

Санкт-Петербургский государственный университет
информационных технологий, механики и оптики

Корнеев Георгий Александрович

**Автоматизация построения
визуализаторов алгоритмов на
основе автоматного подхода**

Специальность 05.13.12. «Системы автоматизированного
проектирования» (приборостроение)

Научный руководитель
доктор технических наук, профессор
Шальто Анатолий Абрамович

Актуальность

- Приборы и устройства усложняются
- Усложнение алгоритмов, применяемых при проектировании
- Усложнение обучению алгоритмам
- Визуализаторы алгоритмов позволяют упростить обучение алгоритмам дискретной математики

Новые результаты, выносимые на защиту

- Метод формального преобразования программ в систему взаимодействующих конечных автоматов (КА)
- Метод формального преобразования программ в систему взаимодействующих КА, обеспечивающую обратимую трассировку исходной программы в прямом и обратном направлениях
- Язык описания визуализаторов алгоритмов дискретной математики
- Технология, автоматизирующая построение визуализаторов алгоритмов дискретной математики

Требования к визуализаторам алгоритмов

1. Интерактивность
2. Двухнаправленность
3. История
4. Автоматический режим работы
5. Отображение хода выполнения программ
6. Комментарии
7. Простота использования
8. Свободный доступ
9. Платформонезависимость
10. Автономность

Анализ систем визуализации

Система визуализации	1	2	3	4	5	6	7	8	9	10
Animal	-	+	±	±	±	+	+	+	+	+
JAWAA	-	-	±	±	±	+	+	+	-	-
BALSA	-	±	+	+	±	±	-	-	-	-
Tango	-	+	+	-	-	-	-	-	-	+
XTango	-	+	+	-	-	-	-	-	-	+
Jeliot	+	-	-	-	+	-	+	+	-	-
Polka	+	-	-	±	±	-	+	-	+	+
Leonardo	+	+	+	+	-	-	-	-	-	+

Метод преобразования программ в систему взаимодействующих КА

1. Преобразование программы к приведенной форме
2. Построение модели данных
3. Построение системы взаимодействующих конечных автоматов

Преобразование к приведенной форме

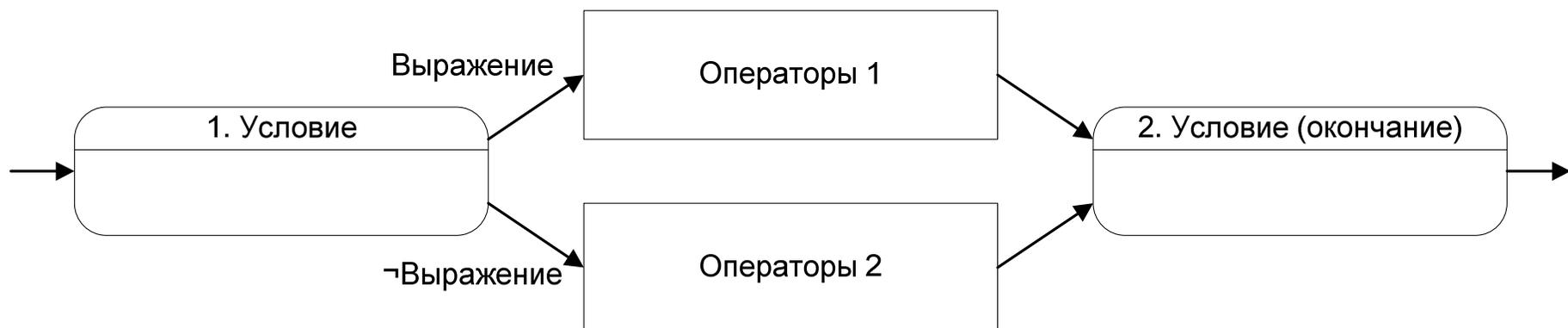
- Допустимые операторы
 - оператор присваивания
 - последовательность операторов
 - оператор ветвления
 - полный оператор ветвления
 - цикл с предусловием
 - вызов процедуры
- Недопустимые операторы заменяются на последовательности допустимых

Построение модели данных

- В модель данных выносятся
 - глобальные переменные
 - локальные переменные
 - аргументы процедур
 - возвращаемые значения функций
- В результате программа разбивается на вычислительную и управляющие части

Построение системы взаимодействующих КА

- Построение производится от вложенных операторов к внешним
- Фрагмент автомата — набор состояний и переходов с одним входом и выходом
- Замыкание фрагментов автоматов

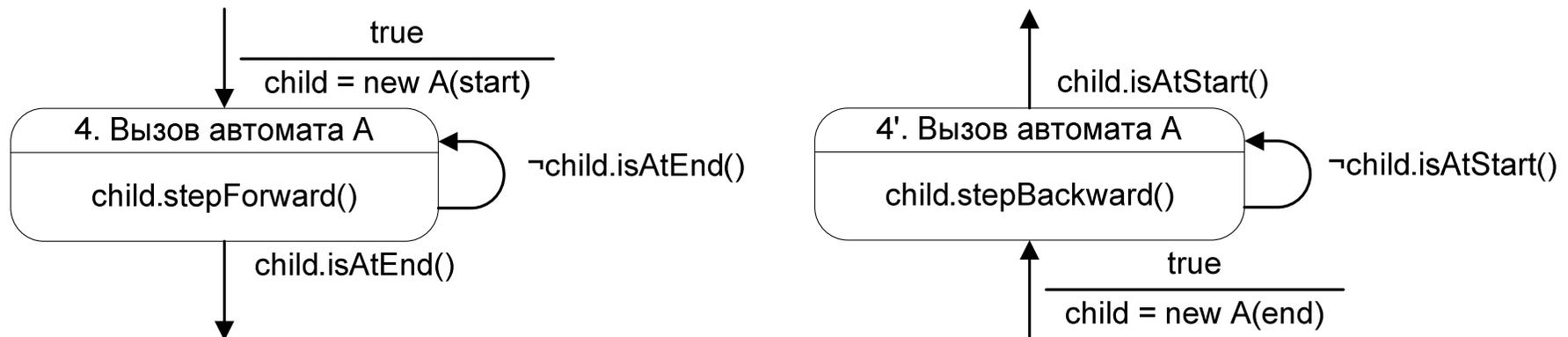


Построение системы КА с поддержкой обратимой трассировки

- Для каждой процедуры строится пара автоматов: прямой и обратный, имеющие общее множество состояний
- Прямой автомат обеспечивает трассировку программы в прямом направлении и сохраняет данные, необходимые при обратном проходе
- Обратный автомат обеспечивает трассировку программы в обратном направлении

Рекурсия

- Экземпляр автомата — соответствует экземпляру процедуры
 - Создание в начальном и конечном состояниях
 - Продвижение на один шаг вперед и назад
 - Определение достижения конечного и начального состояний



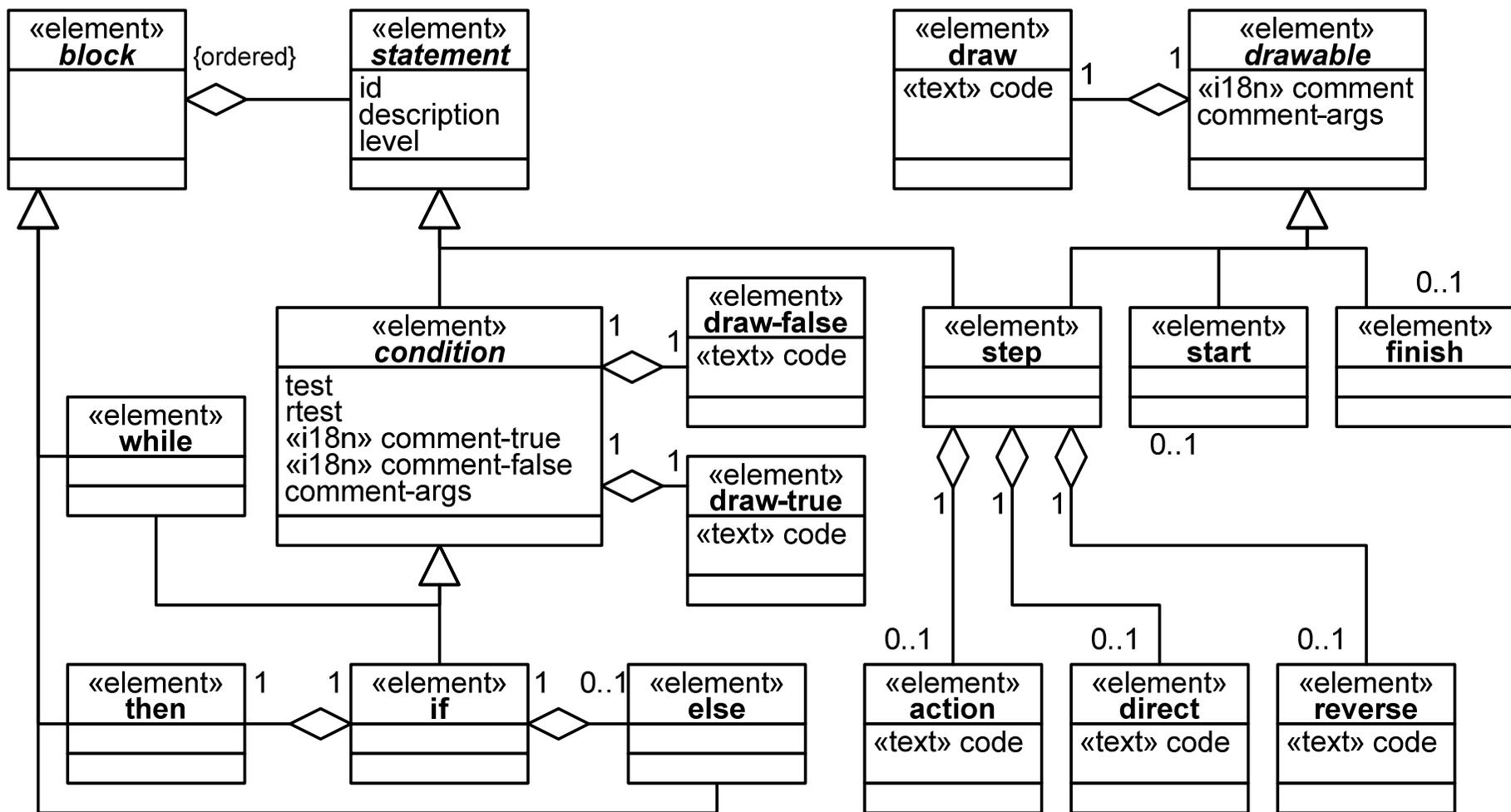
Доказательство корректности методов

- Доказываемые свойства
 - Адекватность
 - Обратимость
 - Полнота
 - Непротиворечивость
 - Отсутствие недостижимых состояний
- Линейность по размеру исходной программы
- Доказательства основаны на структурной индукции

Язык описания логики визуализаторов

- Основан на XML
- Описание визуализируемого алгоритма
 - процедуры
 - переменные
 - операторы
 - комментарии
 - связь с визуальным представлением
- Описание конфигурации визуализатора
 - конфигурация элементов управления
 - конфигурация визуального представления

Фрагмент диаграммы элементов языка описания визуализаторов



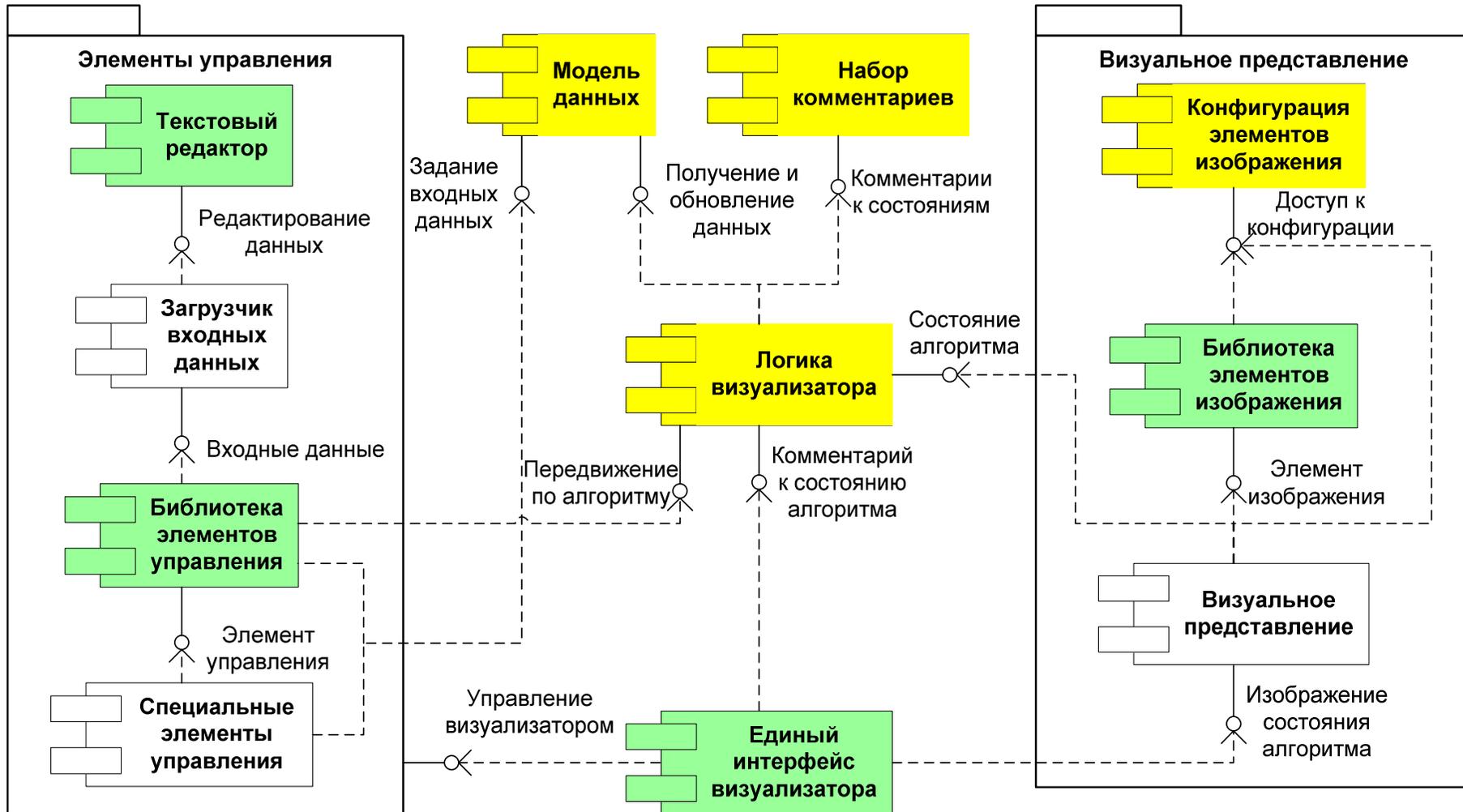
Технология построения визуализаторов алгоритма

1. Постановка задачи и анализ литературы
2. Создание визуализируемой программы
3. Проектирование визуализатора
4. Построение описания визуализируемой программы
5. Реализация визуального представления
6. Реализация элементов управления
7. Интеграция и отладка визуализатора
8. Оформление проектной документации

Система визуализации Vizi

- Статическая часть
 - Поддержка языка описания логики визуализатора
 - Генерация системы автоматов и кода по ней
- Динамическая часть
 - Элементы визуального представления
 - Общие элементы управления
 - Единый интерфейс

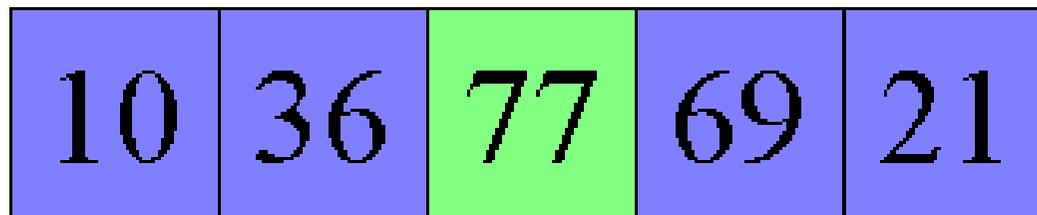
Структура визуализатора



Пример построения визуализатора

Алгоритм поиска максимума в массиве
натуральных чисел

max = 36



77 больше текущего максимума (36)

A control panel with two rows of buttons. The top row contains: '<<', '>>' (with a dotted border), 'Рестарт', 'Авто', '<<', 'Задержка: 1000', '>>', and '?'. The bottom row contains: 'Случайно', 'Сохранить/Загрузить', '<<', 'Элементов: 7', and '>>'.

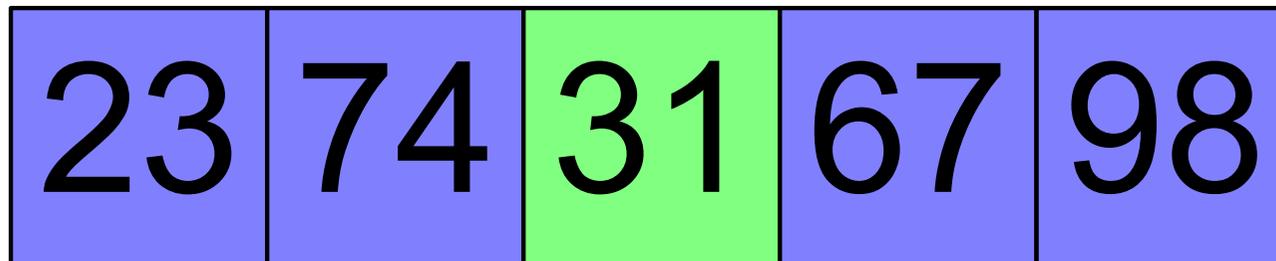
Создание визуализируемой программы

```
int max = 0;
int i = 0;
while (i < a.length) {
    if (max < a[i]) {
        max = a[i];
    }
    i++;
}
```

Проектирование визуализатора

- Проверка на обновление максимума
 - Проверяется необходимость обновить текущий максимум
 - $\{0\}$ больше текущего максимума ($\{1\}$)
 - Текущий элемент подсвечивается зеленым цветом

max=74



Фрагмент описания логики визуализатора

```
<if
  id           = "Cond"
  description  = "Условие"
  test        = "max < a[i]"
  true-comment-ru = "{0} больше текущего максимума ({1})"
  true-comment-en = "{0} greater than current maximum ({1})"
  false-comment-ru = "{0} не больше текущего максимума ({1})"
  false-comment-en = "{0} less than current maximum ({1})"
  comment-args  = "new Integer(a[i]), new Integer(max)"
>
  <draw>
    @visualizer.updateArray(i, 1);
  </draw>
  ...
</if>
```

Построение модели данных

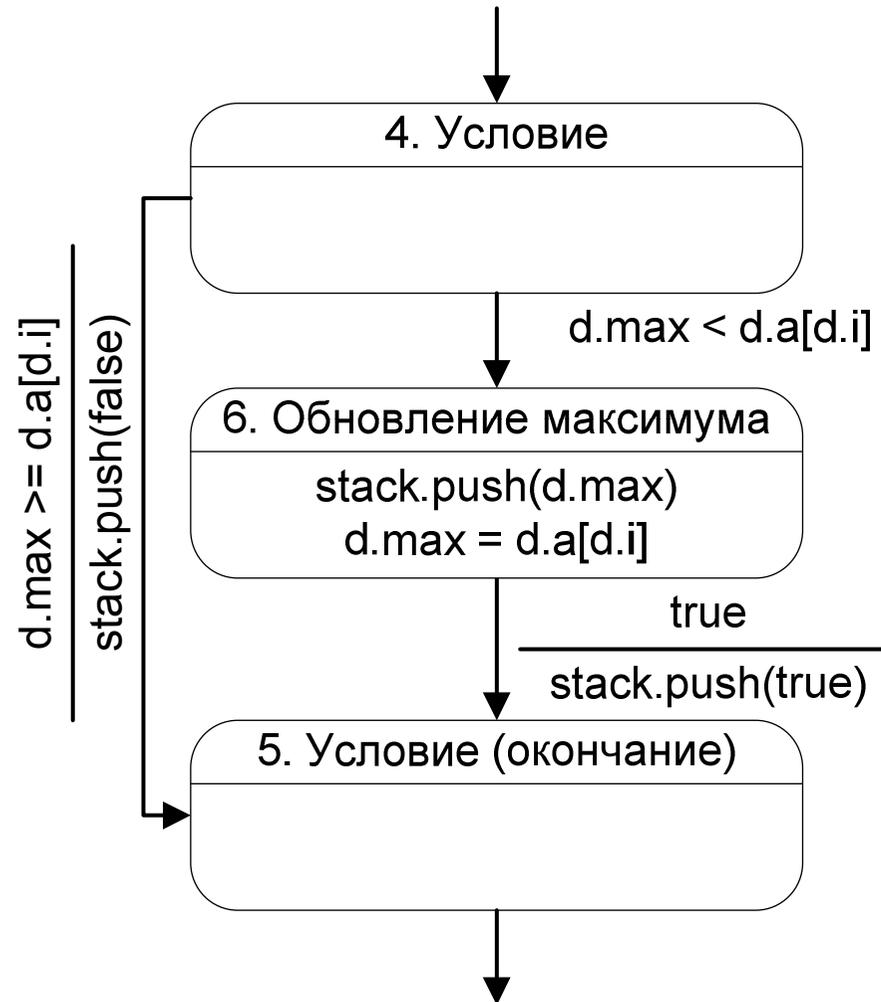
```
public final static class Data {  
    public int max;  
    public int a[];  
    public int Main_i;  
}
```

```
d.max = 0;
```

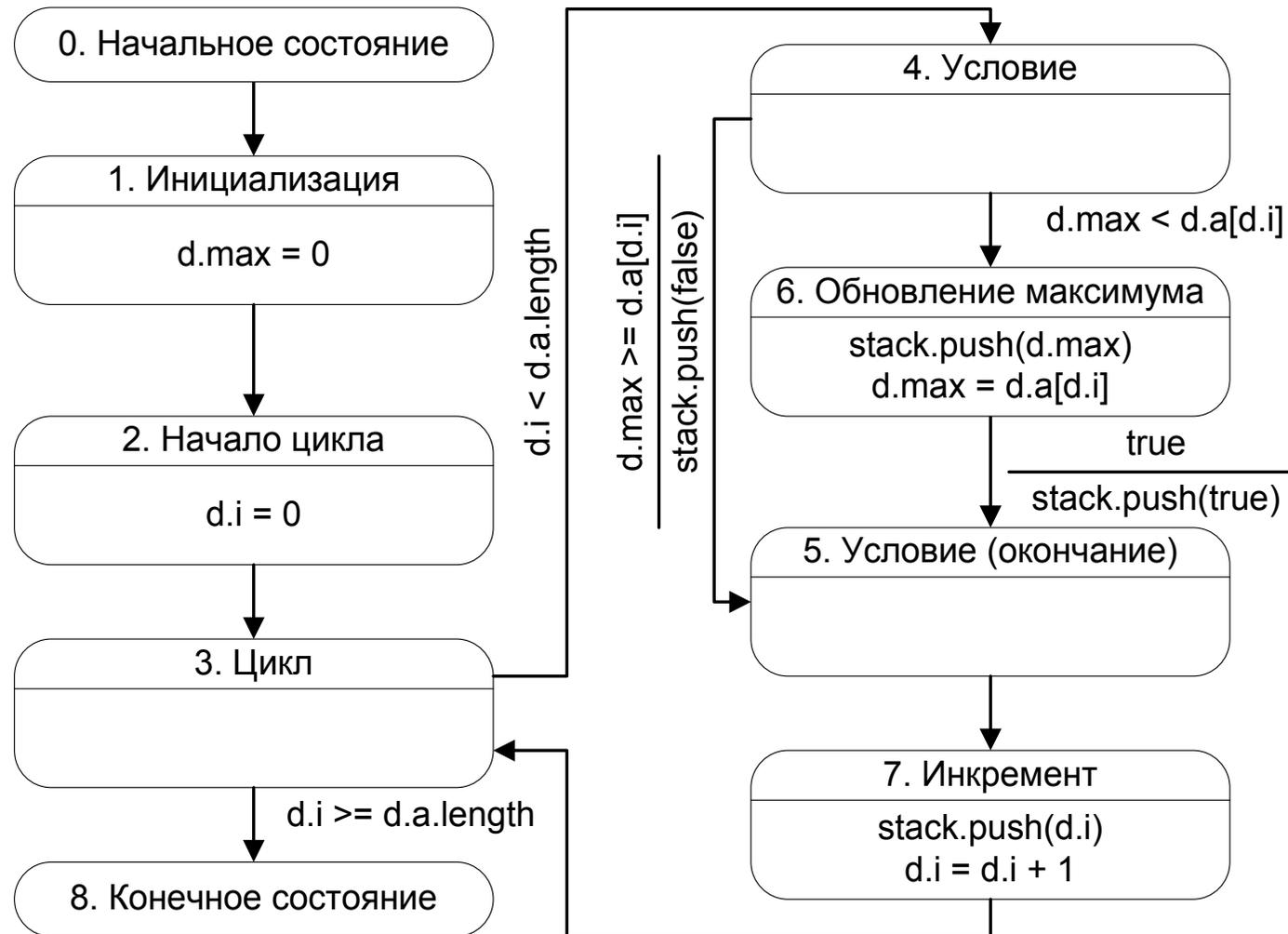
```
d.i = 0;
```

```
while (d.i < d.a.length) {  
    if (d.max < d.a[d.i]) d.max = d.a[d.i];  
    d.i++;  
}
```

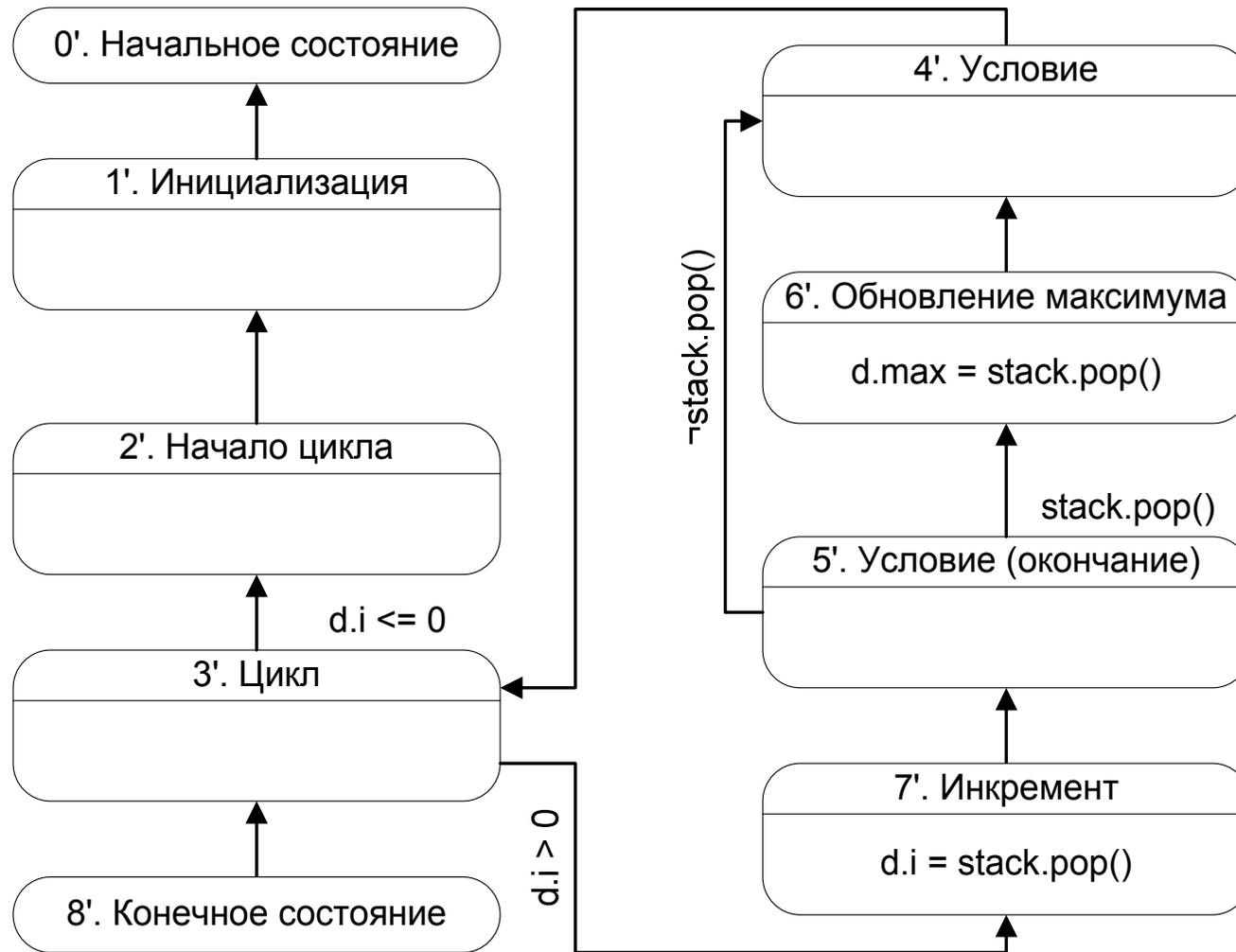
Фрагмент прямого автомата для ветвления



Прямой автомат



Обратный автомат



Сравнительный анализ систем визуализации

Алгоритм	<i>BaseApplet</i> , КБ	<i>Vizi</i> , КБ	Уменьшение размера
Бойера-Мура	18.69	12.06	35.5%
Поиск двусвязных компонент графа	26.54	22.87	13.8%
Обход графа в ширину и глубину	32.34	25.20	22.1%
Форда-Фалкерсона	45.80	26.57	42.0%
Форда-Беллмана	51.39	26.08	49.3%
Дерево отрезков	61.01	27.63	54.7%
2-3 дерева	82.28	33.20	59.6%
Красно-черные деревья	95.20	62.50	34.4%

Внедрение

- Создание и применение визуализаторов
 - Кафедра «Компьютерные технологии» СПбГУ ИТМО
 - Специализированный учебно-научный центр МГУ
- Применение визуализаторов
 - Петрозаводский государственный университет
 - Лицей «Физико-техническая школа»

Визуализаторы алгоритмов, сделанные в СПбГУ ИТМО (1)

- Алгоритм Укконена
- Алгоритм Прима
- Алгоритм Краскала
- Алгоритм Флойда
- Алгоритм Дейкстры
- Алгоритм Форда-Беллмана
- Битонический алгоритм для задачи коммивояжера
- Циклы и разрезы в графах
- Построение максимального паросочитания
- Венгерский алгоритм
- Сортировки слиянием, быстрая, кучей, квадратичные, линейные
- Алгоритм Малхотры-Кумара-Махешвари
- Алгоритм Диница
- 2-3 Деревья
- AVL деревья
- Алгоритм Штрассена
- Дерево отрезков
- Генерация простых строк и построение циклов де Брюина

Всего более 100 визуализаторов

Визуализаторы алгоритмов, сделанные в СПбГУ ИТМО (2)

Визуализатор	Автоматов	Состояний
2-3 дерева	14	195
Алгоритм Малхотры-Кумара-Махешвари	18	89
Алгоритм Диница	8	68
Алгоритм Краскала	6	32
Алгоритм Флойда	2	47
Венгерский алгоритм	8	46
Алгоритм Укконена	2	57
Алгоритм Куна	4	43

Публикации (1)

1. **Корнеев Г.А., Васильев В.Н., Парфенов В.Г., Столяр С.Е.** Визуализаторы алгоритмов как основной инструмент технологии преподавания дискретной математики и программирования / Труды международной научно-методической конференции «Телематика-2001». СПб.: СПбГИТМО (ТУ). 2001.
2. **Корнеев Г.А., Шалыто А.А.** Реализация конечных автоматов с использованием объектно-ориентированного программирования / Труды X международной научно-методической конференции «Телематика-2003». СПб.: СПбГИТМО (ТУ). 2003.
3. **Корнеев Г.А., Казаков М.А., Шалыто А.А.** Построение логики работы визуализаторов алгоритмов на основе автоматного подхода / Труды X международной научно-методической конференции «Телематика-2003». СПб.: СПбГИТМО (ТУ). 2003.
4. **Казаков М.А., Корнеев Г.А., Шалыто А.А.** Разработка логики визуализаторов алгоритмов на основе конечных автоматов // Телекоммуникации и информатизация образования. 2003. № 6.
5. **Корнеев Г.А., Шамгунов Н.Н., Шалыто А.А.** Обход деревьев на основе автоматного подхода // Компьютерные инструменты в образовании. 2004. № 3.
6. **Корнеев Г. А.** Технология разработки визуализаторов алгоритмов / Труды II межвузовской конференции молодых учёных. СПб.: СПбГУ ИТМО, 2005.

Публикации (2)

7. **Корнеев Г.А., Шалыто А.А.** Преобразование программ в систему взаимодействующих конечных автоматов / Труды Второй Всероссийской Научной конференции «Методы и средства обработки информации». М.: МГУ. 2005.
8. **Корнеев Г. А.** Метод преобразования программ в систему взаимодействующих автоматов / Труды II межвузовской конференции молодых учёных. СПб.: СПбГУ ИТМО. 2005.
9. **Корнеев Г.А.** Преобразование программы в систему взаимодействующих автоматов, допускающих двустороннюю трассировку / Материалы политехнического симпозиума 2005 «Молодые ученые — промышленности Северо-Западного региона». СПб.: Политехнический университет. 2005.
10. **Корнеев Г.А., Шалыто А.А.** Построение визуализаторов алгоритмов дискретной математики // Научно-технический вестник СПбГУ ИТМО. Выпуск 23. Высокие технологии в оптических и информационных системах. СПб.: СПбГУ ИТМО. 2005.
11. **Корнеев Г.А., Шалыто А.А.** VIZI — язык описания логики визуализаторов алгоритмов // Научно-технический вестник СПбГУ ИТМО. Выпуск 23. Высокие технологии в оптических и информационных системах. СПб.: СПбГУ ИТМО. 2005.

Заключение

- Разработаны формальные методы преобразования императивных программ в систему взаимодействующих конечных автоматов
- Предложен язык описания визуализаторов алгоритмов
- Разработана технология автоматизированного построения визуализаторов алгоритмов дискретной математики
- Разработана система визуализации *Vizi*, поддерживающая предложенную технологию
- Система визуализации *Vizi* и визуализаторы, построенные на ее основе, успешно внедрены

Спасибо за внимание

Соответствие паспорту специальности

Специальность 05.13.12

«Системы автоматизации проектирования»

Области исследований:

5. Разработка научных основ обучения автоматизированному проектированию.

Подходы к построению визуализаторов

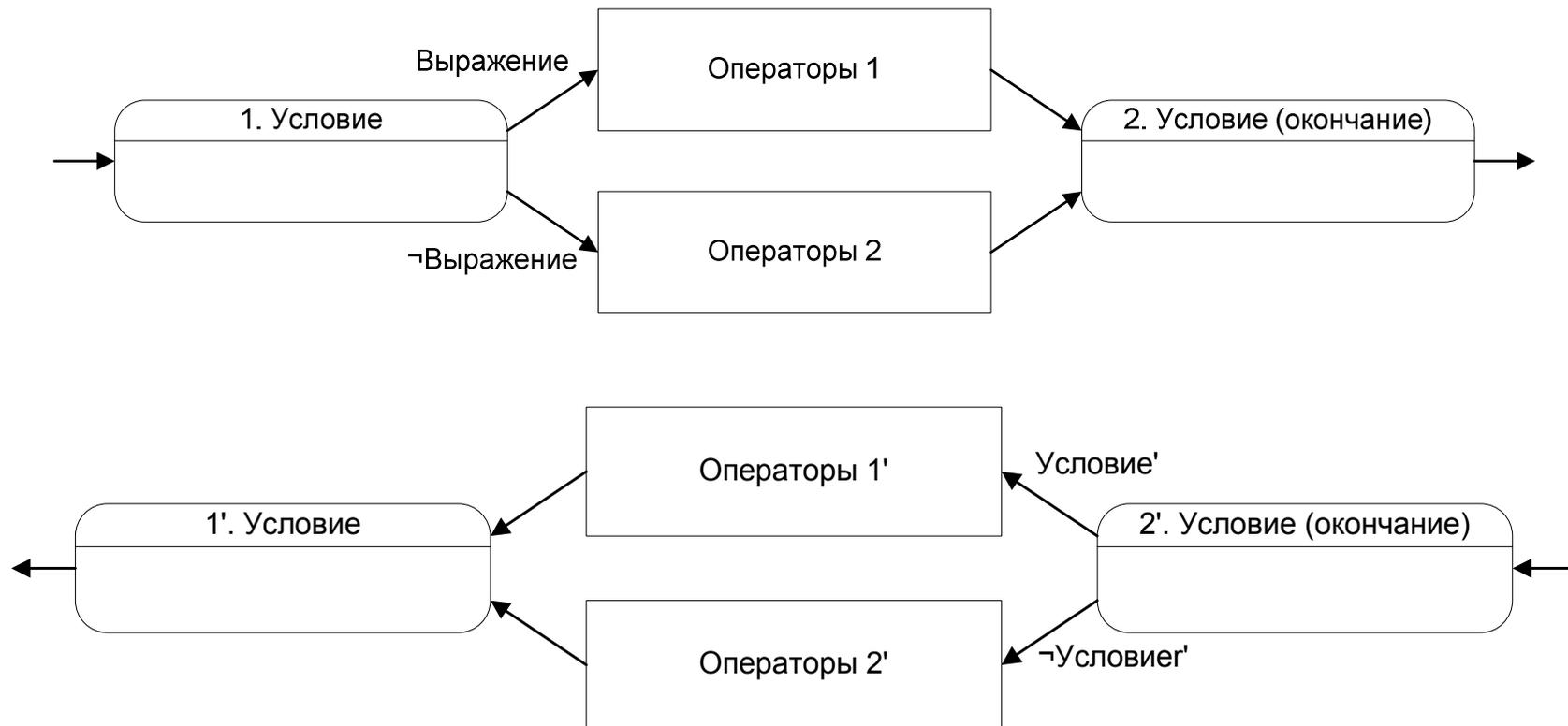
- Визуализаторы программ
 - М. Браун
 - Р. Седжвик

- Визуализаторы данных
 - Дж. Стаско

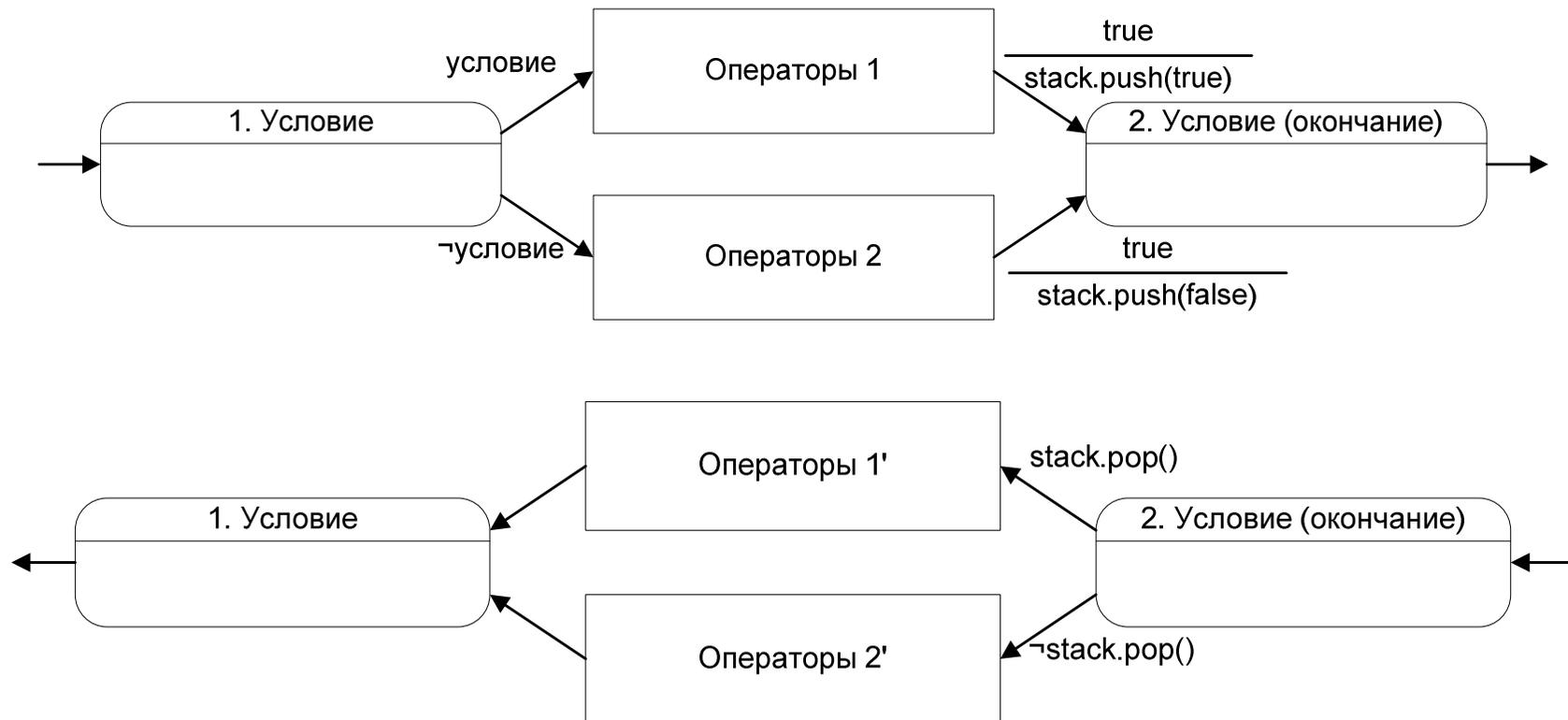
Пример фрагмента кода

```
switch (state) {
  case 0: { // Начальное состояние
    state = 1; // Инициализация
    break;
  } case 1: { // Инициализация
    d.max = 0;
    state = 2; // Иниц. цикла
    break;
  } case 2: { // Иниц. цикла
    d.i = 0;
    state = 3; // Цикл
    break;
  } case 3: { // Цикл
    if (d.Main_i < d.a.length)
      state = 4; // Условие
    else state = END_STATE;
    break;
  }
  case 4: { // Условие
    if (d.max < d.a[d.Main_i])
      state = 6; // Обновление
    else state = 5; // End if
    break;
  } case 5: { // End if
    state = 7; // Increment
    break;
  } case 6: { // Update
    d.max = d.a[d.Main_i];
    state = 5; // End if
    break;
  } case 7: { // Increment
    d.Main_i++;
    state = 3; // Loop
    break;
  }
}
```

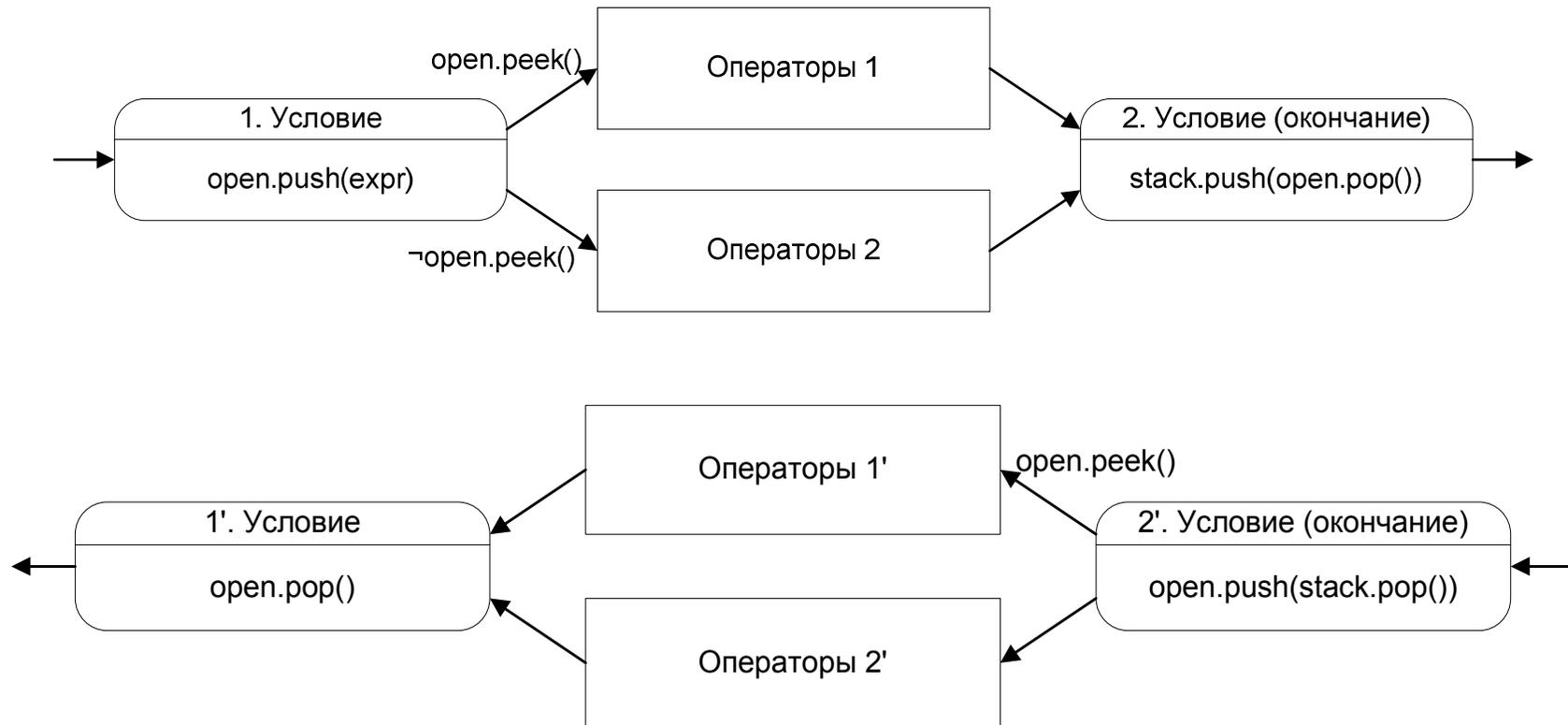
Прямое обращение оператора ветвления



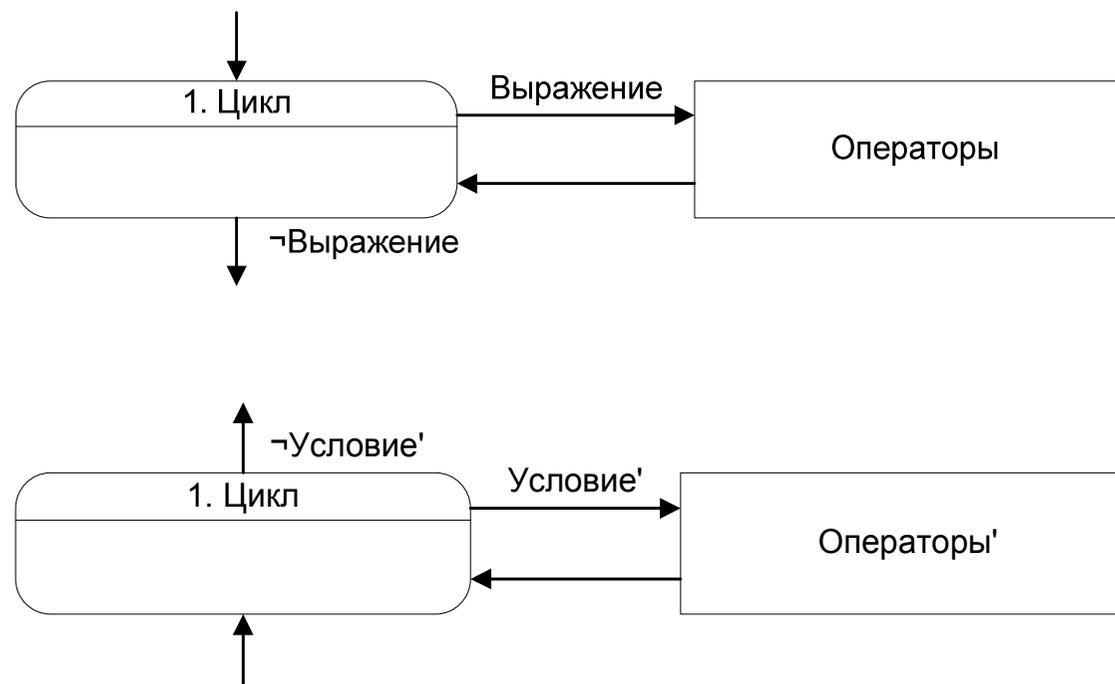
Косвенное обращение оператора ветвления первым способом



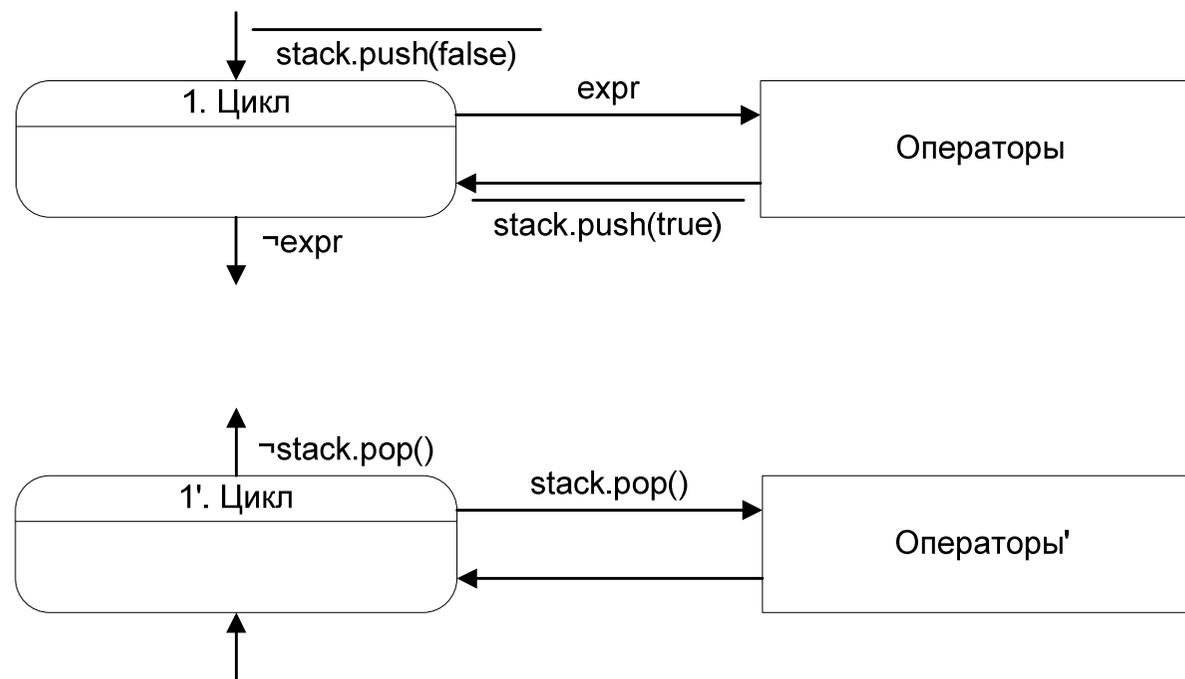
Косвенное обращение оператора ветвления вторым способом



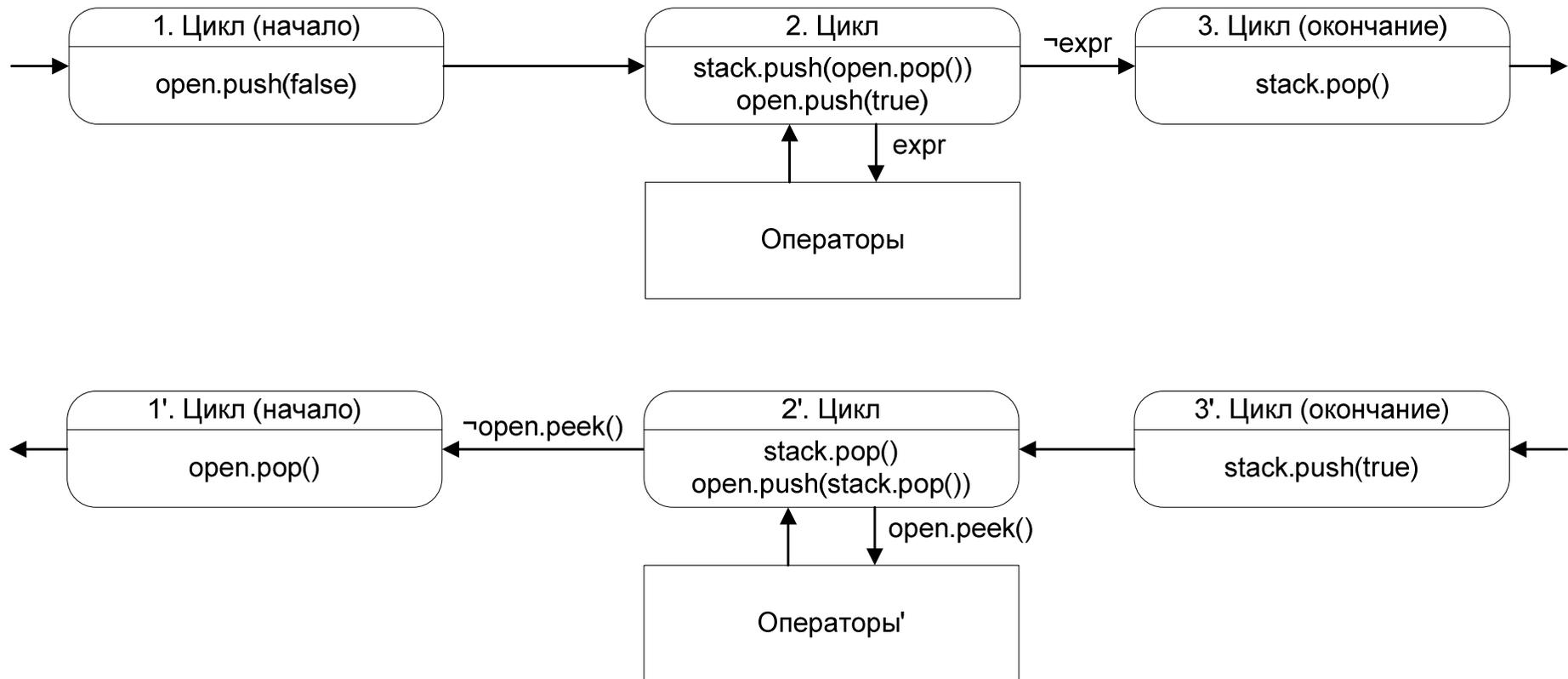
Прямое обращение оператора цикла



Косвенное обращение оператора цикла первым способом



Косвенное обращение оператора цикла вторым способом



Пример автоматов для рекурсивной программы

