

Direct Embedding

Joris Beau

EPFL - LAMP

Project in computer science

Directed by **Martin Odersky**, Supervised by **Vojin Jovanovic**

January 15, 2015

Outline

- 1 Introduction
 - Motivation, Goal & Features
 - Initial constraints
 - Context
- 2 Implementation
 - Simple cases prototype
 - One case example
- 3 Demonstration

Motivation

- Instigated by Slick database library
- Embedding DSLs simply!
- Provide a painless logic for reification
- Direct embedding has advantages:
 - better error messages
 - compile-time

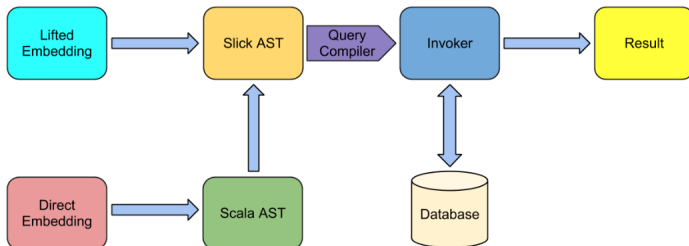
Goal

- ① Educational aspect
- ② Implement a core prototype
- ③ Make it work
- ④ Extend it
- ⑤ Might evolve to a library

Features

- Take care of macro
- Handle Intermediate Representation (IR)
- \implies users code in Scala-like language in Scala projects
- \implies users' friendly

Big Picture of Slick



Direct vs lifted

Direct

- AST generated compile-time
- macro based
- Scala type
- at runtime, errors for unsupported methods
- experimental on Slick

Lifted

- changes into a deep DSL IR
- operator overloading
- deep type
- caught at compilation unsupported operators
- used by Slick, LMS

Reification: Need to modify Scala AST

Beginning

How to modify Scala AST?

Beginning

How to modify Scala AST?

→ **quasiquote?**

Beginning

No quasiquote

but ...

Beginning

but

we can access to the
symbols



@ANNOTATIONS

```
@reifyAs(ClassCons)
class ClassExample {
  ???
}
```

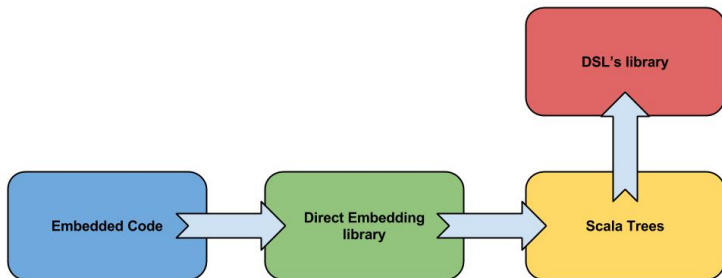
Objects

Cases	Object	Nested	Classes
<code>val value</code>	✓	✓	✓
<code>def foo</code>	✓	✓	✓
<code>def foo(args)</code>	✓	✓	✓
<code>def foo[T, U]: (T, U)</code>	✓	✓	✓
<code>def foo[T, U](t: T, u: U): (T, U)</code>	✓	✓	✓
<code>def foo[T](t₁: T)(...)(t_a: T)</code>	✓	✓	✓

Language specification

if	X
while	X
do while	X
lazy val	X
return	X

Overview



An example: embedded code

Annotate

```
@reifyAs(JustArgs)
def justArgs(x: Int): Int = ???
```

Lift

```
lift {
  ObjectExample.justArgs(1)
}
```


An example: tree

Tree

```
ObjectExample.justArgs(1)  
  annotated  
directembedding.reifyAs(JustArgs)
```

Raw tree

```
Apply(Select(Ident(ch.epfl.directembedding.test.ObjectExample),  
TermName("justArgs" )), List(Literal(Constant(1))))
```

An example: macro

From symbol, arguments, type, we can use the annotation to reify the tree:

Macro

\Rightarrow macro q"..."

Returns

JustArgs.apply(1)

Going further

- Operator
- Recursion
- Raw block of code

Demonstration

Thank you

Special thanks