Основанный на данных синтез кода в IntelliJ IDEA

Выполнил: Владислав Танков, 444 группа
Научный руководитель: к. т. н., доцент Тимофей Александрович Брыксин
Рецензент: Алексей Александрович Шпильман

Задача синтеза кода

- Синтезатор кода позволяет получать алгоритм из неполной его спецификации
- Задача актуальна для промышленного программирования
  - Генерация примеров
  - Генерация тестов
  - Автодополнение
- Существующие синтезаторы используют специфичные формализмы

BSL-синтезаторы

- Bayesian Sketch Learning синтезаторы, представлены в 2017 году
  - Статья: https://arxiv.org/abs/1703.05698v1
- DSL порождается из целевого языка
- Генерация специфичного для библиотек кода
  - Утилитарные методы
- Эталонная реализация — Bayou (JVM/Python)

Пример (Bayou)

```
public class TestIO {
    void delete(String file) {
        Evidence.call(“delete”);
        Evidence.type(“File”);
    }
}
```

```
import java.io.File;

public class TestIO {
    void delete(String file) {
        File f1;
        boolean b1;
        f1 = new File(file);
        b1 = f1.delete();
        return;
    }
}
```

Постановка задачи

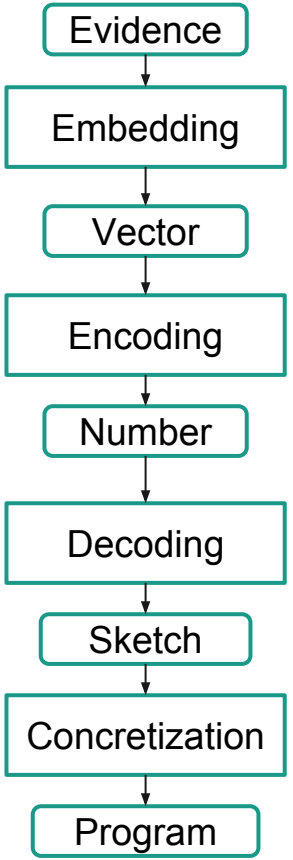
Цель: интегрировать подход BSL-синтезаторов с IntelliJ IDEA

Задачи:

- Разработать архитектуру конфигурируемого BSL-синтезатора
- Реализовать BSL-синтезатор на JVM платформе
- Создать интерфейс взаимодействия с пользователем
- Апробация реализации

Архитектура синтезатора

- Синтез представляется как набор преобразований
  - Вложение
  - Кодирование и декодирование
    - С помощью нейронной сети
  - Конкретизация скетча
- Набор алгоритмов для каждого преобразования

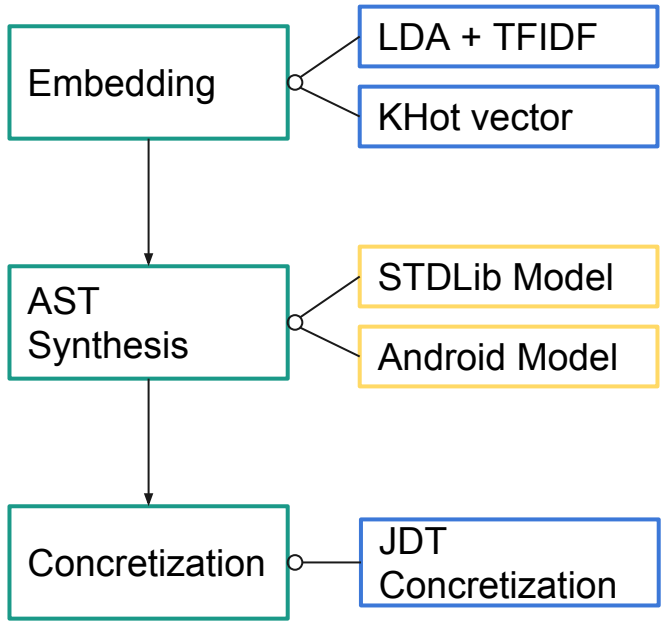


Модель и метамодель

- Метамодель --- набор алгоритмов и значений, параметризующих слои BSL-синтезатора
  - На данный момент есть Java STDlib и Android SDK метамодели
- Модель --- статистическая модель, полученная обучением конкретной метамодели

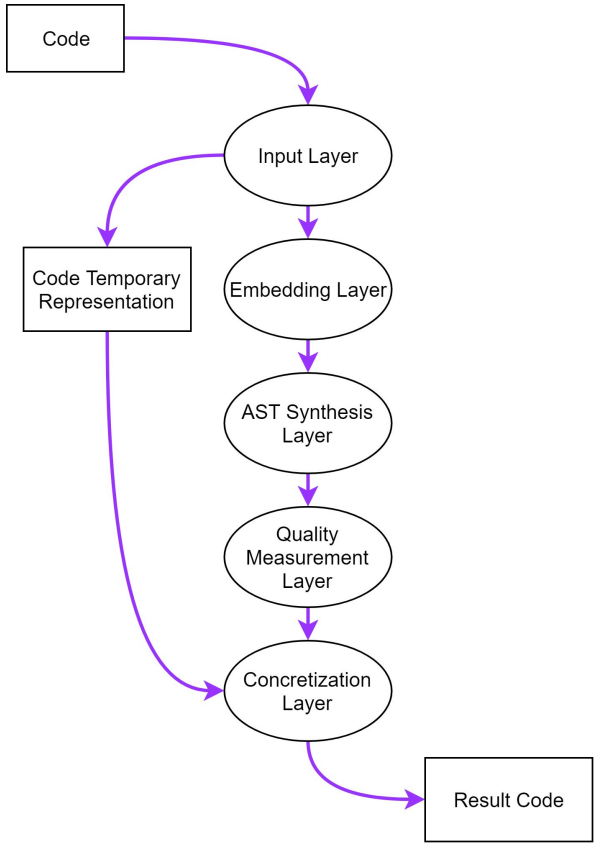
Конфигурируемый BSL-синтезатор

- Каждая метамодель в Bayou --- отдельное приложение
- Вынесем определение метамодели в конфигурацию
- Поддержим алгоритмы для всех метамоделей



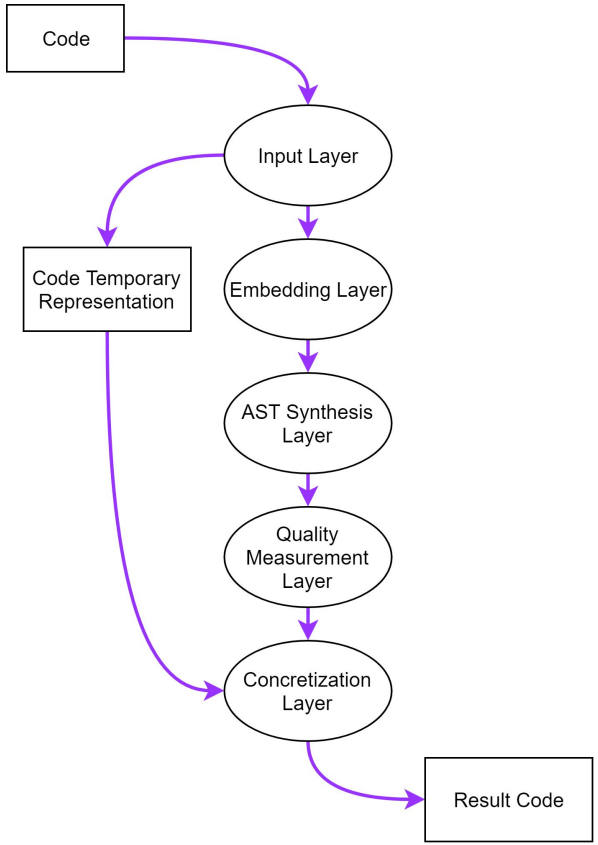
Реализация синтезатора

- Input Layer
  - Разбор код и извлечение свидетельств
- Embedding Layer
  - K-hot для Java STDlib
  - LDA + TF-IDF для Android SDK
- AST Synthesis Layer
  - BED



Реализация синтезатора

- Quality Measurement Layer
  - Дедупликация
  - Верификация
  - Ранжирование
- Concretization Layer
  - Перебор с помощью Eclipse JDT



Интерфейс пользователя

- Библиотека аннотаций
- Собственный язык

```
@SynthesizerType(“Stdlib”)
@ApiType(“File”)
@ApiCall(“delete”)
void delete(String file) {
}
```

```
/**
STDLIB
API:=delete
TYPE:=File
**/
void delete(String file) {
}
```

Пример

```
public class TestIO {
    /**
    STDLIB
    API:=delete
    TYPE:=File
    **/
    void delete(String file) {
    }
}
```

```
import java.io.File;

public class TestIO {
    void delete(String file) {
        File f1;
        boolean b1;
        f1 = new File(file);
        b1 = f1.delete();
        return;
    }
}
```

Пример

```
public class TestIO {
    /**
    STDLIB
    API:=readLine
    TYPE:=File
    **/
   void read(String file) {
    }
}
```

```
import java.io.FileNotFoundException;
import java.io.File;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class TestIO {
   void read(String file) {
      File f1;
      FileReader fr1;
      String s1;
      BufferedReader br1;
      try {
         f1 = new File(file);
         fr1 = new FileReader(f1);
         br1 = new BufferedReader(fr1);
         while ((s1 = br1.readLine()) != null) {}
         br1.close();
      } catch (FileNotFoundException _e) {
      } catch (IOException _e) {
      }
      return;
   }
}
```

Апробация

- 20 задач синтеза Java STDlib
- 20 задач синтеза Android SDK
- Получены эквивалентные результаты во всех случаях

Результаты

- Спроектирован конфигурируемый BSL-синтезатор
- Реализован конфигурируемый BSL-синтезатор
  - Поддержка Android SDK и Java STDLib
- Реализован пользовательский интерфейс
  - Аннотации и язык
- Проведена апробация подтверждающая эквивалентность полученного решения эталонной реализации Bayou
- Работа представлена на конференции SEIM 2018
  - Статья рекомендована для публикации в сборнике рецензируемом SCOPUS