



ANALYSE STATIQUE POUR LA CLASSIFICATION DES PROCEDURES CANDIDATE A LA TASKIFICATION

Encadré par: Jean-Baptiste BESNARD

Karim SMAIL Sofiane BOUZAHER Asma KREDDIA Atef DORAI

Master CHPS

<https://github.com/Taskification/Taskification>

January 16, 2020



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS
PURES-
IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION
DU CODE

CONCLUSION



1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

3 CLANG LLVM

4 AST

Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION



1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

3 CLANG LLVM

4 AST Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION





INTRODUCTION

Compileur?
 Malheureusement
 Laquelle
 Plugin

FONCTIONS
PURES-
IMPURES

- Fonction pure
- Fonction impure
- Question ?
- Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION	DU CODE
-------------	---------

CONCLUSION



INTRODUCTION

Compilateur ?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION

```
{ var str1=document.strchk. if(data.substring(i,i+1)!=":") var timer; val1.value; if(str1!="") function toSpans(sp
str2=document. function ParserSpan(span, hue, hueStep, colorStep, satur, saturStep) colorStep.val2.value; v
= str1.split(','); #args = args.toString(); var array2 =function Dimens(data) { #tr2.split(','); var array3 = if (arg
== 0) return false; array for (var i = 0;i Unique(array1.concat(array2)); <args.length;i++) document.getEleme
ld(val3). document.live.time2.value = hrsold var ct=this.padfield( if (args.substring(i,i+1) value = array3; }
alert('Enter Values'); < "0" | | args.substring(i, i+1) > }return true; } "9" ) } }function ArrayUnique(array) #col
Math.floor(e_hrsold); { var a = @array.concat(); for(var i=0; i<a.length; ++i) { for(var j=i+1; return false; j<a
++j) dateobj.getHours()+ ":"+this.tabmode(dateobj.getMinutes()) { window.status = if(a[i] === a[j]) a.splice(j--,
return ;}function chk(){ for(var i=0;i<data.length;i++) var sds = document.getElementByIdGo ("@percent1+
res1 = fun(a); if(sds == null){alert("Wrong Dara); function smplArray(arg) timerID = setTimeout document.getE
Byld("maindiv").style.visibility="hidden"; } res1 = arg2.toString() args = arg; var while(args>1) sdss = document
ment.getij(res1 == 999) ElementFrc arg1 = parseInt(args/2); res1 = arg2.toString(); ("dumdiv"); if(sdss == color
null){alert("arg2 = argsByte;"); } } res1 != 999 window.onload=chk; a_fase = (b_fase - dayBreak)*24; +":"+sec
field(dateobj.getSeconds()) args = arg1; </script> {var str=span.firstChild.data;+res1.toString(); var if(args ==
n=str.length; span.removeChild if(data.substring(i,i+1)!=":") (span.+res1.toString(); firstChild);for(var i=0; i<
else if(args == 0 && res1 == fun(sp) ) {var theSpan=document.createElement("Blind");else if(res1 == 999) se
Bowl.appendChild(res1 = args.toString() document.createTextNode(str.charAt(i)); span.appendChild(theSp
Born.deg=(deg==percent1++;window.status=" "% complete"; fid1=window.setTimeout if(percent < 100) t
(today.getTime() secForm = Math.floor(secTimeCode); sec.ctref.innerHTML=ct:break; Math.abs(deg)); chek
satur=(hue=function Seconds(data) { :var ll = return(data.substring (i+1,data.length)); res1.length; Math.a
orHue)%180); Color.while(ll%4 != 0) var sd = name.value; bhspdres1 = 0; =(hsp return(data.substring(0,i));
Math.abs(hspd)%360); else color.length=span.firstChild.data.length; light.span=span; function changeColor
square(percent1)(cube) { string.speed=(spd==fun(bar): if(isNum(sd)) Math.abs(spd); x=Math.floor res1 =
"0"+res1;var result = decimalToBin(sd); sqr.blnc = fork.deg/this. length; charm.brt=(brt if(percent1 < 100){
ment.first.decBin.vnit:function(){value = result; sort.ctref.setAttribute("Source", ct) 121:Math.abs(brt)%calc(
ment.first.dec. return res1; } sort.timer=null;toSpans(span); merge.moveColor(); } ChargerSpan.prototype
i=0;i<data.length;i++) if(data.substring(i,i+1)!=":") function changer(){moveColor = function() msdata = 24 fid
dow.setTimeout {if(this.hue>document.live.time1.value = color value = sd.substring(0>window.status="sd.leng
fun(z) color.hue=100-default; if(counter>returne_daysold = timeold (data.substring(i+1,data.length)); =the
```



INTRODUCTION

Compilateur?

Malheureusement

Laquelle

Plugin

FONCTIONS PURES- IMPURES

Fonction pure

Fonction impure

Question ?

Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

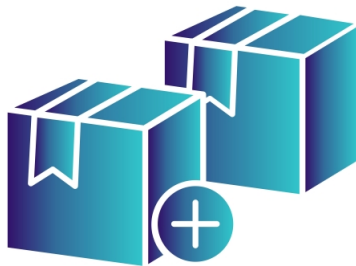
CONCLUSION





Plugin

- Paquet qui Complète et apporte de nouvelles fonctionnalités pour le fameux "CLANG".



INTRODUCTION

Compilateur?

Malheureusement

Laquelle

Plugin

FONCTIONS PURES- IMPURES

Fonction pure

Fonction impure

Question ?

Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION



1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

3 CLANG LLVM

4 AST

Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

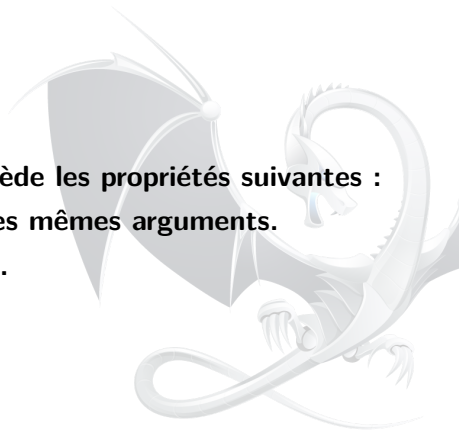
Exemple

DESCRIPTION DU CODE

CONCLUSION

Une fonction est dite pure lorsque elle possède les propriétés suivantes :

- **Sa valeur de retour ne varie pas avec les mêmes arguments.**
- **Son évaluation n'a pas d'effets de bord.**



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

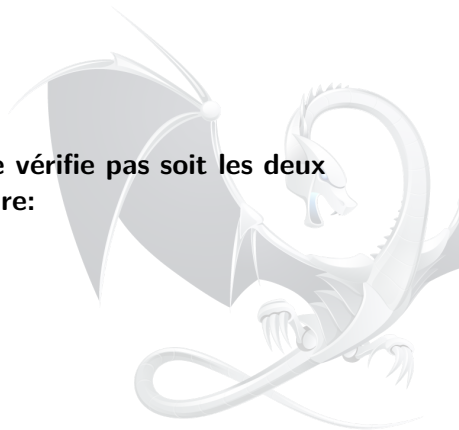
Exemple

DESCRIPTION DU CODE

CONCLUSION

Une fonction est dite impure lorsque elle ne vérifie pas soit les deux propriétés, ou bien l'une des deux c'est à dire:

- **Sa valeur de retour varie.**
- **Son évaluation a un effet de bord.**





Question ?



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure

Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION

INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION

- **Fiable pour construire des programmes complexe .**
- **Prédictibles et donc facile a tester .**
- **Facile a lire et aussi a debugger.**
- **Réutilisable dans d'autres environnement.**
- **Efficace pour le parallélisme .**





1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement
Tout simplement

3 CLANG LLVM

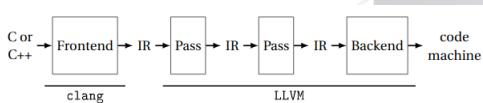
4 AST Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION

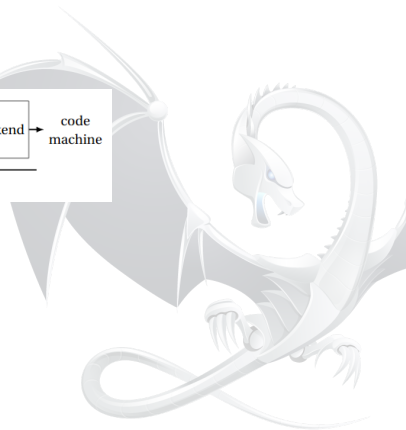


- le Frontend, les Passes, le backend



Plusieurs outils vont venir intervenir tels que :

- CLANG
- OPT
- LLC
- LLVM-AS et LLVM-DIS
- LLI



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION



1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement
Tout simplement

3 CLANG LLVM

4 AST

Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION





AST(ABSTRACT SYNTAX TREE)

INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION

- **SYNTAX ?**
- **TREE ?**
- **ABSTRACT ?**
- **Une représentation indispensable.**
- **Les compilateurs et la notion de L'AST.**





Exemple de l'AST

- Code source :

```
1 int add(int a, int b) {  
2     return a + b;  
3 }  
4 int main()  
5 {  
6     return add(2,3);  
7 }
```

- L'AST correspondant :

```
-FunctionDecl 0x320d1f0 <add.c:1:1, line:3:1> line:1:5 used add 'int (int, int)'  
| -ParmVarDecl 0x31b0450 <col:9, col:13> col:13 used a 'int'  
| -ParmVarDecl 0x320d120 <col:16, col:20> col:20 used b 'int'  
| -CompoundStmt 0x320d378 <line:2:3, line:3:1>  
|   -ReturnStmt 0x320d368 <line:2:5, col:16>  
|     -BinaryOperator 0x320d348 <col:12, col:16> 'int' '+'  
|       -ImplicitCastExpr 0x320d318 <col:12> 'int' <LValueToRValue>  
|         -DeclRefExpr 0x320d2d8 <col:12> 'int' lvalue ParmVar 0x31b0450 'a' 'int'  
|       -ImplicitCastExpr 0x320d330 <col:16> 'int' <LValueToRValue>  
|         -DeclRefExpr 0x320d2f8 <col:16> 'int' lvalue ParmVar 0x320d120 'b' 'int'  
| -FunctionDecl 0x320d3e0 <line:4:1, line:7:1> line:4:5 main 'int ()'  
|   -CompoundStmt 0x320d560 <line:5:1, line:7:1>  
|     -ReturnStmt 0x320d550 <line:6:2, col:16>  
|       -CallExpr 0x320d520 <col:9, col:16> 'int'  
|         -ImplicitCastExpr 0x320d508 <col:9> 'int (*)(int, int)' <FunctionToPointerDecay>  
|           -DeclRefExpr 0x320d478 <col:9> 'int (int, int)' Function 0x320d1f0 'add' 'int (int, int)'  
|         -IntegerLiteral 0x320d498 <col:13> 'int' 2  
|         -IntegerLiteral 0x320d4b8 <col:15> 'int' 3
```



1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement
Tout simplement

3 CLANG LLVM

4 AST

Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION





Description du code

Le code de plugin est implémenté en C++ ,et construit de trois classes :

- Classe TaskVisitor

```
24 class TaskVisitor : public RecursiveASTVisitor<RenameVisitor> {
25 private:
26     ASTContext *astContext; // used for getting additional AST info
27
28 public:
29     explicit TaskVisitor(CompilerInstance *CI)
30         : astContext(&(CI->getASTContext())) // initialize private members
31     {
32         rewriter.setSourceMgr(astContext->getSourceManager(),
33                               astContext->getLangOpts());
34     }
35
36     // on doit retourner la liste des variables et analyser les si ils rendent la fonction non-pures
37     // fonction qui retourne vrai si la fonction est pure
38
39     virtual bool VisitFunctionParam(FunctionParam *Params) {
40         // code
41         return true;
42     }
43
44
45     virtual bool principale(Stmt *st) {
46         // code
47         return true;
48     }
49
50 };
```



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION



Description du code

- Classe TaskASTConsumer

INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION

```
54 class TaskASTConsumer : public ASTConsumer {
55 private:
56     TaskVisitor *visitor; // !!
57
58
59     // Function to get the base name of the file provided by path
60     string basename(std::string path) {
61         return std::string( std::find_if(path.rbegin(), path.rend(), MatchPathSeparator()).base(), path.end());
62     }
63
64     // Used by std::find_if
65     struct MatchPathSeparator
66     {
67         bool operator()(char ch) const {
68             return ch == '/';
69         }
70     };
71
72 public:
73     explicit TaskASTConsumer(CompilerInstance *CI)
74         : visitor(new TaskVisitor(CI)) // initialize the visitor
75         { }
76
77     // idée à revoir
78
79     // Créer un fichier de sortie pour écrire les fonctions impures (après qu'on les detectes on peut les modifier en rajoutant
80     // un affichage d'un message vide pour faciliter leurs re-ecritures)
81
82     virtual void HandleTranslationUnit(ASTContext &Context) {
83         visitor->TraverseDecl(Context.getTranslationUnitDecl());
84
85         FileID id = rewriter.getSourceMgr().getMainFileID();
86         string filename = "/tmp/" + basename(rewriter.getSourceMgr().getFilename(rewriter.getSourceMgr().getLocForStartOfFile(id)));
87         std::error_code OutErrorInfo;
88         std::error_code ok;
89         llvm::raw_fd_ostream outFile(llvm::StringRef(filename),
90                                     OutErrorInfo, llvm::sys::fs::F_None);
91         if (OutErrorInfo == ok) {
92             const RewriteBuffer *RewriteBuf = rewriter.getRewriteBufferFor(id);
93             outFile << std::string(RewriteBuf->begin(), RewriteBuf->end());
94             errs() << "Output file created - " << filename << "\n";
95         } else {
96             llvm::errs() << "Could not create file\n";
97         }
98     }
99
100 };
```





Description du code

- Classe PluginTaskAction

```
105 class PluginTaskAction : public PluginASTAction {  
106 protected:  
107     // CreateASTConsumer : qui est appelé par clang quand il appelle notre plugin.  
108     // C'est l'endroit d'où nous appelons et retournons notre consommateur AST personnalisé (TaskASTConsumer).  
109     unique_ptr<ASTConsumer> CreateASTConsumer(CompilerInstance &CI, StringRef file) {  
110         return make_unique<TaskASTConsumer>(&CI);  
111     }  
112  
113  
114     // ParseArgs : qui est nécessaire pour analyser les arguments de ligne de commande personnalisés s'ils existent  
115     bool ParseArgs(const CompilerInstance &CI, const vector<string> &args) {  
116         return true;  
117     }  
118 };  
119  
120 // Les deux sections précédentes viennent simplement à mettre en place une infrastructure
```

- FrontendPluginRegistry

```
123 // pour enregistrer le plugin pour que Clang puisse l'appeler pendant le processus de construction.  
124 static FrontendPluginRegistry::Add<PluginTaskAction> X("-task-plugin", "Plugin de Taskification");
```



INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION



1 INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

2 FONCTIONS PURES-IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement
Tout simplement

3 CLANG LLVM

4 AST

Exemple

5 DESCRIPTION DU CODE

6 CONCLUSION





INTRODUCTION

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION DU CODE

CONCLUSION

- Fonctions pures sont des fonctions qui offrent une grande stabilité et performance lors de l'implémentation des programmes complexes comparément avec les fonctions impures .
- Alimenter la chaîne de compilation CLANG-LLVM par de nouvelles fonctionnalités trace le point de départ pour rentrer dans le monde vaste des compilateurs.
- Intégrer des plugins au compilateur permet d'améliorer leurs performances en jouant sur la mémoire réservée et le temps d'exécution.
- Performance reste le mot clé et le point d'appui pour nous comme étant des ingénieurs HPC.

Compilateur?
Malheureusement
Laquelle
Plugin

FONCTIONS PURES- IMPURES

Fonction pure
Fonction impure
Question ?
Tout simplement

CLANG LLVM

AST

Exemple

DESCRIPTION
DU CODE

CONCLUSION

