

COMPILEUR DU LANGAGE DECA

Vendredi 24 janvier 2025

BENJELLOUN Otman - Chef de projet
BIRÉE Thomas
BOULOUZ Amine
EL GOUMIRI Rida
LOGINOVA Aleksandra

THE TEAM

Otman : Chef de projet et partie C

Thomas : Partie A et partie C

Amine : Partie B

Rida : Partie B et tests

Aleksandra : Tests

AVANCEMENT DEPUIS LE DERNIER SUIVI



Bugs

- Ambiguïté liées aux environnements d'expressions
- Bon empilement des environnements lors des déclarations (champs de classe, méthodes)
- Gestion de la stack pour la partie C



Implementation

- Partie A Objet - Rapide
- Partie B Objet - Chronophage
- Partie C Objet - Complexe



Documentation

- Manuel utilisateur
- Rapport gestion d'équipe
- Documentation validation
- Documentation développeur
- Documentation extension

```
{  
    println("hello gl13");  
}
```

fichier .deca

choix des options de compilation



```
OBRACE: [@0,1:1='{'<18>,2:0]
PRINTLN: [@1,6:12='println',<2>,3:3]
OPARENT: [@2,13:13='('<20>,3:10]
STRING: [@3,14:25='hello gl13'',<47>,3:11]
CPARENT: [@4,26:26=')',<21>,3:23]
SEMI: [@5,27:27=';',<22>,3:24]
CBRACE: [@6,29:29='}',<19>,4:0]
```

lexer output

```
> [2, 0] Program
+> ListDeclClass [List with 0 elements]
`> [2, 0] Main
  +> ListDeclVar [List with 0 elements]
  `> ListInst [List with 1 elements]
    []> [3, 3] Println
      `> ListExpr [List with 1 elements]
        []> [3, 11] StringLiteral (hello gl13)
```

parser output

```
> [2, 0] Program
+> ListDeclClass [List with 0 elements]
`> [2, 0] Main
  +> ListDeclVar [List with 0 elements]
  `> ListInst [List with 1 elements]
    []> [3, 3] Println
      `> ListExpr [List with 1 elements]
        []> [3, 11] StringLiteral (hello gl13)
          type: string
```

Vérification Contextuelle + Décoration

Code generation



```
TSTO #0 ; Check for stack overflow
BOV stack_overflow_error
ADDSP #0
; Start of the main program
; VTABLE
; Main program
; Beginning of main instructions:
WSTR "hello gl13"
WNL
HALT
; Errors:
stack_overflow_error:
WSTR "ERROR: Stack overflow detected,
WNL
ERROR
; End of the main program
```

ou

```
2 .arch armv7-a
3 .arm
4
5
6 .section .data
7 string_format: .asciz "%s\n"
8 string_val.0: .string "hello gl13"
9 .balign 4
10
11
12 .global main
13 .section .text
14 main:
15   MOV fp, sp
16   MOV sb, sp
17
18
19   LDR r0, =string_format
20   LDR r1, =string_val.0
21   BL printf
22
23
24   MOV r7, #1
25   MOV r0, #0
26   SVC #0
```

IMA

ARM

OPTIONS DE COMPIRATION

OPTION 1

-p : Arrête la compilation après l'étape de construction de l'arbre abstrait et affiche sa décompilation.

OPTION 2

-v : Arrête la compilation après l'étape de vérifications contextuelles.
Ne produit aucune sortie en l'absence d'erreur.

OPTION 3

-r <X> : Limite les registers disponibles à R0 ... R{X-1}, 4 <= X <= 16.

OPTION 4

-n : Supprime les tests indiqués dans les points 11.1 and 11.3 de la Semantique Deca.

OPTION 5

-P : Lance la compilation des fichiers en parallèle lorsque plusieurs fichiers sources sont fournis.

OPTION 6

-arm : Génère du code assembleur pour l'architecture ARM.

LIMITES

Il est très difficile de garantir qu'il n'existe pas de cas complexes où notre compilateur pourrait produire des résultats erronés. Telle est la nature d'un tel projet.

TESTS



Anticipation des tests

Des tests sont conçus et doivent être passés pour valider la réalisation de cette partie.



mvn test

Vérification des options du compilateur, de l'analyse syntaxique, de la vérification contextuelle et de la génération de code.



Couverture de code de **80%**

Pour fr.ensimag.deca



Exemple des erreurs

Illustration de 3 messages d'erreur; une erreur syntaxique, une erreur contextuelle et une erreur de génération de code.

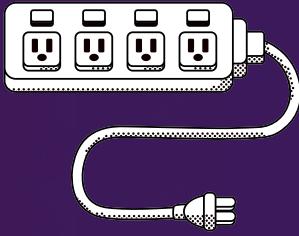
Polymorphisme

Abstraction

Programmation
défensive



- Overrides inutiles
- Verifications répétées.



EXTENSION ARM

**MERCI !
DES QUESTIONS ?**

