

Поддержка системы типов языка Dotty в Scala-плагине для IntelliJ IDEA

Козлов Андрей Дмитриевич

научный руководитель: Подхалюзин Александр Викторович

СПб АУ НОЦНТ РАН

13 июня 2016 г.

Scala —

мультипарадигмальный язык программирования для JVM.

DOT —

исчисление, созданное при попытке теоретически обосновать систему типов Scala.

- доказательство корректности (8 лет работы);
- система типов Scala:
 - 1 слишком сложная;
 - 2 недостаточно надежная.

Dotty —

экспериментальный компилятор Scala, основанный на DOT-исчислении.

23 октября 2015 г.:

- объявлено о возможности производить bootstrap компилятора;
- перечислены дальнейшие задачи, стоящие перед разработчиками языка.

Цель:

Одновременная поддержка систем типов Scala и Dotty в Scala-плагине для IntelliJ IDEA.

Задачи:

- 1 выделение абстракции системы типов;
- 2 статический анализ:
 - синтаксический анализ;
 - вывод типов выражений.

Фазы работы плагина:

- 1 построение абстрактного синтаксического дерева;
- 2 устранение синтаксического сахара (*desugaring*);
- 3 создание типов;
- 4 вывод типов выражений (термов);
- 5 проверка выполнения наложенных ограничений выведенными типами.

Виды синтаксических анализаторов:

- “рукописные”;
- генерируемые:
 - 1 ANTLR;
 - 2 Grammar-Kit.

Реализованное расширение позволяет поддерживать любые диалекты Scala.

Эмулируемые типы в Scala:

- 1 функциональные (`Function_n[-T1, ..., +R]`);
- 2 кортежи (`Tuple_n[-T1, ..., +R]`).

Устранение синтаксического сахара является подпроцессом вывода типов.

Пример дублирующей функциональности:

```
1 class Foo[T <: AnyVal] {} // generic type parameter  
2  
3 class Foo { type T <: AnyVal } // abstract type member
```

В системе типов Dotty устранены:

- 1 расширения типов;
- 2 generic-параметры типов;
- 3 экзистенциальные типы;
- 4 тип “конструктор типа”.

Алгоритм вывода типов не изменился.

Требуют изменения:

- 1 проверка отношения подтипирования;
- 2 проверка отношения эквивалентности;
- 3 алгоритм поиска наибольшей нижней и наименьшей верхней границы набора типов.

Реалии Scala-плагина:

- архитектура для поддержки ровно одного языка;
- отсутствие абстракций, соответствующих этапам статического анализа;
- плохая расширяемость.

- 1 Вынесены интерфейсы, отвечающие за различные фазы статического анализа;
- 2 добавлена возможность регистрировать реализации на уровне проекта;
- 3 добавлены специфичные для Dotty реализации фаз, связанных с системой типов.

- Проекты с различными системам типов;
- синтаксический анализ;
- устранение синтаксического сахара;
- проверка удовлетворения наложенным ограничениям.

- **DOT-исчисление:**
<http://github.com/namin/dot>
- **Компилятор Dotty:**
<http://github.com/lampepfl/dotty>
- **Scala-плагин:**
<http://github.com/JetBrains/intellij-scala>

Вопросы?

```
1 trait A
2 trait B extends A
3
4 trait Foo[T <: A] {
5 }
6
7 type Bar = Foo[A]
8
9 val bar: Bar = new Foo {
10   type Foo$T = B
11 }
```

Устранение синтаксического сахара

```
1 trait A
2 trait B extends A
3
4 trait Foo {
5     type Foo$T <: A
6 }
7
8 type Bar = Foo {
9     type Foo$T = A
10 }
11
12 val bar: Bar = new Foo {
13     type Foo$T = B
14 }
```


Вывод типов

```
1 trait A
2 trait B extends A
3
4 trait Foo {
5     type Foo$T <: A
6 }
7
8 type Bar = Foo {
9     type Foo$T = A
10 }
11
12 val bar: Bar = (new Foo {
13     type Foo$T = B
14 }: Foo {
15     type Foo$T = B
16 })
```

Проверка отношений эквивалентности и подтипирования

- 1 $B <: A \Rightarrow Foo[B] <: Foo[A];$
- 2 $Bar = Foo[A] \Rightarrow Foo[B] <: Bar.$