

Разработка web- ориентированных CASE инструментариев автоматизации построения исходных кодов графоориентированных решателей

Студент: Неклюдов С.А.

Научный руководитель: Соколов А.П.

Введение

Актуальность

Актуальность данной работы обусловлена:

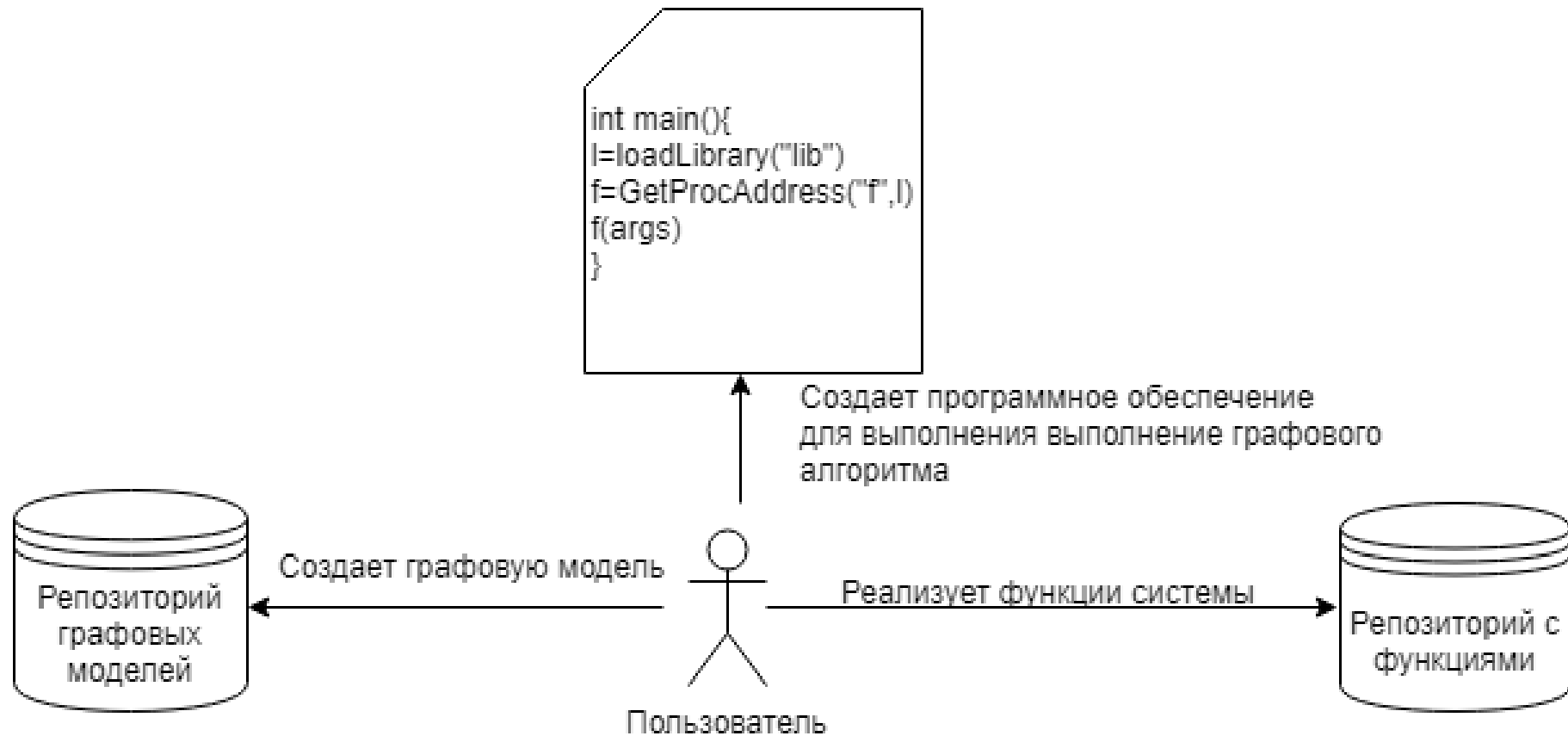
1. Разрастанием программных систем.
2. Желанием автоматизировать труд пользователей и разработчиков прикладного программного обеспечения.

Введение

Цели и задачи работы

- Цель: разработать программное обеспечение генерации исходного кода графоориентированных решателей PBC GCD
- Задачи:
 1. Доработка генератора форм подсистемы ввода-вывода comwps PBC GCD при помощи языка Python и фреймворка Django.
 2. Разработка функции холостого обхода графа в библиотеке comsdk.
 3. Разработка плагина на языке C++, обеспечивающего построение исходного кода решателя на основе модели алгоритма.

Сценарий использования PBC GCD



Архитектура подсистемы ввода - вывода

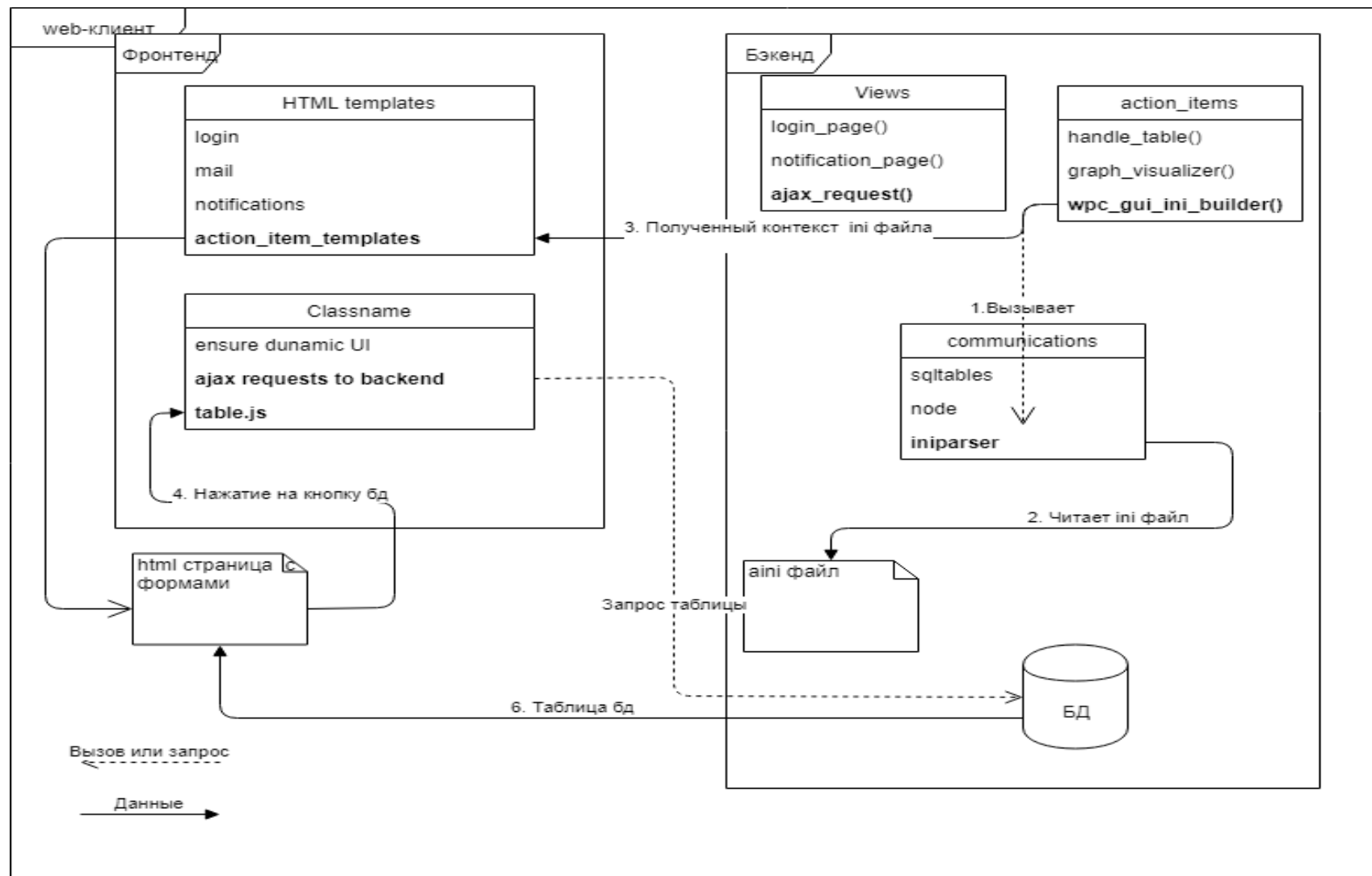
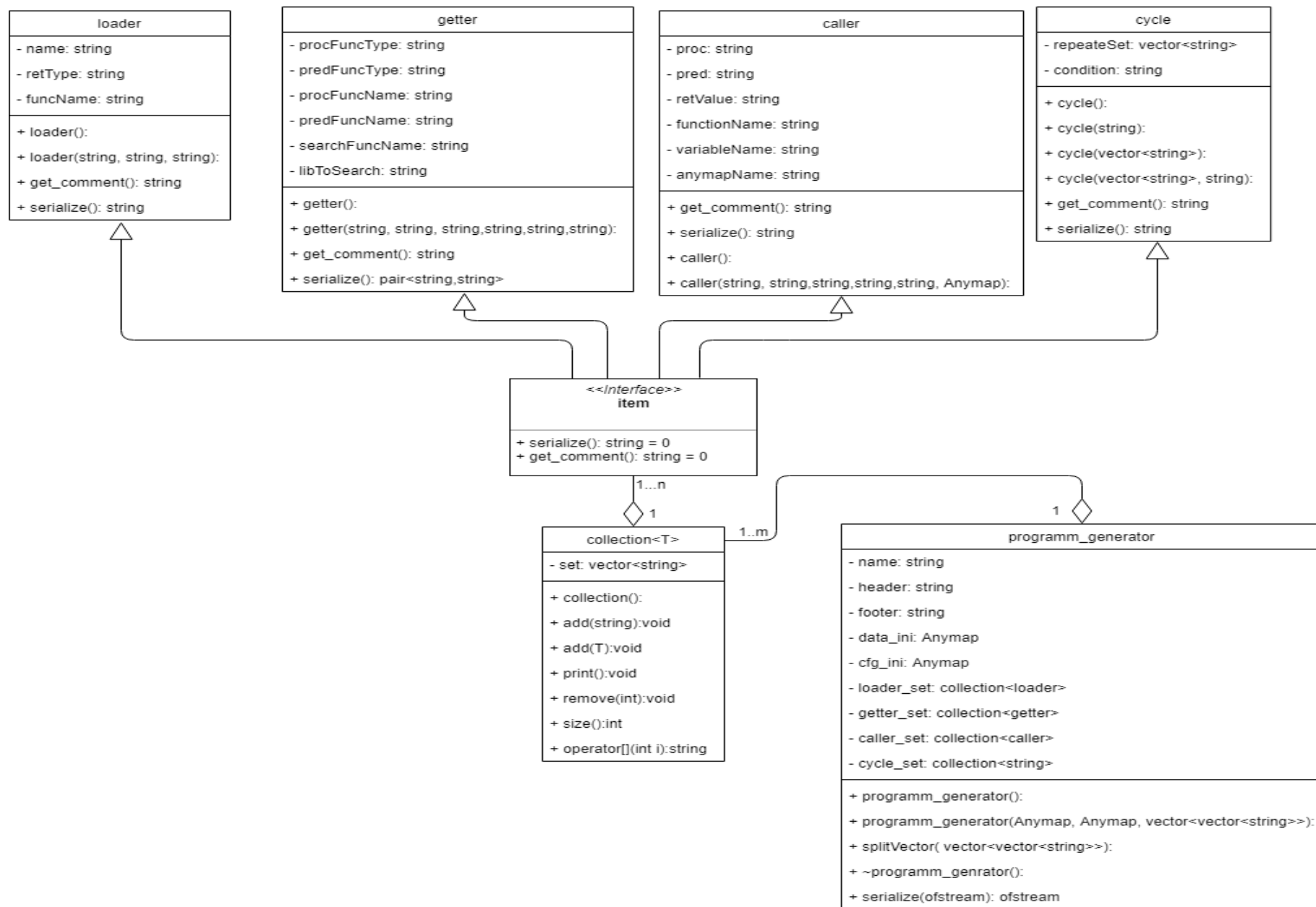
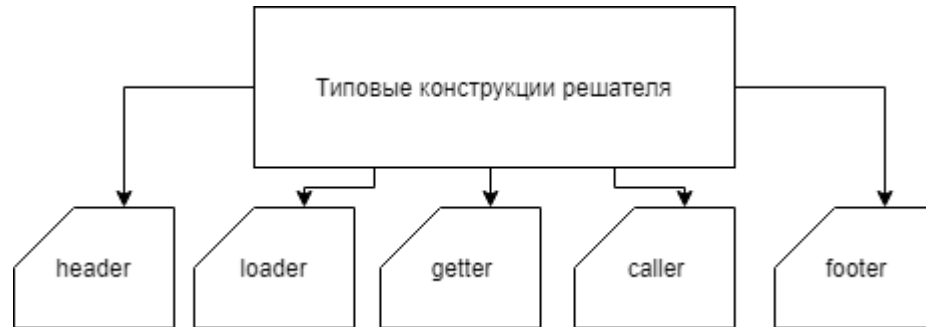


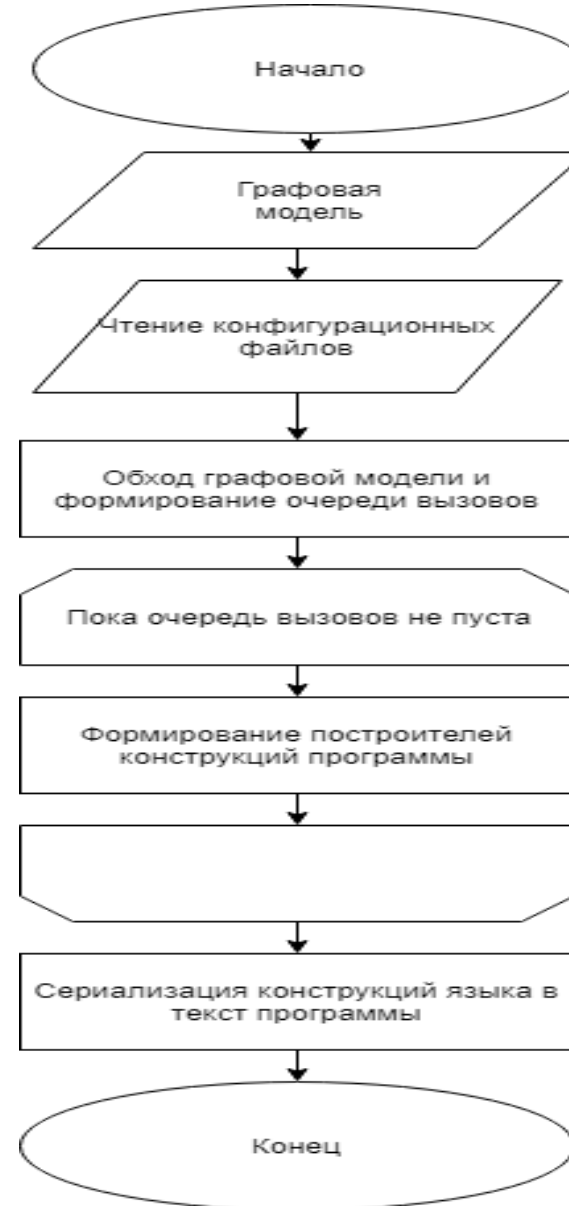
Диаграмма классов генератора кода



Принцип работы генератора



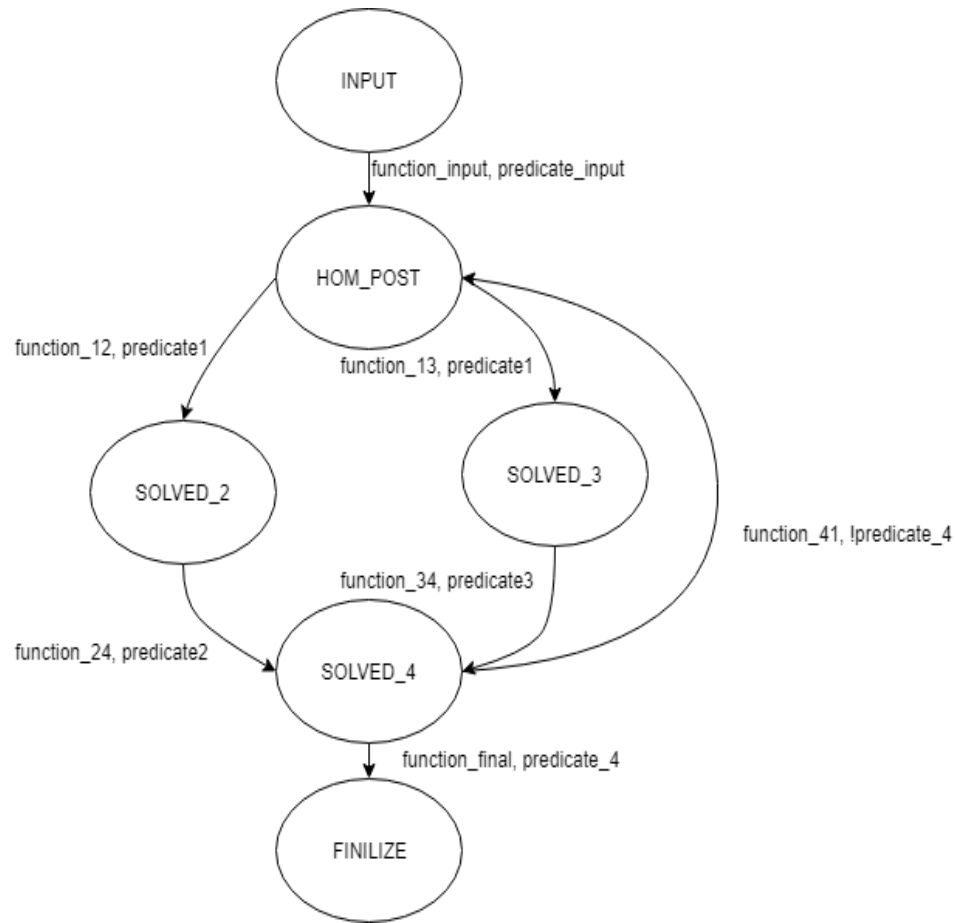
- Модель представляет собой текстовое описание графового алгоритма в формате обмена данными aDot
- Конфигурационные файлы представляются в формате alni
- Обход графовой модели реализован в библиотеке comsdk и позволяет находить циклические конструкции.
- Очередь вызовов является структурой типа `vector<vector<string>>`
- Сериализация – формирование строки типовой конструкции решателя



Сценарий использования РВС GCD с применением генератора



Тестирование



Тестовая модель

cfg.alni

System = Windows

```
Header = "#include <anymap.h>\n\ntypedef int\nprocessorFuncType(Anymap&);\n\ntypedef bool\npredicateFuncType(Anymap&);\n\ntemplate<processorFuncType tg, predicateFuncType tf>\n(const Anymap& p_m) >\n\nint F(Anymap& p_m) {\n\nreturn (tp(p_m)) ? tf(p_m) : tp(p_m);\n\n}\n\nint maint() {\n\nFooter = "}
```

Сгенерированный код

```
#include <anymap.h>

typedef int processorFuncType(AnyMap&);
typedef bool predicateFuncType(const AnyMap&);
template<processorFuncType* tf,predicateFuncType* tp>
int F(AnyMap& p_m)
{
    return (tp(p_m))?tf(p_m):tp(p_m);
}

int main(){
    Anymap input("input.txt");
    Anymap cfg("cfg.ini");
    HMODULE lib_name=LoadLibrary(L"name");
    processorFuncType *proc_function_input=(processorFuncType*)GetProcAddress(lib_name,"function_input");
    predicateFuncType *pred_predicate_input=(predicateFuncType *)GetProcAddress(lib_name, "predicate_input");
    ...
}
```

Заключение

- Созданное ПО позволит ускорить процесс разработки графоориентированных решателей PBC GCD.
- Разработанное ПО послужит ядром подсистемы генерации кода решателей PBC GCD.

Спасибо за внимание!