

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ  
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

---

**Б1.Б.16 Моделирование систем управления**

*(код и наименование дисциплины в соответствии с РУП)*

**Направление подготовки (специальность) 27.03.04 “Управление в технических системах”**

**Профиль образовательной программы** Интеллектуальные системы обработки информации и управления

**Форма обучения очная**

## СОДЕРЖАНИЕ

<b>1. Конспект лекций Моделирование систем.....</b>	<b>4</b>
1.1 Лекция № 1 Общие сведения.....	4
1.2 Лекция № 2 Понятие математической схемы .....	13
1.3 Лекция № 3 Дискретно – стохастические модели .....	21
1.4 Лекция № 4 Обобщённые модели.....	23
1.5 Лекция № 5 Последовательность разработки и машинной реализации моделей...24	
1.6 Лекция № 6 Построение концептуальной модели системы и её формализация.	
1.7 Лекция № 7 Алгоритмизация модели и ее машинная реализация.....	28
1.8 Лекция № 8 Получение и интерпретация результатов моделирования.	
1.9 Лекция № 9 Имитационное моделирование.....	32
<b>2. Методические указания по выполнению практических работ.....</b>	<b>38</b>
2.1 Практическая работа № ПР-1 Предмет теории моделирования. Роль и место моделирования в исследовании систем.....	38
2.2 Практическая работа № ПР-2 «Общие сведения. Математические схемы моделирования систем. Классификация видов моделирования.....	42
2.3 Практическая работа № ПР-3 Принципы подхода в моделировании систем. Классификация видов моделирования систем.....	48
2.4 Практическая работа № ПР-4 Понятие математической схемы. Математическая схема общего вида .....	55
2.5 Практическая работа № ПР-5 Типовые математические схемы. Непрерывно-детерминированные модели ( <i>D</i> -схемы).....	62
2.6 Практическая работа № ПР-6 Дискретно-детерминированные модели ( <i>F</i> -схемы). Конечные автоматы Классификация конечных автоматов .....	66
2.7 Практическая работа № ПР-7 Способы задания работы автоматов.....	68
2.8 Практическая работа № ПР-8 Дискретно-стохастические модели ( <i>P</i> -схемы). Вероятностные автоматы.....	77
2.9 Практическая работа № ПР-9 Дискретно-стохастические модели ( <i>P</i> -схемы).....	80
2.10 Практическая работа № ПР-10 Непрерывно – стохастические модели ( <i>Q</i> – схемы).....	82
2.11 Практическая работа № ПР-11 Обобщенные модели ( <i>A</i> - схемы).....	84
2.12 Практическая работа № ПР-12 Последовательность разработки и машинной реализации моделей.....	87
2.13 Практическая работа № ПР-13 Построение концептуальной модели системы и её формализация.....	89

<b>2.14</b>	<b>Практическая работа № ПР-14</b>	<b>Алгоритмизация модели и её машинная реализация.....</b>	<b>90</b>
<b>2.15</b>	<b>Практическая работа № ПР-15</b>	<b>Получение и интерпретация результатов моделирования.....</b>	<b>92</b>
<b>2.16</b>	<b>Практическая работа № ПР-16</b>	<b>Имитационное моделирование.....</b>	<b>93</b>
<b>2.17</b>	<b>Практическая работа № ПР-17</b>	<b>Среда и функциональная структура языка моделирования GPSS.....</b>	<b>96</b>

# 1. КОНСПЕКТ ЛЕКЦИЙ

## 1.1 Лекция №1 (2 часа).

**Тема:** «Общие сведения»

### 1.1.1 Вопросы лекции:

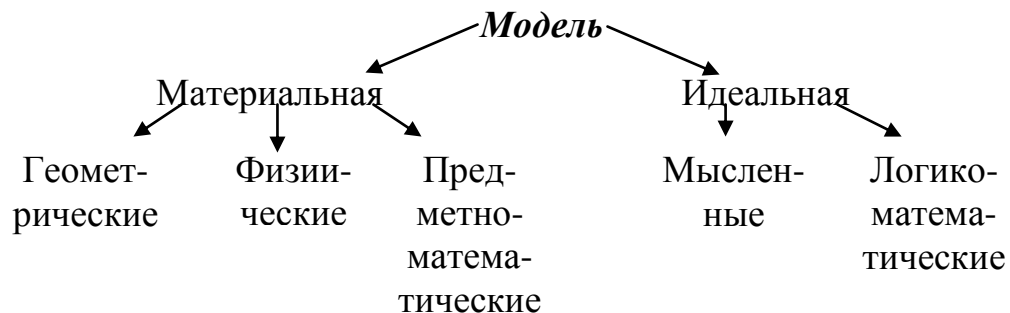
1. Предмет теории моделирования.
2. Роль и место моделирования в исследовании систем.
3. Классификация видов моделирования.
4. Математические схемы моделирования систем.
5. Принципы подхода в моделировании систем.
6. Классификация видов моделирования систем.

### 1.1.2 Краткое содержание вопросов:

1. Предмет теории моделирования.

Моделирование - это замещение одного объекта (оригинала) другим (моделью) и фиксация и изучение свойств модели. Замещение производится с целью упрощения, удешевления, ускорения изучения свойств оригинала.

Модель (лат. *modulus*— мера) — это объект-заместитель объекта-оригинала, обеспечивающий изучение некоторых свойств оригинала.



Компьютерная модель – это программная реализация математической модели, дополненная различными служебными программами (например, рисующими и изменяющими графические образы во времени). Компьютерная модель имеет две составляющие – программную и аппаратную. Программная составляющая так же является абстрактной знаковой моделью. Это лишь другая форма абстрактной модели, которая, однако, может интерпретироваться не только математиками и программистами, но и техническим устройством – процессором компьютера.

Таким образом, моделирование может быть определено как представление объекта моделью для получения информации об этом объекте путем проведения экспериментов с его моделью. Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследования свойств объектов на их моделях называется теорией моделирования.

Теория моделирования — взаимосвязанная совокупность положений, определений, методов и средств создания моделей. Сами модели являются предметом теории моделирования.

Теория моделирования является основной составляющей общей теории систем - системологии, где в качестве главного принципа постулируются осуществимые модели: система представима конечным множеством моделей, каждая из которых отражает определённую грань её сущности.

## 2. Роль и место моделирования в исследовании систем.

Познание любой системы (S) сводится по существу к созданию её модели. Перед изготовлением каждого устройства или сооружения разрабатывается его модель - проект. Любое произведение искусства является моделью, фиксирующее действительность.

Достижения математики привели к распространению математических моделей различных объектов и процессов. Подмечено, что динамика функционирования разных по физической природе систем однотипными зависимостями, что позволяет моделировать их на ЭВМ.

На качественно новую ступень поднялась моделирование в результате разработки методологии имитационного моделирования на ЭВМ.

Сейчас трудно указать область человеческой деятельности, где бы применялось моделирование. Разработаны модели производства автомобилей, выращивания пшеницы, функционирования отдельных органов человека, жизнедеятельности Азовского моря, атомного взрыва, последствий атомной войны.

Специалисты считают, что моделирование становится основной функцией ВС. На практике широко используются АСУ технологическими процессами организационно-экономическими комплексами, процессами проектирования, банки данных и знаний. Но любая из этих систем нуждается в информации об управляемом объекте и модели управляемой объектом, в моделировании тех или иных управляющих решений.

Сами ВС как сложные и дорогостоящие технические системы могут являться объектами моделирования.

Обычно процесс разработки сложной системы осуществляется итерационно с использованием моделирования проектных решений. Если характеристики не удовлетворяют предъявленным требованиям, то по результатам анализа производят корректировку проекта, затем снова проводят моделирование.

При анализе действующих систем с помощью моделирования определяют границы работоспособности системы, выполняют имитацию экспериментальных условий, которые могут возникнуть в процессе функционирования системы. Искусственное создание таких условий на действительной системе затруднено и может привести к катастрофическим последствиям.

Применение моделирования может быть полезным при разработке стратегии развития ВС, её усовершенствования при создании сетей ЭВМ.

В настоящее время при анализе и синтезе сложных (больших) систем получил развитие системный подход, который отличается от классического (или индуктивного - путем перехода от частного к общему и синтезирует (конструирует) систему путем слияния ее компонент, разрабатываемых отдельно) подхода. В отличие от этого системный подход предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежит цель, причем исследуемый объект выделяется из окружающей среды.

Понятие системы и элемента системы. Специалисты по проектированию и эксплуатации сложных систем имеют дело с системами управления различных уровней, обладающими общим свойством - стремлением достичь некоторой цели. Эту особенность учтем в следующих определениях системы.

Система  $S$  — целенаправленное множество взаимосвязанных элементов любой природы.

Внешняя среда  $E$  — множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под ее воздействием.

Понятие модели. Модель — представление объекта, системы или понятия, в некоторой форме, отличного от их реального существования.

Моделирование — во-первых, построение модели, во-вторых, изучение модели, в-третьих, анализ системы на основе данной модели.

При системном подходе к моделированию систем необходимо прежде всего четко определить цель моделирования. Применительно к вопросам моделирования цель возникает из требуемых задач моделирования, что позволяет подойти к выбору критерия и оценить, какие элементы войдут в создаваемую модель  $M$ . Поэтому необходимо иметь критерий отбора отдельных элементов в создаваемую модель.

### 3. Классификация видов моделирования.

Физические модели. В основу классификации положена степень абстрагирования модели от оригинала. Предварительно все модели можно подразделить на 2 группы — физические и абстрактные (математические).

Ф.М. обычно называют систему, эквивалентную или подобную оригиналу, но возможно имеющую другую физическую природу. Виды Ф.М.:

- натуральные;
- квазинатуральные;
- масштабные;
- аналоговые;

Натуральные модели — это реальные исследуемые системы (макеты, опытные образцы). Имеют полную адекватность (соответствия) с системой оригиналом, но дороги.

Квазинатуральные модели — совокупность натуральных и математических моделей. Этот вид используется тогда, когда модель части системы не может быть математической из-за сложности её описания (модель человека оператора) или когда часть системы должна быть исследована во взаимодействии с другими частями, но их ещё не существует или их включение очень дорого (вычислительные полигоны, АСУ).

Масштабная модель — это система той же физической природы, что и оригинал, но отличается от него масштабами. Методологической основой масштабного моделирования является теория подобия. При проектировании ВС масштабные модели могут использоваться для анализа вариантов компоновочных решений.

Аналоговые модели называют системы, имеющие физическую природу, отличающуюся от оригинала, но сходные с оригиналом процессы функционирования. Для создания аналоговой модели требуется наличие математического описания изучаемой системы. В качестве аналоговых моделей используются механические, гидравлические, пневматические и электрические системы. Аналоговое моделирование использует при исследовании средства ВТ на уровне логических элементов и электрических цепей, а так же на системном уровне, когда функционирование системы описывается, например, дифференциальными или алгебраическими уравнениями.

Математические модели. Математические модели представляют собой формализованное представление системы с помощью абстрактного языка, с помощью математических соотношений, отражающих процесс функционирования системы. Для составления математических моделей можно использовать любые математические средства — алгебраическое, дифференциальное, интегральное исчисления, теорию множеств, теорию алгоритмов и т.д. По существу вся математика создана для составления и исследования моделей объектов и процессов.

К средствам абстрактного описания систем относятся также языки химических формул, схем, чертежей, карт, диаграмм и т.п. Выбор вида модели определяется особенностями изучаемой системы и целями моделирования, т.к. исследование модели позволяет получить ответы на определённую группу вопросов. Для получения другой информации может потребоваться модель другого вида. Математические модели можно классифицировать как детерминированные и вероятностные, аналитические, численные и имитационные.

Детерминированное моделирование отображает процессы, в которых предполагается отсутствие всяких случайных воздействий; стохастическое моделирование отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики, т. е. набор однородных реализаций.

Аналитической моделью называется такое формализованное описание системы, которое позволяет получить решение уравнения в явном виде, используя известный математический аппарат.

Численная модель характеризуется зависимостью такого вида, который допускает только частные решения для конкретных начальных условий и количественных параметров моделей.

Имитационная модель — это совокупность описания системы и внешних воздействий, алгоритмов функционирования системы или правил изменения состояния системы под влиянием внешних и внутренних возмущений. Эти алгоритмы и правила не дают возможности использования имеющихся математических методов аналитического и численного решения, но позволяют имитировать процесс функционирования системы и производить вычисления интересующих характеристик. Имитационные модели могут быть созданы для гораздо более широкого класса объектов и процессов, чем аналитические и численные. Поскольку для реализации имитационных моделей служат ВС, средствами формализованного описания ИМ служат универсальные и специальные алгоритмические языки. ИМ в наибольшей степени подходят для исследования ВС на системном уровне.

#### 4. Математические схемы моделирования систем.

Математическая модель — это совокупность математических объектов (чисел, переменных, множеств, векторов, матриц и т.п.) и отношений между ними, адекватно отображающая физические свойства создаваемого технического объекта. Процесс формирования математической модели и использования ее для анализа и синтеза называется математическим моделированием.

При построении математической модели системы необходимо решить вопрос об ее полноте. Полнота модели регулируется, в основном, выбором границы «система  $S$  — среда  $E$ ». Также должна быть решена задача упрощения модели, которая помогает выделить в зависимости от цели моделирования основные свойства системы, отбросив второстепенные.

При переходе от содержательного к формальному описанию процесса функционирования системы с учетом воздействия внешней среды применяют математическую схему как звено в цепочке «описательная модель — математическая схема — математическая (аналитическая или (и) имитационная) модель».

#### 5. Принципы подхода в моделировании систем.

В настоящее время при анализе и синтезе сложных (больших) систем получил развитие системный подход, который отличается от классического (или индуктивного) подхода. Классический подход рассматривает систему путем перехода от частного к общему и синтезирует (конструирует) систему путем слияния ее компонент, разрабатываемых отдельно. В отличие от этого системный подход предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежит цель, причем исследуемый объект выделяется из окружающей среды.

Объект моделирования. Специалисты по проектированию и эксплуатации сложных систем имеют дело с системами управления различных уровней, обладающими общим свойством — стремлением достичь некоторой цели. Эту особенность учтем в следующих определениях системы.

Система или объект  $S$  — целенаправленное множество взаимосвязанных элементов любой природы.

Внешняя среда  $E$  — множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под ее воздействием.

В зависимости от цели исследования могут рассматриваться разные соотношения между самим объектом  $S$  и внешней средой  $E$ . Таким образом, в зависимости от уровня,

на котором находится наблюдатель, объект исследования может выделяться по-разному и могут иметь место различные взаимодействия этого объекта с внешней средой.

С развитием науки и техники сам объект непрерывно усложняется, и уже сейчас говорят об объекте исследования как о некоторой сложной системе, которая состоит из различных компонент, взаимосвязанных друг с другом. Поэтому, рассматривая системный подход как основу для построения больших систем и как базу создания методики их анализа и синтеза, прежде всего необходимо определить само понятие системного подхода.

Системный подход — это элемент учения об общих законах развития природы и одно из выражений диалектического учения. При системном подходе к моделированию систем необходимо прежде всего четко определить цель моделирования. Поскольку невозможно полностью смоделировать реально функционирующую систему (систему-оригинал, или первую систему), создается модель (система-модель, или вторая система) под поставленную проблему.

Таким образом, применительно к вопросам моделирования цель возникает из требуемых задач моделирования, что позволяет подойти к выбору критерия и оценить, какие элементы войдут в создаваемую модель  $M$ . Поэтому необходимо иметь критерий отбора отдельных элементов в создаваемую модель.

Подходы к исследованию систем. Важным для системного подхода является определение структуры системы — совокупности связей между элементами системы, отражающих их взаимодействие. Структура системы может изучаться

1. извне с точки зрения состава отдельных подсистем и отношений между ними,
2. а также изнутри, когда анализируются отдельные свойства, позволяющие системе достигать заданной цели, т. е. когда изучаются функции системы.

В соответствии с этим наметился ряд подходов к исследованию структуры системы с ее свойствами, к которым следует прежде всего отнести структурный подходифункциональный подход.

При структурном подходе выявляются состав выделенных элементов системы  $S$  и связи между ними. Совокупность элементов и связей между ними позволяет судить о структуре системы. Последняя в зависимости от цели исследования может быть описана на разных уровнях рассмотрения. Наиболее общее описание структуры — это топологическое описание, позволяющее определить в самых общих понятиях составные части системы и хорошо формализуемое на базе теории графов.

Менее общим является функциональное описание, когда рассматриваются отдельные функции, т. е. алгоритмы поведения системы, и реализуется функциональный подход, оценивающий функции, которые выполняет система, причем под функцией понимается свойство, приводящее к достижению цели. Поскольку функция отображает свойство, а свойство отображает взаимодействие системы  $S$  с внешней средой  $E$ , то свойства могут быть выражены в виде либо некоторых характеристик элементов и подсистем системы, либо системы  $S$  в целом. При наличии некоторого эталона сравнения можно ввести количественные и качественные характеристики систем. Для количественной характеристики вводятся числа, выражающие отношения между данной характеристикой и эталоном. Качественные характеристики системы находятся, например, с помощью метода экспертных оценок.

Проявление функций системы во времени  $S(t)$ , т. е. функционирование системы, означает переход системы из одного состояния в другое, т. е. движение в пространстве состояний  $Z$ .

## 6. Классификация видов моделирования систем.

В основе моделирования лежит теория подобия, которая утверждает, что абсолютное подобие может иметь место лишь при замене одного объекта другим точно



таким же [4]. При моделировании абсолютное подобие не имеет места и стремятся к тому, чтобы модель достаточно хорошо отображала исследуемую сторону функционирования объекта.

**Классификационные признаки.** В качестве одного из первых признаков классификации видов моделирования можно выбрать степень полноты модели и разделить модели в соответствии с этим признаком на полные, неполные и приближенные. В основе полного моделирования лежит полное подобие, которое проявляется как во времени, так и в пространстве. Для неполного моделирования характерно неполное подобие модели изучаемому объекту. В основе приближенного моделирования лежит приближенное подобие, при котором некоторые стороны функционирования реального объекта не моделируются совсем.

В зависимости от характера изучаемых процессов в системе  $S$  все виды моделирования могут быть разделены на детерминированные и стохастические, статические и динамические, дискретные, непрерывные и дискретно-непрерывные. Детерминированное моделирование отображает детерминированные процессы, т. е. процессы, в которых предполагается отсутствие всяких случайных воздействий; стохастическое моделирование отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики, т. е. набор однородных реализаций. Статическое моделирование служит для описания поведения объекта в какой-либо момент времени, а динамическое моделирование отражает поведение объекта во времени.

Дискретное моделирование служит для описания процессов, которые предполагаются дискретными, соответственно непрерывное моделирование позволяет отразить непрерывные процессы в системах, а дискретно-непрерывное моделирование используется для случаев, когда хотят выделить наличие как дискретных, так и непрерывных процессов.

В зависимости от формы представления объекта (системы  $S$ ) можно выделить мысленное и реальное моделирование.

Мысленное моделирование часто является единственным способом моделирования объектов, которые либо практически нереализуемы в заданном интервале времени, либо существуют вне условий, возможных для их физического создания. Например, на базе мысленного моделирования могут быть проанализированы многие ситуации микромира, которые не поддаются физическому эксперименту. Мысленное моделирование может быть реализовано в виде наглядного, символического и математического.

При наглядном моделировании на базе представлений человека о реальных объектах создаются различные наглядные модели, отображающие явления и процессы, протекающие в объекте. В основу гипотетического моделирования исследователем закладывается некоторая гипотеза о закономерностях протекания процесса в реальном объекте, которая отражает уровень знаний исследователя об объекте и базируется на причинно-следственных связях между входом и выходом изучаемого объекта. Гипотетическое моделирование используется, когда знаний об объекте недостаточно для построения формальных моделей.

Аналоговое моделирование основывается на применении аналогий различных уровней. Наивысшим уровнем является полная аналогия, имеющая место только для достаточно простых объектов.

С усложнением объекта используют аналогии последующих уровней, когда аналоговая модель отображает несколько либо только одну сторону функционирования объекта.

Существенное место при мысленном наглядном моделировании занимает макетирование. Мысленный макет может применяться в случаях, когда протекающие в реальном объекте процессы не поддаются физическому моделированию,

либо может предшествовать проведению других видов моделирования. В основе построения мысленных макетов также лежат аналогии, однако обычно базирующиеся на причинно-следственных связях между явлениями и процессами в объекте. Если ввести условное обозначение отдельных понятий, т. е. знаки, а также определенные операции между этими знаками, то можно реализовать знаковое моделирование и с помощью знаков отображать набор понятий — составлять отдельные цепочки из слов и предложений. Используя операции объединения, пересечения и дополнения теории множеств, можно в отдельных символах дать описание какого-то реального объекта.

В основе языкового моделирования лежит некоторый тезаурус. Последний образуется из набора входящих понятий, причем этот набор должен быть фиксированным. Следует отметить, что между тезаурусом и обычным словарем имеются принципиальные различия. Тезаурус — словарь, который очищен от неоднозначности, т. е. в нем каждому слову может соответствовать лишь единственное понятие, хотя в обычном словаре одному слову могут соответствовать несколько понятий.

Символическое моделирование представляет собой искусственный процесс создания логического объекта, который замещает реальный и выражает основные свойства его отношений с помощью определенной системы знаков или символов.

Математическое моделирование. Для исследования характеристик процесса функционирования любой системы  $S$  математическими методами, включая и машинные, должна быть проведена формализация этого процесса, т. е. построена математическая модель.

Под математическим моделированием будем понимать процесс установления соответствия данному реальному объекту некоторого математического объекта, называемого математической моделью, и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта. Вид математической модели зависит как от природы реального объекта, так и задач исследования объекта и требуемой достоверности и точности решения этой задачи. Любая математическая модель, как и всякая другая, описывает реальный объект лишь с некоторой степенью приближения к действительности. Математическое моделирование для исследования характеристик процесса функционирования систем можно разделить на аналитическое, имитационное и комбинированное.

Для аналитического моделирования характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (алгебраических, интегро-дифференциальных, конечно-разностных и т. п.) или логических условий. Аналитическая модель может быть исследована следующими методами: а) аналитическим, когда стремятся получить в общем виде явные зависимости для искомых характеристик; б) численным, когда, не умея решать уравнений в общем виде, стремятся получить числовые результаты при конкретных начальных данных; в) качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость решения).

Наиболее полное исследование процесса функционирования системы можно провести, если известны явные зависимости, связывающие искомые характеристики с начальными условиями, параметрами и переменными системы  $S$ . Однако такие зависимости удастся получить только для сравнительно простых систем. При усложнении систем исследование их аналитическим методом наталкивается на значительные трудности, которые часто бывают непреодолимыми. Поэтому, желая использовать аналитический метод, в этом случае идут на существенное упрощение первоначальной модели, чтобы иметь возможность изучить хотя бы общие свойства системы. Такое исследование на упрощенной модели аналитическим методом помогает получить ориентировочные результаты для определения более точных оценок другими методами. Численный метод позволяет исследовать по сравнению с аналитическим

методом более широкий класс систем, но при этом полученные решения носят частный характер. Численный метод особенно эффективен при использовании ЭВМ.

В отдельных случаях исследования системы могут удовлетворить и те выводы, которые можно сделать при использовании качественного метода анализа математической модели. Такие качественные методы широко используются, например, в теории автоматического управления для оценки эффективности различных вариантов систем управления.

В настоящее время распространены методы машинной реализации исследования характеристик процесса функционирования больших систем. Для реализации математической модели на ЭВМ необходимо построить соответствующий моделирующий алгоритм.

При имитационном моделировании реализующий модель алгоритм воспроизводит процесс функционирования системы  $S$  во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы  $S$ .

Основным преимуществом имитационного моделирования по сравнению с аналитическим является возможность решения более сложных задач. Имитационные модели позволяют достаточно просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и др., которые часто создают трудности при аналитических исследованиях. В настоящее время имитационное моделирование — наиболее эффективный метод исследования больших систем, а часто и единственный практически доступный метод получения информации о поведении системы, особенно на этапе ее проектирования.

Когда результаты, полученные при воспроизведении на имитационной модели процесса функционирования системы  $S$ , являются реализациями случайных величин и функций, тогда для нахождения характеристик процесса требуется его многократное воспроизведение с последующей статистической обработкой информации и целесообразно в качестве метода машинной реализации имитационной модели использовать метод статистического моделирования. Первоначально был разработан метод статистических испытаний, представляющий собой численный метод, который применялся для моделирования случайных величин и функций, вероятностные характеристики которых совпадали с решениями аналитических задач (такая процедура получила название метода Монте-Карло). Затем этот прием стали применять и для машинной имитации с целью исследования характеристик процессов функционирования систем, подверженных случайным воздействиям, т. е. появился метод статистического моделирования. Таким образом, методом статистического моделирования будем в дальнейшем называть метод машинной реализации имитационной модели, а методом статистических испытаний (Монте-Карло) — численный метод решения аналитической задачи.

Метод имитационного моделирования позволяет решать задачи анализа больших систем  $S$ , включая задачи оценки: вариантов структуры системы, эффективности различных алгоритмов управления системой, влияния изменения различных параметров системы. Имитационное моделирование может быть положено также в основу структурного, алгоритмического и параметрического синтеза больших систем, когда требуется создать систему, с заданными характеристиками при определенных ограничениях, которая является оптимальной по некоторым критериям оценки эффективности.

При решении задач машинного синтеза систем на основе их имитационных моделей помимо разработки моделирующих алгоритмов для анализа фиксированной

системы необходимо также разработать алгоритмы поиска оптимального варианта системы. Далее в методологии машинного моделирования будем различать два основных раздела: статику и динамику,— основным содержанием которых являются соответственно вопросы анализа и синтеза систем, заданных моделирующими алгоритмами.

Комбинированное (аналитико-имитационное) моделирование при анализе и синтезе систем позволяет объединить достоинства аналитического и имитационного моделирования. При построении комбинированных моделей проводится предварительная декомпозиция процесса функционирования объекта на составляющие подпроцессы и для тех из них, где это возможно, используются аналитические модели, а для остальных подпроцессов строятся имитационные модели. Такой комбинированный подход позволяет охватить качественно новые классы систем, которые не могут быть исследованы с использованием только аналитического и имитационного моделирования в отдельности.

Другие виды моделирования. При реальном моделировании используется возможность исследования различных характеристик либо на реальном объекте целиком, либо на его части. Такие исследования могут проводиться как на объектах, работающих в нормальных режимах, так и при организации специальных режимов для оценки интересующих исследователя характеристик (при других значениях переменных и параметров, в другом масштабе времени и т. д.). Реальное моделирование является наиболее адекватным, но при этом его возможности с учетом особенностей реальных объектов ограничены. Например, проведение реального моделирования АСУ предприятием потребует, во-первых, создания такой АСУ, а во-вторых, проведения экспериментов с управляемым объектом, т. е. предприятием, что в большинстве случаев невозможно. Рассмотрим разновидности реального моделирования.

Натурным моделированием называют проведение исследования на реальном объекте с последующей обработкой результатов эксперимента на основе теории подобия. При функционировании объекта в соответствии с поставленной целью удастся выявить закономерности протекания реального процесса. Надо отметить, что такие разновидности натурального эксперимента, как производственный эксперимент и комплексные испытания, обладают высокой степенью достоверности.

С развитием техники и проникновением в глубь процессов, протекающих в реальных системах, возрастает техническая оснащенность современного научного эксперимента. Он характеризуется широким использованием средств автоматизации проведения, применением весьма разнообразных средств обработки информации, возможностью вмешательства человека в процесс проведения эксперимента, и в соответствии с этим появилось новое научное направление — автоматизация научных экспериментов.

Отличие эксперимента от реального протекания процесса заключается в том, что в нем могут появиться отдельные критические ситуации и определяться границы устойчивости процесса. В ходе эксперимента вводятся новые факторы и возмущающие воздействия в процессе функционирования объекта. Одна из разновидностей эксперимента — комплексные испытания, которые также можно отнести к натурному моделированию, когда вследствие повторения испытаний изделий выявляются общие закономерности о надежности этих изделий, о характеристиках качества и т. д. В этом случае моделирование осуществляется путем обработки и обобщения сведений, проходящих в группе однородных явлений. Наряду со специально организованными испытаниями возможна реализация натурального моделирования путем обобщения опыта, накопленного в ходе производственного процесса, т. е. можно говорить о производственном эксперименте. Здесь на базе теории подобия обрабатывают статистический материал по производственному процессу и получают его обобщенные характеристики.

Другим видом реального моделирования является физическое, отличающееся от натурального тем, что исследование проводится на установках, которые сохраняют природу явлений и обладают физическим подобием. В процессе физического моделирования задаются некоторые характеристики внешней среды и исследуется поведение либо реального объекта, либо его модели при заданных или создаваемых искусственно воздействиях внешней среды. Физическое моделирование может протекать в реальном и нереальном (псевдореальном) масштабах времени, а также может рассматриваться без учета времени. В последнем случае изучению подлежат так называемые «замороженные» процессы, которые фиксируются в некоторый момент времени. Наибольшую сложность и интерес с точки зрения верности получаемых результатов представляет физическое моделирование в реальном масштабе времени.

С точки зрения математического описания объекта и в зависимости от его характера модели можно разделить на модели аналоговые (непрерывные), цифровые (дискретные) и аналого-цифровые (комбинированные). Под аналоговой моделью понимается модель, которая описывается уравнениями, связывающими непрерывные величины. Под цифровой понимают модель, которая описывается уравнениями, связывающими дискретные величины, представленные в цифровом виде. Под аналого-цифровой понимается модель, которая может быть описана уравнениями, связывающими непрерывные и дискретные величины.

Особое место в моделировании занимает кибернетическое моделирование, в котором отсутствует непосредственное подобие физических процессов, происходящих в моделях, реальным процессам. В этом случае стремятся отобразить лишь некоторую функцию и рассматривают реальный объект как «черный ящик», имеющий ряд входов и выходов, и моделируют некоторые связи между выходами и входами. Чаще всего при использовании кибернетических моделей проводят анализ поведенческой стороны объекта при различных воздействиях внешней среды.

Таким образом, в основе кибернетических моделей лежит отражение некоторых информационных процессов управления, что позволяет оценить поведение реального объекта. Для построения имитационной модели в этом случае необходимо выделить исследуемую функцию реального объекта, попытаться формализовать эту функцию в виде некоторых операторов связи между входом и выходом и воспроизвести на имитационной модели данную функцию, причем на базе совершенно иных математических соотношений и, естественно, иной физической реализации процесса.

## **1.2 Лекция №2 (2 часа).**

**Тема:** «Понятие математической схемы»

### **1.2.1 Вопросы лекции:**

1. Принципы подхода в моделировании систем.
2. Классификация видов моделирования систем.

### **1.2.2 Краткое содержание вопросов:**

1. Принципы подхода в моделировании систем.

На протяжении всей истории развития теории систем предлагались и применялись различные подходы к представлению (отображению), анализу и проектированию систем.

Традиционный подход, применяющийся в математических исследованиях: определить элементы (переменные, константы) и связать их соответствующим соотношением (формулой, уравнением, системой уравнений), отображающим принцип взаимодействия элементов.

Когда задачи усложнились и такое соотношение не удавалось сразу получить, то предлагалось формировать "пространство состояний" элементов и вводить "меры близости" между элементами этого пространства. Такой подход вначале пытались применить для исследования сложных систем.

Предлагалось обследовать систему, выявить все элементы и связи между ними. Этот подход называли иногда "*перечислением*" системы. При обследовании применялись разные способы: 1) *архивный* (изучение документов и архивов предприятия); 2) *опросный*, или *анкетный* (опрос сотрудников, в том числе с помощью специально разработанных вопросников — анкет).

Однако первые же попытки применить такой подход к исследованию систем управления предприятиями и организациями показали, что "перечислить" сложную систему практически невозможно.

Учитывая трудности "перечисления" систем, предлагались различные подходы к их исследованию и проектированию.

Применение философских категорий — индуктивный и дедуктивный подходы, анализ и синтез — позволяет определить основные принципы исследования. Однако эти категории могут трактоваться и реализовываться по-разному.

Поэтому с самого начала возникновения системных теории предлагались подходы, в большей мере ориентированные на прикладные задачи. Приведем основные из них:

- в начальный период становления теории систем развивался бихевиористский подход (поведение), основанный на исследовании поведения (т.е. функционирования) систем; однако этот подход весьма трудоемок и не всегда реализуем;
- американский ученый М. Месарович предложил подходы, которые назвал *целенаправленным* и *терминальным* (от *терм* — элементарная частица, интересующая исследователя);
- польский ученый Р. Куликовски предложил называть аналогичные подходы *декомпозицией* и *композицией* системы;
- швейцарский астроном Ф. Цвикки предложил и развил *морфологический подход*, который помогает искать полезные объединения элементов путем их комбинаций;
- американская корпорация (ММ) *предложила подход к созданию сложных программ и проектов, названный "дерево целей"*;
- в практике проектирования сложных технических комплексов возникли термины "*язык моделирования*", "*язык автоматизации проектирования*", применяющиеся для отображения взаимосвязей между компонентами проекта; при разработке языков моделирования применяют *математическую логику* и *математическую лингвистику*, в которой есть удобный термин для описания структуры языка — "*тезаурус*", и подход называют иногда *лингвистическим* или *тезаурусным*;
- при исследовании и формировании структур были предложены следующие подходы: путем поиска связей между элементами или, напротив, путем устранения лишних связей.

С учетом рассмотренных подходов на основе обобщения предшествующего опыта сформировалось два основных подхода к отображению систем, первоначально предложенных для формирования структур целей:

а) "сверху" — методы *структуризации* или *декомпозиции*, *целевой* или *целенаправленный* подход;

б) "снизу" — подход, который называют *морфологическим* (в широком смысле), *лингвистическим*, *тезаурусным*, *терминальным*, методом "*языка*" системы. С помощью этого подхода определяется "пространство состояний" системы и реализуется поиск взаимосвязей (мер близости) между элементами.

Подход "снизу" можно реализовать, применяя не только комбинаторные приемы (морфологический и т.п.), но и бихевиористский подход, вариант которого при автоматизации моделирования поведения объектов в настоящее время иногда называют *процессным*, статистические методы, лежащие в основе бизнес-аналитики, методы представления и извлечения знаний, основанные на применении математической логики и математической лингвистики.

Подходы "сверху" и "снизу" называют также *аксиологическим* и *каузальным* соответственно.

Аксиологическое представление системы — отображение системы в терминах *целей* и *целевых функционалов*. Этот термин используют в тех случаях, когда необходимо выбрать подход к отображению системы на начальном этапе моделирования и противопоставить это отображение описанию системы в терминах "перечисления" элементов системы и их непосредственного влияния друг на друга, т.е. каузального представления.

Каузальное представление системы — описание системы в терминах влияния одних переменных на другие, без употребления понятий *цели* и *средств* достижения целей. Этот термин происходит от понятия "*cause*" — причина, т.е. подразумевает причинно-следственные отношения. Применяют каузальное представление в случае предварительного описания системы, когда *цель* сразу не может быть сформулирована и для отображения системы или проблемной ситуации не может быть применено аксиологическое представление.

В 1970—1980-е гг. при проектировании организационных структур были предложены три подхода к решению этой проблемы.

- *Нормативно-функциональный* подход направлен на унификацию организационных форм управления в рамках отрасли. Разработка типовых организационных структур явилась первым шагом на пути внедрения принципов их научно обоснованного построения. Однако ориентация на типовую номенклатуру функций управления и структурных управленческих подразделений не позволяет учесть особенностей конкретных предприятий и условий их деятельности.

- *Функционально-технологический* подход основан на рационализации потоков информации и технологии ее обработки, на формировании и анализе организационно-технологических процедур подготовки и реализации управленческих решений. Этот подход обеспечивает возможность достаточно полно учесть особенности конкретного предприятия (организации), отличается гибкостью и универсальностью. Вместе с тем он

характеризуется высокой трудоемкостью, использованием стабильной номенклатуры сложившихся функций управления, подчинением оргструктуры схеме документооборота.

- *Системно-целевой* подход заключается в построении структуры целей, определении на ее основе функций управления и их организационном оформлении. Преимущества этого подхода заключаются в возможности учитывать особенности объекта управления и условия его деятельности, изменять и расширять состав функций, проектировать разнообразные организационно-правовые формы предприятий. Трудности в использовании подхода связаны с проблемой перехода от совокупности целей и функций к составу и подчиненности структурных звеньев, обеспечивающих их реализацию.

*Обобщающий подход "сверху"*, называемый целевым, целенаправленным, системно-целевым, основан на структуризации или декомпозиции системы *в пространстве*. Этот подход позволяет расчленить исходную большую неопределенность на более обозримые и выбрать методы их анализа и проектирования, сохраняя целостность представления об исследуемой системе или решаемой проблеме на основе иерархической структуры (древовидной, стратифицированной).

*Подход "снизу"*, основанный на анализе пространства состояний, поиске "мер близости" между компонентами с помощью различных, в том числе статистических, методов, морфологического моделирования, отличается большой трудоемкостью. В настоящее время для анализа пространства состояний разработаны методы представления и извлечения знаний, основанные на применении статистических методов, математической логики и математической лингвистики.

В настоящее время для проектирования систем широкое применение нашел подход, кратко называемый *процессным*. Этот подход, который можно считать развитием *функционально-технологического подхода* основан на структуризации во времени, на представлении процессов в форме графов.

Применение функционально-технологического подхода долгое время было практически нереализуемым из-за большой трудоемкости, отсутствия правил и средств автоматизации формирования графов, отображающих процессы в системах. В 1990-е гг. была разработана методология SADT (*Structured Analysis and Design* — структурный анализ и проектирование; предложена Дугласом Россом), представляющая собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. На ее основе разработаны и стали широко применяться функционально-ориентированные и объектно-ориентированные CASE-2 и RAD-3 технологии. Компьютерная реализация методологии SADT получила название IDEF (*Icam Definition*). Основными структурными моделями являются модели процессов IDEF0 и IDEF3, модель данных IDEF1X4. Созданы стандарты IDEF и DFD, ориентированные на анализ процессов (в том числе бизнес-процессов). Для реализации моделей применяются автоматизированные средства — BPWin, ARIS, язык UML (*Unified Modeling Language* — унифицированный язык моделирования). Популярность САБЕ-методологии и технологий базируется на разработке принципов и автоматизации формирования процессов, на развитии методов их формирования (на основе анализа "жизненного цикла" производства, обслуживания или других процессов, причинно-следственных связей и т.п.), что и обеспечило развитие *процессного подхода*, преимущества которого заключаются в возможности учитывать особенности конкретного объекта и условий его деятельности.

## 2. Классификация видов моделирования систем



*Детерминированное моделирование* отображает процессы, в которых предполагается отсутствие всяких случайных воздействий; *стохастическое моделирование* отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики, т. е. набор однородных реализаций. *Статическое моделирование* служит для описания поведения объекта в какой-либо момент времени, *адинамическое моделирование* отражает поведение объекта во времени. *Дискретное моделирование* служит для описания процессов, которые предполагаются дискретными, соответственно непрерывное моделирование позволяет отразить непрерывные процессы в системах, а *дискретно-непрерывное моделирование* используется для случаев, когда хотят выделить наличие как дискретных, так и непрерывных процессов.

В зависимости от формы представления объекта (системы  $S$ ) можно выделить мысленное и реальное моделирование.

*Мысленное моделирование* часто является единственным способом моделирования объектов, которые либо практически нереализуемы в заданном интервале времени, либо существуют вне условий, возможных для их физического создания. Например, на базе мысленного моделирования могут быть проанализированы многие ситуации микромира, которые не поддаются физическому эксперименту. Мысленное моделирование может быть реализовано в виде *наглядного, символического и математического*.

При *наглядном моделировании* на базе представлений человека о реальных объектах создаются различные наглядные модели, отображающие явления и процессы, протекающие в объекте. В основу *гипотетического моделирования* исследователем закладывается некоторая гипотеза о закономерностях протекания процесса в реальном объекте, которая отражает уровень знаний исследователя об объекте и базируется на причинно-следственных связях между входом и выходом изучаемого объекта. Гипотетическое моделирование используется, когда знаний об объекте недостаточно для построения формальных моделей.

*Аналоговое моделирование* основывается на применении аналогий различных уровней. Наивысшим уровнем является полная аналогия, имеющая место только для достаточно простых объектов. С усложнением объекта используют аналогии последующих уровней, когда аналоговая модель отображает несколько либо только одну сторону функционирования объекта.

Существенное место при мысленном наглядном моделировании занимает *макетирование*. Мысленный макет может применяться в случаях, когда протекающие в реальном объекте процессы не поддаются физическому моделированию, либо может предшествовать проведению других видов моделирования. Если ввести условное обозначение отдельных понятий, т. е. знаки, а также определенные операции между этими знаками, то можно реализовать *знаковое моделирование* и с помощью знаков отображать набор понятий — составлять отдельные цепочки из слов и предложений. Используя операции объединения, пересечения и дополнения теории множеств, можно в отдельных символах дать описание какого-то реального объекта.

В основе *языкового моделирования* лежит некоторый тезаурус. Последний образуется из набора входящих понятий, причем этот набор должен быть фиксированным. Следует отметить, что между тезаурусом и обычным словарем имеются принципиальные различия. Тезаурус — словарь, в котором каждому слову может соответствовать лишь

единственное понятие, хотя в обычном словаре одному слову могут соответствовать несколько понятий.

*Символическое моделирование* представляет собой искусственный процесс создания логического объекта, который замещает реальный и выражает основные свойства его отношений с помощью определенной системы знаков или символов.

**Математическое моделирование.** Под *математическим моделированием* будем понимать процесс установления соответствия данному реальному объекту некоторого математического объекта, называемого математической моделью, и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта. Вид математической модели зависит как от природы реального объекта, так и задач исследования объекта и требуемой достоверности и точности решения этой задачи.

Для *аналитического моделирования* характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (алгебраических, интегро - дифференциальных, конечно-разностных и т.п.) или логических условий.

*Имитационное моделирование* позволяет по исходным данным получить сведения о состоянии процесса в определенные моменты времени, дающие возможность оценить характеристики системы  $S$ .

### 1.3 Лекция №3 (2 часа).

**Тема:** «Дискретно – стохастические модели»

#### 1.3.1 Вопросы лекции:

1. Математическая схема общего вида.
2. Типовые математические схемы.
3. Непрерывно-детерминированные модели ( $D$ -схемы)

#### 1.3.2 Краткое содержание вопросов:

1. Математическая схема общего вида.

Для исследования характеристик процесса функционирования любой системы математическими методами должна быть проведена формализация этого процесса, т.е. построена математическая модель. Эта задача решается с помощью математических схем.

**Математическая схема** представляет собой звено при переходе от содержательного к формальному описанию процесса функционирования системы с учётом воздействия внешней среды, т.е. имеет место цепочка «описательная модель – математическая схема – математическая модель». Схематично процесс формализации представлен на рис.1.

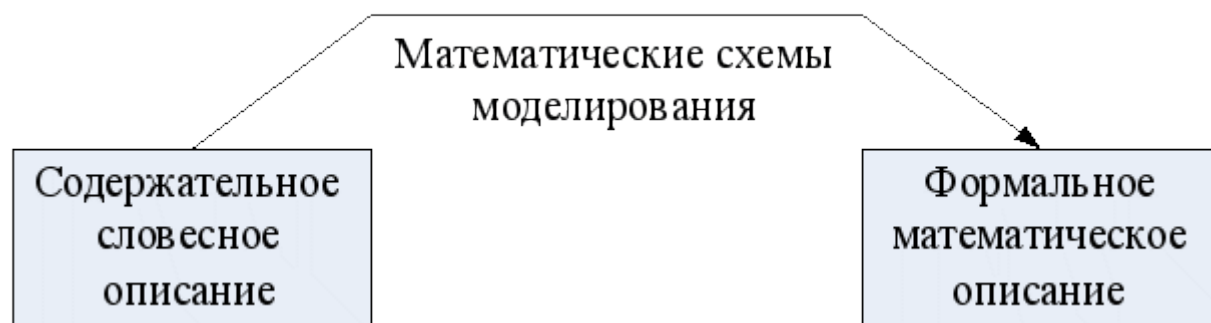


Рис.1 Схема процесса формализации

Введение понятия математической схемы позволяет рассматривать математику не как метод расчёта, а как метод мышления, как средство формулирования понятий, что является важным при переходе от словесного описания системы к формальному представлению процесса её функционирования в виде математической модели. **Исходной информацией** при построении математических моделей процессов функционирования систем служат данные о назначении и условиях работы исследуемой системы, причем уровень абстрагирования зависит от круга тех вопросов, на которые исследователь системы хочет получить ответы с помощью модели.

В отличие от содержательного словесного описания, образующего описательную модель, формальное описание процесса функционирования системы, представляющее собой математическую модель, не допускает неоднозначной интерпретации, так как представляет собой правило, которое необходимо выполнить для получения результата.

При пользовании математической схемой в первую очередь решается вопрос об **адекватности отображения в виде конкретных схем реальных процессов в исследуемой системе**. Кроме того, при построении математической модели необходимо решить вопрос об её полноте. Полнота модели регулируется, в основном, выбором границы между системой и внешней средой. Также должна быть решена задача упрощения модели, которая помогает выделить основные свойства системы, отбросив второстепенные.

В практике моделирования используются математическая схема общего вида и типовые математические схемы. **Математическая схема общего вида** позволяет формализовать широкий класс систем. **Типовые математические схемы**, включающие D–схемы, F–схемы, P–схемы, Q–схемы и A–схемы, не обладают общностью, но имеют преимущества простоты и наглядности.

При использовании математической схемы общего вида модель объекта моделирования, т.е. исследуемой системы, представляется в виде множества величин, описывающих процесс функционирования реальной системы и образующих четыре непересекающихся подмножества (рис. 2):

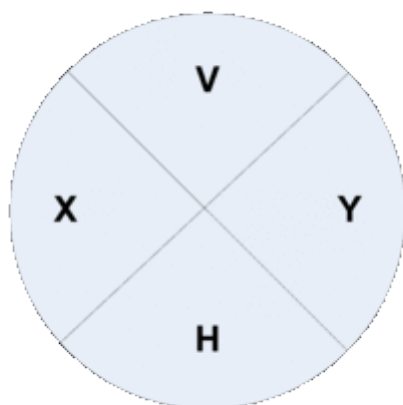


Рис.2 Модель, построенная на основе математической схемы общего вида

- подмножество совокупности входных воздействий на систему  
 $X = \{x_i\}, i = \overline{1, n_x}$
- подмножество совокупности воздействий внешней среды на систему  
 $V = \{v_l\}, l = \overline{1, n_v}$
- подмножество совокупности внутренних (собственных) параметров системы  
 $H = \{h_k\}, k = \overline{1, n_h}$

- подмножество совокупности выходных характеристик системы  

$$Y = \{y_j\}, j = \overline{1, n_y}$$

При моделировании системы входные воздействия  $X$ , воздействия внешней среды  $V$  и внутренние параметры системы  $H$  являются **независимыми (экзогенными) переменными**, а выходные характеристики  $Y$  — **зависимыми (эндогенными) переменными**.

Процесс функционирования системы описывается во времени оператором, который в общем случае преобразует экзогенные переменные в эндогенные в соответствии с соотношением вида  $Y(t) = F_s(X, V, H, t)$ . Эта зависимость называется законом функционирования системы и обозначается  $F_s$ . В общем случае закон функционирования системы  $F_s$  может быть задан в виде функции, функционала, логических условий, в алгоритмической и табличной формах или в виде словесного правила соответствия.

## 2. Типовые математические схемы.

На первоначальных этапах исследования используются типовые схемы: дифференциальные уравнения, конечные и вероятностные автоматы, системы массового обслуживания, сети Петри и т. д.

В качестве детерминированных моделей, когда при исследовании случайные факторы не учитываются, для представления систем, функционирующих в непрерывном времени, используются дифференциальные, интегральные, интегродифференциальные и другие уравнения, а для представления систем, функционирующих в дискретном времени, — конечные автоматы и конечно-разностные схемы.

В качестве стохастических моделей (при учете случайных факторов) для представления систем с дискретным временем используются вероятностные автоматы, а для представления системы с непрерывным временем — системы массового обслуживания и т. д.

Таким образом, при построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно-детерминированный (например, дифференциальные уравнения); дискретно-детерминированный (конечные автоматы); дискретно-стохастический (вероятностные автоматы); непрерывно-стохастический (системы массового обслуживания); обобщенный, или универсальный (агрегативные системы).

## 3. Непрерывно-детерминированные модели (D–схемы)

Непрерывно-детерминированные модели используются при описании и исследовании объектов, для которых отличительными характеристиками являются:

- отсутствие случайностей при работе и управлении объектом моделирования;
- явления в объектах моделирования рассматривают как непрерывные процессы, то есть те, в которых основная переменная, часто это время, является непрерывной величиной.

Модели построенные по этой схеме чаще всего ориентированы на изучение динамики рассматриваемого объекта (отсюда и название D-схема). Поэтому характерным примером использования такого рода схемы являются дифференциальные уравнения.

## 4. Дискретно-детерминированные модели (F–схемы).

ДДМ являются предметом рассмотрения теории автоматов (ТА). ТА - раздел теоретической кибернетики, изучающей устройства, перерабатывающие дискретную

информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени.

#### 1.4 Лекция №3 (2 часа).

Тема: «Дискретно – детерминированные модели»

##### 1.4.1 Вопросы лекции:

1. Конечные автоматы.
2. Классификация конечных автоматов.
3. Способы задания работы автоматов.

##### 1.4.2 Краткое содержание вопросов:

#### 1. Конечные автоматы.

**Конечным автоматом** называется автомат, у которого множество внутренних состояний и входных сигналов (а следовательно, и множество выходных сигналов) являются конечными множествами

Конечный автомат имеет множество внутренних состояний и входных сигналов, являющихся конечными множествами. Автомат задаётся F- схемой:  $F = \langle Z, X, Y, j, y, z_0 \rangle$ ,

где  $Z, X, Y$  - соответственно конечные множества входных, выходных сигналов (алфавитов) и конечное множество внутренних состояний (алфавита).  $z_0 \in Z$  - начальное состояние;  $j(z, x)$  - функция переходов;  $y(z, x)$  - функция выхода.

Автомат функционирует в дискретном автоматном времени, моментами которого являются такты, т. е. примыкающие друг к другу равные интервалы времени, каждому из которых соответствуют постоянные значения входного, выходного сигнала и внутреннего состояния. Абстрактный автомат имеет один входной и один выходной каналы.

Для задания F - автомата необходимо описать все элементы множества  $F = \langle Z, X, Y, j, y, z_0 \rangle$ , т. е. входной, внутренний и выходной алфавиты, а также функции переходов и выходов. Для задания работы F - автоматов наиболее часто используются табличный, графический и матричный способ.

В табличном способе задания используется таблицы переходов и выходов, строки которых соответствуют входным сигналам автомата, а столбцы - его состояниям.

#### 2. Классификация конечных автоматов.

По способу формирования функций выходов выделяют автоматы Мили и Мура.

В автомате Мили (англ. *Mealy machine*) функция выходов  $\lambda$  определяет значение выходного символа по классической схеме абстрактного автомата. Математическая модель автомата Мили и схема рекуррентных соотношений не отличаются от математической модели и схемы рекуррентных соотношений абстрактного автомата. Таким образом, можно дать следующее определение:

Конечным детерминированным автоматом типа Мили называется совокупность пяти объектов

$$A = (S, X, Y, \delta, \lambda),$$

где  $S, X$  и  $Y$  — конечные непустые множества, а  $\delta$  и  $\lambda$  — отображения вида:

$$\delta : S \times X \rightarrow S \text{ и } \lambda : S \times X \rightarrow Y$$

со связью элементов множеств  $S, X$  и  $Y$  в абстрактном времени  $T = 0, 1, 2, \dots$  уравнениями:

$$\begin{aligned} s(t+1) &= \delta(s(t), x(t)), \\ y(t) &= \lambda(s(t), x(t)), t \in T \end{aligned}$$

(Отображения  $\delta$  и  $\lambda$  получили названия, соответственно функции переходов и функции выходов автомата A).

Особенностью автомата Мили является то, что функция выходов является двухаргументной и символ в выходном канале  $y(t)$  обнаруживается только при наличии символа во входном

Зависимость выходного сигнала *только от состояния* представлена в автоматах типа Мура (англ. *Moore machine*). В автомате Мура функция выходов определяет значение выходного символа только по одному аргументу — состоянию автомата. Эту функцию называют также функцией меток, так как она каждому состоянию автомата ставит метку на выходе.

Функциональная схема автомата Мура

Конечным детерминированным автоматом типа Мура называется совокупность пяти объектов:  $A = (S, X, Y, \delta, \mu)$ ,

где  $S, X, Y$  и  $\delta$  — соответствуют определению автомата типа Мили, а  $\mu$  является отображением вида:  $\mu : S \rightarrow Y$ ,

с зависимостью состояний и выходных сигналов во времени уравнением:

$$y(t) = \mu(s(t)), t \in T.$$

Особенностью автомата Мура является то, что символ  $y(t)$  в выходном канале существует все время, пока автомат находится в состоянии  $s(t)$ .

Для любого автомата Мура существует автомат Мили, реализующий ту же самую функцию. И наоборот: для любого автомата Мили существует соответствующий автомат Мура (возможно, со сдвигом по времени, т.е.  $\mu(s(t+1)) = \lambda(s(t), x(t)), t \in T$ ).

### 3. Способы задания работы автоматов.

Чтобы задать конечный автомат  $S$ , необходимо описать все элементы множества  $S = \{A, X, Y, \delta, \lambda\}$ , т.е. необходимо описать входной, выходной алфавиты и алфавит состояний, а также функции переходов  $\delta$  и выходов  $\lambda$ . При этом среди множества  $A = \{a_0, a_1, \dots, a_n\}$  необходимо выделить начальное состояния  $a_0$ , в котором автомат находится в момент времени  $t = 0$ . Существует несколько способов задания работы автомата, но наиболее часто используются табличный и графический. Табличный способ. При этом способе автомат Мили описывается двумя таблицами: таблицей переходов и таблицей выходов. Таблица переходов  $x_j \backslash a_i$   $a_0 \dots a_n$   $x_1 \delta(a_0, x_1) \dots \delta(a_n, x_1) \dots \dots \dots x_m \delta(a_0, x_m) \dots \delta(a_n, x_m)$  Таблица выходов  $x_j \backslash a_i$   $a_0 \dots a_n$   $x_1 \lambda(a_0, x_1) \dots \lambda(a_n, x_1) \dots \dots \dots x_m \lambda(a_0, x_m) \dots \lambda(a_n, x_m)$  Строки этих таблиц соответствуют входным сигналам  $x(t)$ , а столбцы — состояниям. На пересечении столбца  $a_i$  и строки  $x_j$  в таблице переходов ставится состояние  $a_s = \delta[a_i, x_j]$ , в которое автомат перейдет из состояния  $a_i$  под воздействием сигнала  $x_j$ ; а в таблице выходов — соответствующий этому переходу выходной сигнал  $u_g = \lambda[a_i, x_j]$ . Совмещенная таблица переходов и выходов автомата Мили:  $x_j \backslash a_i$   $a_0 \dots a_n$   $x_1 \delta(a_0, x_1) / \lambda(a_0, x_1) \dots \delta(a_n, x_1) / \lambda(a_n, x_1) \dots \dots \dots x_m \delta(a_0, x_m) / \lambda(a_0, x_m) \dots \delta(a_n, x_m) / \lambda(a_n, x_m)$  Задание таблиц переходов и выходов полностью описывает работу конечного автомата, поскольку задаются не только сами функции переходов и выходов, но и также все три алфавита: входной, выходной и алфавит состояний. Для задания автомата Мура требуется одна таблица, поскольку в этом автомате выходной сигнал однозначно определяется состоянием автомата. Отмеченная таблица переходов автомата Мура:  $u_g \lambda(a_0) \dots \lambda(a_n)$   $x_j \backslash a_s$   $a_0 \dots a_n$   $x_1 \delta(a_0, x_1) \dots \delta(a_n, x_1) \dots \dots \dots x_m \delta(a_0, x_m) \dots \delta(a_n, x_m)$  Автомат Мили Автомат Мура В этой таблице каждому столбцу приписан, кроме состояния  $a_i$ , еще и выходной сигнал  $y(t) = \lambda(a(t))$ , соответствующий этому состоянию. Таблица переходов автомата Мура называется отмеченной потому, что каждое состояние отмечено выходным сигналом. Приведем примеры табличного задания автоматов Мили и Мура: По этим таблицам можно найти реакцию автомата на любое входное слово. Например. Для автомата Мили: Для автомата Мура:  $x_1 x_2 x_2 x_1 \dots x_1 x_2 x_2 x_1 \dots$

Графический способ задания автомата (задание автомата с помощью графа). Этот способ основан на использовании ориентированных связных графов. Вершины графов соответствуют состояниям автомата, а дуги – переходам между ними. Две вершины графа  $a_i$  и  $a_s$  соединяются дугой, направленной от  $a_i$  к  $a_s$ , если в автомате имеется переход из  $a_i$  в  $a_s$ , т.е.  $a_s = \delta(a_i, x_j)$ . В автомате Мили дуга отмечается входным сигналом  $x_j$ , вызвавшим переход, и выходным сигналом  $y_g$ , который возникает при переходе. Внутри кружочка, обозначающего вершину графа, записывается состояние.

### 1.5 Лекция №3 (2 часа).

**Тема:** «Дискретно – стохастические модели»

#### 1.5.1 Вопросы лекции:

1. Вероятностные автоматы

#### 1.5.2 Краткое содержание вопросов:

1. Вероятностные автоматы

*Вероятностный автомат* - математическая модель, в функционировании которой участвует некоторый случайный процесс, а вероятность перехода из одного состояния в другое зависит от входных воздействий и предшествующих состояний.

Найти *вероятностные автоматы*, равные произведению и сумме автоматов  $A$  и  $B$ .

О любой конечный вероятностный автомат, ( $1/2$ -распознающий  $L$ , также требует не менее  $p$  состояний.

В *вероятностных автоматах* появление того или иного символа имеет ту или иную вероятность. Имеет смысл сразу же указать три возможных типа автомата, качественно отличающихся друг от друга в функциональном отношении. Это означает, что возбуждение одних и тех же ВХ ДВ автомата, в какой бы момент времени оно ни произошло, всегда вызывает возбуждение соответствующих им одних и тех же выходов.

Начнем с *вероятностных автоматов* и выделим категорию множеств со случайными отображениями. Морфизмы  $\alpha: A \rightarrow B$  - случайные отображения, которые можно интерпретировать как стохастические матрицы. Строки матрицы  $\alpha$  нумеруются элементами множества  $A$ , а столбцы - элементами из  $B$ .

Наконец, для *вероятностного автомата* можно ввести понятие образа, как совокупности элементов, для которых вероятность появления реакции, соответствующей данному обра-

Наконец, для *вероятностного автомата* можно ввести понятие образа, как совокупности элементов, для которых вероятность появления реакции, соответствующей данному образу, больше, чем вероятность др. реакций, и задача состоит тогда в нахождении автомата, имеющего ту же систему образов, что и данный.

Такие системы называются *вероятностными автоматами*.

Как пишет Р. Г. Бухараев, *вероятностный автомат* является математической моделью весьма распространенной физической системы. Оставляя в стороне вопрос о существовании физических систем с индетерминированным поведением, не описываемым статистическими законами, мы можем привести множество примеров, подтверждающих это высказывание. Даже детерминированные конструкции из-за случайных сбоев проявляют стохастическое поведение. Очень важным примером стохастической системы является детерминированная система с очень большим числом состояний, поведение которой ненаблюдаемо в деталях, благодаря чему возникает гомоморфный (в широком смысле) образ этой системы, являющейся стохастической системой. Посредством конструирования вероятностной модели мы в состоянии учесть фактор неопределенности наших знаний о действительных состояниях физической системы, вызванный принципиальным несовершенством процесса измерения

Используя такой способ задания *вероятностных автоматов*, можно ввести теоретико-множественные операции объединения и пересечения вероятностных автоматов по аналогии с операциями над детерминированными автоматами, накладывая, правда, некоторые ограничения на множество стохастических матриц, которые делают довольно узким класс вероятностных автоматов, к которым применимы данные операции. При этом выводы, полученные для теоретико-множественных операций над детерминированными автоматами, справедливы для операций над вероятностными автоматами, разумеется, при сохранении накладываемых ограничений на стохастические матрицы. Поэтому, не останавливаясь на этих операциях, перейдем сразу к алгебраическим операциям умножения, суммирования и суперпозиции, которые применимы к произвольным вероятностным автоматам.

Определим операцию суперпозиции над *вероятностными автоматами*. Так как рассматриваются автоматы без выходов, то полагаем, что алфавит состояний первого автомата совпадает с входным алфавитом второго автомата, к которым применяется операция суперпозиции.

В том случае, когда *вероятностные автоматы* рассматриваются с точки зрения представления событий подобно тому, как это имеет место в случае детерминированных автоматов, при задании вероятностного автомата  $A$  указывается множество  $Q$  отмеченных состояний.

## **1.6 Лекция №5 (2 часа).**

**Тема:** «Последовательность разработки и машинной реализации моделей»

### **1.6.1 Вопросы лекции:**

1. Методологические аспекты моделирования.
2. Требования пользователя к модели.
3. Этапы моделирования систем.

### **1.6.2 Краткое содержание вопросов:**

1. Методологические аспекты моделирования.

С развитием вычислительной техники наиболее эффективным методом исследования больших систем стало машинное моделирование, без которого невозможно решение многих крупных народнохозяйственных проблем. Поэтому одной из актуальных задач подготовки специалистов является освоение теории и методов математического моделирования с учетом требований системности, позволяющих не только строить модели изучаемых объектов, анализировать их динамику и возможность управления машинным экспериментом с моделью, но и судить в известной мере об адекватности



создаваемых моделей исследуемым системам, о границах применимости и правильно организовать моделирование систем на современных средствах вычислительной техники

Прежде чем рассматривать математические, алгоритмические, программные и прикладные аспекты машинного моделирования, необходимо изучить общие методологические аспекты для широкого класса математических моделей объектов, реализуемых на средствах вычислительной техники. Моделирование с использованием средств вычислительной техники (ЭВМ, АВМ, ГВК) позволяет исследовать механизм явлений,

протекающих в реальном объекте с большими или малыми скоростями, когда в натурных экспериментах с объектом трудно (или невозможно) проследить за изменениями, происходящими в течение короткого времени, или когда получение достоверных результатов сопряжено с длительным экспериментом. При необходимости машинная модель дает возможность как бы «растягивать» или «сжимать» реальное время, так как машинное моделирование связано с понятием системного времени, отличного от реального. Кроме того, с помощью машинного моделирования в диалоговой системе можно обучать персонал АСОИУ принятию решений в управлении объектом, например при организации деловой игры, что позволяет выработать необходимые практические навыки реализации процесса управления [12, 29, 46, 53].

Сущность машинного моделирования системы состоит в проведении на вычислительной машине эксперимента с моделью, которая представляет собой некоторый программный комплекс, описывающий формально и (или) алгоритмически поведение элементов системы  $S$  в процессе ее функционирования, т. е. в их взаимодействии друг с другом и внешней средой  $E$ . Машинное моделирование с успехом применяют в тех случаях, когда трудно четко сформулировать критерий оценки качества функционирования системы и цель ее не поддается полной формализации, поскольку позволяет сочетать программно-технические возможности ЭВМ со способностями человека мыслить неформальными категориями. В дальнейшем основное внимание будет уделено моделированию систем на универсальных ЭВМ как наиболее эффективному инструменту исследования и разработки АСОИУ различных уровней, а случаи использования АВМ и ГВК будут специально оговариваться.

## 2. Требования пользователя к модели.

Сформулируем основные требования, предъявляемые к модели  $M$  процесса функционирования системы  $S$ .

1. Полнота модели должна предоставлять пользователю возможность получения необходимого набора оценок характеристик системы с требуемой точностью и достоверностью.

2. Гибкость модели должна давать возможность воспроизведения различных ситуаций при варьировании структуры, алгоритмов и параметров системы.

3. Длительность разработки и реализации модели большой системы должна быть по возможности минимальной при учете ограничений на имеющиеся ресурсы.

4. Структура модели должна быть блочной, т. е. допускать возможность замены, добавления и исключения некоторых частей без переделки всей модели.

5. Информационное обеспечение должно предоставлять возможность эффективной работы модели с базой данных систем определенного класса.

6. Программные и технические средства должны обеспечивать эффективную (по быстродействию и памяти) машинную реализацию модели и удобное общение с ней пользователя.

7. Должно быть реализовано проведение целенаправленных (планируемых) машинных экспериментов с моделью системы с использованием аналитико-имитационного подхода при наличии ограниченных вычислительных ресурсов.

С учетом этих требований рассмотрим основные положения, которые справедливы при моделировании на ЭВМ систем  $S$ , а также их подсистем и элементов. При машинном моделировании системы  $S$  характеристики процесса ее функционирования определяются на основе модели  $M$ , построенной исходя из имеющейся исходной информации об объекте моделирования. При получении новой информации об объекте его модель пересматривается и уточняется с учетом новой информации, т. е. процесс моделирования, включая разработку и машинную реализацию модели, является итерационным. Этот итерационный процесс продолжается до тех пор, пока не будет получена модель  $M$ , которую можно считать адекватной в рамках решения поставленной задачи исследования и проектирования системы  $S$ .

Моделирование систем с помощью ЭВМ можно использовать в следующих случаях [5, 29, 41]: а) для исследования системы  $S$  до того, как она спроектирована, с целью определения чувствительности характеристики к изменениям структуры, алгоритмов и параметров объекта моделирования и внешней среды; б) на этапе проектирования системы  $S$  для анализа и синтеза различных вариантов системы и выбора среди конкурирующих такого варианта, который удовлетворял бы заданному критерию оценки эффективности системы при принятых ограничениях; в) после завершения проектирования и внедрения системы, т. е. при ее эксплуатации, для получения информации, дополняющей результаты натурных испытаний (эксплуатации) реальной системы, и для получения прогнозов эволюции (развития) системы во времени.

Существуют общие положения, применяемые ко всем перечисленным случаям машинного моделирования. Даже в тех случаях, когда конкретные способы моделирования отличаются друг от друга и имеются различные модификации моделей, например в области машинной реализации моделирующих алгоритмов с использованием конкретных программно-технических средств, в практике моделирования систем можно сформулировать общие принципы, которые могут быть положены в основу методологии машинного моделирования [29, 35, 46].

Итак, общие требования к моделям.

1. Модель должна быть *актуальной*. Это значит, что модель должна быть нацелена на важные для лиц, принимающих решения, проблемы.
2. Модель должна быть *результативной*. Это значит, что полученные результаты моделирования могут найти успешное применение. Данное требование может быть реализовано только в случае правильной формулировки требуемого результата.
3. Модель должна быть *достоверной*. Это значит, что результаты моделирования не вызовут сомнения. Данное требование тесно связано с понятием адекватности, то есть, если модель неадекватна, то она не может давать достоверных результатов.
4. Модель должна быть *экономичной*. Это значит, что эффект от использования результатов моделирования превышает расходы ресурсов на ее создание и исследование.

Эти требования (обычно их называют внешними) выполнимы при условии обладания моделью внутренними свойствами.

Модель должна быть:

1. *Существенной*, т. е. позволяющей вскрыть сущность поведения системы, вскрыть неочевидные, нетривиальные детали.
2. *Мощной*, т. е. позволяющей получить широкий набор существенных сведений.
3. *Простой* в изучении и использовании, легко просчитываемой на компьютере.
4. *Открытой*, т. е. позволяющей ее модификацию.

В заключение темы сделаем несколько замечаний. Трудно ограничить область применения математического моделирования. При изучении и создании промышленных и военных систем практически всегда можно определить цели, ограничения и

предусмотреть, чтобы конструкция или процесс подчинялись естественным, техническим и (или) экономическим законам.

Круг аналогий, которые можно использовать в качестве моделей, также практически неограничен. Следовательно, надо постоянно расширять свое образование в конкретной области, но, в первую очередь, в математике.

В последние десятилетия появились проблемы с неясными и противоречивыми целями, диктуемыми политическими и социальными факторами. *Математическое моделирование* в этой области пока еще проблематично. Что это за проблемы? Защита от загрязнения окружающей среды; предсказаний извержений вулканов, землетрясений, цунами; рост городов; руководство боевыми действиями и ряд других. Но, тем не менее, "процесс пошел", прогресс не остановим, и *проблемы моделирования* таких сверхсложных систем постоянно находят свое разрешение. Здесь следует отметить лидирующую роль отечественных ученых и, в первую очередь, академика Н. Н. Моисеева, его учеников и последователей.

### 3. Этапы моделирования систем.

Рассмотрим основные этапы моделирования системы  $S$ , к числу которых относятся: построение концептуальной модели системы и ее формализация; алгоритмизация модели системы и ее машинная реализация; получение и интерпретация результатов моделирования системы.

Взаимосвязь перечисленных этапов моделирования систем и их составляющих (подэтапов) может быть представлена в виде сетевого

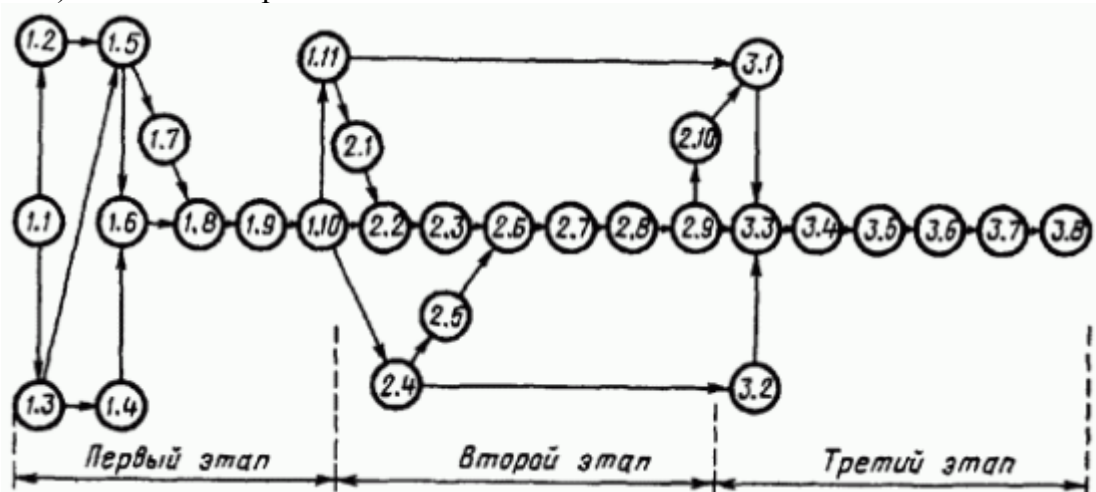


Рис. 3.1. Взаимосвязь этапов моделирования систем

графика, показанного на рис. 3.1. Перечислим эти подэтапы: 1.1-постановка задачи машинного моделирования системы; 1.2 — анализ задачи моделирования системы; 1.3-определение требований к исходной информации об объекте моделирования и организация ее сбора; 1.4 — выдвижение гипотез и принятие предположений; 1.5 — определение параметров и переменных модели; 1.6 — установление основного содержания модели; 1.7 — обоснование критериев оценки эффективности системы; 1.8 — определение процедур аппроксимации; 1.9 — описание концептуальной модели системы; 1.10 — проверка достоверности концептуальной модели; 1.11 — составление технической документации по первому этапу; 2.1 — построение логической схемы модели; 2.2 — получение математических соотношений; 2.3 — проверка достоверности модели системы; 2.4 — выбор инструментальных средств для моделирования; 2.5 — составление плана выполнения работ по программированию; 2.6 — спецификация и построение схемы программы;

2.7 — верификация и проверка достоверности схемы программы;

2.8 — проведение программирования модели; 2.9 — проверка достоверности программы; 2.10 — составление технической документации по второму этапу; 3.1 — планирование машинного эксперимента с моделью системы; 3.2 — определение требований к вычислительным средствам; 3.3 — проведение рабочих расчетов; 3.4 — анализ результатов моделирования системы; 3.5 — представление результатов моделирования; 3.6 — интерпретация результатов моделирования; 3.7 — подведение итогов моделирования и выдача рекомендаций; 3.8 — составление технической документации по третьему этапу.

Таким образом, процесс моделирования системы  $S$  сводится к выполнению перечисленных подэтапов, сгруппированных в виде трех этапов. На этапе построения концептуальной модели <sup>[1]</sup> и ее формализации проводится исследование моделируемого объекта с точки зрения выделения основных составляющих процесса его

функционирования, определяются необходимые аппроксимации и получается обобщенная схема модели системы  $S$ , которая преобразуется в машинную модель  $M$  на втором этапе моделирования путем последовательной алгоритмизации и программирования модели. Последний третий этап моделирования системы сводится к проведению согласно полученному плану рабочих расчетов на ЭВМ с использованием выбранных программно-технических средств, получению и интерпретации результатов моделирования системы  $S$  с учетом воздействия внешней среды  $E$ . Очевидно, что при построении модели и ее машинной реализации при получении новой информации возможен пересмотр ранее принятых решений, т. е. процесс моделирования является итерационным. Рассмотрим содержание каждого из этапов более подробно.

### **1.7 Лекция №7 (2 часа).**

**Тема:** «Алгоритмизация модели и ее машинная реализация»

#### **1.7.1 Вопросы лекции:**

1. Типы состояний
2. Обозначение символов
3. Подэтапы второго этапа моделирования

#### **1.7.2 Краткое содержание вопросов:**

1. Типы состояний

Алгоритмизация моделей систем и их машинная реализация на втором этапе моделирования — математическая модель, сформированная на первом этапе, воплощается в конкретную машинную модель. Этот этап представляет собой этап практической деятельности, направленной на реализацию идей и математических схем в виде машинной модели  $M$  процесса функционирования системы  $S$ .

При рассмотрении процессов функционирования некоторых систем можно обнаружить, что для них характерны два типа состояний:

1) особые, присущие процессу функционирования системы только в некоторые моменты времени (моменты поступления входных или управляющих воздействий, возмущений внешней среды и т. п.);

2) не особые, в которых процесс находится все остальное время.

Особые состояния характерны тем, что функции состояний изменяются скачкообразно, а между особыми состояниями изменение координат происходит плавно и непрерывно или не происходит совсем.

Не особые состояния при моделировании не рассматриваются.

Удобной формой представления логической структуры моделей процессов функционирования систем и машинных программ является схема. На различных этапах моделирования составляются обобщенные и детальные логические схемы моделирующих алгоритмов, а также схемы программ.

Обобщенная (укрупненная) схема моделирующего алгоритма задает общий порядок действий при моделировании системы она показывает, что необходимо выполнить на очередном шаге моделирования, например обратиться к датчику случайных чисел.

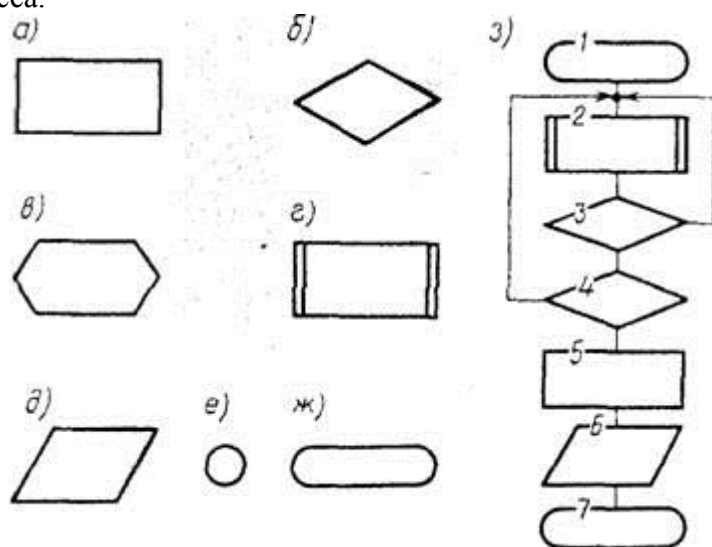
Детальная схема моделирующего алгоритма содержит уточнения, отсутствующие в обобщенной схеме, она показывает не только, что следует выполнить на очередном шаге моделирования системы, но и как это выполнить.

Логическая схема моделирующего алгоритма представляет собой логическую структуру модели процесса функционирования системы. Логическая схема указывает упорядоченную во времени последовательность логических операций, связанных с решением задачи моделирования. Схема программы отображает порядок программной реализации моделирующего алгоритма с использованием конкретного математического обеспечения.

Схема программы представляет собой интерпретацию логической схемы моделирующего алгоритма разработчиком программы на базе конкретного алгоритмического языка.

Логическая схема алгоритма и схема программы могут быть выполнены как в укрупненной, так и в детальной форме. Для начертания этих схем используется набор символов, определяемых ГОСТ 19.701 — 90 (ИСО 5807 — 85) «Единая система программной документации».

Некоторые наиболее употребительные в практике моделирования на ЭВМ символы показаны на рис. 2., где изображены основные, специфические и специальные символы процесса.



К ним относятся: основной символ:

а — процесс — символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться); специфические символы процесса:

б — решение — символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа (соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути);

в — подготовка — символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы);

г — предопределенный процесс — символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле);

д — ручная операция — символ отображает любой процесс, выполняемый человеком; специальные символы:

е — соединитель — символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте (соответствующие символы-соединители должны содержать одно и то же уникальное обозначение);

ж — терминатор — символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы алгоритма, внешнее использование или пункт назначения данных).

## 2. Обозначение символов

Удобной формой представления логической структуры моделей является схема. На различных этапах моделирования составляются обобщенные и детальные логические схемы моделирующих алгоритмов, а также схемы программ. Обобщенная (укрупненная) схема моделирующего алгоритма задает общий порядок действий при моделировании систем без каких-либо уточняющих деталей.

Обобщенная схема показывает, что необходимо выполнить на очередном шаге моделирования. Детальная схема моделирующего алгоритма содержит уточнения, отсутствующие в обобщенной схеме.

Детальная схема показывает не только то, что следует выполнить на очередном шаге моделирования системы, но и как это выполнить.

Логическая схема моделирующего алгоритма представляет собой логическую структуру модели процесса функционирования системы S. Логическая схема указывает упорядоченную во времени последовательность логических операций, связанных с решением задачи моделирования. Схема программы отображает порядок программной реализации моделирующего алгоритма с использованием конкретных математического обеспечения и алгоритмического языка.

Логическая схема алгоритма и схема программы могут быть выполнены как в укрупненной, так и в детальной форме. Наиболее употребительные в практике моделирования на ЭВМ символы показаны на рис. 3.3, где изображены основные, специфические и специальные символы процесса.

К ним относятся: основной символ: а — процесс; специфические символы процесса: б — решение; в — подготовка; г — предопределенный процесс; д — ручная операция; специальные символы: е — соединитель; ж — терминатор. Пример изображения схемы моделирующего алгоритма показан на рис. 3.3, з. Обычно схема является наиболее удобной формой представления структуры моделирующих алгоритмов, например в виде граф-схемы (рис. 3.3, и).

Здесь — начало, — конец, — вычисление, — формирование, — проверка условия, — счетчик, — выдача результата, , где g — общее число операторов моделирующего алгоритма.

В качестве пояснения к граф-схеме алгоритма в тексте дается раскрытие содержания операторов, что позволяет упростить представление алгоритма, но усложняет работу с ним.

## 3. Подэтапы второго этапа моделирования

2.1. Построение логической схемы модели. Рекомендуется строить модель по блочному принципу. При построении блочной модели проводится разбиение процесса

функционирования системы на отдельные достаточно автономные подпроцессы. Блоки такой модели бывают двух типов: основные и вспомогательные. Каждый основной блок соответствует некоторому реальному подпроцессу, имеющему место в моделируемой системе S, а вспомогательные блоки представляют собой лишь составную часть машинной модели, они не отражают функции моделируемой системы и необходимы лишь для машинной реализации, фиксации и обработки результатов моделирования.

2.2. Получение математических соотношений. Одновременно с выполнением подэтапа построения логической схемы модели необходимо получить, если это возможно, математические соотношения в виде явных функций, т. е. построить аналитические модели. Схема машинной модели Мм должна представлять собой полное отражение заложенной в модели концепции и иметь:

- а) описание всех блоков модели с их наименованиями;
- б) единую систему обозначений и нумерацию блоков;
- в) отражение логики модели процесса функционирования системы;
- г) задание математических соотношений в явном виде.

2.3. Проверка достоверности модели системы. Эта проверка является первой из проверок, выполняемых на этапе реализации модели. Поэтому определение достоверности модели можно считать наиболее важной проблемой при моделировании систем. При этом проверяются:

- а) возможность решения поставленной задачи;
- б) точность отражения замысла в логической схеме;
- в) полнота логической схемы модели;
- г) правильность используемых математических соотношений.

2.4. Выбор инструментальных средств для моделирования. Вопрос о выборе ЭВМ сводится к обеспечению следующих требований:

- а) наличие необходимых программных и технических средств;
- б) доступность выбранной ЭВМ для разработчика модели;
- в) обеспечение всех этапов реализации модели;
- г) возможность своевременного получения результатов.

2.5. Составление плана выполнения работ по программированию. План при использовании универсальной ЭВМ должен включать в себя:

- а) выбор языка (системы) программирования модели;
- б) указание типа ЭВМ и необходимых для моделирования устройств;
- в) оценку примерного объема необходимой оперативной и внешней памяти;
- г) ориентировочные затраты машинного времени на моделирование;
- д) предполагаемые затраты времени на программирование и отладку программы на ЭВМ.

2.6. Спецификация и построение схемы программы. Наличие логической блок-схемы модели позволяет построить схему программы, которая должна отражать:

- а) разбиение модели на блоки, подблоки и т. д.;
- б) особенности программирования модели;
- в) проведение необходимых изменений;
- г) возможности тестирования программы;
- д) оценку затрат машинного времени;
- е) форму представления входных и выходных данных.

2.7. Верификация и проверка достоверности схемы программы. Верификация программы — доказательство того, что поведение программы соответствует спецификации на программу. Эта проверка является второй на этапе машинной реализации модели системы.

2.8. Проведение программирования модели. При достаточно подробной схеме программы, которая отражает все операции логической схемы модели, можно приступить к программированию модели.

2.9. Проверка достоверности программы. Эта последняя проверка на этапе машинной реализации модели, которую необходимо проводить:

- а) обратным переводом программы в исходную схему;
- б) проверкой отдельных частей программы при решении различных тестовых задач;
- в) объединением всех частей программы и проверкой ее в целом на контрольном примере моделирования варианта системы S.

2.10. Составление технической документации по второму этапу. Для завершения этапа машинной реализации модели необходимо составить техническую документацию, содержащую:

- а) логическую схему модели и ее описание;
- б) адекватную схему программы принятые обозначения;
- в) полный текст программы;
- г) перечень входных и выходных величин с пояснениями;
- д) инструкцию по работе с программой;
- е) оценку затрат машинного времени на моделирование с указанием требуемых ресурсов ЭВМ.

Таким образом, на этом этапе разрабатывается схема модели системы S, проводится ее алгоритмизация и программирование с использованием конкретных программно-технических средств.

## **1.8 Лекция №8-9 (4 часа).**

**Тема:** «Имитационное моделирование»

### **1.7.1 Вопросы лекции:**

1. Общие сведения
2. Блоки
3. Транзакты
4. Списки
5. Оборудование.

### **1.8.2 Краткое содержание вопросов:**

1. Общие сведения

Специализированные языки имитационного моделирования делятся на языки схемных символов и операторные языки. Соответственно двум видам имитации они также классифицируются на:

- языки моделирования непрерывных процессов;
- языки моделирования дискретных процессов.

Языки моделирования непрерывных процессов обеспечивают блочное построение моделей и при этом обладают мощностью и удобством алгебраической и логической алгоритмизации.

Языки моделирования дискретных процессов делятся на 4 категории:

- языки, ориентированные на события,
- языки, ориентированные на действия,
- языки, ориентированные на процессы,



- языки, ориентированные на потоки сообщений (транзактов). В ряде случаев данная категория не выделяется отдельно, а рассматривается как подмножество языков процессов.

В языке, ориентированном на события, каждое событие представляется в виде мгновенного события в модельном времени с таким регламентом, который обеспечивает наступление события именно в тот момент времени, когда динамическое состояние показывает, что сложились условия для его появления. Для того, чтобы появление событий в модельном времени подчинялось установленному регламенту, исполнительная программа автоматически определяет их очередности. В число языков, ориентированных на события, входят Симскрипт, Симком и Симпак.

В языках, ориентированных на зависящие от времени действия, эти действия представляются в модельном времени как мгновенные. Программируя на этих языках, пользователь не регламентирует действия программы, а лишь указывает на условия, при которых они могут произойти. В подобных языках программа составляется из двух частей: обеспечивающей проверку и обеспечивающей действие. Исполнительные программы, просматривают набор всех условий перед сдвигом времени на очередной интервал, что позволяет определить, могут ли произойти какие-либо события. Перед очередным сдвигом модельного времени происходит сканирование всех программ по действиям для определения, какая из них включена в работу. Исполнение команд изменения состояния и сдвига времени в той части программы, которая обеспечивает действие, разрешается тогда, когда выполняются все контролируемые условия. Если хоть одно из них осталось невыполненным, команды действия пропускаются. Циклическое сканирование всех программ имитации действий дает возможность учитывать вероятности включения каждой из них и все возможные взаимодействия. Языки, ориентированные на действия: CSL, ESP, Форсим IV, GSP и Милитран.

Создание языков, ориентированных на процессы, представляет собой попытку объединить краткость языков, ориентированных на действия, с эффективностью языков, ориентированных на события. Процессы могут быть гибкими и регламентированными, их можно прерывать или дополнять подчиненными подпроцессами, можно программировать задержки в них самих или в других процессах до появления необходимых условий и т.д. Наиболее важное свойство языков, ориентированных на процессы, заключается в том, что написанная на подобном языке программа работает так же, как несколько программ, управляемых независимо одна от другой либо посредством просмотра действий, либо посредством регламентирования событий. Особенностью программирования, позволяющей реализовать это свойство, является использование некоторой точки повторной активизации (точки реактивации); это, в сущности, указатель, сообщающий программе обработки данных, с какого места должно продолжаться ее выполнение после исполнения тех или иных команд, задерживающих этот процесс. Исполнительная программа в этих языках реализует еще более важную функцию, чем в языках, ориентированных на действия или события. В эту категорию входят такие языки, как Симула, OPS и SOL.

Блок-схемы алгоритмов, являясь важным инструментом описания логики поведения и взаимодействия элементов системы, моделируемой на одном из языков для дискретных систем, как в программировании, так и в теории конструирования моделей подобных систем играют небольшую роль. Применяя эти языки, пользователь строит имитационную модель на основе операторов, которые описывают условия, необходимые для выполнения действий, результаты этих действий, а также определяют временные взаимосвязи между элементами системы и действиями, в которых эти элементы участвуют. В языках, ориентированных на

процессы, моделирование системы проводится путем движения транзактов (сообщений) через стилизованные блоки действий. По мере прохода транзактов через блоки происходит сдвиг системного времени; решения принимаются в результате воспроизведения логики функционирования моделируемой системы. Языки, попадающие в эту категорию, называют транзактными. Программисту, знакомому с идеей, положенной в основу блок-схем алгоритмов, на этих языках программировать нетрудно. В этом случае специализированные блоки соединяются в структуры, отображающие логику действий и различные потоки моделируемой системы. Сама система представляется в виде блоков; моделирующая программа создает транзакты, передвигает их по определенным блокам и производит действия, связанные с каждым блоком. Поскольку эти специальные блоки модели являются одновременно и основными программными операторами, составление модели на языке блок-схем равносильно написанию программы. Однако легкость освоения этих языков покупается ценой потери гибкости.

К языкам этой категории принадлежат GPSS и BOSS. Язык моделирования дискретных систем GPSS (GeneralPurposeSystemSimulator) является одним из самых распространенных в мире языков имитационного моделирования. Данный язык был разработан фирмой IBM в начале 70-х годов XX в., а в настоящее время система моделирования GPSS/PC (различных версий) является торговой маркой фирмы MINUTEMAN Software.

GPSS — это интерпретирующая языковая система, применяющаяся, в основном, для имитации пространственно-временного движения объектов различной природы при фиксированной структуре блочной схемы. В основу языка GPSS положены результаты исследований разнообразных дискретных систем, показавшие, что любую систему можно описать с помощью необходимого набора абстрактных элементов, или объектов. Формальными основными моделями, для имитации которых используется GPSS, являются системы массового обслуживания, конечные и вероятностные автоматы, сети Петри, агрегаты. Логические правила поведения таких систем описываются аналогичным в каждом случае набором операций.

Основными структурами данных языковой системы GPSS являются блоки, транзакты, списки, оборудование.

## 2. Блоки

Для задания объектов моделируемой системы и операций над ними в моделирующей системе GPSS используется *совокупность* блоков, имеющих специальное обозначение и формат описания (записи). Каждому блоку соответствует графическое изображение на блок-схеме, а чтобы применить язык моделирования GPSS, блоки блок-схемы заменяются соответствующими операторами GPSS. Каждый блок системы имеет входы и выходы, с помощью которых осуществляется их связь в модели. Блок является обязательным элементом любой модели на GPSS. В системе GPSS/PC версии 2.0 всего имеется 46 блоков.

Стрелки между блоками указывают маршруты потоков сообщений (транзактов). В блоках могут происходить события четырех основных типов:

1. Создание или уничтожение транзактов;
2. Изменение числового атрибута объекта;
3. Задержка сообщения на определенный период времени;
4. Изменение маршрута транзакта.

## 3. Транзакты

*Транзакт* — это некоторое сообщение, которое поступает извне на вход системы и подлежит обработке. В терминах систем массового обслуживания транзактам можно сопоставить требования, сообщения или заявки, поступающие в систему на обслуживание. Транзакт — обязательный элемент каждой модели на GPSS. Транзакты создаются в определенных точках модели, продвигаются интерпретатором через блоки, а затем уничтожаются, производя при этом воздействия на объекты, образующие модель системы. С каждым транзактом в GPSS связаны *параметры*. Параметры могут использоваться для связи конкретных числовых данных с этим транзактом.

Сообщения движутся от блока к блоку так, как движутся элементы, которые они представляют. Каждое продвижение считается событием, которое должно происходить в конкретный момент времени. Интерпретатор GPSS автоматически определяет моменты наступления событий. В тех случаях, когда событие не может произойти, хотя момент его наступления подошел (например, при попытке занять устройство, когда оно уже занято), транзакт прекращает продвижение до снятия блокирующего условия.

Структурно транзакт в моделируемой системе не определяется, его существование предполагается после генерации и прохождения по заданным компонентам системы. Работа системы происходит под действием транзактов, которые определяют собой пространственно-временное продвижение реальных объектов в реальной системе. Само моделирование заключается в продвижении сообщений (транзактов) от блока к блоку аналогично функционированию реальной системы.

#### 4. Списки

В GPSS транзакты хранятся в списках. *Списки* представляют собой структуры данных, в которых размещена информация о транзактах. Списки относятся к элементам внутренней организации GPSS, они обеспечивают внутреннюю логику работы модели системы. Сообщение в любой момент времени находится в одном из пяти списков: текущих событий, будущих событий, прерываний, пользователя, синхронизации.

В *списке текущих событий* транзакт находится либо в активном состоянии (в способности к продвижению к следующему блоку), либо в состоянии задержки (когда движение транзакта блокируется какими-либо объектами модели). Транзакты расположены в порядке убывания приоритета. Внутри класса транзакты с одинаковыми приоритетами располагаются в порядке поступления их в список.

*Список будущих событий* содержит транзакты, которые могут начать движение в модели в будущие моменты времени. Такие транзакты размещаются в списке строго в порядке возрастания времени начала движения по отношению к значению текущего времени. Приоритеты не влияют на порядок транзактов в рассматриваемом списке.

*Список прерываний* содержит транзакты, обслуживание которых (дальнейшее продвижение по модели) прервано на одном или более устройствах. В списке прерываний транзакты расположены в порядке поступления. Транзакты находятся в списке прерываний до тех пор, пока не будут сняты все условия прерываний.

*Список пользователя* представляет собой список транзактов, удаленных из списка текущих событий и помеченных как временно неактивных.

*Список синхронизации* или список синхронизируемых транзактов содержит транзакты, которые в данный момент находятся в состоянии сравнения.

Списки призваны оптимизировать работу моделирующей системы, представляющей собой программу-интерпретатор. В задачу интерпретатора

входит, прежде всего, определение ближайшего следующего события, которое должно произойти. После того как определено следующее событие (события), интерпретатор должен выполнить ряд проверок, чтобы определить, существуют ли в системе условия для наступления этого события, если условия существуют, то интерпретатор переходит к обработке события. При этом интерпретатор может собирать некоторую статистическую информацию. Эта статистика впоследствии является частью выходных данных, по которым можно судить о работе моделируемой системы. Основной цикл программы выявления ближайшего следующего события и его обработка повторяется столько раз, сколько нужно для завершения моделирования. Таким образом, основной задачей, которую выполняет интерпретатор, является просмотр возможных событий. Также должен учитываться вопрос об изменении условного времени при организации просмотра возможных событий. Общую внутреннюю последовательность событий можно разбить на три фазы: изменение значения условного времени, просмотр списка текущих событий, движение транзактов.

## 5. Оборудование.

Для того, чтобы моделировать обслуживающие приборы реальных систем, в GPSS используются объекты типа «оборудование». К объектам этого типа относятся:

- устройства,
- логические ключи.

Устройства моделируют реальные объекты, в которых может происходить обработка транзактов. Устройства могут быть одноканальными и многоканальными.

Одноканальные устройства (в дальнейшем их будем называть устройствами) представляют собой оборудование, которое в любой момент времени может быть занято только одним сообщением. Если другое сообщение попытается захватить устройство, то это сообщение задерживается до тех пор, пока устройство не освободится. Интерпретатор автоматически подсчитывает общее время занятости устройства, что позволяет определить коэффициент использования каждого устройства, и общее число сообщений, занимавших устройство, что позволяет вычислить среднее время занятости устройства одним сообщением.

Многоканальные устройства представляют собой объекты типа «оборудование» для параллельной обработки, они могут быть использованы несколькими транзактами (сообщениями) одновременно. Пользователь определяет емкость каждого многоканального устройства, используемого в модели, а интерпретатор ведет учет числа единиц многоканальных устройств, занятых в каждый момент времени. Если сообщение пытается занять больше единиц многоканального устройства, чем свободно в данный момент, обработка этого сообщения задерживается до того момента, пока в многоканальном устройстве освободится достаточный объем. Интерпретатор автоматически ведет подсчет числа транзактов, входящих в многоканальное устройство, а также определяет среднее число единиц многоканальных устройств, занятых одним сообщением, и среднее время пребывания транзакта в многоканальном устройстве. Эти статистические данные позволяют определить, насколько эффективно используются в системе объекты параллельной обработки и достаточна ли их емкость.

Ранее происходившие в моделируемой системе события могут заблокировать и изменить движение последующих событий, поэтому в GPSS предусмотрена возможность прерывания устройств. Для моделирования таких ситуаций введены логические ключи. Управление ключами осуществляется транзактами, продвигающимися по модели. Транзакт может устанавливать логические ключи в положение «включено» или «выключено». Впоследствии состояние ключа (свободно,

занято, прервано, не прервано) может быть проверено другими транзактами для определения пути их следования. Транзакт может установить ключ (установка ключа соответствует состоянию «включено»), сбросить его (состояние «выключено») или инвертировать.

Также к объектам GPSS относятся арифметические и булевские переменные, функции, ячейки сохраняемых величин и матрицы ячеек сохраняемых величин, а также очереди:

- **арифметические переменные** позволяют вычислять арифметические выражения. В выражении переменной используют операторы, арифметические действия и вызовы библиотечных функций;

- **булевские переменные** позволяют пользователю проверять в одном блоке GPSS одновременно несколько условий, исходя из состояния или значения объектов и их атрибутов;

- **функции** позволяют проводить вычисления непрерывных или дискретных функциональных зависимостей между аргументом функции (независимая величина) и зависимым значением функции. Широко применяются, например, для задания случайного интервала времени между генерацией двух сообщений;

- **ячейки сохраняемых величин и матрицы ячеек сохраняемых величин** используются для хранения некоторой числовой информации. Например, значение, занесенное в ячейку, может представлять собой длину очереди в какой-то конкретной точке модели. Любое сообщение может произвести запись информации в эти объекты. Впоследствии записанную в эти объекты информацию может считывать любое сообщение. Например, в случае моделирования работы порта для построения зависимости ожидания судов в очереди определенной длины от времени достаточно через каждые X единиц времени (X задается пользователем) заносить значение длины очереди в ячейки (каждое значение в отдельную ячейку). В конце счета значения этих ячеек будут распечатаны, и по ним можно построить искомую зависимость;

в моделируемой системе движение потока транзактов может быть задержано из-за недоступности оборудования. В этом случае задержанные сообщения ставятся в **очередь**. Учет очередей составляет одну из основных функций интерпретатора GPSS. Пользователь может определить специальные точки в модели, в которых нужно собирать статистику об очередях. Тогда интерпретатор автоматически будет собирать статистику об очередях (длину очереди; среднее время пребывания в очереди; общее число сообщений, поступающих в очередь; число сообщений, задерживающихся в каждой очереди; число сообщений, прошедших какую-либо точку модели без задержки).

Помимо стандартной статистической информации относительно устройств, памяти и очередей, накапливаемой интерпретатором автоматически, пользователь может собрать дополнительную статистическую информацию путем регистрации значений контролируемых параметров, их классификации и определения частот попаданий в каждый класс.

Описанные выше объекты в модели системы можно считать присутствующими неявно, т.е. их существование предполагается. Пользователю нет необходимости определять каждый объект и описывать, как его использовать. Он только описывает те действия, которые производятся над этими объектами. Анализируя выходную статистику, пользователь может установить, какое влияние эти действия оказали на объект.

Для того, чтобы обеспечить правильную временную последовательность событий в модели исследуемой системы, организованы **часы**, хранящие значения текущего момента в модели. В отличие от обычных часов, часы в GPSS меняют значение только для того, чтобы указать время наступления ближайшего события. Например, если текущее значение часов модели равно 2, а очередное событие должно

наступить в момент времени 7, то значение часов увеличивается сразу на 5 единиц. Единицы времени в модели не обязательно должны быть конкретными единицами времени, такими, как секунда или час. Основной единицей времени в модели можно выбрать любую единицу, которая позволит получить необходимую точность моделирования.

## **2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ**

### **2.1 Практическая работа №1 (2 часа).**

**Тема:** «Предмет теории моделирования. Роль и место моделирования в исследовании систем»

**2.1.1 Цель работы:** Получение практических навыков, необходимых для математического анализа и синтеза конструкций, устройств, технологических процессов.

**2.1.2 Задачи работы:** Овладение основными математическими методами и теориями, применяемыми в решение типовых задач в области проектирования.

#### **2.1.3 Описание (ход) работы:**

Построить пространство событий для испытания «бросание 6-гранного игрального кубика». Определить вероятности случайных событий A, B, C и D:

A – выпадение четного числа очков;

B – выпадение нечетного числа очков;

C – выпадение не более 2-х очков;

D – выпадение более 5-ти очков.

*Решение.* Построить пространство событий – это значит, определить все возможные элементарные события для испытания. В данной задаче элементарными событиями являются:

$a_1$  – выпадение числа 1;

$a_2$  – выпадение числа 2;

$a_3$  – выпадение числа 3;

$a_4$  – выпадение числа 4;

$a_5$  – выпадение числа 5;

$a_6$  – выпадение числа 6.

Пространство событий представляет собой множество:

$$\Omega = \{a_1, a_2, a_3, a_4, a_5, a_6\}.$$

Следовательно, общее число элементарных событий  $n = 6$ .

а) Случайному событию А «выпадение четного числа очков» благоприятствуют элементарные события  $a_2, a_4$  и  $a_6$ . Представим случайное событие А в виде множества:

$$A = \{a_2, a_4, a_6\}.$$

Следовательно, число благоприятствующих элементарных событий  $m = 3$ .

Подставляя в классическую формулу определения вероятности значения для  $n$  и  $m$ , получим:

$$P(A) = m/n = 3/6 = 1/2.$$

б) Случайному событию В «выпадение нечетного числа очков» благоприятствуют элементарные события  $a_1, a_3$  и  $a_5$ . Представим случайное событие В в виде множества:

$$B = \{a_1, a_3, a_5\}.$$

Следовательно, число благоприятствующих элементарных событий  $m = 3$ .

С помощью классической формулы определения вероятности получим:

$$P(B) = m/n = 3/6 = 1/2.$$

в) Случайному событию С «выпадение не более 2-х очков» благоприятствуют элементарные события  $a_1$  и  $a_2$ :

$$C = \{a_1, a_2\}.$$

Число благоприятствующих элементарных событий  $m = 2$ .

Вероятность события С:

$$P(C) = m/n = 2/6 = 1/3.$$

г) Случайному событию D «выпадение более 5-ти очков» благоприятствует одно элементарное событие  $a_6$ :

$$D = \{a_6\}.$$

$$m = 1$$

Вероятность события D:

$$P(D) = m/n = 1/6.$$

### **Пример 2.**

Построить пространство событий для испытания «бросание 2-х игральных кубиков – белого и черного». Элементарным событием является пара чисел, выпавших на верхних гранях кубиков. Определить вероятности случайных событий А, В и С:

А – сумма чисел равна 6;

В – сумма чисел более 9;

С – выпал дубль.

*Решение.* Для задачи с двумя объектами (кубиками) пространство событий удобно представить в виде таблицы, где по вертикали располагаются возможные события для первого объекта, а по горизонтали – возможные значения для другого объекта. В клетках таблицы записываются элементарные события:

События (2-й объект)	События (1-ый объект)				
	$x_1$	$x_2$	$x_3$	...	$x_k$
$y_1$	$a_{11}$	$a_{12}$	$a_{13}$	...	$a_{1k}$
$y_2$	$a_{21}$	$a_{22}$	$a_{23}$	...	$a_{2k}$

...	...	...	...	...	...
$y_r$	$a_{r1}$	$a_{r2}$	$a_{r3}$	...	$a_{rk}$

В условиях задачи пространству событий соответствует таблица (элементарными событиями является пара чисел):

Количество очков, выпавших на белом кубике	Количество очков, выпавших на черном кубике					
	1	2	3	4	5	6
1	1 : 1	1 : 2	1 : 3	1 : 4	1 : 5	1 : 6
2	2 : 1	2 : 2	2 : 3	2 : 4	2 : 5	2 : 6
3	3 : 1	3 : 2	3 : 3	3 : 4	3 : 5	3 : 6
4	4 : 1	4 : 2	4 : 3	4 : 4	4 : 5	4 : 6
5	5 : 1	5 : 2	5 : 3	5 : 4	5 : 5	5 : 6
6	6 : 1	6 : 2	6 : 3	6 : 4	6 : 5	6 : 6

или

Количество очков, выпавших на белом кубике	Количество очков, выпавших на черном кубике					
	1	2	3	4	5	6
1	$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$	$a_{16}$
2	$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$	$a_{26}$
3	$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$	$a_{35}$	$a_{36}$
4	$a_{41}$	$a_{42}$	$a_{43}$	$a_{44}$	$a_{45}$	$a_{46}$
5	$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{55}$	$a_{56}$
6	$a_{61}$	$a_{62}$	$a_{63}$	$a_{64}$	$a_{65}$	$a_{66}$

Общее количество элементарных событий соответствует количеству клеток:  $n = 36$ .

а) Случайному событию А «сумма чисел равна 6» благоприятствуют элементарные события  $a_{15}$ ,  $a_{24}$ ,  $a_{33}$ ,  $a_{42}$  и  $a_{51}$ . Представим случайное событие А в виде множества:

$$A = \{a_{15}, a_{24}, a_{33}, a_{42}, a_{51}\}.$$

Следовательно, число благоприятствующих элементарных событий  $m = 5$ .

Подставляя в классическую формулу определения вероятности значения для  $n$  и  $m$ , получим:

$$P(A) = m/n = 5/36.$$

б) Случайному событию В «сумма чисел более 9» благоприятствуют все элементарные события, при которых сумма чисел равна 10, 11 или 12:  $a_{46}$ ,  $a_{55}$ ,  $a_{64}$ ,  $a_{56}$ ,  $a_{65}$  и  $a_{66}$ .

$$B = \{a_{46}, a_{55}, a_{56}, a_{64}, a_{65}, a_{66}\}.$$

Число благоприятствующих элементарных событий  $m = 6$ .

Вероятность события В:

$$P(B) = m/n = 6/36 = 1/6.$$

в) Случайному событию С «выпал дубль» благоприятствуют все элементарные события, при которых на обоих кубиках выпало одинаковое число очков:  $a_{11}$ ,  $a_{22}$ ,  $a_{33}$ ,  $a_{44}$ ,  $a_{55}$  и  $a_{66}$ .

$$C = \{a_{11}, a_{22}, a_{33}, a_{44}, a_{55}, a_{66}\}.$$

Число благоприятствующих элементарных событий  $m = 6$ .

Вероятность события С:



$$P(C) = m/n = 6/36 = 1/6.$$

**Указание:** если испытание заключается в том, что два объекта последовательно извлекаются из одной и той же совокупности (два шара из одного ящика, две карты из одной колоды и т.п.), при построении пространства событий необходимо исключить повторы (один и тот же шар нельзя вынуть дважды). Тогда пространство событий будет выглядеть следующим образом:

События (извлекается 2-й объект)	События (извлекается 1-ый объект)				
	$x_1$	$x_2$	$x_3$	...	$x_k$
$x_1$	—	$a_{12}$	$a_{13}$	...	$a_{1k}$
$x_2$	$a_{21}$	—	$a_{23}$	...	$a_{2k}$
$x_3$	$a_{31}$	$a_{32}$	—	...	$a_{3k}$
...	...	...	...	—	...
$x_k$	$a_{k1}$	$a_{k2}$	$a_{k3}$	...	—

#### 1) ЗАДАНИЯ «Классическое определение вероятности»

**1** Построить пространство событий для испытания «две монеты бросают последовательно». Определить вероятности случайных событий А, В и С:

А – хотя бы на одной из монет выпадет "решка";

В – на одной из монет выпадет "решка", а на другой – "орел";

С – на первой монете выпадет "решка", а на второй – "орел".

**2** В ящике находятся 15 шаров, пронумерованных от 1 до 15.

Построить пространство событий для испытания «из ящика вытаскивают один шар». Определить вероятности случайных событий А, В, С и D:

А – вытаскен шар с номером 5;

В – вытаскен шар с номером, не большим 5;

С – вытаскен шар с номером, кратным 5;

D – вытаскен шар с номером, кратным 3 или 4.

**3** В ящике находятся 5 шаров, пронумерованных от 1 до 5.

Построить пространство событий для испытания «из ящика вытаскивают последовательно два шара». Элементарным событием является пара номеров на вытасканных шарах. Определить вероятности случайных событий А, В и С:

А – один из шаров имеет номер 5;

В – сумма номеров на вытасканных шарах равна 5;

С – сумма номеров на вытасканных шарах является кратной 3.

**4** В группе 20 студентов. У четырех студентов фамилия начинается на букву «А», у трех – на «О», у трех – на «У», у двух – на «К», у двух – на «П», у остальных – на «С».

Построить пространство событий для испытания «преподаватель вызывает одного студента». Определить вероятности случайных событий А, В, С и D:

- А – фамилия студента начинается на букву «О» или «П»;  
В – фамилия студента начинается на согласную букву;  
С – фамилия студента начинается на букву, которая расположена в первой половине алфавита (до буквы «Р»);  
D – фамилия студента начинается на согласную букву, которая расположена в первой половине алфавита (до буквы «Р»).

**5** Набирая номер телефона, абонент забыл последнюю цифру.

Построить пространство событий для испытания «абонент набирает последнюю цифру наугад». Определить вероятности случайных событий А, В, С и D:

- А – абонент набрал нечетную цифру;  
В – абонент набрал цифру меньше 4;  
С – абонент набрал нечетную цифру больше 5;  
D – абонент набрал нужную цифру.

**6** В классе 10 мальчиков и 12 девочек. Из девочек половина – отличницы. Из мальчиков отличниками являются трое.

Построить пространство событий для испытания «учитель вызывает к доске одного школьника». Определить вероятности случайных событий А, В, С и D:

- А – будет вызвана девочка;  
В – будет вызвана девочка-отличница;  
С – будет вызван школьник, учащийся на «отлично»;  
D – будет вызван мальчик, учащийся на отлично, или любая из девочек.

**7** В ящике находятся 6 шаров: 3 белых, 2 синих и 1 красный.

Построить пространство событий для испытания «из ящика вытаскивают последовательно два шара». Определить вероятности случайных событий А, В и С:

- А – хотя бы один из шаров – синий;  
В – оба шара – белые;  
С – первым вытащен красный шар.

**8** Школьник купил 20 лотерейных билетов, из которых 3 подарил маме, 3 – папе, 2 – бабушке, 2 – дедушке, 1 – брату, остальные оставил себе.

Построить пространство событий для испытания «один из приобретенных лотерейных билетов оказался выигрышным». Определить вероятности случайных событий А, В, С и D:

- А – выигрышный билет достался школьнику;  
В – выигрышный билет достался родителям;  
С – выигрышный билет не достался брату;  
D – выигрышный билет не достался никому.

**9** Школьник купил 6 лотерейных билетов, из которых 1 подарил маме, 1 – папе, 2 – брату и 2 оставил себе.

Построить пространство событий для испытания «два из приобретенных лотерейных билетов оказались выигрышными». Определить вероятности случайных событий А, В и С:

- А – оба выигрышных билета достались школьнику;
- В – ни один из выигрышных билетов не достался брату;
- С – хотя бы один из выигрышных билетов достался родителям.

## **2.2 Практическая работа №2 (2 часа).**

**Тема:** «Общие сведения. Математические схемы моделирования систем. Классификация видов моделирования»

**2.2.1 Цель работы:** Освоение методов математического моделирования объектов и систем управления.

**2.2.2 Задачи работы:** Отработка практических навыков компьютерного математического моделирования при проектировании и исследовании различных объектов и систем управления.

### **2.2.3 Описание (ход) работы:**

Новый ресторан может быть построен в южной или северной части города. Если ресторан будет построен в северной части города, то вероятность его успешной работы в течение первого года равна 90%. Если построить ресторан в южной части, то вероятность успешной работы в первый год равна 65%. Северный участок удастся приобрести с вероятностью 40%.

Найти вероятность следующих событий:

- а) ресторан будет построен в южной части города;
- б) ресторан будет построен в южной части города и его работа в первый год будет успешной;
- в) ресторан будет построен в северной части города, но его работа в первый год не будет успешной;
- г) работа ресторана в первый год не будет успешной при условии, что он будет построен в северной части города;
- д) работа ресторана в течение первого года будет успешной;
- е) ресторан построен в северной части города при условии, что его работа в первый год оказалась успешной.

*Решение.*

Обозначим события:

- $A_1$  – ресторан будет построен в северной части города;
- $A_2$  – ресторан будет построен в южной части города;
- $B_1$  – ресторан будет работать успешно в течение первого года;
- $B_2$  – ресторан не будет успешно работать в течение первого года.

Обозначим исходные вероятности:

$P(A_1) = 40\% = 0,4$  – вероятность того, что ресторан будет построен в северной части города;

$P(B_1|A_1) = 90\% = 0,9$  – условная вероятность того, что ресторан будет работать успешно в течение первого года при условии, что он будет построен в северной части города;

$P(B_1|A_2) = 65\% = 0,65$  – условная вероятность того, что ресторан будет работать успешно в течение первого года при условии, что он будет построен в южной части города.

а) Необходимо найти вероятность события  $A_2$ .

События  $A_1$  и  $A_2$  являются противоположными, так как если ресторан не будет построен в северной части города, то он будет построен в южной части:

$$A_2 = \bar{A}_1.$$

Вероятность события  $A_1$  известна из условия задачи:  $P(A_1) = 0,4$ .

Вероятность события  $A_2$  можно определить как вероятность противоположного события:

$$P(A_2) = P(\bar{A}_1) = 1 - P(A_1) = 1 - 0,4 = 0,6.$$

*Ответ:* вероятность того, что ресторан будет построен в южной части города, составляет 0,6 или 60%.

б) Событие «ресторан будет построен в южной части города и его работа в первый год будет успешной» представляет собой одновременное выполнение событий  $A_2$  и  $B_1$ , т.е. произведение этих событий:  $A_2 * B_1$ .

События  $A_2$  и  $B_1$  являются зависимыми, так как по условию задачи вероятность успешной работы ресторана различна в зависимости от того, в какой части города он построен.

Поскольку эти события являются зависимыми, для расчета вероятности их произведения применим теорему о вероятности произведения событий:

$$P(A_2 * B_1) = P(A_2) * P(B_1|A_2).$$

Вероятность события  $A_2$  рассчитана в п. а), условная вероятность события  $B_1$  при условии, что наступило событие  $A_2$  ( $P(B_1|A_2)$ ) известна из условий задачи. Тогда:

$$P(A_2 * B_1) = 0,6 * 0,65 = 0,39.$$

*Ответ:* вероятность того, что ресторан будет построен в южной части города и его работа в первый год будет успешной, составляет 0,39 или 39%.

в) Событие «ресторан будет построен в северной части города, но его работа в первый год не будет успешной» представляет собой одновременное выполнение событий  $A_1$  и  $B_2$ , т.е. произведение этих событий:  $A_1 * B_2$ .

Событие  $A_1$  может происходить либо одновременно с событием  $B_1$ , либо одновременно с событием  $B_2$  (ресторан либо работает успешно, либо нет), поэтому событие  $A_1$  можно представить как сумму произведений соответствующих событий:

$$A_1 = A_1 * B_1 + A_1 * B_2.$$

События  $A_1*B_1$  и  $A_1*B_2$  являются несовместными, следовательно вероятность события  $A_1$  можно представить как сумму вероятностей этих событий (применить следствие из теоремы о вероятности суммы):

$$P(A_1) = P(A_1*B_1) + P(A_1*B_2).$$

Отсюда:

$$P(A_1*B_2) = P(A_1) - P(A_1*B_1).$$

Вероятность события  $A_1$  известна из условия задачи.

События  $A_1$  и  $B_1$  являются зависимыми (обоснование приведено в п. б).

Для расчета вероятности произведения зависимых событий применим теорему о вероятности произведения событий:

$$P(A_1*B_1) = P(A_1) * P(B_1|A_1).$$

Все вероятности известны из условия задачи:

$$P(A_1*B_1) = 0,4 * 0,9 = 0,36.$$

Следовательно,

$$P(A_1*B_2) = 0,4 - 0,36 = 0,04.$$

*Ответ:* вероятность того, что ресторан будет построен в северной части города, но его работа в первый год не будет успешной, составляет 0,04 или 4%.

г) Вероятность события «работа ресторана в первый год не будет успешной при условии, что он будет построен в северной части города» представляет собой условную вероятность  $P(B_2|A_1)$ . Эту вероятность можно определить из теоремы о вероятности произведения зависимых событий:

$$P(A_1*B_2) = P(A_1) * P(B_2|A_1).$$

Отсюда:

$$P(B_2|A_1) = P(A_1*B_2) / P(A_1).$$

Вероятность события  $A_1$  известна из условия задачи, вероятность события  $A_1*B_2$  найдена в п. в):

$$P(B_2|A_1) = 0,04 / 0,4 = 0,1.$$

*Ответ:* вероятность того, что работа ресторана в первый год не будет успешной при условии, что он будет построен в северной части города, составляет 0,1 или 10%.

д) Необходимо найти вероятность события  $B_1$  «ресторан будет работать успешно в течение первого года».

Событие  $B_1$  может происходить либо одновременно с событием  $A_1$ , либо одновременно с событием  $A_2$  (ресторан построен либо в северной, либо в южной части города), поэтому событие  $B_1$  можно представить как сумму произведений соответствующих событий:

$$B_1 = A_1 * B_1 + A_2 * B_1.$$

В этом случае вероятность события  $B_1$  можно рассчитать по формуле полной вероятности:

$$P(B_1) = P(A_1) * P(B_1|A_1) + P(A_2) * P(B_1|A_2).$$

Вероятность события  $A_2$  была найдена в п. а), все остальные составляющие вероятности известны из условия задачи:

$$P(B_1) = 0,4 * 0,9 + 0,6 * 0,65 = 0,75.$$

*Ответ:* вероятность того, что работа ресторана в течение первого года будет успешной, составляет 0,75 или 75%.

е) Вероятность события «ресторан построен в северной части города при условии, что его работа в первый год оказалась успешной» представляет собой условную вероятность  $P(A_1|B_1)$ , которую можно рассчитать по формуле Байеса:

$$P(A_1 | B_1) = \frac{P(A_1) \cdot P(B_1 | A_1)}{P(B_1)} = \frac{P(A_1 * B_1)}{P(B_1)}.$$

Вероятность события  $A_1 * B_1$  найдена в п. в), вероятность события  $B_1$  – в п. д):

$$P(A_1|B_1) = 0,36 / 0,75 = 0,48.$$

*Ответ:* вероятность того, что ресторан построен в северной части города при условии, что его работа в первый год оказалась успешной, составляет 0,48 или 48%.

## **2) Задачи по теме «Основные теоремы теории вероятностей»**

**1** На предприятии рассматривается вопрос о выпуске новой зубной пасты. При обсуждении стратегии сделаны следующие выводы: маркетинговое исследование будет удачным с вероятностью 0,65; при условии удачного маркетингового исследования вероятность успешного выпуска товара на рынок равна 0,55; полная вероятность успешного выпуска товара на рынок составляет 0,4.

а) Найдите вероятность того, что маркетинговое исследование окажется удачным и выпуск товара на рынок также окажется успешным.

б) Найдите условную вероятность того, что выпуск товара на рынок окажется успешным при условии отсутствия успеха в маркетинговом исследовании.

**2** Магазин изучает модель поведения своих покупателей. Вероятность того, что посещение магазина завершится покупкой, составляет 0,35. Вероятность того, что покупатель был в этом магазине в течение предыдущего месяца, равна 0,2. Из тех, кто ничего не купил, в последний месяц посещали магазин 12% (условная вероятность).

а) Какой процент покупателей часто посещают магазин, но редко делают покупки (эту категорию покупателей составляют те, кто не совершает покупку и был в магазине в течение прошлого месяца)?

б) Найдите условную вероятность того, что посетитель совершит покупку при условии, что он был в магазине в течение прошлого месяца.

**3** Для типичных посетителей кондитерского магазина вероятность покупки конфет составляет 0,23, вероятность покупки печенья равна 0,76, а условная вероятность покупки печенья при условии покупки конфет составляет 0,85.

а) Найдите вероятность покупки типичным посетителем и конфет, и печенья.

б) Найдите вероятность того, что типичный посетитель делает покупку (покупает либо конфеты, либо печенье).

**4** Организация часто принимает участие в конкурсах на выполнение различных научных проектов. Если при этом разрабатывается детальный финансовый план (30% всех проектов), то существует условная вероятность 80%, что удастся заключить контракт.

Если производятся только быстрые расчеты, то в этом случае условная вероятность заключения контракта составляет только 10%.

а) Найдите вероятность того, что удастся добиться заключения контракта.

б) Если заключить контракт не удалось, чему равна условная вероятность, что был разработан детальный финансовый план?

**5** Из всех телефонных звонков, на которые отвечает сотрудник отдела сбыта, в 75% случаев запрашивается информация, а в 15% случаев сразу делается заказ (без запроса информации). Кроме того, в 12% звонков после запроса информации также делается и заказ.

а) Найдите условную вероятность того, что телефонный звонок приводит к получению заказа, если в этом же звонке запрашивалась информация.

б) Найдите условную вероятность того, что покупатель выбрал товар заранее (звонок не связан с получением информации при условии, что сделан заказ).

**6** Организация подала заявку на участие в конкурсе на крупный государственный заказ. По оценкам экспертов существует вероятность в 35%, что предпочтение будет отдано заявкам конкурентов. Однако руководитель считает, что даже если это произойдет, то с вероятностью 10% он все равно сможет заключить контракт, убедив комиссию. С другой стороны, если предпочтение изначально будет отдано заявке данной организации, существует вероятность 5%, что контракт будет потерян в результате действий конкурентов.

а) Найдите вероятность того, что контракт удастся заключить.

б) Найдите условную вероятность того, что предпочтение отдано заявке данной организации при условии, что она заключила контракт.

**7** Вероятность успешной продажи нового товара в Москве равна 0,6. Вероятность его успешной продажи в Санкт-Петербурге составляет 0,7. А вероятность того, что товар будет успешно продаваться в обоих городах, равна 0,55.

а) Найдите вероятность успешной продажи товара хотя бы в одном из городов.

б) Найдите условную вероятность, что товар будет успешно продаваться в Санкт-Петербурге при условии, что он успешно продается в Москве.

**8** Предприятие начинает выпускать новые товары – детскую коляску и детское автомобильное кресло. Вероятность того, что эти товары будут иметь успех на рынке, составляет соответственно 0,8 и 0,75. Если коляски будут пользоваться успехом, то можно увеличить продажи кресел, предлагая их покупателям колясок: в этом случае продажи кресел будут успешными с условной вероятностью 0,85.

а) Найдите вероятность того, что успешными окажутся продажи хотя бы одного из товаров.

б) Найдите вероятность того, успех не будет достигнут ни для одного из товаров.

**9** Организация продает вязаные изделия по каталогам. Установлено, что 6% корреспондентов, получивших каталог, заказали комплект – шапочку и шарф, а 4%

корреспондентов заказали варежки. Причем из тех, кто заказал комплект, варежки заказали 55%.

- а) Какой процент корреспондентов заказали и комплект, и варежки?
- б) Какой процент корреспондентов не заказали вообще ничего?

### 2.3 Практическая работа №3 (2 часа).

**Тема:** «Принципы подхода в моделировании систем. Классификация видов моделирования систем»

**2.3.1 Цель работы:** Изучение основных идей моделирования систем, этапов и видов моделирования.

**2.3.2 Задачи работы:** Изучить основные понятия теории моделирования; основные типы моделей процессов и систем; основные требования, предъявляемые к разработке математических моделей

#### 2.3.3 Описание (ход) работы:

Стрелок производит четыре выстрела по мишени. Вероятность попадания в мишень при каждом выстреле равна 0,4. За каждое попадание стрелку насчитывается 5 очков.

- а) Построить ряд распределения числа полученных очков (случайная величина  $X$ ) по биномиальному закону. Проверить условие нормировки.
- б) Построить многоугольник распределения.
- в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.
- г) Записать функцию распределения случайной величины  $X$ , построить график функции.
- д) Найти вероятности того, что стрелок получит:
  - менее 10 очков;
  - от 5 до 15 очков;
  - более 10 очков.

*Решение.*

а) Дискретная случайная величина  $X$  – число полученных очков. Определим ее возможные значения:

- $x_1 = 0$ , если стрелок ни разу не попал в мишень;
- $x_2 = 5$ , если стрелок попал в мишень 1 раз;
- $x_3 = 10$ , если стрелок попал в мишень 2 раза;
- $x_4 = 15$ , если стрелок попал в мишень 3 раза;
- $x_5 = 20$ , если стрелок попал в мишень 4 раза.

Чтобы построить ряд распределения, нужно определить вероятности, соответствующие каждому возможному значению случайной величины:

$x_i$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$p_i$	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$



Для определения вероятностей используем формулу биномиального закона распределения (формулу Бернулли):

$$P_n(k) = C_n^k \cdot p^k \cdot (1-p)^{n-k},$$

где  $n$  – количество независимых испытаний (4 выстрела,  $n = 4$ );

$p$  – вероятность события (попадания) в каждом из испытаний ( $p = 0,4$ );

$k$  – количество произошедших событий (попаданий);

$C_n^k$  – число сочетаний из  $n$  элементов по  $k$  – неупорядоченные наборы по  $k$  элементов, взятых из  $n$ . Число сочетаний определяется по формуле:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Вероятность появления значения  $x_1$  соответствует вероятности того, что стрелок ни разу не попал в мишень (событие ни разу не произошло), вероятность появления значения  $x_2$  соответствует вероятности того, что стрелок попал в мишень 1 раз (событие произошло 1 раз) и т.д.:

$x_i$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$p_i$	$P_4(0)$	$P_4(1)$	$P_4(2)$	$P_4(3)$	$P_4(4)$

Рассчитаем вероятности:

$$k=0 \quad P_4(0) = C_4^0 \cdot p^0 \cdot (1-p)^{4-0}$$

$$C_4^0 = \frac{4!}{0!(4-0)!} = \frac{24}{1 \cdot 24} = 1$$

$$P_4(0) = 1 \cdot 0,4^0 \cdot 0,6^4 = 1 \cdot 0,1296 = 0,1296$$

$$k=1 \quad P_4(1) = C_4^1 \cdot p^1 \cdot (1-p)^{4-1}$$

$$C_4^1 = \frac{4!}{1!(4-1)!} = \frac{24}{1 \cdot 6} = 4$$

$$P_4(1) = 4 \cdot 0,4^1 \cdot 0,6^3 = 4 \cdot 0,4 \cdot 0,216 = 0,3456$$

$$k=2 \quad P_4(2) = C_4^2 \cdot p^2 \cdot (1-p)^{4-2}$$

$$C_4^2 = \frac{4!}{2!(4-2)!} = \frac{24}{2 \cdot 2} = 6$$

$$P_4(2) = 6 \cdot 0,4^2 \cdot 0,6^2 = 6 \cdot 0,16 \cdot 0,36 = 0,3456$$

$$k=3 \quad P_4(3) = C_4^3 \cdot p^3 \cdot (1-p)^{4-3}$$

$$C_4^3 = \frac{4!}{3!(4-3)!} = \frac{24}{6 \cdot 1} = 4$$

$$P_4(3) = 4 \cdot 0,4^3 \cdot 0,6^1 = 4 \cdot 0,064 \cdot 0,6 = 0,1536$$

$$k=4 \quad P_4(4) = C_4^4 \cdot p^4 \cdot (1-p)^{4-4}$$

$$C_4^4 = \frac{4!}{4!(4-4)!} = \frac{24}{24 \cdot 1} = 1$$

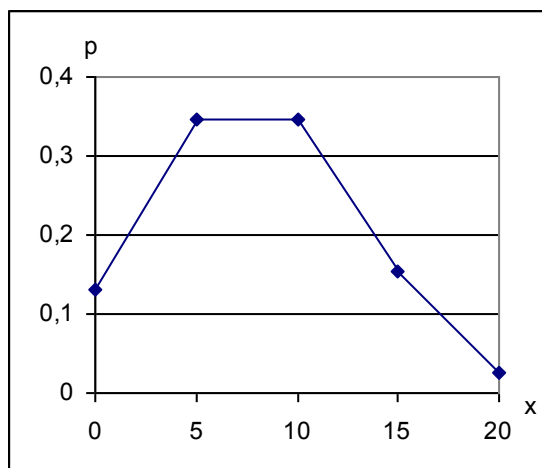
$$P_4(4) = 1 * 0,4^4 * 0,6^0 = 0,0256 * 1 = 0,0256$$

Построим ряд распределения:

$x_i$	0	5	10	15	20
$p_i$	0,1296	0,3456	0,3456	0,1536	0,0256

Проверим условие нормировки:  $0,1296 + 0,3456 + 0,3456 + 0,1536 + 0,0256 = 1$ .

б) Многоугольник распределения – диаграмма, которая позволяет наглядно представить закон распределения случайной величины X:



в) Математическое ожидание дискретной случайной величины определяется по формуле:

$$M(X) = \sum_{i=1}^n x_i p_i ,$$

где  $x_i$  – возможные значения случайной величины X;

$p_i$  – соответствующие этим значениям вероятности.

Рассчитаем математическое ожидание случайной величины X:

$$M(X) = 0 * 0,1296 + 5 * 0,3456 + 10 * 0,3456 + 15 * 0,1536 + 20 * 0,0256 = 8$$

*Вывод:* среднее число очков, которое может получить стрелок, равно 8.

Дисперсию дискретной случайной величины определим по формуле:

$$D(X) = \sum_{i=1}^n x_i^2 p_i - [M(X)]^2$$

$$D(X) = 0^2 * 0,1296 + 5^2 * 0,3456 + 10^2 * 0,3456 + 15^2 * 0,1536 + 20^2 * 0,0256 - 8^2 = 24$$

Среднее квадратичное отклонение представляет собой корень из дисперсии:

$$\sigma(X) = \sqrt{D(X)} = \sqrt{24} = 4,9$$

*Вывод:* число полученных очков может отличаться от математического ожидания (равного 8) в среднем на 4,9.

г) Определим функцию распределения случайной величины X.

Аналитическая запись функции распределения складывается из отдельных записей для каждого диапазона, на которые разбивается числовая ось возможными значениями случайной величины  $X$ . Число диапазонов равно  $n+1$ , где  $n$  – число возможных значений случайной величины.

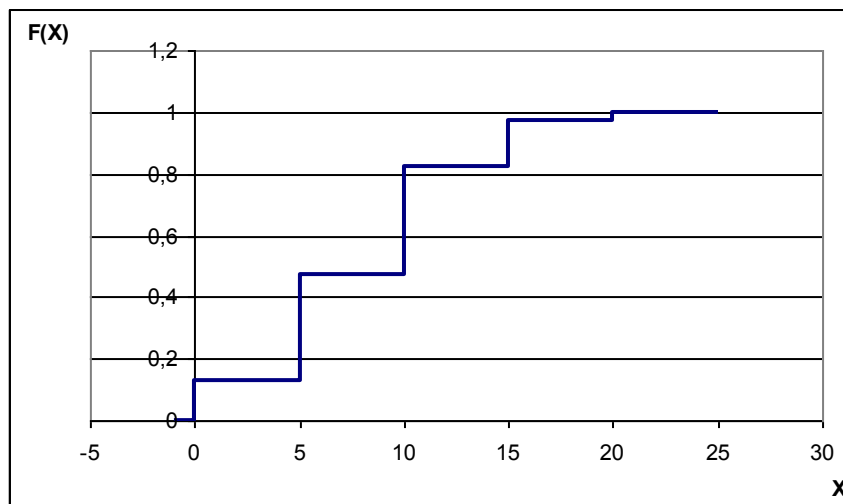
$n=5$ , значит, число диапазонов равно 6.

Для дискретной случайной величины  $F(x) = \sum_{x_i < x} P(X = x_i)$ ,

т.е. чтобы найти значение функции распределения для значения аргумента  $x$ , нужно сложить все вероятности, соответствующие значениям случайной величины, меньшим, чем  $x$ .

$$F(x) = \begin{cases} \text{при } x \leq 0 & F(x) = P(X < 0) = 0, \\ \text{при } 0 < x \leq 5 & F(x) = P(X = 0) = 0,1296, \\ \text{при } 5 < x \leq 10 & F(x) = P(X = 0) + P(X = 5) = 0,1296 + 0,3456 = 0,4752, \\ \text{при } 10 < x \leq 15 & F(x) = P(X = 0) + P(X = 5) + P(X = 10) = 0,1296 + 0,3456 + 0,3456 = 0,8208, \\ \text{при } 15 < x \leq 20 & F(x) = P(X = 0) + P(X = 5) + P(X = 10) + P(X = 15) = \\ & = 0,1296 + 0,3456 + 0,3456 + 0,1536 = 0,9744, \\ \text{при } x > 20 & F(x) = P(X = 0) + P(X = 5) + P(X = 10) + P(X = 15) + P(X = 20) = \\ & = 0,1296 + 0,3456 + 0,3456 + 0,1536 + 0,0256 = 1 \end{cases}$$

Построим график функции  $F(x)$ :



д) Чтобы определить вероятность попадания значений случайной величины в заданный интервал, следует использовать формулу:

$$P(a \leq X < b) = F(b) - F(a),$$

где  $a$  – нижняя граница заданного интервала;  
 $b$  – верхняя граница заданного интервала;  
 $F(b)$  – функция распределения случайной величины при  $X = b$ ;  
 $F(a)$  – функция распределения случайной величины при  $X = a$ .

Определим вероятность того, что стрелок получит менее 10 очков:

$a = 0, \quad b = 10$

$$P(0 \leq X < 10) = F(10) - F(0)$$

Значения функции распределения при  $X=10$  и  $X=0$  определим из аналитической записи функции  $F(x)$  в п. г):

значение  $X=0$  входит в первый интервал, значит,  $F(0) = 0$ ,

значение  $X=10$  входит в третий интервал, значит,  $F(10) = 0,4752$ .

$$P(0 \leq X < 10) = 0,4752 - 0 = 0,4752.$$

Таким образом, в 47,52% случаев стрелок получит менее 10 очков.

Определим вероятность того, что стрелок получит от 5 до 15 очков.

Преобразуем заданный интервал таким образом, чтобы он соответствовал формуле для определения вероятности:

$$P(5 \leq X \leq 15) = P(5 \leq X < 20)$$

$$a = 5, \quad b = 20$$

$$P(5 \leq X < 20) = F(20) - F(5) = 0,9744 - 0,1296 = 0,8448.$$

Таким образом, в 84,48% случаев стрелок получит от 5 до 15 очков.

Определим вероятность того, что стрелок получит более 10 очков.

Преобразуем заданный интервал таким образом, чтобы он соответствовал формуле для определения вероятности:

$$P(10 < X < \infty) = P(15 \leq X < \infty)$$

$$a = 15, \quad b = \infty$$

$$P(15 \leq X < \infty) = F(\infty) - F(15) = 1 - 0,8208 = 0,1792.$$

Таким образом, в 17,92% случаев стрелок получит более 10 очков.

### **3) Задачи по теме «Построение закона распределения дискретной случайной величины»**

**1** По дисциплине «Теория вероятностей и математическая статистика» предусмотрено 16 лекций. Вероятность посещения студентом любой лекции составляет 90%.

а) Построить ряд распределения числа посещенных лекций (случайная величина  $X$ ) по биномиальному закону. Проверить условие нормировки.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что студент посетит более половины лекций.

**2** Бетонные блоки поступают на строительную площадку с интенсивностью 2 блока/час.

а) Построить ряд распределения количества блоков, поступивших за 2 часа, (случайная величина  $X$ ) по закону Пуассона. Использовать 10 первых значений случайной величины.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что количество блоков, поступивших за 2 часа, составит от 3 до 5 шт.

**3** Учебник издан тиражом 100 000 экземпляров. Вероятность того, что учебник сброшюрован неправильно, равна 0,0001.

а) Построить ряд распределения количества бракованных книг (случайная величина  $X$ ) по закону Пуассона. Использовать 13 первых значений случайной величины.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что тираж содержит менее 8 бракованных книг.

**4** Фирма по доставке горячих обедов планирует позвонить в 8 крупных организаций с предложением своих услуг. Каждый звонок с вероятностью 20% приводит к заказу.

а) Построить ряд распределения числа заказанных обедов (случайная величина  $X$ ) по биномиальному закону. Проверить условие нормировки.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что более половины фирм закажут горячие обеды.

**5** В неудачный день на фондовой бирже 80% ценных бумаг падает в цене. Оценивается портфель, содержащий 12 ценных бумаг.

а) Построить ряд распределения количества ценных бумаг, упавших в цене, (случайная величина  $X$ ) по биномиальному закону. Проверить условие нормировки.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность падения в цене менее половины ценных бумаг.

**6** Завод отправил на базу 500 изделий. Вероятность повреждения изделия в пути равна 0,002.

а) Построить ряд распределения количества изделий, поврежденных в пути, (случайная величина  $X$ ) по закону Пуассона. Использовать 8 первых значений случайной величины.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что в пути будет повреждено хотя бы одно изделие.

**7** На заседании совета директоров должно состояться голосование по важному вопросу. Число директоров – 10 человек, а вероятность того, что каждый из них будет голосовать «за», составляет 0,53.

а) Построить ряд распределения числа положительных голосов (случайная величина  $X$ ) по биномиальному закону. Проверить условие нормировки.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что решение будет принято, т.е. что большинство директоров выскажутся за него.

**8** На автоматическую телефонную станцию (АТС) поступает в среднем 3 вызова в минуту.

а) Построить ряд распределения количества вызовов, поступивших за 2 минуты (случайная величина  $X$ ) по закону Пуассона. Использовать 10 первых значений случайной величины.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что за 2 минуты на станцию поступит от 3 до 5 вызовов.

**9** На ткацком станке нить обрывается в среднем 0,375 раза в течение часа работы станка.

а) Построить ряд распределения количества обрывов нити за 8 часов (случайная величина  $X$ ) по закону Пуассона. Использовать 10 первых значений случайной величины.

б) Построить многоугольник распределения.

в) Определить математическое ожидание, дисперсию и среднее квадратичное отклонение случайной величины  $X$ . Сделать выводы.

г) Записать функцию распределения случайной величины  $X$ , построить график функции.

д) Найти вероятность того, что за 8 часов число обрывов нити будет заключено в границах от 2 до 4.

## Приложение А

### Кодификатор заданий

		Первая буква фамилии								
		А Л Х	Б М Ц	В Н Ч	Г О Ш	Д П Щ	Е Р Э	Ж С Ю	З Т Я	И К У Ф
Последняя цифра шифра	0	5.1 6.12 7.14	5.2 6.13 7.15	5.3 6.14 7.16	5.4 6.15 7.17	5.5 6.16 7.1	5.6 6.17 7.2	5.7 6.18 7.3	5.8 6.19 7.4	5.9 6.20 7.5
	1	5.10 6.3 7.5	5.11 6.4 7.6	5.12 6.5 7.7	5.13 6.6 7.8	5.14 6.7 7.9	5.15 6.8 7.10	5.1 6.9 7.11	5.2 6.10 7.12	5.3 6.11 7.13
	2	5.4 6.14 7.13	5.5 6.15 7.14	5.6 6.16 7.15	5.7 6.17 7.16	5.8 6.18 7.17	5.9 6.19 7.1	5.10 6.20 7.2	5.11 6.1 7.3	5.12 6.2 7.4
	3	5.13 6.5 7.4	5.14 6.6 7.5	5.15 6.7 7.6	5.1 6.8 7.7	5.2 6.9 7.8	5.3 6.10 7.9	5.4 6.11 7.10	5.5 6.12 7.11	5.6 6.13 7.12
	4	5.7 6.16 7.12	5.8 6.17 7.13	5.9 6.18 7.14	5.10 6.19 7.15	5.11 6.20 7.16	5.12 6.1 7.17	5.13 6.2 7.1	5.14 6.3 7.2	5.15 6.4 7.3
	5	5.15 6.7 7.3	5.13 6.8 7.4	5.11 6.9 7.5	5.9 6.10 7.6	5.7 6.11 7.7	5.5 6.12 7.8	5.3 6.13 7.9	5.1 6.14 7.10	5.14 6.15 7.11
	6	5.12 6.18 7.11	5.10 6.19 7.12	5.8 6.20 7.13	5.6 6.1 7.14	5.4 6.2 7.15	5.2 6.3 7.16	5.15 6.4 7.17	5.13 6.5 7.1	5.11 6.6 7.2
	7	5.9 6.9 7.2	5.7 6.10 7.3	5.5 6.11 7.4	5.3 6.12 7.5	5.1 6.13 7.6	5.14 6.14 7.7	5.12 6.15 7.8	5.10 6.16 7.9	5.8 6.17 7.10
	8	5.6 6.20 7.10	5.4 6.1 7.11	5.2 6.2 7.12	5.15 6.3 7.13	5.13 6.4 7.14	5.11 6.5 7.15	5.9 6.6 7.16	5.7 6.7 7.17	5.5 6.8 7.1
	9	5.3 6.11 7.1	5.1 6.12 7.2	5.14 6.13 7.3	5.12 6.14 7.4	5.10 6.15 7.5	5.8 6.16 7.6	5.6 6.17 7.7	5.4 6.18 7.8	5.2 6.19 7.9

#### 2.4 Практическая работа №4 (2 часа).

**Тема:** «Понятие математической схемы. Математическая схема общего вида»

**2.4.1 Цель работы:** Целью работы является изучение матричных способов представления графов.

**2.4.2 Задачи работы:** Научиться изучать математические схемы с помощью матричных графов

**2.4.3 Описание (ход) работы:**

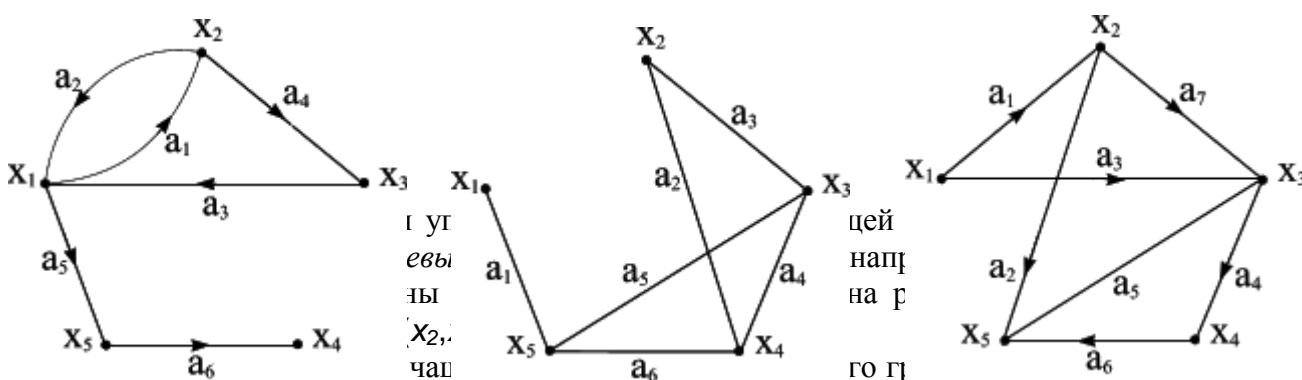
## ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

В последнее время теория графов стала простым, доступным и мощным средством решения вопросов, относящихся к широкому кругу проблем. Это проблемы проектирования интегральных схем и схем управления, исследования автоматов, логических цепей, блок-схем программ, экономики и статистики, химии и биологии, теории расписаний и дискретной оптимизации.

### ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

*Граф*  $G$  задается множеством точек или вершин  $x_1, x_2, \dots, x_n$  (которое обозначается через  $X$ ) и множеством линий или *ребер*  $a_1, a_2, \dots, a_n$  (которое обозначается символом  $A$ ), соединяющих между собой все или часть этих точек. Таким образом, граф  $G$  полностью задается (и обозначается) парой  $(X, A)$ .

Если ребра из множества  $A$  ориентированы, что обычно показывается стрелкой, то они называются *дугами*, и граф с такими ребрами называется *ориентированным* графом (рисунок 1(а)). Если ребра не имеют ориентации, то граф называется *неориентированным* (рисунок 1(б)). В случае когда  $G=(X, A)$  является ориентированным графом и мы хотим пренебречь направленностью дуг из множества  $A$ , то неориентированный граф, соответствующий  $G$ , будем обозначать как  $G=(X, A)$ .



множества вершин  $X$  и *соответствия*  $\Gamma$ , которое показывает, как между собой связаны вершины. Соответствие  $\Gamma$  называется *отображением* множества  $X$  в  $X$ , а граф в этом случае обозначается парой  $G=(X, \Gamma)$ .

Рисунок 1(а) – ориентированный граф; (б) – неориентированный граф; (в) – ориентированный граф. Для графа на рисунке 1(а) имеем  $\Gamma(x_1)=\{x_2, x_3\}$ , т. е. вершины  $x_2$  и  $x_3$  являются конечными вершинами дуг, исходящих из начальной вершины  $x_1$ .

$$\Gamma(x_2)=\{x_1, x_3\}, \quad \Gamma(x_3)=\{x_1\}, \quad \Gamma(x_4)=\emptyset - \text{пустое множество}, \quad \Gamma(x_5)=\{x_4\}.$$

В случае неориентированного графа или графа, содержащего и дуги, и неориентированные ребра (см., например, графы, изображенные на рисунках 1(б) и 1(в)), предполагается, что соответствие  $\Gamma$  задает такой эквивалентный ориентированный граф, который получается из исходного графа заменой каждого неориентированного ребра двумя противоположно направленными дугами, соединяющими те же самые вершины. Так, например, для графа, приведенного на рисунке 1(б), имеем  $\Gamma(x_5)=\{x_1, x_3, x_4\}$ ,  $\Gamma(x_1)=\{x_5\}$  и др.

Поскольку *прямое соответствие* или *образ* вершины  $\Gamma(x_i)$  представляет собой множество таких вершин  $x_j \in X$ , для которых в графе  $G$  существует дуга  $(x_i, x_j)$ , то через  $\Gamma^{-1}(x_j)$  естественно обозначить множество вершин  $x_k$ , для которых в  $G$  существует дуга



$(x_k, x_i)$ . Такое отношение принято называть *обратным соответствием* или *прообразом* вершины. Для графа, изображенного на рисунке 1(а), имеем

$$\Gamma^{-1}(x_1) = \{x_2, x_3\}, \quad \Gamma^{-1}(x_2) = \{x_1\} \text{ и т. д.}$$

Вполне очевидно, что для неориентированного графа  $\Gamma^{-1}(x_i) = \Gamma(x_i)$  для всех  $x_i \in X$ .

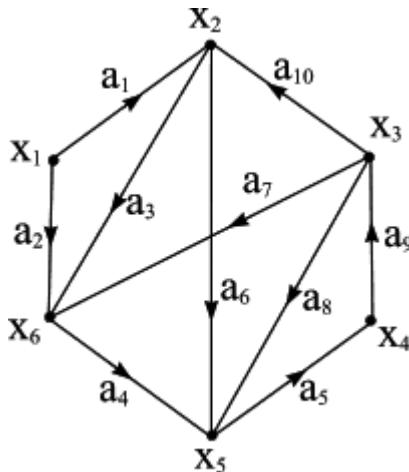
Когда отображение  $\Gamma$  действует не на одну вершину, а на множество вершин  $X_q = \{x_1, x_2, \dots, x_q\}$ , то под  $\Gamma(X_q)$  понимают объединение  $\Gamma(x_1) \cup \Gamma(x_2) \cup \dots \cup \Gamma(x_q)$ , т. е.  $\Gamma(X_q)$  является множеством таких вершин  $x_j \in X$ , что для каждой из них существует дуга  $(x_i, x_j)$  в  $G$ , где  $x_i \in X_q$ . Для графа, приведенного на рисунке 1(а),  $\Gamma(\{x_2, x_5\}) = \{x_1, x_3, x_4\}$  и  $\Gamma(\{x_1, x_3\}) = \{x_2, x_5, x_1\}$ .

Отображение  $\Gamma(\Gamma(x_i))$  записывается как  $\Gamma^2(x_i)$ . Аналогично "тройное" отображение  $\Gamma(\Gamma(\Gamma(x_i)))$  записывается как  $\Gamma^3(x_i)$  и т. д. Для графа, показанного на рисунке 1(а), имеем:

$$\Gamma^2(x_1) = \Gamma(\Gamma(x_1)) = \Gamma(\{x_2, x_5\}) = \{x_1, x_3, x_4\};$$

$$\Gamma^3(x_1) = \Gamma(\Gamma^2(x_1)) = \Gamma(\{x_1, x_3, x_4\}) = \{x_2, x_5, x_1\} \quad \text{и т. д.}$$

Аналогично понимаются обозначения  $\Gamma^{-2}(x_i)$ ,  $\Gamma^{-3}(x_i)$  и т. д.



Дуги  $a = (x_i, x_j)$ ,  $x_i \neq x_j$ , имеющие общие концевые вершины, называются *смежными*. Две вершины  $x_i$  и  $x_j$  называются смежными, если какая-нибудь из двух дуг  $(x_i, x_j)$  и  $(x_j, x_i)$  или обе одновременно присутствуют в графе. Так, например, на рисунке 2 дуги  $a_1$ ,  $a_{10}$ ,  $a_3$  и  $a_6$  как и вершины  $x_5$  и  $x_3$ , являются смежными, в то время как дуги  $a_1$  и  $a_5$  или вершины  $x_1$  и  $x_4$  не являются смежными.

Число дуг, которые имеют вершину  $x_i$  своей начальной вершиной, называется *полустепенью исхода* вершины  $x_i$ , и, аналогично, число дуг, которые имеют  $x_i$  своей конечной вершиной, называется *полустепенью захода* вершины  $x_i$ .

Рисунок 2.

захода вершины  $x_i$ .

Таким образом, на рисунке 2 полустепень исхода вершины  $x_3$ , обозначаемая через  $\deg^+(x_3)$ , равна  $|\Gamma(x_3)| = 3$ , и полустепень захода вершины  $x_3$ , обозначаемая через  $\deg^-(x_3)$ , равна  $|\Gamma^{-1}(x_3)| = 1$ .

Очевидно, что сумма полустепеней захода всех вершин графа, а также сумма полустепеней исхода всех вершин равны общему числу дуг графа  $G$ , т. е.

$$\sum_{i=1}^n \deg^+(x_i) = \sum_{i=1}^n \deg^-(x_i) = m, \quad (1)$$

где  $n$  - число вершин и  $m$  - число дуг графа  $G$ .

Для неориентированного графа  $G=(X, \Gamma)$  *степень* вершины  $x_i$  определяется аналогично - с помощью соотношения  $\deg(x_i) \equiv |\Gamma(x_i)| = |\Gamma^{-1}(x_i)|$ .

*Петлей* называется дуга, начальная и конечная вершины которой совпадают. На рисунке 3, например, дуги  $a_3$  и  $a_{10}$  являются петлями.

## МАТРИЧНЫЕ ПРЕДСТАВЛЕНИЯ

### МАТРИЦА СМЕЖНОСТИ

Пусть дан граф  $G$ , его матрица смежности обозначается через  $A=[a_{ij}]$  и определяется следующим образом:

$a_{ij}=1$ , если в  $G$  существует дуга  $(x_i, x_j)$ ,  
 $a_{ij}=0$ , если в  $G$  нет дуги  $(x_i, x_j)$ .

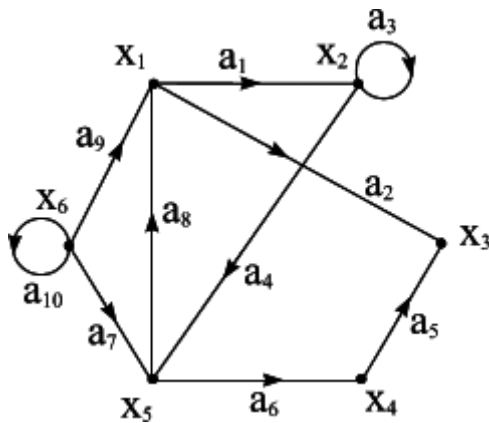


Рисунок 3.

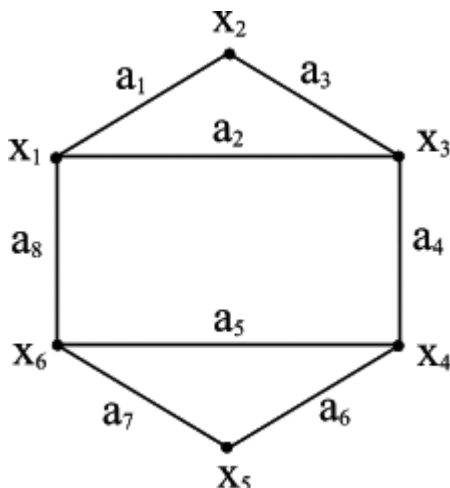
Таким образом, матрица смежности графа, изображенного на рисунке 3, имеет вид

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$	0	1	1	0	0	0
$x_2$	0	1	0	0	1	0
$x_3$	0	0	0	0	0	0
$x_4$	0	0	1	0	0	0
$x_5$	1	0	0	1	0	0
$x_6$	1	0	0	0	1	1

Матрица смежности полностью определяет структуру графа. Например, сумма всех элементов строки  $x_i$  матрицы дает полустепень исхода вершины  $x_i$ , а сумма элементов столбца  $x_i$  - полустепень захода вершины  $x_i$ . Множество столбцов, имеющих 1 в строке  $x_i$  есть множество  $\Gamma(x_i)$ , а множество строк, которые имеют 1 в столбце  $x_i$  совпадает с множеством  $\Gamma^{-1}(x_i)$ .

Петли на графе представляют собой элементы, имеющие 1 на главной диагонали матрицы, например  $a_{22}$ ,  $a_{66}$  для графа, изображенного на рисунке 3.

В случае неориентированного графа матрица смежности является симметричной относительно главной диагонали (рисунок 4).



	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$	0	1	1	0	0	1
$x_2$	1	0	1	0	0	0
$x_3$	1	1	0	1	0	0
$x_4$	0	0	1	0	1	1
$x_5$	0	0	0	1	0	1
$x_6$	1	0	0	1	1	0

Рисунок 4.

### МАТРИЦА ИНЦИДЕНТНОСТИ

Пусть дан граф  $G$  с  $n$  вершинами и  $m$  дугами. Матрица инцидентности графа  $G$  обозначается через  $B=[b_{ij}]$  и является матрицей размерности  $n \times m$ , определяемой следующим образом:

- $b_{ij}=1$ , если  $x_i$  является начальной вершиной дуги  $a_j$ ;
- $b_{ij}=-1$ , если  $x_i$  является конечной вершиной дуги  $a_j$ ;
- $b_{ij}=0$ , если  $x_i$  не является концевой вершиной дуги  $a_j$ .

Для графа, приведенного на рисунке 3, матрица инцидентности имеет вид:

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
$x_1$	1	1	0	0	0	0	0	-1	-1	0
$x_2$	-1	0	±1	1	0	0	0	0	0	0
$x_3$	0	-1	0	0	0	0	0	0	0	0
$x_4$	0	0	0	0	-1	-1	0	0	0	0
$x_5$	0	0	0	-1	1	1	-1	1	0	0
$x_6$	0	0	0	0	0	0	1	0	1	±1

Поскольку каждая дуга инцидентна двум различным вершинам (за исключением случая, когда дуга образует петлю), то каждый столбец содержит один элемент, равный 1, и один - равный -1. Петля в матрице инцидентности не имеет адекватного математического представления (в программной реализации допустимо задание одного элемента  $b_{ij}=1$ ).

Если  $G$  является неориентированным графом (рисунок 4), то его матрица инцидентности определяется следующим образом:

- $b_{ij}=1$ , если  $x_i$  является концевой вершиной дуги  $a_j$ ;
- $b_{ij}=0$ , если  $x_i$  не является концевой вершиной дуги  $a_j$ .

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
$x_1$	1	1	0	0	0	0	0	1
$x_2$	1	0	1	0	0	0	0	0
$x_3$	0	1	1	1	0	0	0	0
$x_4$	0	0	0	1	1	1	0	0
$x_5$	0	0	0	0	0	1	1	0
$x_6$	0	0	0	0	1	0	1	1

Ма  
трица  
инциден  
тности,  
как  
способ  
задания  
графов,

успешно применяется при описании мультиграфов (графов, в которых смежные вершины могут соединяться несколькими параллельными дугами).

### 3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

По заданной матрице смежности построить неориентированный граф, составить таблицу степеней вершин, матрицу инцидентности, таблицу расстояний и условных радиусов, найти радиус и центр графа.

1)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

2)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

4)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\left\| \begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{array} \right\|$$

5)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

6)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

7)

$$A(G) = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

8)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

9)

$$A(G) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

## 2.5 Практическая работа №5 (2 часа).

**Тема:** «Типовые математические схемы. Непрерывно-детерминированные модели ( $D$ -схемы)»

**2.5.1 Цель работы:** Приобретение практических навыков анализа предметной области, информационных задач и построения концептуальной модели базы данных.

**2.5.2 Задачи работы:** Изучение типовых математических схем на примере информационных задач.

### 2.5.3 Описание (ход) работы:

Вариант1.

Задача – организация учебного процесса в вузе:

\* Студенты: паспортные данные, адрес, дата зачисления, номер приказа, факультет, группа, является ли старостой, кафедра (специализация), изучаемые (изученные) предметы, оценки, задолженности, стипендия.

\* Учебные курсы: название, факультет(ы), групп(ы), кафедра, семестр(ы), форма отчётности, число часов.

\* Преподаватели: паспортные данные, адрес, телефон, фотография, кафедра, должность, учёная степень, начальник (зав. кафедрой), предмет(ы), число ставок, зарплата.

.

Вариант2.

Учет и выдача книг в библиотеке вуза:

\* Книги: авторы, название, раздел УДК, раздел (техническая, общественно-политическая и т.п.), место и год издания, издательство, количество страниц, иллюстрированность, цена, дата покупки, номер сопроводительного документа (чек, счёт/накладная), вид издания (книги, учебники, брошюры, периодические издания), инвентарный номер (есть только для книг и некоторых учебников), длительность использования читателями (год, две недели, день), электронная версия книги или ее реферата (отсканированный текст).

\* Читатели: номер читательского билета, ФИО, год рождения, адрес, дата записи, вид

(студент, аспирант, преподаватель, сотрудник), курс, номер группы, названия взятых книг и даты их выдачи.

#### Вариант3.

Отдел кадров некоторой компании.

\* Сотрудники: ФИО, паспортные данные, фотография, дом. и моб. телефоны, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, тип(ы) работы, задание(я), проект(ы), размер зарплаты, форма зарплаты (почасовая, фиксированная).

\* Отделы: название, комната, телефон(ы), начальник, размер финансирования, число сотрудников.

\* Проекты: название, дата начала, дата окончания, размер финансирования, тип финансирования (периодический, разовый), задачи и их исполнители, структура затрат и статьи расходов.

#### Вариант4.

Отдел поставок некоторого предприятия:

\* Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.

\* Сырьё: тип, марка, минимальный запас на складе, время задержки, цена, продукты, при производстве которых используется, потребляемые объёмы (необходимый, реальный, на единицу продукции).

#### Вариант5.

Пункт проката видеозаписей (внутренний учёт).

\* Видеокассеты: идентификационный номер видеокассеты, тип видеокассет, дата его создания, компания-поставщик, число штук данного типа (общее, в магазине, выдано в настоящее время, выдано всего, выдано в среднем за месяц), общая длительность записей; записи видеокассет: название, длительность, категория, год выпуска и производитель (оригинала).

\* Клиенты: ФИО, паспортные данные, адрес, телефон; заказы, т.е. взятые видеокассеты (сейчас и в прошлом): номер, дата выдачи, дата возвращения, общая стоимость заказа.

#### Вариант6.

Пункт проката видеозаписей (информация для клиентов).

\* Видеокассеты: краткое описание, внешний вид (этикетка), марка (пустой) видеокассеты, цена за единицу прокатного времени (например: 1 день, 3 дня, неделя), есть ли в наличии, общая длительность записей; записи на видеокассете: название, длительность, жанр (категория), тема, год и страна выпуска (оригинала), кинокомпания, описание, актеры, режиссер.

\* Заказы: идентификационные номера и названия выданных видеокассет, дата выдачи, дата возвращения (продления), общая стоимость заказа, возвращены ли кассеты заказа.

#### Вариант7.

Кинотеатры (информация для зрителей).

\* Фильмы: название, описание, жанр (категория), длительность, популярность (рейтинг, число проданных билетов в России и в мире), показывается ли сейчас (сегодня, на

текущей неделе), в каких кинотеатрах показывается, цены на билеты (в т.ч. средние).

\* Кинотеатры: название, адрес, схема проезда, описание, число мест (в разных залах, если их несколько), акустическая система, широкоэкранность, фильмы и цены на них: детские и взрослые билеты в зависимости от сеанса (дневной, вечерний и т.п.) и от категории мест (передние, задние и т.п.); сеансы показа фильмов (дата и время начала).

#### Вариант8.

Ресторан (информация для посетителей).

\* Меню: дневное или вечернее, список блюд по категориям.

\* Блюда: цена, название, вид кухни, категории (первое, второе и т.п.; мясное, рыбное, салат и т.п.), является ли вегетарианским, компоненты блюда, время приготовления, есть ли в наличии.

\* Компоненты блюд: тип (гарнир, соус, мясо и т.п.), калорийность, цена, рецепт, время приготовления, есть ли в наличии, ингредиенты (продукты) и их расходы на порцию.

#### Вариант9.

Задача- информационная поддержка деятельности склада.

База данных должна содержать информацию о наименовании товара, его поставщике, количестве, цене товара, конечном сроке реализации, сроке хранения на складе. Торговый склад производит уценку хранящейся продукции. Если продукция хранится на складе дольше 10 месяцев, то она уценивается в 2 раза, а если срок хранения превысил 6 месяцев, но не достиг 10, то в 1,5 раза. Ведомость уценки товаров должна содержать информацию: наименование товара, количество товара(шт.), цена товара до уценки, срок хранения товара, цена товара после уценки, общая стоимость товаров после уценки.

#### Вариант10.

Задача – информационная поддержка деятельности адвокатской конторы. БД должна осуществлять:

ведение списка адвокатов;

ведение списка клиентов;

ведение архива законченных дел.

Необходимо предусмотреть:

получение списка текущих клиентов для конкретного адвоката;

определение эффективности защиты (максимальный срок минус полученный срок) с учётом оправданий, условных сроков и штрафов;

определение неэффективности защиты (полученный срок минус минимальный срок);

подсчёт суммы гонораров (по отдельным делам) в текущем году;

получение для конкретного адвоката списка текущих клиентов, которых он защищал ранее (из архива, с указанием полученных сроков и статей).

#### Вариант11.

Задача – информационная поддержка деятельности гостиницы.

БД должна осуществлять:

ведение списка постояльцев;

учёт забронированных мест;

ведение архива выбывших постояльцев за последний год.

Необходимо предусмотреть:

получение списка свободных номеров (по количеству мест и классу);

получение списка номеров (мест), освобождающихся сегодня и завтра;

выдачу информации по конкретному номеру;

автоматизацию выдачи счетов на оплату номера и услуг;



получение списка забронированных номеров;  
проверку наличия брони по имени клиента и/или названию организации

#### Вариант12.

Описание предметной области:

- В компании несколько отделов.
- В каждом отделе есть некоторое количество сотрудников, занятых в нескольких проектах и размещающихся в нескольких офисах.
- Каждый сотрудник имеет план работы, т.е. несколько заданий, которые он должен выполнить. Для каждого такого задания существует ведомость, содержащая перечень денежных сумм, полученных сотрудником за выполнение этого задания.
- В каждом офисе установлено несколько телефонов.

В базе данных должна храниться следующая информация.

- Для каждого отдела: номер отдела ( уникальный), его бюджет и личный номер сотрудника, возглавляющего отдел ( уникальный).
- Для каждого сотрудника: личный номер сотрудника ( уникальный), номер текущего проекта, номер офиса, номер телефона, название выполняемого задания вместе с датой и размером выплат, проведенных в качестве оплаты за выполнение данного задания.
- Для каждого проекта : номер проекта ( уникальный) и его бюджет.
- Для каждого офиса : номер офиса ( уникальный), площадь в квадратных футах, номера всех установленных в нем телефонов.

#### Вариант13.

Задача – информационная поддержка деятельности спортивного клуба. БД должна осуществлять:

ведение списков спортсменов и тренеров;

учёт проводимых соревнований (с ведением их архива);

учёт травм, полученных спортсменами.

Необходимо предусмотреть:

возможность перехода спортсмена от одного тренера к другому;

составление рейтингов спортсменов;

составление рейтингов тренеров;

выдачу информации по соревнованиям;

выдачу информации по конкретному спортсмену;

подбор возможных кандидатур на участие в соревнованиях (соответствующего уровня мастерства, возраста и без травм).

#### Вариант14.

Задача – информационная поддержка деятельности аптечного склада.

В аптечном складе хранятся лекарства. Сведения о лекарствах содержатся в специальной ведомости: наименование лекарственного препарата; количество (в шт.); цена; срок хранения на складе (в месяцах). Лекарства поступают на склад ежедневно от разных поставщиков, отпускаются два раза в неделю по предварительным заказам аптек.

Выяснить, сколько стоит самый дорогой и самый дешевый препарат; сколько препаратов хранится на складе более 3 месяцев; сколько стоят все препараты, хранящиеся на складе, отыскать препараты, остаток которых равен нулю , ниже требуемого по заказам.

#### Вариант15.

“Электронный журнал посещаемости”

Предметная область представлена следующими документами:

Список студентов

Журнал посещаемости

Расписание занятий

Предусмотреть учет пропусков по уважительным, неуважительным причинам. Подсчет пропусков по каждому студенту, за неделю, месяц, заданный период, по конкретному предмету.

Вариант16.

«Итоги сессии»

База данных должна содержать информацию о двух последних сессиях студентов.

Источником информации являются экзаменационные ведомости. Необходимо проводить анализ успеваемости по специальностям, формам обучения, курсам, группам, предметам, вычислять средний балл по указанным критериям, а также число каждой оценки .

## **2.6 Практическая работа №6 (2 часа).**

**Тема:** «Дискретно-детерминированные модели (F–схемы). Конечные автоматы  
Классификация конечных автоматов»

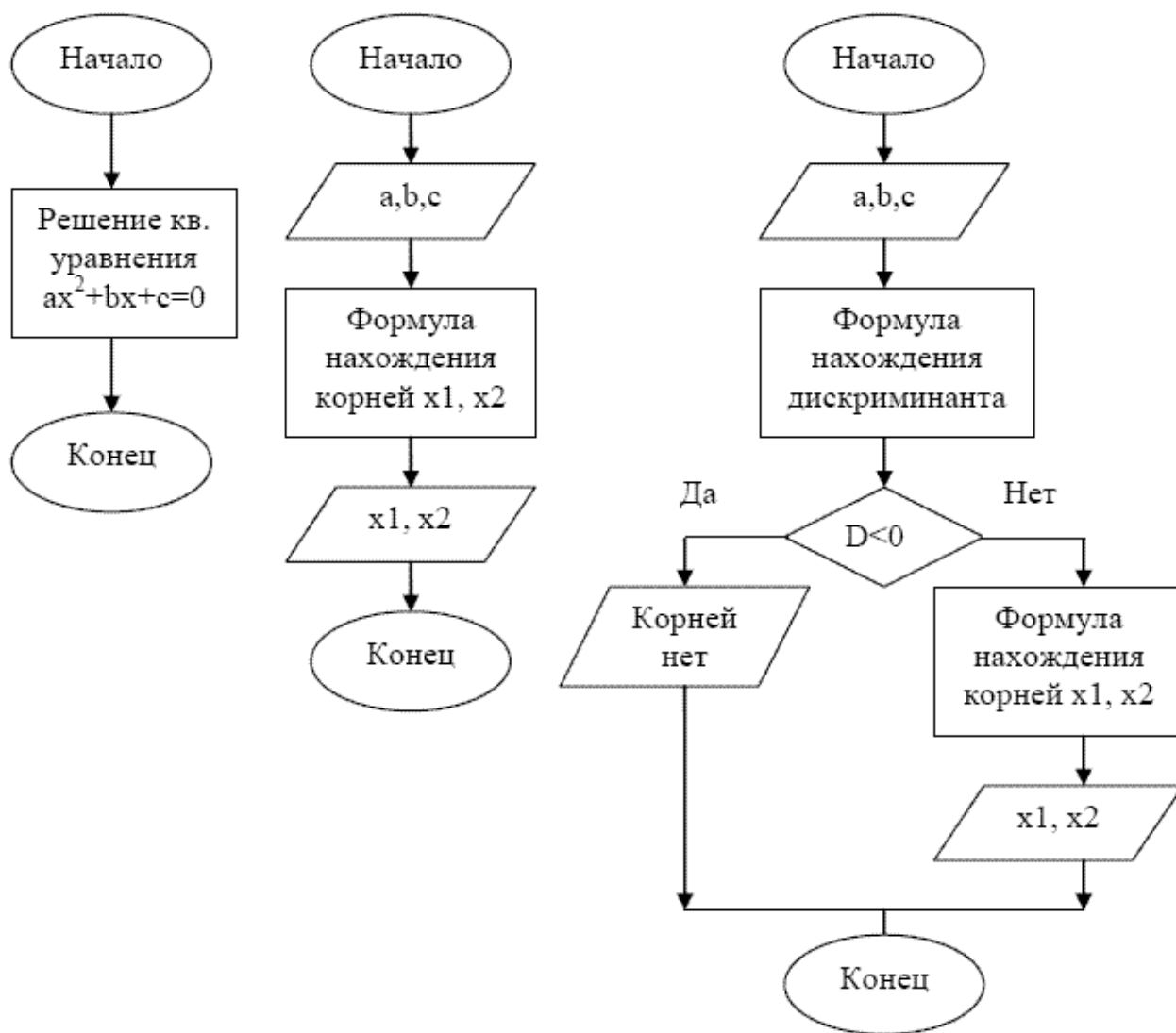
**2.6.1 Цель работы:** Изучить схему дискретно-детерминированной модели.

**2.6.2 Задачи работы:** Научиться разрабатывать программы для решения квадратных уравнений.

**2.6.3 Описание (ход) работы:**

*Задание 1.* Технология разработки программы решения квадратного уравнения. Заметим, что для начального шага разработки программы имеем в качестве входных данных коэффициенты  $a, b, c$  квадратного уравнения  $ax^2 + bx + c = 0$ , а на выходе – значения двух корней  $x_1, x_2$ .

Ответ:



Пошаговая детализация построения алгоритма

На рисунке проиллюстрирована пошаговая детализация процесса построения алгоритма.

Задания:

1. Дана функция  $y = y - ax^2 + |bx|$ ; Вычислить значение  $y$  при заданном  $a$ ,  $b$  и  $x$ .

2. Построить блок-схему алгоритма для вычисления  $u = \frac{|a|x + \sin x}{b}$ , при условии, что  $x \neq 0$ . Результат вывести на экран.

3. Построить блок-схему алгоритма для вывода длины двух катетов прямоугольного треугольника и нахождения длины гипотенузы и площади треугольника (использовать теорему Пифагора).

4. Построить блок-схему алгоритма для вычисления  $R = \sin[x] + \cos x^2$ . Результат вывести на экран.

$$Y = \frac{5x - \frac{x^2}{2}}{x - \frac{x^2}{3}}$$

5. Построить блок-схему алгоритма для вычисления  $x \neq 0$ . Результат вывести на экран.
6. Построить блок-схему алгоритма для вычисления длины окружностей с радиусами  $R1$  и  $R2$  и одним центром, а также площади кольца, образованного этими окружностями.
7. Построить блок-схему алгоритма для вычисления:  $Y = |x| - \sin(x) - a$ . Результат вывести на экран.
8. Построить блок-схему алгоритма для вычисления:  $U = \frac{|x| - a \sin x}{a - x}$  при условии, что  $a \neq x$ . Результат вывести на экран.
9. Построить блок-схему алгоритма для нахождения объема куба и площади его сторон по произвольному ребру куба  $a$ .

## 2.7 Практическая работа №7 (2 часа).

**Тема:** «Способы задания работы автоматов»

**2.7.1 Цель работы:** Проектирование цифровых устройств на основе программируемых логических интегральных схем.

**2.7.2 Задачи работы:** Рассмотреть возможности реакции автомата в течение текущего такта.

### 2.7.3 Описание (ход) работы:

Построение автомата Мили

Разметка ГСА для определения состояний УА, функционирующего в соответствии с моделью автомата Мили.

Набор состояний определяется путем отметки МП:  
для автомата Мили:

- символом  $Q_0$  отмечается вход первой вершины, следующей за начальной, а также вход конечной вершины;
- входы вершин, следующих за операторными, отмечаются символами  $Q_i$ ;
- входы двух различных вершин, за исключением конечной, не могут быть отмечены одинаковыми символами;
- вход вершины может отмечаться только одним символом.

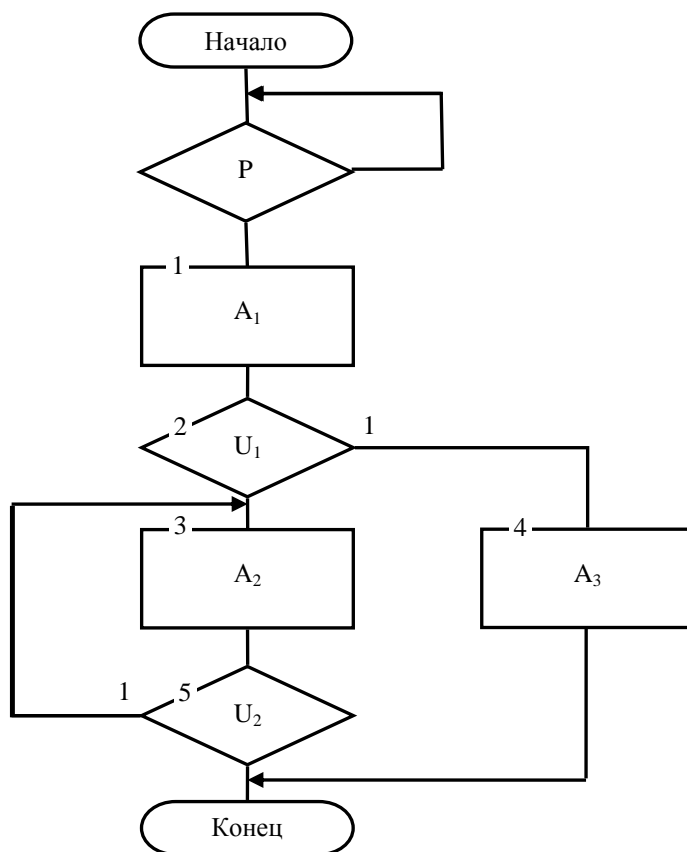


Рис. 2.1 Микропрограмма автомата

### Построение графа автомата Мили

Если состояниям  $Q_0, Q_1, Q_2$  поставить в соответствие вершины графа, а путям – дуги, направленные из вершины  $Q_i$  в  $Q_j$  и помеченные наборами значений «входные переменные» / «выходные переменные», то полученный граф будет определять закон функционирования автомата Мили:

$$Q(t+1) = \delta [Q(t), U(t)]$$

$$A(t) = \lambda [Q(t), U(t)],$$

где  $Q = \{Q_0, Q_1, Q_2\}$ ,

$$A = \{A_1, A_2, A_3\},$$

$$U = \{U_1, U_2\},$$

$$t = 0, 1, \dots$$

Дуги на графе автомата отмечаются только теми входными сигналами, которые определяют возможность перехода из состояния  $Q_i$  в  $Q_j$ , и теми выходными сигналами, которые в данной ситуации принимают значение 1. При этом предполагается, что все остальные выходные сигналы имеют нулевое значение.

Когда автомат не работает (микропрограмма не выполняется), он находится в состоянии  $Q_0$ . При запуске (инициировании микропрограммы) автомат сохраняет состояние  $Q_0$  в течение одного такта, за время которого выполняются микрооперации, соответствующие текущим значениям входных сигналов. По окончании первого такта автомат переключается в очередное состояние  $Q_j$ , предписанное законом функционирования.

Для запуска автомата используется специальный сигнал  $P$ , который относится к группе входных сигналов и имеет длительность, равную такту. Чтобы исключить возможность появления выходных сигналов в моменты, когда автомат находится в состоянии  $Q_0$  и не работает, дугам, исходящим из вершины  $Q_0$  дополнительно приписывается запускающий сигнал  $P$ , только при единичном значении которого

выходным сигналам присваивается 1 и становится возможным переход автомата в следующее состояние. При  $P=0$  автомат находится в  $Q_0$  и все выходные сигналы равны 0.

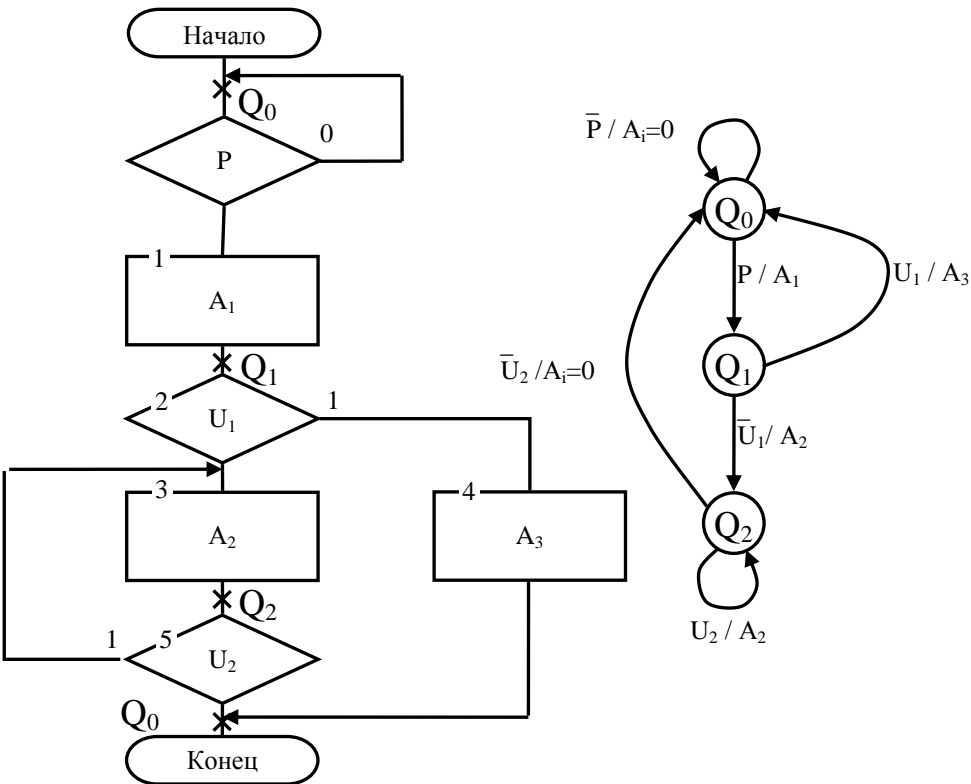


Рис. 2.2 Микропрограмма и граф интерпретирующего её автомата Мили

Переход от микропрограммы к автомату Мили иллюстрируется рис. 2.2, на котором показаны микропрограмма с метками (×) и граф интерпретирующего её автомата Мили.

Условия перехода по микропрограмме от одной метки состояния к другой задают функции переходов. Эти условия записываются в виде конъюнкции входных сигналов ( $P, A_1, A_2, A_3$ ). Для каждого перехода фиксируется также набор выходных переменных ( $U_1, U_2$ ), принимающих при переходе единичное значение (задание функции выходов).

От графа, интерпретирующего микропрограмму, можно перейти к её технической реализации.

Прежде всего, запишем закон функционирования автомата в виде таблицы переходов.

Составление структурной таблицы автомата и кодирование его состояний  
Таблица 2.1. Переходы автомата Мили

Исходное состояние	Входной набор	Выходной набор	Следующее состояние
$Q_0$	$\bar{P}$	-	$Q_0$
$Q_0$	$P$	$A_1$	$Q_1$
$Q_1$	$U_1$	$A_3$	$Q_0$
$Q_1$	$\bar{U}_1$	$A_2$	$Q_2$

$Q_2$	$U_2$	$A_2$	$Q_2$
$Q_2$	$\bar{U}_2$	-	$Q_0$

Зная число состояний автомата  $m$ , определим число триггеров, необходимых для реализации его памяти,  $K = \log_2 m$ .

Для приведённого примера  $m=3 \Rightarrow K = \log_2 3 \approx 2$  требуется два триггера.

Закодируем состояния автомата следующим образом:

$Q_0 = 00$ ;  $Q_1 = 01$ ;  $Q_2 = 10$ .

Для построения функций возбуждения и выходов используется структурная таблица, в которую добавлены два столбца кодов и столбец с перечнем сигналов возбуждения триггеров, формируемых на переходе.

Таблица 2.2. Переходы автомата Мили с кодами состояний

Исх. сост.	Код исх. сост.	Вх. набор	Вых. набор	След. сост.	Код след. сост.	Сигналы возбуждения	
						RS	
$Q_0$	00	$\bar{P}$	-	$Q_1$	00	-	
$Q_0$	00	$P$	$A_1$	$Q_1$	01	$S_2$	
$Q_1$	01	$\bar{U}_1$	$A_3$	$Q_0$	00	$R_2$	
$Q_1$	01	$\bar{U}_1$	$A_2$	$Q_2$	10	$S_1 R_2$	
$Q_2$	10	$\bar{U}_2$	$A_2$	$Q_2$	10	-	
$Q_2$	10	$\bar{U}_2$	-	$Q_0$	00	$R_1$	

Функции возбуждения выходов:

$$A_1 = Q_0 P$$

$$A_2 = Q_1 \bar{U}_1 \vee Q_2 U_2 \quad (1)$$

$$A_3 = Q_1 U_1$$

Подставляя в (1) выражения состояний через состояния  $p_1$  и  $p_2$  триггеров памяти

$$Q_0 = \bar{p}_1 \bar{p}_2$$

$$Q_1 = \bar{p}_1 p_2$$

$$Q_2 = p_1 \bar{p}_2$$

Имеем

$$A_1 = \bar{p}_1 \bar{p}_2 P$$

$$A_2 = \bar{p}_1 p_2 \bar{U}_1 \vee p_1 \bar{p}_2 U_2 \quad (2)$$

$$A_3 = \bar{p}_1 p_2 U_1$$

Функции возбуждения памяти зависят от типа триггера (RS, T, JK, D и др.)

Логика работы определяется количеством входов и особенностями схемы. В зависимости от влияния, оказываемого на состояние триггера, его входы имеют следующие обозначения:

- R (Reset — сброс) — вход (раздельной) установки триггера в нулевое состояние,  $Q = 0$
- S (Set — установка) — вход (раздельной) установки триггера в единичное состояние,  $Q = 1$
- T (Toggle — релаксатор) — счетный вход триггера

- J (Jerk — внезапное включение) — вход установки JK-триггера в единичное состояние,  $Q = 1$  (- вход для синхронизируемой установки в 1)
- K (Kill — внезапное выключение) —  $Q = 0$  (вход для синхронизируемой установки в 0);
- D (Delay — задержка) — вход установки триггера в единичное или нулевое состояние на время, равное одному такту
- C (Clock — часы) — вход синхронизирующих тактовых импульсов
- Y - вход разрешения.

Входы и выходы триггеров так же, как и других логических элементов, могут быть прямыми и инверсными, т.е. наличие сигнала определяется высоким или низким уровнем напряжений соответственно.

### RS-триггеры

Выберем для определённости RS-триггеры с отдельными входами.

Таблица 2.3. Состояния RS триггера

$R^t$	$S^t$	$Q^t$	$Q^{t+1}$	Режим
0	0	0/1	0/1	Хранение
0	1	0/1	1	Уст. 1
1	0	0/1	0	Уст. 0
1	1	0/1	0*	Запрещ. режим

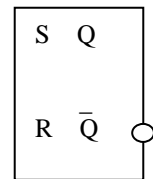


Рис. 2.4. УГО RS триггера

В столбце сигналов возбуждения указывается сигнал  $S_k$ , если триггер  $T_k$  на переходе  $(a_i, a_j)$  переключается из состояния 0 в состояние 1, и сигнал  $R_k$ , если триггер переключается из состояния 1 в состояние 0.

$$R_1 = Q_2 \quad \bar{U}_2 = p_1 \quad \bar{p}_2 \quad \bar{U}_2$$

$$S_1 = Q_1 \quad \bar{U}_1 = \bar{p}_1 p_2 \quad \bar{U}_1$$

$$R_2 = Q_1 \quad \bar{U}_1 \vee Q_1 U_1 = Q_1 (\bar{U}_1 \vee U_1) = Q_1 = \bar{p}_1 p_2$$

$$S_2 = Q_0 P = \bar{p}_1 \quad \bar{p}_2 P$$



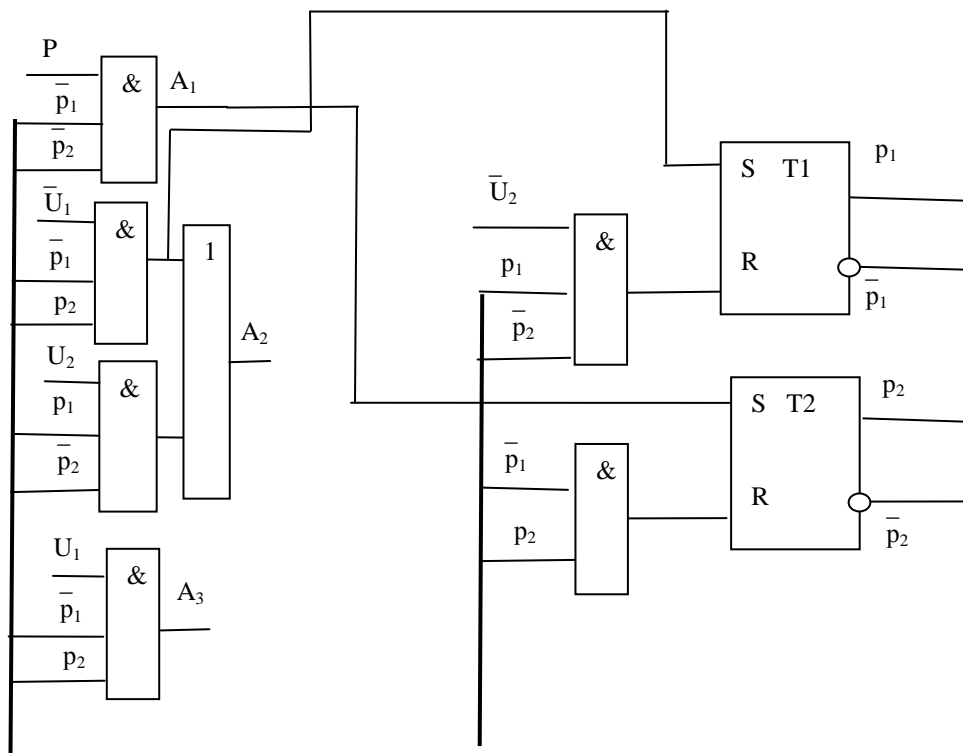


Рис. 2.5. Схема 1 на RS триггерах

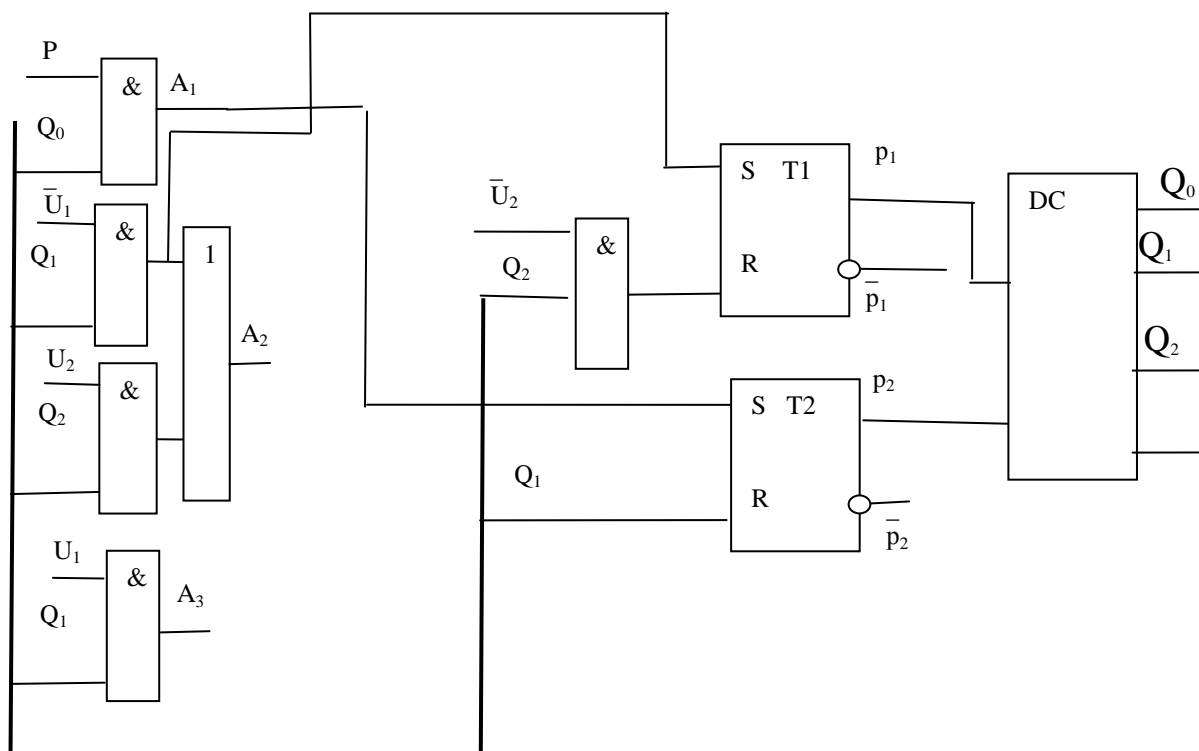


Рис. 2.6. Схема 2 на RS триггерах

### D-триггеры

D-триггер предназначен для хранения состояния (1 или 0) на один период тактовых импульсов (с задержкой на 1 такт). Его переходы отражены в табл.

Таблица 2.4. Переходы D-триггера

Входные сигналы	Состояние $q_t$		Режим
D	0	1	
0	0	0	Установка 0
1	1	1	Установка 1

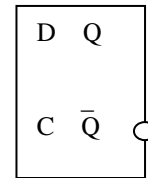


Рис. 2.7. УГО D-триггера

Сигнал  $D_i = 1$  возбуждения триггера в следующее состояние должен формироваться всякий раз, когда при переходе следующее состояние равно 1

Таблица 2.5. Переходы автомата Мили с кодами состояний

Исх. сост.	Код исх. сост.	Вх. набор	Вых. набор	След. сост.	Код след. сост.	Сигналы возбуждения	
						RS	D
$Q_0$	00	$\bar{P}$	-	$Q_1$	00	-	-
$Q_0$	00	$P$	$A_1$	$Q_1$	01	$S_2$	$D_2$
$Q_1$	01	$U_1$	$A_3$	$Q_0$	00	$R_2$	-
$Q_1$	01	$\bar{U}_1$	$A_2$	$Q_2$	10	$S_1 R_2$	$D_1$
$Q_2$	10	$U_2$	$A_2$	$Q_2$	10	-	$D_1$
$Q_2$	10	$\bar{U}_2$	-	$Q_0$	00	$R_1$	-

$$D_1 = Q_1 \bar{U}_1 \vee Q_2 U_2 = \bar{p}_1 p_2 \bar{U}_1 \vee p_1 \bar{p}_2 U_2$$

$$D_2 = Q_0 P = \bar{p}_1 \bar{p}_2 P$$

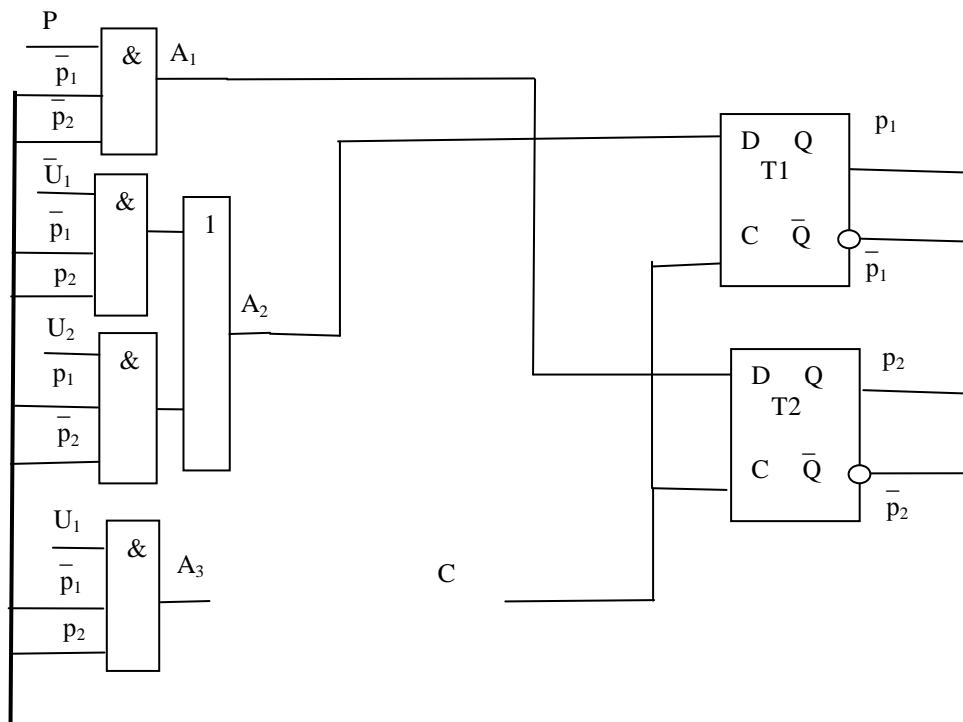


Рис. 2.6. Схема 1 на D триггерах

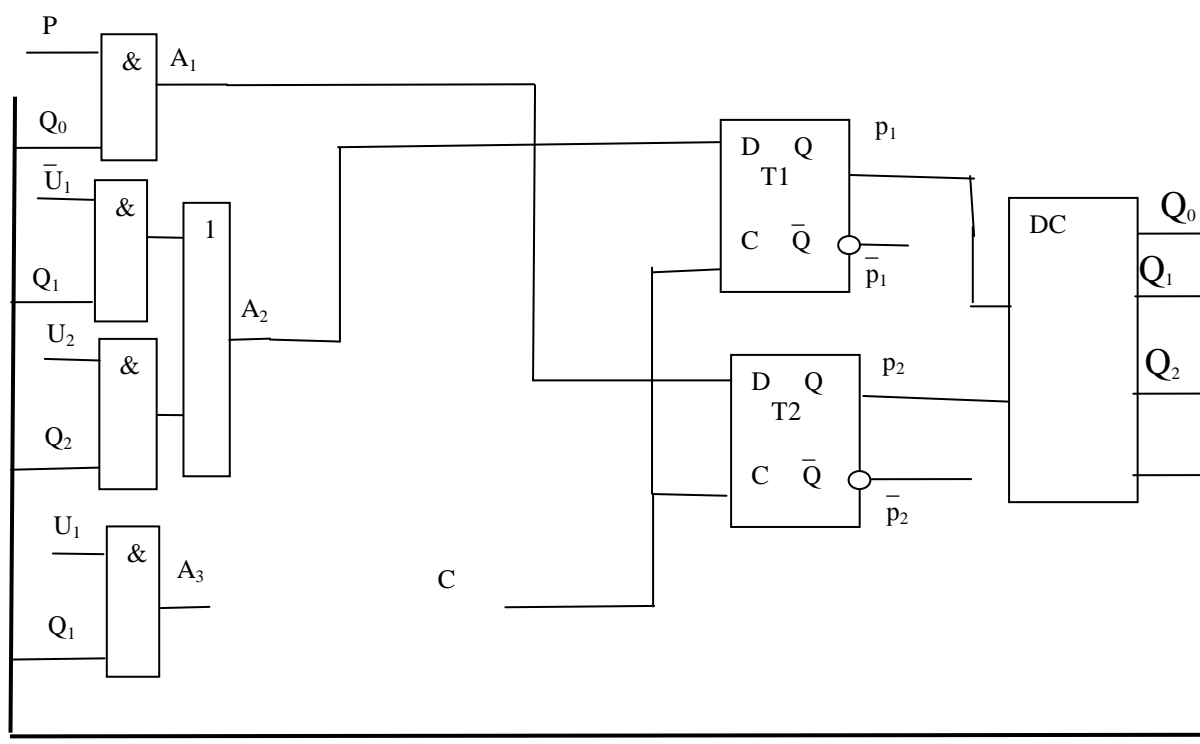


Рис. 2.7. Схема 2 на D триггерах

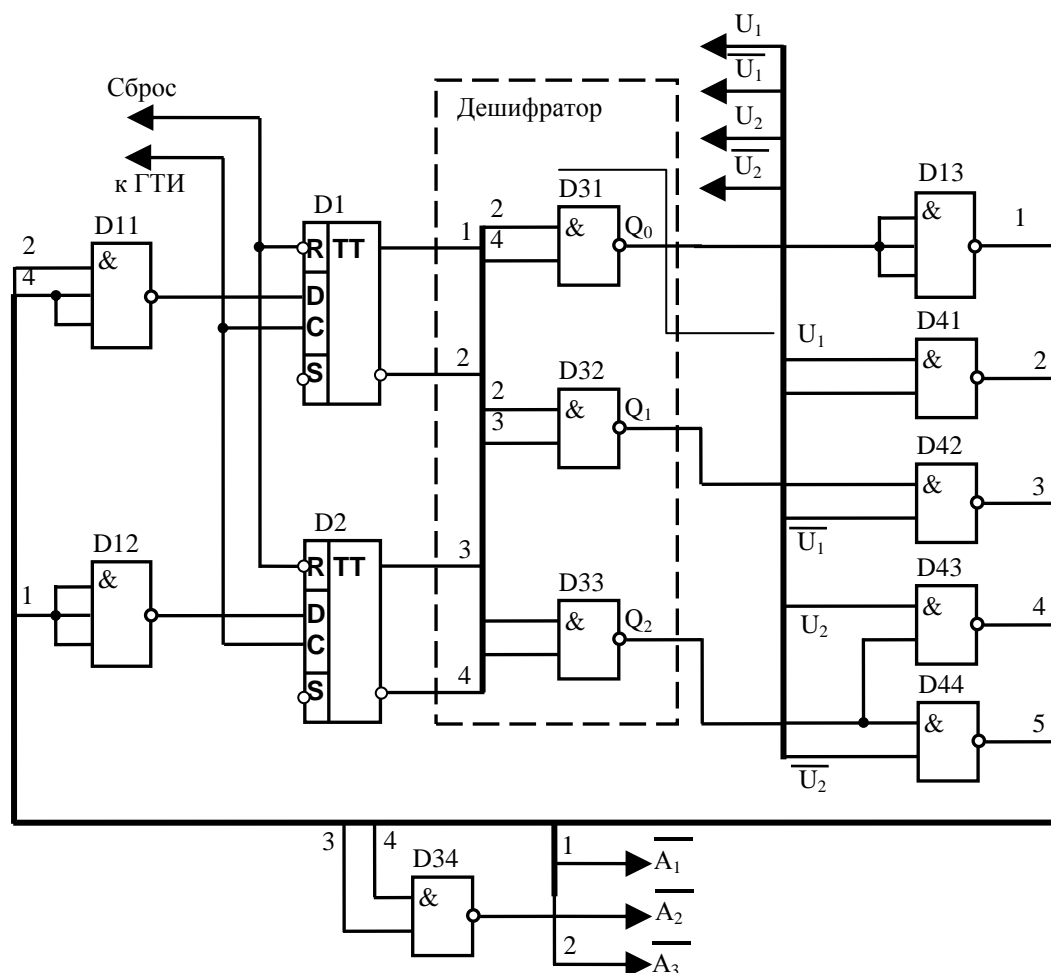


Рис. 2.8. Схема 3 на D триггерах

Схема дешифрации состояний состоит из элементов D3.1, D3.2, D3.3.

На элементах D1.3, D41- D44 собрана схема, вырабатывающая управляющие сигналы состояний  $Q_0$ ,  $Q_1$ ,  $Q_2$  и множество выходных сигналов  $\{A\}$ .

Схема функционирует так, что на её выходах только одно логическое значение "0", все остальные - "1". Зная этот выход (в разные моменты времени он разный) и граф микропрограммы, можно точно определить текущие и следующие состояния автомата, выходные управляющие сигналы. Поэтому тем, на какие элементы D1 или D2 заведён этот вывод (провод), и определяются следующие состояния автомата.

Объединение по схеме "монтажная логика" некоторых выводов этих элементов (D13, D4) согласно графу функционирования автомата даёт множество управляющих выходных сигналов.

Элементы D1.1, D1.2 служат для формирования функций возбуждения триггеров.

### Построение автомата Мура

**Разметка ГСА для определения состояний УА, функционирующего в соответствии с моделью автомата Мили**

Набор состояний определяется путем отметки МП:  
для автомата Мура:

- символом  $Q_0$  отмечается начальная и конечная вершины;
- символами  $Q_i$  отмечаются каждая операторная вершина;
- входы двух различных вершин, за исключением конечной, не могут быть отмечены одинаковыми символами;

- вход вершины может отмечаться только одним символом.

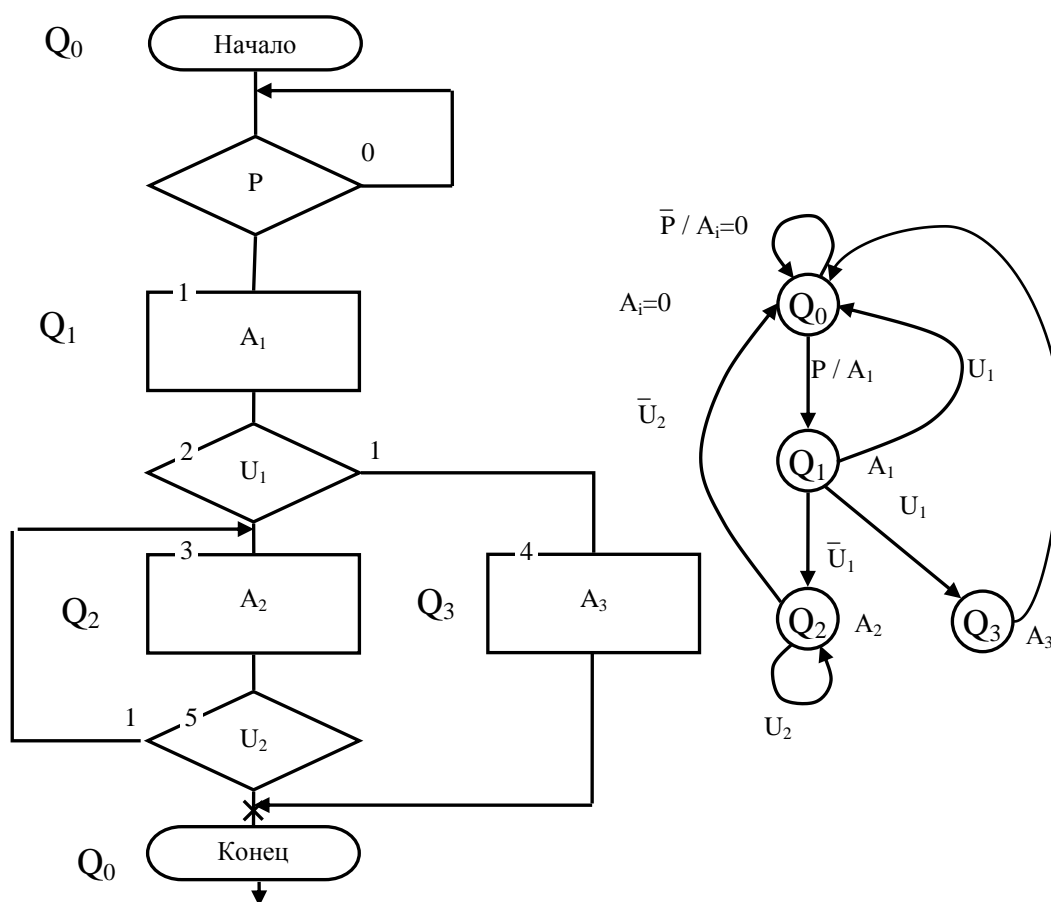


Рис. 2.9 Микропрограмма и граф интерпретирующего её автомата Мехили

Переход от микропрограммы к автомату Мили иллюстрируется рис. 2.2, на котором показаны микропрограмма с метками (×) и граф интерпретирующего её автомата Мили.

## 2.8 Практическая работа №8 (2 часа).

**Тема:** «Дискретно-стохастические модели (Р-схемы). Вероятностные автоматы»

**2.8.1 Цель работы:** Проследить разновидности вероятностных автоматов.

**2.8.2 Задачи работы:** Изучить применение схем вероятностных автоматов.

### 2.8.3 Описание (ход) работы:

При дискретно-стохастическом подходе к формализации процесса функционирования системы  $S$  подход остается аналогичный рассмотренному конечному автомату, то влияние фактора стохастичности можно проследить разновидности таких автоматов, а именно на вероятностных (стохастических) автоматах.

В общем виде вероятностный автомат (англ. probabilistic automat) можно определить как дискретный потактный преобразователь информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано статистически.

Применение схем вероятностных автоматов (Р - схем) имеет важное значение для разработки методов проектирования дискретных систем, проявляющих статистически закономерное случайное поведение, для выяснения алгоритмических возможностей таких систем и обоснования границ целесообразности их использования, а также для решения задач синтеза по выбранному критерию дискретных стохастических систем, удовлетворяющих заданным ограничениям.

множество  $G$ , элементами которого являются всевозможные пары  $(x_i, z_s)$ , где  $x_i$  и  $z_s$  — элементы входного подмножества  $X$  и подмножества состояний  $Z$  соответственно. Если существуют две такие функции  $j$  и  $y$ , то с их помощью осуществляются отображения  $G \rightarrow Z$  и  $G \rightarrow Y$ , то говорят, что  $F = \langle Z, X, Y, \phi, \psi \rangle$  определяет автомат детерминированного типа.

Введем в рассмотрение более общую математическую схему. Пусть  $\Phi$  — множество всевозможных пар вида  $(x_i, y_j)$ , где  $y_j$  — элемент выходного подмножества  $Y$ . Потребуем, чтобы любой элемент множества  $G$  индуцировал на множестве  $\Phi$  некоторый закон распределения следующего вида:

$$\begin{array}{ccccccc} \text{Элементы из } \Phi & \cdots & (z_1, y_1) & \cdots & (z_1, y_2) & \cdots & (z_K, y_{J-1}) & (z_K, y_J) \\ (x_i, z_k) & \cdots & b_{11} & & b_{12} & \cdots & b_{K(J-1)} & b_{KJ} \end{array}$$

$$\sum_{i=1}^K \sum_{j=1}^J b_{ij} = 1$$

При этом  $b_{kj}$  — вероятности перехода автомата в состояние  $z_k$  появления на выходе сигнала  $y_j$  если он был в состоянии  $z_s$  и на его вход в этот момент времени поступил сигнал  $x_i$ . Число таких распределений, представленных в виде таблиц, равно числу элементов множества  $G$ . Обозначим множество этих таблиц через  $B$ . Тогда четверка элементов  $B = \langle Z, X, Y, B \rangle$  называется вероятностным автоматом (Р - автоматом).

Пусть элементы множества  $G$  индуцируют некоторые законы распределения на подмножествах  $Y$  и  $Z$ , что можно представить соответственно в виде:

$$\begin{array}{ccccccc} \text{Элементы из } Y & \cdots & y_1 & y_2 & \cdots & y_{J-1} & y_J \\ (x_i, z_s) & \cdots & q_1 & q_2 & \cdots & q_{J-1} & q_J \\ \text{Элементы из } Z & \cdots & z_1 & z_2 & \cdots & z_{K-1} & z_K \\ (x_i, z_s) & \cdots & z_1 & z_2 & \cdots & z_{K-1} & z_K \end{array}$$

$$\sum_{i=1}^K z_i = 1 \quad \text{и} \quad \sum_{i=1}^K q_i = 1$$

При этом  $z_k$  и  $q_k$  — вероятности перехода Р-автомата в состояние  $z_k$  и появления выходного сигнала  $y_k$  при условии, что Р-автомат находился в состоянии  $z_s$  и на его вход поступил входной сигнал  $x_i$ .

Если для всех  $k$  и  $j$  имеет место соотношение  $q_i z_i = b_{ij}$ , то такой Р-автомат называется вероятностным автоматом Милли. Это требование означает выполнение условия независимости распределений для нового состояния Р-автомата и его выходного сигнала.

определение выходного сигнала Р-автомата зависит лишь от того состояния, в котором находится автомат в данном такте работы. Другими словами, пусть каждый элемент выходного подмножества  $Y$  индуцирует распределение вероятностей выходов, имеющее следующий вид:

$$\begin{array}{ccccccc} \text{Элементы из } Y & \cdots & y_1 & y_2 & \cdots & y_{I-1} & y_I \\ z_k & \cdots & s_1 & s_2 & \cdots & s_{I-1} & s_I \end{array}$$

$$\sum_{i=1}^I s_i = 1$$

Здесь  $s_i$  — вероятность появления выходного сигнала  $y_i$  при условии, что Р-автомат находился в состоянии  $z_k$ .

для всех  $k$  и  $i$  имеет место соотношение  $z_i s_i = b_{ii}$ , то такой Р-автомат называется вероятностным автоматом Мура.

Понятие Р-автоматов Мили и Мура введено по аналогии с детерминированным F-автоматом, задаваемым  $F = \langle Z, X, Y, \Phi, \Psi \rangle$ . Частным случаем Р-автомата, задаваемого

как  $P = \langle Z, X, Y, B \rangle$  являются автоматы, у которых либо переход в новое состояние, либо выходной сигнал определяются детерминированно. Если выходной сигнал Р-автомата определяется детерминированно, то такой автомат называется Y-детерминированным вероятностным автоматом. Аналогично, Z-детерминированным вероятностным автоматом называется Р-автомат, у которого выбор нового состояния является детерминированным.

Y-детерминированный Р-автомат задаётся таблицей переходов и таблицей выходов. Первую из этих таблиц можно представить в виде квадратной матрицы размерности  $K \times K$ , которую будем называть матрицей переходных вероятностей или просто матрицей переходов Р-автомата. В общем случае такая матрица переходов имеет вид

$$P_P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1K} \\ p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots \\ p_{K1} & p_{K2} & \dots & p_{KK} \end{bmatrix}$$

Для описания Y-детерминированного Р-автомата необходимо задать начальное распределение вероятностей вида

$$\begin{matrix} Z & \dots & z_1 & z_2 & \dots & z_{K-1} & z_K \\ D & \dots & d_1 & d_2 & \dots & d_{K-1} & d_K \end{matrix}$$

Здесь  $d_k$  — вероятность того, что в начале работы Р-автомат находится в состоянии

$$k. \text{ При этом } \sum_{k=1}^K d_k = 1.$$

Y-детерминированный Р-автомат можно задать в виде ориентированного графа, вершины которого сопоставляются состояниям автомата, а дуги — возможным переходам из одного состояния в другое. Дуги имеют веса, соответствующие вероятностям перехода  $p_{ij}$ , а около вершин графа пишутся значения выходных сигналов, индуцируемых этими состояниями рис 1.

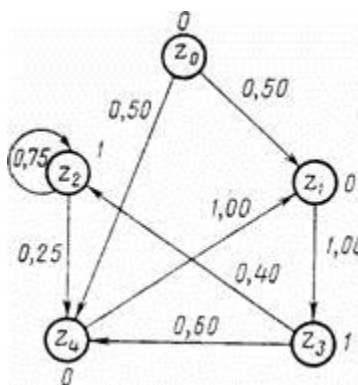


рис 1.

Р-автоматы могут использоваться как генераторы марковских последовательностей, которые необходимы при построении и реализации процессов функционирования систем  $S$  или воздействий внешней среды  $E$ .

Для оценки различных характеристик исследуемых систем, представляемых в виде Р-схем, кроме случая аналитических моделей можно применять и имитационные модели, реализуемые, например, методом статистического моделирования.

## 2.9 Практическая работа №9 (2 часа).

**Тема:** «Дискретно-стохастические модели (Р-схемы). Вероятностные автоматы»

**2.9.1 Цель работы:** Проследить разновидности вероятностных автоматов.

**2.9.2 Задачи работы:** Изучить применение схем вероятностных автоматов.

### 2.9.3 Описание (ход) работы:

При дискретно-стохастическом подходе к формализации процесса функционирования системы  $S$  подход остается аналогичный рассмотренному конечному автомату, то влияние фактора стохастичности можно проследить разновидности таких автоматов, а именно на вероятностных (стохастических) автоматах.

В общем виде вероятностный автомат (англ. probabilistic automat) можно определить как дискретный потактный преобразователь информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано статистически.

Применение схем вероятностных автоматов (Р - схем) имеет важное значение для разработки методов проектирования дискретных систем, проявляющих статистически закономерное случайное поведение, для выяснения алгоритмических возможностей таких систем и обоснования границ целесообразности их использования, а также для решения задач синтеза по выбранному критерию дискретных стохастических систем, удовлетворяющих заданным ограничениям.

множество  $G$ , элементами которого являются всевозможные пары  $(x_i, z_s)$ , где  $x_i$  и  $z_s$  — элементы входного подмножества  $X$  и подмножества состояний  $Z$  соответственно. Если существуют две такие функции  $j$  и  $u$ , то с их помощью осуществляются отображения  $G \rightarrow Z$  и  $G \rightarrow Y$ , то говорят, что  $F = \langle Z, X, Y, \phi, \psi \rangle$  определяет автомат детерминированного типа.

Введем в рассмотрение более общую математическую схему. Пусть  $\Phi$  — множество всевозможных пар вида  $(x_i, y_j)$ , где  $y_j$  — элемент выходного подмножества  $Y$ . Потребуем, чтобы любой элемент множества  $G$  индуцировал на множестве  $\Phi$  некоторый закон распределения следующего вида:

Элементы из $\Phi$	...	$(z_1, y_1)$	...	$(z_1, y_2)$	...	$(z_K, y_{J-1})$	$(z_K, y_J)$
$(x_i, z_k)$	...	$b_{11}$		$b_{12}$	...	$b_{K(J-1)}$	$b_{KJ}$

$$\sum_{i=1}^K \sum_{j=1}^J b_{ij} = 1$$

При этом  $b_{kj}$  — вероятности перехода автомата в состояние  $z_k$  появления на выходе сигнала  $y_j$  если он был в состоянии  $z_s$  и на его вход в этот момент времени поступил сигнал  $x_i$ . Число таких распределений, представленных в виде таблиц, равно числу элементов множества  $G$ . Обозначим множество этих таблиц через  $B$ . Тогда

четверка элементов  $B = \langle Z, X, Y, B \rangle$  называется вероятностным автоматом (Р - автоматом).

Пусть элементы множества  $G$  индуцируют некоторые законы распределения на подмножествах  $Y$  и  $Z$ , что можно представить соответственно в виде:

Элементы из $Y$	...	$y_1$	$y_2$	...	$y_{J-1}$	$y_J$
$(x_i, z_s)$	...	$q_1$	$q_2$	...	$q_{J-1}$	$q_J$
Элементы из $Z$	...	$z_1$	$z_2$	...	$z_{K-1}$	$z_K$
$(x_i, z_s)$	...	$z_1$	$z_2$	...	$z_{K-1}$	$z_K$



При этом  $\sum_{k=1}^K z_k = 1$  и  $\sum_{k=1}^K q_k = 1$  где  $z_k$  и  $q_k$  — вероятности перехода Р-автомата в состояние  $z_k$  и появления выходного сигнала  $y_k$  при условии, что Р-автомат находился в состоянии  $z_s$  и на его вход поступил входной сигнал  $x_i$ .

Если для всех  $k$  и  $j$  имеет место соотношение  $q_k z_k = b_{kj}$ , то такой Р-автомат называется вероятностным автоматом Мили. Это требование означает выполнение условия независимости распределений для нового состояния Р-автомата и его выходного сигнала.

определение выходного сигнала Р-автомата зависит лишь от того состояния, в котором находится автомат в данном такте работы. Другими словами, пусть каждый элемент выходного подмножества  $Y$  индуцирует распределение вероятностей выходов, имеющее следующий вид:

$$\begin{array}{ccccccc} \text{Элементы из } Y & \dots & y_1 & y_2 & \dots & y_{K-1} & y_K \\ z_k & \dots & s_1 & s_2 & \dots & s_{I-1} & s_I \end{array}$$

$$\sum_{i=1}^I s_i = 1$$

Здесь  $s_i$  — вероятность появления выходного сигнала  $y_i$  при условии, что Р-автомат находился в состоянии  $z_k$ .

для всех  $k$  и  $i$  имеет место соотношение  $z_k s_i = b_{ki}$ , то такой Р-автомат называется вероятностным автоматом Мура.

Понятие Р-автоматов Мили и Мура введено по аналогии с детерминированным F-автоматом, задаваемым  $F = \langle Z, X, Y, \Phi, \Psi \rangle$ . Частным случаем Р-автомата, задаваемого

как  $P = \langle Z, X, Y, B \rangle$  являются автоматы, у которых либо переход в новое состояние, либо выходной сигнал определяются детерминированно. Если выходной сигнал Р-автомата определяется детерминированно, то такой автомат называется Y-детерминированным вероятностным автоматом. Аналогично, Z-детерминированным вероятностным автоматом называется Р-автомат, у которого выбор нового состояния является детерминированным.

Y-детерминированный Р-автомат задаётся таблицей переходов и таблицей выходов. Первую из этих таблиц можно представить в виде квадратной матрицы размерности  $K \times K$ , которую будем называть матрицей переходных вероятностей или просто матрицей переходов Р-автомата. В общем случае такая матрица переходов имеет вид

$$P_P = \begin{vmatrix} p_{11} & p_{12} & \dots & p_{1K} \\ p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots \\ p_{K1} & p_{K2} & \dots & p_{KK} \end{vmatrix}$$

Для описания Y-детерминированного Р-автомата необходимо задать начальное распределение вероятностей вида

$$\begin{array}{ccccccc} Z & \dots & z_1 & z_2 & \dots & z_{K-1} & z_K \\ D & \dots & d_1 & d_2 & \dots & d_{K-1} & d_K \end{array}$$

Здесь  $d_k$  — вероятность того, что в начале работы Р-автомат находится в состоянии

$$k. \text{ При этом } \sum_{k=1}^K d_k = 1.$$

Y-детерминированный Р-автомат можно задать в виде ориентированного графа, вершины которого сопоставляются состояниям автомата, а дуги — возможным переходам из одного состояния в другое. Дуги имеют веса, соответствующие вероятностям

перехода  $p_{ij}$ , а около вершин графа пишутся значения выходных сигналов, индуцируемых этими состояниями рис 1.

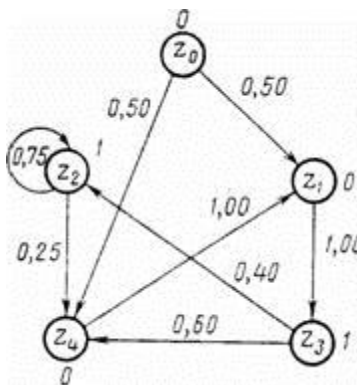


рис 1.

P-автоматы могут использоваться как генераторы марковских последовательностей, которые необходимы при построении и реализации процессов функционирования систем  $S$  или воздействий внешней среды  $E$ .

Для оценки различных характеристик исследуемых систем, представляемых в виде P-схем, кроме случая аналитических моделей можно применять и имитационные модели, реализуемые, например, методом статистического моделирования.

## 2.10 Практическая работа №10 (2 часа).

**Тема:** «Непрерывно-стохастические модели (Q-схемы)»

**2.10.1 Цель работы:** Рассмотреть непрерывно-стохастические модели на примере систем массового обслуживания.

**2.10.2 Задачи работы:** Изучение непрерывно - стохастических моделей.

**2.10.3 Описание (ход) работы:**

При непрерывно-стохастическом подходе в качестве типовых математических схем применяется система массового обслуживания (англ. queueingsystem), которые будем называть Q-схемами. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

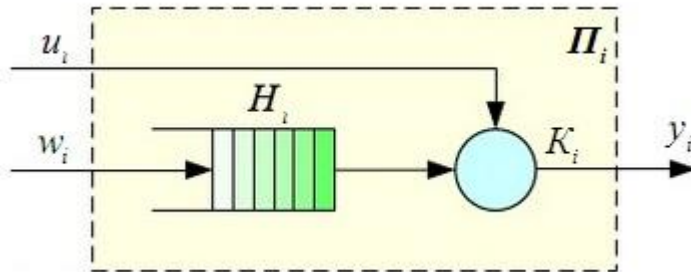
В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д.

При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. Остановимся на основных понятиях массового обслуживания, необходимых для использования Q-схем, как при аналитическом, так и при имитационном.

В любом элементарном акте обслуживания можно выделить две основные составляющие:

- ожидание обслуживания заявки;
- собственно обслуживание заявки.

Это можно изобразить в виде некоторого  $i$ -го прибора обслуживания  $\Pi_i$  (рис. 1), состоящего из накопителя заявок  $H_i$ , в котором может одновременно находиться  $l_i = 0, L_i^H$  заявок, где  $L_i^H$  — емкость  $i$ -го накопителя, и канала обслуживания заявок (или просто канала)  $K_i$ . На каждый элемент прибора обслуживания  $\Pi_i$ , поступают потоки событий: в накопитель  $H_i$  — поток заявок  $w_i$  на канал  $K_i$  — поток обслуживания  $u_i$ .



Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Поток событий называется однородным, если он характеризуется только моментами поступления этих событий (вызывающими моментами). Поток неоднородных событий характеризуется моментами поступления этих событий и набором признаков события. Например, применительно к процессу обслуживания для неоднородного потока заявок могут быть заданы принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т.п.

Обычно в приложениях при моделировании различных систем применительно к элементарному каналу обслуживания  $K_i$ , можно считать, что поток заявок  $w_i \in W$ , т.е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе  $K_i$  образует подмножество неуправляемых переменных, а поток обслуживания  $u_i \in U$ , т.е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных. Заявки, обслуженные каналом  $K_i$  и заявки, покинувшие прибор  $\Pi_i$  по различным причинам необслуженными (например, из-за переполнения накопителя  $H_i$ ), образуют выходной поток  $y_i \in Y$ , т.е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а Q-схемы, образуемые композицией многих элементарных приборов обслуживания (сети массового обслуживания). Для задания Q-схемы необходимо использовать оператор сопряжения R, отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой.

Для задания Q-схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе  $H_i$  и обслуживания заявок каналом  $K_i$ . Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов. В зависимости от динамики приоритетов в Q-схемах различают статические и динамические приоритеты. Статические приоритеты назначаются заранее и не зависят от состояний Q-схемы, т.е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя  $H_i$  на обслуживание каналом  $K_i$  можно выделить относительные и абсолютные приоритеты. Относительный приоритет означает, что заявка с более высоким приоритетом, поступившая в накопитель  $H_i$ , ожидает окончания обслуживания предшествующей заявки каналом  $K_i$  и только после этого

занимает канал. Абсолютный приоритет означает, что заявка с более высоким приоритетом, поступившая в накопитель  $H_i$ , прерывает обслуживание каналом  $K_i$  заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из  $K_i$  заявка может либо покинуть систему, либо может быть снова записана на какое-то место в  $H_i$ ).

При рассмотрении алгоритмов функционирования приборов обслуживания  $\Pi_i$  необходимо также задать набор правил, по которым заявки покидают  $H_i$  и  $K_i$ .

Весь набор возможных алгоритмов поведения заявок в Q-схеме можно представить в виде некоторого оператора алгоритмов поведения заявок  $A$ .

Таким образом, Q-схема, описывающая процесс функционирования системы массового обслуживания любой сложности, однозначно задается в виде

$$Q = \langle W, U, H, Z, R, A \rangle$$

Возможности оценки характеристик с использованием аналитических моделей теории массового обслуживания являются весьма ограниченными по сравнению с требованиями практики исследования и проектирования систем, формализуемых в виде Q-схем. Несравненно большими возможностями обладают имитационные модели, позволяющие исследовать Q-схему без ограничений. На работу с Q-схемами при машинной реализации моделей ориентированы многие языки имитационного моделирования например SIMULA, SIMSCRIPT, GPSS и др.

## **2.11 Практическая работа №11 (2 часа).**

**Тема:** «Обобщённые модели (А-схемы)»

**2.11.1 Цель работы:** Изучение обобщённых моделей (А-схемы).

**2.11.2 Задачи работы:** Описать поведение непрерывных и дискретных, детерминированных и стохастических систем.

### **2.11.3 Описание (ход) работы:**

Этот подход позволяет описывать поведение непрерывных и дискретных, детерминированных и стохастических систем, т. е. по сравнению с рассмотренными является обобщенным (универсальным) и базируется на понятии агрегативной системы (от англ. aggregatesystem), представляющей собой формальную схему общего вида, которую будем называть А-схемой .

Анализ существующих средств моделирования систем и задач, решаемых с помощью метода моделирования на ЭВМ, неизбежно приводит к выводу, что комплексное решение проблем, возникающих в процессе создания и машинной реализации модели, возможно лишь в случае, если моделирующие системы имеют в своей основе единую формальную математическую схему, т. е. А-схему.

Такая схема должна одновременно выполнять несколько функций:

- являться адекватным математическим описанием системы  $S$ ;
- служить основой для построения алгоритмов и программ при машинной реализации модели  $M$ ;
- позволять в упрощенном варианте (для частных случаев) проводить аналитические исследования.
- при агрегативном подходе сначала дается формальное определение объекта моделирования – агрегативной системы, которая является математической схемой, отображающей системный характер изучаемых объектов.

При агрегативном описании сложный объект (система) разбивается на конечное число частей (подсистем), сохраняя при этом связи, обеспечивающие их взаимодействие. Если некоторые из полученных подсистем оказываются в свою очередь еще достаточно сложными, то процесс их разбиения продолжается до тех пор пока не образуются

подсистемы, которые в условиях рассматриваемой задачи моделирования могут считаться удобными для математического описания. В результате такой декомпозиции сложная система представляется в виде многоуровневой конструкции из взаимосвязанных элементов, объединенных в подсистемы различных уровней.

В качестве элемента А-схемы выступает агрегат, а связь между агрегатами осуществляется с помощью оператора сопряжения R. агрегат сам может рассматриваться как А-схема, т. е. может разбиваться на элементы (агрегаты) следующего уровня.

Любой агрегат характеризуется следующими множествами: моментов времени T, входных X и выходных Y сигналов, состояний Z в каждый момент времени t. Состояние агрегата в момент времени t обозначается как  $z(t) \in Z$ , а входные и выходные сигналы — как  $x(t) \in X$  и  $y(t) \in Y$  соответственно.

Будем полагать, что переход агрегата из состояния  $z(t_1)$  в состояние  $z(t_2) \neq z(t_1)$  происходит за малый интервал времени, т. е. имеет место скачок  $\delta z$ . Переходы агрегата из состояния  $z(t_1)$  в  $z(t_2)$  определяются собственными (внутренними) параметрами самого агрегата  $h(t) \in H$  и входными сигналами  $x(t) \in X$ .

Для описания скачков состояний  $\delta z$  в особые моменты времени  $t\delta$  будем использовать случайный оператор W, представляющий собой частный случай оператора U, т. е.

$$z(t\delta + 0) = W[t\delta, z(t\delta)].$$

В множестве состояний Z выделяется такое подмножество  $Z(Y)$ , что если  $z(t\delta)$  достигает  $Z(Y)$ , то это состояние является моментом выдачи выходного сигнала, определяемого оператором выходов

$$y = G[t\delta, z(t\delta)].$$

Таким образом, под агрегатом будем понимать любой объект, определяемый упорядоченной совокупностью рассмотренных множеств T, X, Y, Z,  $Z(Y)$ , H и случайных операторов V, U, W, G.

Последовательность входных сигналов, расположенных в порядке их поступления в А-схему, будем называть входным сообщением или x - сообщением. Последовательность выходных сигналов, упорядоченную относительно времени выдачи, назовем выходным сообщением или y - сообщением.

Существует класс больших систем, которые ввиду их сложности не могут быть формализованы в виде математических схем одиночных агрегатов, поэтому их формализуют некоторой конструкцией из отдельных агрегатов  $A_n$ , которую назовем агрегативной системой или А-схемой. Для описания некоторой реальной системы S в виде А-схемы необходимо иметь описание как отдельных агрегатов  $A_n$ , так и связей между ними.

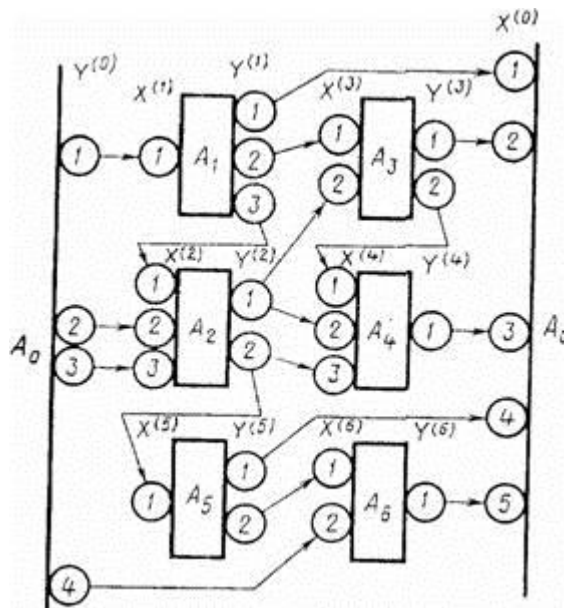
Для построения агрегата вводятся предположения о закономерностях функционирования А-схем, в соответствии реальной системой:

1) взаимодействие между А-схемой и внешней средой E, а также между отдельными агрегатами внутри системы S осуществляется при передаче сигналов, причем взаимные влияния, имеющие место вне механизма обмена сигналами, не учитываются;

2) для описания сигнала достаточно некоторого конечного набора характеристик;

3) элементарные сигналы мгновенно передаются в А-схеме независимо друг от друга по элементарным каналам;

4) к входному контакту любого элемента А-схемы подключается не более чем один элементарный канал, к выходному контакту - любое конечное число элементарных каналов при условии, что ко входу одного и того же элемента А-схемы направляется не более чем один из упомянутых элементарных каналов.



Структура агрегативной системы

Взаимодействие А-схемы с внешней средой Е рассматривается как обмен сигналами между внешней средой Е и элементами А-схемы. В соответствии с этим внешнюю среду Е можно представить в виде фиктивного элемента системы  $A_0$ , вход которого содержит  $I_0$  входных контактов, а выход —  $J_0$  выходных контактов. Сигнал, выдаваемый А-схемой во внешнюю среду Е, принимается элементом  $A_0$  как входной сигнал, состоящий из элементарных сигналов  $x_1(0)(t)$ ,  $x_2(0)(t)$ , ...,  $x_{I_0}(0)(t)$ . Сигнал, поступающий в А-схему из внешней среды Е, является выходным сигналом элемента  $A_0$  и состоит из элементарных сигналов  $y_1(0)(t)$ ,  $y_2(0)(t)$ , ...,  $y_{J_0}(0)(t)$ .

Таким образом, каждый  $A_n$  (в том числе и  $A_0$ ) как элемент А-схемы в рамках принятых предположений о механизме обмена сигналами достаточно охарактеризовать множеством входных контактов  $X_1(n)$ ,  $X_2(n)$ , ...,  $X_{I_n}(n)$  которое обозначим  $\{X_i(n)\}$ , и множеством выходных контактов  $Y_1(n)$ ,  $Y_2(n)$ , ...,  $Y_{J_n}(n)$  которое обозначим  $\{Y_j(n)\}$ , где . Полученная пара множеств  $\{X_i(n)\}$ ,  $\{Y_j(n)\}$  является математической моделью элемента  $A_n$  используемого для формального описания сопряжения его с прочими элементами А-схемы и внешней средой Е.

Если в А-схеме к контакту  $X_i(n)$  не подключен никакой элементарный канал, то оператор  $R$  не определен на этом контакте  $X_i(n)$ . Оператор  $R$  называется оператором сопряжения элементов (агрегатов) в А-схему. Совокупность множеств  $\{X_i(n)\}$ ,  $\{Y_l(k)\}$  и оператор  $R$  образуют схему сопряжения элементов в систему  $S$ . Оператор сопряжения  $R$  можно задать в виде таблицы, в которой на пересечении строк с номерами элементов (агрегатов)  $n$  и столбцов с номерами контактов  $i$  располагаются пары чисел  $k, l$ , указывающие номер элемента  $k$  и номер контакта  $l$ , с которым соединен контакт  $X_i(n)$ .

Таким образом, использование обобщенной типовой математической схемы моделирования, т. е. А-схемы, в принципе не отличается от рассмотренных ранее D-, F-, P-, N-, Q-схем. Для частного случая, а именно для кусочно-линейных агрегатов, результаты могут быть получены аналитическим методом. В более сложных случаях, когда применение аналитических методов неэффективно или невозможно, прибегают к имитационному методу, причем представление объекта моделирования в виде А-схемы может являться тем фундаментом, на котором базируется построение имитационной системы и ее внешнего и внутреннего математического обеспечения. Стандартная форма представления исследуемого объекта в виде А-схемы приводит к унификации не только алгоритмов имитации, но и к возможности применять стандартные методы обработки и анализа результатов моделирования системы  $S$ .

Рассмотренные примеры использования типовых математических схем (F-, D, Q-, N-, A-схем) позволяют формализовать достаточно широкий класс больших систем, с которыми приходится иметь дело в практике исследования и проектирования сложных систем.

## **2.12 Практическая работа №12 (2 часа).**

**Тема:** «Последовательность разработки и машинной реализации моделей»

**2.12.1 Цель работы:** Рассмотреть алгоритм реализации машинных моделей.

**2.12.2 Задачи работы:** Реализация моделей с помощью ЭВМ.

**2.12.3 Описание (ход) работы:**

В настоящее время наиболее эффективным методом исследования систем является машинное моделирование, без которого невозможно решение многих задач.

Моделирование с использованием компьютера позволяет исследовать механизм явлений, протекающих в реальном объекте с большими или малыми скоростями, когда в натурных экспериментах с объектом трудно (или невозможно) проследить за изменениями, протекающими в течение короткого времени, или когда получение достоверных результатов сопряжено с длительным экспериментом.

Сущность машинного моделирования системы состоит в проведении на компьютере эксперимента с моделью, которая представляет собой некоторый программный комплекс, описывающий формально и (или) алгоритмически поведение элементов системы в процессе её функционирования, т.е. в их взаимодействии друг с другом и внешней средой.

Основные требования, предъявляемые к модели процесса функционирования системы.

1. Полнота модели должна предоставлять исследователю возможность получения необходимого набора оценок характеристик системы с требуемой точностью и достоверностью.

2. Гибкость модели должна давать возможность воспроизведения различных ситуаций при варьировании структуры, алгоритмов и параметров системы.

3. Длительность разработки и реализации модели системы должна быть по возможности минимальной при учёте ограничений на имеющиеся ресурсы.

4. Структура модели должна быть блочной, т.е. допускать возможность замены, добавления и исключения некоторых частей без переделки всей модели.

5. Информационное обеспечение должно предоставлять возможность эффективной работы модели с базой данных систем определённого класса.

6. Программные и технические средства должны обеспечивать эффективную (по быстродействию и памяти) машинную реализацию модели и удобное общение с ней пользователя.

7. Должно быть реализовано проведение целенаправленных (планируемых) машинных экспериментов с моделью системы с использованием аналитико-имитационного подхода при наличии ограничений.

При машинном моделировании системы характеристики процесса её функционирования определяются на основе модели, построенной исходя из имеющейся исходной информации об объекте моделирования. При получении новой информации об объекте его модель пересматривается и уточняется, т.е. процесс моделирования является итерационным. Этот итерационный процесс продолжается до тех пор, пока не будет получена модель, которую можно считать адекватной в рамках решения поставленной задачи исследования и проектирования системы.

Моделирование систем на ЭВМ целесообразно использовать в следующих случаях:

а) для исследования системы до того, как она спроектирована, с целью оценки эффективности будущей системы, а также определения чувствительности характеристик системы к изменениям структуры, алгоритмов и параметров объекта моделирования и внешней среды;

б) на этапе проектирования системы для анализа и синтеза различных вариантов системы и выбора среди конкурирующих такого варианта, который удовлетворял бы заданному критерию оценки эффективности системы при принятых ограничениях;

в) после завершения проектирования и внедрения системы, т.е. при её эксплуатации, для получения информации, дополняющей результаты натурных испытаний реальной системы, и для получения прогнозов эволюции системы во времени.

Основными этапами моделирования систем являются [8]:

- построение концептуальной модели системы и её формализация;
- алгоритмизация модели системы и её машинная реализация;
- получение и интерпретация результатов моделирования системы.

Взаимосвязь перечисленных этапов моделирования систем и их составляющих (подэтапов) представлена в виде сетевого графика (рис. 3.1).

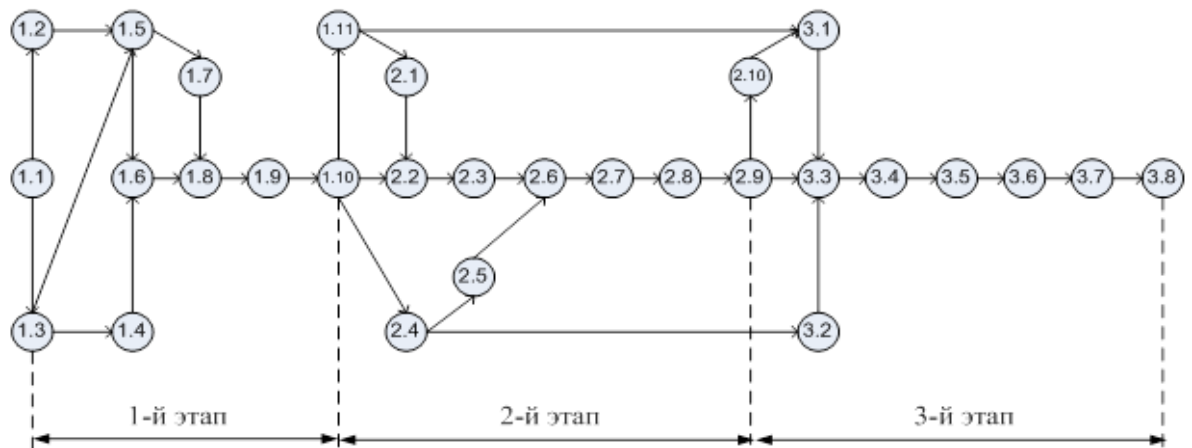


Рис. 3.1. Взаимосвязь этапов моделирования систем

К подэтапам относятся: 1.1 – постановка задачи машинного моделирования системы; 1.2 – анализ задачи моделирования системы; 1.3 – определение требований к исходной информации об объекте моделирования и организация её сбора; 1.4 – выдвижение гипотез и принятие предложений; 1.5 – определение параметров и переменных модели; 1.6 – установление основного содержания модели; 1.7 – обоснование критериев оценки эффективности системы; 1.8 – определение процедур аппроксимации; 1.9 – описание концептуальной модели системы; 1.10 – проверка достоверности концептуальной модели; 1.11 – составление технической документации по первому этапу; 2.1 – построение логической схемы модели; 2.2 – получение математических соотношений; 2.3 – проверка достоверности модели системы; 2.4 – выбор вычислительных средств для моделирования; 2.5 – составление плана выполнения работ по программированию; 2.6 – построение схемы программы; 2.7 – проверка достоверности схемы программы; 2.8 – проведение программирования модели; 2.9 – проверка достоверности программы; 2.10 – составление технической документации по второму этапу; 3.1 – планирование машинного эксперимента с моделью системы; 3.2 – определение требований к вычислительным средствам; 3.3 – проведение рабочих расчётов; 3.4 – анализ результатов моделирования системы; 3.5 – представление результатов моделирования; 3.6 – интерпретация результатов моделирования; 3.7 –



подведение итогов моделирования и выдача рекомендаций; 3.8 – составление технической документации по третьему этапу.

Таким образом, процесс моделирования системы сводится к выполнению перечисленных подэтапов, сгруппированных в виде трёх этапов. На этапе построения концептуальной модели  $M_K$  и её формализации проводится исследование моделируемого объекта с точки зрения выделения основных составляющих процесса его функционирования, определяются необходимые аппроксимации и получается обобщённая схема модели системы, которая преобразуется в машинную модель  $M_M$  на втором этапе моделирования путём последовательной алгоритмизации и программирования модели. Последний этап моделирования системы сводится к проведению согласно разработанному плану рабочих расчётов на ЭВМ с использованием выбранных программно-технических средств, получению и интерпретации результатов моделирования системы. Очевидно, что при построении модели и её машинной реализации при получении новой информации возможен пересмотр ранее принятых решений.

### **2.13 Практическая работа №13 (2 часа).**

**Тема:** «Построение концептуальной модели системы и её формализация»

**2.13.1 Цель работы:** Изучение концептуальной модели.

**2.13.2 Задачи работы:** Построение концептуальной модели.

**2.13.3 Описание (ход) работы:**

На первом этапе проведения моделирования конкретного объекта (системы) на базе ЭВМ необходимо построить концептуальную (содержательную) модель  $M$  процесса функционирования системы, а затем провести её формализацию, т. е. основным содержанием этапа является переход от словесного описания объекта моделирования к его математической (аналитико-имитационной) модели. Наиболее ответственными моментами в этой работе является упрощение описания системы, т. е. отделение собственно системы  $S$  от внешней среды  $E$  и выбор основного содержания модели путем отбрасывания всего второстепенного с точки зрения поставленной цели моделирования.

Постановка задачи машинного моделирования. Необходимо дать четкую формулировку задачи, обратив особое внимание на существование такой задачи и необходимость машинного моделирования, на выбор методики решения задачи с учетом имеющихся машинных ресурсов, на определение масштаба задачи и возможность ее разбиения на подзадачи.

Анализ задачи моделирования. Проведение детального анализа поставленной задачи моделирования должно способствовать преодолению дальнейших трудностей с минимальными затратами. На этом подэтапе работа по анализу задачи сводится к выбору критериев оценки процесса функционирования исследуемой системы  $S$  (если они не заданы), выделению эндогенных и экзогенных переменных модели  $M$ , выбору возможных методов идентификации, выполнению предварительного анализа следующих двух этапов моделирования.

Определение требований к исходной информации. Необходимо сформулировать требования к исходной информации об объекте моделирования, оценить достаточность имеющейся информации об объекте для его машинного моделирования, подготовить имеющиеся априорные сведения об объекте, провести анализ имеющихся экспериментальных данных о подобном классе систем.

Выдвижение гипотез и принятие предположений. При выдвижении гипотез и принятии предположений учитывается следующее: объем априорной информации для решения задачи; подзадачи, для решения которых информации недостаточно; ограничения на ресурсы при решении задачи; ожидаемые результаты моделирования.

Гипотезы при построении модели системы  $S$  служат для заполнения “пробелов” в понимании задачи моделирования исследователем, а предположения принимаются относительно известных данных, которые не отвечают требованиям (ограничениям, ресурсам) решаемой задачи и служат обычно для упрощения модели.

Определение параметров и переменных. Прежде чем перейти к описанию математической модели необходимо определить параметры системы, входные и выходные переменные, воздействия внешней среды. Описание каждого параметра и переменной дается в следующей форме: определение и краткая характеристика; символ обозначения и единица измерения; диапазон изменения (для переменных); место применения в модели.

Установление основного содержания модели. На этом подэтапе определяется основное содержание и выбирается метод построения модели системы, которые разрабатываются на основе принятых гипотез и предположений. При этом учитываются следующие особенности: формулировка задачи моделирования системы; структура системы  $S$  и алгоритмы ее поведения, воздействия внешней среды  $E$ ; возможные методы и средства решения задачи моделирования.

Обоснование критериев оценки эффективности системы. Для возможности оценки качества процесса функционирования моделируемой системы  $S$  необходимо выбрать некоторую совокупность критериев оценки эффективности, т. е. в математической постановке задача сводится к получению соотношения для оценки эффективности как функции параметров и переменных системы, с учетом воздействий внешней среды  $E$ .

Определение процедур аппроксимации. Для возможности получения числовых значений интересующих характеристик системы  $S$  необходимо в процессе моделирования провести аппроксимации, для чего обычно используются: детерминированная, когда результаты моделирования однозначно определяются по данной совокупности входных воздействий и параметров системы; вероятностная, когда предполагается, что случайные элементы влияют на результаты моделирования и необходимо получить информацию о законах распределения выходных переменных; определения средних значений, когда при моделировании представляют интерес средние значения выходных переменных при наличии случайных факторов.

Описание концептуальной модели. Проводится описание концептуальной модели в абстрактных терминах и понятиях с использованием типовых математических схем, т. е. реализуется переход к математической модели системы, окончательно принимаются гипотезы и предположения, обосновывается выбор процедур аппроксимации реальных процессов при построении модели.

Проверка достоверности модели. Проверять достоверность концептуальной модели достаточно сложно, так как процесс ее построения является эвристическим, и такая модель описывается в абстрактных терминах и понятиях. Проверка достоверности должна включать: проверку замысла модели; оценку достоверности исходной информации; рассмотрение постановки задачи моделирования; анализ принятых аппроксимаций; исследование гипотез и предположений.

Составление технической документации. Технический отчет по этапу включает в себя: подробную постановку задачи моделирования системы  $S$ ; анализ задачи моделирования системы; критерии оценки эффективности системы; параметры и переменные модели системы; гипотезы и предположения, принятые при построении модели; описание модели в абстрактных терминах и понятиях; описание ожидаемых результатов моделирования системы.

## **2.14 Практическая работа №14 (2 часа).**

**Тема:** «Алгоритмизация модели и её машинная реализация»

**2.14.1 Цель работы:** Рассмотреть алгоритм машинной реализации.

**2.14.2 Задачи работы:** Реализация идей и математических схем в виде машинной модели.

**2.14.3 Описание (ход) работы:**

На втором этапе моделирования системы математическая модель, сформулированная ранее, воплощается в конкретную машинную модель. Этап представляет собой практическую деятельность, направленную на реализацию идей и математических схем в виде машинной модели, ориентированной на использование конкретных программно-технических средств.

Построение логической схемы модели. Рекомендуется строить модель по блочному принципу. Блоки такой модели бывают двух типов: основные и вспомогательные. Каждый основной блок соответствует некоторому реальному подпроцессу, имеющему место в моделируемой системе  $S$ , а вспомогательные блоки представляют собой лишь составную часть машинной модели, они не отражают функции моделируемой системы и необходимы лишь для машинной реализации, фиксации и обработки результатов моделирования.

Получение математических соотношений. Одновременно с выполнением подэтапа 2.1 необходимо, где это возможно, получить математические соотношения в виде явных функций, т. е. построить аналитические модели. Схема машинной модели должна представлять собой полное отражение заложенной в модели концепции и иметь описание всех блоков модели с их наименованиями; единую систему обозначений и нумерацию блоков; отражение логики модели; задание математических соотношений в явном виде.

Проверка достоверности модели. Проверка должна дать ответ на вопрос, насколько логическая схема модели системы и используемые математические соотношения отражают замысел модели, сформированный на первом этапе. При этом проверяются: возможность решения поставленной задачи; точность отражения замысла в логической схеме; полнота логической схемы модели; правильность используемых математических соотношений.

Выбор вычислительных средств для моделирования. Необходимо окончательно решить вопрос о выборе ЭВМ для реализации модели на основе следующих требований: наличие необходимого математического обеспечения; доступность выбранной ЭВМ для разработчика модели; обеспечение всех этапов реализации модели; возможность своевременного получения результатов моделирования.

Составления плана выполнения работ по программированию. План при использовании ЭВМ должен включать в себя: выбор языка (системы) программирования модели; указание типа ЭВМ и необходимых для моделирования устройств; оценку примерного объема необходимой памяти; ориентировочные затраты машинного времени на моделирование, программирование и отладку программы на ЭВМ.

Построение схемы программы. Схема программы должна отражать: разбиение модели на блоки, подблоки и т. д.; особенности программирования модели; проведение необходимых изменений; возможности тестирования программы; оценку затрат машинного времени; форму представления входных и выходных данных. Особое внимание должно быть уделено особенностям выбранного для реализации модели языка.

Проверка достоверности схемы программы. При этом проводится проверка соответствия каждой операции, представленной в схеме программы, аналогичной ей операции в логической схеме модели.

Проведение программирования модели. Если имеется адекватная схема программы, то программирование представляет собой работу только для программиста без участия и помощи со стороны разработчика модели.

Проверка достоверности программы. Проводится: обратным переводом программы в исходную схему; проверкой отдельных частей программы при решении различных тестовых задач; объединением всех частей программы и проверкой ее в целом на контрольном примере моделирования варианта системы  $S$ .

Составление технической документации. Документация содержит: логическую схему модели и ее описание; адекватную схему программы и принятые обозначения; полный текст программы; перечень входных и выходных величин с пояснениями; инструкцию по работе с программой; оценку затрат машинного времени на моделирование с указанием требуемых ресурсов ЭВМ.

## **2.15 Практическая работа №15 (2 часа).**

**Тема:** «Получение и интерпретация результатов моделирования»

**2.15.1 Цель работы:** Рассмотреть этапы получения и интерпретации результатов моделирования.

**2.15.2 Задачи работы:** Построение и реализация модели системы S.

### **2.15.3 Описание (ход) работы:**

На третьем этапе моделирования — этапе получения и интерпретации результатов моделирования — ЭВМ используется для проведения рабочих расчетов по составленной и отлаженной программе. Результаты этих расчетов позволяют проанализировать и сформулировать выводы о характеристиках процесса функционирования моделируемой системы S.

Планирование машинного эксперимента. Должен быть составлен план проведения эксперимента с моделью системы S с указанием комбинаций переменных и параметров, для которых проводится моделирование. Различают стратегическое и тактическое планирование машинного эксперимента. При стратегическом планировании ставится задача построения оптимального плана эксперимента для достижения цели, поставленной перед моделированием (например, оптимизация структуры системы S). Тактическое планирование машинного эксперимента преследует частные цели (например, определение необходимого числа реализаций при заданной точности и достоверности результатов моделирования).

Определение требований к вычислительным средствам. Окончательно формулируются требования по времени загрузки вычислительных средств с указанием внешних устройств.

Проведение рабочих расчетов. Расчеты обычно включают в себя: подготовку наборов исходных данных для ввода в ЭВМ; проверку исходных данных, подготовленных для ввода; проведение расчетов на ЭВМ; получение выходных данных, т. е. результатов моделирования.

Анализ результатов моделирования системы. Планирование машинного эксперимента позволяет вывести необходимое количество данных и определить метод их анализа. Следует выводить результаты, которые нужны для проведения дальнейшего анализа, и полнее использовать возможности ЭВМ для статистической обработки результатов моделирования и представления их в наиболее наглядной форме, например в виде графиков, гистограмм, схем и т. п. Вычисление статистических характеристик перед выводом результатов повышает эффективность применения машины и сводит к минимуму обработку выходной информации после ее выдачи пользователю.

Представление результатов моделирования. Целесообразно в каждом конкретном случае выбрать наиболее подходящую форму (таблицы, графики, диаграммы, гистограммы, схемы и т. п.). В большинстве случаев наиболее простой формой считаются таблицы, хотя графики более наглядно иллюстрируют результаты моделирования системы S.

Интерпретация результатов моделирования. Основное содержание этого подэтапа — переход от информации, полученной в результате машинного эксперимента с моделью

М, к информации применительно к объекту моделирования, на основании которой и будут делаться выводы относительно характеристик процесса функционирования исследуемой системы S.

Подведение итогов моделирования. Должны быть отмечены главные особенности полученных в соответствии с планом эксперимента результатов, проведена проверка гипотез и предположений и сделаны выводы на основании полученных результатов моделирования.

Составление технической документации. Документация должна включать в себя: план проведения машинного эксперимента; наборы исходных данных для моделирования; результаты моделирования системы; анализ и оценку результатов моделирования; выводы по полученным результатам моделирования; указание путей дальнейшего совершенствования машинной модели и возможных областей ее приложения.

Таким образом, процесс моделирования системы S сводится к выполнению перечисленных этапов моделирования. На этапе построения концептуальной модели проводится исследование моделируемого объекта, определяются необходимые аппроксимации и строится обобщенная схема модели, которая преобразуется в машинную модель на втором этапе моделирования путем последовательного построения логической схемы модели и схемы программы. На последнем этапе моделирования проводят рабочие расчеты на ЭВМ, получают и интерпретируют результаты моделирования системы.

Рассмотренная последовательность этапов и подэтапов отражает наиболее общий подход к построению и реализации модели системы S.

## **2.16 Практическая работа №16 (2 часа).**

**Тема:** «Имитационное моделирование»

**2.16.1 Цель работы:** Овладение навыком имитационного моделирования.

**2.16.2 Задачи работы:** Провести серию вычислительных экспериментов.

**2.16.3 Описание (ход) работы:**

*Компьютерное моделирование* как новый метод научных исследований основывается на:

1. *построении математических моделей* для описания изучаемых процессов;
2. *использовании новейших вычислительных машин*, обладающих высоким быстродействием (миллионы операций в секунду) и способных вести диалог с человеком.

Суть *компьютерного моделирования* состоит в следующем: на основе математической модели с помощью ЭВМ проводится серия вычислительных экспериментов, т.е. исследуются свойства объектов или процессов, находятся их оптимальные параметры и режимы работы, уточняется модель. Например, располагая уравнением, описывающим протекание того или иного процесса, можно изменяя его *коэффициенты*, начальные и граничные условия, исследовать, как при этом будет вести себя *объект*. *Имитационные модели* - это проводимые на ЭВМ *вычислительные эксперименты* с математическими моделями, имитирующими поведение реальных объектов, процессов или систем.

Реальные процессы и системы можно исследовать с помощью двух типов математических моделей: аналитических и имитационных.

В аналитических моделях поведение реальных процессов и систем (РПС) задается в виде явных *функциональных зависимостей* (уравнений линейных или нелинейных, дифференциальных или интегральных, систем этих уравнений). Однако получить эти зависимости удастся только для сравнительно простых РПС. Когда явления сложны и многообразны исследователю приходится идти на упрощенные представления сложных РПС. В результате аналитическая модель становится слишком грубым приближением к

действительности. Если все же для сложных РПС удастся получить аналитические модели, то зачастую они превращаются в трудноразрешимую проблему. Поэтому исследователь вынужден часто использовать *имитационное моделирование*.

*Имитационное моделирование* представляет собой численный метод проведения на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими поведение реальных объектов, процессов и систем во времени в течение заданного периода. При этом функционирование РПС разбивается на элементарные явления, подсистемы и модули. Функционирование этих элементарных явлений, подсистем и модулей описывается набором алгоритмов, которые имитируют элементарные явления с сохранением их *логической структуры* и последовательности протекания во времени.

*Имитационное моделирование* - это совокупность методов *алгоритмизации* функционирования объектов исследований, программной реализации алгоритмических описаний, организации, планирования и выполнения на ЭВМ вычислительных экспериментов с математическими моделями, имитирующими функционирование РПС в течение заданного периода.

Под алгоритмизацией функционирования РПС понимается пооперационное описание работы всех ее функциональных подсистем отдельных модулей с уровнем детализации, соответствующем комплексу требований к модели.

"*Имитационное моделирование*" (ИМ)- это двойной термин. "Имитация" и "*моделирование*" - это синонимы. Фактически все области науки и техники являются моделями реальных процессов. Чтобы отличить математические модели друг от друга, исследователи стали давать им дополнительные названия. Термин "*имитационное моделирование*" означает, что мы имеем дело с такими математическими моделями, с помощью которых нельзя заранее вычислить или предсказать поведение системы, а для предсказания поведения системы необходим *вычислительный эксперимент* (имитация) на математической модели при заданных исходных данных.

Основное достоинство ИМ:

1. возможность описания поведения компонент (элементов) процессов или систем на высоком уровне детализации;
2. отсутствие ограничений между параметрами ИМ и состоянием внешней среды РПС;
3. возможность исследования динамики взаимодействия компонент во времени и пространстве параметров системы;

Эти достоинства обеспечивают имитационному методу широкое распространение.

Рекомендуется использовать *имитационное моделирование* в следующих случаях:

1. Если не существует законченной постановки задачи исследования и идет процесс познания объекта моделирования. *Имитационная модель* служит средством изучения явления.
2. Если аналитические методы имеются, но математические процессы сложны и трудоемки, и *имитационное моделирование* дает более простой способ решения задачи.
3. Когда кроме оценки влияния параметров (переменных) процесса или системы желательно осуществить наблюдение за поведением компонент (элементов) процесса или системы (ПС) в течение определенного периода.
4. Когда *имитационное моделирование* оказывается единственным способом исследования сложной системы из-за невозможности наблюдения явлений в реальных условиях (реакции термоядерного синтеза, исследования космического пространства).
5. Когда необходимо контролировать протекание процессов или поведение систем путем замедления или ускорения явлений в ходе имитации.
6. При подготовке специалистов для новой техники, когда на *имитационных моделях* обеспечивается возможность приобретения навыков в эксплуатации новой техники.

7. Когда изучаются новые ситуации в РПС. В этом случае имитация служит для проверки новых стратегий и правил проведения натурных экспериментов.

8. Когда особое значение имеет последовательность событий в проектируемых ПС и модель используется для предсказания узких мест в функционировании РПС.

Однако ИМ наряду с достоинствами имеет и недостатки:

1. Разработка хорошей ИМ часто обходится дороже создания аналитической модели и требует больших временных затрат.

2. Может оказаться, что ИМ неточна (что бывает часто), и мы не в состоянии измерить степень этой неточности.

3. Зачастую исследователи обращаются к ИМ, не представляя тех трудностей, с которыми они встретятся и совершают при этом ряд ошибок методологического характера.

И тем не менее ИМ является одним из наиболее широко используемых методов при решении задач синтеза и анализа сложных процессов и систем.

Одним из видов *имитационного моделирования* является статистическое *имитационное моделирование*, позволяющее воспроизводить на ЭВМ функционирование сложных случайных процессов.

При исследовании сложных систем, подверженных случайным возмущениям используются вероятностные аналитические модели и вероятностные *имитационные модели*.

В вероятностных аналитических моделях влияние случайных факторов учитывается с помощью задания вероятностных характеристик случайных процессов (законы распределения вероятностей, спектральные плотности или корреляционные функции). При этом построение вероятностных аналитических моделей представляет собой сложную *вычислительную задачу*. Поэтому вероятностное аналитическое *моделирование* используют для изучения сравнительно простых систем.

Подмечено, что введение случайных возмущений в *имитационные модели* не вносит принципиальных усложнений, поэтому исследование сложных случайных процессов проводится в настоящее время, как правило, на *имитационных моделях*.

В вероятностном *имитационном моделировании* оперируют не с характеристиками случайных процессов, а с конкретными случайными числовыми значениями параметров ПС. При этом результаты, полученные при воспроизведении на *имитационной модели* рассматриваемого процесса, являются случайными реализациями. Поэтому для нахождения объективных и устойчивых характеристик процесса требуется его многократное воспроизведение, с последующей статистической обработкой полученных данных. Именно поэтому исследование сложных процессов и систем, подверженных случайным возмущениям, с помощью *имитационного моделирования* принято называть статистическим моделированием.

Статистическая модель случайного процесса - это *алгоритм*, с помощью которого имитируют работу сложной системы, подверженной случайным возмущениям; имитируют взаимодействие элементов системы, носящих вероятностный характер.

При реализации на ЭВМ статистического *имитационного моделирования* возникает задача получения на ЭВМ случайных числовых последовательностей с заданными вероятностными характеристиками. Численный метод, решающий задачу генерирования последовательности случайных чисел с заданными законами распределения, получил название "*метод статистических испытаний*" или "*метод Монте-Карло*".

Так как *метод Монте-Карло* кроме статистического моделирования имеет *приложение* к ряду численных методов (взятие интегралов, решение уравнений), то целесообразно иметь различные термины.

Итак, статистическое *моделирование* - это способ изучения сложных процессов и систем, подверженных случайным возмущениям, с помощью *имитационных моделей*.

*Метод Монте-Карло* - это численный метод, моделирующий на ЭВМ псевдослучайные числовые последовательности с заданными вероятностными характеристиками.

Методика статистического моделирования состоит из следующих этапов:

1. Моделирование на ЭВМ *псевдослучайных последовательностей* с заданной корреляцией и законом распределения вероятностей (*метод Монте-Карло*), имитирующих на ЭВМ случайные значения параметров при каждом испытании;
2. Преобразование полученных числовых последовательностей на *имитационных математических моделях*.
3. Статистическая обработка результатов моделирования.

Обобщенный *алгоритм* метода статистических испытаний представлен на [рис. 1](#).



Рисунок 1. Обобщенный алгоритм метода статистических испытаний

## 2.17 Практическая работа №17 (2 часа).

**Тема:** «Среда и функциональная структура языка моделирования GPSS»

**2.17.1 Цель работы:** Изучить возможность оценки эффекта конструкторских решений в чрезвычайно сложных системах реального мира.

**2.17.2 Задачи работы:** Рассмотреть общецелевые системы моделирования сложных систем.

### 2.17.3 Описание (ход) работы:

GPSS (GeneralPurposeSimulationSystem) – общецелевая система моделирования сложных систем, разработанная Джеффри Гордоном. Первоначально разрабатывалась и поддерживалась компанией IBM. В настоящее время имеются версии различных разработчиков. GPSS World – самая современная версия GPSS для персональных ЭВМ и ОС Windows. Разработана компанией MinutemanSoftware. В России и странах СНГ распространяется ООО «Элина-Компьютер» (Казань).

Подробная информация о GPSS World и условиях его распространения может быть получена по адресу: <http://www.elina-computer.ru>.

Информация о системе на русском языке:

<http://www.gpss.ru> – Портал GPSS.RU, посвященный имитационному моделированию с использованием GPSS и не только,

<http://gpss-forum.narod.ru> – Сайт Юрия Носкова о GPSS.

GPSS World – общецелевая система имитационного моделирования



Система GPSS World, разработанная компанией MinutemanSoftware (США), – это мощная среда компьютерного моделирования общего назначения, разработанная для профессионалов в области моделирования. Это комплексный моделирующий инструмент, охватывающий области как дискретного, так и непрерывного компьютерного моделирования, обладающий высочайшим уровнем интерактивности и визуального представления информации.

Использование GPSS World дает возможность оценить эффект конструкторских решений в чрезвычайно сложных системах реального мира.

GPSS World основан на оригинальном языке компьютерного моделирования GPSS, что означает GeneralPurposeSimulationSystem – общецелевая система моделирования. В основном этот язык был разработан Джеффри Гордоном приблизительно в 1960 году в IBM и привнес множество важных концепций в каждую из коммерческих реализаций языков компьютерного моделирования дискретных событий, разработанных с тех пор. GPSS World – это прямое развитие языка моделирования GPSS/PC, одной из первых реализаций GPSS для персональных компьютеров. После своего появления в 1984 году GPSS/PC и его последующие версии сохранили тысячам пользователей миллионы долларов. В настоящее время версия GPSS World для ОС Windows имеет расширенные возможности, включая пользовательскую среду с интегрированными функциями работы с Интернет.

GPSS World разработан для оперативного получения достоверных результатов с наименьшими усилиями. В соответствии с этими целями в GPSS World хорошо проработана визуализация процесса моделирования, а также встроены элементы статистической обработки данных. Сильная сторона GPSS World – это его прозрачность для пользователя.

Прозрачность для пользователя ценна по трем причинам. Во-первых, опасно полагаться на непрозрачное моделирование типа “черный ящик”, внутренние механизмы функционирования которого скрыты от пользователя. Мало того, что в этом случае нельзя быть уверенным, подходит ли оно для какого-либо конкретного случая, но и невозможно гарантировать, что оно работает, как задумано. Во-вторых, удачные имитационные модели являются очень ценными и пригодны в течение длительного периода времени. Возможно, потребуется, чтобы новые сотрудники ознакомились с внутренними процессами модели, а это почти невозможная задача, если модель не имеет высокого уровня прозрачности. В-третьих, одним из наиболее эффективных, но наименее известных преимуществ компьютерного имитационного моделирования является возможность проникновения в самую суть поведения системы, когда опытный профессионал в области моделирования может видеть внутреннюю динамику в наиболее важные моменты времени процесса моделирования.

GPSS World был разработан с целью решить все эти проблемы. GPSS World является объектно-ориентированным языком. Его возможности визуального представления информации позволяют наблюдать и фиксировать внутренние механизмы функционирования моделей. Его интерактивность позволяет одновременно исследовать и управлять процессами моделирования. С помощью встроенных средств анализа данных можно легко вычислить доверительные интервалы и провести дисперсионный анализ. Кроме того, теперь есть возможность автоматически создавать и выполнять сложные отсеивающие и оптимизирующие эксперименты.

GPSS World был разработан, чтобы полностью использовать возможности вашей вычислительной системы. Использование механизма виртуальной памяти позволяет моделям реально достигать размера миллиарда байт. Вытесняющая многозадачность и многопоточность обеспечивают высокую скорость реакции на управляющие воздействия и дают возможность GPSS World одновременно выполнять множество задач. Это также означает, что система моделирования GPSS World может использовать вычислительные возможности, предоставляемые симметричными многопроцессорными архитектурами (SMP).

GPSS World сочетает в себе функции дискретного и непрерывного моделирования. Возможность перехода из дискретной фазы моделирования в непрерывную фазу и обратно обеспечивает тесную связь с непрерывным моделированием. В непрерывной фазе могут быть установлены пороговые значения, управляющие созданием транзактов в дискретной фазе.

Последняя версия GPSS World 4.3.2. (от 8 ноября 2001 года) включает в себя массу нововведений, позволяющих проводить более эффективные исследования и сделать работу с системой максимально простой и удобной для пользователя.

#### Транслятор

Высокоэффективный транслятор является частью программы GPSS World, которая создает объекты “Процесс моделирования”. Перед включением в объект “Процесс моделирования” все операторы модели проходят трансляцию. Точно так же интерактивные операторы транслируются в глобальной области видимости прежде, чем они будут переданы существующему объекту “Процесс моделирования”.

#### Архитектура

На уровне интерфейса GPSS World представляет собой реализацию архитектуры “документ-вид”, общей для всех приложений операционной системы Windows. Объекты могут быть открыты в нескольких окнах, изменены и сохранены на постоянных носителях информации. Привычное меню главного окна и блокировка недоступных команд меню, не отвлекая внимания, направляет пользователя к конечной цели. GPSS World был разработан с целью достичь тесной интерактивности даже в многозадачной среде с использованием виртуальной памяти.

#### Многозадачность

Многопоточная архитектура GPSS World позволяет совместно запускать несколько процессов моделирования и экспериментов. Одновременно выполняются не только обновление окон, пользовательский ввод, дисковый ввод-вывод, печать и процесс моделирования, но также в одно и то же время может быть запущено любое количество процессов моделирования.

#### Виртуальная память

Процессы моделирования непосредственно не ограничиваются размером физической памяти с произвольным доступом (ОЗУ), в которой выполняется объект “Процесс моделирования”. Используя механизм виртуальной памяти, модели могут достигать размера до гигабайта. Количество объектов также ограничивается только обеспечиваемым размером файла подкачки. Для достижения оптимальной производительности необходимо использовать значительный объем реальной памяти. Выделение и управление памятью для объектов происходит невидимо для пользователя. Объекты автоматически создаются до тех пор, пока не потребуется дополнительная информация.

#### Интерактивность

GPSS World поддерживает высокий уровень интерактивности даже во время процесса моделирования. Используя команду главного меню окна модели Command (Команда), ускоряющие клавиши или настройки модели, закрепляя за функциональными клавишами собственные команды, вы можете передавать существующему объекту “Процесс моделирования” любой оператор. Вы можете использовать диалоговое окно “SimulationCommand” (“Команда”) для ввода операторов, отсутствующих в выпадающем меню, а с помощью команды INCLUDE вы можете посылать процессу моделирования интерактивные операторы любой сложности.

#### Визуализация

GPSS World отличается высоким уровнем визуализации выполняющегося процесса моделирования. Для наблюдения и взаимодействия с процессом моделирования используются двадцать различных окон, соответствующих большей части объектов GPSS.

Для получения, сохранения и печати визуального представления состояния процесса моделирования не требуется дополнительных усилий, кроме операций с окнами.

#### Анимация

В GPSS World существует ряд анимационных возможностей. Уровень их реализма изменяется от абстрактной визуализации, не требующей никаких усилий, до высоко реалистических динамических изображений, включающих в себя сложные элементы, созданные пользователем.

#### Абстрактная анимация

В GPSS World встроена функция комплексной стилизованной анимации. Эта функция обеспечивается большим количеством окон, которые динамически отображают объекты GPSS в процессе моделирования по мере изменения их состояния. Для создания такой анимации не требуется дополнительных усилий. Графические изображения могут быть сохранены для включения в отчеты и/или распечатаны.

#### Пост-процессорная анимация

GPSS World имеет внешний интерфейс, который может поддерживать анимационные пакеты, управляемые трассировочными данными. Таким способом могут быть разработаны фотореалистичные анимации. Для использования этого интерфейса необходимо создать выходной поток, содержание и форматирование данных которого удовлетворяет правилам анимационного пакета, поставляемого сторонним разработчиком.

#### Оперативная анимация

GPSS World имеет набор PLUS-процедур динамического вызова, позволяющих вызывать функции во внешних исполняемых файлах. Это обеспечивает оперативную связь с анимационными пакетами других разработчиков.

#### Язык моделирования

GPSS World является реализацией GPSS, общецелевой системы моделирования, улучшенной встроенным языком программирования PLUS – языком программирования низкого уровня моделирования.

Эта версия GPSS включает в себя 53 типа блоков и 25 команд, а также более чем 35 системных числовых атрибутов, которые обеспечивают текущие переменные состояния, доступные в любом месте модели.

PLUS – это небольшой, но эффективный процедурный язык программирования, созданный из 12 типов операторов. Его эффективность во многом обеспечивается большой библиотекой процедур, содержащей математические функции и функции манипуляции со строками, и большого набора вероятностных распределений.

В GPSS World модель определяется как последовательность операторов. Это операторы GPSS, операторы PLUS-процедур или операторы PLUS-экспериментов. За исключением списков данных функции, все операторы GPSS должны состоять из одной текстовой строки длиной до 250 символов. Любой оператор GPSS может входить в модель и сохраняться в файле модели или может быть передан процессу моделирования в интерактивном режиме.

#### Нововведения

В языке моделирования было сделано множество изменений. Было исключено понятие управляющих операторов. Теперь все операторы, не являющиеся операторами блоков GPSS, называются просто командами. Номера строк игнорируются в GPSS World. Если же они используются, то должны начинаться в первом столбце текстовой строки.

Также в GPSS World не применяется автоматическое округление. Теперь необходимо явно использовать процедуру Int() или какой-либо другой метод, если требуется округлять промежуточные числовые результаты. Это справедливо для всех выражений, а также для операторов VARIABLE и BARIABLE. Теперь значения, возвращаемые системными числовыми атрибутами (СЧА), в зависимости от конкретного СЧА могут быть целого, вещественного или строчного типа. Даже СЧА, возвращающие

значения в тысячных долях, выполняют это с помощью вещественных чисел двойной точности, лежащих в интервале от 0 до 1000 включительно. “Старые” округления и целочисленные СЧА могут быть использованы в коммерческой версии GPSS World в режиме совместимости с GPSS/PC, который рассматривается далее.

PLUS-операторы могут занимать любое количество текстовых строк. Операторы GPSS, кроме списков данных функции, должны находиться на одной текстовой строке, максимальная длина которой теперь составляет 250 символов.

Математические выражения, заключенные в скобки, теперь могут использоваться почти везде, где используются СЧА. Выражения являются неотъемлемой частью языка PLUS, но они также используются для расширения возможностей операндов в операторах GPSS. Если выражения используются в операндах операторов GPSS, они должны быть заключены в скобки. Также были добавлены новые типы операторов.

К студенческой и коммерческой версии GPSS World были добавлены новые возможности:

- Автоматический генератор отсеивающего эксперимента;
- Автоматический генератор оптимизирующего эксперимента;
- Теперь в студенческой версии доступны кадры состояния;
- Библиотечная процедура дисперсионного анализа (ANOVA);
- Переменное количество аргументов в процедуре PolyCatenate();
- Библиотечные процедуры динамического вызова внешних функций;
- Возможность трассировки PLUS;
- Пакетный режим работы;
- Команда EXIT с различными режимами сохранения файлов;
- Диалоговые окна создания блоков;
- Настраиваемые интервалы табуляции;
- PLUS-процедуры для операций с потоками данных.

Новые операторы GPSS

Новые блоки

Были добавлены следующие новые блоки:

ADOPT – Изменяет номер семейства.

DISPLACE – Меняет транзакту следующий по порядку блок в последовательности блоков.

PLUS – Вычисляет значение PLUS-выражения и сохраняет результат в параметре.

INTEGRATION – Включает/выключает интегрирование переменной пользователя.

OPEN – Инициализирует поток данных.

CLOSE – Закрывает поток данных.

READ – Считывает следующую строку данных из потока данных.

WRITE – Передает значение величины в поток данных.

SEEK – Изменяет указатель строки в потоке данных.

Новые команды

Были добавлены операторы, обеспечивающие управление интегрированием непрерывных переменных, применение составных файлов модели и библиотек процедур, а также вызов PLUS-экспериментов.

CONDUCT – Выполняет зарегистрированный PLUS-эксперимент.

EXIT – Завершает сеанс работы с GPSS World, при этом может сохранять объекты.

INTEGRATE – Автоматически вычисляет интеграл для переменной пользователя.

INCLUDE – Считывает и транслирует дополнительные файлы модели.

Новые СЧА

Новый системный числовой атрибут A1 возвращает семейство активного транзакта. Его удобно использовать совместно с новым блоком ADOPT, который позволяет изменять семейство активного транзакта.

Полиморфные типы данных

Переменные могут принимать значения одного из четырех типов. Переменные, управляемые пользователем, такие, как ячейки, элементы матриц, параметры транзактов и переменные пользователя, могут принимать целочисленное, вещественное, строковое и неопределенное значение. Значения времени могут быть целыми или вещественными двойной точности.

Преобразования типов происходит автоматически. Процедуры, которые требуют в качестве аргумента значение определенного типа данных, принудительно преобразуют аргумент к соответствующему типу. Например, если вы передаете строку процедуре, которая требует числовое значение, будет использован числовой эквивалент этой строки. Точно так же, если вы пытаетесь записать (WRITE) числовое значение, оно будет автоматически преобразовано в текстовую строку.

Строковые величины имеют много вариантов использования. Они могут использоваться в потоках данных для создания отчетов и файлов с результатами, а также для прямого доступа к внутренним данным. Строковые константы обозначаются заключением строки между парой двойных кавычек. Для представления двойных кавычек внутри строки используется пара двойных кавычек. Например, для представления одной строки внутри другой строки необходимо использовать 6 символов кавычек. Внутренняя строка (подстрока) заключается между двумя парами двойных кавычек, а итоговая строка будет заключаться между еще одной парой двойных кавычек. Библиотека процедур содержит много строковых процедур, которые могут использоваться для создания и манипулирования строковыми данными.

Теперь ячейкам и матрицам может быть присвоено значение UNSPECIFIED (неопределенный тип данных). Команда INITIAL позволяет присваивать это значение ячейкам, элементам матрицы и даже целым матрицам. Если элемент данных UNSPECIFIED был использован в операции, которая требует некоторого значения, произойдет останов по ошибке. Библиотечная процедура дисперсионного анализа (ANOVA) теперь распознает элементы UNSPECIFIED в матрице результатов как недоступные результаты прогонов эксперимента.

#### Многомерные матрицы

Матрицы могут иметь до 6 измерений. Динамические окна “Matrix” (“Матрица”) позволяют просматривать любые поперечные сечения матрицы, одновременно можно открыть любое количество таких окон. Для использования во время исполнения процедуры могут быть созданы временные матрицы. Для инициализации матрицы с тремя и более измерениями используются присваивающие PLUS-операторы.

#### Выражения

GPSS World поддерживает широкое использование выражений. Они могут использоваться в PLUS-процедурах или в операторах GPSS (если заключены в скобки). Это означает, что в операндах блоков и команд могут производиться эффективные вычисления. Выражения могут производить простые вычисления, вызывать процедуры, выполняющие математические или строковые операции, производить выбор вероятностного распределения или выполнять заданные пользователем алгоритмы, включая файловый ввод-вывод.

#### Совместимость

GPSS World совместим с GPSS/PC и обычно выдаёт результаты, которые статистически неотличимы от результатов, выдаваемых GPSS/PC. Этот уровень совместимости может быть достигнут исправлением некоторых отличий и запуском процесса моделирования.

Кроме того, доступен ещё более высокий уровень совместимости, называемый режимом совместимости с GPSS/PC. В большинстве случаев можно достигнуть точного повторения результатов. Тем не менее, GPSS World использует новую исполняемую библиотеку. Применяемый в нём метод округления чисел с плавающей запятой немного отличается от используемого в GPSS/PC. Но даже в этом случае большинство моделей

GPSS/PC с небольшими изменениями могут давать идентичные результаты при выполнении под управлением коммерческой версии GPSS World в режиме совместимости с GPSS/PC. Последовательность действий, которой вы должны придерживаться, подробно рассматривается далее, после описания отличий GPSS World от GPSS/PC.

#### Отличия от GPSS/PC

GPSS World имеет много отличий от GPSS/PC. GPSS World основывается на идее, что текстовый объект “Модель” формируется и/или модифицируется, а затем транслируется с целью создания объекта “Процесс моделирования”. Этим он отличается от GPSS/PC, в котором используются понятия файла программы и процесса моделирования.

В GPSS World была полностью заменена операция загрузки модели, используемая в GPSS/PC. Вместо последовательного ввода и просмотра каждой строки, одну за другой, в GPSS World применяется полноэкранный текстовый редактор и команда меню CreateSimulation (Создать процесс моделирования). Любые ошибки, обнаруженные во время трансляции, сохраняются в очереди сообщений об ошибках, поэтому они могут быть легко найдены и исправлены. Быстродействие транслятора гарантирует, что обнаружение и исправление ошибок происходит очень быстро.

GPSS World не выполняет операцию предотвращения ошибки при каждом нажатии клавиши, как это было в GPSS/PC. Вместо этого для создания процесса моделирования GPSS World использует транслятор модели. Это ускоряет время загрузки модели в сотни раз. Однако это привело к изменению механизма обнаружения ошибок. Теперь ошибки обнаруживаются в течение трансляции и могут быть исправлены с помощью команды NextError (Следующая ошибка) из меню Edit (Правка) главного окна. Курсор автоматически перемещается к ошибке, а в строке состояния внизу главного окна появляется сообщение об этой ошибке.

Возможно, наиболее заметным изменением по сравнению с GPSS/PC является отсутствие необходимости в нумерации строк, которая фактически игнорируется в GPSS World. Это означает, что расположение блоков больше не определяется номером строки оператора блока, а определяется только относительным положением этой строки в файле (файлах) модели, переданном транслятору. Т.к. новый оператор INCLUDE позволяет объектам “Модель” содержать в себе другие файлы текстового формата, объекты “Модель” представляют собой считываемую транслятором последовательность операторов, которая определяет расположение блоков в процессе моделирования. Несмотря на то, что блоки больше не могут быть вставлены во время процесса моделирования, для большинства целей будет проще повторно транслировать модель, если необходимо изменить структуру блоков, т.к. трансляция выполняется очень быстро. При использовании режима ручного моделирования сохраняется высокий уровень интерактивности. Это означает, что любой оператор может использоваться в течение процесса моделирования в качестве интерактивного оператора.

В GPSS World был сохранён высокий уровень интерактивности. Любой оператор модели может быть передан для выполнения выполняющемуся процессу моделирования. PLUS-процедуры могут быть определены или даже переопределены “на лету”. Команды GPSS могут использоваться для переопределения объекта или управления процессом моделирования. Блоки могут использоваться, так же, как и в GPSS/PC, для создания временного блока, в который направляется активный транзакт. Все это выполняется после трансляции модели с помощью меню Command (Команда).

В GPSS World не используется автоматическое округление. Теперь для округления промежуточных числовых результатов необходимо явно использовать процедуру Int() или какой-либо другой метод. Это касается и всех выражений, а также операторов VARIABLE и BVARIABLE.

Теперь значения, возвращаемые системными числовыми атрибутами (СЧА), в зависимости от конкретного СЧА могут быть целого, вещественного или строчного типа.

Даже СЧА, возвращающие значения в тысячных долях, выполняют это с помощью вещественных чисел двойной точности, лежащих в интервале от 0 до 1000 включительно. “Старые” округления и целочисленные СЧА могут быть использованы в коммерческой версии GPSS World в режиме совместимости с GPSS/PC.

Использование в GPSS World многозадачной архитектуры также привело к некоторым изменениям. Прежде всего, теперь для представления команд или изменения состояния посылаются служебные сообщения. Таким образом, оперативное обновление окна выполняется посредством очереди сообщений, поступающих от процесса моделирования. Точно так же большинство команд, полученных объектом “Процесс моделирования”, перед выполнением помещаются в очередь команд процесса моделирования. Исключения составляют только команды SHOW и HALT, которые выполняются немедленно. Кроме того, команда HALT удаляет из очереди все оставшиеся команды.

Остальные изменения перечислены далее и более подробно рассматриваются в других частях данного руководства пользователя:

PLUS-операторы могут включать в себя любое количество текстовых строк. Хотя операторы GPSS, кроме списков данных функции, должны находиться на одной текстовой строке, максимальная длина которой теперь составляет 250 символов.

Математические выражения, заключённые в скобки, могут теперь использоваться практически везде, где используются СЧА.

Системные числовые атрибуты в зависимости от конкретного типа теперь могут возвращать целочисленные, вещественные или строковые значения. В режиме совместимости с GPSS/PC СЧА, за исключением модификаторов функций, возвращают только целочисленные значения.

Введен новый системный числовой атрибут, A1, который возвращает номер семейства активного транзакта.

Блок HELP больше не используется. Его заменили блоком PLUS, который поддерживает вложение в модель законченных процедур, а также блоками потоков данных, которые поддерживают связь с внешними файлами и программами.

Управляющий оператор MICROWINDOW был заменен окном “Expression” (“Выражения”), которое может быть открыто с помощью подменю Window (Окно).

Управляющий оператор PLOT был заменен окном “Plot” (“График”), которое может быть открыто с помощью подменю Window (Окно).

Управляющий оператор END был заменен командой EXIT, которая может завершать сеанс работы с GPSS World. END теперь является ключевым словом языка PLUS.

Управляющий оператор ANOVA был заменен библиотечной процедурой ANOVA.

Управляющий оператор EVENTS был заменен окнами “CEC Snapshot” (“Кадр CTC”) и “FEC Snapshot” (“Кадр СБС”), которые могут быть открыты с помощью подменю Window (Окно).

Управляющий оператор GROUPS был заменен окнами “NumericGroupsSnapshot” (“Кадр числовых групп”) и “TransactionGroupsSnapshot” (“Кадр групп транзактов”), которые могут быть открыты с помощью подменю Window (Окно).

Управляющий оператор RESULT был заменен блоками управления потоками ввода-вывода, которые могут автоматически записывать данные в файл результатов.

Управляющий оператор WINDOW был заменен специальным подменю Window (Окно) в меню главного окна.

Управляющий оператор USERCHAINS был заменен окном “UserchainsSnapshot” (“Кадр списков пользователя”), которое может быть открыто с помощью подменю Window (Окно).

Управляющий оператор ANITRACE больше не используется. Анимация в GPSS World обеспечивается потоками данных и постпроцессорами сторонних разработчиков.

Блок MOVE больше не используется. Анимация в GPSS World обеспечивается потоками данных и постпроцессорами сторонних разработчиков.

Окно “Positions” и файл POSITION.GPS больше не используются. Анимация в GPSS World обеспечивается потоками данных и анимационными пакетами сторонних разработчиков.

Символ “@”, обозначавший вложение файлов, был заменен оператором INCLUDE, который использует в качестве операнда полный путь доступа к файлу, заключенный в двойные кавычки.

Системный числовой атрибут Z1 больше не отображает общий объем физической памяти. Теперь он возвращает значение максимального объема памяти, который может быть выделен, возвращаемое операционной системой.

Следующие функции управления исходным текстом, присутствовавшие в GPSS/PC, были заменены графическим интерфейсом пользователя, полноэкранным редактором и соответствующими опциями: DELETE, DISPLAY, DOS, EDIT, RENUMBER и SAVE.

Ячейками, матрицами, именованными величинами и параметрами транзактов теперь поддерживаются значения целочисленного, вещественного или строкового типа данных. Значения времени могут быть целочисленными или вещественными. Целочисленная арифметика неограниченной точности, применявшаяся в GPSS/PC, больше не используется.

Команды REPORT теперь всегда функционируют в режиме NOW. Операнд A больше не используется и должен быть нулевым.

Теперь PLUS-выражениях допускается использование процедур из библиотеки PLUS-процедур. Также поддерживаются процедуры обработки строк и более двадцати встроенных вероятностных распределений.

Матрицы GPSS теперь могут иметь до 6 измерений. Любое двумерное поперечное сечение матрицы может динамически просматриваться в окне “Matrix” (“Матрица”).

Изменились приоритеты операторов. Новые приоритеты рассматриваются в Главе 3. Если возникают какие-либо проблемы, необходимо выделить скобками все операции в выражениях из программных файлов GPSS/PC.