{-149} et $(2-2^{-23}) \cdot 2^{127}$. De même que pour les nombres entiers, les nombres flottants sont reconnues par la méthode `parseFloat()` de la classe `Float` qui limite les flottants pris en considérations à cet intervalle.
- ★ Le compilateur a un pile limité : A partir d'une dizaine d'appels, le compilateur indique un dépassement de pile.
- ★ La transformation du code en Bytecode (fichier `.class`) est limitée aux codes dérivables depuis la grammaire Deca Sans Objet ,Ainsi le bloc de la méthode `main` est englobé dans une fonction `main` d'un fichier `.class` portant le même nom du fichier source .