



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Московский государственный технический университет имени Н.  
Э. Баумана

(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

---

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

---

## Отчёт по лабораторной работе № 7 по курсу «Анализ алгоритмов»

Тема Графовые модели

Студент Гаврилюк В. А.

Группа ИУ7-51Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Волкова Л. Л.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>2</b>
<b>1 Определение графовых моделей</b>	<b>3</b>
<b>2 Фрагмент кода</b>	<b>4</b>
<b>3 Графовые модели</b>	<b>5</b>
3.1 Информационный граф . . . . .	5
3.2 Информационная история . . . . .	6
3.3 Граф управления . . . . .	7
3.4 Операционная история . . . . .	8
<b>ЗАКЛЮЧЕНИЕ</b>	<b>10</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>11</b>

# ВВЕДЕНИЕ

**Цель работы** — получение навыка построения графовых моделей для анализа программного кода.

Для достижения цели необходимо выполнить следующие задачи:

- дать определение графовых моделей;
- выделить законченный фрагмент кода, который содержит вложенный цикл;
- по выделенному фрагменту кода выполнить построение четырех графов: информационного графа, информационной истории, графа управления и операционной истории;
- сделать вывод о применимости графовых моделей к задаче анализа программного кода.

# 1 Определение графовых моделей

Граф  $G$  можно записать как упорядоченную пару  $G = (V, E)$ , где:

- $V$  — непустое множество вершин;
- $E$  — множество рёбер, каждое из которых соединяет две вершины из  $V$ .

Графовая модель программы — ориентированный граф с конечным числом вершин и дуг, где элементы системы отображаются как вершины, а отношения (связи) между ними — как дуги [1]. Отношение в графовых моделях может быть информационным (по передаче данных) и операционным (по передаче управления). В таблице 1.1 представлено описание различных графовых моделей.

Таблица 1.1 – Описание графовых моделей

Название модели	Вершины	Дуги
Информационный граф	Операторы	Информационные отношения
Граф управления	Операторы	Операционные отношения
Операционная история	Срабатывания операторов	Операционные отношения
Информационная история	Срабатывания операторов	Информационные отношения

## 2 Фрагмент кода

На листинге 2.1 приведён рассматриваемый в рамках данной работы фрагмент кода. Подпрограмма из лабораторной работы №5 выполняет извлечение данных о рецептах из HTML документа.

Листинг 2.1 – Фрагмент кода

```
func (p Parser) extractIngredients(doc *goquery.Document)
    []*models.Ingredient {
    var ingredients []*models.Ingredient          // 1
    seen := make(map[string]struct{})             // 2
    containers := doc.Find("div.ingredient-list") // 3
    for i := 0; i < containers.Length(); i++ { // 4
        container := containers.Eq(i) // 5
        hiddenInputs := container.
            Find("input[type='hidden'][data-declensions]") // 6
        for j := 0; j < hiddenInputs.Length(); j++ { // 7
            input := hiddenInputs.Eq(j) // 8
            ingredientName := input.Parent().
                Find("span.recipe_ingredient_title").Text() // 9
            if _, ok := seen[ingredientName]; ingredientName != "" &&
                !ok { // 10
                seen[ingredientName] = struct{}{} // 11
                unit := p.extractUnit(input) // 12
                quantity := p.extractQuantity(input) // 13
                if unit == "" || quantity <= 0 { // 14
                    fmt.Printf("Warning: invalid ingredient data for %s\n",
                        ingredientName)// 15
                }
                ingredients = append(ingredients, &models.Ingredient{ //
                    16
                    Name:      ingredientName,
                    Unit:      unit,
                    Quantity: quantity,
                })
            }
        }
        fmt.Printf("Processed %d hidden inputs in container %d\n",
            hiddenInputs.Length(), i) // 17
    }
    return ingredients // 18
}
```

## 3 Графовые модели

### 3.1 Информационный граф

На рисунке 3.1 приведён информационный граф, построенный на основе рассмотренного фрагмента кода.

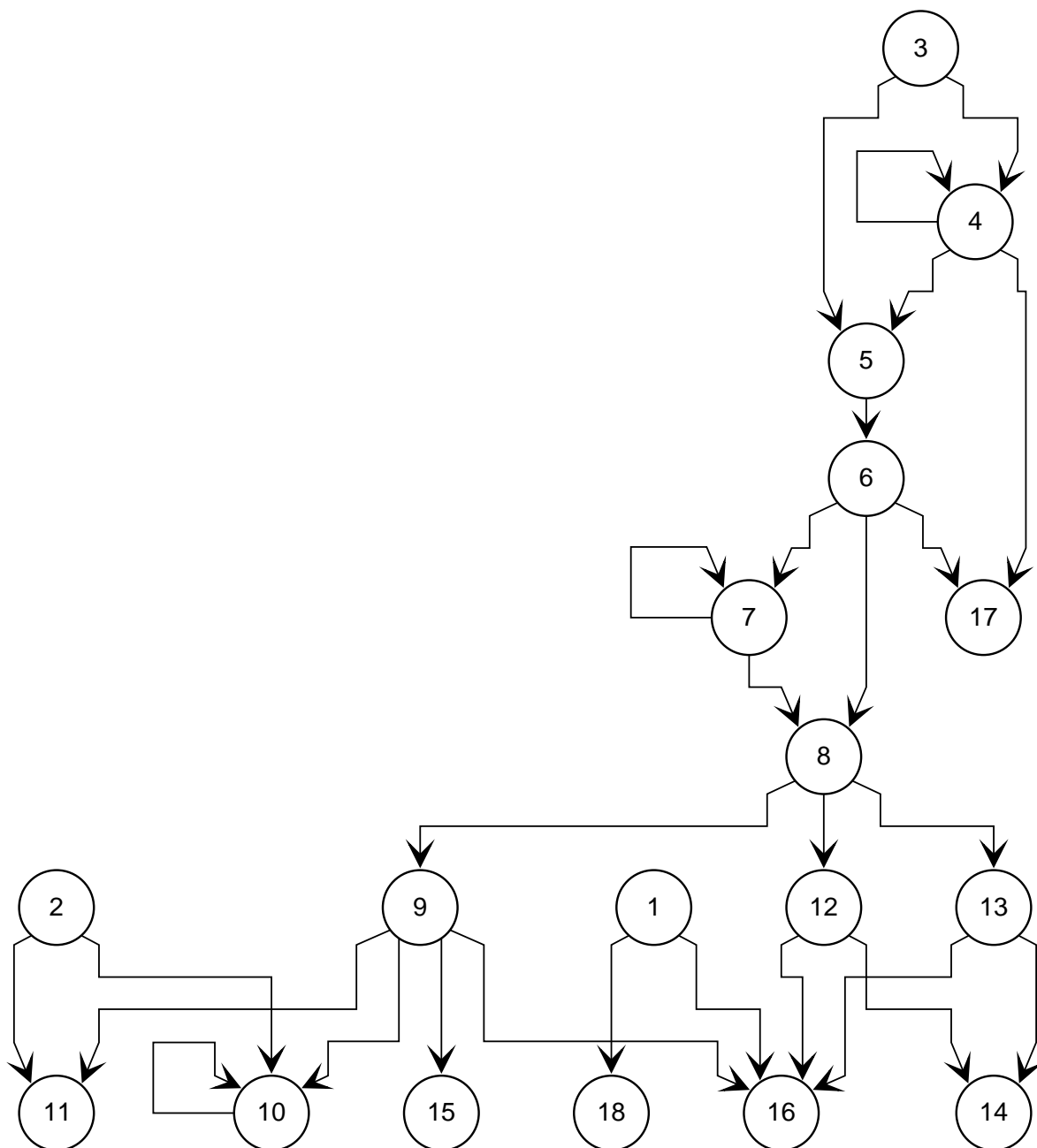


Рисунок 3.1 – Информационный граф

## 3.2 Информационная история

На рисунке 3.2 представлена информационная история, построенная на основе рассмотренного фрагмента кода.

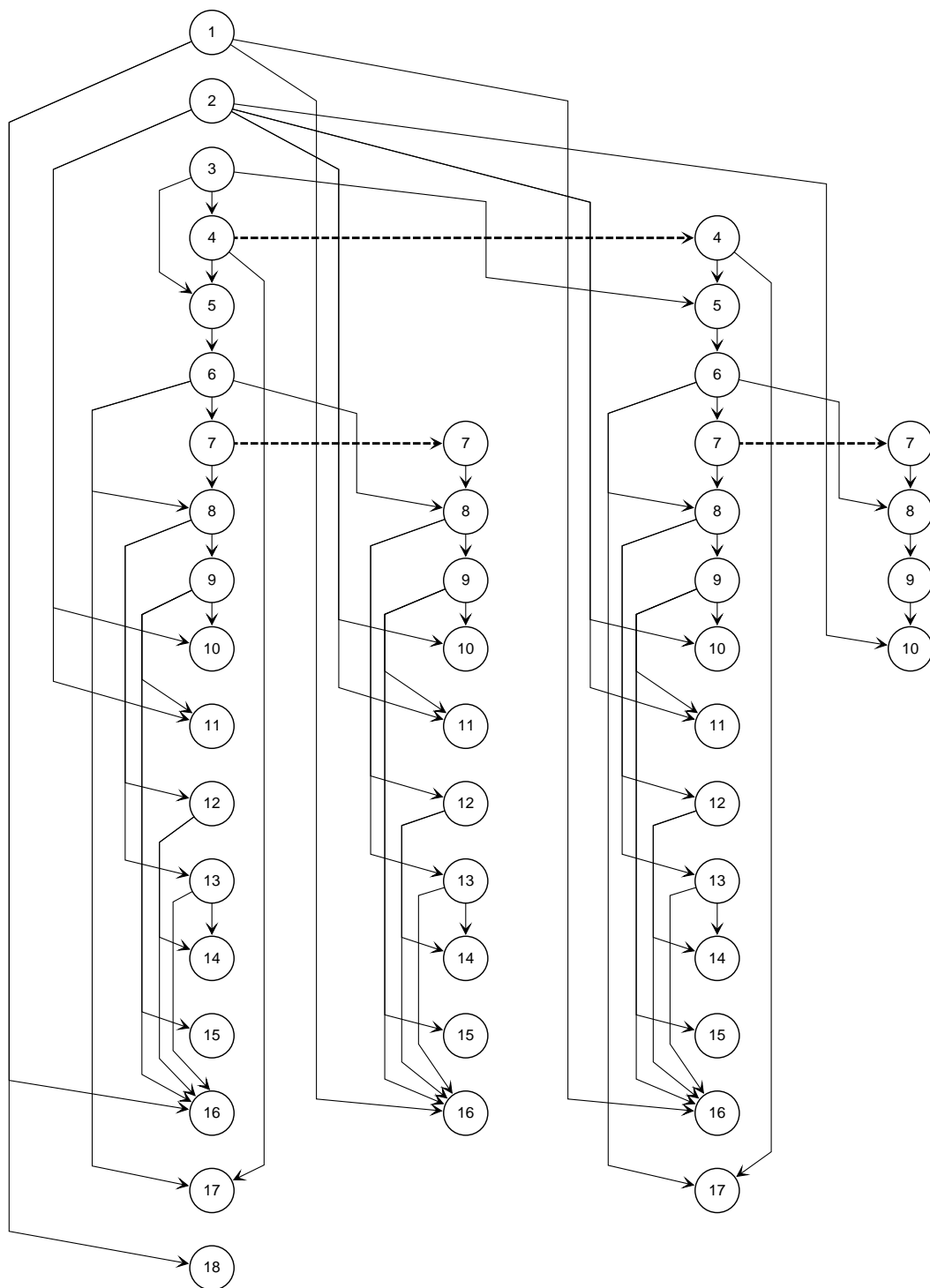


Рисунок 3.2 – Информационная история

### 3.3 Граф управления

На рисунке 3.3 продемонстрирован граф управления, построенный на основе рассмотренного фрагмента кода.

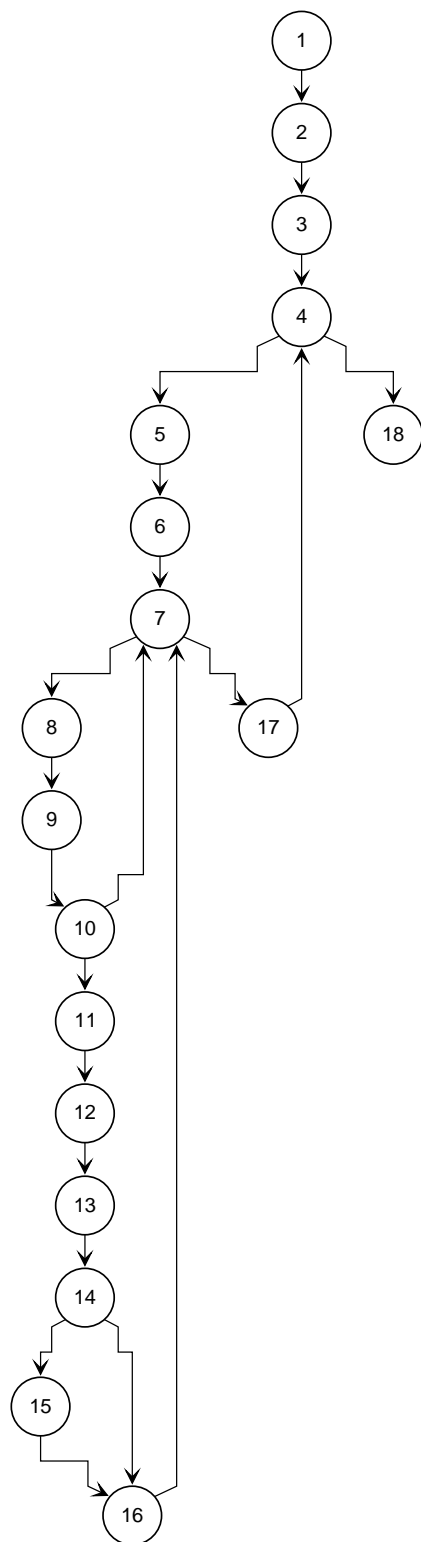


Рисунок 3.3 – Граф управления



### 3.4 Операционная история

На рисунке 3.4 изображена операционная история, построенная на основе рассмотренного фрагмента кода.

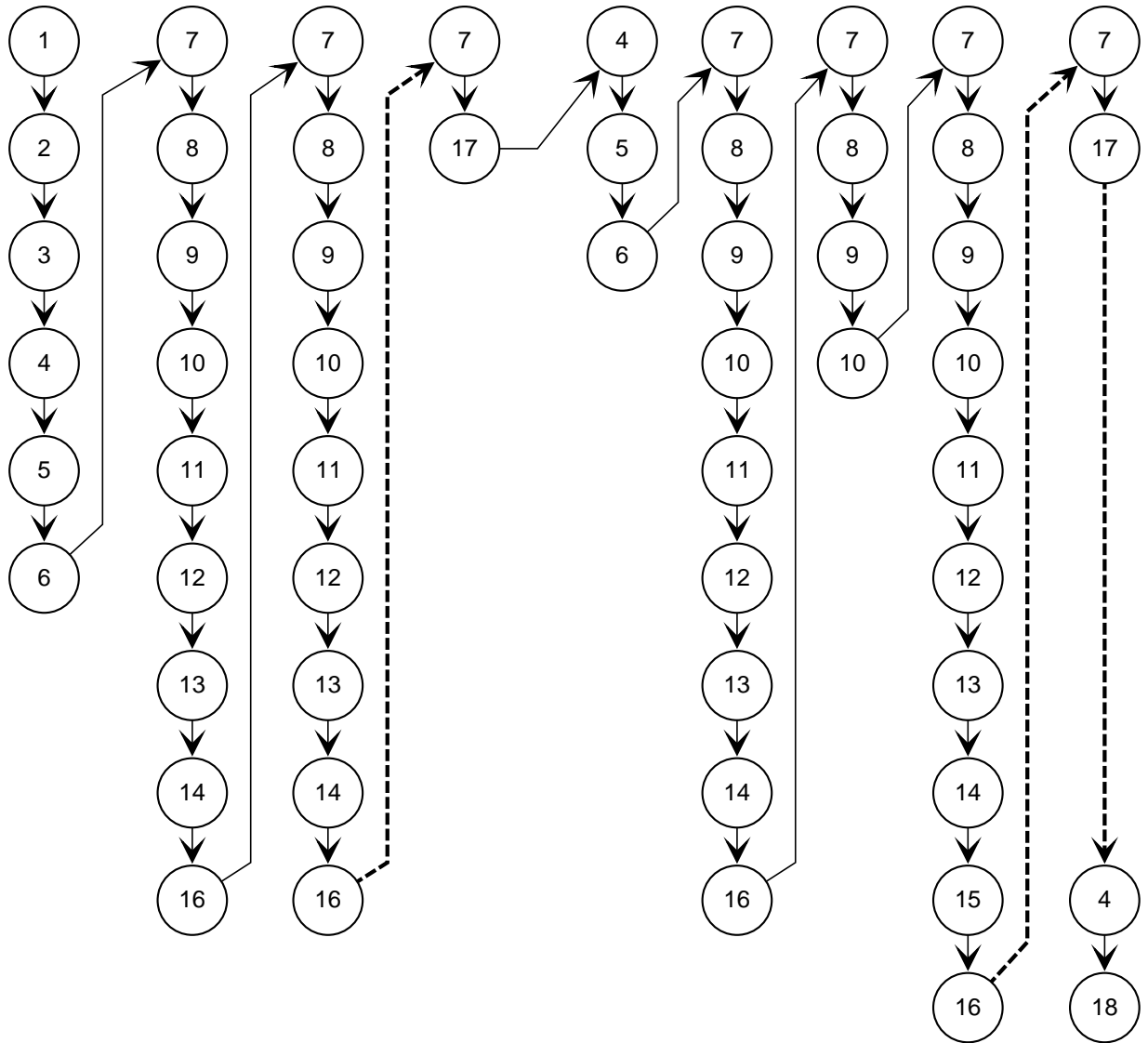


Рисунок 3.4 – Операционная история

## Вывод

Использование графов упрощает структурированное представление логики программы и потока данных. Информационный граф и информационная история демонстрируют, как данные передаются и преобразуются, что помогает в их отладке и оптимизации. Граф управления отображает логику ветвлений и циклов, позволяя проверить корректность тестирования. Операционная история подробно описывает пошаговое выполнение кода, что является

полезным свойством для диагностики ошибок и повышения эффективности выполнения программы.

Таким образом, графовые модели позволяют визуализировать информационные и операционные связи между объектами программного кода, облегчая их анализ.

## ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы были выполнены следующие задачи:

- было дано определение графовых моделей;
- выделен законченный фрагмент кода, который содержит вложенный цикл;
- по выделенному фрагменту кода были построены четыре графа: информационный граф, информационная история, граф управления и операционная история;
- сделан вывод о применимости графовых моделей к задаче анализа программного кода.

Все задачи работы выполнены. Цель достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Кирияничков В. А.* Методика построения операционных графовых моделей программ // Известия СПбГЭТУ «ЛЭТИ». — 2018. — Т. 2. — С. 16—21.