

Теория автоматов в программировании

Лекция 1
Ф. Н. Царев
08.09.2009



План курса

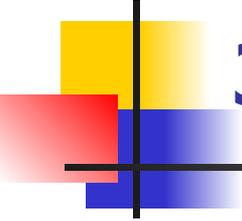
- Основные понятия автоматного программирования
- Инструментальные средства автоматного программирования
- Применение генетических алгоритмов
- Верификация автоматных программ
- Текстовые языки автоматного программирования
- ...



Преподаватели курса

- Шалыто А. А.
- Царев Ф. Н.
- ...

Место и время проведения занятий



- Пятница, 17-20
- Аудитория 218, 219 или 146



Как получить зачет?

- 5 семестр
- Сдать лабораторную работу по генетическим алгоритмам
- Сообщить тему своей курсовой работы

Виртуальная лаборатория по ГА



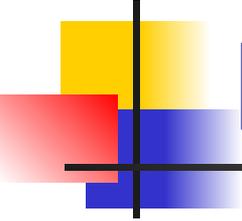
- Два варианта: Java или C#
- Сайт is.ifmo.ru, раздел «Генетические алгоритмы», подраздел «Лабораторные работы»
- Сдать работу = сдать программу + выложить на сайт отчет



Как сдать курсовую работу?

- 6 семестр
- Написать программу
- Написать проектную документацию
- Выложить ее на сайт is.ifmo.ru
- Не забывать записываться в календарь Шалыто

Цель выполнения курсовой работы



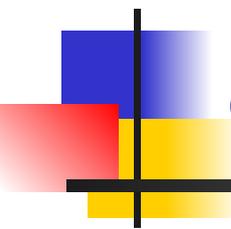
- Привести ее в такое состояние, чтобы было не стыдно выкладывать в Интернет



Материалы по курсу

- Сайт кафедры «Технологии программирования» по автоматному программированию и мотивации к творчеству is.ifmo.ru
- Книга Н. Поликарпова, А. Шалыто Автоматное программирование
- Материалы лекций

1.1 Области применения автоматного программирования





1.1.1. Классификация программ по Харелу

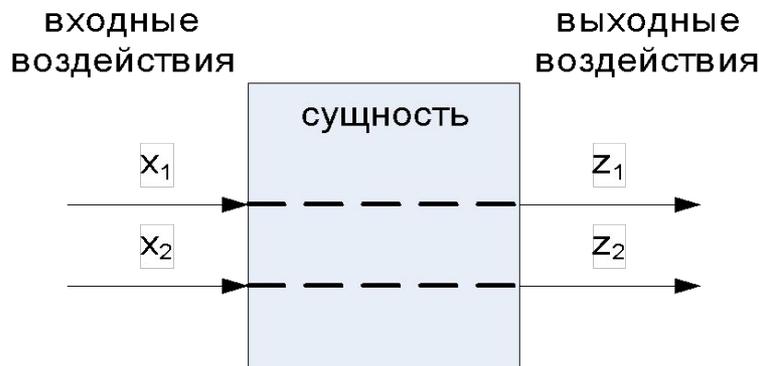
- Трансформирующие системы
 - некоторое преобразование входных данных
 - например: компиляторы, архиваторы
- Интерактивные системы
 - взаимодействуют с окружающей средой в режиме диалога
 - например: текстовые редакторы
- Реактивные системы
 - обмен со средой сообщениями, в темпе задаваемом средой
 - например: системы контроля



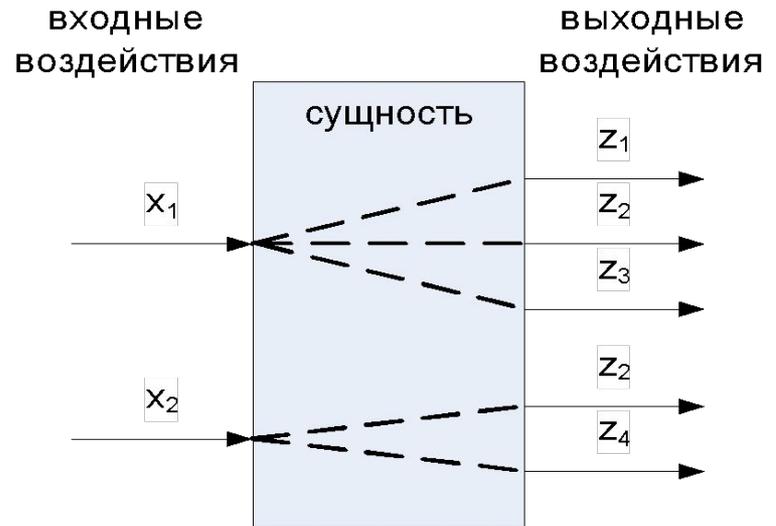
1.1.2. Критерии применимости

- «Сложное поведение»
 - поведение, зависящее от состояния
 - реакция зависит от предыстории
- «Простое поведение»
 - поведение, не зависящее от состояния
 - реакция зависит только от воздействия

1.1.2. Критерии применимости



Сущность с простым поведением



Сущность со сложным поведением

Пример использования: ЭЛЕКТРОННЫЕ ЧАСЫ

- Простое поведение
 - Н – увеличивает на единицу число часов
 - М – увеличивает на единицу число минут



Пример использования:

ЭЛЕКТРОННЫЕ ЧАСЫ

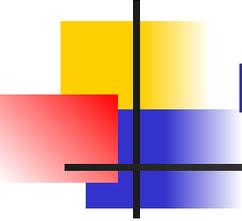
- Сложное поведение
 - H – увеличивает на единицу число часов
 - M – увеличивает на единицу число минут
 - A – включает и выключает настройку «будильник»





1.1.3. Идеи автоматного программирования:

- отделение логики от семантики
- описание логики при автоматном подходе строго структурировано

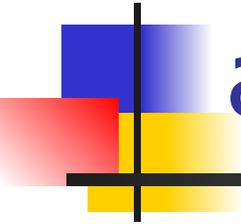


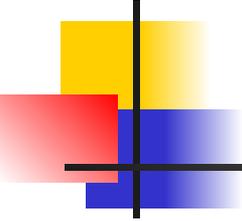
1.1.4. Рекомендации при использовании автоматного подхода

- используйте автоматный подход при создании любой программной системы, в которой есть сущности со сложным поведением
- используйте автоматный подход при создании только тех компонент системы, которые являются сущностями со сложным поведением

1.2. Основные понятия

автоматного программирования

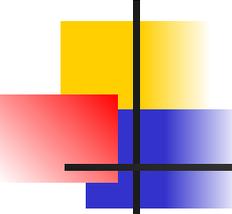




1.2.1. Основные понятия

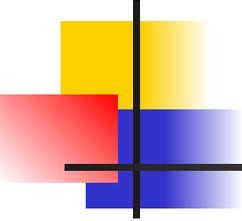
- *Состояние*

- особая величина, которая в неявной форме объединяет все входные воздействия прошлого, влияющие на реакцию сущности в настоящий момент времени



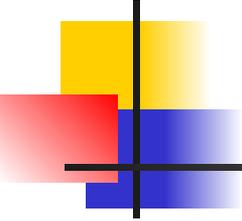
1.2.1. Основные понятия

- *Свойства состояния системы:*
 - текущее состояние несет в себе всю информацию о прошлом системы, необходимую для определения ее реакции на любое входное воздействие, формируемое в момент времени t_0
 - не требуется знание предыстории



1.2.1. Основные понятия

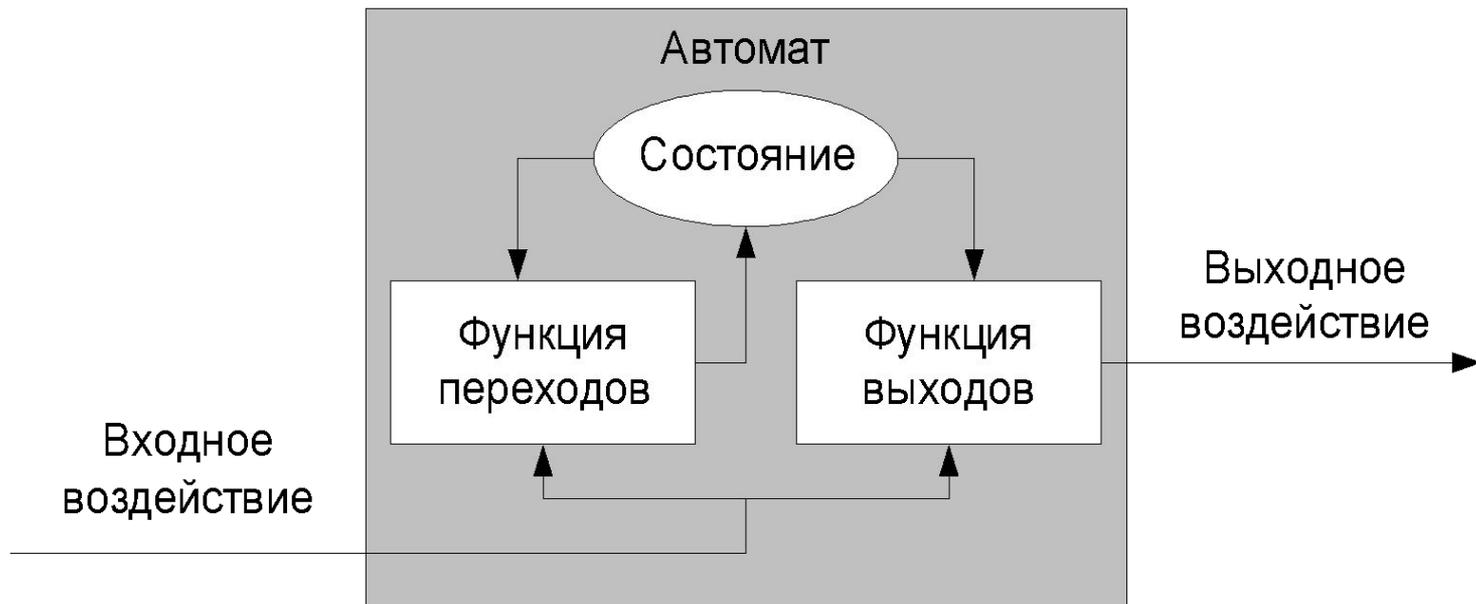
- *Входное воздействие*
 - это вектор, составляющие которого - события и входные переменные
- *Функция переходов*
 - правила, по которым происходит смена состояний
- *Выходное воздействие*



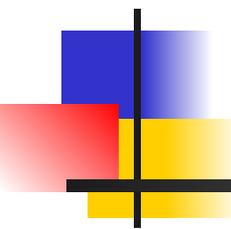
1.2.1. Основные понятия

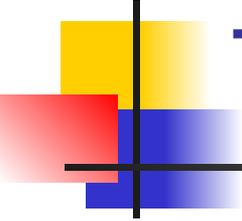
- ***Функция выходов***
 - правила формирования выходных воздействий
- ***Автомат без выходов (конечный)***
 - совокупность конечного множества состояний и конечного множества входных воздействий

1.2.2. Конечный автомат



1.3. Парадигма автоматного программирования

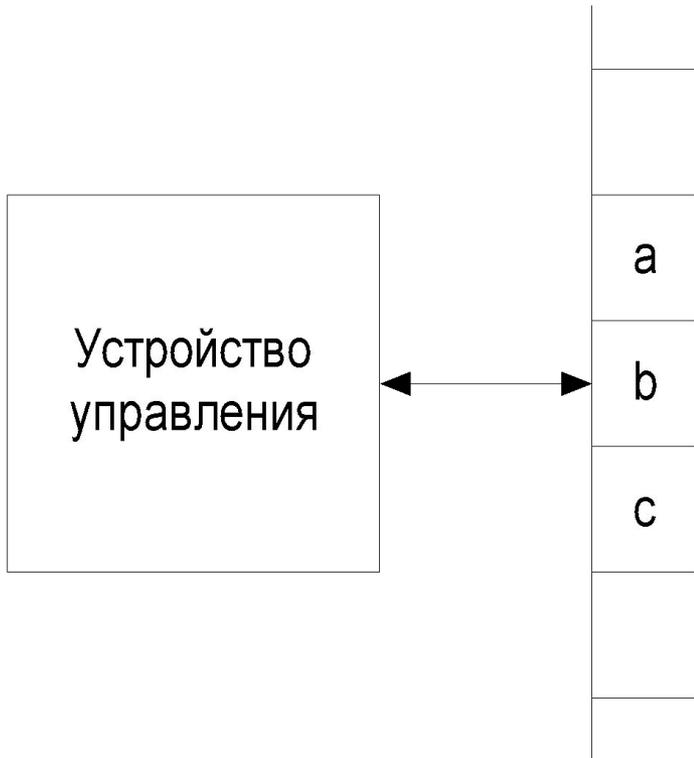




Тезис Тьюринга-Черча

Все, что можно «вычислить», «запрограммировать» или «распознать» в любом смысле (из формально определенных в настоящее время) можно вычислить, запрограммировать или распознать с помощью подходящей машины Тьюринга

1.3.1. Машина Тьюринга



*Машина Тьюринга
состоит из 2-х частей:*

- Устройство управления
- Запоминающее устройство - лента



1.3.1. Машина Тьюринга

- Устройство управления представляет собой конечный автомат
 - единственное входное воздействие:
 - символ, считанный с ленты
 - два выходных воздействия:
 - символ, записываемый на ленту
 - указание головке сдвинуться на одну ячейку в ту или иную сторону, либо остаться на месте

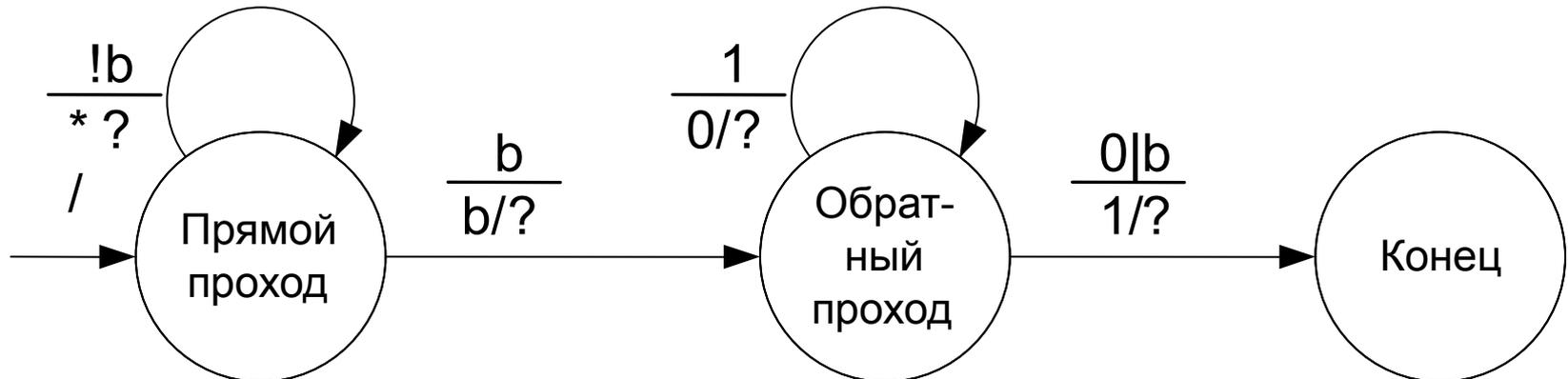


1.3.2. Программирование на Машине Тьюринга

- *Реализация функции инкремент:*
 - двигаться вправо, пока не встретится пустой символ
 - сдвинуться на одну ячейку влево
 - пока в текущей ячейке находится '1', заменять его на '0' и двигаться влево
 - если в текущей ячейке находится '0' или 'blank', записать в ячейку '1' и завершить работу

1.3.3. Краткое описание поведения машины

- *Граф переходов, где:*
 - вершины - состояния автомата
 - дуги – переходы между состояниями





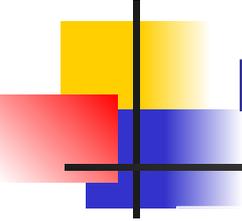
1.3.4. Выводы по работе машины Тьюринга

- Для того, чтобы задать алгоритм для машины Тьюринга, достаточно описать ее поведение в каждом из трех состояний управляющего автомата
- Состояния управляющего автомата определяют *действия* машины, а состояние ленты – *результат* этих действий



1.3.4. Выводы по работе машины Тьюринга

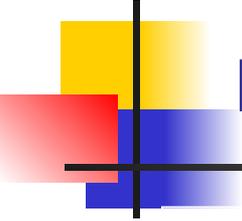
- Состояния устройства управления
 - следует явно перечислять, отображать на графе переходов
 - качественные состояния машины
 - называют *управляющими*
- Состояния ленты
 - в программе в явном виде не участвуют, построить граф переходов между ними невозможно
 - количественные состояния машины
 - называют *вычислительными*



1.3.5. Управляющие и вычислительные состояния

Управляющие состояния

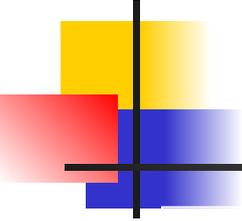
- Их относительно немного
- Каждое из них имеет вполне определенный смысл и качественно отличается от других
- Они определяют действия, которые совершает сущность



1.3.5. Управляющие и вычислительные состояния

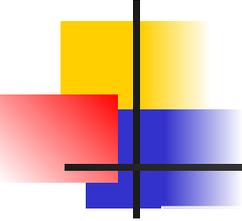
Вычислительные состояния

- Их количество либо бесконечно, либо конечно, но очень велико
- Большинство из них не имеет смысла и отличается от остальных лишь количественно
- Они непосредственно определяют лишь результаты действий



1.3.6. Сущность со сложным поведением

- Управляющая часть
 - управляющий автомат
 - отвечает за логику поведения – выбор выполняемых действий, зависящий от текущего состояния и входных воздействий, а также за переход в новое состояние



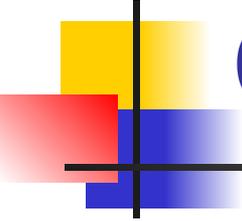
1.3.6. Сущность со сложным поведением

- Управляемая часть
 - объект управления
 - отвечает за выполнение действий, выбранных для выполнения управляющей частью, и, возможно, за формирование некоторых компонентов входных воздействий для управляющей части – *обратных связей*



1.3.6. Сущность со сложным поведением

- Парадигма автоматного программирования состоит в представлении сущностей со сложным поведением в виде автоматизированных объектов управления
- *Автоматизированный объект управления* - объект управления, интегрированный вместе с системой управления в одно устройство



Спасибо за внимание

- Следующий раз – в пятницу, 11 сентября в 17-20
- А. А. Шалыто