Direct Embedding

Joris Beau

EPFL - LAMP
Project in computer science
Directed by **Martin Odersky**, Supervised by **Vojin Jovanovic**

January 14, 2015

Outline

Motivation

- embedding DSLs, simply!
- Slick

Goal

1. Implement a core prototype
2. Make it work
3. Extend it

Beginning

No quasiquote

but ...

Beginning

but

we can access to the symbols $\rightarrow$ @ANNOTATIONS

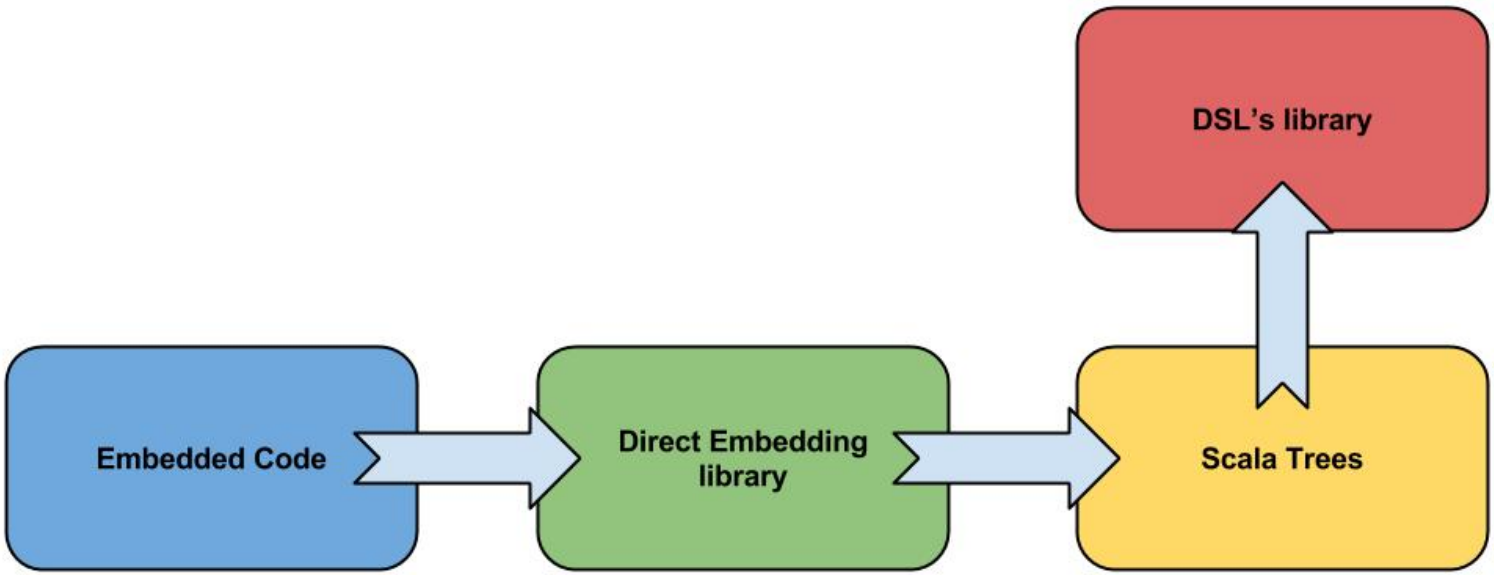
Objects

Cases	Object	Nested	Classes
val value	✓	✓	✓
def foo	✓	✓	✓
def foo(args)	✓	✓	✓
def foo[T, U]: (T, U)	✓	✓	✓
def foo[T, U](t: T, u: U): (T, U)	✓	✓	✓
def foo[T](t_1: T)(...)(t_a: T)	✓	✓	✓

Language specification

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

Overview


DSL's library
Embedded Code
Direct Embedding library
Scala Trees

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

$\Longrightarrow$ macro q"..."

Returns

JustArgs.apply(1)

Going further

- ...

Demo

Thank you