

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Казаков Матвей Алексеевич

**Методы построения визуализаторов
алгоритмов дискретной математики
на основе автоматного подхода**

**Специальность 05.13.06 – Автоматизация и управление
технологическими процессами и производствами
(образование)**

**Научный руководитель
доктор технических наук, профессор
Парфенов Владимир Глебович**



Федеральные государственные образовательные стандарты

- Повышают роль самостоятельной работы студентов
- Требуют увеличения объема интерактивных занятий

Визуализаторы алгоритмов

- Являются средством интерактивного и индивидуального обучения алгоритмам **в динамике**
- Упрощают изучение дискретной математики и программирования

Построение визуализаторов

- Требуется высоких трудозатрат
- Необходимость создания методов для снижения трудоемкости

Автоматный подход

- Позволяет явным образом выделять состояния – контрольные точки визуализации
- Может служить методологической основой при построении визуализаторов

Построение визуализаторов обеспечивает знакомство школьников и студентов 1-2 курсов с проектным подходом

Разработка методов построения визуализаторов алгоритмов дискретной математики на основе автоматного подхода – **актуальна**

Результаты, выносимые на защиту



1. Метод построения визуализаторов на основе конечных автоматов
2. Метод построения визуализаторов на основе системы взаимодействующих автоматов
3. Метод обеспечения простой анимации визуализаторов на основе автоматного подхода

Что такое визуализатор?



**Динамический набор иллюстраций,
отображающих**

1. Входные, выходные и служебные данные.
2. Процесс воздействия алгоритма на эти данные, процесс принятия решений.
3. Комментарии к каждому действию алгоритма.

**Визуализатор пошагово иллюстрирует
процесс исполнения алгоритма**

Подходы к построению визуализаторов



Существующий

- Традиционный (эвристический)

Предлагаемые

- Автоматный эвристический
- Автоматный формализованный

Разработанные методы на основе автоматного подхода



I. Базовый метод построения визуализаторов

II. Метод на основе системы взаимодействующих автоматов

- для декомпозиции сложных алгоритмов и отображения различных уровней подробности

III. Метод обеспечения простой анимации

- для поддержки плавного отображения изменений в динамических иллюстрациях

I. Базовый метод на примере «пузырьковой» сортировки (1)



1. Постановка задачи

Отсортировать массив чисел $a[1..n]$ по возрастанию

2. Решение задачи «пузырьковая сортировка»

Сортировка происходит за $n-1$ проход массива. На каждом проходе последовательно сравниваются пары соседних элементов. В случае нарушения порядка производится их обмен. При этом один из элементов («пузырек») перемещается на искомую позицию в конец массива («всплывает»).

3. Выбор визуализируемых данных

- массив чисел;*
- индексы элементов во время сравнения/обмена.*

I. Базовый метод на примере «пузырьковой» сортировки (2)



4. Анализ алгоритма для целей визуализации

Визуализатор должен отображать движение «пузырька»: производимые при этом сравнения, обмены и отсортированную часть массива.

5. Реализация алгоритма решения задачи на псевдокоде

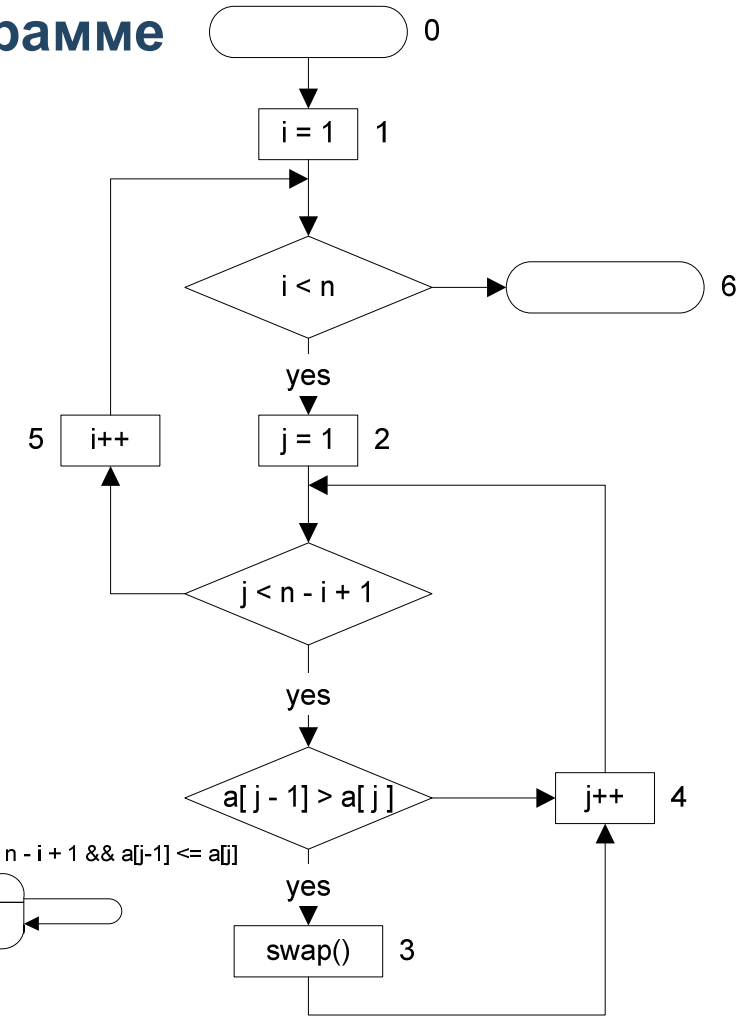
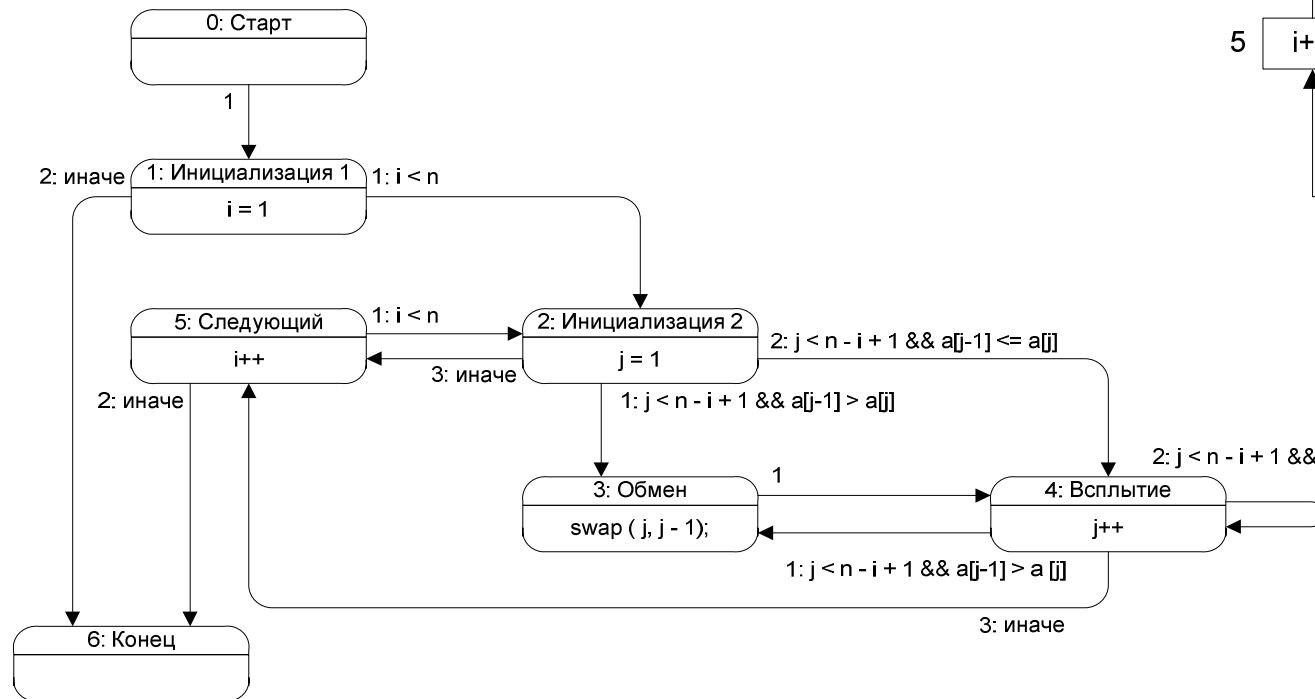
```
цикл для i = 1, 2, ..., n - 1
  цикл для j = 1, 2, ..., n - 1:
    если a[j-1] > a[j], то:
      обменять местами элементы a[j-1] и a[j]
```

6. Реализация алгоритма на выбранном языке программирования (Java), тестирование

```
for (int i = 1; i < n; i++) {
  for (int j = 1; j < n - i + 1; j++) {
    if (a[j - 1] > a[j]) {
      swap(j, j - 1); // всплытие
    }
  }
}
```

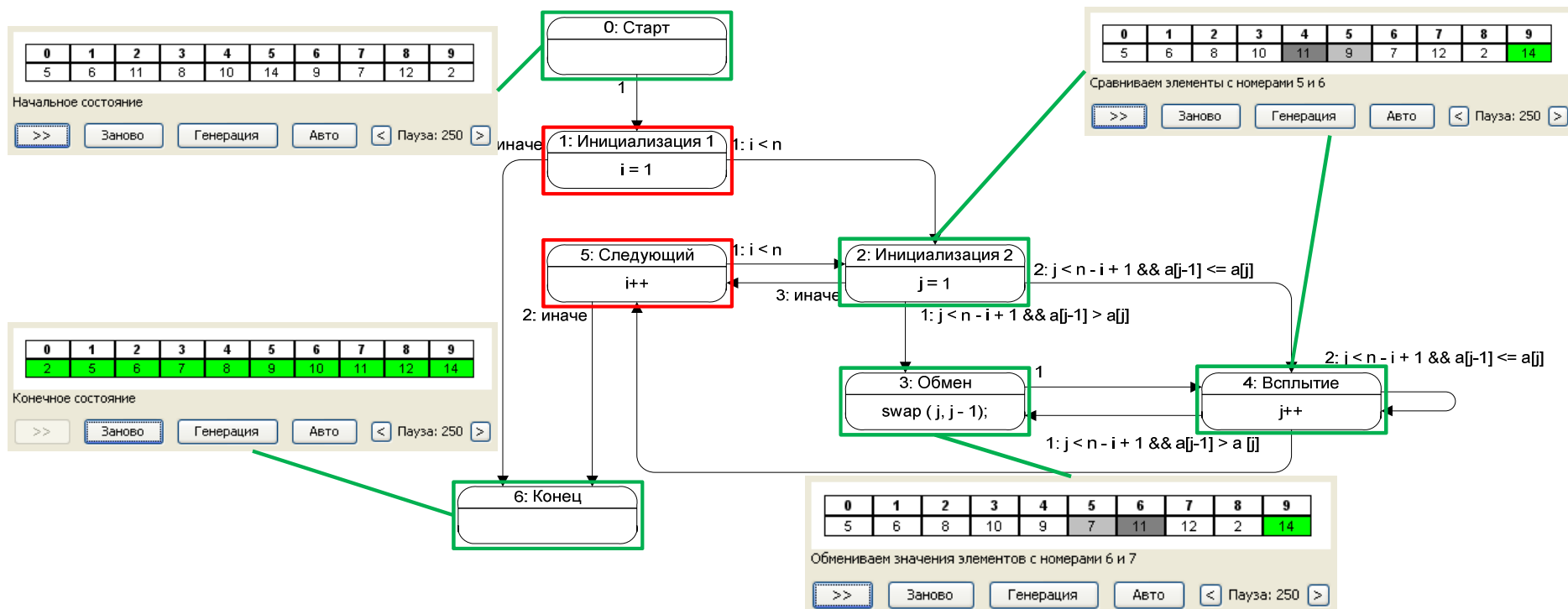
I. Базовый метод на примере «пузырьковой» сортировки (3)

7. Построение схемы алгоритма по программе
8. Преобразование схемы алгоритма в граф переходов автомата Мура на основе кодирования операторных вершин



I. Базовый метод на примере «пузырьковой» сортировки (4)

9. Формирование набора невизуализируемых состояний (красный цвет)
10. Выбор интерфейса визуализатора
11. Сопоставление иллюстраций и комментариев с состояниями автомата



I. Базовый метод на примере «пузырьковой» сортировки (5)



12. Разработка архитектуры программы визуализатора на базе паттерна «Модель-Вид-Контроллер»
13. Программная реализация визуализатора путем формального перехода от диаграмм автоматов к коду

Визуализатор «пузырьковой» сортировки

0	1	2	3	4	5	6	7	8	9
5	6	11	8	10	14	9	7	12	2

Начальное состояние

>> Заново Генерация **Стоп** < Пауза: 500 >

II. Метод на основе системы взаимодействующих автоматов (1)



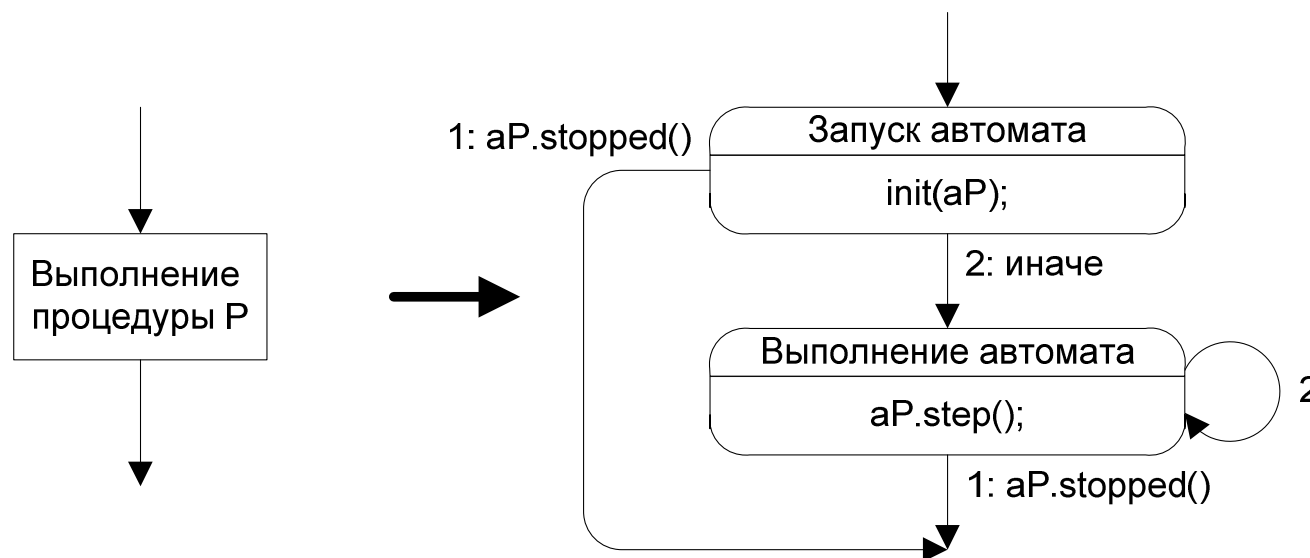
Для декомпозиции сложных алгоритмов
и отображения различных уровней подробности

- **Формирование системы автоматов**
 - Программа → процедуры
 - Процедура → схема алгоритма
 - Схема алгоритма → автомат
 - Организация взаимодействия автоматов
- **Построение визуализатора**
 - Выбор невизуализируемых состояний для системы автоматов
 - Формирование иллюстраций для состояний системы автоматов
 - Назначение уровней автоматов → многоуровневая визуализация

II. Метод на основе системы взаимодействующих автоматов (2)

Организация взаимодействия системы автоматов

- Запуск вложенного автомата
- Ожидание остановки вложенного автомата



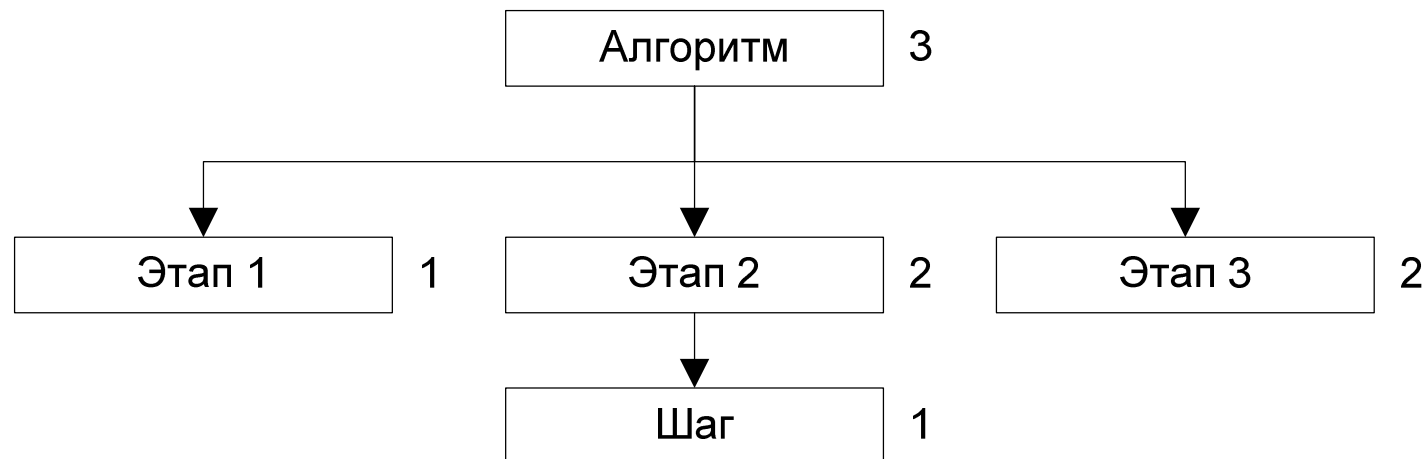
Управление системой автоматов происходит через главный автомат

II. Метод на основе системы взаимодействующих автоматов (3)



Многоуровневая визуализация

- Каждому автомату назначается уровень



- При визуализации на уровне N все иллюстрации, соответствующие более низким уровням, пропускаются

Сравнение методов на примере «задачи о рюкзаке»



Особенность	Базовый метод	Метод на основе системы автоматов
Число автоматов	1	6
Общее ее число состояний автоматов	16	32
Число строк кода для реализации автоматов	187	455

- Реализация визуализатора усложняется, но благодаря декомпозиции упрощается восприятие алгоритма учащимся

III. Метод обеспечения простой анимации (1)



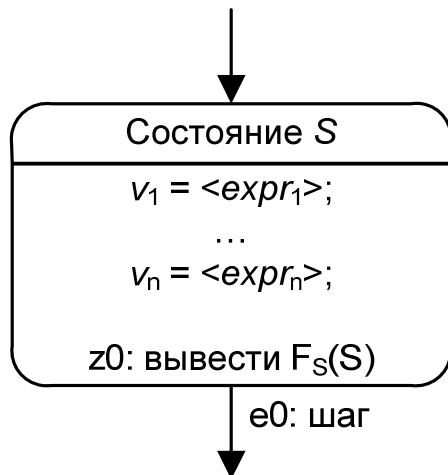
Поддержка плавного отображения изменений в динамических иллюстрациях

- **Выбор состояний для введения анимации**
- **Построение двух автоматов**
 - преобразованного управляющего автомата
 - анимационного автомата
- **Построение функции вывода динамических иллюстраций**

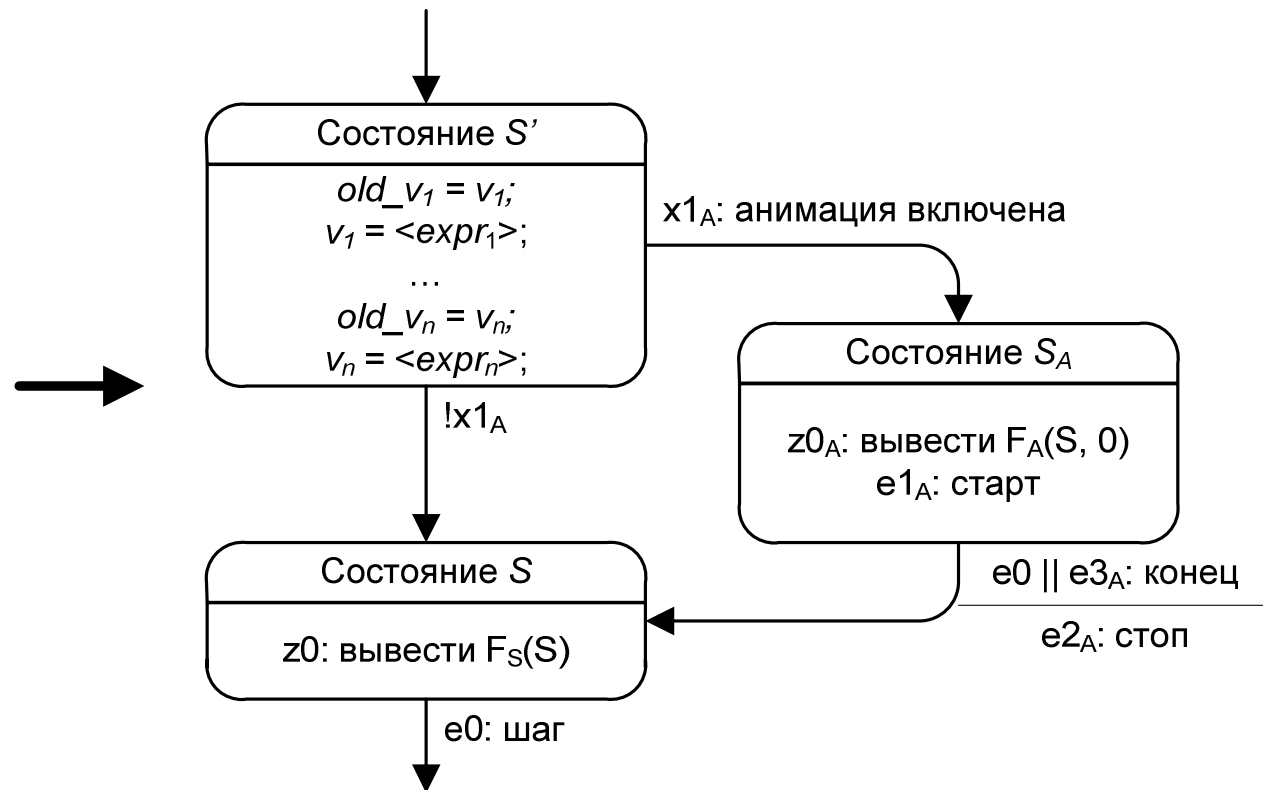
III. Метод обеспечения простой анимации (2)

Построение преобразованного управляющего автомата

Обычный визуализатор

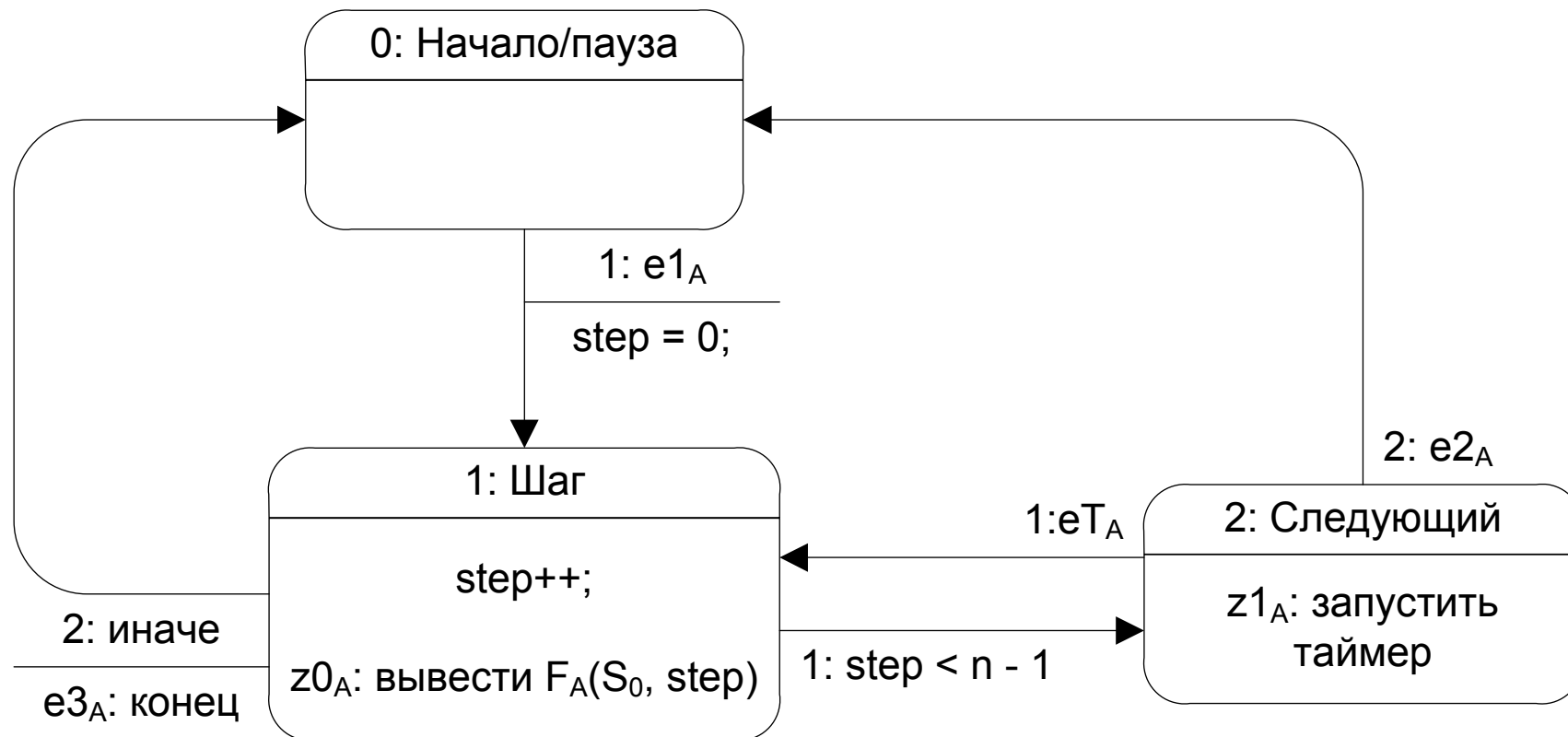


Визуализатор с поддержкой анимации



III. Метод обеспечения простой анимации (3)

Построение анимационного автомата



III. Метод обеспечения простой анимации (4)

Функция вывода динамических иллюстраций

Рассмотрим переход управляющего автомата

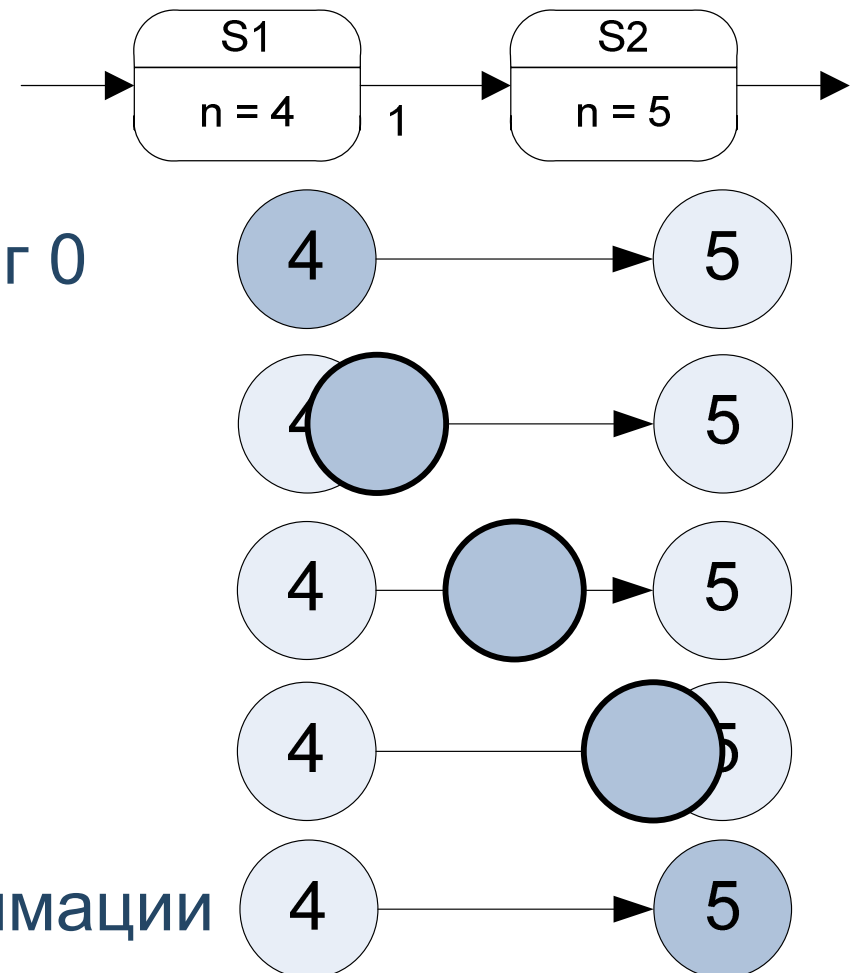
состояние S1 = состояние S2, шаг 0

состояние S2, шаг 1

состояние S2, шаг 5

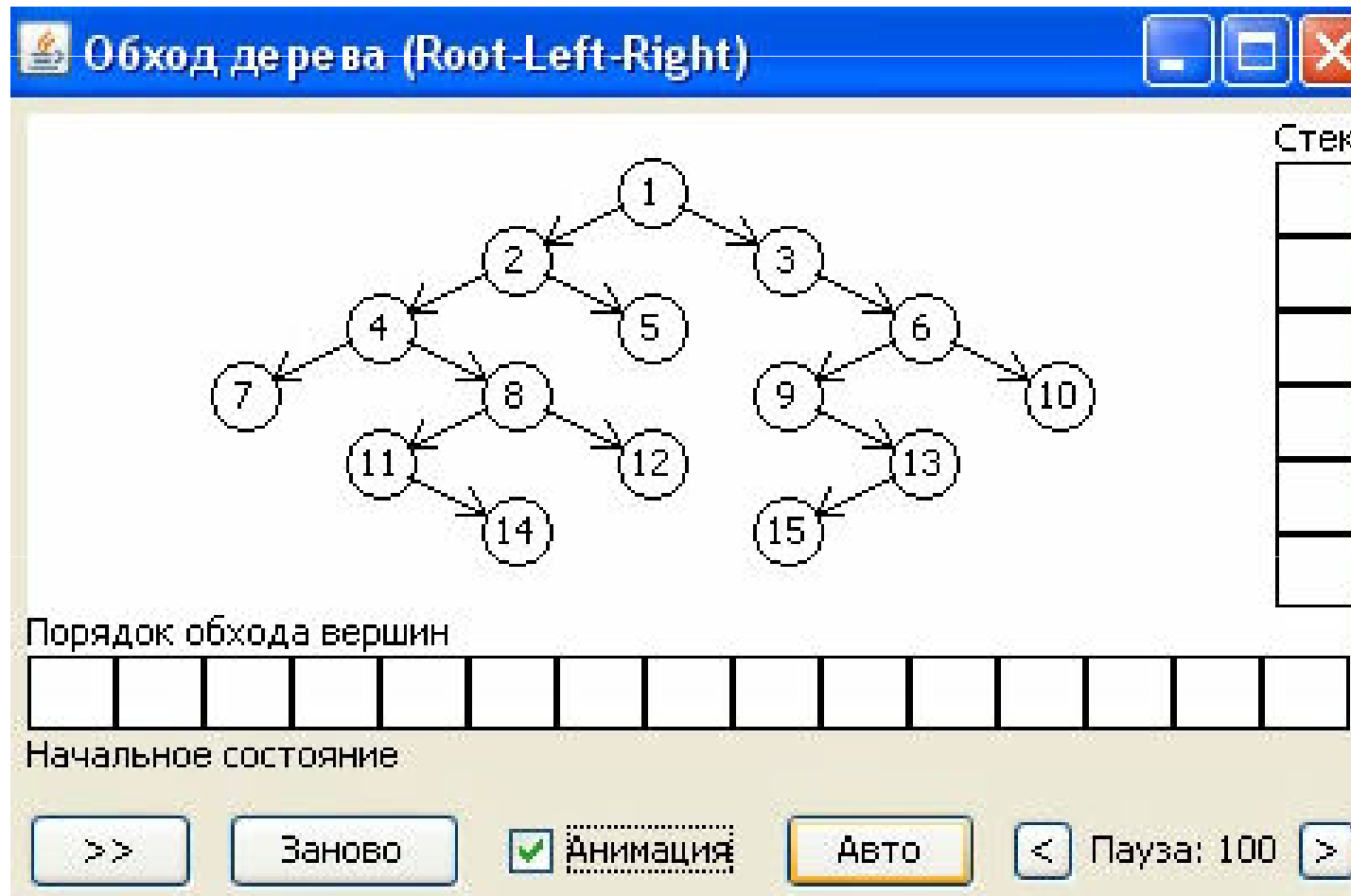
состояние S2, шаг 9

состояние S2, шаг 10 – конец анимации



Пример визуализатора с поддержкой анимации

Визуализатор алгоритма обхода дерева



- **Результаты диссертации получены в ходе выполнения семи НИР (РФФИ и Минобрнауки)**
- **Кафедра «Компьютерные технологии» СПбГУ ИТМО**
 - Курсы
 - Дискретная математика
 - Алгоритмы и структуры данных
 - Теория автоматов в программировании
 - Интернет-школа программирования
- **Специализированный учебно-научный центр МГУ**
 - Курс информатики

Эффективность предложенных методов



Подход	Время студента, час		Время преподавателя, час
	Простые алгоритмы	Сложные алгоритмы	
Традиционный (эвристический)	80-120	120-180	8-10
Автоматный эвристический	10-20	40-80	2-3
Автоматный формализованный	2-5	20-40	менее одного часа

Пример проекта построения визуализатора (1)



Визуализатор «Сортировка подсчетом». Авторы В.А. Добровольский, А.В. Степук, 2004 год

<http://is.ifmo.ru/vis/countsort/>

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
Кафедра «Технологии программирования»

Главная / Визуализаторы / Сортировка подсчетом
(версия для печати)

Сортировка подсчетом

© В.А. Добровольский, А.В. Степук

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Проектная документация в формате PDF
 Исходные коды

Визуализатор (online)

Аннотация

В работах, приведенных на сайте <http://is.ifmo.ru>, показана целесообразность применения автоматного программирования для построения визуализаторов.

Данная работа подтверждает успешность этого направления. В работе показана возможность визуализации еще одного типа алгоритмов.

Сортировка подсчетом — оригинальный вид сортировки массивов положительных чисел, работающей за линейное время.

Меню

- Главная
- Новости
- Автографы
- Автоматы
- Автоматное программирование в школе
- Беллетристика
- Важное
- Верификация
- Визуализаторы**
- Внедрение автоматного программирования
- Генетические алгоритмы
- Геном
- Движение
- Дипломы
- Диссертации
- Инновации
- Искусственный интеллект
- Искусство
- Клеточные автоматы
- Книги
- Конференции
- Люди
- Мотивация
- Мысли
- Наука
- Нейронные сети
- О нас

Визуализатор

Сортировка подсчетом

Сортировка подсчетом позволяет отсортировать за линейное время последовательность, в которой каждый элемент — целое положительное число в известном диапазоне (не превосходящее заранее известного k). В реализации визуализатора k равно десяти.

Идея алгоритма состоит в том, чтобы для каждого элемента x предварительно подсчитать, сколько элементов входной последовательности меньше x . После этого x записывается напрямую в выходной массив в соответствии с этим числом. Например, если семнадцать элементов входного массива меньше x , то в выходном массиве x должен быть записан на место с номером восемнадцать.

Используются следующие обозначения:
 $A[1..n]$ — входная последовательность;
 $C[1..k]$ — вспомогательный массив из k элементов;
 $B[1..n]$ — выходная отсортированная последовательность.

Работа алгоритма заключается в следующем.
В элемент $C[i]$ заносится количество элементов массива A , равных i . Затем находится частичные суммы последовательности $C[1], C[2], \dots, C[k]$, каждая из которых является количеством элементов, не превосходящих i . После этого каждый элемент $A[i]$ из входного массива помещается в выходной массив B на позицию, равную значению элемента $C[A[i]]$.

Массив отсортирован. Работа алгоритма завершена.

Авторы: В.А. Добровольский, А.В. Степук
Руководитель: А.А. Шальго

Санкт-Петербургский государственный университет информационных технологий, механики и оптики.
Кафедра "Компьютерные технологии"

[Исходные коды](#)

Copyright 2004 by [Vladimir Dobrovolski](#)

Пример проекта построения визуализатора (2)



Проектная документация

Оглавление	
Введение	3
1. Постановка задачи	3
2. Решение задачи	3
3. Выбор визуализируемых переменных	4
4. Анализ алгоритма для его визуализации	4
5. Алгоритм решения задачи	4
6. Реализация алгоритма на языке <i>Java</i>	4
7. Построение схемы алгоритма по программе	5
8. Преобразование схемы алгоритма в граф переходов автомата Мили	7
9. Формирование набора невизуализируемых переходов	8
10. Выбор интерфейса визуализатора	9
11. Сопоставление иллюстраций и комментариев с состояниями автомата	10
12. Архитектура программы визуализатора	13
13. Программная реализация визуализатора	13
Заключение	15
Литература	15
Приложение. Исходный текст визуализатора	15

Введение

В работах, приведенных на сайте [1], показана целесообразность применения автоматного программирования для построения визуализаторов. Визуализатор отображает пошаговое выполнение алгоритма, передвигаясь от состояния к состоянию. Такая логика соответствует работе автомата.

Целью данной работы является построение визуализатора алгоритма сортировки подсчетом на основе автоматного программирования [2].

Заключение

Использование автоматного подхода для создания визуализаторов была развита в работах на сайте [1].

Данная работа подтверждает успешность этого направления.

Литература

1. Сайт кафедры «Технологии программирования». <http://is.ifmo.ru>
2. Казаков М.А., Шалыто А.А. Использование автоматного программирования для реализации визуализаторов //Компьютерные инструменты в образовании. 2004. № 2.

- Автоматный подход впервые применен к решению задачи построения визуализаторов алгоритмов дискретной математики, которые до этого строились эвристически.
- Разработанные методы формализуют переход от алгоритма к его визуализатору, позволяя реализовать многоуровневую визуализацию и анимацию.
- Разработанные методы успешно внедрены в учебный процесс. Они повысили эффективность создания визуализаторов и обеспечили эффективную поддержку проектного подхода.

Основные достижения



- По теме диссертации опубликовано 21 работа, в том числе пять из перечня ВАК
- Премия Правительства РФ за 2008 год в области образования «Инновационная система поиска и подготовки высококвалифицированных специалистов в области производства программного обеспечения на основе проектного и соревновательного подходов»