



Факультет компьютерных наук

Системное программирование

Москва 2025

Статический анализ программ на языке Scala на основе унифицированного абстрактного синтаксического дерева

Static analysis of Scala programs based on the unified abstract syntax tree

Выполнил:

Пасилецкий Даниил Олегович

Руководитель:

Профессор Базовая кафедра
«Системное программирование» ИСП
РАН, факультета компьютерных наук,
Белеванцев Андрей Андреевич

Консультант:

Стажер-исследователь ИСП РАН,
Афанасьев Виталий Олегович



Scala

```
1. object Magic extends App {  
2.   case class ~>[A](a: A) { def ~>[B](b: B) = (a,b) }  
3.   implicit class 🧙(s: Symbol) { def →(x: Any) = s.name -> x }  
4.  
5.   val config = new ~>('port → 8080) ~> ('host → "localhost")  
6.  
7.   def apply(f: PartialFunction[Any,Unit]) = f  
8.   val logic = apply { case (('port, p), ('host, h)) => println(s"$h:$p") }  
9.  
10.  logic(config)  
11. }
```

Листинг 1. Пример кода на Scala



Проблема

В анализаторе Svace уже есть поддержка языка Scala, но она работает только на уровне **байткода JVM**, что не позволяет обнаруживать ряд ошибок, которые **видны только в исходном коде**.

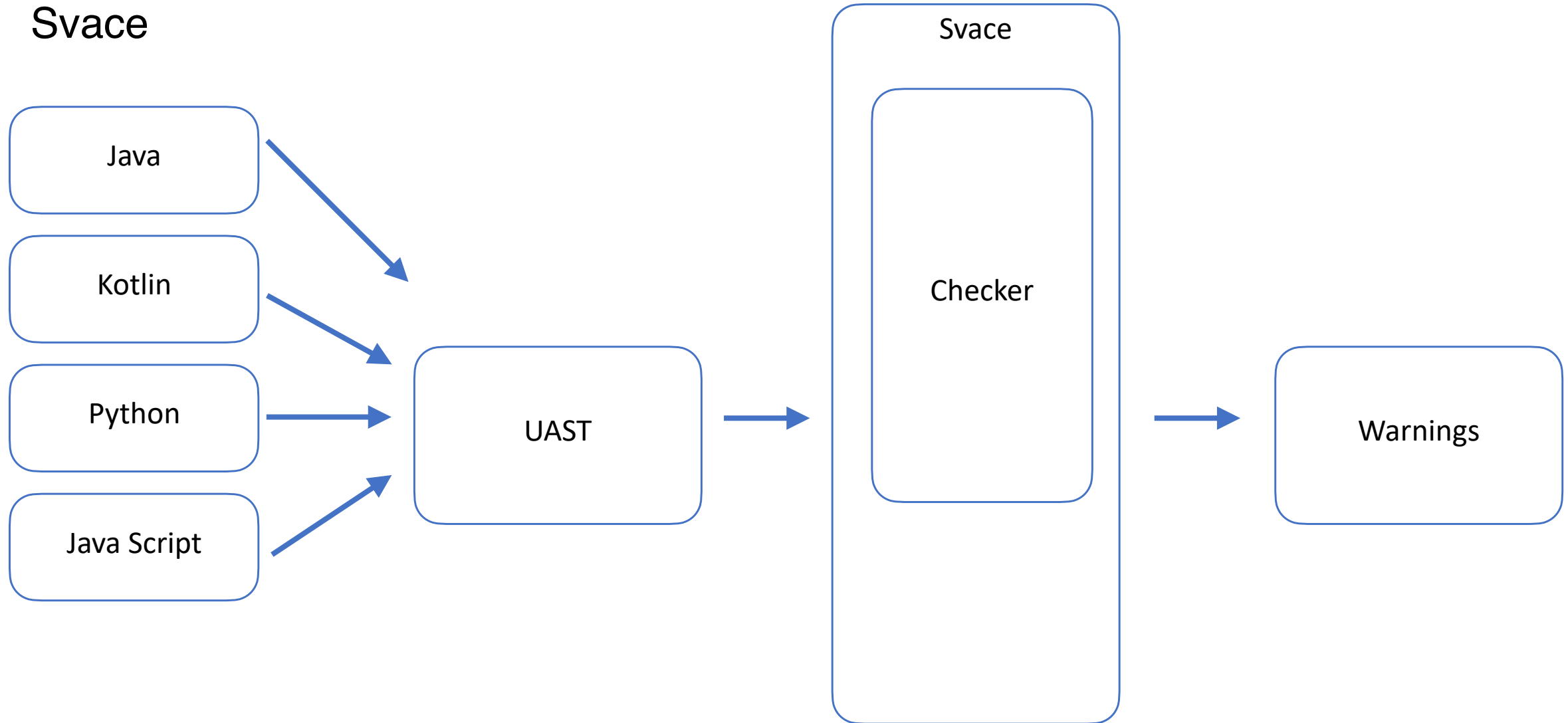
```
1. case class User(name: String, age: Int)
2. val user = User("Alice", 30)
3. user match {
4.     case User(age, name) => println(s"$name - $age")
5. }
```

Ouptut: 30 - Alice

Листинг 2. Несовпадение case class и unapply
(pattern matching)

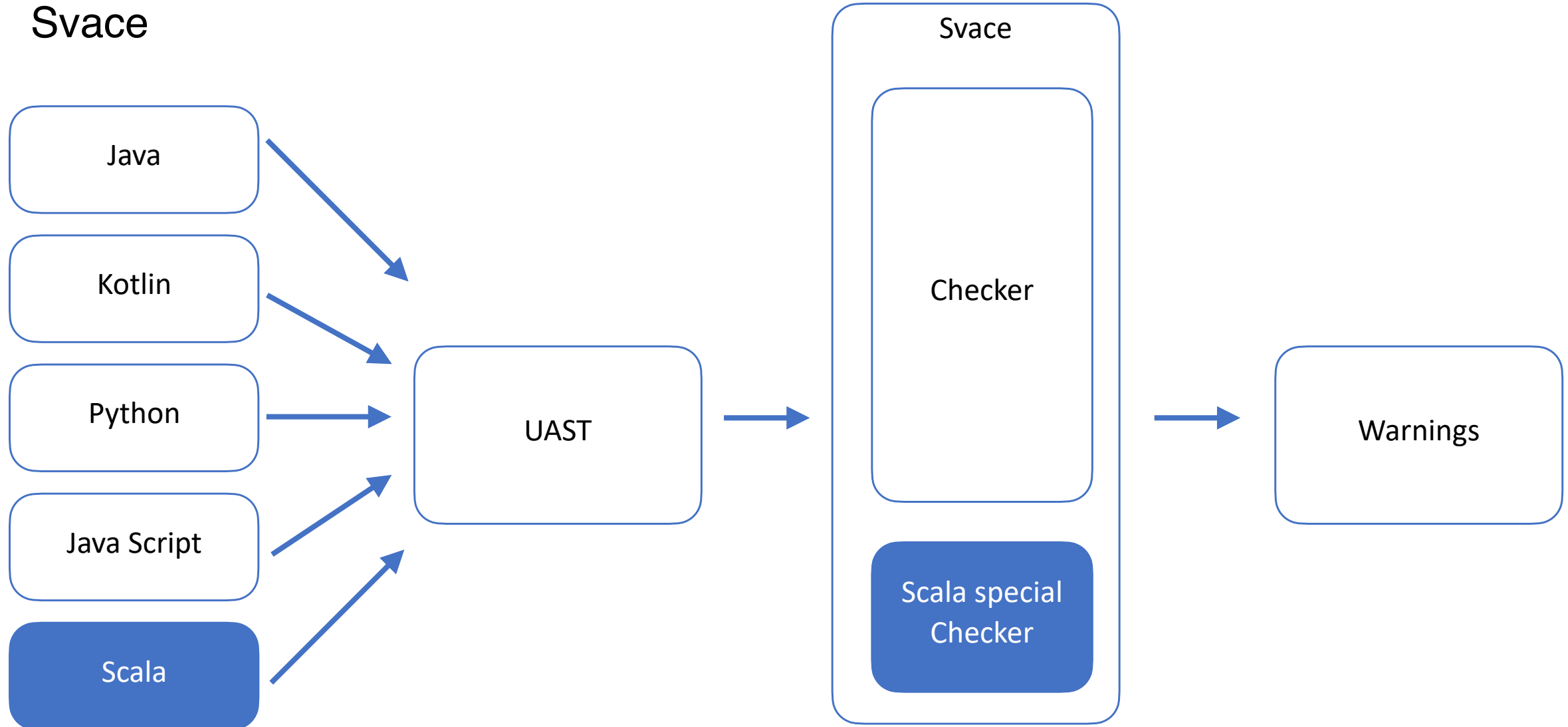


Svace





Svace





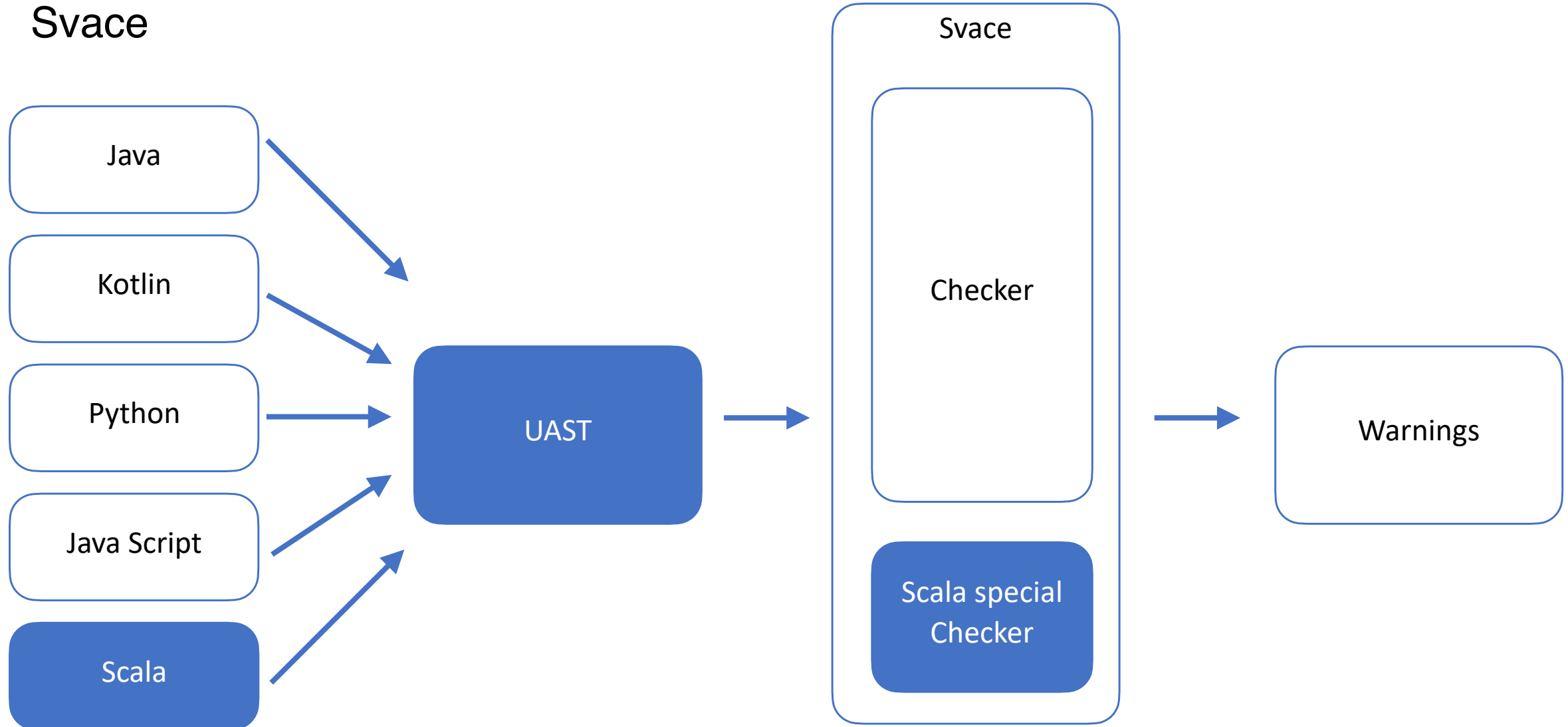
Scala

```
1. package A
2. package B
3. package C {
4.     object Hello {
5.         def main(args: Array[String]) = {
6.             println("Hello, world")
7.         }
8.     }
9. }
```

Листинг 3. Пример кода HelloWorld с пакетами



Svace





Постановка задачи

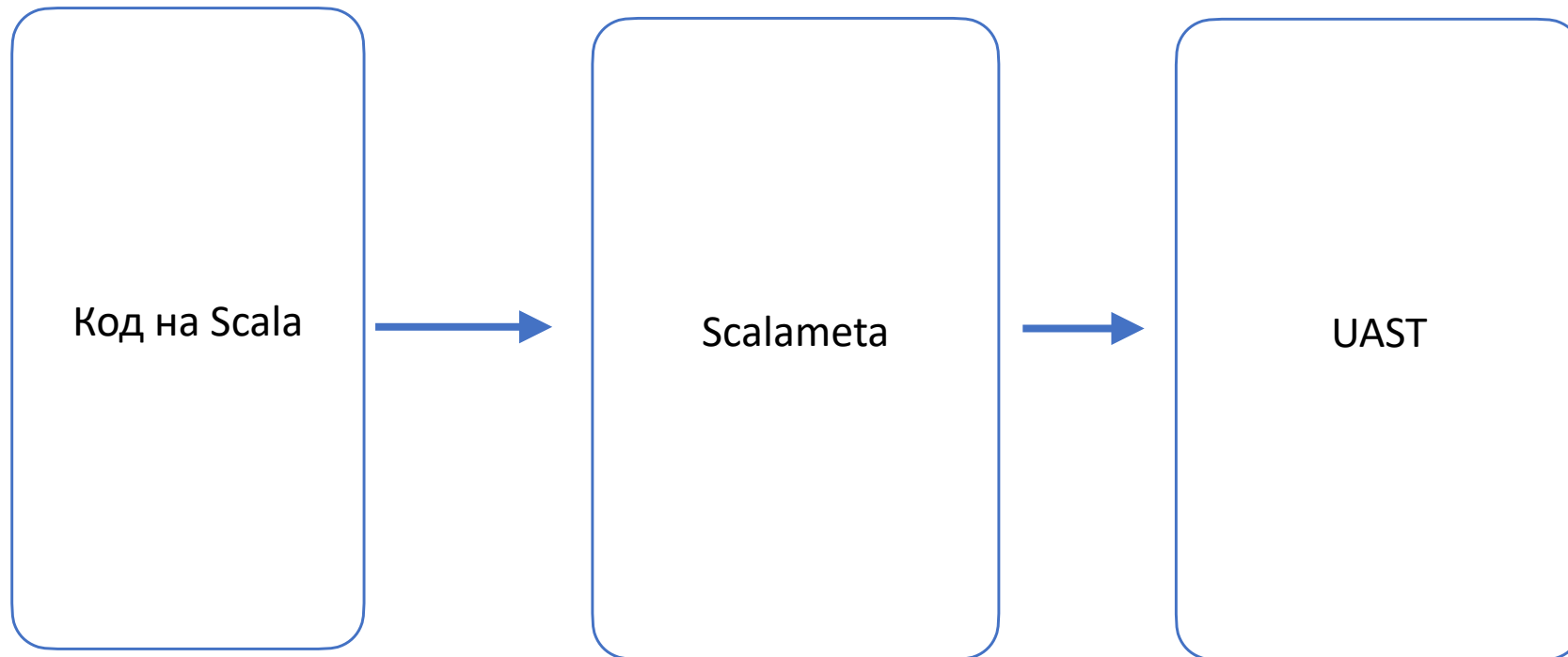
Цель: Добавить поддержку анализа исходного кода Scala на основе абстрактного синтаксического дерева в систему статического анализа Svace.

Для достижение этой цели необходимо выполнить следующие задачи:

1. Разработать транслятор из исходного кода на языке Scala (версии 2.0) в унифицированное абстрактное синтаксическое дерево;
2. Разработать Scala специфичные чекеры;
3. Провести экспериментальную оценку результата



Транслятор





Оценка результатов

Набор данных:

Открытые проекты написанные на Scala 2.

Объём: несколько сотен тысяч строк кода.

Сценарий анализа

Запускаем стандартный анализ Scala на уровне JVM байткода

И запускаем анализ с AST модулем

Метрики

Количество найденных ошибок

Время анализа



План работ

21.11.2025

01.01.2026

01.03.2026

Базовые программы
Scala -> UAST

Транслятор
Scala -> UAST

Специализированные
чекеры для Scala



Факультет компьютерных наук

Системное программирование

Москва 2025

Статический анализ программ на языке Scala на основе унифицированного абстрактного синтаксического дерева

Static analysis of Scala programs based on the unified abstract syntax tree

Выполнил:
Пасилецкий Даниил Олегович

Руководитель:
Профессор Базовая кафедра
«Системное программирование» ИСП
РАН, факультета компьютерных наук,
Белеванцев Андрей Андреевич

Консультант:
Стажер-исследователь ИСП РАН,
Афанасьев Виталий Олегович