Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Верификация и анализ программ

Отчет по лабораторной работе №1

Выполнил студент гр. 3540901/21502 Джеус А.С.
Преподаватель Ицыксон В.М.
20 ноября 2022 г.

Санкт-Петербург
2022

Содержание

1. Техническое задание

Создать программу с графическим интерфейсом, предназначенную для построения и отображения графа потока управления (Control Flow Graph) для функции, написанной на языке Java. Построение CFG следует вести, исходя из соответствующего абстрактного синтаксического дерева (Abstract Syntax Tree), для получения которого допускается использование готовых библиотек.

2. Ход работы

2.1. Сторонние библиотеки

Для написания программы были использованы следующие библиотеки:

- JavaFX – создания графического интерфейса;
- JavaParser – построение AST;
- Graphviz-java – визуализация графа, описанного на языке DOT.

2.2. Метод решения

Для представления структуры данных, являющейся графом потока управления, был создан класс ControlFlowGraph, который хранит информацию об узлах графа и соединениях между ними, а также реализует функционал конкатенации двух графов в один и представления графа в виде его текстового описания на языке DOT.

Для построения CFG из AST был создан класс JavaCFGBuilder, который содержит следующие методы:

- String getCFGDescription(String code) – принимает введенный пользователем код на языке Java; с помощью библиотеки JavaParser формирует AST, которое затем передает на обработку в метод *methodProcessing*; возвращает описание искомого CFG на языке DOT;
- ControlFlowGraph methodProcessing(MethodDeclaration method) – принимает экземпляр класса MethodDeclaration, содержащий AST метода, для которого необходимо построить CFG; возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемый метод;

- ControlFlowGraph getNestedBlockCFG(List<Node> block, ControlFlowGraph superCFG) – возвращает экземпляр класса ControlFlowGraph, описывающий вложенный блок команд;
- ControlFlowGraph expressionProcessing(List<Node> expression) – возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемое выражение;
- ControlFlowGraph ifProcessing(IfStmt ifStmt) – возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемый условный оператор;
- ControlFlowGraph returnProcessing(ReturnStmt returnStmt) – возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемое выражение return;
- ControlFlowGraph forProcessing(ForStmt forStmt) – возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемый цикл for;
- ControlFlowGraph whileProcessing(WhileStmt whileStmt) – возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемый цикл while;
- ControlFlowGraph switchProcessing(SwitchStmt switchStmt) – возвращает экземпляр класса ControlFlowGraph, описывающий обрабатываемый оператор switch.

Графический интерфейс программы состоит из следующих элементов:

- Текстовое поле. Используется для ввода кода функции на языке Java;
- Кнопка «Copy DOT». При нажатии запускается построение CFG для введенного кода функции; результат в виде текстового описание графа на языке DOT помещается в буфер обмена.
- Кнопка «Build CFG». При нажатии запускается построение CFG для введенного кода функции; результат в виде текстового описание графа на языке DOT преобразуется в графическое изображение с помощью библиотеки Graphviz-java, затем полученная картинка открывается в стандартном просмотрщике изображений.

2.3. Код программы

Код программы находится в GitHub репозиторий проекта:

https://github.com/MickeyMouseMouse/JavaControlFlowGraphBuilder.

3. Скриншоты программы

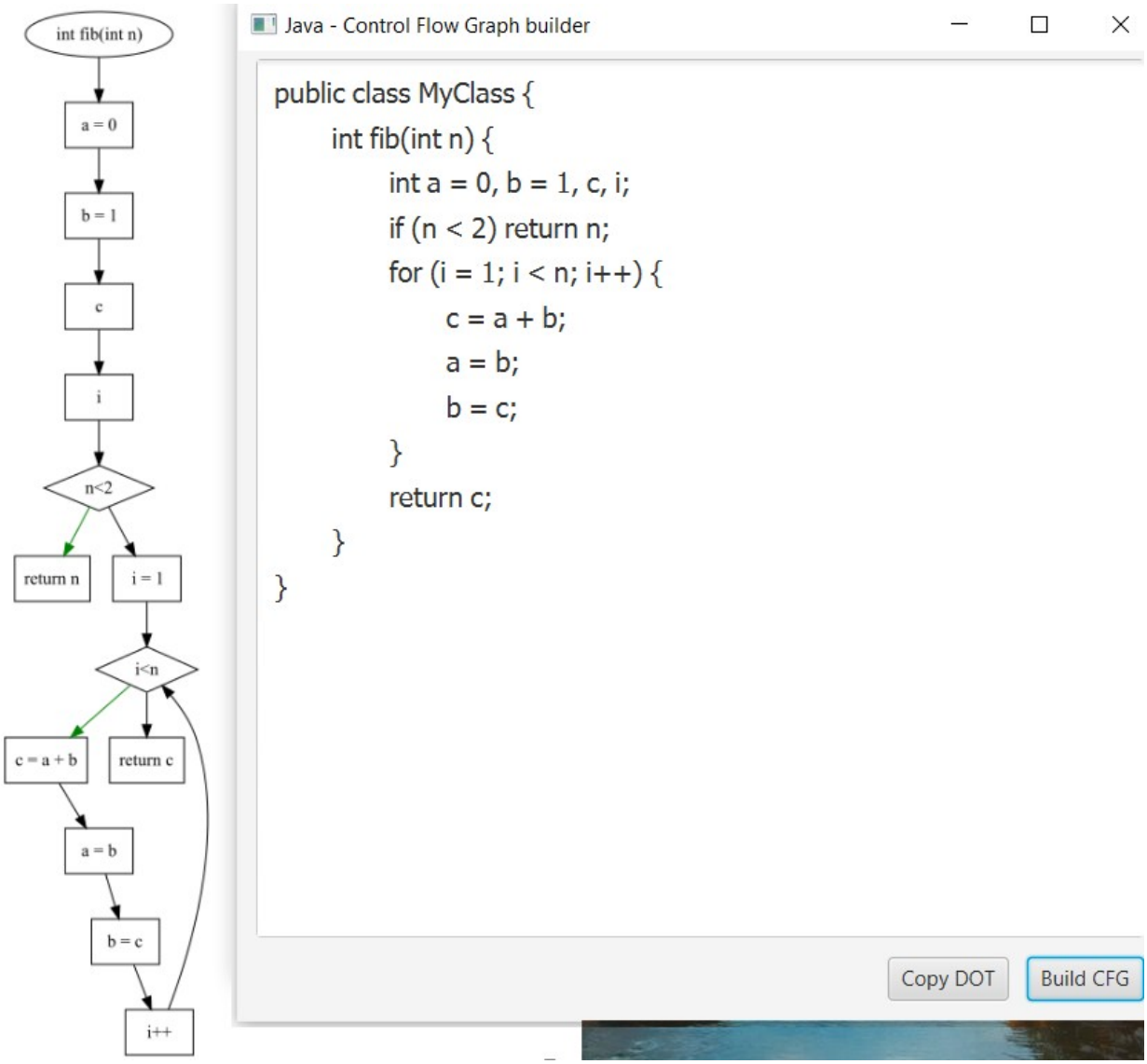

Рис.1.

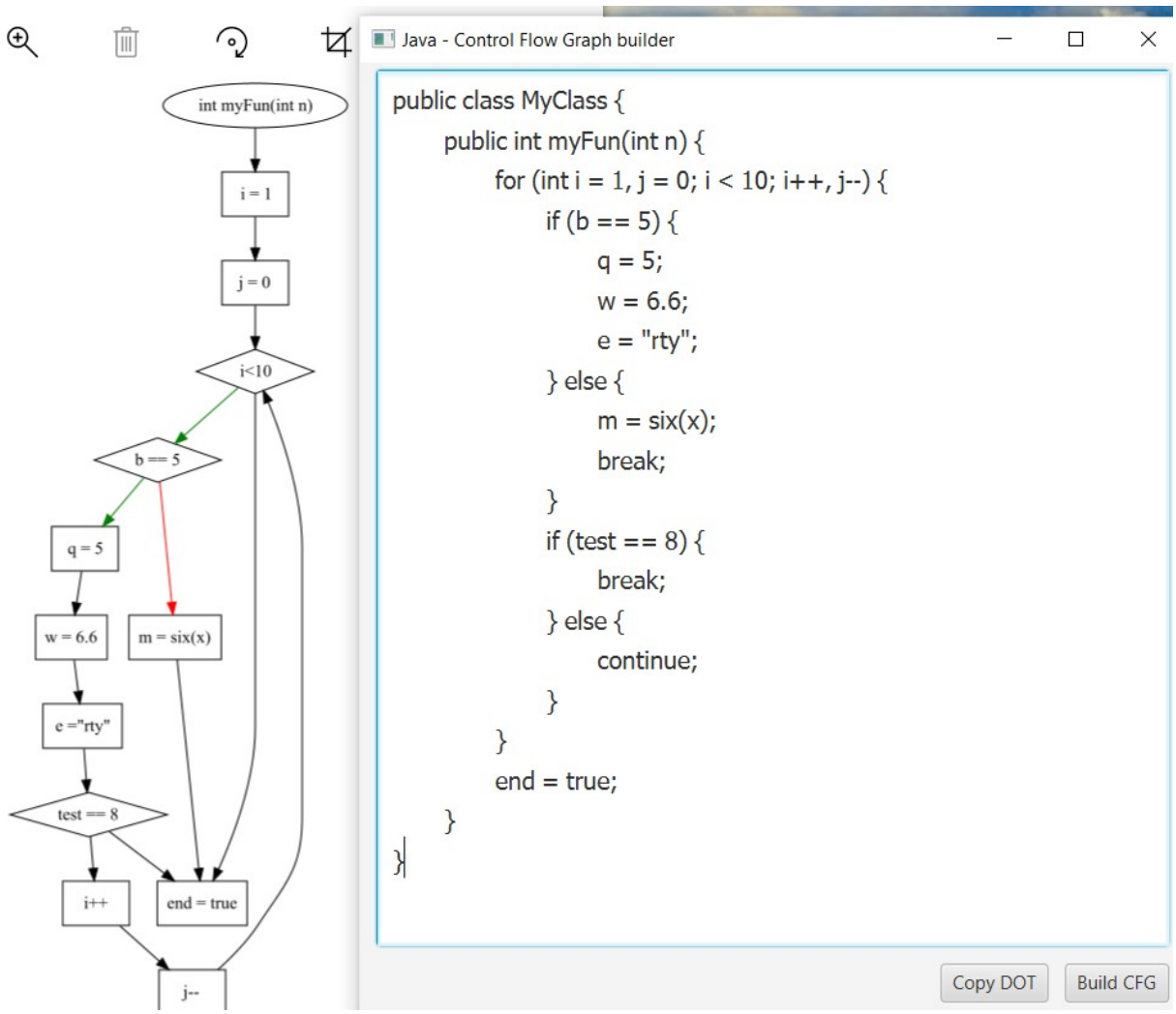
Java - Control Flow Graph builder
public class MyClass {
public int myFun(int n) {
for (int i = 1, j = 0; i < 10; i++, j--) {
if (b == 5) {
q = 5;
w = 6.6;
e = "rty";
} else {
m = six(x);
break;
}
if (test == 8) {
break;
} else {
continue;
}
}
end = true;
}
}
Copy DOT
Build CFG
int myFun(int n)
i = 1
j = 0
i<10
b == 5
q = 5
w = 6.6
m = six(x)
e ="rty"
test == 8
i++
end = true
j--

Рис.2.

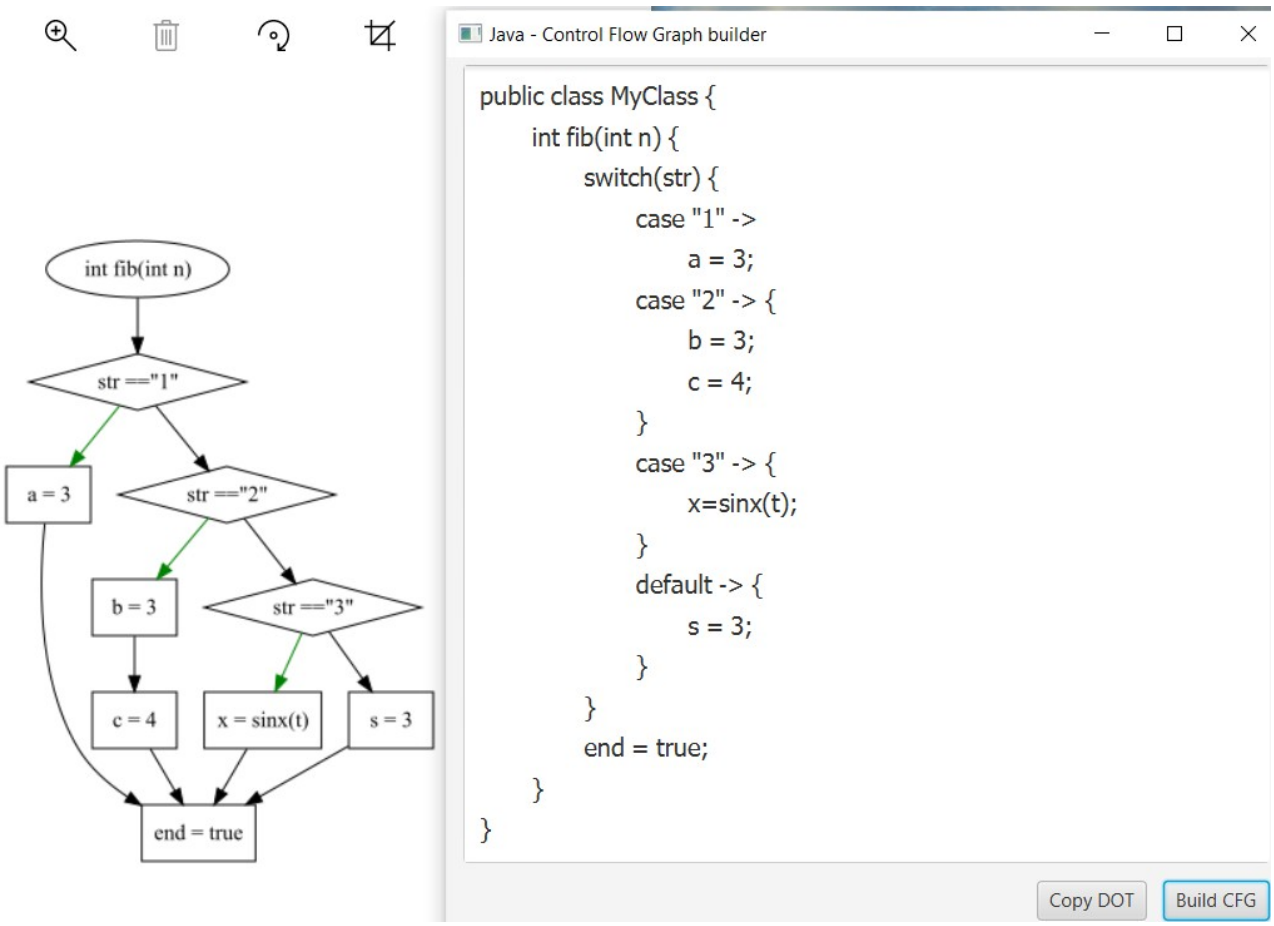
Java - Control Flow Graph builder
public class MyClass {
int fib(int n) {
switch(str) {
case "1" ->
a = 3;
case "2" -> {
b = 3;
c = 4;
}
case "3" -> {
x=sinx(t);
}
default -> {
s = 3;
}
}
end = true;
}
}
Copy DOT
Build CFG
int fib(int n)
str =="1"
a = 3
str =="2"
b = 3
str =="3"
c = 4
x = sinx(t)
s = 3
end = true

Рис.3.