

Compilation de Mini-Lustre vers LLVM

Lemaire & Patault

Université Paris-Saclay

January 15, 2022

1 Introduction

- Présentation LLVM
- Utilisation LLVM

2 Travail Réalisé

- Montrer un peu le code
- Benchmarks
- Démonstration

Présentation Générale

- Utilisation de l'interface C++ de LLVM par les bindings OCaml

↳ `Llvm.declare_function "main" ...`

- Génération du code LLVM IR (fichiers .ll)

```
declare i32 @main() #0 {  
  entry:  
    ...  
}
```

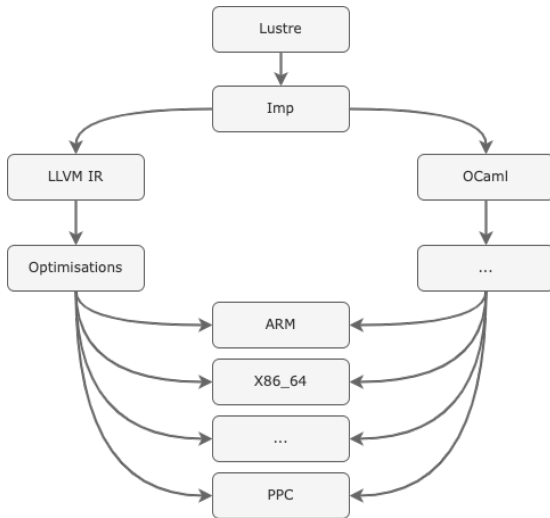
Pourquoi compiler vers LLVM ?

■ “Assembleur de haut niveau”

- Typage
- Pointeurs
- Vecteurs
- Tableaux
- Structures
- Fonctions
- ...

■ Bindings simples d'utilisation

■ Compilateur optimisant vers toutes les architectures



1 Introduction

- Présentation LLVM
- Utilisation LLVM

2 Travail Réalisé

- Montrer un peu le code
- Benchmarks
- Démonstration

Exemple de transformation

```
let a = 2 in  
let a = 2 in  
let a = 2 in  
let a = 2 in  
let a = 2 in  
let a = 2 in  
let a = 2 in  
let a = 2 in
```

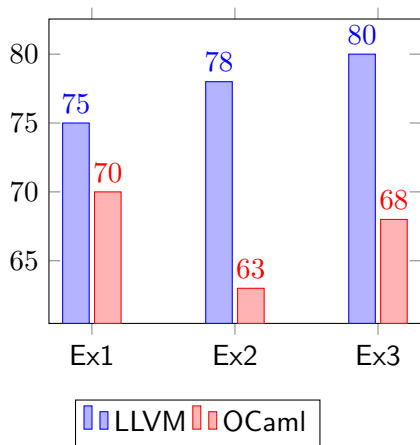
 \mapsto

```
store float %11, float* %aux4, align 4  
%12 = load float, float* %aux4, align 4  
store float %12, float* %f, align 4  
%13 = alloca %tuple_t_0, align 8  
%14 = getelementptr inbounds %tuple_t_0, %tuple_t_0*  
%15 = load i32, i32* %aux1, align 4  
store i32 %15, i32* %14, align 4
```

Performances

	file1.mls	file2.mls	file3.mls
OCaml	11s	9s	19s
LLVM	9s	8s	1s

Table: Table des performances comparées en fonction du langage cible



En direct