ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ

ОГЛАВЛЕНИЕ

1 КОРОТКО О ГЛАВНОМ GPSS

Язык GPSS (GeneralPurposeSimulationSystem), ориентированный на процессы, разработан еще в 1961 г., но продолжает широко использоваться. Язык реализован в ряде программ имитационного моделирования, так, версия программы GPSS/PC в среде Windows создана в 2000 г.

Модель (программа) на языке GPSS представляет собой последовательность операторов (их называют блоками), отображающих события, происходящие в СМО при перемещениях транзактов. Поскольку в интерпретаторах GPSS реализуется событийный метод и в СМО может быть одновременно много транзактов, то интерпретатор будет попеременно исполнять разные фрагменты программы, имитируя продвижения транзактов в текущий момент времени до их задержки в некоторых устройствах или очередях.

Операторы (блоки) GPSS имеют следующий формат:

<метка><имя_оператора><поле_операндов> [<комментарий>]

Метка может занимать позиции, начиная со второй, имя оператора – с восьмой, поле операндов – с девятнадцатой, комментарий обязательно отделяется от поля операндов пробелом.

Поле операндов может быть пусто, иметь один или более операндов, обозначаемых ниже при описании блоков символами **A** , **B** , **C** ,... Операндами могут быть идентификаторы устройств, накопителей, служебные слова и стандартные числовые атрибуты (СЧА). К СЧА относятся величины, часто встречающиеся в разных задачах. Это, например, такие операнды, как **S** – объем занятой памяти в накопителе, **F** – состояние устройства, **Q** – текущая длина очереди, **P** – параметр транзакта (каждый транзакт может иметь не более **L** параметров, где **L** зависит от интерпретатора), **V** – целочисленная переменная

(вещественная и булева переменные обозначаются **FV** и **BV** соответственно), **X** – хранимая переменная (переменная, для которой автоматически подсчитывается статистика), **K** – константа, **AC1** – текущее время, **FN** – функция, **RN** – случайная величина, **RN1** – случайная величина, равномерно распределенная в диапазоне [0, 1] и др. При этом ссылки на СЧА записываются в виде **<СЧА>$<идентификатор>**. Например, **Q$ORD** означает очередь **ORD** или **FN$COS** — ссылка на функцию **COS**.

Рассмотрим наиболее часто встречающиеся операторы, сопровождая знакомство с ними простыми примерами моделей. Источники заявок обычно описываются блоком:

```
GENERATE A,B,C,D,E
```

Здесь **A** и **B** служат для задания интервалов между появлениями заявок, при этом можно использовать один из следующих вариантов:

- интервал – равномерно распределенная в диапазоне [**A-B,A+B**] случайная величина;
- интервал – значение функции, указанной в **B**, умноженной на **A**;

C – задержка в выработке первого транзакта; **D** — число вырабатываемых источником заявок; **E** — приоритет заявок. Если **D** пусто, то число вырабатываемых транзактов неограничено. Например:

```
GENERATE 6,FN$EXP,,15
```

Этот оператор описывает источник, который вырабатывает 15 транзактов с интервалами, равными произведению числа 6 и значения функции EXP;

```
GENERATE 36,12
```

Здесь число транзактов неограничено, интервалы между транзактами – случайные числа в диапазоне **[24, 48]**. Функции, на которые имеются ссылки в операторах, должны быть описаны с помощью блока следующего типа:

```
M     FUNCTION A,B
```

За ним следует строка, начинающаяся с первой позиции :

```
X1,Y1/X2,Y2/X3,/.../Xn,Yn
```

Здесь метка **M** – идентификатор функции, **A** – аргумент функции, **B** – тип функции, **Xi** и **Yi** — координаты узловых точек функции, заданной таблично. Например:

```
EXP FUNCTION RN1,C12
0,0/.2,.22/.4,.51/.5,.6/.6,.92/.7,1.2/.8,1.61/.9,2.3/.95,3/
.99,4.6/.999,6.9/1,1000
```

Это описание непрерывной (**C**) функции **EXP**, заданной таблично **12**-ю узловыми точками, аргументом является случайная равномерно распределенная величина в диапазоне [0, 1]; или:

```
BBB FUNCTION(RN1), *4,D6
1,2/2,5/3,11/4,20/5,18/6,12/7,9
```

Дискретная (**D**) функция **BBB** задана 6-ю узловыми точками, аргумент — четвертый параметр транзакта, возбудившего обращение к функции **BBB**.

Здесь аргумент задан с использованием косвенной адресации, признаком которой является символ *****, т.е. запись ***4** означает, что аргументом является величина, указанная в 4-м параметре транзакта, вызвавшего функцию (в данном примере можно было бы использовать равноценную запись ***p4**). В общем случае

косвенная адресация выполняется путем записи операнда в виде **СЧА*СЧА.** Например: **Q*p5** – длина очереди с именем, записанным в парамтере 5 транзакта.

Тразакты могут порождаться и оператором размножения:

SPLIT A,B,C

Новые транзакты порождаются, когда в данный блок входит некоторый транзакт. При этом создается семейство транзактов, включающее основной (вошедший в блок) транзакт и **A** его копий. Основной транзакт переходит в следующий по порядку блок, а его копии переходят в блок с меткой **B**. Для различения транзактов параметр **C** основного транзакта увеличивается на 1, а транзактов-копий – на 2, 3, 4,... и т. д.

Обратное действие – сборка транзактов выполняется операторами:

ASSEMBLE A

GATHER A

Согласно оператору **ASSEMBLE** первый из вошедших в блок транзактов выйдет из него только после того, как в этот блок придут еще **A-1** транзактов того же семейства. Второй оператор отличается от предыдущего тем, что из блока выходят все **A** транзактов.

Операторы **занятия** транзактом и **освобождения** от обслуживания устройства **A**:

SEIZE A

RELEASE A

Задержка в движении транзакта по СМО описывается оператором:

ADVANCE

A,B

A и **B** имеют тот же смысл, что и в операторе **GENERATE**[1].

ПРИМЕРЫ в **finder_mod.pdf** (248 стр.).

1. ***A*** *и* ***B*** *задают интервал между появлениями заявок, юзаем один из следующих вариантов:*
   - *интервал – равномерно распределенная в диапазоне* ***[A-B,A+B]*** *случайная величина;*
   - *интервал – значение функции, указанной в* ***B****, умноженной на* ***A****;*)

2 МОДЕЛИРОВАНИЕ

2.1 Философские аспекты моделирования.

Объектом называется всё то, на что направлена человеческая деятельность.

В научном исследовании большую роль играет понятие **гипотезы** – определенное предсказание, основанное на небольшом количестве опытных данных, наблюдениях, догадках. Быстрая проверка гипотезы может быть проведена в ходе специально поставленных экспериментов.

При формировании и проверке правильности гипотезы в качестве метода суждения используется **аналогия**. **Аналогией** называется суждение о каком либо частном сходстве двух объектов.

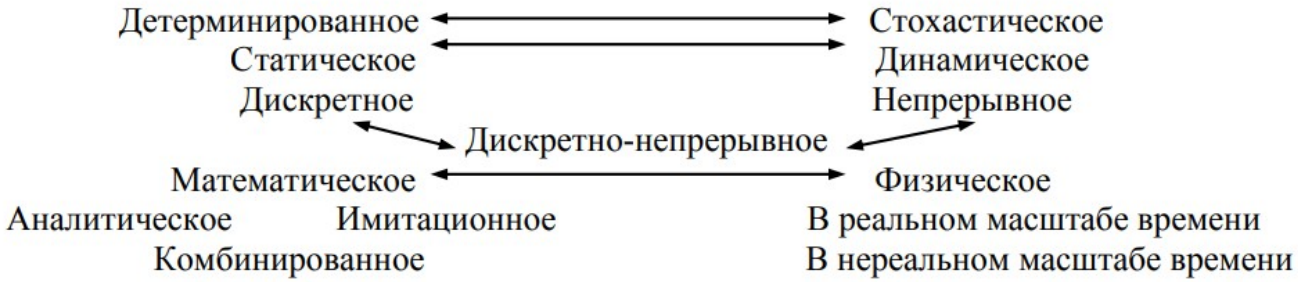
Современные научные гипотезы создаются как правило по аналогии проверенным на практике положениям. Таким образом, аналогия связывает гипотезу с экспериментом.

Гипотезы и аналогии, отражающие реальный объктивно-существующий мир, должны обладать наглядностью или сводиться к удобным для исследования логическим схемам. Такие логические схемы, упрощающие рассуждения и позволяющие проводить эксперименты, уточняющие природу явлений, называются моделями.

Модель – объект - заместитель объекта оригинала, обеспечивающий изучение некоторых свойств оригинала. Замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели называется **моделированием**.

2.2 Классификация видов моделирования.

В зависимости от характера изучаемых процессов в некоторой сложной системе все виды моделирования можно разделить:



- **Детерминированное** моделирование отображает детерминированные процессы, т.е. такие, в которых отсутствуют всякие случайные воздействия.
- **Стохастическое** моделирование отображает случайные, вероятностные процессы и события.
- **Статическое** служит для описания сложной системы в конкретный момент времени.
- **Динамическое** отражает поведение системы во времени.
- **Дискретное** моделирование используется для описания процессов, происходящих в дискретные моменты времени.
- **Непрерывное** используется для описание непрерывных во времени процессов.
- **Дискретно-непрерывное** используется для тех случаев, когда хотят отразить наличие как дискретных, так и непрерывных процессов в системе.
- Под **математическом моделированием** будем понимать процесс установления данному реальному объекту некоторого математического объекта, называемого математической моделью и исследование этой модели,

позволяющее получить характеристики реального объекта. Любая математическая модель, как и всякая другая, описывает реальный объект лишь с некоторой степенью приближения.

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

Аналитические модели могут быть исследованы тремя способами:

1. **Аналитическим**. Получение в общем виде зависимости выходных характеристик от исходных.

2. **Численным**. Нельзя решить сложные уравнения в общем виде. Результаты получают для конкретных начальных данных.

3. **Качественным**. Нет возможности получения конкретных решений, но можно выделить некоторые свойства объектов или решений уравнений, например, оценить устойчивость решения.

- При **имитационном моделировании** алгоритм, реализующий модель, воспроизводит процесс функционирования системы во времени. Имитируются элементарные явления, составляющие процесс, с сохранением логической структуры объекта и последовательности протекания процесса во времени. Это позволяет по исходным данным получить сведения о состоянии процесса в определенные моменты времени. Преимуществом имитационного моделирования является возможность решения более сложных задач.

Имитационные модели позволяют достаточно просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные

характеристики системы, многочисленные случайные воздействия. Когда результаты, полученные имитационной моделью, являются реализацией случайных величин и функций, то для нахождения характеристик процесса функциональной системы необходимо его многократное воспроизведение с последующей статистической обработкой.

- **Комбинированное моделирование** при анализе сложных систем позволяет объединить достоинства отдельных методов. В нем проводят декомпозицию процесса функционирования сложной системы на подпроцессы и для тех, где можно используют аналитические модели, где нельзя – имитационное моделирование.

2.3 Технические средства математического моделирования

2.3.1 Цифровая техника

Цифровая техника является дискретной. Основная проблема – быстродействие (не догнать реальное время) слишком сложен механизм.

2.3.2 Аналоговая техника

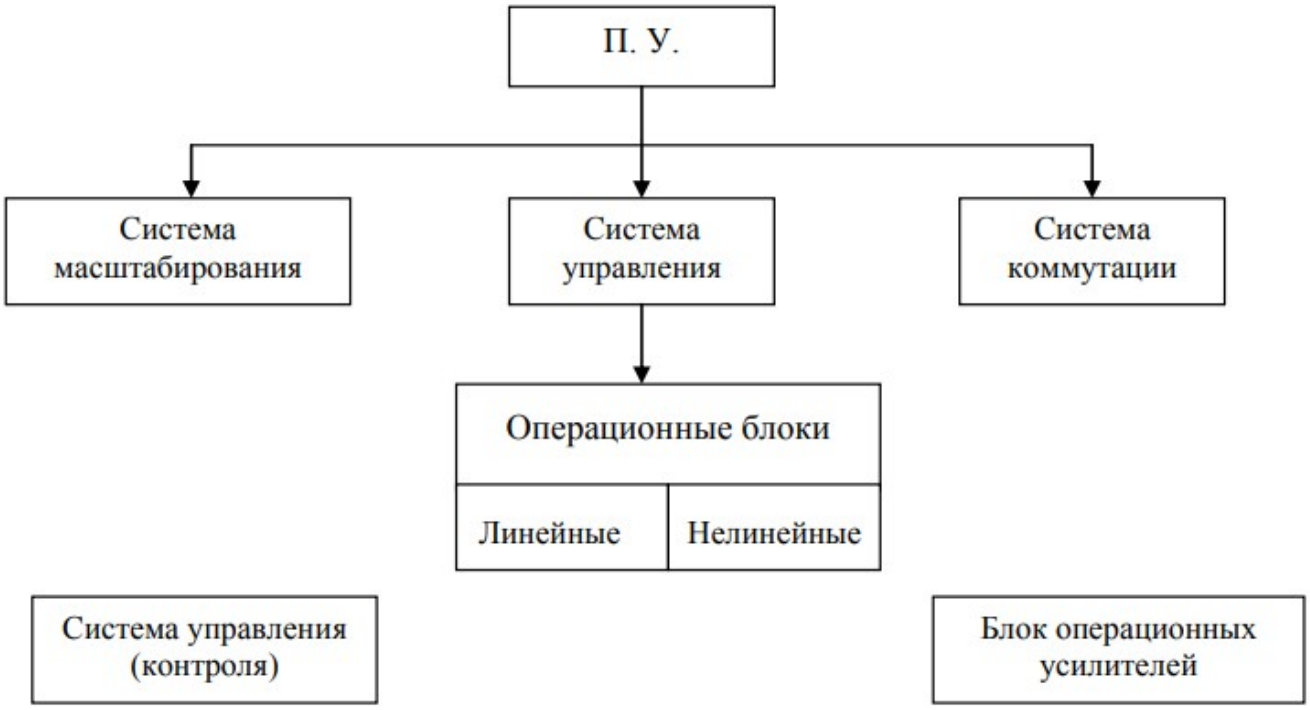
В отличие от дискретной техники в основе аналоговой лежит принцип моделирования, а не счета. При использовании в качестве модели некоторой задачи электронных цепей, каждой переменной величине ставится в соответствие определенную переменную величину электрической цепи. При этом основой построения такой модели является **изоморфизм** - подобие исследуемой задачи и соответствующей электрической модели. При определении критерия подобия используют специальные приемы масштабирования, соответствующие заданным параметрам.

Согласно своим вычислительным возможностям АВМ наиболее приспособлены для исследования объектов, динамика которых описывается обыкновенными дифференциальными уравнениями и уравнениями в частных производных, реже - алгебраическими, следовательно, АВМ можно отнести к классу специальных машин.

В общем случае под АВМ понимаем совокупность электрических элементов, организованных с систему, позволяющих изоморфно моделировать динамику изучаемого объекта. Функциональные блоки АВМ должны реализовывать весь комплекс арифметикологических операций.

АВМ делятся по мощности (степень дифференциальных уравнений):

- малые ($n \leq 10$)
- средние ($10 \leq n \leq 20$)
- большие ($n \leq 20$)



Теги: Система масштабирования, Система управления, Система коммутации, Операционные блоки, Система управления (контроля), Блок операционных усилителей.

2.3.3 Гибридные ВМ

Широкий класс ВС, использующий как аналоговый, так и дискретный метод представления и обработки информации.

Подклассы гибридных ВМ:

1. АВМ с цифровыми методами численного анализа
2. АВМ, программируемые с помощью ЦВМ
3. АВМ с цифровым управлением и логикой
4. АВМ с цифровыми элементами (цифровые вольтметры, память)
5. ЦВМ с аналоговыми арифметическими устройствами
6. ЦВМ, допускающие программирование аналогового типа.

В АВМ накладывают и складывают сигналы.

АВМ ↔ система сопряжения АВМ – ЦВМ.

ЦВМ ↔ электрическая система согласования

2.4 Формал. и алгоритмиз. процесса функционир. сложных систем

Сущность компьютерного моделирования состоит в проведении эксперимента с моделью, которая обычно представляет собой некоторый программный комплекс, описывающий формально или алгоритмически поведение элементов в системе в процессе ее функционирования, т.е. во взаимодействии друг с другом и с внешней средой.

Основные требования, предъявляемые к модели, отображающей функционирование некоторой системы:

1. **полнота модели** - должна предоставлять пользователю возможность получения необходимого набора характеристик, оценок системы, с требуемой точностью и достоверностью.
2. **гибкость модели** – должна давать возможность воспроизведения различных ситуаций при варьировании структуры, алгоритмов и параметров модели. При чем структура должна быть блочной – допускать возможность замены, добавления, исключения некоторых частей без переделки всей модели.
3. **машинная реализация модели** – должна соответствовать имеющимся ресурсам. Ресурсы - технические, экономические. Пример: автомобиль, критерий – мин. расход топлива.

Процесс моделирования включает разработку и машинную реализацию модели, является итерационным. Этот итерационный процесс продолжается до тех пор, пока не будет получена модель, которую можно считать адекватной в рамках решения поставленной задачи (исследования и проектирования).

2.5 Основные этапы моделирования больших систем

1. Построение концептуальной (описательной) модели системы и ее формализация
2. Алгоритмизация модели и ее компьютерная реализация.
3. Получение и интерпретация результатов моделирования

2.5.1 1 этап

Формулируется модель и строится ее формальная схема. Основное назначение – переход от содержательного описания объекта к его математической модели. Исходный материал – описание.

Последовательность действий:

1. Проведение границы между системой и внешней средой
2. Исследование объекта с точки зрения основных составляющих процесса
3. Переход от содержательного описания системы к формализованному описанию свойств процесса функционирования – непосредственно к концептуальной модели. Переход от описания к модели сводится к исключению из рассмотрения некоторых второстепенных элементов описания. Причем полагается, что они не оказывают существенного влияния на ход процесса. То, что осталось, разбивается на группы:

   Блоки 1 группы – имитатор воздействия внешней среды.

   Блоки 2 группы – модель процесса функционирования.

   Блоки 3 группы – вспомогательные, служат для машинной реализации 1 и 2 групп. 9
4. Процесс функционирования системы так разбивается на подпроцессы, чтобы построение модели отдельных процессов было элементарно и не вызывало особых затруднений.

2.5.2 2 этап

Математическая модель реализуется в конкретную программу. Основной этап – блочная логическая схема.

Последовательность действий:

1. Разработка схемы моделирующего механизма
2. Разработка схемы программы.
3. Выбор технических средств для реализации компьютерной программы

4. Программирование, отладка
5. Проверка достоверности программы на тестовых примерах.
6. Составление технической документации – логические схемы, схемы программы, текст, спецификация, иллюстрации и т. д.

2.5.3 3 этап

Компьютер используется для проведения рабочих расчетов по готовой программе. Результаты этих расчетов позволяют проанализировать и сделать выводы по характеристикам процесса функционирования исследуемой системы.

Последовательность действий:

1. Планирования машинного эксперимента (активный/пассивный) Составление плана проведения эксперимента, с указанием переменных и параметров, для которых должен проводиться эксперимент. Главная задача – дать максимальный объем информации об объекте при минимальных затратах машинного времени.
2. Проведений рабочих расчетов (контрольная калибровка модели).
3. Статистическая обработка результатов и их интерпретация.
4. Составления отчетов

При **стратегическом планировании** ставится задача построения оптимального плана эксперимента для достижения цели, поставленной перед моделированием (оптимизация структуры, алгоритмов и параметров).

Тактическое планирование преследует частные цели оптимальной реализации каждого конкретного эксперимента из множества необходимых – оно задается при стратегическом планировании.

2.5.4 Тактическое планирование

Тактическое планирование связано с вопросами **эффективности** и определением способа проведения испытаний, намеченных планом эксперимента. Тактическое планирование прежде всего связано с решением задач:

1. Определение начальных условий в той мере, в которой они влияют на достижение
2. Сокращение размеров выборки при уменьшении дисперсии решения.

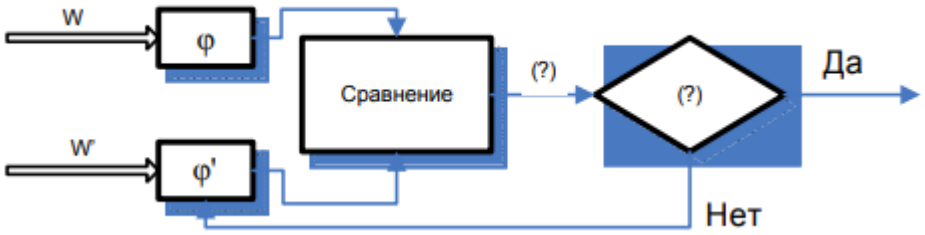
Для получения соотношений, связывающих характеристики, описывающие функционирование q-схемы, вводят некоторые допущения относительно входных потоков, функций распределения, длительности обслуживания запросов и дисциплины обслуживания. Для таких систем в качестве типовой математической модели будем рассматривать теорию Марковских процессов.

Случайный процесс, протекающий в некоторой системе S, называется Марковским процессом, если он обладает следующими свойствами:

1. для каждого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от её состояния в настоящий момент ($t = t_0$) и не зависит от того, когда и каким образом система пришла в это состояние.

Другими словами в Марковском случайном процессе будущее развитие зависит только от от настоящего состояния и не зависит от предыстории процесса. Для Марковского процесса разработаны уравнения Колмогорова. В общем виде они представляются: F = (P’(t),P(t),lambda). Лямбда – набор некоторых коэффициентов. В общем случае – просто вектор.

2.6 Итеративная калибровка модели



=объект=>[ф]+

[сравнение по критерию]->ошибка допустима? -да-> пропускаем, y'

=модель=>[ф]+

Три класса ошибок (по убыванию страшности):

1.Ошибки формализации. Неполное/недостаточно подробное модели объекта или предметной области.

2.Ошибки решения. Некорректный/слишком упрощенный метод построения модели.

3.Ошибка задания параметров модели (на стадии эксперимента).

2.6.1 Проверка адекватности и корректировки модели

Проверка адекватности модели некоторой системы заключается в анализе ее соразмерности и равнозначности системы. Адекватность нарушается из-за идеализации внешних условий и режимов функционирования; пренебрежением некоторых случайных факторов.

Простейшая мера адекватности применительно к ^ схеме калибровки:

Yмодели, Yобъекта. DeltaY – разность по модулю, либо относительная погрешность. Считают, что модель адекватна с системой, если вероятность того, что отклонение deltaY не превысит некоторой величины delta, больше допустимой вероятности.

Практическое использование критерия невозможно, потому что:

- для проектируемых/модернизируемых систем отсутствует информация о характеристике Y объекта
- система оценивается не по одной, а по множеству характеристик
- характеристики могут быть случайными величинами и функциями
- отсутствует возможность априорного точного задания предельных отклонений и допустимых вероятностей

На практике оценка адекватности обычно проводится путем экспертного анализа разумности результатов моделирования.

Выделяют следующие виды проверок:

- проверка моделей элементов
- проверка моделей внешних воздействий
- проверка концептуальной модели
- проверка способов измерения и вычисления выходных характеристика

Корректировка модели. Если по результатам проверки адекватности выявляются недопустимые рассогласования (объекта и модели), необходимо вносить изменения:

глобальные - в случае обнаружения методических ошибок в концептуальной/математической модели

локальные - связаны с уточнением некоторых параметров и алгоритмов. Выполняются путем замены моделей компонентов системы и внешних воздействий на эквивалентные, но более точные.

параметрические - изменения некоторых специальных параметров, называемых калибровочными

III этап завершается определением и фиксацией области пригодности модели, под которой понимается: множество условий, при соблюдении которых точность результатов моделирования находится в допустимых пределах.

2.7 Сети Петри

Сеть Петри – математическая абстракция, один из формализмов – на практике занимает положение между цифровым автоматом и вероятностным. ПО любой сложности можно формализовать графом (автомат с памятью/автомат мили-мура). Остаются нюансы, которые с помощью автоматного подхода не моделируются (проявляется на уровне регистровых передач) – событию нельзя придать какие-либо характеристики, *помимо* его собственно свершения (почему совершилось и т.п.).

Сеть Петри состоит из 4-х элементов[2]:

- множество позиций,
- множество переходов,
- входная функция,
- выходная функция.

Сети Петри позволяют ввести состояния, внутри которых используются «фишки». Моделирование производится с помощью «запуска» сети – формально, фишки передвигаются по графу. Можно получить цветную сеть с использованием разноцветных фишек. Можем получить модель для определения работоспособности программы и поиска тупиковых ситуаций.

Сети Петри можно представлять с двух точек зрения:

1. теория множеств (абстрактная математика)
2. графовое представления

2. Не из его лекций!

2.7.1 Обыкновенные сети Петри

Математическая модель дискретных динамических систем (параллельных программ, операционных систем, компьютеров и компонентов, вычислительных сетей), ориентированная на анализ и синтез таких систем. Даёт обнаружение блокировок, тупиковых ситуаций, узких мест при выполнении заданий, автоматический синтез параллельных программ, синтез компонент компьютера и т.д.

Формально, **сеть Петри** – кортеж $PN = (\theta, P, T, F, M_0)$

Если может сработать несколько переходов, то срабатывает любой из них. Функционирование сети останавливается, если при некоторой маркировке (тупиковая маркировка) ни один из её переходов не может сработать.

При одной и той же начальной маркировке, сеть Петри может порождать различные последовательности срабатывания её переходов (в силу недетерминированности её функционирования). **Эти последовательности образуют слова в некотором алфавите**. Множество всех возможных слов, порождаемых сетью Петри, называется языком сети Петри. **Две сети Петри эквивалентны, если порождают один и тот же язык.**

В отличие от конечных автоматов, в терминах которых описывается глобальное состояние системы, сети Петри акцентируют внимание на локальных событиях (переходах), локальных условиях (позициях) и локальных же связях между оными. Поэтому в терминах сетей Петри, более адекватно, чем с помощью автоматов, моделируется поведение распределенных асинхронных систем.

2.7.2 Графы сетей

Теоретико-графовым представлением сети Петри является двудольный ориентированный мультиграф. Этот граф содержит:

- позиции(места), кружки

- переходы, вертикальные планки
- ориентированные дуги, стрелки, соединяющие позиции с переходами и наоборот.

Кратные дуги обозначаются несколькими параллельными (можно указывать число). Благодаря наличию кратных дуг, сеть Петри есть мультиграф. Благодаря двум типам вершин, граф называется двудольным. Поскольку дуги имеют направления, граф является ориентированным.

2.7.3 Основные свойства сетей Петри

1. **Свойство ограниченности.** Позиция p в сети называется ограниченной, если для любой достижимой в сети маркировки М существует такое К, что $\mu_i \le k$.

Сеть называется **ограниченной**, если все её позиции ограничены. ^ сеть – неограниченна, потому что есть неограниченный рост μ_2 .

2. **Свойство безопасности.** Сеть называется безопасной, если при любой достижимой маркировке $\mu_i \ge 1 \; \forall \, i = \overline{1,n}$ (n –число позиций). Следовательно, в безопасной сети вектор маркировок состоит только из нулей или единиц (является двоичным словом).
3. **Свойство консервативности.** Сеть называется консервативной, если сумма фишек во всех позициях остается постоянной при работе сети.

$$\sum_{i=1}^{n} \mu_i = const \quad .$$

4. **Свойство живости.** Переход t_j в сети ПН называется потенциально живым, если существует достижимая из M_0 маркировка М', при которой t_j может сработать. Если t_j является потенциально живым при любой достижимой в сети маркировке, то он называется живым.

Переход t_j , не являющийся живым при начальной маркировке M0, называется мёртвым при этой маркировке. Маркировка M0 в этом случае называется $t_{j\,тупиковая}$ для перехода. Если маркировка M0 является $t_{j\,тупиковая}$ для всех j, то она называется тупиковой маркировкой. Другими словами, при тупиковой маркировке не может сработать ни один переход.

Если рассматривать *дерево* всех маркировок, тупик будет листовой вершиной. Переход называется устойчивым, если никакой другой переход не может лишить его возможности сработать при наличии для этого необходимых условий.

Последовательность маркировок $M_1, \ldots, M_p$, в которой последующая маркировка через функцию переходов может быть выражена из предыдущей, образует цикл в том случае, когда $M_0 = M_p$. Фактически, каждому циклу соответствует последовательность слов свободного языка сети.

Далее не расписываю **прям пипец**, т. к. сказал, что много спрашивать не будет… ¯_(ツ)_/¯

йцвйцвйцв