

Функциональный взгляд на визиторы

Продолжение: менее функциональный взгляд

А. М. Пеленицын
apel@sfedu.ru

Южный федеральный университет
Институт математики, механики и компьютерных наук им. И. И. Воровича
Кафедра информатики и вычислительного эксперимента

29 апреля 2016
Семинар «Языки программирования и компиляторы»



Содержание

- 1 Решения Оливейры
 - Объектные алгебры (2012)
 - «Тривиальное» ковариантное решение (2016)
- 2 Заключение: смежные задачи, литература



Идея: заменить конструкторы узлов на обобщённые методы

Объектная алгебра

```
interface IntAlg<A> {
    A lit(int x);
    A add(A e1, A e2);
}
```

Использование

```
static <A> A e(IntAlg<A> a)
    { return a.add(
        a.add(a.lit(4),
              a.lit(2)),
        a.lit(1));
    }
...
out.print(e(new Eval()));
```

Визитор

```
class Eval implements IntAlg<Integer> {
    public Integer lit(int x)
        { return x; }
    public Integer add(Integer e1, Integer e2)
        { return e1 + e2; }
}
```



Идея: расширяемость на основе (ковариантного) изменения типов узлов-потомков, Scala

Типы узлов

```

trait Exp { def eval() : Int }
trait Lit extends Exp {
  val x: Int
  def eval() = x }
trait Add extends Exp {
  val e1, e2 : Exp
  def eval() = e1.eval + e2.eval }
  
```

Визитор 1: печать

```

trait ExpP extends Exp {def print(): String}
trait LitP extends Lit with ExpP {
  def print() = "" + x }
trait AddP extends Add with ExpP {
  val e1, e2 : ExpP // type refined!
  def print() =
    "(" + e1.print + "+" + e2.print + ")" }
  
```

Визитор 2: сбор списка литералов

```

trait ExpC extends Exp {
  def collectLit(): List[Int] }
trait LitC extends Lit with ExpC {
  def collectLit(): List[Int] = x::List() }
trait AddC extends Add with ExpC {
  val e1, e2 : ExpC
  def collectLit() : List[Int] =
    e1.collectLit ::: e2.collectLit }
  
```

Узлы, принимающие 2 визитора

```

trait ExpPC extends ExpP with ExpC
trait LitPC extends LitP with LitC
  with ExpPC
trait AddPC extends AddP with AddC
  with ExpPC {
  val e1, e2 : ExpPC }
  
```

Википедия: Expression Problem#Solutions

- Multimethods[11]
 - Open classes[12]
 - Coproducts of functors[13]
 - Type classes[14]
 - Tagless-final[15] /
Object algebras[16]
- [11] Статья 95-го года.
- [12] Статья 2000-го года: MultiJava.
- [13] W. Swierstra (2008). “Data Types à La Carte”.
- [14] JavGI (2011).
- [15] J. Carette, O. Kiselyov, C. Shan (2009). “Finally Tagless, Partially Evaluated: Tagless Staged Interpreters for Simpler Typed Languages”.
- [16] Оливейра (2012).



Литература или Почему это интересно функциональщикам?

Embedded DSL

- 1 W. Swierstra (2008). [Data Types à La Carte](#) // JFP.
- 2 J. Carette, O. Kiselyov, C. Shan (2009). [Finally Tagless, Partially Evaluated: Tagless Staged Interpreters for Simpler Typed Languages](#) // JFP.
- 3 J. Gibbons (2014). [Folding Domain—Specific Languages: Deep and Shallow Embeddings](#) // ICFP.
- 4 A. Löh (2015) et al. [Type-level Web APIs with Servant](#) // WGP.

Статьи [2]–[4] ссылаются на [1].

Хаскель-аналог объектных алгебр можно найти в [4, п. 4.5].



Популярные ссылки

- Реализация по Киселёву на нашем форуме.
- Тред на Реддите: Is “Solving the Expression Problem” worth the bother?
- Реализация A-ля карт и смежных техник на Hackage.

