



# Курс «Моделирование»

Лектор – Ланцов Владимир Николаевич,  
кафедра ВТ

# Состав курса:

- Лекции – 34 ч.
- Лаборат. работы – 17 ч. (4 работы с 9-й нед.)
- Практические – 34 ч.
- К.Р. - нет
- Экзамен – есть
- Рейтинг-контроль
  - 6 неделя – по лекциям
  - 12 неделя – по практическим + лекциям
  - 17 неделя – по практическим + лаб.работам + лекциям

# Содержание лекций

- Лекция 1. Общие сведения о моделировании.
- Лекция 2. Моделирование ВС с помощью систем массового обслуживания.
- Лекция 3. Организация моделирования СМО.
- Лекция 4. Модели источников входного потока заявок.
- Лекция 5. Языки имитационного моделирования.
- Лекция 6. Планирование машинного эксперимента.
- Лекция 7. Аналитическое моделирование.
- Лекция 8. Сети Петри.
- Лекция 9. Цепи Маркова.

# Литература

- Основная:

- 1. Ланцов В.Н. Моделирование. Часть 1. /уч.пособие. – ВлГУ, 1999
- 2. Советов Б.А., Яковлев С.А. Моделирование систем. – М.: Высш.шк., 1985
- 3. Ланцов В.Н. Моделирование: метод. указания к лаб. работам/ ВлГУ, 1998

- Дополнительная:

- 4. Ланцов В.Н. Моделирование. Часть 2. /уч.пособие. ВлГУ, 2001
- 5. Шрайбер Т.Д. Моделирование на GPSS. – М.: Машиностроение, 1980
- 6. Автоматизация проектирования вычислительных систем. Языки моделирования и базы данных /Под ред. М.Брейера. – М.: Мир, 1979
- 7. Советов Б.Я., Яковлев С.А. Моделирование систем: Лабораторный практикум.-М.: Высш.шк., 1989

# Лекция 1. Общие сведения о моделировании

## ● **Содержание:**

- **Введение**
- **Основные определения теории моделирования**
- **Классификация математических моделей**
- **Методика получения математических моделей элементов**
- **Обобщенная схема моделирования**



# Лекция 1. Общие сведения о моделировании

- Введение:

- Необходимость внедрения средств ВТ в различные отрасли н/х
- Требования к уровню подготовки специалистов повышаются, необходимы знания по методикам анализа и моделирования на базе современных программных систем
- Полное и всестороннее исследование ВС возможно только с помощью средств моделирования
- «Моделирование» базовая дисциплина специальности, отдельный раздел дипломных и курсовых проектов

# Лекция 1. Общие сведения о моделировании

- **Основные определения теории моделирования:**
  - **Объект моделирования** – все, на что направлена деятельность человека
  - **Гипотеза** – предсказание, основанное на небольшом количестве опытных данных
  - **Аналогия** – суждение о частном сходстве двух объектов. Часто гипотеза создается по аналогии.
  - Наглядная и удобная форма для сравнения и представления гипотез и аналогий – объекты-заменители для объектов-оригиналов (**модели**)
  - **Моделирование** – это замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-заменителя (**модели**)
  - **Теория моделирования** – теория замещения одних объектов другими объектами (**моделями**) и исследование свойств объектов на их моделях.

# Лекция 1. Общие сведения о моделировании (продолж.)

## ● Основные определения теории моделирования (продолж.):

- Если результаты моделирования подтверждаются (свойства совпадают), могут служить основой для прогнозирования процессов, то говорят, что модель адекватна объекту.

*Адекватность* сильно зависит от цели моделирования и принятых критериев.

- Более обобщенно - моделирование это метод опосредованного познания объекта-оригинала.

Различают:

- моделирование как познавательный процесс обработки информации из внешней среды;

- моделирование как построение системы-модели (второй системы), связанной определенными соотношениями подобия с системой-оригиналом (первой системой).



# Лекция 1. Общие сведения о моделировании (продолж.)

- **Основные определения теории моделирования (продолж.):**
  - ВС и комплексы относятся к одним из самых сложных систем. Проектирование и исследование их невозможно без использования различных видов моделирования.
  - На всех уровнях представления ВС учет особенностей: сложность структуры, неоднозначность связей между элементами, неоднозначность алгоритмов поведения при различных условиях, большое количество параметров и переменных, неполнота и недетерминированность исходной информации, разнообразие и вероятностный характер воздействий, стохастичность связей и т.д.
  - На разных этапах применение метода моделирования преследует конкретные цели, а эффективность метода зависит от того, насколько грамотно разработчик использует возможности моделирования.

# Лекция 1. Общие сведения о моделировании (продолж.)

## ● Основные определения теории моделирования (продолж.):

- Выделяют два подхода:

аналитический и численный.

Аналитический - основан на использовании формальных методов доказательств и решения в аналитическом виде, где модель представляется аналитическими зависимостями, а решение находится в явном виде.

Находит все меньшее применение, значительные трудности при моделировании сложных систем, упрощение моделей - к получению недостоверных результатов.

Численный подход основан на математическом моделировании процессов функционирования объектов. Математическое моделирование – это исследование объекта путем создания его математической модели и оперирования этой моделью с целью получения полезной информации о свойствах объекта.

Математическая модель – это совокупность математических объектов (чисел, скалярных переменных, векторов, графов и т.п.) и связывающих их отношений.

Альтернативой математического моделирования является физическое макетирование. У математического моделирования ряд преимуществ: меньшие сроки на подготовку моделирования; значительно меньшая материалоемкость, особенно при моделировании крупногабаритных объектов; возможность выполнения экспериментов на критических режимах, которые привели бы к разрушению физического макета и др.

# Лекция 1. Общие сведения о моделировании (продолж.)

- Основные определения теории моделирования (продолж.):

-В зависимости от характера изучаемых процессов все виды моделирования могут быть разделены на:

Детерминированное моделирование отображает детерминированные процессы, т.е. отсутствие всяких случайных воздействий;

Стохастическое моделирование отображает вероятностные процессы и события, оцениваются средние характеристики.

Статическое моделирование служит для описания поведения объекта в какой-либо момент времени.

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 в случаях, когда хотят выделить наличие как дискретных, так и непрерывных процессов.

# Лекция 1. Общие сведения о моделировании (продолж.)

- **Классификация математических моделей:**

Представим модель в виде

$$Y(t) = F(X, Q, t). \quad (1)$$

Где  $X$  - совокупность (в общем случае вектор) входных воздействий и воздействий внешней среды;  $Q$  - совокупность внутренних (собственных) параметров системы;  $Y$  - совокупность выходных характеристик системы. Входные воздействия, воздействия внешней среды и внутренние параметры - являются независимыми переменными, а выходные характеристики являются зависимыми переменными. Процесс функционирования системы описывается оператором  $F$ , который преобразует независимые переменные в зависимые

- Соотношение (1) является математическим описанием поведения объекта моделирования во времени  $t$ , т.е. отражает его динамические свойства. Поэтому модели такого вида назыв. *динамическими моделями*. Для *статических моделей* соотношение (1) преобразуется в
- $Y = F(X, Q)$ .
- Если математическое описание не содержит элементов случайности или они не учитываются, то модель называется *детерминированной*
- $Y(t) = F(X, t)$ .
- Детерминированная модель является частным случаем стохастической модели.

# Лекция 1. Общие сведения о моделировании (продолж.)

- Классификация математических моделей (продолж.):

Признак классификации	Тип математической модели
1. Характер отображаемых свойств объекта	Структурные, функциональные
2. Принадлежность к иерархическому уровню	Микроуровень, макроуровень, метауровень
3. Степень детализации описания внутри одного уровня	Полные, макромодели
4. Способ представления свойств объекта	Аналитические, алгоритмические, имитационные
5. Способ получения модели	Теоретические, эмпирические



# Лекция 1. Общие сведения о моделировании (продолж.)

- **Классификация математических моделей (продолж.):**

- Структурная ММ отображает геометрические и топологические (структурные) свойства (формы и взаимное расположение, состав и взаимосвязи). Представляется в виде уравнений поверхностей и линий, графов, таблиц, списков, неравенств и т.д.

- Функциональная ММ отражает физические и информационные состояния и процессы, последовательно сменяющих друг друга состояний в объекте. Типичная форма представления - системы уравнений.

- При использовании блочно-иерархического подхода приводит к появлению деления моделей на уровни описания. Количество иерархических уровней определяется сложностью объектов и возможностями систем моделирования. Однако для большинства предметных областей в технике можно выделить три обобщенных уровней:

- Микроуровень
  - Макроуровень
  - Метауровень,

различающихся степенью детализации рассмотрения процессов в объекте.

# Лекция 1. Общие сведения о моделировании (продолж.)

- Классификация математических моделей (продолж.):
- ММ на микроуровне отражает физические процессы, протекающих в непрерывных пространстве и времени (сплошная или трехмерная среда).
- Типичные ММ на микроуровне - системы дифференциальных уравнений в частных производных (ДУЧП) с заданными краевыми условиями. Независимыми переменными здесь являются пространственные координаты и время (уравнения Ламе для механики упругих сред, уравнения Навье-Стокса для гидравлики, уравнения теплопроводности для термодинамики и т.д.).
- Для решения этих уравнений используются методы конечных разностей и граничных элементов. Возможно единое ПО для разных классов объектов.
- Для элементов ВС модели описывают состояние в сплошной среде элементов и деталей узлов ЭВМ. Математическая модель – в виде ДУЧП (уравнения теплопроводности, диффузии, электродинамики, упругости, непрерывности, переноса, Пуассона и т.п.) с заданными краевыми условиями. В качестве переменных этих уравнений используются время  $t$  и пространственные координаты  $x_i$ . Зависимыми параметрами являются температура, концентрация основных и неосновных носителей, напряженности полей и т.д.
- ДУЧП можно представить в общем виде как
$$L \varphi(\mathbf{Z}) = f(\mathbf{Z}),$$
- где  $\mathbf{Z} = (t, x_1, x_2, x_3, \dots)$  – вектор независимых параметров;  $f(\mathbf{Z})$  – внешнее воздействие на сплошную среду;  $L$  – оператор дифференцирования;  $\varphi(\mathbf{Z})$  – уравнение, описывающее физическую природу рассматриваемого объекта.
- Если из уравнения исключить время  $t$ , то уравнение будет стационарным. Вместо трех пространственных координат можно использовать одну или две.

# Лекция 1. Общие сведения о моделировании (продолж.)

- Классификация математических моделей (продолж.):

**Пример:** уравнение теплопроводности. Это уравнение является нестационарным и одномерным

$$\partial T / \partial t - a_{t*} \partial^2 T / \partial x^2 = 0,$$

где  $T$  – температура,  $L = \partial / \partial t - a_{t*} \partial^2 T / \partial x^2$ ,  $\varphi(Z) = T(t,x)$ ,  
 $f(Z) = 0$ ,  $Z = (t,x)$ .

- Возможности ММ в виде ДУЧП ограничены отдельными компонентами (фрагментами). С увеличением размерностей решаемых задач анализ ММ на микроуровне приводит к чрезмерным вычислительным затратам, поэтому вводят какие-либо упрощения или допущения и переходят к следующему уровню описания. Осуществляется переход от непрерывного к дискретному пространству при сохранение непрерывного времени.

# Лекция 1. Общие сведения о моделировании (продолж.)

## ● Классификация математических моделей (продолж.):

- На макроуровне используют укрупненную дискретизацию пространства по функциональному признаку, что приводит к представлению ММ на этом уровне в виде систем обыкновенных дифференциальных уравнений (ОДУ) с заданными начальными условиями.
- В этих уравнениях независимой переменной является время, а зависимыми переменными являются переменные, характеризующие состояние укрупненных элементов, например силы и скорости механических систем, напряжения и силы тока электрических систем, давления и расходы гидравлических и пневматических систем и т.п.
- Системы ОДУ строятся на основе компонентных уравнений для дискретных элементов и топологических уравнений, указывающих связи между элементами. На этом уровне возможно единое математическое и программное обеспечение для разных классов технических объектов.

# Лекция 1. Общие сведения о моделировании (продолж.)

- Классификация математических моделей (продолж.):
  - Для узлов ЭВМ непрерывные модели формируются на основе компонентных, например:  
 $i(t) = U(t)/R, i(t) = C dU/dt, U(t) = L di/dt$
  - Используя топологические связи (например, для метода узловых потенциалов), можно получить систему дифференциальных уравнений:  
 $F(U', U, t) = 0, \text{ где } U' = dU/dt \text{ или } U' = f(U, t)$
  - Для дискретных моделей на макроуровне все состояния дискретны, в частном случае это булевы переменные (0, 1). Непрерывное время также заменяется на дискретные значения (такты моделирования):  $\Delta t_k = t_k - t_{k-1}$
  - Модель может быть представлена в виде конечного автомата, который описывается с помощью системы логических уравнений, например:  $V' = F(V, U)$ .



# Лекция 1. Общие сведения о моделировании (продолж.)

- Классификация математических моделей (продолж.):
  - При повышении размерностей решаемых уравнений (порядок систем ОДУ превышает  $10^3$ , оперировать моделью становится затруднительным, и поэтому обычно переходят к следующему, более высокому уровню.
  - На метауровне в качестве элементов принимают достаточно сложные совокупности компонентов. В моделях отражаются процессы преобразования информации, а не сигналов, как на макроуровне.
  - Метауровень характеризуется большим разнообразием типов используемых ММ. Применяют:
    - системы ОДУ (с переменными, относящимися к выводам компонента)
    - *функциональное моделирование*, развитое для анализа систем автоматического управления (САУ)
    - *моделями систем массового обслуживания (СМО)*
    - *сети Петри.*

# Лекция 1. Общие сведения о моделировании (продолж.)

## ● Классификация математических моделей (продолж.):

- Полная ММ – модель, в которой фигурируют переменные, характеризующие состояния всех имеющихся межэлементных связей (т.е. состояния всех элементов объекта).
- Макромодель - ММ, в которой отображаются состояния значительно меньшего числа межэлементных связей, что соответствует более укрупненному описанию объекта. Понятия "полная ММ" и "макромодель" относительны.
- Аналитические ММ представляют собой явные аналитические выражения, связывающие выходные параметры с входными и внутренними параметрами.
- Алгоритмические ММ выражают связи выходных параметров с внутренними и внешними параметрами в виде алгоритма.
- Имитационная ММ отражает поведение объекта при заданных, изменяющихся во времени внешних воздействиях.

# Лекция 1. Общие сведения о моделировании (продолж.)

- **Классификация математических моделей (продолж.):**
  - Математические модели разделяются на математические модели компонентов (элементов) и математические модели систем (строятся на основе математических моделей компонентов).
  - *Теоретические* ММ создаются в результате исследования процессов и их закономерностей, присущих рассматриваемому классу объектов и явлений.
  - *Эмпирические* ММ - в результате изучения внешних проявлений свойств объекта с помощью измерений и обработки результатов измерений.
  - Для большинства технических объектов используют типовые элементы, количество типов которых обычно невелико, а модели предназначены для многократного использования. Поэтому разработка ММ элементов выполняется сравнительно редко. Единожды созданные ММ элементов в дальнейшем многократно используются при разработке разнообразных систем на базе этих элементов. Типичные примеры - модели транзисторов и диодов, модели стандартных микросхем и т.п.
  - Математические модели систем строятся на основе формальных правил или алгоритмов и часто выполняются в автоматическом режиме по математическим моделям компонентов. Изучение этих методов - одна из целей данного курса.

# Лекция 1. Общие сведения о моделировании (продолж.)

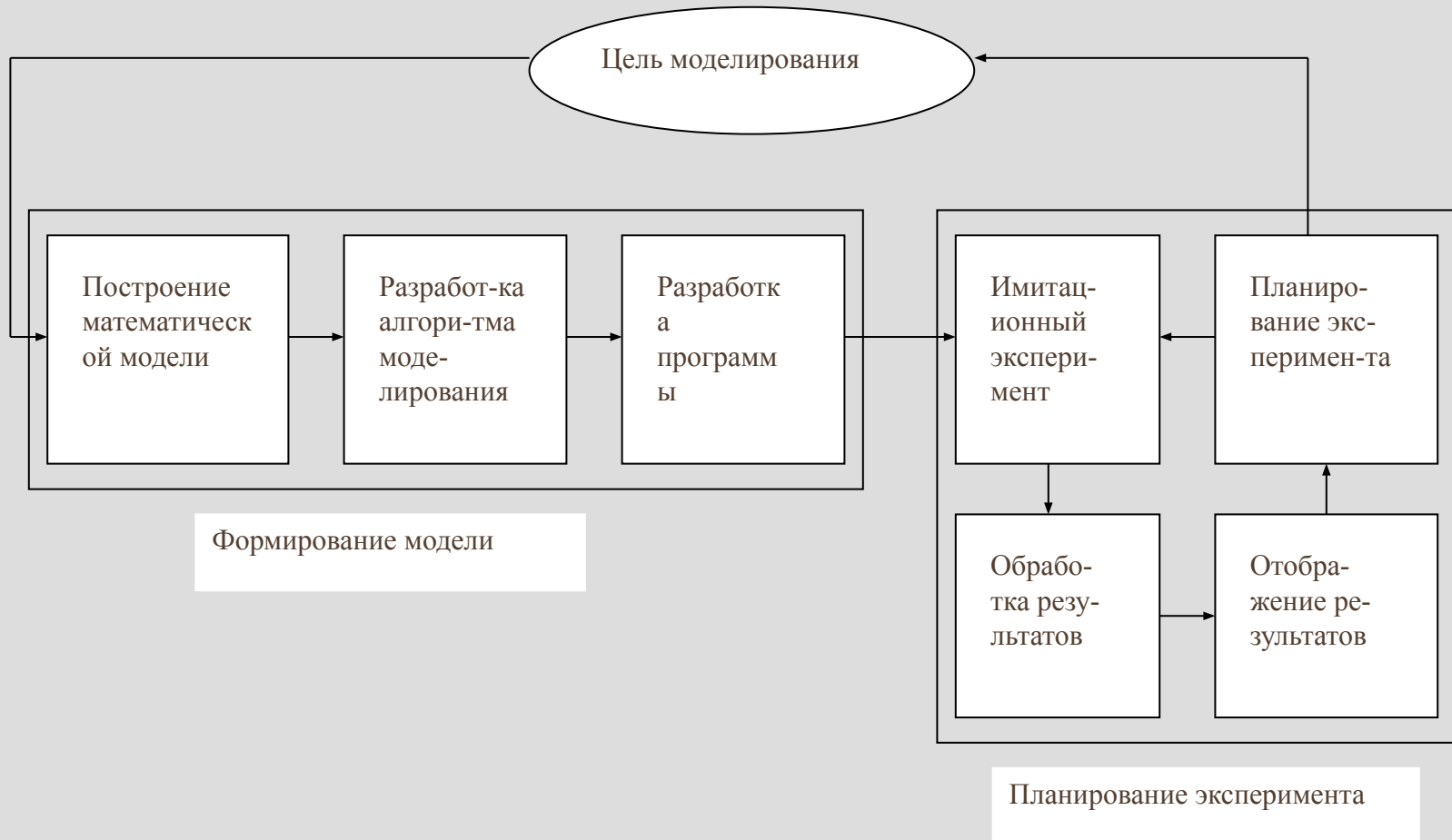
- **Методика получения математических моделей элементов:**

Процедура получения ММЭ включает в себя следующие операции:

1. Отбор свойств объекта, которые подлежат отражению в модели (анализ возможных применений модели).
2. Сбор исходной информации о выбранных свойствах объекта (опыт и знания инженера, научно-техническая справочная литература, описания прототипов, близкие по своим свойствам имеющиеся ММ, результаты экспериментальных измерений параметров и т.п.)
3. Синтез структуры ММ (общий вид математических соотношений без конкретных числовых значений параметров, эквивалентные схемы, графы и т.п.). Это наиболее ответственная операция.
4. Расчет числовых значений параметров ММ с помощью минимизации погрешности между поведением модели и результатами эксперимента либо физического, либо численного с использованием более точных ММ.
5. Оценка точности и адекватности (области адекватности) ММ. Для оценки точности используются экспериментальные значения, не применявшиеся при четвертой операции.

# Лекция 1. Общие сведения о моделировании (продолж.)

- Обобщенная схема моделирования:





# Лекция 1. Общие сведения о моделировании (продолж.)

- Обобщенная схема моделирования (продолж.):
  - На первом этапе моделирования решается ряд задач: построения модели, выбора алгоритма моделирования и программная реализация. Основным назначением первой задачи является переход от содержательного описания объекта к его математической модели.
  - Второй этап моделирования – выполнение машинного эксперимента, рабочих расчетов по составленной и отлаженной программе. Здесь решаются задачи: планирования машинного эксперимента с моделью (начальные условия, число прогонов модели), проведение рабочих расчетов (исходные данные, получение результатов), анализ результатов моделирования (отбор, статистика), представление результатов моделирования (таблицы, графики, схемы), интерпретация результатов моделирования (интерпретация по отношению к моделируемому объекту), подведение итогов моделирования и выработка рекомендаций (выводы, проверка гипотез, рекомендации по использованию).

# Лекция 2. Моделирование ВС с помощью систем массового обслуживания

- Содержание:
  - Задачи и особенности системного уровня
  - Краткие сведения о СМО
  - Схема имитационного моделирования
  - Имитационные модели ВС в СМО

# Лекция 2. Моделирование ВС с помощью систем массового обслуживания

## ● Задачи и особенности системного уровня:

- На системном уровне разрабатываются структурные схемы ЭВА, в связи с чем данный уровень называют *структурным уровнем*.
- Ведется укрупненное рассмотрение всей ЭВМ или ВС в целом. Элементы – это крупные узлы ЭВМ (процессоры, каналы, запоминающие устройства, периферийные устройства), хотя степень детализации может быть и очень глубокой.
- С алгоритмической позиции – это разработка архитектуры ЭВА (форматы представления различных данных, системы и форматы команд, организация адресации; принципы выполнения операций, приоритеты, дисциплины обслуживания; общие технические характеристики ЭВА (производительность или пропускная способность, надежность и др.).

# Лекция 2. Моделирование ВС с помощью систем массового обслуживания

## ● **Задачи и особенности системного уровня:**

### Основные задачи:

- 1) Определение принципов организации ВС.**
- 2) Выбор архитектуры и разделение функций на программную и аппаратную реализацию.**
- 3) Разработка структурной схемы ВС.**
- 4) Определение требований на выходные параметры и характеристики компонентов, формирование ТЗ на отдельные компоненты ВС.**



# Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

- Задачи и особенности системного уровня:

- Особенности системного моделирования:

- *Функционирование ВС рассматривается с информационной точки зрения (процесс преобразования информации), абстрагирование от физической сущности протекающих в системе процессов.*
  - *Характер информационных процессов в ВС в значительной мере определяется программным обеспечением (совокупность аппаратных и программных средств).*
  - *Моделирование выполняется как многовариантный анализ: варьируя характеристиками элементов и связями между ними, определяют такую структуру ВС, которая обладает заданными (необходимыми) характеристиками.*
  - *Задачи моделирования отличаются большим разнообразием и трудно поддаются формализации, трудно автоматизировать процедуру составления математических моделей ВС на основе моделей компонентов.*
  - *В ЭВМ с помощью математических моделей имитируется процесс ввода, обработки и вывода данных (метод имитационного моделирования).*
  - *Соответствие ВС своему назначению невозможно проверить на примере одной задачи, решаемой в ВС. Необходимо обобщение на весь класс решаемых задач. Такое обобщение достигается при учете статистических закономерностей поступления и обработки данных, т.е. используются вероятностные методы.*
  - *Наиболее часто ВС моделируются с позиций систем массового обслуживания (СМО).*
  - *Возможна постоянная детализация моделей ВС. Число уровней детализации может быть очень большим.*



# Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

## ● Краткие сведения о СМО:

- Системы массового обслуживания (СМО) – класс математических схем для формализации процессов функционирования систем на основе процессов 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 в ЭВМ. Характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т.е. стохастический характер.

- Элементы СМО:

СМО - это системы, предназначенные для обслуживания (обработки) потока заявок (решаемых задач) с помощью совокупности устройств (обслуживающих аппаратов - ОА).

- ОА относятся к так называемым статическим объектам или ресурсам. Такими объектами могут быть ЭВМ, отдельные устройства ЭВМ, внешние устройства и т.п. Так как обработка данных может выполняться как аппаратными, так и программными средствами, то программные средства также относят к ресурсам.

- Элементы динамического типа – это заявки, или транзакты (решаемые в ВС задачи).

## Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

### ● Краткие сведения о СМО (продолж.):

#### Элементы СМО (продолж.):

- **Функционирование СМО** – это процесс прохождения заявок через ОА.
- **ОА** может быть в состоянии «занято» (если заявка вошла в ОА на обслуживание) или «свободно». ОА также характеризуется длиной очереди заявок к нему.
- **Заявки** характеризуются состояниями «на обслуживании» (если она занимает ОА) и «в ожидании» (если она находится в очереди).
- **Дисциплина обслуживания** – это правило, по которому заявки поступают из очереди на обслуживание в ОА. Наиболее распространенные: FIFO (first input – first output), «первым пришел – первым обслужен», LIFO (last input – first output) – здесь заявки выбираются из конца очереди.

## Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

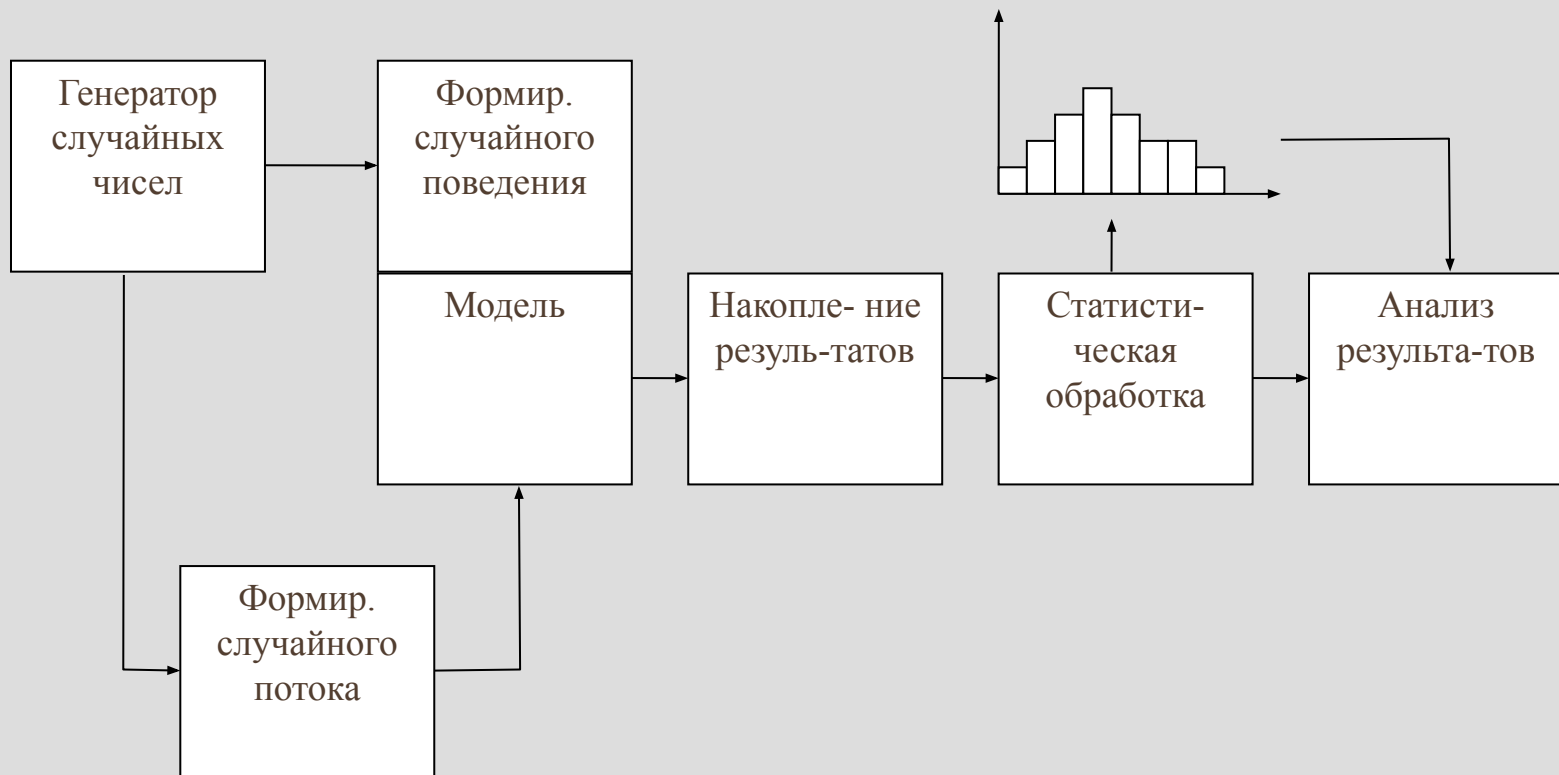
### ● Краткие сведения о СМО (продолж.):

#### Элементы СМО (продолж.):

- Приоритет – это преимущества на обслуживание одной заявки перед другими. Если все заявки имеют одинаковый приоритет, то система называется бесприоритетной.
- Приоритеты могут быть динамическими и статическими в зависимости от того, возможно или нет изменение приоритетов в процессе выполнения задачи.
- Любое изменение в состоянии системы является событием. Считается, что события происходят мгновенно в дискретные моменты времени.
- СМО могут быть одно- и многоканальные в зависимости от числа параллельно работающих каналов. Замкнутые СМО – это когда в системе циркулирует постоянное число заявок.

# Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

- **Схема имитационного моделирования:**

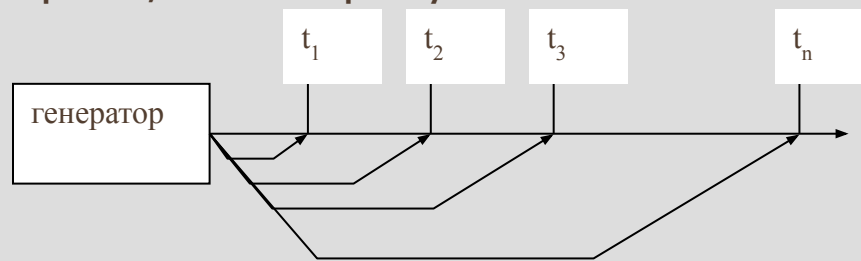


# Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

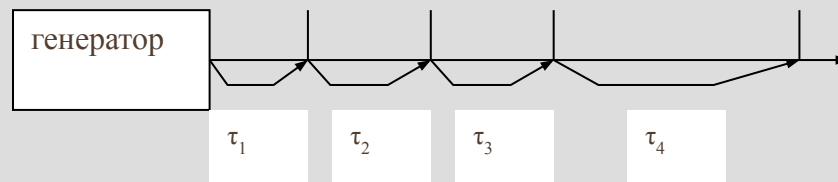
## ● Имитационные модели ВС в СМО:

– Модель любой сложности ВС можно построить из четырех типов моделей: источников заявок, устройств, памяти и узлов.

1. Модель источника входного потока заявок (генератор заявок) – это алгоритм, по которому вычисляются моменты появления заявок.



абсолютные  
времена



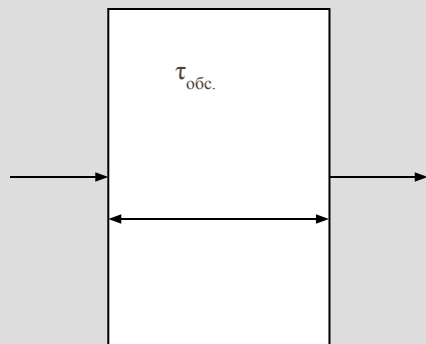
относительные  
времена



# Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

- Имитационные модели ВС в СМО:

2. Модель устройства – это алгоритм выработки значений интервалов (времени) обслуживания заявки в данном устройстве. Чаще всего это алгоритм выработки значений случайной величины, распределенной по заданному закону.

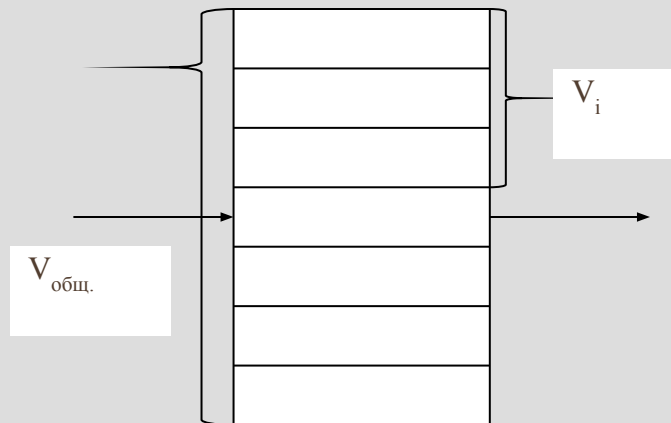


$\tau_{обс.}$  – время работы данного устройства

# Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

- Имитационные модели ВС в СМО:

3. Модель памяти – это алгоритм, определяющий объем памяти, необходимый для обслуживания заявки  $V_i$ . Обычно объем памяти определяется как реализация случайной величины, причем закон распределения и его параметры зависят от типа заявки.



## Лекция 2. Моделирование ВС с помощью систем массового обслуживания (продолж.)

- Имитационные модели ВС в СМО:
  4. Модель узла. Связи источников и ОА в значительной мере определяются функционированием программного обеспечения (ПО), а не физическими связями между различными устройствами. Именно ПО определяет маршрут прохождения заявок по ВС. Для имитации связей используются специальные элементы-модели, называемые узлами. Модели узлов бывают нескольких типов, например: для организации разветвлений в зависимости от типа заявки или определенных признаков; для организации вероятностных разветвлений; для изменения типа заявок и др.

# Лекция 3. Организация моделирования СМО

- **Содержание:**

- Организация моделирования в СМО
- Событийное моделирование
- Принципы работы имитационных моделей
- Организация программного обеспечения систем имитационного моделирования на основе СМО.

# Лекция 3. Организация моделирования СМО

- Организация моделирования в СМО:
  - Имитационная модель представляет собой алгоритм, состоящий из упорядоченных обращений к моделям элементов. Последовательность обращений определяется свойствами моделируемой ВС и режимом ее функционирования.
  - В процессе имитации в модели происходит изменение дискретного времени. Время изменяется после того, как заканчивается имитация очередной группы событий, относящихся к текущему моменту времени.
  - В процессе имитации происходит накопление данных. Процесс имитации заканчивается, когда текущее время превысит заданный отрезок времени моделирования или когда входные источники выработают заданное количество заявок.



# Лекция 3. Организация моделирования СМО

- Организация моделирования в СМО:

- Организация имитационного моделирования ВС будет заключаться в определении происходящих в системе событий и их привязки к моментам дискретного времени и элементам системы.
- Математическая модель ВС представляет собой программу, включающую ряд подпрограмм. Ряд основных подпрограмм обеспечивает генерацию и продвижение заявок по системе. Исполнение такой программы по отношению к конкретной заявке выявляет ее маршрут и времена прохождения этапов маршрута.
- Однако поведение заявки в системе не является независимым, оно обуславливается событиями, в которых фигурируют также другие заявки. Поэтому процесс имитации на ЭВМ должен отображать хронологию событий в последовательности, имеющей место в реальной ВС.

# Лекция 3. Организация моделирования СМО

- **Организация моделирования в СМО:**

- Существуют две проблемы организации моделирования:

1. Процессы в реальной системе зависят от времени, т.е. необходимо создавать аналог системного времени в модели и соотносить все процессы в зависимости от него.

2. В реальной системе информационные процессы протекают параллельно. В модели это реализовать невозможно, поэтому параллельные процессы надо имитировать как последовательные, обеспечивая при этом все причинно-следственные связи реальной системы.

- Существуют два способа реализации имитационного моделирования:

1. Потактовое моделирование.
2. Событийное моделирование.

# Лекция 3. Организация моделирования СМО

- Организация моделирования в СМО:

- **Потактовое моделирование.**

- Расчет всей системы выполняется через некоторый достаточно малый интервал времени (такт). Выполняется последовательный анализ всех блоков модели в момент времени по известному состоянию в момент времени  $t$ .

- Новое состояние модели рассчитывается по алгоритмам модели с учетом статистики поведения. На каждом такте моделирования мы обращаемся ко всем моделям и проверяем состояние данной модели в новый момент времени (не изменилось ли оно). Если такт моделирования выбрали слишком большим, то мы можем пропустить какие-либо события, поэтому такт выбирается очень малым.

- Недостатки потактового моделирования:

- 1. Необходимо выбирать очень маленький шаг.
        2. Требуется обращение ко всем моделям, независимо от того, произошли там события или нет (большие вычислительные затраты).

- Достоинство:

- 1. Простота программной реализации.

# Лекция 3. Организация моделирования СМО

## ● Событийное моделирование:

- Событийное моделирование за счет некоторого усложнения алгоритма программы позволяет обойти недостатки потактового моделирования.
- Для соблюдения при имитации правильной хронологии событий все события, которые в данный момент модельного времени уже известны (можно предвидеть), упорядочиваются по времени наступления этих событий и оформляются в виде списка - списка будущих событий (СБС). Элементами списка являются имена или ссылки на имена заявок, с которыми связаны данные события. Иногда выделяют отдельно начало этого списка, куда относят события, которые должны произойти в текущий момент времени. Этот список называется списком текущих событий (СТС).

# Лекция 3. Организация моделирования СМО

## ● **Событийное моделирование:**

- Исполнение программы на каждом очередном шаге моделирования (новом моменте модельного времени) будет начинаться с выбора события из СТС и связанной с этим событием заявки, например заявка А.
- Программа должна выполнять все действия с этой заявкой, пока заявка не задержится, например, в очереди или в обслуживающем аппарате.
- После того как программа прерывает имитацию прохождения заявкой А, она переходит к обработке очередного события из СТС. Пусть это событие связано с заявкой В. Программа исполняется, начиная с того оператора, который определил задержку заявки В, и доходит до такого оператора, который означает новую задержку заявки В. Если эта задержка, например из-за обслуживания в некотором устройстве, то при этом определяется длительность обслуживания и, следовательно, момент времени, в который это обслуживание закончится. Далее ссылка на заявку В включается в СБС с указанием времени свершения этого будущего события - окончания обслуживания заявки В в устройстве.



# Лекция 3. Организация моделирования СМО

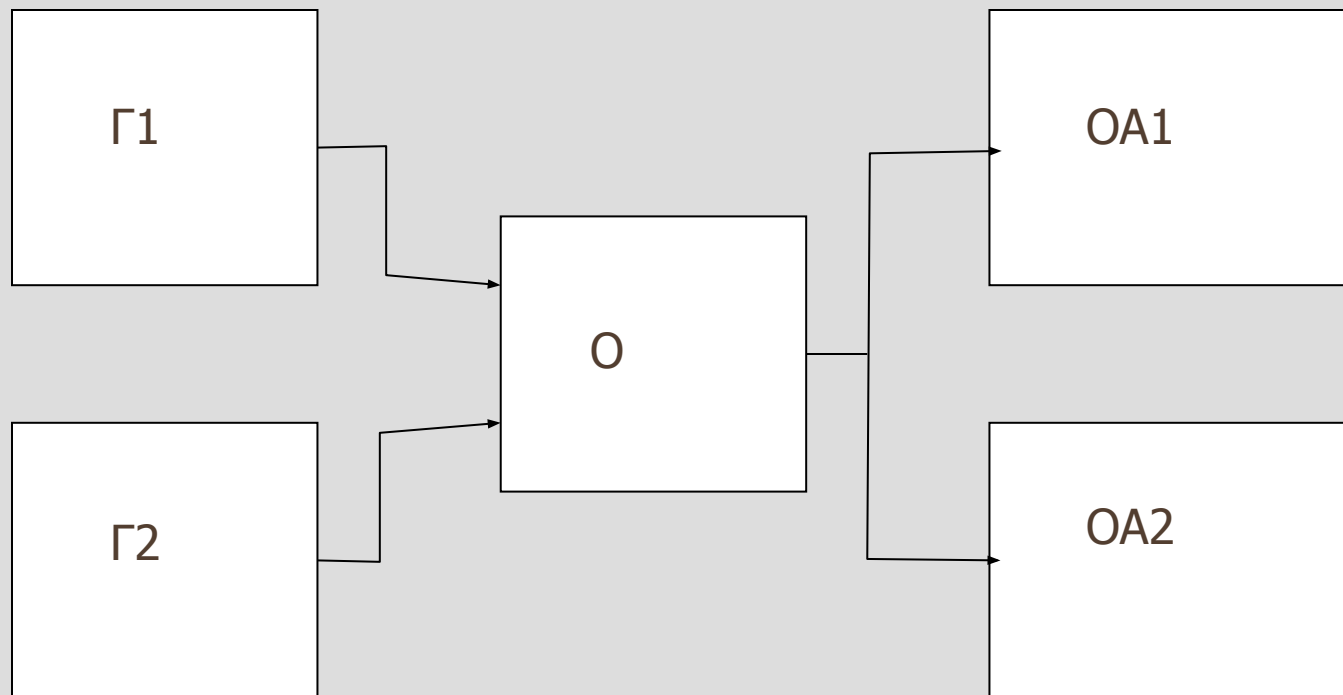
## ● Событийное моделирование:

- После этого выбирается очередной элемент из СТС и имитируется продвижение новой заявки.
- После того как все события текущего времени  $t_i$  исчерпываются, модельное время увеличивается до величины  $t_{i+1}$ , равной времени ближайшего по времени события, и тогда заявки для момента  $t_{i+1}$  будут новым списком СТС. Процесс имитации продолжается по тому же алгоритму.
- Таким образом, при использовании СБС моделирование выполняется путем обращения к моделям тех элементов ВС, с которыми связаны происходящие события, и выполняется только в те моменты времени, в которые произошли какие-либо события. Такое моделирование, избирательное по месту и времени событий, называется событийным. Достоинством его по сравнению с потактовым моделированием является значительное сокращение вычислительных затрат, так как шаг по времени здесь является оптимальным (от события до события), и обращение происходит на каждом шаге только к тем моделям элементов, в которых произошли события.

# Лекция 3. Организация моделирования СМО

- **Событийное моделирование:**

**Пример: изменение модельного времени с использованием СБС**



# Лекция 3. Организация моделирования СМО

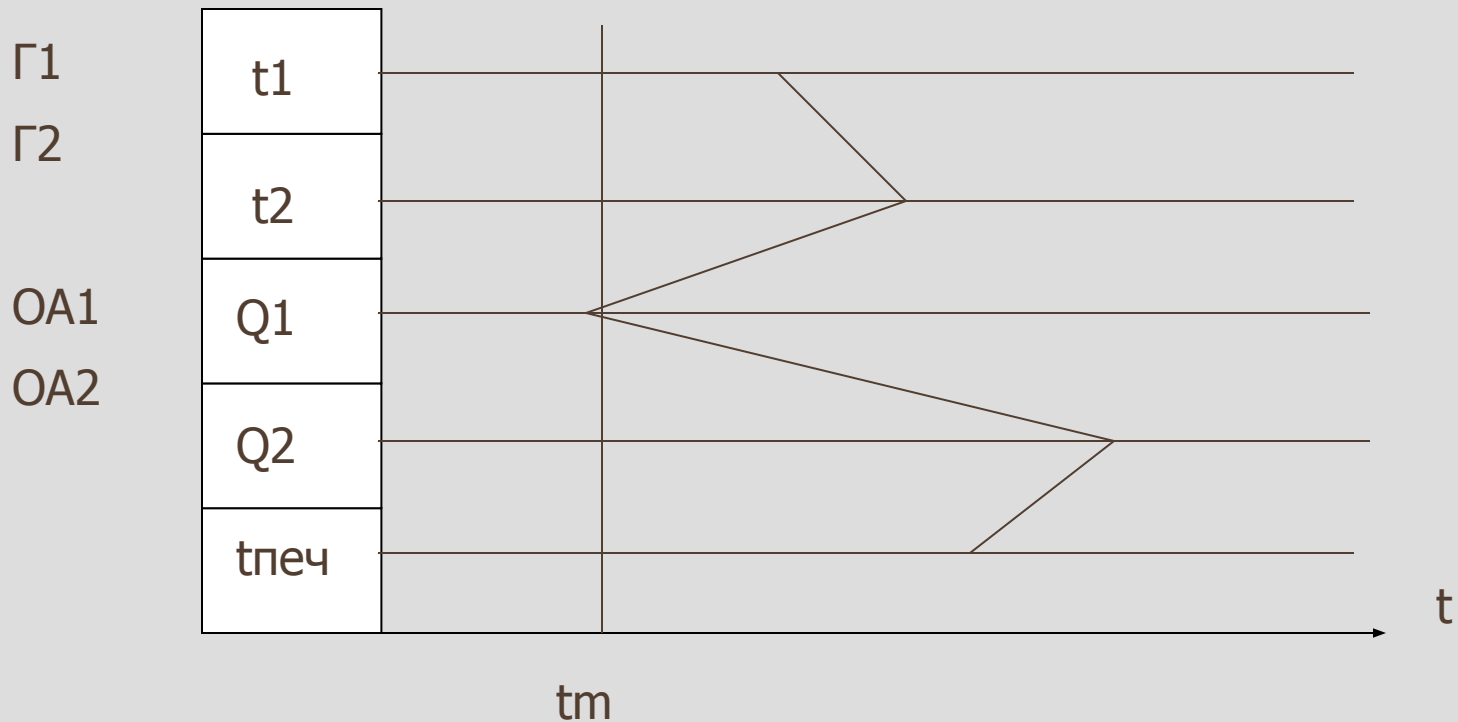
- **Событийное моделирование:**

**Пример:** Система содержит два источника заявок ( $\Gamma 1$  и  $\Gamma 2$ ), два обслуживающих аппарата ( $OA1$  и  $OA2$ ) и общую очередь к ним ( $O$ ). Кроме событий, непосредственно влияющих на работу системы, в СБС заносятся также моменты времени печати статистических сведений ( $t_{печ.}$ ) и др. Список будущих событий для этой системы представим на следующих иллюстрациях, где отражены моменты поступления заявок от  $\Gamma 1$  ( $t1$ ) и от  $\Gamma 2$  ( $t2$ ); моменты выборки заявок из очереди аппаратами  $OA1$  ( $Q1$ ) и  $OA2$  ( $Q2$ ) (предполагается, что моменты выборки заявок из очереди аппаратами совпадают с моментами времени окончания обслуживания предыдущих заявок в этих аппаратах).

# Лекция 3. Организация моделирования СМО

- Событийное моделирование:

СБС



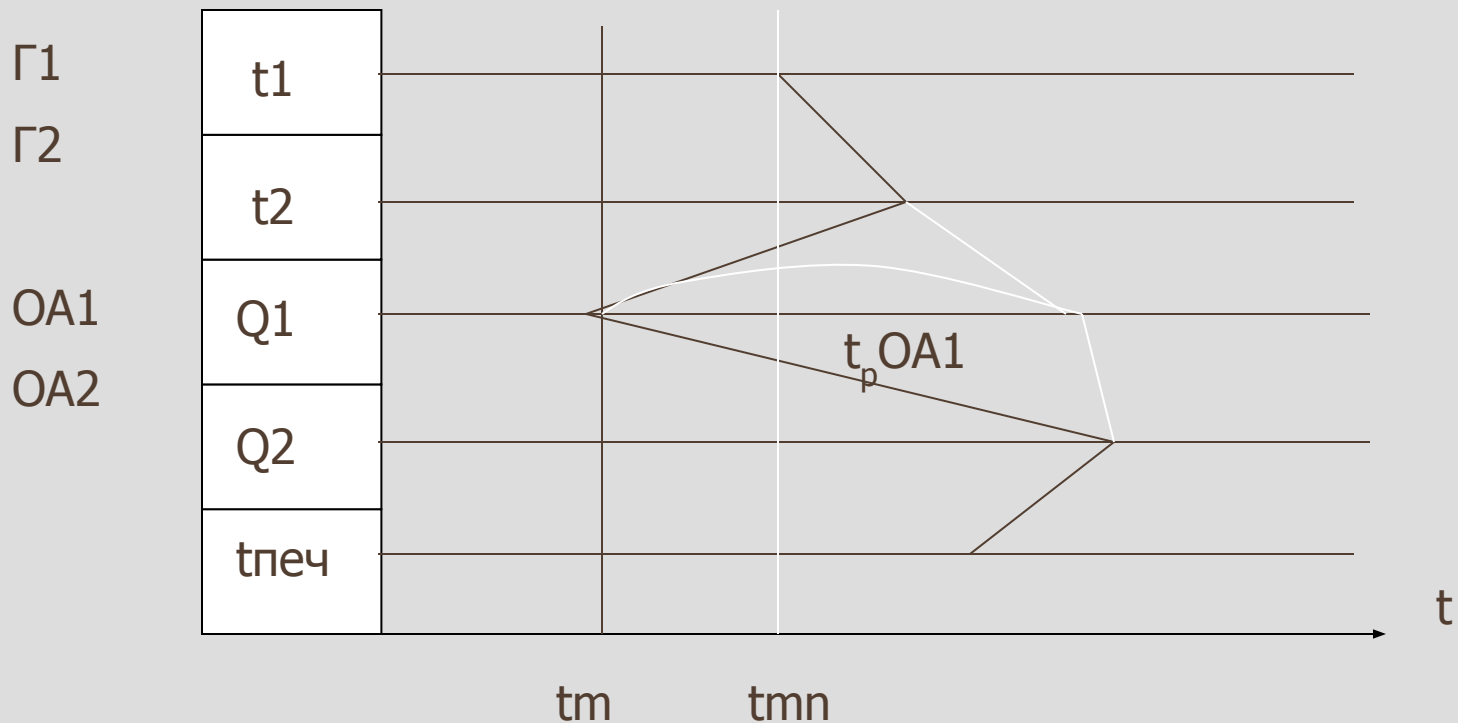
# Лекция 3. Организация моделирования СМО

- **Событийное моделирование:**
  - Упорядочивание СБС выполняется визуально.
  - Для нашего случая алгоритм моделирования будет выглядеть следующим образом:
    1. выборка из СБС ближайшего события  $Q1$ ;
    2. обращение к подпрограмме (модели)  $OA1$ , связанного с событием  $Q1$ ;
    3. выполнение подпрограммы  $OA1$ . Результатом работы подпрограммы  $OA1$  будет время обслуживания заявки в данном устройстве ( $tr.OA1$ ) и соответственно время выборки из очереди следующей заявки в  $OA1$ ;
    4. будущее время  $Q1$  становится известным и значение  $tM + tr. OA1$  заносится в СБС, заменяя старое (предыдущее) значение;
    5. очередь уменьшается на одну заявку;
    6. СБС упорядочивается, и модельное время увеличивается до ближайшего события из СБС ( $tMН$ ).
  - Процесс моделирования далее продолжается с выбора очередного ближайшего по времени события ( $t1$ , для данного примера).



# Лекция 3. Организация моделирования СМО

- Событийное моделирование:



# Лекция 3. Организация моделирования СМО

- Основные принципы работы имитационного моделирования:

**1. Исходными данными для имитационного моделирования являются:**

- Сведения о структуре системы (состав, связи).
- Сведения о дисциплинах обслуживания в каждом ОА.
- Значения параметров ОА.
- Значения параметров входных потоков заявок.

**2. Результатом имитационного моделирования являются следующие рассчитываемые характеристики:**

- Пропускная способность (производительность системы) – среднее количество обслуживаемых системой заявок в единицу времени.
- Вероятность обслуживания заявки (для тех систем, где заявки могут покидать систему необслуженными).
- Коэффициенты загрузки оборудования для каждого ОА – отношение полезного времени обслуживания к общему времени моделирования.
- Средние длины очередей к ОА.

# Лекция 3. Организация моделирования СМО

- Основные принципы работы имитационного моделирования (продолж.):
  3. *Имитационная модель* представляет собой алгоритм, состоящий из упорядоченных обращений к моделям элементов. Последовательность обращений определяется свойствами анализируемой ВС и принципами ее функционирования.
  4. *Модельное время* изменяется после того, как закончится имитация событий, относящихся к текущему моменту времени. Имитация заканчивается либо когда текущее время превысит конечное время моделирования, либо когда входные источники выработают необходимое количество заявок.
  5. *Оценка результатов* выполняется на основе принятия решений: по правильности выбранных параметров и структуры ВС, устранению узких мест в системе, изменению дисциплины обслуживания и т.д.

# Лекция 3. Организация моделирования СМО

- Организация программного обеспечения систем имитационного моделирования на основе СМО:
  - Программная реализация алгоритмов моделирования и языков описания СМО обычно оформляется в виде *пакетов прикладных программ (ППП)* сложной структуры.
  - В состав программного обеспечения входят:
    1. Управляющий модуль (управляет обращением к подпрограммам, осуществляет ввод исходных данных, упорядочивает СБС, изменяет модельное время, осуществляет продвижение заявок).
    2. Пакеты прикладных подпрограмм (исполнимые модули) – математические модели компонентов, математические модели источников входных потоков, подпрограммы статистической обработки и т.д.
    3. База данных системы (СБС, массивы очередей, числовые параметры устройств и заявок).

# Лекция 3. Организация моделирования СМО

- Организация программного обеспечения систем имитационного моделирования на основе СМО:

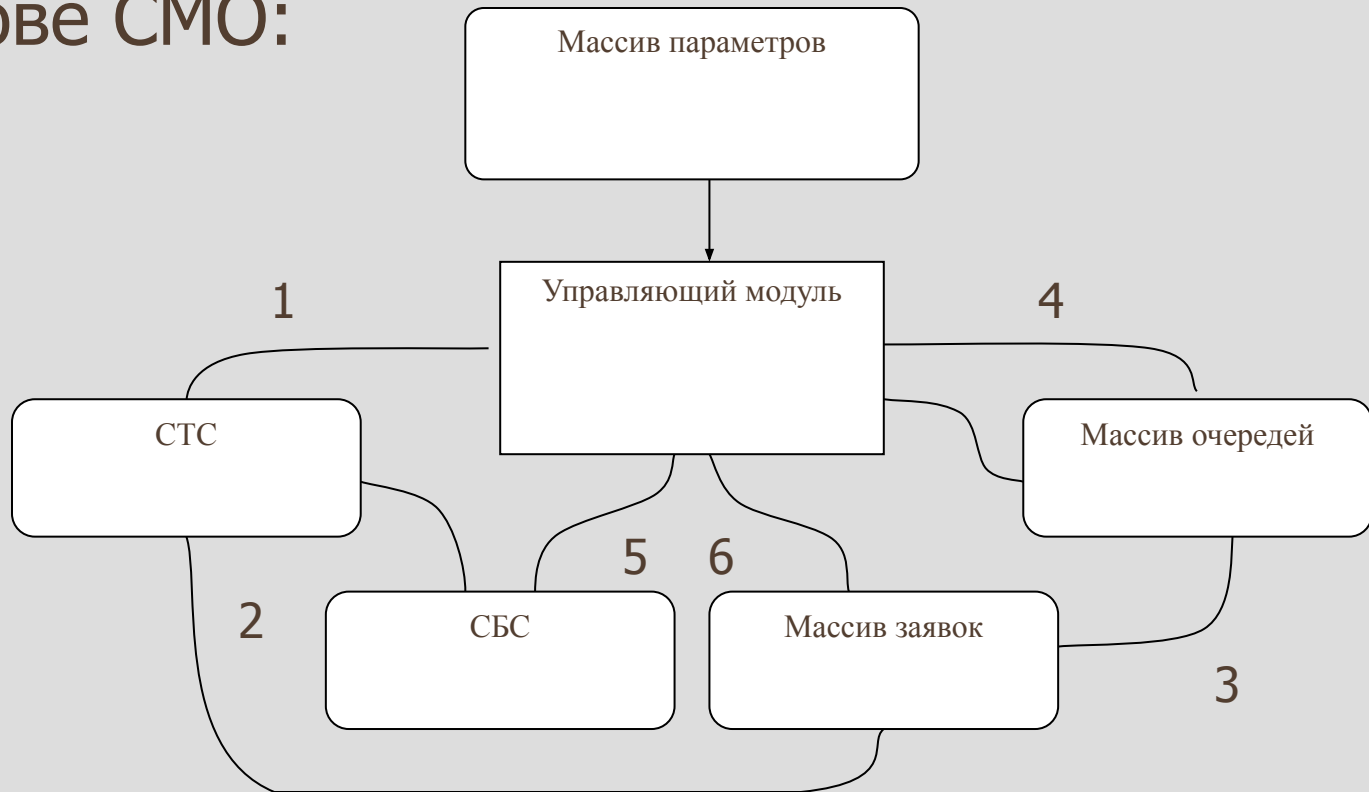
**База данных содержит следующие основные массивы данных:**

- **Массив очередей** содержит информацию о заявках в очереди и упорядочен по устройствам.
- **СТС и СБС** упорядочены по моментам наступления событий. Их элементы – это ссылки на заявки. По ссылкам могут быть найдены адреса хранения всех сведений о самих заявках.
- **Массив заявок** хранит сведения о каждой заявке: имя, тип, приоритет и время очередного события, связанного с этой заявкой, место нахождения заявки в модели.
- **Массив параметров элементов** – числовые параметры законов распределения, дисциплины обслуживания, накапливаемые суммы, длины очередей и вычисляемые характеристики.



# Лекция 3. Организация моделирования СМО

- Организация программного обеспечения систем имитационного моделирования на основе СМО:



# Лекция 3. Организация моделирования СМО

- Организация программного обеспечения систем имитационного моделирования на основе СМО:

**Каждый этап вычислений начинается с выборки очередного события из СТС (1). Элементом списка является ссылка (2) на заявку (адрес в массиве заявок). Обозначим имя обрабатываемой заявки А. По извлеченным из массива заявок сведениям (3) определяется имя освобождающегося ОА. Если к этому ОА имеется очередь (4), то ОА начинает обслуживать некоторую заявку В из очереди в соответствии с заданной дисциплиной обслуживания. При этом происходит обращение к подпрограмме рассматриваемого ОА, определяется время обслуживания, и тем самым становится предвиденным новое событие – окончание обслуживания заявки В. Ссылка на это событие помещается в список будущих событий (5), и выполняется изменение сведений о заявке В в массиве заявок (6). Далее имитируется продвижение задержанной в очереди заявки А (если нет других заявок в очереди). Это продвижение прерывается, когда производится запрос на обслуживание заявки А в некотором ОА. При этом происходит обращение к массиву очередей по имени этого ОА и либо поступление заявки А на обслуживание, либо постановка ее в очередь. После обработки всех этих действий процесс имитации повторяется, и выбирается очередной элемент из СТС.**

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ заявок

- Содержание:

- Генераторы случайных чисел
- Моделирование равномерного закона распределения
- Формирование нормального закона распределения
- Распределение Пуассона
- Формирование произвольного закона распределения.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- **Генераторы случайных чисел:**

- При статистическом моделировании одним из основных вопросов является моделирование стохастических воздействий. Для статистического моделирования характерно большое число операций со случайными числами. Кроме того, результаты статистического моделирования существенно зависят от качества последовательностей случайных чисел. Поэтому наличие простых и экономичных способов формирования случайных чисел требуемого качества во многом определяет возможность моделирования на ЭВМ.

- Существуют три способа генерации случайных чисел (СЧ): аппаратный (физический), табличный (файловый) и алгоритмический (программный).

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Генераторы случайных чисел (продолж.):
  - Аппаратный способ - случайные числа вырабатываются специальным электронным устройством – генератором (датчиком) случайных чисел.
  - Достоинствами являются: отсутствие дополнительных вычислительных операций ЭВМ по выработке случайных чисел, необходима только операция обращения к внешнему устройству ЭВМ (датчику), не ограничен запас чисел, не требуется дополнительная память ЭВМ, высокая скорость работы, генерация "хороших" последовательностей случайных цифр и чисел.
  - Недостатками являются: отсутствие гарантии качества последовательности непосредственно во время моделирования системы на ЭВМ (требуется периодические проверки), нельзя повторно получать при моделировании одинаковые последовательности чисел, необходимость дополнительного устройства.



# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Генераторы случайных чисел (продолж.):

Табличный способ - случайные числа, оформленные в виде таблицы, помещаются в виде файла в память ЭВМ. Имеются специальные справочники, где содержатся таблицы случайных чисел, полученные в результате многократного выполнения каких-либо физических опытов. Эта таблица из справочника может быть введена в ЭВМ. Таблицы представляют собой последовательности цифр от 0 до 9, имеющих равную вероятность появления. Использование таких таблиц очень простое: если требуется случайная цифра, то берется первая попавшаяся позиция из таблицы. Если требуется  $n$ -позиционное случайное число, то выбирается  $n$  последовательных очередных цифр таблицы. Цифры выбираются в любом направлении и с любого места.

Достоинства: генерация действительно случайных чисел, требуется однократная проверка, можно воспроизводить последовательности.

Недостатки: запас чисел ограничен, требуется большой объем памяти ЭВМ.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Генераторы случайных чисел (продолж.):

Алгоритмический способ - используются специальные алгоритмы и реализующие их программы.

Достоинствами являются: требуется однократная проверка, можно многократно воспроизводить последовательности чисел, занимает мало места в памяти ЭВМ, не требуется внешнее дополнительное устройство.

Недостатками являются: запас чисел последовательности ограничен ее периодом, более значительные затраты машинного времени.

- Все алгоритмы генераторов СЧ основаны на выборе следующего СЧ как функции от предыдущего, поэтому получаемое СЧ не является полностью случайным, т.к. вычисляется по определенному алгоритму. Такие числа называются псевдослучайными. Эти последовательности очень похожи на действительно случайные числа, а пригодность их применения в процессах моделирования определяется специальными тестами.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Генераторы случайных чисел (продолж.):

В качестве основных тестов используют проверки на равномерность, стохастичность и независимость.

- Проверка на равномерность осуществляется по гистограмме с использованием косвенных признаков.

- Проверка стохастичности наиболее часто проводится методами комбинаций и серий. Сущность метода комбинаций сводится к определению закона распределения длин участков между единицами (нулями) или закона распределения (появления) числа единиц (нулей) в  $n$ -разрядном двоичном числе. Серией называется любой отрезок последовательности случайных чисел, состоящий из следующих друг за другом элементов одного и того же рода. Число элементов в отрезке называется длиной серии.

- Проверка независимости проводится на основе вычисления корреляционного момента. При моделировании с использованием программных генераторов равномерных последовательностей важными характеристиками качества является *длина периода  $P$  и длина отрезка апериодичности  $L$* .

- Длина отрезка апериодичности псевдослучайной последовательности означает, что все числа в пределах отрезка апериодичности не повторяются. Очевидно, что использование при моделировании систем последовательности чисел, длина которой больше отрезка апериодичности  $L$ , может привести к повторению испытаний, что не дает новых статистических данных. Из-за того, что разрядная сетка ЭВМ конечна, при формировании последовательности чисел рано или поздно одно из них совпадет с ранее вычисленным числом. После этого все формируемые далее числа будут совпадать с уже имеющимися и при этом будут следовать в том же порядке.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Моделирование равномерного распределения:

Для имитационного моделирования требуются случайные числа с различными законами распределения – нормальным, равномерным, экспоненциальным и другими. Основой для их генерации является последовательность случайных чисел, распределенных по равномерному закону на интервале  $(0, 1)$ .

Непрерывная случайная величина имеет равномерное распределение в интервале  $(a, b)$ , если ее функция плотности имеет вид

$$f(x) = \begin{cases} 1/(b-a), & a \leq x \leq b \\ 0, & x < a, x > b. \end{cases}$$

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Моделирование равномерного распределения:

- На ЭВМ с  $n$ -разрядными числами вместо непрерывной совокупности равномерных случайных чисел интервала  $(0, 1)$  используют дискретную последовательность  $2^n$  случайных чисел того же интервала. Закон распределения такой дискретной последовательности называют *квазиравномерным распределением*.

- Наибольшее применение в практике моделирования нашли алгоритмы вида

$$x_{i+1} = F(x_i),$$

представляющие собой рекуррентные соотношения, для которых начальное значение  $x_0$  и параметры функции  $F$  заданы, например

$$X_{i+1} = \lambda X_i \pmod{M},$$

где  $\lambda, X_i, M$  - неотрицательные целые числа.

- Для реализации на ЭВМ наиболее удобно, когда  $M = p^k$ , где  $p$  - число цифр в системе счисления, принятой в ЭВМ (например,  $p = 2$  для двоичной):  $k$  - длина разрядной сетки (например  $k = 32$  для 32-разрядной ЭВМ). В этом случае вычисление остатка от деления на  $M$  сводится к выделению  $k$  младших разрядов делимого, а преобразование целого числа  $X_i$  в рациональную дробь из интервала  $(0, 1)$  осуществляется подстановкой слева от  $X_i$  двоичной запятой.



# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Формирование нормального закона распределения:
  - Нормальное распределение является одним из важнейших непрерывных распределений. Все методы базируются на использовании равномерно распределенных случайных чисел.
  - Один из часто применяемых методов основан на центральной предельной теореме, которая гласит, что сумма независимых одинаково распределенных случайных чисел с математическим ожиданием  $a_m$  и среднеквадратическим отклонением  $\sigma_m$  образует асимптотически случайное число с нормальным законом распределения и математическим ожиданием  $a = N a_m$  и среднеквадратическим отклонением  $\sigma = \sigma_m \sqrt{N}$ , где  $N$  – число суммируемых чисел. Расчеты показывают, что уже при сравнительно небольших  $N$  (8, 12) сумма имеет распределение, близкое к нормальному.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Формирование нормального закона распределения:

Нормированное распределение с  $M(x) = 0$ ,  $D(x) = 1$  можно получить, воспользовавшись преобразованием

$$y_{\text{норм}} = \sqrt{12/N} * (\sum x_{\text{равн}} - N/2).$$

В частности, при  $N=12$

$$\text{получим } y_{\text{норм}} = \sum x_{\text{равн}} - 6.$$

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- **Распределение Пуассона:**

- Пусть необходимо получить случайные числа, имеющие закон распределения Пуассона  $P(m) = \lambda^m e^{-\lambda} / m!$ .
- Для этого распределения можно также воспользоваться предельной теоремой Пуассона, которая гласит, что если при проведении  $N$  независимых испытаний вероятность свершения события  $A$  в  $i$ -ом испытании равна  $P_i$ , то относительная частота появления события  $m/N$  при  $N \rightarrow \infty$  сходится по вероятности к среднему из вероятностей  $P_i$ .
- Таким образом, если  $p$  – вероятность наступления события  $A$  при одном испытании, то вероятность наступления  $m$  событий в  $N$  независимых испытаниях при  $N \rightarrow \infty, p \rightarrow 0, Np = \lambda$  асимптотически равна  $P(m)$ .
- Для реализации алгоритма выбирается достаточно большое  $N$ , такое, чтобы  $p = \lambda/N < 1$ , проводятся серии по  $N$  независимых испытаний, в каждом из которых событие  $A$  происходит с вероятностью  $p$ , и подсчитывается число случаев  $Y_j$  фактического наступления события  $A$  в серии с номером  $j$ . Числа  $Y_j$  будут приближенно следовать закону Пуассона, причем тем точнее, чем больше  $N$ . Практически  $N$  должно выбираться, чтобы  $p \leq 0,1 - 0,2$ . (Алгоритм – в учебнике).

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Формирование произвольного закона распределения:

Рассмотрим два метода.

1. **Метод обратной функции.** Он основывается на следующей теореме: Если случайная величина (СВ)  $x$  имеет плотность распределения  $P(x)$ , то

$$j \text{ равн} = F(x) = \int_{-\infty}^x P(x) dx$$

подчиняется равномерному закону распределения в интервале  $(0, 1)$  независимо от вида  $P(x)$ . Отсюда вытекает способ моделирования случайных чисел  $x$  с произвольной плотностью вероятности  $P(x)$ .

Алгоритм: 1) Моделируют СВ  $j$  с равномерным законом распределения в диапазоне  $(0, 1)$ . 2) Решают интегральное уравнение относительно верхнего предела:  $x_i$ . Значение верхнего предела  $x_i$  и будет СВ по закону  $P(x)$ . Если  $P(x)=0$  при  $x < x_0$ , то вместо  $-\infty$  нижний предел можно заменить на  $x_0$ .

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Формирование произвольного закона распределения:

Графически эта процедура представлена на рис.9. Здесь для каждой случайной величины  $j_i$  с равномерным распределением на интервале  $(0, 1)$  находится соответствующая величина  $x_i$ , у которой плотность распределения  $P(x)$ .

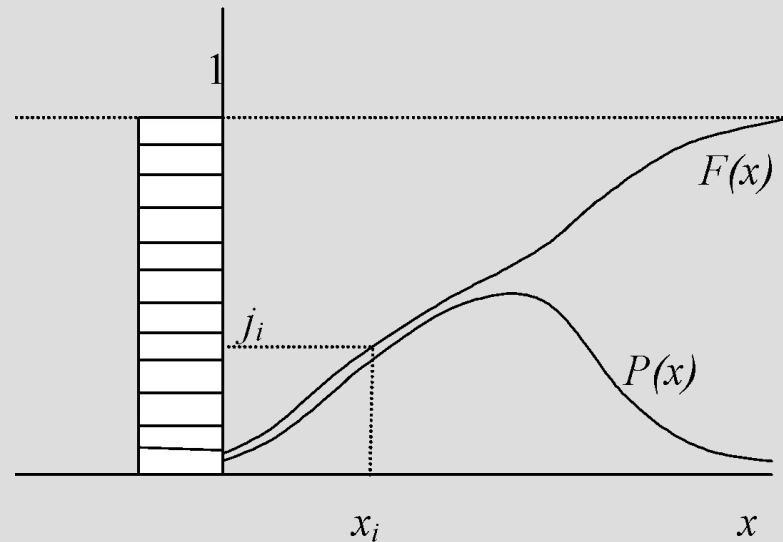


Рис. 9



# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- Формирование произвольного закона распределения:

**Пример 1.** Необходимо получить случайные числа с экспоненциальным законом распределения. Для экспоненциальной случайной величины  $x_i$ , распределенной по закону

$$p(x) = a \cdot e^{-a(x-x_0)}, \quad x_0 < x < \infty$$

имеем  $x_i$

$$j_i = \int_{x_0}^{x_i} a \cdot e^{-a(x-x_0)} dx \quad \text{откуда} \quad x_i = x_0 - 1/a \cdot \ln(1 - j_i)$$

**Пример 2.** Необходимо получить случайные числа с законом распределения  $p(x) = \lambda(1 - \lambda x/2)$ ,  $0 < x \leq 2/\lambda$ .

Воспользовавшись приведенным выше алгоритмом, получим

$$\lambda(x_j - \lambda x_j^2/4) = y_j$$

$$\text{Отсюда} \quad x_j = 2(1 - \sqrt{1 - y_j})/\lambda$$

$$\text{или} \quad x_j = 2(1 - \sqrt{y_j})/\lambda.$$

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- **Формирование произвольного закона распределения:**

**Достоинство** метода обратной функции: малые вычислительные затраты при реализации вычисления в аналитическом виде.

**Недостаток**: не все законы распределения можно представить в аналитическом виде. Приходится прибегать к численным методам решения, что увеличивает машинное время на получение каждого случайного числа. Даже в тех случаях, когда интеграл берется в аналитическом виде, получаются формулы, содержащие действия логарифмирования, извлечения корня и т.п., которые приводят к увеличению вычислительных затрат.

- Поэтому в практике моделирования часто пользуются другими, более универсальными и иногда приближенными способами.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ ЗАЯВОК

- **Формирование произвольного закона распределения:**

## Метод Неймана.

Если случайная величина  $x$  определена на конечном интервале  $(a, b)$  и плотность ее ограничена  $P(x) \leq M_0$ , то можно использовать следующий алгоритм:

1) Генерируются два значения  $x_1$  и  $x_2$  случайной величины, равномерно распределенной на интервале  $(0, 1)$ .

2) Вычисляются координаты точки  $C(n_1; n_2)$ :

$$n_1 = a + x_1 \cdot (b - a);$$

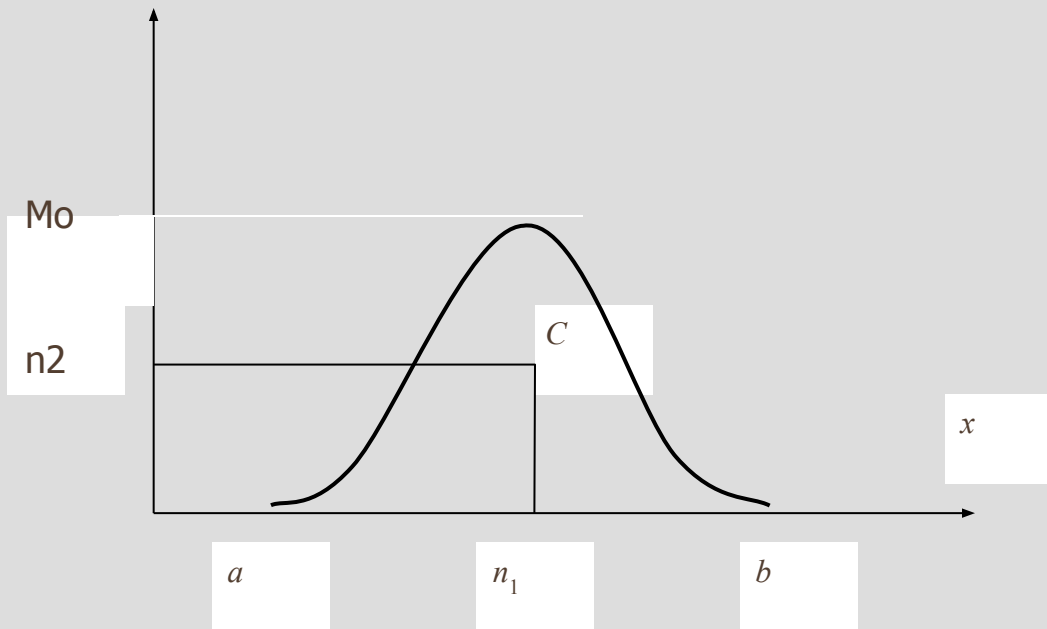
$$n_2 = x_2 \cdot M_0.$$

3) Если точка  $C$  лежит под кривой необходимого закона распределения  $P(x)$ , то в качестве очередного СЧ по заданному закону  $P(x)$  выбирают  $n_1$ , а если над кривой, то пара  $x_1$  и  $x_2$  отбрасывается и выбирается новая пара.

# Лекция 4. Модели источников ВХОДНЫХ ПОТОКОВ заявок

- Формирование произвольного закона распределения:

**Метод Неймана.** Достоинством метода является его высокая надежность и применимость к любому закону распределения. К недостатку следует отнести большие вычислительные затраты.



# Лекция 5. Языки имитационного моделирования

- **Содержание:**
  - Особенности языков моделирования.
  - Классификация языков моделирования.
  - Пример моделирования ВС.



# Лекция 5. Языки имитационного моделирования

- Особенности языков моделирования:
  - На всех других уровнях, кроме системного, заданная структура объекта однозначно дает ее математическую модель, которая получается автоматически по моделям компонентов и топологическим связям.
  - Структурные схемы ВС отображают физические связи между устройствами и состав аппаратных средств. Однако реализация тех или иных связей на системном уровне зависит от программного обеспечения, которое на структурной схеме никак не отображается.
  - Таким образом, необходимо преобразовывать структуру на каждом шаге ее функционирования так, чтобы отражались только активизированные связи между устройствами, т.е. можно говорить о переменной структуре ВС.

# Лекция 5. Языки имитационного моделирования

- Особенности языков моделирования:
  - Это значительно усложняет подготовку задания для моделирования на ЭВМ и в отличие от других уровней делает ее двухэтапной:
    - На первом этапе разрабатывается математическая модель ВС, которая обычно не является формальной и выполняется человеком. Большая трудоемкость составления математической модели - главная проблема систем имитационного моделирования. Модель представляется либо в виде схемы, либо в виде алгоритма.
    - Второй этап похож на другие уровни моделирования и заключается в представлении результатов первого этапа на языке систем имитационного моделирования.

# Лекция 5. Языки имитационного моделирования

- **Классификация языков моделирования:**
  - Модель ВС в виде алгоритма - в качестве входных языков системного уровня очень хорошо подходят алгоритмические языки.
  - Но язык моделирования должен отражать определенную структуру понятий для описания широкого класса явлений. Высокий уровень проблемной ориентации языка моделирования значительно упрощает разработку моделей, а специально предусмотренные в ней возможности сбора, обработки и вывода результатов моделирования позволяют быстро и подробно анализировать исходы имитационного эксперимента.
  - Основными моментами, характеризующими качество языков моделирования, являются: удобство описания процесса функционирования системы, удобство ввода исходных данных моделирования и варьирования структуры, алгоритмов и параметров модели, возможность реализации статистического моделирования, эффективность выполнения анализа и вывода результатов моделирования, простота отладки и контроля работы моделирующей программы, доступность восприятия и использования языка.

# Лекция 5. Языки имитационного моделирования

- **Классификация языков моделирования:**
  - К настоящему времени выделяют *языки непрерывного и дискретного моделирования*, предназначенные для имитации непрерывных и дискретных процессов.
  - Примером языка моделирования непрерывных систем, путем представления моделируемой системы в виде уравнений в конечных разностях является язык DYNAMO. Ряд систем использует прямо системы дифференциальных уравнений, например система MIMIC.
  - В рамках дискретного подхода можно выделить несколько групп языков моделирования, различающихся принципами организации процесса моделирования. Это языки, ориентированные на события, процессы и активности (действия, работы). Здесь мы рассмотрим лишь классификацию по способу описания задания на моделирование.

# Лекция 5. Языки имитационного моделирования

- **Классификация языков моделирования:**

- Алгоритмические языки.**

- Целесообразность использования алгоритмических языков объясняется тем, что модель ВС представляется в виде алгоритма. Поэтому известно немало случаев использования алгоритмических языков (Алгол, Фортран и ПЛ/1) при моделировании систем.

- Достоинства:**

- Необходимо знание только одного языка алгоритмического.
    - Необходимо только стандартное ПО.
    - Большая гибкость разработки программ, а также отладки и использования модели.

- Недостатки:**

- Неудобство описаний имитационных моделей.
    - Отсутствие лаконичных средств для операций с очередями, управления временем, синхронизации процессов в модели.
    - Трудоемкость программирования.
    - Сложность внесения изменений.



# Лекция 5. Языки имитационного моделирования

- **Классификация языков моделирования:**

- Расширение алгоритмических языков**

- (алгоритмические языки, расширенные библиотеками подпрограмм для работы с типичными операциями имитационного моделирования).**

- Языки SIMSCRIPT, SMPL и FORSIM, созданные на основе алгоритмического языка Фортран. Языки SIMULA и SOL, созданные на основе языка Алгол. Языки СЛЭНГ, НЕДИС, АРГОН – разработаны в нашей стране.**

# Лекция 5. Языки имитационного моделирования

- Классификация языков моделирования:

## Общецелевые языки имитационного моделирования.

Характерные черты:

- Специально ориентированы на имитационное моделирование; удобство программирования модели (что играет существенную роль при разработке крупных систем-моделей).
- Концептуальная направленность на СМО. Язык соответствует определениям и терминам СМО.
- Это не просто высокоуровневые языки - это системы моделирования.
- Представителями этого класса языков являются: GPSS, НЕДИС, СИМУЛА 2, СЛЭНГ, СИМСКРИПТ 2.
- К недостаткам можно отнести:
  1. Несколько большие вычислительные затраты при расчетах по имеющимся моделям по сравнению с компиляторами алгоритмических языков.
  2. Необходимость сопровождения на ЭВМ дополнительной системы, кроме имеющихся универсальных традиционных компиляторов.
  3. Необходимость изучения дополнительного языка, часто с менее кропотливо отработанной документацией, чем в традиционных компиляторах алгоритмических языков.

# Лекция 5. Языки имитационного моделирования

## ● Классификация языков моделирования:

### Специализированные языки.

Целью разработки этого класса является дальнейшее повышение эффективности имитационного моделирования специально для ВС. Характерные черты:

- Ориентация не на СМО, а на ВС.
- Включают ПО как часть ВС.
- Служат и для отладки алгоритмов операционных систем ЭВМ.

### Недостаток:

В связи с большой специализацией применение этих языков ограничено только задачами моделирования ВС, однако сокращается объем описаний моделей ВС по сравнению с описанием на общецелевых и алгоритмических языках в 4-10 и в 10-80 раз соответственно.

- Представителями данного класса являются языки: МПЛ/ВС, CSS, OASIS.

# Лекция 5. Языки имитационного моделирования

- **Классификация языков моделирования:**

**Выводы:**

**1) Создание больших имитационных моделей, включая разработку алгоритма, кодирование и отладку, требует многих месяцев работы нескольких отделов высококвалифицированных специалистов. Изменения, вносимые в модель, также приводят к большим затратам времени и средств.**

**2) Продолжаются работы по созданию новых языков, их упрощению, удобству работы, легкости внесения изменений.**

# Лекция 5. Языки имитационного моделирования

- **Пример моделирования ВС:**

**Требуется исследовать процесс функционирования ВС, состоящей из трех мини-ЭВМ (АРМ1, АРМ2 и АРМ3), коммутатора канала (КК), осуществляющего сопряжение мини-ЭВМ и ЭВМ центрального вычислительного комплекса (ЦВК) в режиме полудуплексной связи. Требуется оценить загрузку КК, ЦВК и средние времена ожидания обслуживания заявок, поступающих от АРМ.**





# Лекция 5. Языки имитационного моделирования

- **Пример моделирования ВС:**

**1) Первый этап заключается в подготовке описания модели. Рассмотрим решение этой задачи на примере составления схемы сетевой имитационной модели (СИМ).**

**- Сетевая имитационная модель составляется на основе алгоритма функционирования ВС, который мы представим далее. В СИМ будут представлены те же устройства, что и в структурной схеме, но будут введены дополнительно источники заявок  $T1 - T3$  и узлы трех типов:**

**- Узлы типа  $P (P1, P2, P3)$  - имитируют направление заявок либо к источникам, либо к КК в соответствии с заданными вероятностями.**

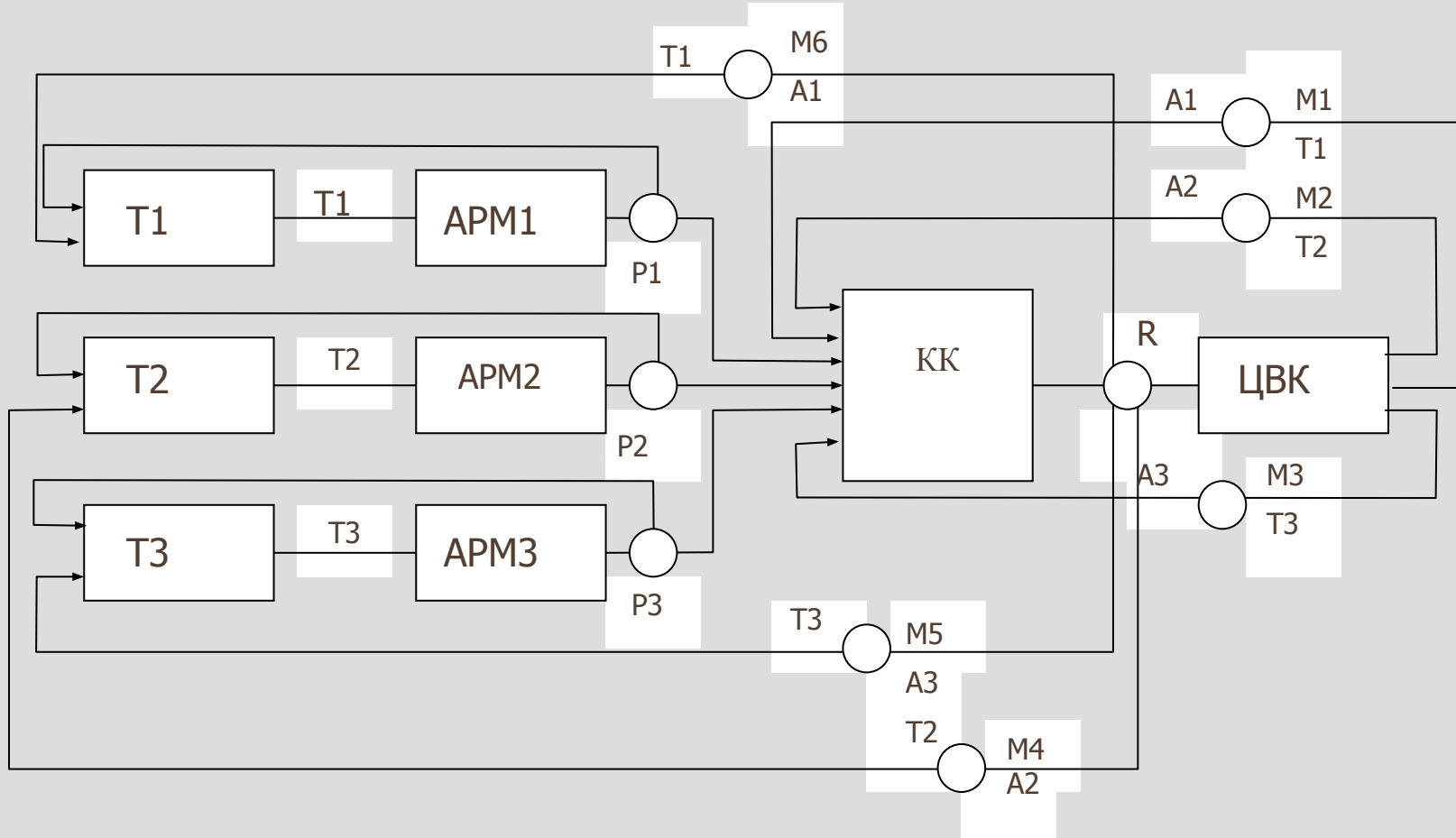
**- Узлы типа  $R (R1)$  - имитируют распределение заявок по всевозможным направлениям в зависимости от их типа.**

**- Узлы типа  $M (M1, M2, M3, M4, M5, M6)$  - изменяют тип заявки.**

**- Источники заявок  $T1, T2$  и  $T3$  имитируют работу операторов в диалоговом режиме с мини-ЭВМ через устройства ввода-вывода, имеющиеся в АРМ.**

# Лекция 5. Языки имитационного моделирования

- Пример моделирования ВС:



# Лекция 5. Языки имитационного моделирования

- Пример моделирования ВС:

**Исходными данными для моделирования являются:**

- числовые параметры источников входных заявок;
- числовые параметры ОА: АРМ, КК, ЦВК;
- вероятности решения задач на АРМ.

**Алгоритм функционирования:**

Работа начинается с генерации каждым источником по одной заявке, которым присваиваются соответственно типы  $T_1$ ,  $T_2$  и  $T_3$ . Тем самым имитируется первое обращение оператора к ЭВМ. Затем эти обращения обрабатываются в АРМ и в зависимости от характера решаемых задач (указываемых в исходных данных) может вызываться обращение к ЦВК или запрос (ответ) оператору. В СИМ эти две альтернативы отображаются узлами  $P_1$ ,  $P_2$  и  $P_3$ , а вероятности задаются в исходных данных. Задачи, направленные в ЦВК, проходят КК, возможно с ожиданием в очереди, а затем через узел  $R_1$  поступают в ЦВК, где либо поступают на обслуживание, либо ожидают в очереди. Заявки после освобождения из ЦВК (в виде результатов решения) направляются через КК к оператору, опять возможно с ожиданием в очереди, а чтобы в узле  $R_1$  они опять не попали в ЦВК, в узлах  $M_1$ ,  $M_2$  и  $M_3$  тип заявок изменяется с  $T$  на  $A$ . Далее в узлах  $M_4$ ,  $M_5$  и  $M_6$  восстанавливается исходный тип заявок ( $T$ ). Затем, поступившие в АРМ заявки (решения), могут либо выводиться на печать, либо дополнительно обрабатываться на мини-ЭВМ и инициировать генерацию новых заявок  $T_1$ ,  $T_2$  и  $T_3$ .

# Лекция 5. Языки имитационного моделирования

- Пример моделирования ВС:

**2) Второй этап состоит в описании полученной СИМ средствами входного языка системы моделирования. Ниже приведен текст описания модели на языке GPSS.**

- 1 GENERATE ,,, 1
- 2 M1 ASSIGN 1, MM1
- 3 SEIZE ARM1
- 4 ADVANCE 2, F2
- 5 RELEASE ARM1
- 6 TRANSFER 0.75, M7, M2
- 7 M2 SEIZE T1
- 8 ADVANSE 4, F2
- 9 RELEASE T1
- 10 TRANSFER .4, M1, M11
- 11 GENERATE ,,, 1
- 12 M3 ASSIGN 1, MM3
- 13 SEIZE ARM2
- 14 ADVANCE 2, F2
- 15 RELEASE ARM2
- 16 TRANSFER 0.75, M7, M4
- 17 M4 SEIZE T2
- 18 ADVANSE 4, F2
- 19 RELEASE T2
- 20 TRANSFER .4, M3, M11
- 21 GENERATE ,,, 1

- 22 M5 ASSIGN 1, MM5 Моделирование

- 23 .....

# Лекция 6. Планирование машинного эксперимента

- **Содержание:**
  - Планирование эксперимента
  - Обработка и анализ результатов моделирования
  - Анализ и интерпретация результатов моделирования



# Лекция 6. Планирование машинного эксперимента

- Планирование эксперимента:

Планирование эксперимента с программной моделью связано с вопросами эффективного использования ресурсов ЭВМ и определением конкретных способов проведения испытаний модели. Планирование эксперимента связано прежде всего с решением проблем:

- 1) определения начальных условий и их влияния на достижение установившегося результата при моделировании;
- 2) обеспечения точности и достоверности результатов моделирования.

# Лекция 6. Планирование машинного эксперимента

- Планирование эксперимента:

- Определение начальных условий и их влияние на результаты моделирования.**

- Проблема из-за искусственного характера процесса функционирования модели, в отличие от реальной системы. Поэтому при очередном прогоне модели требуется определенное время для достижения установившихся режимов работы (занятия устройств и очередей, памяти), соответствующих условиям функционирования реальной системы. Начальный период работы модели искажается из-за влияния начальных условий запуска модели. Существует несколько способов, например:

- а) Исключить из рассмотрения информацию о модели, полученную в начальной части периода моделирования, т.е. запоминать и обрабатывать результаты работы модели через достаточно большой период от начала моделирования, который соответствует установившемуся режиму.

- б) Задавать такие начальные условия для модели, которые соответствовали бы работе реальной системы.

- Но все имеют недостатки, так в первом случае увеличивается значительно время моделирования, во втором подходе существует проблема - еще до начала моделирования надо знать условия работы реальной системы.

# Лекция 6. Планирование машинного эксперимента

- **Планирование эксперимента:**  
**Обеспечение точности и достоверности результатов моделирования.**

**Эта проблема возникает из-за того, что мы используем вероятностное моделирование, и никакой машинный эксперимент принципиально не дает точного результата. Инженер всегда решает компромиссную задачу сокращения вычислительных затрат при увеличении точности.**

**Нет универсальных способов решения этих задач в первую очередь из-за того, что законы распределения в сложных моделях неизвестны. К тому же показатели качества при моделировании не могут быть точно оценены, в лучшем случае можно получить только некоторую оценку такого показателя.**

# Лекция 6. Планирование машинного эксперимента

- **Планирование эксперимента:**

- Обеспечение точности и достоверности результатов моделирования.**

Пусть  $E$  - показатель качества системы,  $E'$  - оценка показателя качества системы, в общем случае  $E' \neq E$ . При этом  $E$  называется *точностью* (абсолютной) *оценки*. Вероятность того, что неравенство  $(E - E') < \varepsilon$  выполняется, называется *достоверностью оценки*  $Q = P\{|E - E'| < \varepsilon\}$ .

- Основным приемом при решении задач, когда закон распределения в сложных моделях неизвестен, является выдвижение предположений о характере законов распределения случайной величины.

- Рассмотрим пример определения взаимосвязи точности и достоверности результатов моделирования с количеством реализаций  $N$  при программном эксперименте, когда в качестве показателя выступает вероятность  $p$ .

# Лекция 6. Планирование машинного эксперимента

- Планирование эксперимента:

**Обеспечение точности и достоверности результатов моделирования.**

Существует две постановки задачи:

1. При заданных величинах точности и достоверности ( $\varepsilon$  и  $Q$ ) определить необходимое число испытаний  $N$ .
2. При заданном  $N$  найти  $\varepsilon$  и  $Q$ .

Рассмотрим пример определения взаимосвязи точности и числа испытаний при расчете вероятности  $p$  свершения события  $A$ , где в качестве оценки вероятности выступает частность  $\hat{p} = P(A) = m/N$ , где  $m$  – число свершений события  $A$ .

**Тогда соотношение, связывающее точность и достоверность оценок с количеством реализаций, будет иметь вид**

$$Q = P\{|p - m/N| < \varepsilon\}, \quad Q = P\{p - \varepsilon \stackrel{(3)}{<} m/N < p + \varepsilon\}.$$



# Лекция 6. Планирование машинного эксперимента

- Планирование эксперимента:

## Обеспечение точности и достоверности результатов моделирования.

Опуская выкладки по определению математического ожидания и дисперсии, оценки несмещенности и применения центральной предельной теоремы, можно сделать заключение, что частность  $m/N$  при достаточно больших  $N$  можно описывать нормальным законом распределения с математическим ожиданием  $p$  и дисперсией  $p(1-p)/N$ .

Поэтому соотношение (3) с учетом теоремы Лапласа можно переписать как:

$$P\left\{(p - \varepsilon) < \frac{m}{N} < (p + \varepsilon)\right\} = \Phi_0\left(\frac{p + \varepsilon - p}{\sqrt{p(1-p)}} \sqrt{N}\right) - \Phi_0\left(\frac{p - \varepsilon - p}{\sqrt{p(1-p)}} \sqrt{N}\right) = Q,$$

где  $\Phi_0(X) = \frac{1}{\sqrt{2\pi}} \int_0^X e^{-\frac{t^2}{2}} dt$  — интеграл вероятностей.

# Лекция 6. Планирование машинного эксперимента

## Обеспечение точности и достоверности результатов моделирования.

Учитывая, что  $\Phi_0(-z) = 1 - \Phi_0(z)$ , получим

$$2\Phi_0\left(\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right) = 1 + Q; \quad \Phi_0\left(\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}}\right) = (1 + Q)/2 = \phi.$$

Тогда  $\frac{\varepsilon\sqrt{N}}{\sqrt{p(1-p)}} = t_\phi$  где  $t_\phi$  - квантиль нормального закона

распределения порядка  $\phi = (1 + Q)/2$ , находится из специальных таблиц. В результате точность оценки можно определить как

$$\varepsilon = t_\phi \sqrt{\frac{p(1-p)}{N}},$$

т.е. точность оценки обратно пропорционально  $\sqrt{N}$ . Количество реализаций определяется соотношением

$$N = t_\phi^2 p(1-p) / \varepsilon^2.$$

# Лекция 6. Планирование машинного эксперимента

- **Планирование эксперимента:**

**Обеспечение точности и достоверности результатов моделирования.**

**Пример: Чтобы оценить, как примерно соотносится число реализаций с точностью и достоверностью, рассмотрим пример расчета количества реализаций  $N$ , когда в качестве показателя эффективности используется вероятность  $p$  при достоверности  $Q = 0,95$  ( $t\phi = 1,96$ ) и точности  $\epsilon = 0,01; 0,02; 0,05$ .**

**- Так как значения  $p$  до проведения моделирования (эксперимента) неизвестны, то вычислим множество оценок  $N$  для диапазона возможных значений  $p$  от 0 до 1 с шагом 0,1. Результаты расчетов представлены в табл.**

# Лекция 6. Планирование машинного эксперимента

- Планирование эксперимента:

**Обеспечение точности и достоверности результатов моделирования.**

Вероятность, $p$	Точность,		
	0,05	0,02	0,01
0,1 (0,9)	140	900	3600
0,2 (0,8)	250	1500	6200
0,3 (0,7)	330	2100	8400
0,4 (0,6)	380	2300	9400
0,5	390	2400	9800

# Лекция 6. Планирование машинного эксперимента

- **Планирование эксперимента: Обеспечение точности и достоверности результатов моделирования.**

Чаще всего на начальных стадиях моделирования, когда решается вопрос выбора количества реализаций  $N$ , значение  $p$  неизвестно. Поэтому на практике выполняют предварительное моделирование для произвольно выбранного значения  $N_0$ , определяют  $p_0$ , а затем по точной формуле вычисляют, используя вместо  $p$  значение  $p_0$ , необходимое количество реализаций  $N$ .

При очень высоких точностях по этим формулам невозможно оценить малые вероятности. Для оценивания малых вероятностей с высокой точностью необходимо очень большое число реализаций. На практике, для оценивания вероятностей порядка  $p \sim 10^{-k}$ , количество реализаций выбирают равным  $N = 10^{k+1}$ . Очевидно, что даже для сравнительно простых моделей метод статистического моделирования приводит к большим затратам машинного времени.



# Лекция 6. Планирование машинного эксперимента

- **Обработка и анализ результатов моделирования:**  
**При выборе методов обработки результатов моделирования существенную роль играют две особенности машинного эксперимента с моделью.**
  - 1) Вероятностное моделирование на ЭВМ требует большого числа прогонов модели и хранения большого числа статистических данных. Для решения этой проблемы используют специальные рекуррентные алгоритмы обработки, которые позволяют по ходу моделирования вычислять оценки пользуясь достаточно простыми асимптотическими формулами.**
  - 2) Сложность ВС и моделей делает невозможным давать априорно суждение о законах распределения характеристик модели, поэтому при моделировании используются непараметрические оценки и оценки моментов распределения, а не сами распределения.**

# Лекция 6. Планирование машинного эксперимента

- **Обработка и анализ результатов моделирования:**  
**Рассмотрим некоторые удобные для программной реализации методы оценки распределений при достаточно большом объеме выборки. Математическое ожидание и дисперсия случайной величины  $\xi$  соответственно имеют вид**

$$a = m[\xi] = \int f(x) \cdot x dx$$

$$\delta^2 = D[\xi] = m[(x-a)^2] = \int (x-a)^2 \cdot f(x) \cdot x dx$$

**Так как плотность распределения априори неизвестна, то определить эти моменты при проведении эксперимента нельзя, поэтому приходится использовать некоторые оценки моментов при конечном числе реализаций  $N$ . В качестве таких оценок используются**

$$\hat{x} = \hat{a} = 1/N \cdot \sum x_i$$

$$S^2 = \hat{\delta}^2 = 1/N \cdot \sum (x_i - \hat{x})^2$$

# Лекция 6. Планирование машинного эксперимента

- **Обработка и анализ результатов моделирования:**  
**К качеству оценок, полученных в результате статистической обработки результатов моделирования, предъявляются следующие требования:**
  - 1) Несмещенность** оценки - равенство математического ожидания оценки определяемому параметру  $m[\hat{g}] = g$ , где  $\hat{g}$  - оценка параметра  $g$ .
  - 2) Эффективность** оценки - минимальность среднего квадрата ошибки данной оценки  $m[(\hat{g}_{min} - g)^2] \leq m[(\hat{g}_i - g)^2]$ , где  $\hat{g}_{min}$  - рассматриваемая оценка,  $\hat{g}_i$  - любая другая оценка.
  - 3) Состоятельность** оценки - сходимость по вероятности при  $N \rightarrow \infty$  к оцениваемому параметру  $\lim p(|\hat{g} - g| \geq \varepsilon) = 0$ .

# Лекция 6. Планирование машинного эксперимента

- **Обработка и анализ результатов моделирования:**  
При реализации на ЭВМ сложных моделей при большом числе прогонов получается значительный объем информации. Поэтому необходимо так организовать процесс вычислений и хранения результатов моделирования, чтобы оценки искомых характеристик формировались постепенно по ходу моделирования и без специального запоминания всей информации. Рассмотрим более экономичные формулы вычисления оценок:  
а) расчет вероятности наступления события  $A$ . В качестве оценки для искомой вероятности  $p=P(A)$  используется частота наступления события  $m/N$ , где  $m$  - число свершений события  $A$ ;  $N$  - общее число исходов. Такая оценка вероятности является состоятельной, несмещенной и эффективной. В памяти ЭВМ достаточно одной ячейки, где накапливается число  $m$ , при условии, что  $N$  задано заранее;

# Лекция 6. Планирование машинного эксперимента

- **Обработка и анализ результатов моделирования:**
  - б) закон распределения.** Область возможных значений случайной величины разбивается на  $n$  интервалов. Затем накапливается количество попаданий случайной величины в эти интервалы  $m_k$ . Оценкой для вероятности попадания случайной величины в интервал с номером  $k$  служит величина  $m_k / N$ . Необходимо фиксировать  $n$  значений  $m_k$ , т.е. требуется иметь  $n$  ячеек памяти;
  - в) среднее значение.** Накапливается сумма возможных значений случайной величины  $y_k$ . Тогда среднее значение  $\hat{y} = 1/N \sum y_k$ . Требуется всего лишь одна ячейка для накапливаемой суммы.
  - г) оценка дисперсии.** В качестве оценки можно использовать выражение  $S^2 = 1/N \cdot \sum (y_k - \hat{y})^2$ , но непосредственное вычисление по этой формуле нерационально, так как здесь используется среднее значение, которое изменяется в процессе накопления и неизвестно в момент промежуточных вычислений. Более рационально использовать  $S^2 = (\sum y_k^2 - (\sum y_k)^2 / N) / (N-1)$  и накапливать две суммы.



# Лекция 6. Планирование машинного эксперимента

- Анализ и интерпретация результатов моделирования:

Решение этой задачи может быть выполнено различными методами в зависимости от целей и вида характеристик. Рассмотрим особенности использования двух методов: корреляционного и регрессионного анализа.

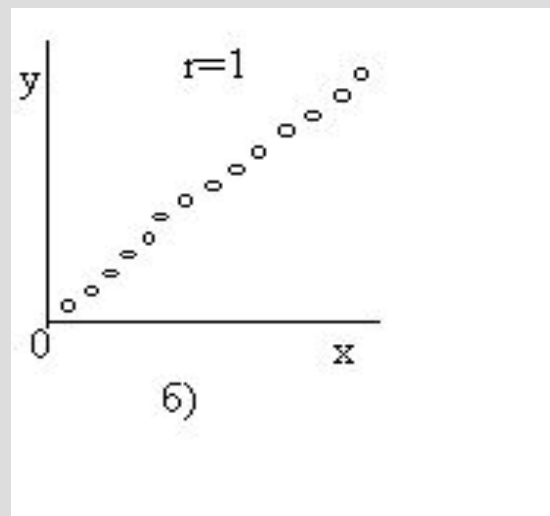
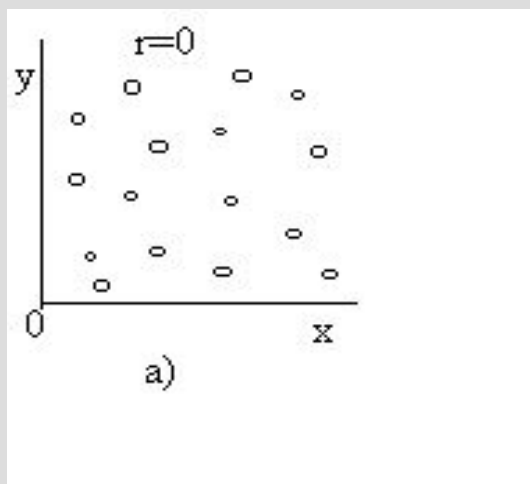
Корреляционный анализ используется для определения связи между двумя и более случайными процессами, наблюдаемыми при моделировании. Фактически он сводится к оценке разброса первой случайной величины  $x$  относительно среднего значения второй случайной величины  $\hat{y}$ . Существование этих связей можно выразить с помощью коэффициента корреляции:

# Лекция 6. Планирование машинного эксперимента

- Анализ и интерпретация результатов моделирования:

$$r_{xe} = (\sum x_k y_k - N \cdot \hat{x} \cdot \hat{y}) / \sqrt{\{(\sum x_k^2 - N \cdot \hat{x}^2) \cdot (\sum y_k^2 - N \cdot \hat{y}^2)\}}.$$

Где  $|r_{xe}| \leq 1$  – коэффициент корреляции (безразмерная величина).



# Лекция 6. Планирование машинного эксперимента

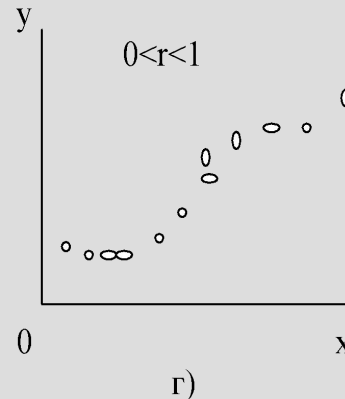
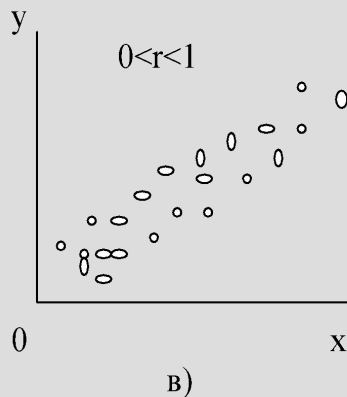
- Анализ и интерпретация результатов моделирования:

При  $r = 0$  считается, что статистически два процесса независимы (рис. а). При  $|r|=1$  считается, что статистически два процесса полностью зависимы - линейная зависимость для рис. б), причем если  $r>0$ , то говорят о положительной корреляции, т.е. большие значения одной случайной величины соответствуют большим значениям другой. Случай  $0<r<1$  соответствует либо наличию линейной корреляции с рассеянием (рис. в), либо наличию нелинейной корреляции результатов (рис. г).

# Лекция 6. Планирование машинного эксперимента

- Анализ и интерпретация результатов моделирования:

При анализе важно отметить, что даже если удалось установить тесную связь между двумя переменными, то отсюда еще не следует их взаимообусловленность. При статистическом моделировании наличие такой связи может иметь место, например, из-за коррелированности последовательностей псевдослучайных чисел.



# Лекция 6. Планирование машинного эксперимента

- Анализ и интерпретация результатов моделирования:  
**Регрессионный анализ дает возможность построить модель, наилучшим образом соответствующую набору данных, полученных в ходе машинного эксперимента. В регрессионном анализе минимизируется функция ошибки, являющаяся разностью между моделью и данными эксперимента, а в качестве такой функции берется сумма квадратов ошибок.**

**- Рассмотрим особенности регрессионного анализа результатов моделирования при построении линейной модели. На рис. показаны точки полученные в эксперименте с моделью.**

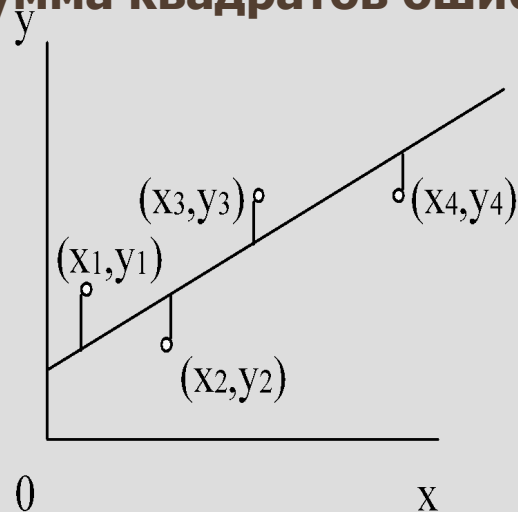


Рис. 27



# Лекция 6. Планирование машинного эксперимента

- Анализ и интерпретация результатов моделирования:
  - Далее предполагается, что модель может быть описана линейным уравнением в виде прямой линии  $\hat{y} = f(x) = b_0 + b_1 x$ , где  $\hat{y}$  - величина, получаемая по регрессионной модели.
  - Требуется получить такие значения коэффициентов  $b_0$  и  $b_1$ , при которых сумма квадратов ошибок является минимальной. Ошибки  $e_i$  для каждой экспериментальной точки определяются как расстояние по вертикали от точки до линии регрессии  $\hat{y} = f(x)$ .
  - Выражение для ошибок  $e_i = \hat{y}_I - y_I = b_0 + b_1 x_i - y_I$ , а функция ошибки  $F_0 = \sum (b_0 + b_1 x_i - y_I)^2$ . Для нахождения  $b_0$  и  $b_1$ , при которых  $F_0$  является минимальной, применяют методы минимизации, например метод наименьших квадратов. Условия:  $\partial F_0 / \partial b_0 = 0$ ;  $\partial F_0 / \partial b_1 = 0$ .
  - Дифференцируя, получим:
$$\partial F_0 / \partial b_0 = 2(N b_0 + b_1 \sum x_i - \sum y_I) = 0;$$
$$\partial F_0 / \partial b_1 = 2 b_0 \sum x_i + 2 b_1 \sum x_i^2 - 2 \sum x_i y_I = 0.$$

# Лекция 6. Планирование машинного эксперимента

- Анализ и интерпретация результатов моделирования:
  - Решая систему из двух уравнений, можно получить значения  $b_0$  и  $b_1$ .

$$\begin{pmatrix} N \sum_{i=1}^N x_i \\ \sum_{i=2}^N x_i \sum_{i=1}^N x_{2i} \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N y_i \end{pmatrix}.$$

$$b_0 = \left( \sum_{i=1}^N y_i \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \sum_{i=1}^N x_i y_i \right) / \left[ N \sum_{i=1}^N x_i^2 - \left( \sum_{i=1}^N x_i \right)^2 \right];$$

$$b_1 = \left( N \sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i \right) / \left[ N \sum_{i=1}^N x_i^2 - \left( \sum_{i=1}^N x_i \right)^2 \right].$$

- Аналогично могут быть оценены коэффициенты уравнения регрессии и для случая нелинейной аппроксимации

# Лекция 7. Аналитическое моделирование

## ● **Содержание:**

- **Основные определения вероятностных событий и потоков**
- **Примеры аналитического моделирования ВС**
- **Ограничения аналитического моделирования**

# Лекция 7. Аналитическое моделирование

- Основные определения вероятностных событий и потоков:

Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Поток событий называется однородным, если он характеризуется только моментами наступления этих событий и задается последовательностью  $\{t_i\} = \{0 < t_1 \leq t_2 \dots \leq t_i \leq \dots\}$ , где  $\{t_i\}$  – момент наступления  $i$ -го события. Однородный поток событий также может быть задан в виде последовательности промежутков времени между двумя соседними событиями  $\{\tau_i\}$ , где  $\{\tau_i\} = t_{i+1} - t_i$ .

Потоком неоднородных событий называется последовательность, характеризующаяся моментами наступления и набором признаков для этих событий  $\{t_i, f_i\}$ . В качестве признаков могут быть: принадлежность к тому или иному источнику заявок; наличие приоритета; возможность обслуживания тем или иным ОА.

# Лекция 7. Аналитическое моделирование

- Основные определения вероятностных событий и потоков:
  - Если  $\{t_i\}$  и  $\{t_{i+1}\}$  являются независимыми, то такой поток называется поток с ограниченным последствием. Поток называется ординарным, если вероятность того, что в системе находится больше чем одна заявка в момент времени  $t$  на интервале  $dt$   $P_{>1}(t, dt)$ , пренебрежимо мала по сравнению с вероятностью того, что в системе находится одна заявка в момент времени  $t$  на интервале  $dt$   $P_1(t, dt)$ , т.е.  $P_1(t, dt) \gg P_{>1}(t, dt)$ .
  - Если для любого интервала  $dt$   
$$P_0(t, dt) + P_1(t, dt) + P_{>1}(t, dt) = 1,$$
как сумма вероятностей событий, образующих полную группу, то для ординарного потока событий  
$$P_0(t, dt) + P_1(t, dt) = 1, \quad P_{>1}(t, dt) = 0.$$



# Лекция 7. Аналитическое моделирование

- Основные определения вероятностных событий и потоков:
  - Стационарным называется поток, для которого вероятность появления того или иного числа событий на интервале  $t$  зависит лишь от длины этого участка  $t$  и не зависит от момента времени, на котором расположен этот участок  $t$ .
  - Интенсивностью (плотностью) ординарного потока событий называют предел:  $\lim [P_1(t, dt) / dt = \lambda(t)$ .
  - Для стационарного потока событий  $\lambda(t) = \lambda = const$  и означает среднее число событий в системе за единицу времени.

# Лекция 7. Аналитическое моделирование

## ● Основные определения вероятностных событий и потоков:

Классификация систем:

- Системы с потерями - это системы, в которых отсутствуют очереди.
- Системы с ожиданием - когда все очереди бесконечной емкости.
- Системы смешанные - когда очередь имеет ограниченную длину.

Системы в зависимости от работы с приоритетами делятся на бесприоритетные (приоритеты равны), системы с абсолютным приоритетом и системы с относительным приоритетом. В зависимости от динамики приоритетов различают статические и динамические приоритеты. Статические приоритеты являются фиксированными в пределах решения конкретной задачи моделирования и назначаются заранее. Динамические приоритеты могут менять свои значения при моделировании в зависимости от возникающих ситуаций.

# Лекция 7. Аналитическое моделирование

- Основные определения вероятностных событий и потоков:

Относительный приоритет означает, что заявка с более высоким приоритетом ожидает окончания обслуживания предшествующей заявки и только после этого занимает устройство. Абсолютный приоритет означает, что заявка с более высоким приоритетом при поступлении в очередь прерывает обслуживание заявки с более низким приоритетом и сама занимает устройство (вытесненная заявка может либо покинуть систему, либо поступает опять в очередь, либо занимает устройство сразу после окончания обслуживания заявки с более высоким приоритетом).

# Лекция 7. Аналитическое моделирование

- **Пример аналитического моделирования ВС:**

При аналитическом моделировании выделяют все внутренние параметры системы ( $x$ ), вектор внешних ( $Q$ ) параметров (в качестве него используются параметры входных потоков заявок) и вектор выходных результатов ( $Y$ ).

$$Y = f(X, Q).$$

- Аналитическим моделированием называется представление в виде аналитической зависимости  $Y$  как функции от  $X$  и  $Q$  для системы в целом, а также аналитическое разрешение этого уравнения.

- Пример. Моделирование системы с одним обслуживающим аппаратом и очередью к нему.

- Допустим, что процесс обслуживания начинается при отсутствии заявок в очереди. Тогда состояние СМО описывается следующей системой уравнений:

# Лекция 7. Аналитическое моделирование

- Пример аналитического моделирования ВС:

$$P_n(t + \Delta t) = P_n(t)(1 - \lambda\Delta t)(1 - \mu\Delta t) + P_{n-1}(t)\lambda\Delta t(1 - \mu\Delta t) + P_{n+1}(t)\mu\Delta t(1 - \lambda\Delta t), n \geq 1;$$

$$P_0(t + \Delta t) = P_0(t)(1 - \lambda\Delta t) + P_1(t)\mu\Delta t(1 - \lambda\Delta t), n = 0,$$



# Лекция 7. Аналитическое моделирование

- Пример аналитического моделирования ВС:

где  $P_n(t)$  - вероятность нахождения в системе  $n$  заявок в момент времени  $t$ .

- Модель получена из рассуждений, что вероятность нахождения в системе  $n$  заявок в момент времени  $(t+\Delta t)$  равна вероятности нахождения в системе  $n$  заявок в момент  $t$ , умноженной на вероятность того, что за время  $\Delta t$  в систему не поступит ни одной заявки и ни одна заявка не будет обслужена, плюс вероятность нахождения в системе  $(n-1)$  заявок в момент  $t$ , умноженная на вероятность того, что за время  $\Delta t$  поступит одна заявка и ни одна заявка не будет обслужена, плюс вероятность нахождения в системе  $(n+1)$  заявок в момент  $t$ , умноженная на вероятность того, что за время  $\Delta t$  одна заявка покинет систему и не поступит ни одной заявки.

# Лекция 7. Аналитическое моделирование

## ● Пример аналитического моделирования ВС:

- Упростим уравнения, сначала пренебрежем членами порядка малости  $(\Delta t)^2$ . Тогда

$$(1-\lambda\Delta t)(1-\mu\Delta t) \approx 1-(\lambda+\mu)\Delta t;$$

$$\lambda\Delta t(1-\mu\Delta t) \approx \lambda\Delta t;$$

$$\mu\Delta t(1-\lambda\Delta t) \approx \mu\Delta t.$$

- После подстановки получим:

$$P_n(t+\Delta t) = P_n(t)[1-(\lambda+\mu)\Delta t] + P_{n-1}(t)\lambda\Delta t + P_{n+1}(t)\mu\Delta t, n \geq 1;$$

$$P_0(t+\Delta t) = P_0(t)(1-\lambda\Delta t) + P_1(t)\mu\Delta t, n=0.$$

- Следующий шаг решения данного уравнения - освобождение от  $\Delta t$  и получение дифференциальных уравнений. Для этого перенесем  $P_n(t)$  влево и устремим  $\Delta t$  к нулю. В итоге получим:

$$dP_n(t)/dt = -(\lambda+\mu)P_n(t) + \lambda P_{n-1}(t) + \mu P_{n+1}(t), n \geq 1;$$

$$dP_0(t)/dt = -\lambda P_0(t) + \mu P_1(t), n=0.$$

# Лекция 7. Аналитическое моделирование

## ● Пример аналитического моделирования ВС:

- Решение дифференциального уравнения можно выполнить для стационарного состояния  $\rho = \lambda/\mu < 1$ . Приравняв нулю производные по времени и исключив таким образом время  $t$  из уравнений, получим систему алгебраических уравнений

$$(\lambda + \mu) P_n = \lambda P_{n-1} + \mu P_{n+1}, n \geq 1;$$

$$\lambda P_0 = \mu P_1, n=0.$$

Или  $(1 + \rho) P_n = \rho P_{n-1} + P_{n+1}, n \geq 1;$

$$P_1 = \rho P_0, n=0.$$

- Далее попытаемся найти выражения для математического ожидания числа заявок, находящихся в очереди, и среднего времени ожидания в очереди.

Для этого нам необходимо выразить  $P_n$  через величины исходных данных, т. е.  $\rho$ . Пусть в первом уравнении  $n=1$ . Тогда  $(1 + \rho) P_1 = \rho P_0 + P_2$ . Подставив сюда  $P_1$  из второго уравнения, находим  $P_2 = \rho^2 P_0$ . Повторяя эти операции для следующих  $n$  получим  $P_n = \rho^n P_0$ .

# Лекция 7. Аналитическое моделирование

## ● Пример аналитического моделирования ВС:

- Далее учтем, что  $\sum P_n = 1$ , так как это сумма вероятностей того, что в системе нет ни одной заявки, имеется одна заявка, две заявки и т.д., и сумма этих вероятностей равна единице, так как рассматриваются все возможные состояния (полная группа) системы. Поэтому  $\sum \rho^n P_0 = 1$ , или  $\sum \rho^n P_0 = P_0 \sum \rho^n = P_0 / (1-\rho) = 1$ . Откуда  $P_0 = 1-\rho$ . Следовательно,  $P_0 = \rho^n (1-\rho)$ . Полученное выражение представляет собой геометрическое распределение, для него математическое ожидание числа заявок, находящихся в системе (ОА):

$$m_n = \sum n P_n = (1-\rho) \sum n \rho^n = \rho (1-\rho). \text{ Дисперсия:}$$

$$D_n = \sum (n - m_n)^2 P_n = \sum n^2 P_n - [ \rho / (1-\rho) ]^2 = m_n + m_n^2 .$$

Математическое ожидание числа заявок, находящихся в очереди:

$$L_n = \sum (n - 1) P_n = m_n - \rho = \rho / (1-\rho) - \rho = \rho^2 / (1-\rho) .$$

Среднее время ожидания заявок в очереди

$$\hat{t}_n = L_n / \lambda = \rho^2 / [\lambda(1-\rho)] .$$

# Лекция 7. Аналитическое моделирование

- Применение и ограничения аналитического моделирования:  
Аналитическое моделирование применяется для простых систем и элементов. Возможности аналитического моделирования сильно ограничены, часто требуется принятие некоторых упрощающих предположений.  
- Основные *ограничения* аналитического моделирования:
  - 1) Входные потоки заявок должны быть стационарными, ординарными и с отсутствием последствия. Такие потоки называют простейшими.
  - 2) Интервалы времени между моментами поступления заявок должны подчиняться экспоненциальному закону распределения.
  - 3) Приоритеты заявок не учитываются.
  - 4) Дисциплина обслуживания должна быть типа FIFO.
  - 5) Времена обслуживания заявок в ОА также подчиняются экспоненциальному закону.Применения:
  - 1) Для ориентировочных оценок ВС в ряде частных случаев.
  - 2) В качестве макромоделей для отдельных фрагментов ВС при имитационном моделировании.



# Лекция 7. Аналитическое моделирование

## ● Пример определения вероятности безотказной работы системы.

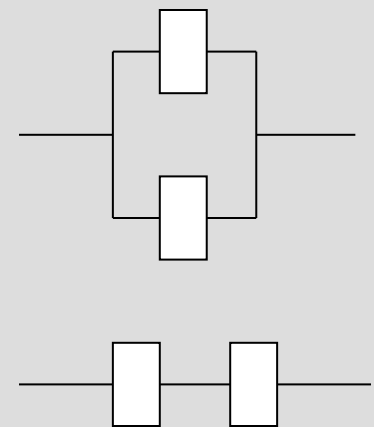
- Каждый блок системы характеризуется собственной вероятностью безотказной работы. Задача состоит в определении вероятности отказа системы за некоторое время  $t$ .

- Вероятность отказа каждого блока равна  $Q_i(t) = \int P(t)dt$ , где  $P(t)$  – плотность распределения вероятности отказов.

- Для параллельно соединенных блоков, имеющих вероятности отказов  $Q_1(t)$  и  $Q_2(t)$ , суммарная вероятность отказов определяется как

$Q_{пар}(t) = Q_1(t) \cdot Q_2(t)$ . Для последовательно соединенных блоков вероятность отказа составит

$Q_{посл}(t) = Q_1(t) + Q_2(t) - Q_1(t) \cdot Q_2(t)$ .



# Лекция 8. Сети Петри

- Содержание:
  - Общие сведения о сетях Петри
  - Примеры сетей Петри
  - Виды и свойства сетей Петри
  - Модель двухуровневой ВС

# Лекция 8. Сети Петри

- **Общие сведения о сетях Петри:**

- Теория сетей Петри стала очень популярной и предназначена для работы с дискретными параллельными и асинхронными системами.

- Основанная в начале 60-х годов немецким математиком Карлом Петри (диссертация 1962 года), в настоящее время она содержит большое количество моделей, методов и средств анализа и моделирования, имеющих обширное количество приложений практически во всех отраслях вычислительной техники и даже вне ее.

- Популярность сетей Петри вызвана удачным представлением различных типов объектов, присутствующих во многих моделируемых системах и "событийным" подходом к моделированию.

**Сети Петри (Petri net)** — в первую очередь это математический аппарат для моделирования динамических дискретных систем.

# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

- Сеть Петри кроме математических определений может быть представлена в графическом виде, в виде двудольного ориентированного графа, состоящего из вершин двух типов — позиции (places, места) и переходов (transfer), соединённых между собой дугами.
- Позиции изображаются окружностями, переходы - отрезками прямой (прямоугольником, утолщенными черточками). Дуги в сетях Петри - направленные.



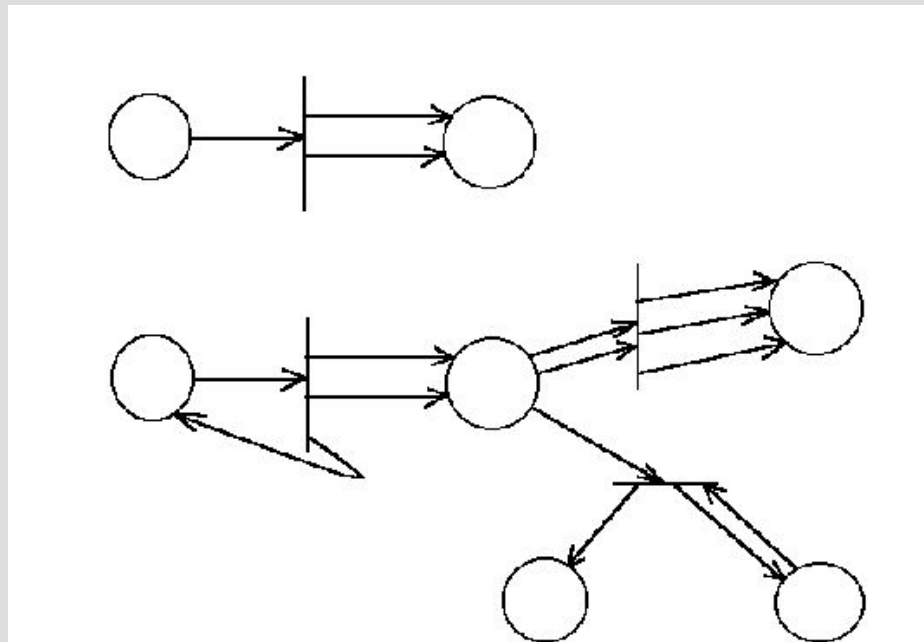
# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

Каждая дуга связывает вершины, только разных классов. Либо начало дуги совпадает с позицией и тогда конец этой дуги совпадает с переходом, либо наоборот.

Следующие рисунки иллюстрируют это:

- допустимые примеры

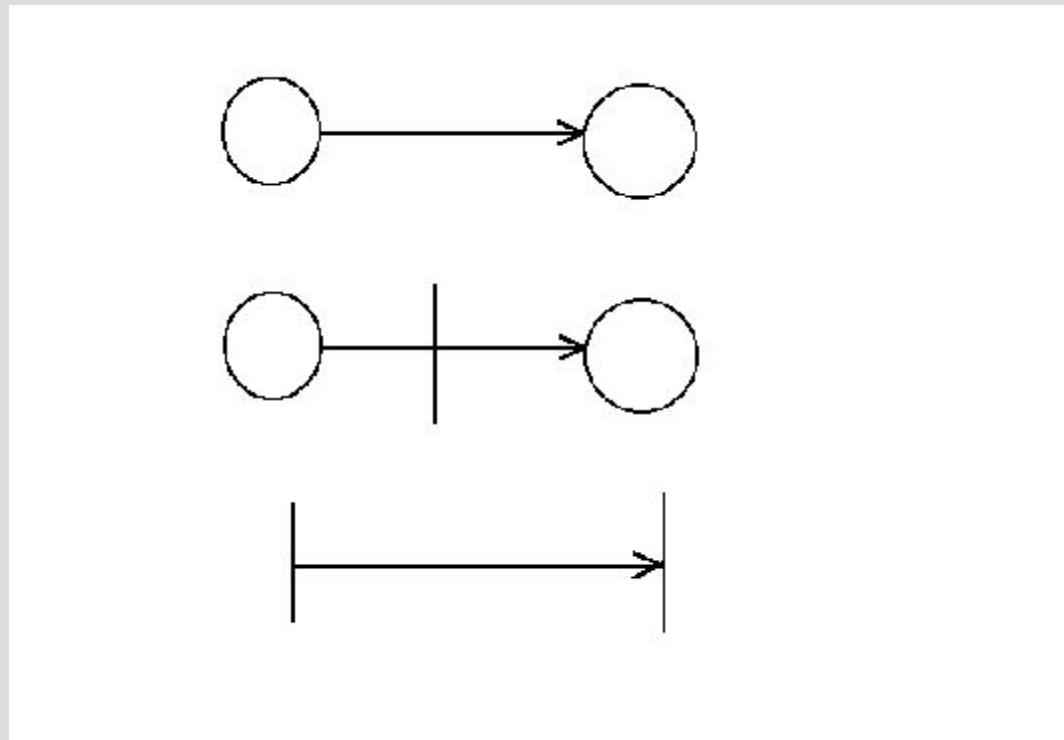




# Лекция 8. Сети Петри

- **Общие сведения о сетях Петри:**

- недопустимые примеры

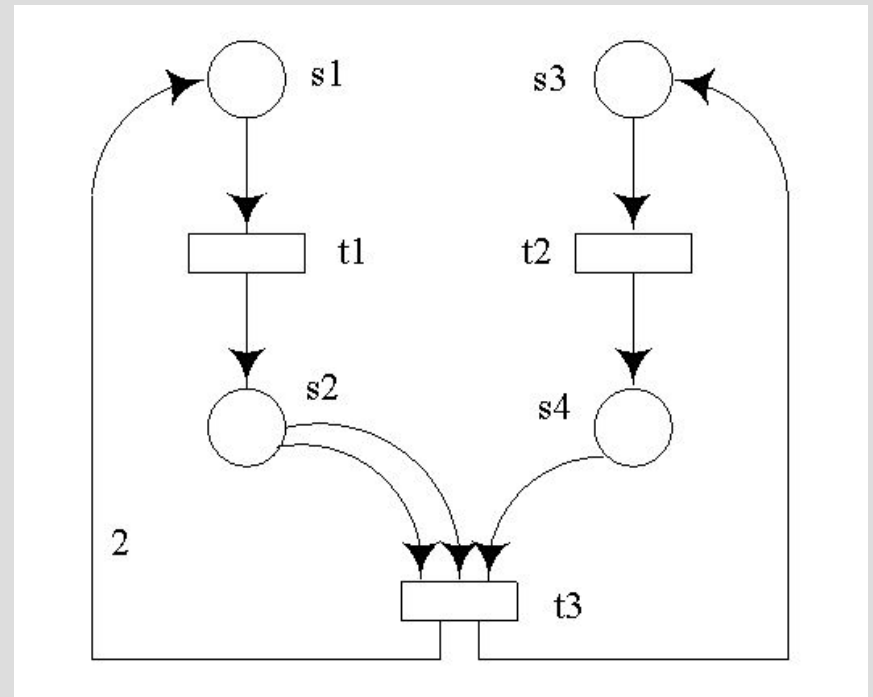


# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

Различают также:

- **входная позиция** некоторого конкретного перехода – позиция, из которой исходят дуги, входящие в данный переход (входные дуги).
- **выходная позиция** некоторого конкретного перехода - позиция, в которую входят дуги, исходящие из данного перехода (выходные дуги).
- Переходы и позиции обычно нумеруются и обозначаются:  
переходы –  $t_n$ , позиции (места) –  $p_n (s_n)$



# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

- Математически Сеть Петри представляется множеством векторов

$$S = f(P, T, I, O).$$

где  $P$  - вектор позиций,  $T$  - вектор переходов,  $I$  - вектор входных функций и  $O$  - вектор выходных функций.

- В позициях могут размещаться маркеры (метки, фишки, tokens), способные перемещаться по сети.

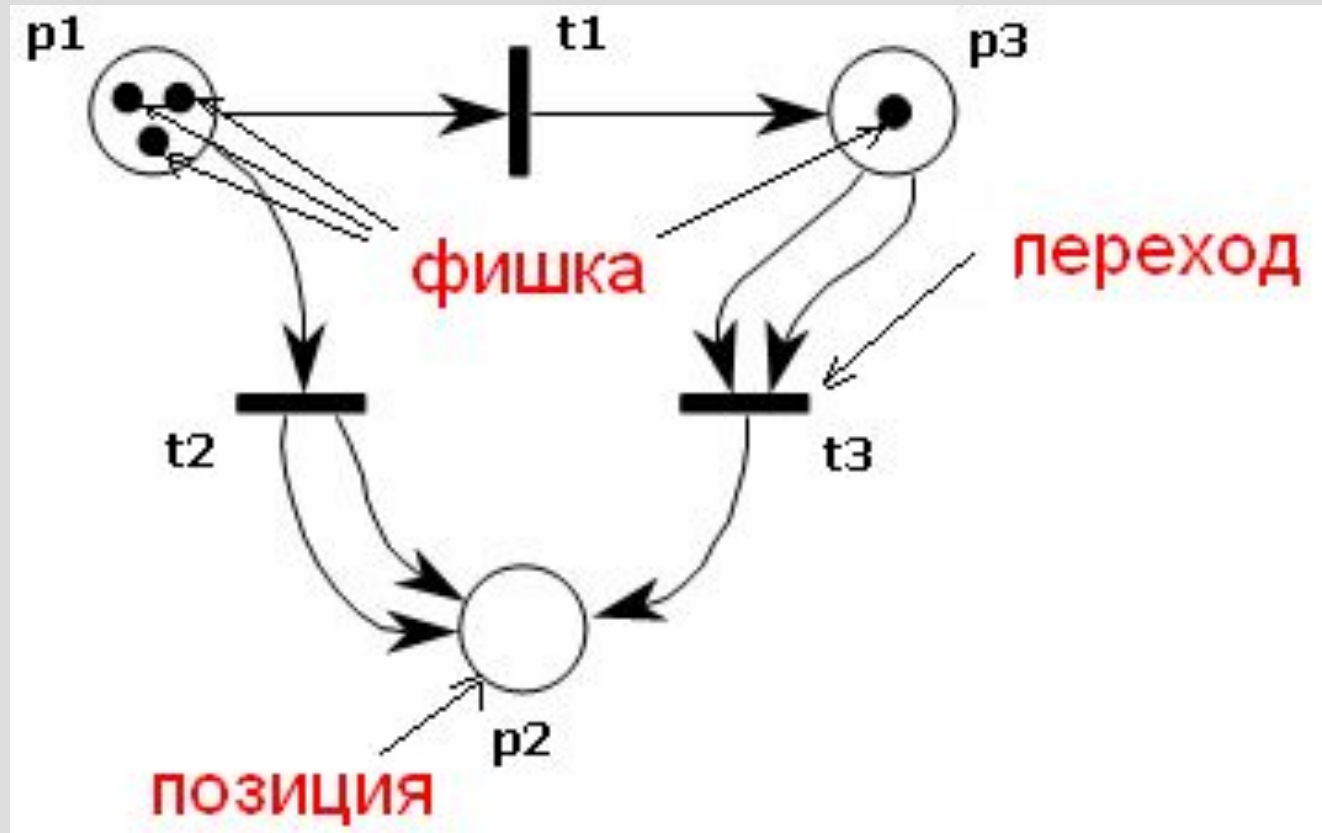
- Графически маркировка изображается в виде точек, располагающихся в кружках, соответствующих местам (позициям) сети.

- Каждой позиции сети ставится в соответствие натуральное число, указывающее количество фишек в данной позиция.

- Это число называют **разметкой позиции**, а совокупность таких чисел для всех позиций сети называют **разметкой (маркировкой) сети**. В этом случае сеть называется маркированной. Отсутствие меток в некотором месте говорит о нулевой маркировке этого места.

# Лекция 8. Сети Петри

- **Общие сведения о сетях Петри:**
  - Пример маркированной сети:



# Лекция 8. Сети Петри

- Общие сведения о сетях Петри:
  - Модель представляет собой последовательность событий.
  - Каждому возможному событию в системе соответствует определенный переход. Событие будет происходить, если будут удовлетворены определенные условия.
  - Каждому условию соответствует определенная позиция. Выполнение условий отображается с помощью маркеров (точки).
    - Число состояний сети Петри (СП) определяется числом возможных маркировок. Если позиции пронумерованы (проиндексированы), то состояние системы определяется вектором состояний (или маркировкой), например,  $P(0,1,0,0,1,\dots)$ .



# Лекция 8. Сети Петри

- Общие сведения о сетях Петри:
    - Как и в системах массового обслуживания в сетях Петри есть объекты двух типов: динамические — метки (маркеры) внутри позиций и статические — им соответствуют вершины сети Петри.
    - Каждое изменение маркировки называют событием, т.к. каждое событие связано со срабатыванием определенного перехода.
- Условие (позиция) имеет емкость:
- Условие не выполнено - емкость равна нулю
  - Условие выполнено - емкость равна 1.
  - Условие выполнено с  $n$ -кратным запасом - емкость равна  $n$ .
- Последовательность событий образует моделируемый процесс.

# Лекция 8. Сети Петри

- Общие сведения о сетях Петри:

Математические правила срабатывания переходов, конкретизируют следующим образом:

- переход срабатывает, если для каждой из его входных позиций выполняется условие  $N_i \geq K_i$ , где  $N_i$  — число маркеров в  $i$ -й входной позиции,  $K_i$  — число дуг, идущих от  $i$ -й позиции к переходу; при срабатывании перехода число маркеров в  $i$ -й входной позиции уменьшается на  $K_i$ , а в  $j$ -й выходной позиции увеличивается на  $M_j$ , где  $M_j$  — число дуг, связывающих переход с  $j$ -й позицией.

Иногда в позициях указывают не метки, а число  $N$ , где  $N$  означает емкость данной позиции. В этом случае для дуг, соединяющих позиции и переходы могут быть использованы кратные дуги, где число  $N$  называется *кратностью дуги*, которое графически изображается рядом с дугой. Дуги, имеющие единичную кратность, будут обозначаться без приписывания единицы.

# Лекция 8. Сети Петри

## ● **Общие сведения о сетях Петри:**

- Т.о. модель представляет собой последовательность событий.
- Каждому возможному событию в системе соответствует определенный переход.
- Событие будет происходить, если будут удовлетворены определенные условия.
- Каждому условию соответствует определенная позиция.
- Выполнение условий отображается с помощью маркеров.
- Число состояний сети Петри (СП) определяется числом возможных маркировок.
- Графическое изображение сети Петри удобно для восприятия и легко преобразуется в исходную программу имитационного моделирования.

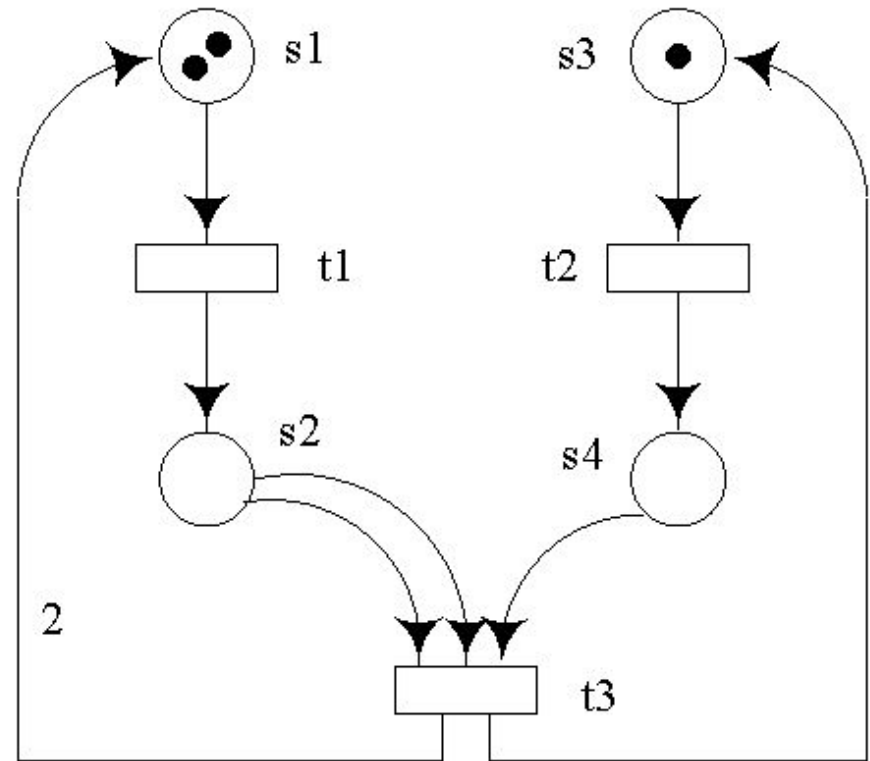
# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

Пример маркированной сети.

В начальной маркировке позиция  $S1$  имеет две метки, позиция  $S3$  - одну метку, а позиции  $S2, S4$  - ни одной метки.

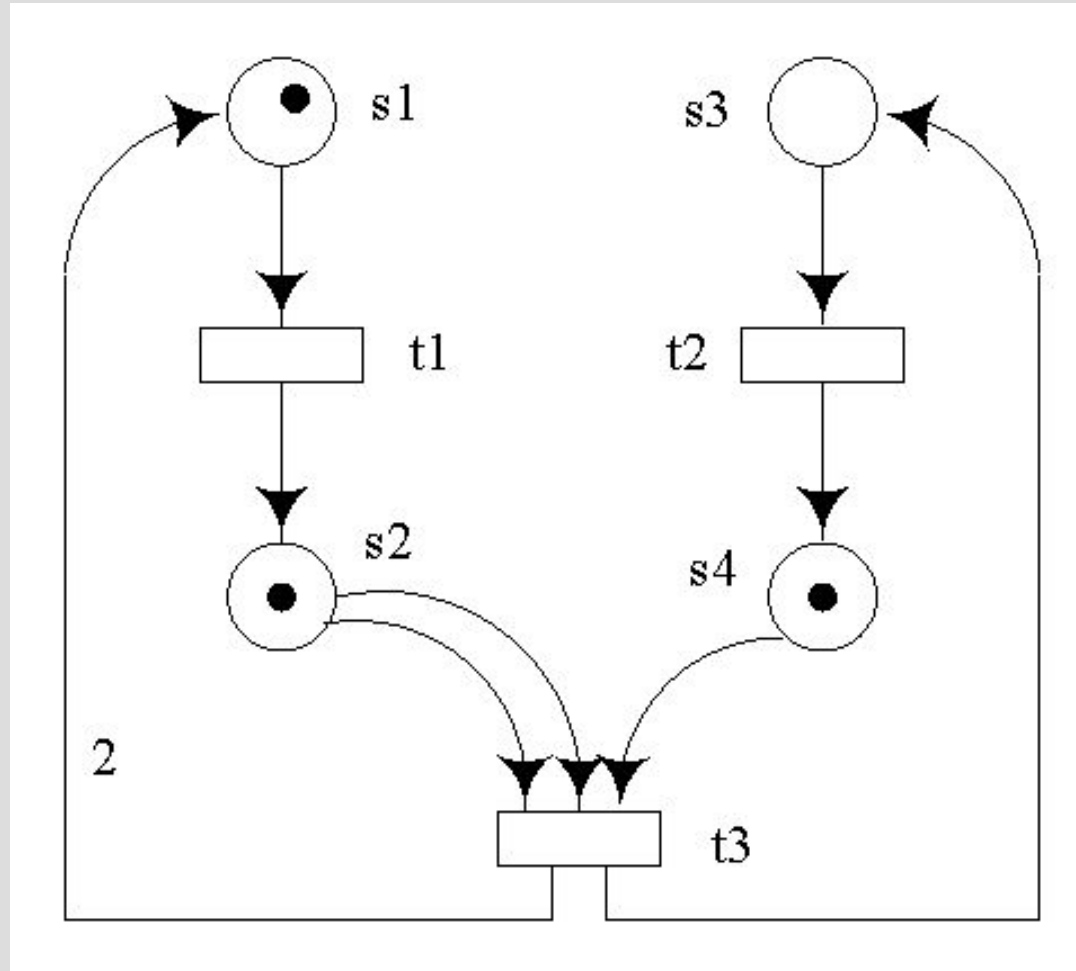
Пусть сработают переходы  $t1$  и  $t2$



# Лекция 8. Сети Петри

- **Общие сведения о сетях Петри:**

Получим:





# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

**Пример:** модель простой ВС.

- Требуется описать с помощью сети Петри работу группы пользователей на единственной рабочей станции  $WS$  при заданных характеристиках потока запросов на пользование  $WS$  и характеристиках поступающих задач.
    - В модели данного примера переходы описывают следующие события:  $t_{12}$  – (событие) появление задачи на входе  $WS$ ,  $t_{10}$  – (событие) начало решения задачи,  $t_{11}$  – (событие) окончание решения задачи,  $t_{13}$  – (событие) выход решенной задачи из  $WS$  .
- Позиция  $P_1$  соответствует условию, что во входной очереди имеется задача,  $P_2$  – (условие) решение окончено,  $P_3$  – (условие) процессор свободен,  $P_4$  – (условие) в выходной очереди имеется задача.

# Лекция 8. Сети Петри

- Общие сведения о сетях Петри:

Начальная маркировка в данном примере задана исходным вектором  $P_0 = (P_1, P_2, P_3, P_4) = (0, 0, 1, 0)$ .

Приход задачи в *WS* отображается запуском перехода  $t_{12}$  и появлением маркера в позиции  $P_1$ , вектор состояния будет иметь вид -  $(1, 0, 1, 0)$ .

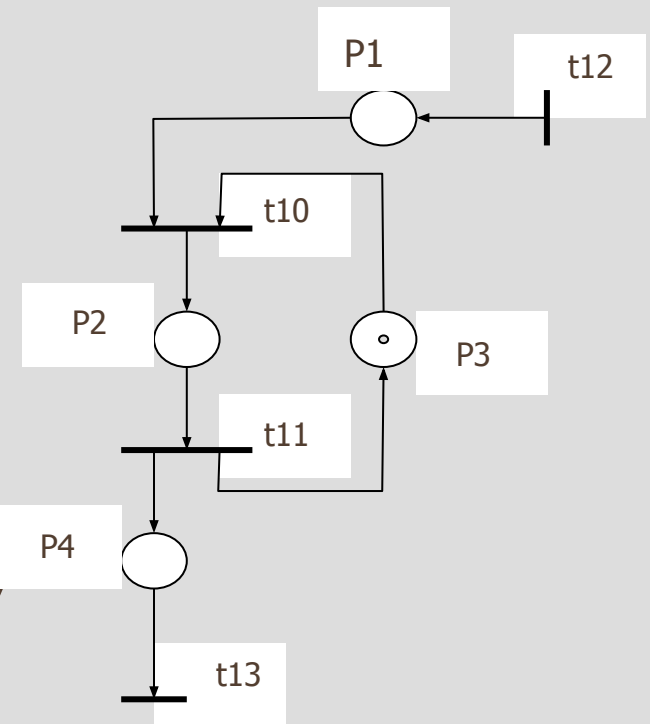
Теперь удовлетворяются условия для срабатывания перехода  $t_{10}$ , в результате удаляются маркеры из позиций  $P_1$  и  $P_3$ , помещается маркер в позицию  $P_2$ .

Тогда маркировка будет иметь вид -  $(0, 1, 0, 0)$ . Далее выполняются условия для срабатывания перехода  $t_{11}$ ,

появляется новая маркировка -  $(0, 0, 1, 1)$ .

Выполняются условия для срабатывания  $t_{13}$ ,

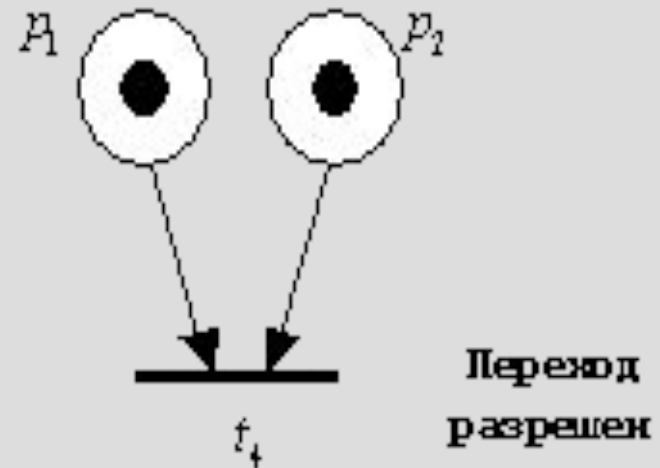
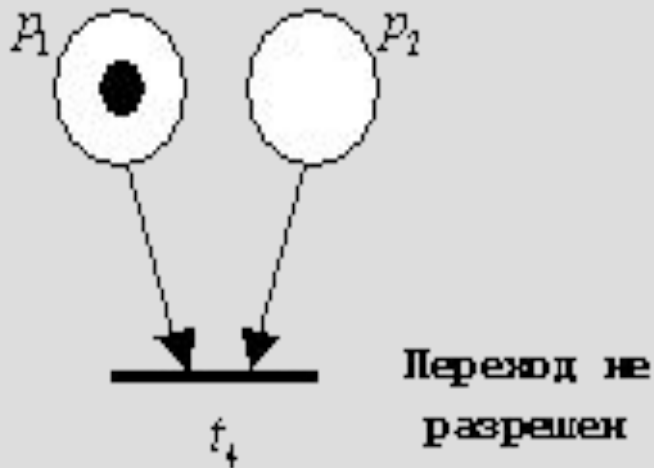
и система приходит в исходное состояние -  $(0, 0, 1, 0)$ .



# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

Переход называется разрешенным, если каждая из его входных позиций имеет число фишек по крайней мере равное числу дуг из позиции в переход. Например, если позиции  $P_1$  и  $P_2$  служат входами для перехода  $t_4$ , тогда  $t_4$  разрешен, если  $P_1$  и  $P_2$  имеют хотя бы по одной фишке:



# Лекция 8. Сети Петри

- **Общие сведения о сетях Петри:**

Для перехода  $t_7$  с входным

комплектom  $\{p_6, p_6, p_6\}$  позиция  $p_6$  должна обладать по крайней мере тремя фишками, для того чтобы  $t_7$  был разрешен:



Переход не разрешен

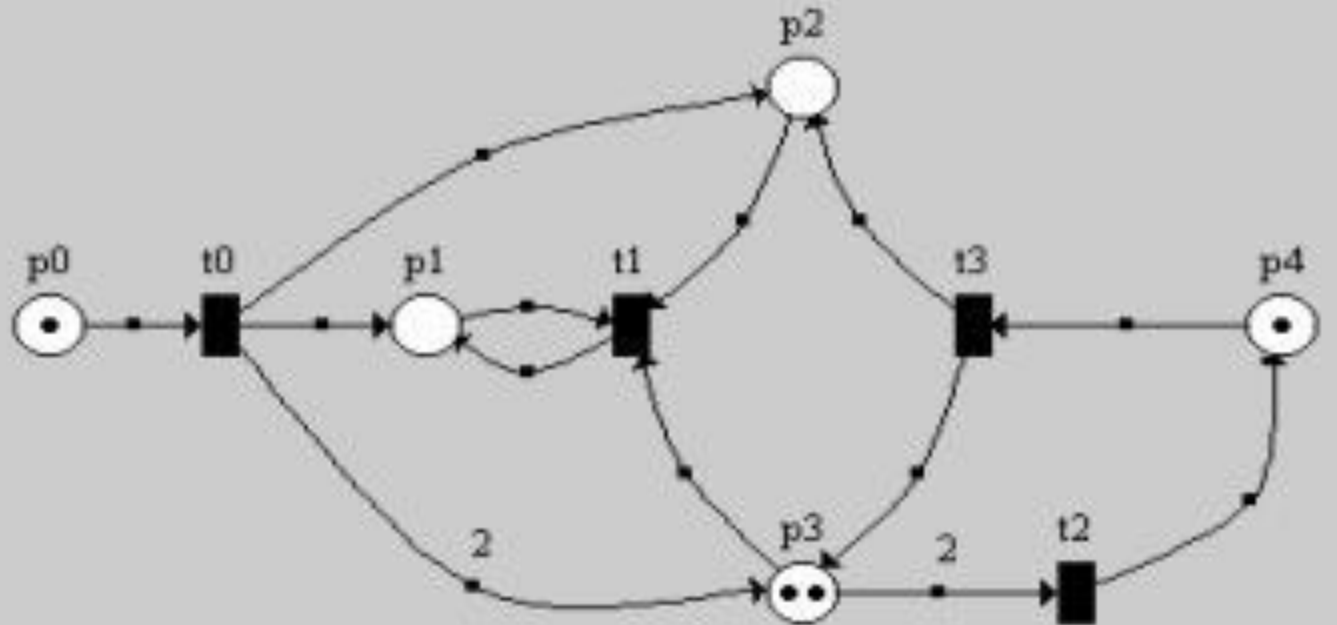


Переход разрешен

# Лекция 8. Сети Петри

## ● Общие сведения о сетях Петри:

Переход запускается удалением всех разрешающих фишек из его входных позиций и последующим помещением в каждую из его выходных позиций, по одной фишке для каждой дуги. Из приведенных ниже графических иллюстраций все сразу станет понятно:

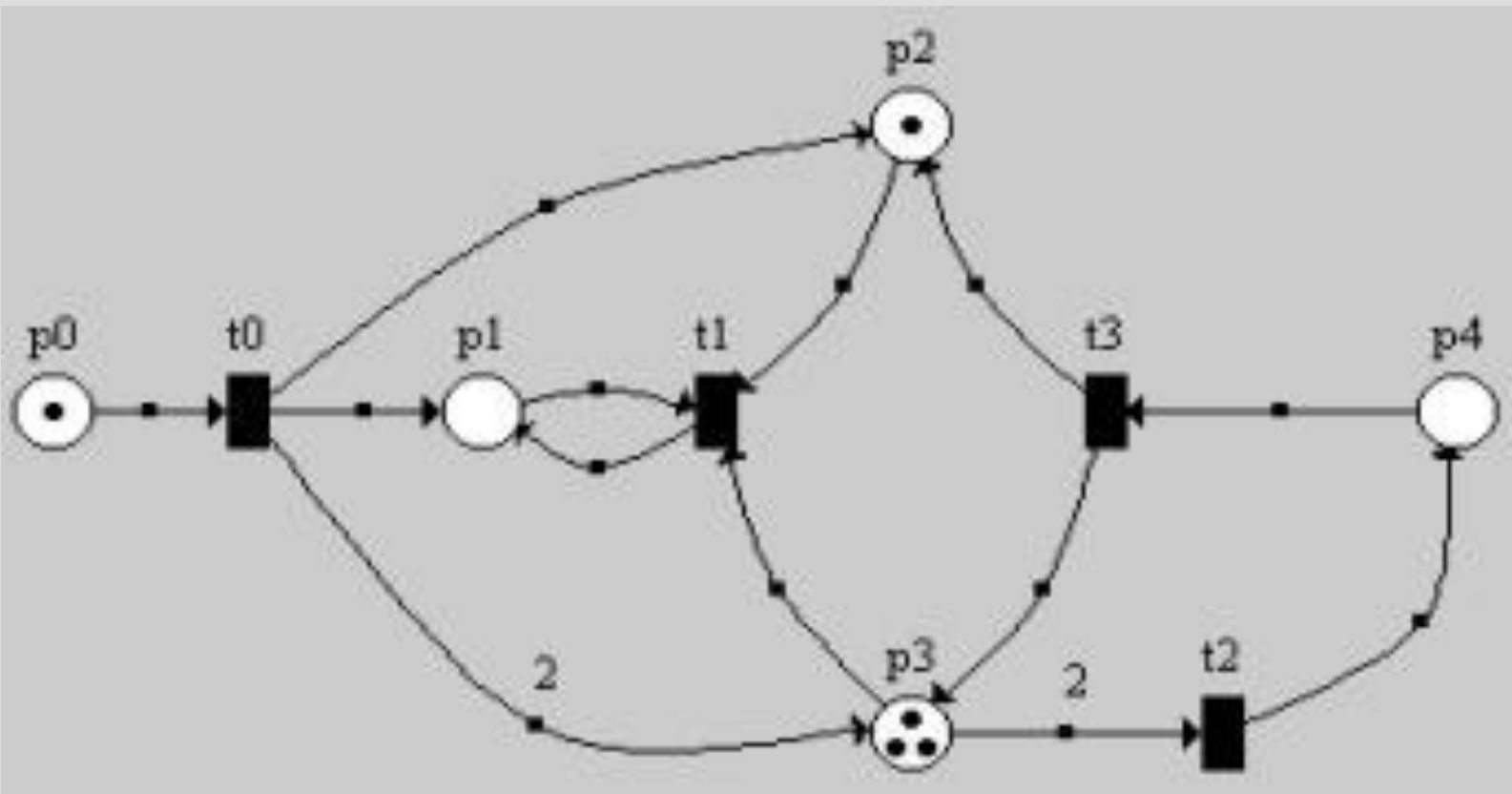


В данной сети переходы t0, t2 и t3 разрешены.



# Лекция 8. Сети Петри

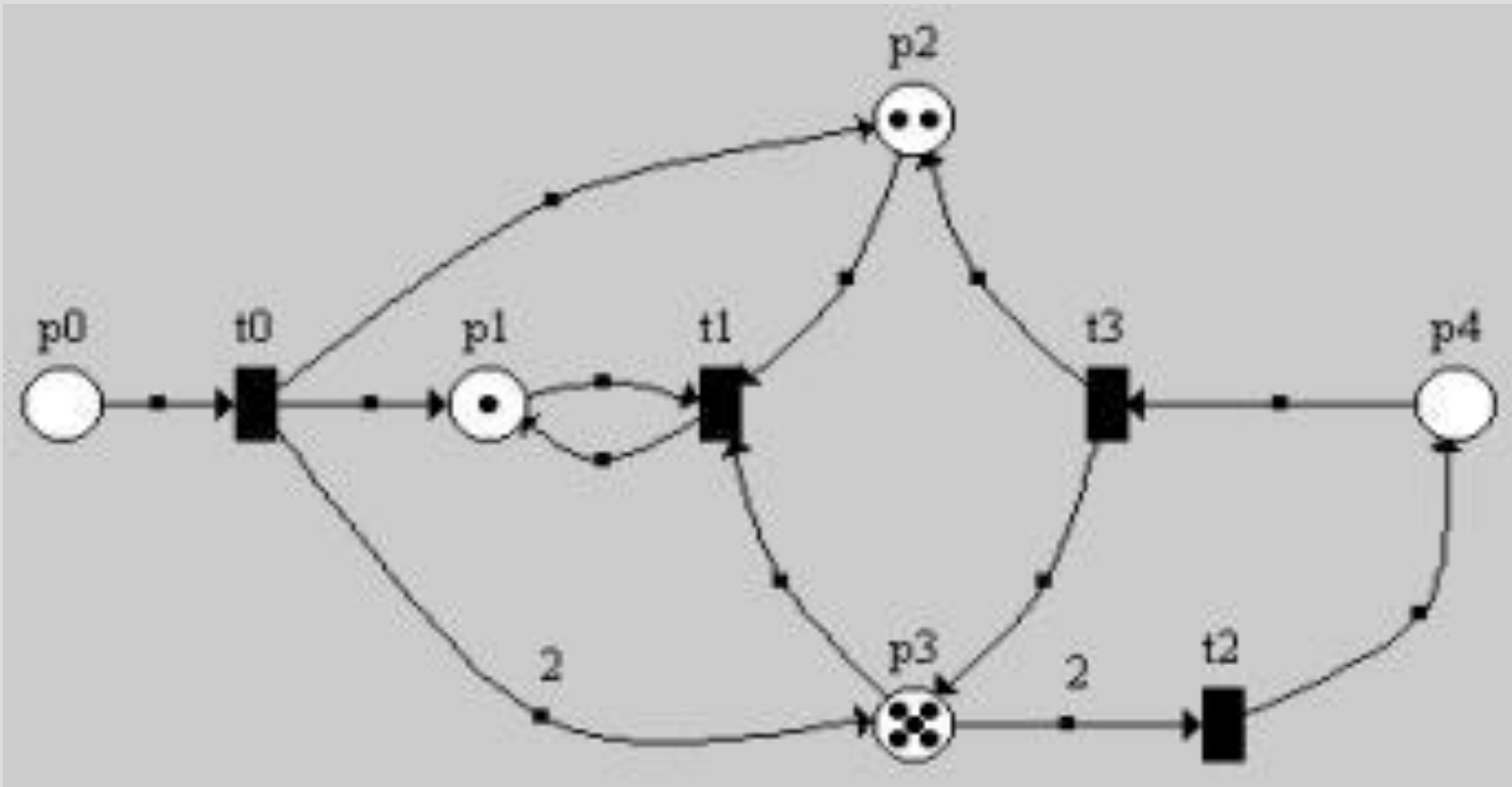
- **Общие сведения о сетях Петри:**



Маркировка, полученная в результате запуска перехода t3.

# Лекция 8. Сети Петри

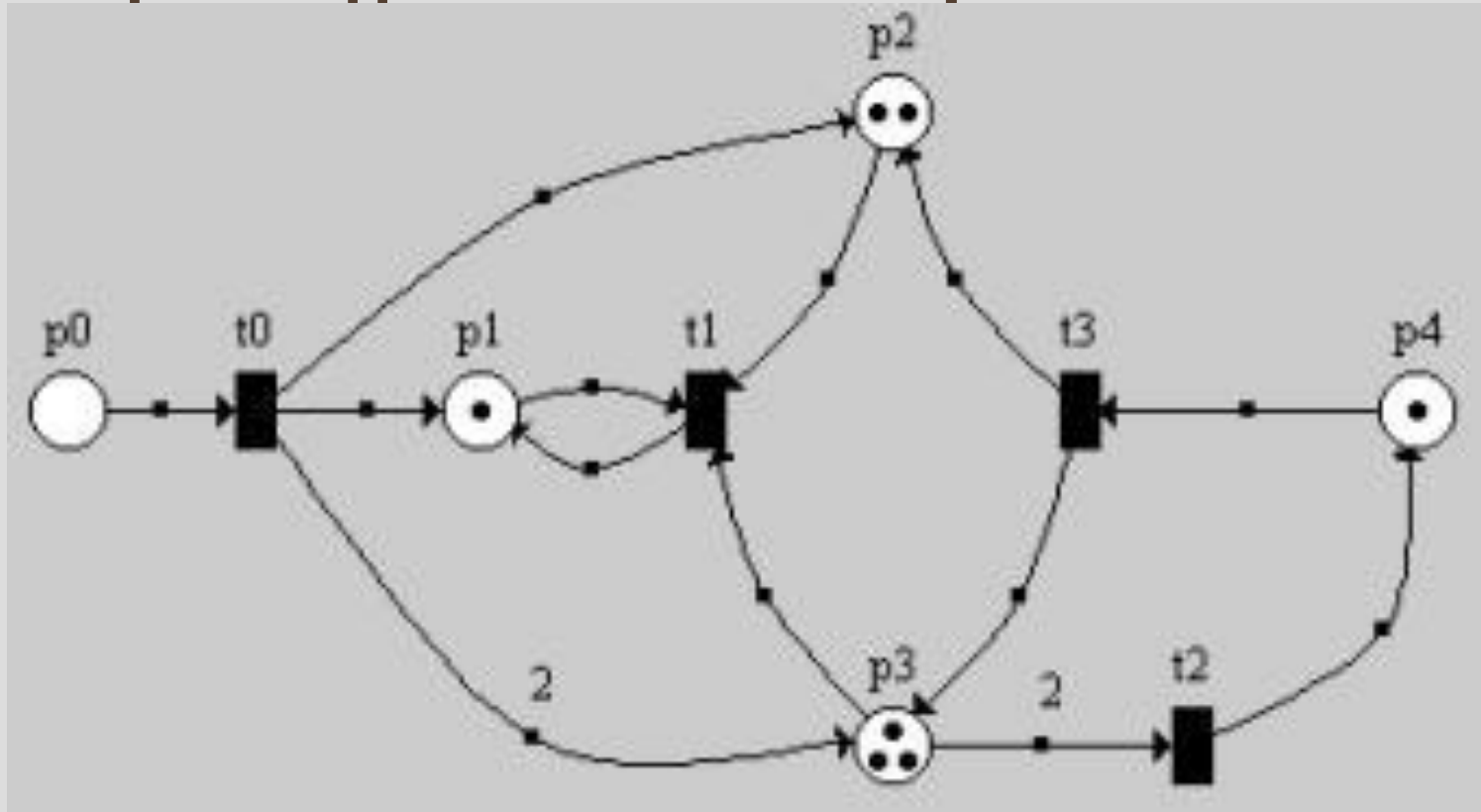
- **Общие сведения о сетях Петри:**



Маркировка, полученная в результате запуска перехода  $t_0$ . (обратите внимание - из  $t_0$  в  $p_3$  идет ДВЕ дуги!!! Поэтому в  $p_3$  добавилось ДВЕ фишки).

# Лекция 8. Сети Петри

- **Общие сведения о сетях Петри:**



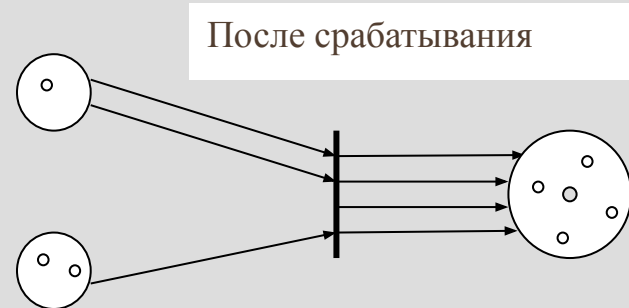
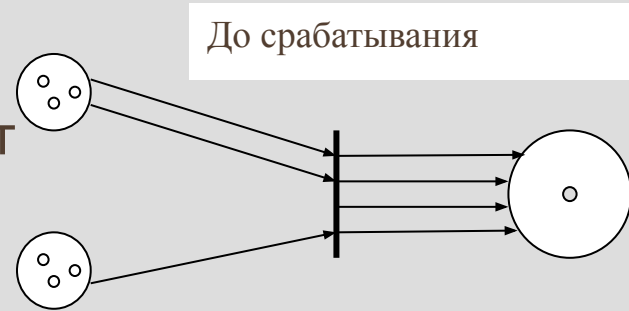
Маркировка, полученная в результате запуска перехода t2 (и снова обратите внимание на ДВЕ дуги, идущие из p3 в t2).

# Лекция 8. Сети Петри

- Общие сведения о сетях Петри:

Всю эту игру можно продолжать до тех пор, пока хотя бы один из переходов будет разрешен. Поиск тупиковых ситуаций (когда ни один из переходов не является разрешенным) – тема отдельного обсуждения.

В общем случае в позиции может быть более одного маркера. Рис. Иллюстрирует правила срабатывания переходов при наличии нескольких маркеров.



# Лекция 8. Сети Петри

- Виды сетей Петри:

Можно вводить ряд дополнительных правил и условий в алгоритмы моделирования, получая ту или иную разновидность сетей Петри.

Некоторые виды сетей Петри:

1. Временная сеть Петри
2. Стохастическая сеть Петри
3. Функциональная сеть Петри
4. Цветная сеть Петри
5. Ингибиторная сеть Петри
6. Иерархическая сеть
7. WF-сети
8. Сети с приоритетами.



# Лекция 8. Сети Петри

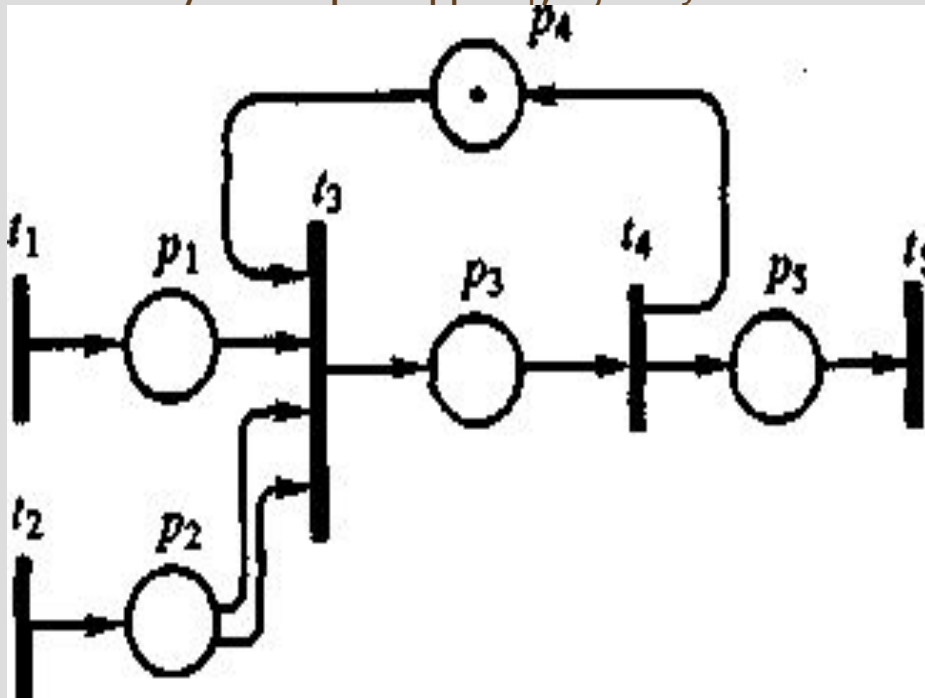
- Виды сетей Петри:

1. Так, прежде всего полезно ввести модельное время, чтобы моделировать не только последовательность событий, но и их привязку ко времени. Это осуществляется приданием переходам веса — продолжительности (задержки) срабатывания, которую можно определять, используя задаваемый при этом алгоритм. Полученную модель называют ***временной сетью Петри***.

# Лекция 8. Сети Петри

- Виды сетей Петри:

Пример. Требуется описать с помощью сети Петри функционирование системы из предприятий *A*, *B* и *C*. Предприятия *A* и *B* поставляют узлы *X1* и *X2* соответственно, а на предприятии *C* происходит сборка, в каждый сборочный узел входит один узел *X1* и два узла *X2*. На рис. предприятиям *A*, *B* и *C* соответствуют переходы  $t_1$ ,  $t_2$  и  $t_3$ .



# Лекция 8. Сети Петри

- Виды сетей Петри:

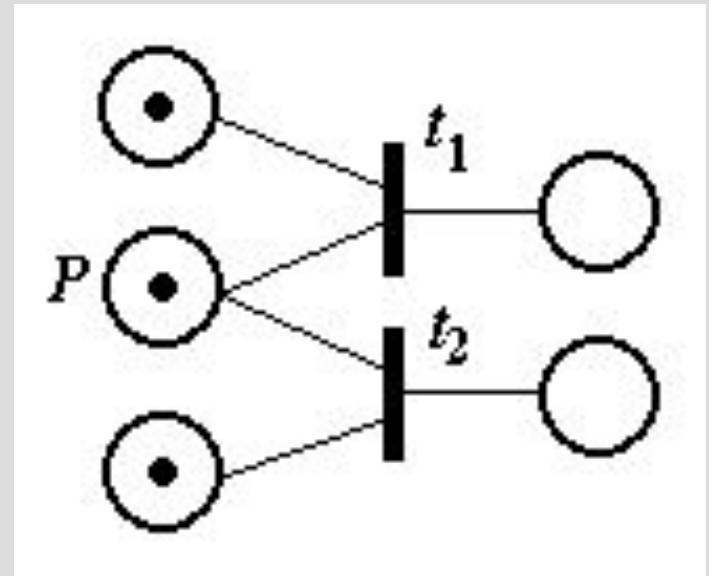
Срабатывание перехода  $t_3$  происходит только в том случае, если, во-первых, в позиции  $p_1$  имеется метка, а в позиции  $p_2$  - не менее двух меток, что означает поступление от предприятия  $A$  и  $B$  соответствующих комплектующих, и, во-вторых, имеется метка в позиции  $p_4$ , что означает, что предприятие  $C$  закончило сборку предыдущего изделия и готово приступить к сборке следующего. Пока очередное изделие не будет собрано, метки в  $p_4$  не будет, следовательно, запросы, пришедшие во входные позиции  $p_1$  и  $p_2$ , вынуждены ожидать срабатывания перехода  $t_4$ . Переходам  $t_1$ ,  $t_2$  и  $t_3$  поставлены в соответствие процедуры вычисления задержек срабатывания. Задержки в первых двух переходах равны интервалам времени между появлениями готовых узлов, задержка в  $t_3$  равна времени сборки изделия.

# Лекция 8. Сети Петри

- Виды сетей Петри:

2. Если задержки являются случайными величинами, то сеть называют **стохастической сетью Петри**. В стохастических сетях возможно введение вероятностей срабатывания возбужденных переходов. Так, на рис. представлен фрагмент сети Петри, иллюстрирующий конфликтную ситуацию — маркер в позиции  $P$  может запустить либо переход  $t_1$ , либо переход  $t_2$ .

В стохастической сети предусматривается вероятностный выбор срабатывающего перехода в таких ситуациях.

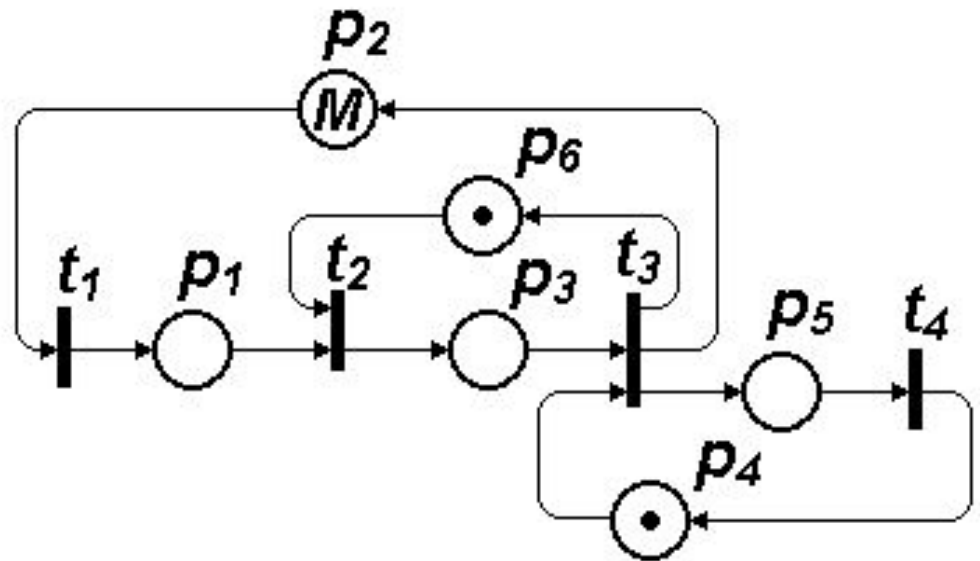


# Лекция 8. Сети Петри

- Виды сетей Петри:

**Пример.** Требуется описать с помощью сети Петри процессы возникновения и устранения неисправностей в некоторой технической системе, состоящей из множества однотипных блоков; в запасе имеется один исправный блок; известны статистические данные об интенсивностях возникновения отказов и длительностях таких операций, как поиск неисправностей, замена и ремонт отказавшего блока.

Поиск и замену отказавшего блока производит одна бригада, а ремонт замененного блока — другая бригада. Отметим, что при числе меток в позиции, равном  $M$ , можно в ней не ставить  $M$  точек, а записать в позиции значение  $M$ .





# Лекция 8. Сети Петри

- Виды сетей Петри:

В нашем примере значение  $M$  в позиции  $P2$  соответствует числу имеющихся в системе блоков. Переходы отображают следующие события:  $t1$ — отказ блока,  $t2$  — поиск неисправного блока,  $t3$  — его замена,  $t4$ — окончание ремонта.

Очевидно, что при непустой позиции  $P2$  переход  $t1$  срабатывает, но с задержкой, равной вычисленному случайному значению моделируемого отрезка времени между отказами. После выхода маркера из  $t1$  он попадает через  $P1$  в  $t2$ , если имеется метка в позиции  $P6$ , это означает, что обслуживающая систему бригада специалистов свободна и может приступить к поиску возникшей неисправности. В переходе  $t2$  метка задерживается на время, равное случайному значению длительности поиска неисправности. Далее маркер оказывается в  $P3$  и, если имеется запасной блок (маркер в  $P4$ ), то запускается переход  $t3$ , из которого маркеры выйдут в  $P2$ ,  $P5$  и  $P6$  через отрезок времени, требуемый для замены блока. После этого в  $t4$  имитируется восстановление неисправного блока.

# Лекция 8. Сети Петри

- Виды сетей Петри:

3. Если задержки определяются как функции некоторых аргументов, которыми могут быть количества маркеров в каких-либо позициях, состояния некоторых переходов и т.п., то имеем **функциональную сеть Петри**. *Функциональная СП* характеризуется тем, что отражает не только последовательность событий, но и процессы обработки некоторых потоков данных. Для этого в описание каждого перехода добавляется алгоритм обработки данных.

# Лекция 8. Сети Петри

- Виды сетей Петри:

4. Во многих задачах динамические объекты могут быть нескольких типов, и для каждого типа нужно вводить свои алгоритмы поведения в сети. В этом случае каждый маркер должен иметь хотя бы один параметр, обозначающий тип маркера. Такой параметр обычно называют цветом; цвет можно использовать как аргумент в функциональных сетях. Сеть при этом называют ***цветной сетью Петри***.

Понятие цветности в них тесно связано с понятиями переменных, типов данных, условий и других конструкций, более приближенных к языкам программирования. К примеру, цветные сети Петри дополняются введением типов фишек и помогают избежать многократного дублирования фрагментов сети в сложных системах.

# Лекция 8. Сети Петри

- Виды сетей Петри:

5. Среди других разновидностей сетей Петри следует упомянуть *ингибиторные сети Петри*, характеризующиеся тем, что в них возможны запрещающие (ингибиторные) дуги.

Наличие маркера во входной позиции, связанной с переходом ингибиторной дугой, означает запрещение срабатывания перехода.

# Лекция 8. Сети Петри

## ● Виды сетей Петри:

**6. Иерархическая сеть** — содержит переходы, в которые вложены другие, возможно, также иерархические, сети. Срабатывание такого перехода характеризует выполнение полного жизненного цикла вложенной сети.

Для наглядности такие переходы часто обозначаются большим прямоугольником. Такой переход начинает выполнение с изъятия фишек из входных позиций и завершает, соответственно, помещая фишки в выходные. Основное отличие от простого перехода заключается в том, что между этими двумя моментами могут срабатывать другие переходы.

В момент активации составного перехода из его входных позиций извлекаются фишки, после чего вложенная сеть начинает функционирование исходя из своей начальной разметки. Когда после очередного срабатывания внутреннего перехода вложенной сети в ней не остается разрешенных переходов, она прекращает работу. В этот момент соответствующий составной переход помещает фишки в выходные позиции и переходит в пассивное состояние. При следующей активации составного перехода вложенная сеть снова размечается в соответствии со своей начальной разметкой и начинает новый жизненный цикл.



# Лекция 8. Сети Петри

- **Виды сетей Петри:**

7. **WF-сети** – это подкласс сетей Петри, называемые также сетями потоков работ, применяют для моделирования потоков работ в WorkFlow системах.
8. **Сети Петри с приоритетами** добавляют к разрешенным переходам приоритеты и тем самым позволяют снизить недетерминированность срабатываний, ограничивая множество разрешенных переходов группой переходов с наивысшим приоритетом.

# Лекция 8. Сети Петри

- **Анализ сетей Петри:**

Анализ сложных систем на базе сетей Петри можно выполнять посредством имитационного моделирования, представленных моделями сетей Петри. При этом задают входные потоки заявок и определяют соответствующую реакцию системы.

Выходные параметры СМО рассчитывают путем обработки накопленного при моделировании статистического материала.

Возможен и другой подход к использованию сетей Петри для анализа объектов, исследуемых на системном уровне. Он не связан с имитацией процессов и основан на исследовании системных характеристик и свойств поведения модели, которые характеризуют системные (общие) свойства, не связанные с конкретными внешними воздействиями. Многие такие общие свойства систем могут быть легко получены по СП.

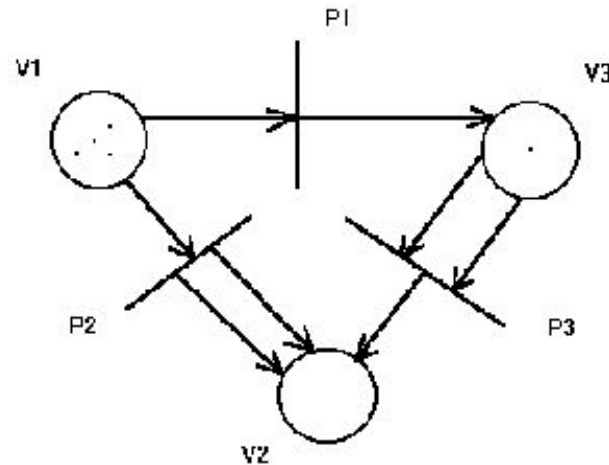
# Лекция 8. Сети Петри

## ● Анализ сетей Петри:

Например, если срабатывает переход P1, то изымается из позиции V1 одна фишка и увеличивается количество фишек в позиции V3 на одну.

Если срабатывает переход P2, то изымается одна фишка из позиции V1 и помещается в позицию V2 две фишки. Переход срабатывает, если количество фишек в каждой входной позиции перехода не меньше количества дуг, соединяющих эту позицию с переходом.

Например, переход P3 не может сработать, ибо в позиции V3 находится только одна фишка, а дуг, связывающих V3 и P3 - две. В самом деле эта сеть, в которой переход P3 не может сработать. Но если сработает переход P1, то количество фишек в позиции V3 – увеличится. Теперь P3 - сработает



# Лекция 8. Сети Петри

- **Анализ сетей Петри:**

Разметка сети меняется постоянно. Возможно, что в результате этих изменений некоторый переход потеряет возможность срабатывать, или наоборот - приобретет ее. Последовательное срабатывание переходов и соответствующее изменение разметки сети называют процессом функционирования сети.

Собственно говоря, предметом теоретического исследования сетей Петри и является процесс их функционирования, т.е. возможные последовательности срабатывания переходов и свойства получаемых при этом разметок сети. И, как обычно в математике, такие исследования формулируются, как правило, в виде утверждений двух основных типов - утверждение о существовании и утверждения об обязательности.

# Лекция 8. Сети Петри

- **Анализ сетей Петри:**

## **Конечные разметки сети**

Одна из основных проблем в теории сетей Петри - задача о конечности функционирования сети (системы) (о достижении тупиковой разметки и т.д.).

Суть проблемы состоит в ответе на вопрос для данной конкретной сети - существует ли такая последовательность срабатывания переходов, которая приводит сеть к тупиковой разметке (т.е. разметке, при которой ни один переход не может сработать)?

На предыдущем рис. видно, что последовательность  $P2, P2, P2, P2$  (т.е. четыре подряд срабатывания перехода  $P2$ ) делают дальнейшее срабатывание любого перехода в данной сети – невозможным.

Более того, анализ сети позволяет утверждать, что эта сеть всегда приходит к тупиковой разметке. Это математическое утверждение (теорема!) может быть строго доказано.



# Лекция 8. Сети Петри

- **Анализ сетей Петри:**

Свойство достижения конечной разметки присуще далеко не всем сетям.

Например, на рис.4 приведен пример сети всегда приходящей к тупиковой разметке

на рис.5 - сеть никогда

не "попадает в тупик",

на рис. 6 - сеть, которая

может остановиться, а

может и нет.

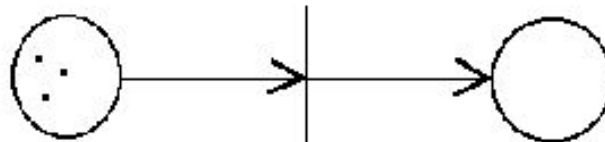


Рис.4

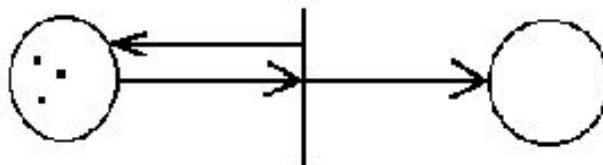


Рис. 5

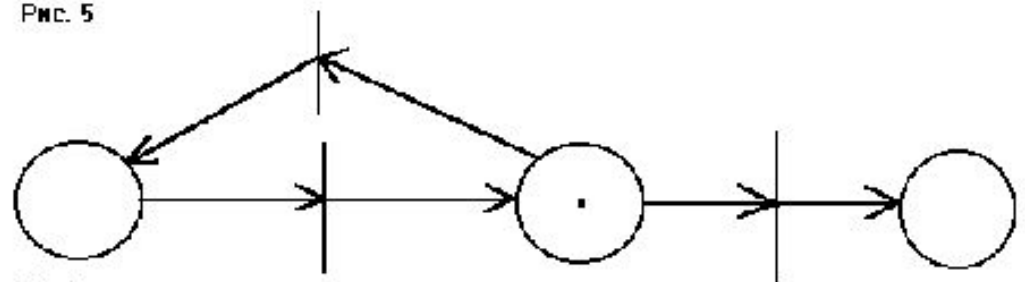


Рис.6

# Лекция 8. Сети Петри

- **Анализ сетей Петри:**

Другое направление исследования функционирования сети Петри связано с изменением количества фишек в конкретной или произвольной позиции в процессе функционирования сети.

Под ограниченностью понимают свойство сети не допускать превышения количества фишек в конкретной или произвольной позиции некоторого

фиксированного числа.

Например, сеть на рис.7 является ограниченной при любом срабатывании сети количество фишек в любой позиции не превысит 1. На рис.8 нет Тупиков, но не является ограниченной.

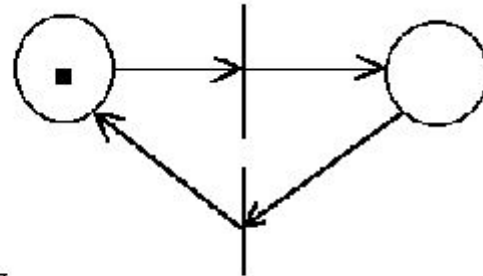


Рис.7

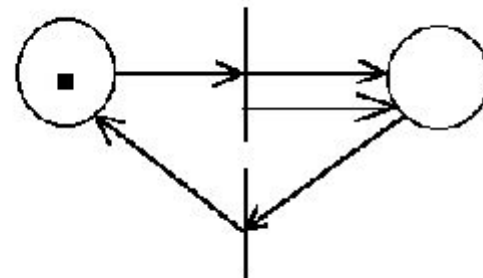


Рис.8

# Лекция 8. Сети Петри

- **Свойства сетей Петри:**

Основными свойствами сети Петри являются:

1. Ограниченность или K-ограниченность
2. Безопасность
3. Сохраняемость
4. Достижимость
5. Живость

# Лекция 8. Сети Петри

- Свойства сетей Петри:

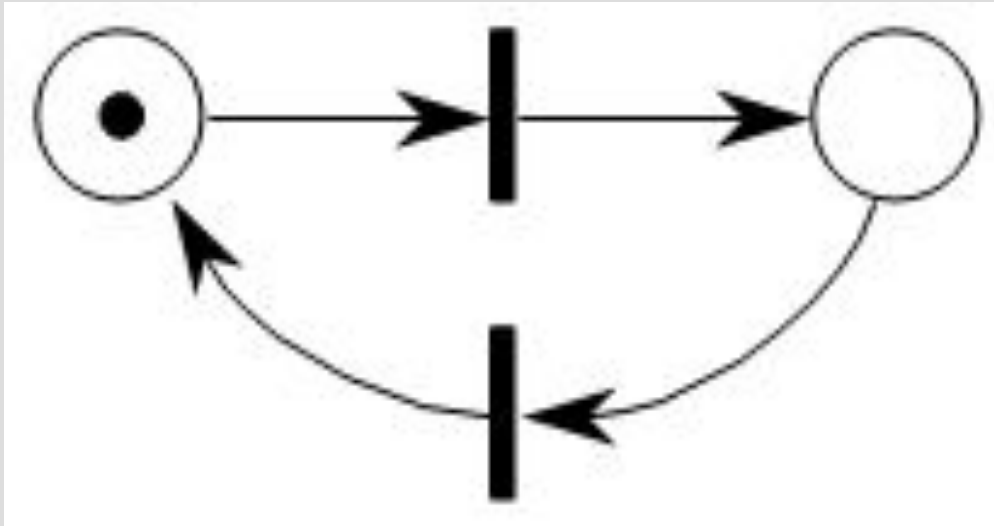
**1. Ограниченность (или  $K$ -ограниченность)** имеет место, если число меток в любой позиции сети не может превысить значения  $K$ . При проектировании систем определение  $K$  позволяет обоснованно выбирать емкости накопителей. Возможность неограниченного роста числа меток свидетельствует об опасности неограниченного роста длин очередей.

Сеть Петри ограничена, если ограничены все ее позиции.

# Лекция 8. Сети Петри

- **Свойства сетей Петри:**

Если известны емкости всех позиций, и наибольшая из них  $k_{max}$ , то сеть называют  **$k_{max}$ -ограниченной**.

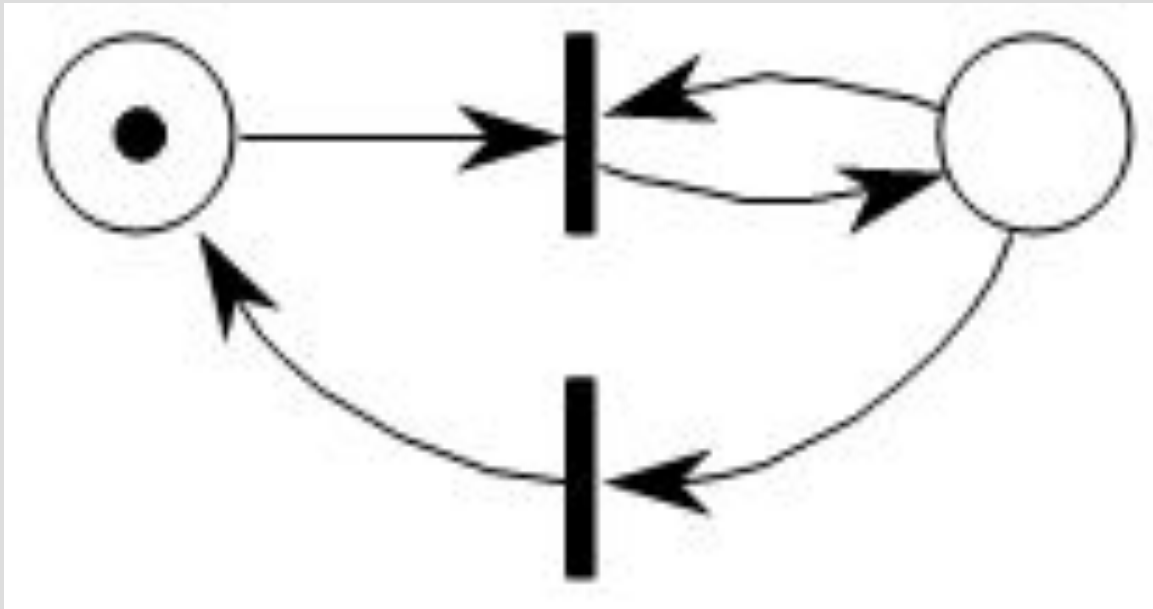


Ограниченная сеть



# Лекция 8. Сети Петри

- Свойства сетей Петри:



Не ограниченная сеть

# Лекция 8. Сети Петри

- **Свойства сетей Петри:**

2. **Безопасность** — частный случай ограниченности, а именно это 1-ограниченность. Если для некоторой позиции установлено, что она безопасна, то ее можно представлять одним триггером.

# Лекция 8. Сети Петри

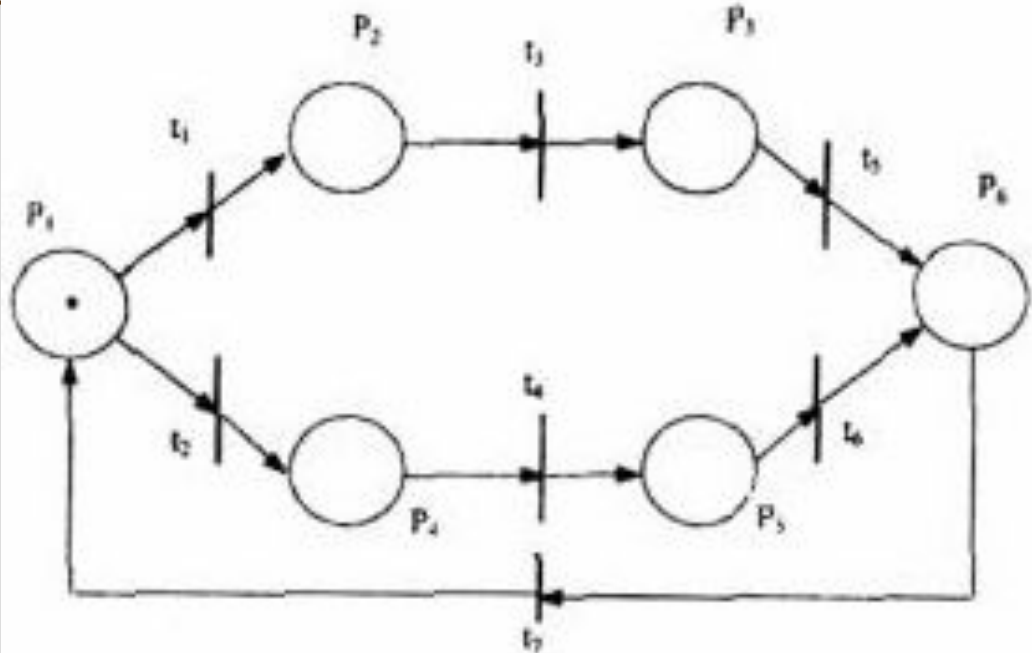
- Свойства сетей Петри:

3. **Сохраняемость** характеризуется постоянством загрузки ресурсов, т. е.

$$\sum_j A_{ij} N_i = \text{const},$$

где  $N_i$  — число маркеров в  $i$ -й позиции  $A_{ij}$  — весовой коэффициент.

Сеть Петри называют **сохраняющей**, если число циркулирующих в ней объектов постоянно.



# Лекция 8. Сети Петри

- **Свойства сетей Петри:**

Переход **сети Петри** называют **тупиковым**, если в процессе функционирования сеть может оказаться в состоянии, в котором этот переход заблокирован. Нетупиковый переход называют **активным**.

активность уровня 0, если он никогда не может быть активирован (пассивный переход);

активность уровня 1, если существует состояние (достижимое из начального), в котором он активирован;

активность уровня 2, если для всякого целого  $n$  существует последовательность срабатывания переходов, в которой данный переход присутствует по крайней мере  $n$  раз;

активность уровня 3, если существует бесконечная последовательность срабатывания переходов, в которой данный переход присутствует неограниченно часто;

активность уровня 4, если для любого достижимого состояния существует последовательность срабатываний, приводящая в такое состояние, в котором этот переход активирован (активный переход).

# Лекция 8. Сети Петри

- Свойства сетей Петри:

4. **Достижимость.** Состояние  $S$  достижимо в сети Петри, если существует цепочка срабатываний переходов, ведущая из начального состояния в  $S$ . Состояние  $S'=(P1' \dots Pn')$  покрывает состояние  $S''=(P1'' \dots Pn'')$ , если для каждого  $i=1, \dots, n$  имеет место  $Pi' \geq Pi''$ , т.е. имеет место  $S' \geq S''$ .

**Задача достижимости** (покрываемость) состоит в том, чтобы для данной сети и состояния  $S$  определить достижимо ли  $S$  (достижимо состояние, покрывающее  $S$ ). Задачи достижимости и покрываемости можно рассматривать применительно к набору не всех, а лишь некоторых позиций, т.е. к подсостояниям сети Петри.



# Лекция 8. Сети Петри

- Свойства сетей Петри:

4. *Достижимость*  $\mathbf{M}_k \rightarrow \mathbf{M}_j$  характеризуется возможностью достижения маркировки  $\mathbf{M}_j$  из состояния сети, характеризуемого маркировкой  $\mathbf{M}_k$ .

*Достижимость* – позволяет определить множество маркировок, в которые возможны переходы в процессе функционирования системы.

# Лекция 8. Сети Петри

- Свойства сетей Петри:

5. **Живость сети Петри** определяется возможностью срабатывания любого перехода при функционировании моделируемого объекта. Отсутствие живости означает либо избыточность аппаратуры в проектируемой системе, либо свидетельствует о возможности возникновения зацикливаний, тупиков, блокировок.

**Живость** – это свойство системы, означающее, что из любого состояния, достижимого из начального, возможен переход в любое другое достижимое состояние.

# Лекция 8. Сети Петри

- **Свойства сетей Петри:**

В основе исследования перечисленных свойств сетей Петри лежит анализ достижимости.

Один из методов анализа достижимости любой маркировки из состояния  $M_0$  — построение графа достижимости.

Начальная вершина графа отображает  $M_0$ , а остальные вершины соответствуют остальным маркировкам. Дуга из  $M_i$  в  $M_j$  означает событие  $M_i \rightarrow M_j$  и соответствует срабатыванию перехода.

В сложных сетях граф может содержать чрезмерно большое число вершин и дуг. Однако при построении графа можно не отображать все вершины, так как многие из них являются дублями.

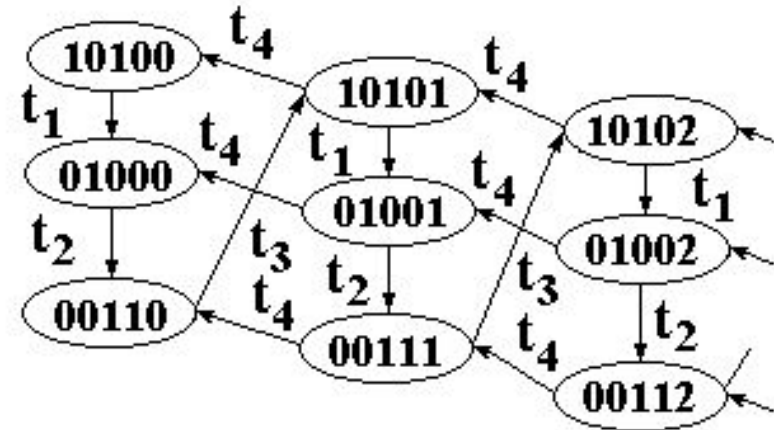
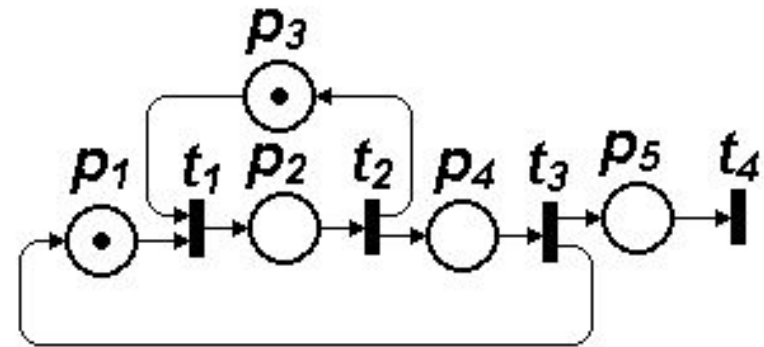
Тупики обнаруживаются по отсутствию разрешенных переходов из какой-либо вершины, т.е. по наличию листьев — терминальных вершин. Неограниченный рост числа маркеров в какой-либо позиции свидетельствует о нарушениях ограниченности.

# Лекция 8. Сети Петри

- Свойства сетей Петри:

## Пример 1

На рисунке вершины графа изображены в виде маркировок. дуги помечены срабатывающими переходами. Сеть является неограниченной и живой, так как метки могут накапливаться в позиции  $p_5$ , срабатывают все переходы, тупики отсутствуют.

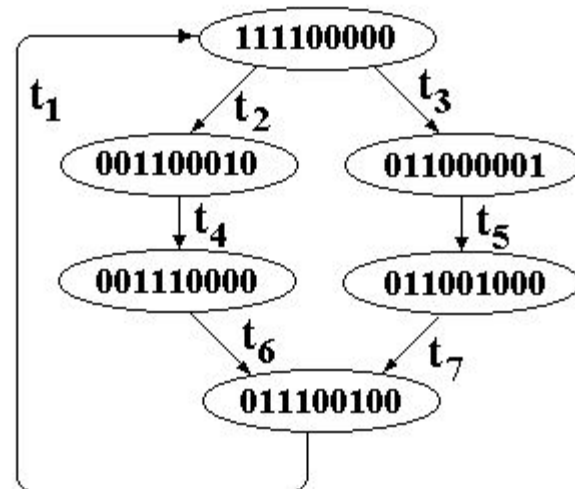
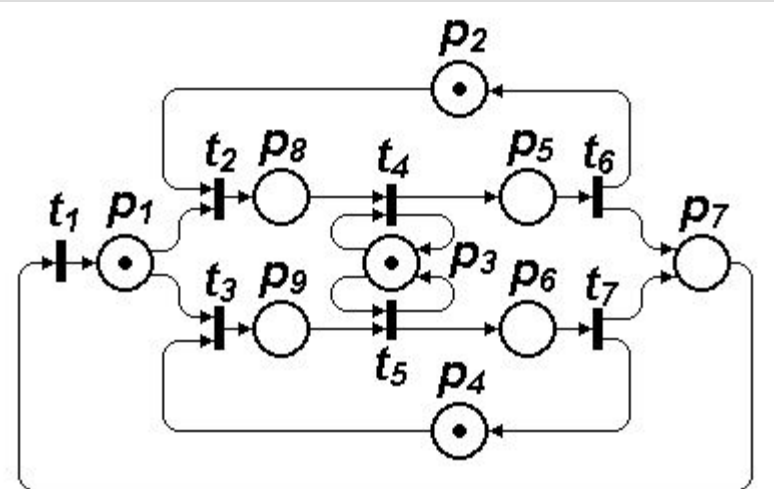


# Лекция 8. Сети Петри

- Свойства сетей Петри:

## Пример 2

Сеть, моделирующая  
Двухпроцессорную  
вычислительную систему  
с общей памятью,  
является безопасной,  
живой, все разметки  
достижимы.



# Лекция 8. Сети Петри

- Примеры применения сетей Петри:

## **Защита программ**

Сеть Петри является идеальным инструментом для моделирования переходов при выполнении определенных условий.

Не секрет, что большинство защит сводится к максимальному сокрытию и недопущения нахождение пресловутого условного перехода. При этом, обертка системы может состоять из самых изощренных анти-отладочных приемов, программа может быть запакована множеством упаковщиков, и любыми другими средствами, но в середине всей этой внешней мишуры находится один единственный переход.



# Лекция 8. Сети Петри

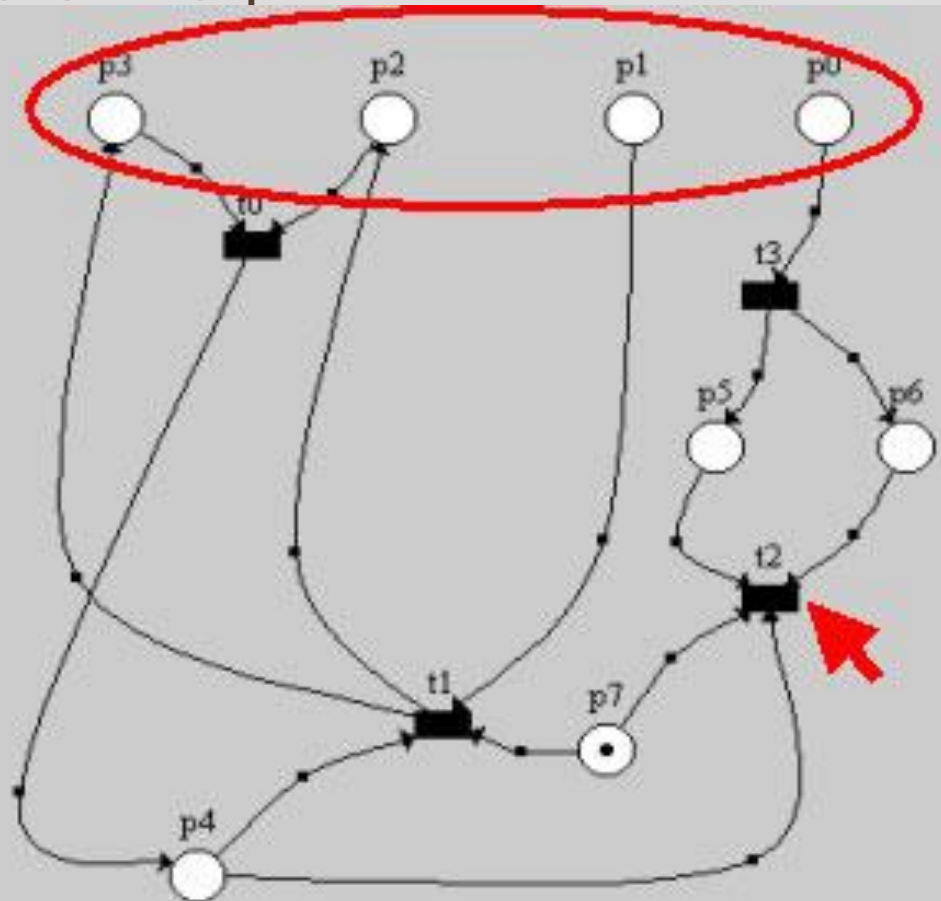
- Примеры применения сетей Петри:

## Защита программ

Допустим, что при начальной маркировке можно задавать кол-во фишек только в позициях  $p_0 \dots p_3$ . ( $p_7$  изначально содержит одну фишку). Остальные позиции недоступны для маркирования.

Зададим *ключевому* переходу  $t_2$  наименьший приоритет, после чего он (переход  $t_2$ ) выполниться тогда и только тогда, когда начальная маркировка будет в  $p_0 \dots p_3$  следующим образом:

$p_0=1, p_1=0, p_2=1, p_3=1$ . При всех остальных начальных маркировках переход  $t_2$  не сработает. Попробуйте проверить.



# Лекция 8. Сети Петри

- Примеры применения сетей Петри:

## Защита программ

Теперь представим, что каждый переход – это поток, который проверяет определенный набор бит на ВХОДЕ (входные позиции) и в зависимости от результата устанавливает биты на ВЫХОДЕ. Так, поток  $t_3$  проверяет фишки (биты) в позиции  $p_0$  и в зависимости от того есть там фишки (бит установлен) или нет (бит сброшен) устанавливает биты на выходных позициях ( $p_5$  и  $p_6$ ).

Допустим, пользователю необходимо зарегистрировать программу, введя 4-битный ключ. Биты ключа (фишки) помещаются в позиции  $p_0, p_1, p_2$  и  $p_3$ . Этим обеспечивается начальная маркировка сети.

Программа будет считаться зарегистрированной, если переход  $t_2$  сработает (т.е. будет активным). Как уже было показано выше – переход  $t_2$  сработает только при такой начальной маркировке:  $p_0=1, p_1=0, p_2=1, p_3=1$ .

Все это легко можно реализовать программно.

# Лекция 8. Сети Петри

- Примеры применения сетей Петри:

## **Защита программ**

Решая задачу достижимости с использованием дерева достижимости мы фактически прибегаем к методу полного перебора. К данному методу вообще можно свести любую задачу. Поэтому защищенность здесь определяется только временем и ресурсами, затраченными на полный перебор.

На нашем примере легко видеть, что найти правильный ключ (т.е. правильную начальную маркировку) можно перебрав всего лишь  $2^4 = 16$  значений. Это можно сделать даже вручную, а взглянув на представление программы в виде сети Петри можно догадаться интуитивно за несколько секунд.

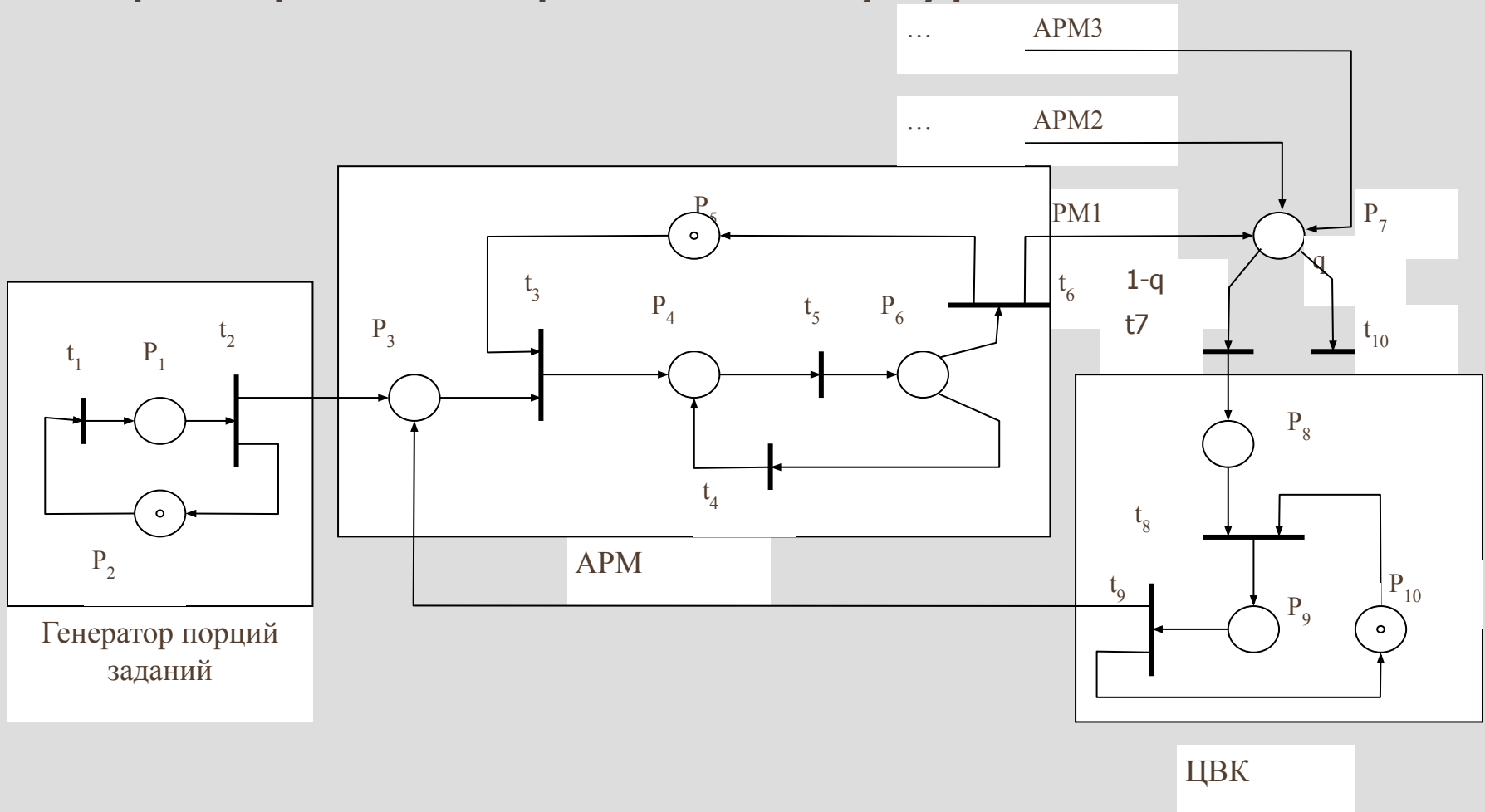
Использование ключей длиной менее  $2^{128}$  бит на сегодняшний день не может обеспечить защиту на приличном уровне, именно поэтому предложенная в примере сеть Петри не годится для практической защиты и приведена лишь в качестве наглядной иллюстрации.

# Лекция 8. Сети Петри

- **Пример моделирования двухуровневой ВС:**
  - Рассмотрим моделирование ВС, которую исследовали на языке GPSS при условии, что отсутствует КК и модель отражает работу только одного АРМ.
  - Это временная стохастическая сеть.
  - Считаются заданными параметры распределения случайной величины: время реакции пользователя на результаты обработки очередной порции информации; время обработки порции информации в АРМ и ЦВК; вероятность направления задания в ЦВК ( $q_1$ ); вероятность окончания решения ( $q_2$ ) после обработки очередной порции.

# Лекция 8. Сети Петри

- Пример моделирования двухуровневой ВС:



# Лекция 8. Сети Петри

- Пример моделирования двухуровневой ВС:
  - Позиции  $P1$  и  $P2$  и переходы  $t1$  и  $t2$  отражают поведение источника входных потоков заявок, имитирующего поступление запросов на использование АРМ.
  - Позиции  $P3$  и  $P4$  и переходы  $t3$  и  $t4$  соответствуют работе пользователя за дисплеем АРМ.
  - Позиции  $P5$  и  $P6$  и переходы  $t5$  и  $t6$  соответствуют работе АРМ (обработка информации).
  - Из  $P6$  заявки с вероятностью  $q1$  поступают на  $t6$  и с вероятностью  $(1 - q1)$  поступают на  $t4$ . Переход  $t5$  регулирует задержку в АРМ.
  - Позиция  $P7$  и переход  $t7$  отображают направление задач с вероятностью  $(1 - q2)$  на решение в ЦВК. Позиции  $P8, P9, P10$  и переходы  $t8, t9$  отображают обработку информации в ЦВК. Выход задач из системы происходит через переход  $t10$ .



# Лекция 9. Цепи Маркова

- **Содержание:**
  - Основные определения
  - Примеры
  - Применение

# Лекция 9. Цепи Маркова

## ● Основные определения:

- Цепь Маркова (*Markov Chain*) — последовательность случайных событий с конечным или счётным бесконечным числом исходов, характеризующаяся тем свойством, что при фиксированном настоящем будущее независимо от прошлого.
- Марков Андрей Андреевич, 1856—1922.
- Марков родился в Рязанской губернии.
- В 1878 окончил Петербургский университет. С 1880 — приват-доцент, с 1886 — профессор, а с 1905 — заслуженный профессор Петербургского университета, с 1896 — ординарный академик Императорской Санкт-Петербургской Академии Наук.

# Лекция 9. Цепи Маркова

## ● Основные определения:

- **Определение 1. Процесс**, протекающий в физической системе, называется **марковским**, если в любой момент времени вероятность любого состояния системы в будущем зависит только от состояния системы в текущий момент и не зависит от того, каким образом система пришла в это состояние.
- **Определение 2. Цепью Маркова** называется последовательность испытаний, в каждом из которых появляется только одно из  $k$  несовместных событий  $A_i$  из полной группы. При этом условная вероятность  $p_{ij}(s)$  того, что в  $s$ -ом испытании наступит событие  $A_j$  при условии, что в  $(s-1)$ -ом испытании наступило событие  $A_i$ , не зависит от результатов предшествующих испытаний.

# Лекция 9. Цепи Маркова

## ● Основные определения:

- **Определение 3. Цепью Маркова с дискретным временем** называется цепь, изменение состояний которой происходит в определенные фиксированные моменты времени.
- **Определение 4. Цепью Маркова с непрерывным временем** называется цепь, изменение состояний которой возможно в любые случайные моменты времени.

Пусть  $P_{ij}(n)$  вероятность того, что в результате  $n$  шагов (испытаний) система перейдет из состояния  $i$  в состояние  $j$ . Например,  $P_{36}(10)$  – вероятность перехода за 10 шагов из третьего состояния в шестое. Отметим, что при  $n=1$  эта вероятность сводится просто к переходной вероятности  $P_{ij}(1) = p_{ij}$ .

# Лекция 9. Цепи Маркова

- **Основные определения:**
  - **События** называются **состояниями системы**, а **испытания** – **изменениями состояний системы**.
  - Область значений случайных величин  $\{X_n\}$  называется **пространством состояний** цепи, а номер  $n$  — **номером шага**.
  - **Определение 5. Однородной** называется цепь Маркова, если условная вероятность  $p_{ij}$  перехода системы из состояния  $i$  в состояние  $j$  не зависит от номера испытания. Вероятность  $p_{ij}$  называется **переходной вероятностью**. Для однородных цепей вместо  $p_{ij}(s)$  используют обозначение  $P_{ij}$ .
  - В противном случае цепь Маркова называется **неоднородной**.

# Лекция 9. Цепи Маркова

## ● Основные определения:

- **Определение 6.** Допустим, число состояний конечно и равно  $k$ . Тогда матрица, составленная из условных вероятностей перехода будет иметь вид:

$$P_1 = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1k} \\ p_{21} & p_{22} & \dots & p_{2k} \\ \dots & \dots & \dots & \dots \\ p_{k1} & p_{k2} & \dots & p_{kk} \end{pmatrix}$$

- Эта матрица **называется матрицей перехода** системы. Т.к. в каждой строке содержатся вероятности событий, которые образуют полную группу, то сумма элементов каждой строки матрицы равна единице.

$$\sum_j P_{ij}(n) = 1, \quad \forall n \in \mathbb{N}$$



# Лекция 9. Цепи Маркова

## ● Основные определения:

- **Равенство Маркова.** Вопрос - как, зная переходные вероятности  $P_{ij}$ , найти вероятности перехода состояния  $i$  в состояние  $j$  за  $n$  шагов. Рассмотрим промежуточное (между  $i$  и  $j$ ) состояние  $r$ . Из первоначального состояния  $i$  за  $m$  шагов система перейдет в промежуточное состояние  $r$  с

вероятностью  $P_{ir}(m)$

, после чего за оставшиеся  $n-m$  шагов из промежуточного состояния  $r$  перейдет в конечное состояние  $j$  с вероятностью  $P_{rj}(n-m)$ . По формуле полной вероятности

$$P_{ij}(n) = \sum_{r=1}^k P_{ir}(m) P_{rj}(n-m)$$

Эту формулу называют равенством Маркова.

# Лекция 9. Цепи Маркова

## ● Основные определения:

- Зная матрицу перехода  $P_{ij} = P_{ij}(1)$  за один шаг, т.е.  $P_1$  можно найти вероятности перехода за два шага  $P_{ij}(2)$ , а значит, и саму матрицу  $P_2$  перехода, далее – по известной  $P_2$  матрице  $P_3$  – найти и т.д.
- Полагая в равенстве Маркова  $n=2$ ,  $m=1$  получим:

$$P_{ij}(2) = \sum_{r=1}^k P_{ir}(1) P_{rj}(2-1)$$

или 
$$P_{ij}(2) = \sum_{r=1}^k P_{ir} P_{rj} .$$

- В матричном виде это можно записать, как  $P_2 = P_1 P_1$ .
- Полагая  $n=3$ ,  $m=2$ , получим  $P_3 = P_1 P_2 = P_1 \cdot P_1^2 = P_1^3$ . В общем случае справедливо соотношение

$$P_n = P_1^n$$

# Лекция 9. Цепи Маркова

## ● Основные определения:

- Пример. Пусть матрица перехода равна

$$P_1 = \begin{pmatrix} 0,4 & 0,6 \\ 0,3 & 0,7 \end{pmatrix} .$$

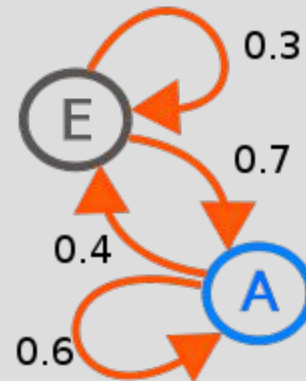
- Требуется найти матрицу перехода

$$P_2 = \begin{pmatrix} P_{11}(2) & P_{12}(2) \\ P_{21}(2) & P_{22}(2) \end{pmatrix} .$$

- Умножая матрицу  $P_1$  саму на себя, получим

$$P_2 = \begin{pmatrix} 0,34 & 0,66 \\ 0,33 & 0,67 \end{pmatrix}$$

- На основе матрицы перехода системы можно построить граф состояний системы. Удобно для наглядного представления цепи.



# Лекция 9. Цепи Маркова

## ● Основные определения:

### Классификация состояний цепи Маркова

**1. Возвратное состояние** — это состояние марковской цепи, посещаемое ею бесконечное число раз.

Дана однородная цепь Маркова с дискретным временем.

- Пусть  $f_{ii}^{(n)}$  — вероятность, того, что выйдя из состояния  $i$ , вернуться в него ровно за  $n$  шагов. Тогда  $f_{ii} = \sum_{n=1}^{\infty} f_{ii}^{(n)}$

— вероятность, выйдя из состояния  $i$ , вернуться в него (за конечное или бесконечное время).

- Состояние  $i$  называется **возвратным (рекуррентным)**, если  $f_{ii} = 1$ . В противном случае состояние называется **невозвратным (транзиентным)**.

# Лекция 9. Цепи Маркова

## ● Основные определения:

### Классификация состояний цепи Маркова

#### 2. Достижимое состояние

Пусть  $\{X_n\}_{n \in \mathbb{N}}$  — однородная цепь Маркова с дискретным временем. Состояние  $j$  называется **достижимым** из состояния  $i$ , если существует  $n = n(i, j)$  такое, что

$$p_{ij}^{(n)} \equiv \mathbb{P}(X_n = j \mid X_0 = i) > 0$$

Пишут  $i \rightarrow j$ .

- Состояния  $i$  и  $j$  называются **сообщающимися**, если  $i \rightarrow j$  и  $j \rightarrow i$ . Пишем:  $i \leftrightarrow j$ .

- Цепь Маркова называется **неприводимой**, если любое состояние  $S_j$  может быть достигнуто из любого другого состояния  $S_i$  за конечное число переходов. В этом случае все состояния цепи - **сообщающиеся**

# Лекция 9. Цепи Маркова

## ● Основные определения:

### Классификация состояний цепи Маркова

#### 3. Периодическое состояние

— это такое состояние цепи Маркова, которое навещается цепью только через промежутки времени, кратные фиксированному числу.

Дана однородная цепь Маркова  $P^n = (p_{ij}^{(n)})$ . Рассмотрим последовательность  $p_{jj}^{(n)}, n \in \mathbb{N}$ . Число

$$d(j) = \gcd \left( n \in \mathbb{N} \mid p_{jj}^{(n)} > 0 \right)$$

где  $\gcd$  обозначает наибольший общий делитель, называется **периодом** состояния  $j$ .

Таким образом, период состояния  $j$  равен  $d(j)$ , если из того, что  $p_{jj}^{(n)} > 0$  следует, что  $n$  делится на  $d(j)$ .



# Лекция 9. Цепи Маркова

## ● Примеры

1. Если заданы начальное распределение вероятностей и матрица перехода, то вероятности состояний системы на  $n$ -м шаге вычисляются по рекуррентной формуле (частный случай равенства Маркова):

$$P_i(n) = \sum_{j=1}^k P_j(n-1) \cdot P_{ji}$$

Рассмотрим пример (функционирование прибора). Пусть прибор в течение одних суток может находиться в одном из двух состояний – исправном ( $S_1$ ) и неисправном ( $S_2$ ). В результате массовых наблюдений за работой прибора составлена следующая матрица перехода:

$$P_{ij} = \begin{pmatrix} 0,8 & 0,2 \\ 0,9 & 0,1 \end{pmatrix}$$

# Лекция 9. Цепи Маркова

## ● Примеры

где  $P_{11}$  – вероятность того, что прибор останется в исправном состоянии;

$P_{12}$  – вероятность перехода прибора из исправного в неисправное состояние;

$P_{21}$  – вероятность перехода прибора из неисправного в исправное состояние;

$P_{22}$  – вероятность того, что прибор останется в состоянии «неисправен».

Пусть вектор начальных вероятностей состояний прибора задан соотношением  $P(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ , т.е.

$P_1(0) = 0$  и  $P_2(0) = 1$  (в начальный момент прибор был неисправен). Требуется определить вероятности состояния прибора через трое суток.

# Лекция 9. Цепи Маркова

## ● Примеры

Решение: Используя матрицу перехода, определим вероятности состояний после первого шага (после первых суток):

$$P_1(1) = P_1(0) \cdot P_{11} + P_2(0) \cdot P_{21} = 0 \cdot 0,8 + 1 \cdot 0,9 = 0,9$$

$$P_2(1) = P_1(0) \cdot P_{12} + P_2(0) \cdot P_{22} = 0 \cdot 0,2 + 1 \cdot 0,1 = 0,1$$

Вероятности состояний после второго шага (вторых суток) равны:

$$P_1(2) = P_1(1) \cdot P_{11} + P_2(1) \cdot P_{21} = 0,9 \cdot 0,8 + 0,1 \cdot 0,9 = 0,81$$

$$P_2(2) = P_1(1) \cdot P_{12} + P_2(1) \cdot P_{22} = 0,9 \cdot 0,2 + 0,1 \cdot 0,1 = 0,19$$

Вероятности состояний после третьего шага (третьих суток) равны:

$$P_1(3) = P_1(2) \cdot P_{11} + P_2(2) \cdot P_{21} = 0,81 \cdot 0,8 + 0,19 \cdot 0,9 = 0,819$$

$$P_2(3) = P_1(2) \cdot P_{12} + P_2(2) \cdot P_{22} = 0,81 \cdot 0,2 + 0,19 \cdot 0,1 = 0,181$$

Таким образом, вероятность того, что прибор будет находиться в исправном состоянии равна 0,819, и того, что в неисправном – соответственно 0,181.

# Лекция 9. Цепи Маркова

## ● Применение

### 1. Определение авторства текста

- Формальный анализ текста - определение авторства текста.
- Частоты употребления пар букв очень хорошо характеризуют автора.
- Первые пробные шаги предпринял еще А.А. Марков. Изучал распределение доли гласных и согласных среди первых 20000 букв "Евгения Онегина".
- Исследовались простые параметры авторского стиля на огромном числе произведений писателей XVIII-XX веков, доказано, что доля всех служебных слов в длинном прозаическом произведении является т.н. авторским инвариантом.
- Метод основывается на цепи Маркова. По тем известным произведениям автора вычисляется матрица переходных частот употреблений пар букв. Для каждого автора оценивается вероятность того, что именно он написал анонимный фрагмент текста. Автором анонимного текста полагается тот, у которого вычисленная оценка вероятности больше.

# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

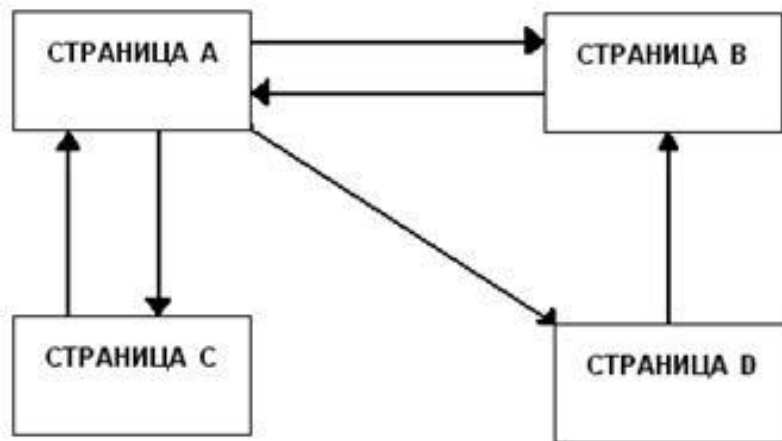
- При поиске и ранжировании документов Google основывается на содержании страницы, ключевых словах в заголовке, описании и на значении PageRank.
- PageRank - число, отображающее важность страницы.
- Для расчета PageRank используется следующая формула:  
$$PR(A) = (1-d) + d \cdot ( PR(T1)/C(T1) + \dots + PR(TN)/C(TN))$$
где  $PR(A)$  – PageRank страницы  $A$ ;  $T1... Tn$  – страницы, ссылающиеся на  $A$ ;  $C(T1)$  – количество ссылок страницы  $T1$ ;  $d$  – коэффициент затухания, находится в пределах от 0 до 1, обычно равен 0,85.
- Ограничение:  $\sum PR_j = N$ ,  $j=1...N$ ,  $N$  – количество страниц в сети.

# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

Пример, сеть из четырех страниц



На первом шаге (итерации) расчет (при условии, что PageRank всех страниц равен 1) дает

$$PR(A) = (1-0,85) + 0,85 \cdot ( PR(B)/1 + PR(C)/1) = 0,15 + 0,85 \cdot ( 1/1 + 1/1) = 1,85.$$

$$PR(B) = (1-0,85) + 0,85 \cdot ( PR(A)/3 + PR(D)/1) = 0,15 + 0,85 \cdot ( 1/3 + 1/1) = 1,28.$$

$$PR(C) = (1-0,85) + 0,85 \cdot ( PR(A)/3) = 0,15 + 0,85 \cdot ( 1/3) = 0,43.$$

$$PR(D) = (1-0,85) + 0,85 \cdot ( PR(A)/3) = 0,15 + 0,85 \cdot ( 1/3) = 0,43.$$



# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

На втором этапе, подставляем не 1, а полученные значения:

$$PR(A) = 0,15 + 0,85 \cdot (1,28/1 + 0,43/1) = 1,609.$$

$$PR(B) = 0,15 + 0,85 \cdot (1,85/3 + 0,43/1) = 1,425.$$

$$PR(C) = 0,15 + 0,85 \cdot (1,85/3) = 0,674.$$

$$PR(D) = 0,15 + 0,85 \cdot (1,85/3) = 0,674.$$

После 10 шагов - сходимость, результат:

$$PR(A) = 1,637;$$

$$PR(B) = 1,136;$$

$$PR(C) = 0,614;$$

$$PR(D) = 0,614.$$

Сумма всех PR равна 4, ограничение выполняется.

# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

Страница A имеет самый высокий PR, страницы C и D – наиболее низкий, т.е. вероятность попасть на страницу A больше всего.

Поэтому если все четыре страницы, будут по содержанию соответствовать какому-то запросу поиска, то несмотря на достаточно похожее содержание страниц, после учета значений PR, страница A окажется на первом месте, B – на втором, а C и D – на третьем.

Данный процесс является Марковским.

Обозначим шаги (этапы) процесса за  $t = 0, 1, \dots$

Формула расчета в матричном виде выглядит следующим образом:

$$P(t+1) = P(t) \cdot P. \text{ Отсюда следует, что } P(t) = P(0) \cdot P^t$$

$P(*) = P(*) \cdot P$  - предельный вектор вероятностей  $P(*)$ , является единственным и не зависит от вектора начальных вероятностей  $P(0)$  и определяются только матрицей вероятностей перехода.

# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

Матрица вероятностей перехода:

P=	A	B	C	D
A	0	1/3	1/3	1/3
B	1	0	0	1
C	1	0	0	0
D	0	1	0	0

Первая строка - со страницы A можно перейти на B, C, и D.

Вероятность того, что после A пользователь выберет C равна 1/3.

Вероятность выбора D или C тоже равна 1/3.

Вторая строка - со страницы B можно попасть только на A, вероятность этого перехода равна 1. Третья строка – с C можно попасть только на A, вероятность перехода равна 1. Четвертая строка – с D можно попасть только на B, вероятность перехода равна 1.

# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

Вектор начальных вероятностей (т.к. четыре страницы и пользователь может оказаться на любой из них с одинаковой вероятностью):

$$P(0) = \langle 1/4, 1/4, 1/4, 1/4 \rangle.$$

Для расчета вектора  $P(1)$ ,  $P(0)$  умножим на матрицу вероятностей перехода. Получим  $P(1) = \langle 0,5; 0,3333; 0,0833; 0,0833 \rangle$ .

Для вектора  $P(2)$ :  $P(2) = P(1) \cdot P$ . После расчета получаем значения  $P(2) = \langle 0,4167; 0,25; 0,167; 0,167 \rangle$ .

Значение предельного вектора вероятностей  $P(*)$ , который не зависит от вектора начальных вероятностей  $P(0)$  и определяется только матрицей вероятностей перехода.

В нашем случае  $P(*) = \langle 0,429; 0,286; 0,143; 0,143 \rangle$ .

42% пользователей будут на странице А, 29% будут на странице В и на страницах С и D будет по 14% посетителей.

# Лекция 9. Цепи Маркова

## ● Применение

### 2. Поисковые системы. Применение PageRank

Теперь с полученным вектором  $P^*$  произведем следующие действия:

$$0,85 \cdot 4P^* + (1 - 0,85)$$

Проведем расчеты:

$$\text{Для A: } 0,85 \cdot 4 \cdot 0,429 + 0,15 = 1,607.$$

$$\text{Для B: } 0,85 \cdot 4 \cdot 0,286 + 0,15 = 1,12.$$

$$\text{Для C: } 0,85 \cdot 4 \cdot 0,143 + 0,15 = 0,635.$$

$$\text{Для D: } 0,85 \cdot 4 \cdot 0,143 + 0,15 = 0,635.$$

Получается что, рассчитав предельный вектор вероятностей  $P^*$ , умножив его на единичный вектор, умноженный на  $4d$ , и прибавив к нему единичный вектор, умноженный на  $(1-d)$ , мы получили значения практически равные значениям PR, рассчитанным ранее.

# Лекция 9. Цепи Маркова

## ● Применение

### 3. Автоматическое формирование текста и спама

- Рассмотрим текст, состоящий из слов  $w$ .
- Представим процесс, в котором состояниями являются слова, так что когда он находится в состоянии  $(S_i)$  система переходит в состояние  $(S_j)$  согласно матрице переходных вероятностей.
- Сначала «обучают» систему: на каком-то тексте оценивают переходные вероятности.
- Затем строят траектории марковской цепи.
- Увеличение смысла в тексте, построенном при помощи алгоритма цепей Маркова возможно только при увеличении порядка, где состоянием является не одно слово, а множества — пары  $(u, v)$ , тройки  $(u, v, w)$  и т.д.
- Причем смысл начнет появляться при увеличении размерности порядка как минимум до среднего количества слов в типовой фразе исходного текста. Рост смысловой нагрузки текста в цепях Маркова высоких порядков происходит значительно медленнее, чем падение уникальности текста.



# Лекция 9. Цепи Маркова

## ● Применение

### 3. Автоматическое формирование текста и спама

- Эта технология сейчас очень широко применяется (к сожалению) в *Интернете* для создания web-контента и поискового спама.
- Увеличить трафик на свой сайт и повысить его рейтинг в поисковых системах, можно помещая на свои страницы как можно больше ключевых слов для поиска. Для этого используют тексты, созданные генератором на основе марковской цепи.
- В настоящее время одной из основных проблем поиска является распространение поискового спама. Поисковый спам создается для завышения оценки страницы в поисковой системе, по сравнению с ее истинной ценностью. В соответствии с современными оценками поисковый спам составляет около 22% всего содержимого сети Интернет. Одним из распространенных способов автоматического создания большого количества текстов является генерация текстов на основе цепей Маркова, порождающих большое количество в целом бессмысленных, но локально связанных текстов.

# Лекция 9. Цепи Маркова

## ● Применение

### 4. Другие

- Расчет сумм для обязательного страхования автогражданской ответственности
- Расчет результатов лотерей и т.п.

### Литература

1. Беляев А., Гаврилов М., Масальских А., Медвинский М. Марковские процессы, 2004.
2. Алексеев В. Марковские процессы принятия решений, 2002.
3. Беляев А. Марковские цепи, 2002.