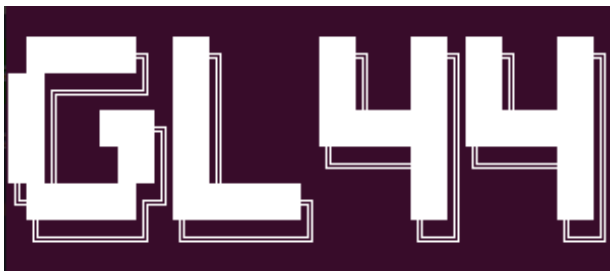

Documentations
Projet de Génie Logiciel

Version 2.3.1

GL44
Aymeric Baron - Carmel Benoit - Eliott Dumont
Jérôme Duveau - Jules Roques
January 22, 2024

Contents

1 Manuel utilisateur **2**
1.1 Spécificités de notre compilateur 2
1.1.1 Ses avantages 2
1.1.2 Ses limites 2
1.2 Gestion des messages d'erreur 3
1.3 Utiliser notre compilateur 3

1 Manuel utilisateur

1.1 Spécificités de notre compilateur

Cette section présente notre compilateur sous tous ses aspects, sur son fond (ses avantages et faiblesses) et dans sa forme (utilisation et limites).

1.1.1 Ses avantages

Le compilateur permet de compiler l'ensemble des programmes Deca n'utilisant pas les notions de cast, instance of ainsi que tout ce qui tourne autour des classes. De ce fait, il est capable de convertir tous les fichiers contenant instructions, structures conditionnelles et tout ce que cela implique, à condition que le programme soit correct (syntaxiquement et contextuellement).

De plus, notre compilateur est robuste aux erreurs de l'utilisateur. Les erreurs sont relevées correctement pour tous les programmes n'utilisant pas les notions de cast et instance of (fonctionne aussi pour les programmes avec classe).Dans le cas d'une erreur lexicale (token non reconnu, par exemple) ou syntaxique, le compilateur renvoie un message d'erreur précis avec la position de l'erreur. Sinon, dans le cas d'erreur contextuelle, le compilateur saura indiquer à l'utilisateur l'origine de l'erreur afin de simplifier la correction de celle-ci.

Le compilateur n'utilise pas de registres inutilement. En effet, l'algorithme de gestion de mémoire n'est pas aussi naïf que celui présenté dans le cours vidéo. Par exemple, lors d'une opération arithmétique binaire, il est proposé de systématiquement charger dans un registre les opérandes droite et gauche. Dans notre compilateur, seule l'opérande gauche est chargée, ce qui permet d'économiser de l'espace mémoire.
De plus, la gestion de pile est fonctionnelle (testée par plusieurs programmes), et assure la cohérence des résultats même lorsque beaucoup de registres sont utilisés simultanément.

Notre compilateur est également doté d'une classe Math dans la bibliothèque standard qui permet à l'utilisateur de calculer le sinus, le cosinus, l'arc-sinus, l'arc-tangente et l'ulp d'un flottant. La spécification de la méthode ulp est la même qu'en java. La précision des fonctions trigonométriques sera détaillée dans la documentation propre à l'extension TRIGO.

1.1.2 Ses limites

Comme spécifié dans le paragraphe précédent, ce compilateur ne permet pas de compiler des programmes utilisant des classes.

1.2 Gestion des messages d'erreur

Différentes erreurs sont relevées par le compilateur :

- Les messages d'erreurs provenant du lexer et du parser. Ils renvoient la place de l'erreur dans le fichier Deca ainsi que sa nature.
- Les messages d'erreurs dus au contexte. Ils renvoient le fichier Deca relevant un problème suivi de la localisation de l'erreur ainsi que la nature de celle-ci sous la forme :

  ```
  *fichier.deca:ligne:colonne: Nature du probleme
  ```

  Pour les erreurs dérivant d'une règle de la syntaxe contextuelle de Deca, celle-ci est spécifiée dans le message d'erreur.

1.3 Utiliser notre compilateur

Pour utiliser le compilateur , il suffit de créer un fichier Deca, puis on utilise le script suivant :

```
$src/main/bin/decac option *fichier.deca
```

Par exemple grâce à la commande :

```
$src/main/bin/decac src/test/deca/syntax/valid/provided/hello.deca
```

on exécutera le fichier hello.deca sans options.

Ci-après la liste des options disponibles sur notre compilateur :

- -b (banner) : affiche une bannière indiquant le nom de l'équipe (doit être utilisé seule).
- -p (parse) : arrête decac après l'étape de construction de l'arbre, et affiche la décompilation de ce dernier (i.e. s'il n'y a qu'un fichier source à compiler, la sortie doit être un programme deca syntaxiquement correct).
- -v (verification) : arrête decac après l'étape de vérifications (ne produit aucune sortie en l'absence d'erreur).
- -n (no check) : supprime les tests à l'exécution spécifiés dans les points 11.1 et 11.3 de la sémantique de Deca.
- -r X (registers) : limite les registres banalisés disponibles à R0 ... RX-1, avec 4 <= X <= 16.
- -d (debug) : active les traces de debug. Répéter l'option plusieurs fois pour avoir plus de traces.
- -P (parallel) : s'il y a plusieurs fichiers sources, lance la compilation des fichiers en parallèle (pour accélérer la compilation).
- -s (show) : affiche le fichier assembleur généré sur la sortie standard.