

# СЕТИ ПЕТРИ

## 1. ВВЕДЕНИЕ В СЕТИ ПЕТРИ

*Модель* — это представление, как правило, в математических терминах наиболее характерных черт изучаемого объекта или системы. *Сети Петри* это инструмент для математического моделирования и исследования сложных систем. Цель представления системы в виде сети Петри и последующего анализа этой сети состоит в получении важной информации о структуре и динамическом поведении моделируемой системы. Эта информация может использоваться для оценки моделируемой системы и выработки предложений по ее усовершенствованию. Впервые сети Петри предложил немецкий математик Карл Адам Петри.

### 1.1. Природа систем, моделируемых сетями Петри.

Сети Петри предназначены для моделирования систем, которые состоят из множества взаимодействующих друг с другом *компонент*. При этом компонента сама может быть системой. Действиям различных компонент системы присущ параллелизм. Примерами таких систем могут служить вычислительные системы, в том числе и параллельные, компьютерные сети, программные системы, обеспечивающие их функционирование, а также экономические системы, системы управления дорожным движением, химические системы, и т. д.

### 1.2. Подходы к проектированию систем с помощью сетей Петри.

В одном из подходов к проектированию и анализу систем сети Петри используются, как вспомогательный инструмент анализа. Здесь для построения системы используются общепринятые методы проектирования. Затем построенная система моделируется сетью Петри, и модель анализируется. Если в ходе анализа в проекте найдены изъяны, то с целью их устранения проект модифицируется. Модифицированный проект затем снова моделируется и анализируется. Этот цикл повторяется до тех пор, пока проводимый анализ не

приведет к успеху.

Другой подход предполагает построение проекта сразу в виде сети Петри. Методы анализа применяются только для создания проекта, не содержащего ошибок. Затем сеть Петри преобразуется в реальную рабочую систему.

В первом случае необходима разработка методов моделирования систем сетями Петри, а во втором случае должны быть разработаны методы реализации сетей Петри системами.

## 2. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

### 2.1. Теоретико-множественное определение сетей Петри.

В сетях Петри события и условия представлены абстрактными символами из двух непересекающихся алфавитов, называемых соответственно множеством *переходов*  $T = \{t_1, t_2, \dots, t_m\}$  и множеством *мест*  $P = \{p_1, p_2, \dots, p_n\}$ . В графическом представлении сетей переходы изображаются "барьерами", а места — кружками.



Переход и его входные и выходные места.

. Сеть Петри  $N$  является четверкой  $N=(P,T,I,O)$ , где

- $I: T \rightarrow P^*$  — входная функция, сопоставляющая переходу мультимножество его входных позиций;
- $O: T \rightarrow P^*$  — выходная функция, сопоставляющая переходу мультимножество его выходных позиций.

### 2.2. Маркировка сетей Петри.

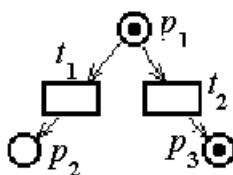
Маркировка — это размещение по позициям сети Петри фишек, изображаемых на графе сети Петри точками. Фишки используются для определения выполнения сети Петри. Количество фишек в позиции при

выполнении сети Петри может изменяться от 0 до бесконечности.

Маркировка  $\mu$  сети Петри  $N=(P,T,I,O)$  есть функция, отображающая множество позиций  $P$  в множество неотрицательных целых чисел  $Nat$  (где число из  $Nat$  обозначает количество фишек, помещаемых в соответствующую позицию).

Маркированная сеть Петри  $N=(P,T,I,O,\mu)$  определяется совокупностью структуры сети Петри  $(P,T,I,O)$  и маркировки  $\mu$

Графического представления маркированной сети Петри.



$\mu=<1,0,1>$ .

Множество всех маркировок сети Петри бесконечно. Если фишек, помещаемых в позицию слишком много, то удобнее не рисовать фишки в кружке этой позиции, а указывать их количество.

### 2.3. Правила выполнения сетей Петри.

Сеть Петри выполняется посредством запусков переходов. Запуск перехода управляется фишками в его входных позициях и сопровождается удалением фишек из этих позиций и добавлением новых фишек в его выходные позиции.

Переход может запускаться только в том случае, когда он разрешен. Переход называется *разрешенным*, если каждая из его входных позиций содержит число фишек, не меньшее, чем число дуг, ведущих из этой позиции в переход (или кратности входной дуги).

Запуски могут осуществляться до тех пор, пока существует хотя бы один разрешенный переход. Когда не останется ни одного разрешенного перехода, выполнение *прекращается*.

Если запуск произвольного перехода  $t$  преобразует маркировку  $\mu$  сети Петри в новую маркировку  $\mu'$ , то будем говорить, что  $\mu'$  достижима из  $\mu$  посредством запуска перехода  $t$  и обозначать этот факт, как  $\mu \rightarrow \mu'$ . Это понятие очевидным образом обобщается для случая последовательности запусков разрешённых переходов. Через  $R(N, \mu)$  обозначим множество всех достижимых маркировок из начальной маркировки  $\mu$  в сети Петри  $N$ .

### 3. МОДЕЛИРОВАНИЕ СИСТЕМ НА ОСНОВЕ СЕТЕЙ ПЕТРИ

В этом разделе рассмотрим метод моделирования на основе сетей Петри, а также его применение для моделирования параллельных систем взаимодействующих процессов и решения ряда классических задач из области синхронизации процессов.

#### 3.1. События и условия.

Представление системы сетью Петри основано на двух основополагающих понятиях: *событиях* и *условиях*. Возникновением событий управляет состояние системы, которое может быть описано множеством условий. Условие может принимать либо значение «истина», либо значение «ложь».

Возникновение события в системе возможно, если выполняются определённые условия – *предусловия* события. Возникновение события может привести к выполнению других условий – *постусловий* события. В качестве примера рассмотрим следующую ниже задачу моделирования.

Моделирование последовательной обработки запросов сервером базы данных. Сервер находится в состоянии ожидания до тех пор, пока от пользователя не поступит запрос, который он обрабатывает и отправляет результат такой обработки пользователю.

Условиями для рассматриваемой системы являются:

а) сервер ждет;

- б) запрос поступил и ждет;
- в) сервер обрабатывает запрос;
- г) запрос обработан.

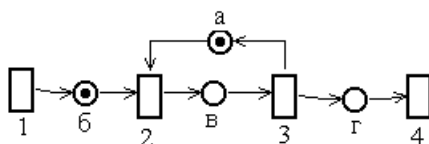
Событиями для этой системы являются:

1. Запрос поступил.
2. Сервер начинает обработку запроса.
3. Сервер заканчивает обработку запроса.
4. Результат обработки отправляется.

Для перечисленных событий можно составить следующую таблицу их пред- и постусловий

Событие	Предусловия	Постусловия
1	нет	б
2	а. б	в
3	в	г. а
4	г	нет

Такое представление системы легко моделировать сетью Петри. В сети Петри условия моделируются позициями, события — переходами. При этом входы перехода являются предусловиями соответствующего события; выходы — постусловиями. Возникновение события моделируется запуском соответствующего перехода. Выполнение условия представляется фишкой в позиции, соответствующей этому условию. Запуск перехода удаляет фишки, представляющие выполнение предусловий и образует новые фишки, которые представляют выполнение постусловий.



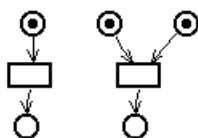
### 3.2. Одновременность и конфликт.

Важная особенность сетей Петри — это их *асинхронная* природа. В сетях Петри отсутствует измерение времени. В них учитывается лишь важнейшее свойство времени – частичное упорядочение событий.

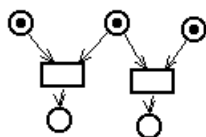
Выполнение сети Петри (или поведение моделируемой системы) рассматривается здесь как *последовательность* дискретных событий, которая является одной из возможных. Если в какой-то момент времени разрешено более одного перехода, то любой из них может стать «следующим» запускаемым.

Переходы в сети Петри, моделирующей некоторую систему, представляют её *примитивные* события (длительность которых считается равной 0), и в один момент времени может быть запущен только один разрешённый переход.

Моделирование одновременного (параллельного) возникновения независимых событий системы в сети Петри демонстрируется на следующем рисунке.



В этой ситуации два перехода являются разрешенными и не влияют друг на друга в том смысле, что могут быть запущены один вслед за другим в любом порядке.



Другая ситуация в приведённой ниже сети Петри.

Эти два разрешённые перехода находятся в *конфликте*, т. е. запуск одного из

них удаляет фишку из общей входной позиции и тем самым запрещает запуск другого. Таким образом, моделируются взаимоисключающие события системы.

### **3.3. Моделирование параллельных систем взаимодействующих процессов.**

#### **3.3.1. Моделирование последовательных процессов.**

Вырожденным случаем параллельной системы процессов является система с одним процессом. Сначала рассмотрим, как сетью Петри может быть представлен отдельный процесс. Отдельный процесс описывается программой на одном из существующих языков программирования.

Последовательная программа на абстрактном языке программирования, вычисляющая  $Y!$  и произведение всех чётных чисел из отрезка  $[1, Y]$  для произвольного положительного целого  $Y$ .

```
begin
  read(Y);
  X1:=1;
  X2:=1;
  while Y>0 do
    begin
      if mod(Y,2)=0
      then begin
        X1:=X1*Y;
      end;
      X2:=X2*Y;
      Y:=Y-1;
    end;
  write(X1);
  write(X2);
end
```

Программа представляет два различных аспекта процесса: вычисление и управление. Сети Петри удачно представляют структуру и управление программ. Они предназначены для моделирования упорядочения действий и потока информации, а не для действительного вычисления самих значений.

Стандартный способ представления структуры программы и потока управления в ней — это *блок-схемы*, которые в свою очередь могут быть представлены сетями Петри. Блок-схема программы состоит из узлов двух типов (принятия решения, обозначаемых ромбами, и вычисления, обозначаемых прямоугольниками) и дуг между ними.

Блок-схема программы

a: read(Y); X1:=1; X2:=1;

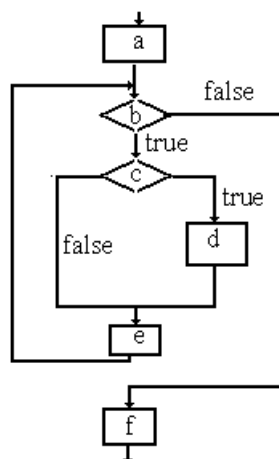
b: Y>0

c: mod(Y,2)=0

d: X1:=X1\*Y;

e: X2:=X2\*Y; Y:=Y-1;

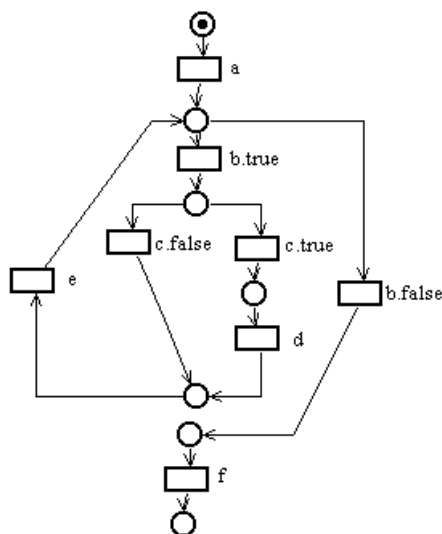
f: write(X1); write(X2);



В сети Петри, моделирующей блок-схему, узлы блок-схемы представляются переходами сети Петри как показано ниже, а дуги блок-схемы — позициями сети Петри.



Сеть Петри, представляющая блок-схему программы



Фишка в сети Петри представляет счетчик команд блок-схемы.

### 3.3.2. Моделирование взаимодействия процессов.

Параллельная система может строиться несколькими способами. Один из способов состоит в простом объединении процессов, без взаимодействия во время их одновременного выполнения. Так, например, если система строится этим способом из двух процессов, каждый из которых может быть представлен сетью Петри, то сеть Петри моделирующая одновременное выполнение двух процессов, является простым объединением сетей Петри для каждого из двух процессов. Начальная маркировка составной сети Петри имеет две фишки, по одной в каждой сети, представляя первоначальный счетчик команд процесса.

Такой способ введения параллелизма имеет низкое практическое значение. Далее будем рассматривать параллельные системы процессов, допускающие взаимодействие процессов во время их параллельного выполнения.

Существуют различные виды взаимодействия (синхронизации) процессов, в том числе: взаимодействие посредством общей памяти; - посредством передачи сообщения различных видов.

*Таким образом, для моделирования сетями Петри параллельных систем процессов, помимо последовательных процессов, необходимо уметь моделировать различные механизмы взаимодействия (синхронизации) процессов.*

Далее покажем, как сети Петри могут моделировать различные механизмы синхронизации процессов, на основе решения с помощью сетей Петри ряда задач, ставших классическими в области синхронизации.

### **3.3.3. Задача о взаимном исключении.**

Пусть несколько процессов разделяют общую переменную, запись, файл или другой элемент данных. Для обновления разделяемого элемента данных процесс должен сначала считать старое значение, затем вычислить новое и, наконец, записать его на то же место. Если два процесса  $P_1$  и  $P_2$  в одно и то же время пытаются выполнить такую последовательность действий, то могут возникнуть искажение данных.

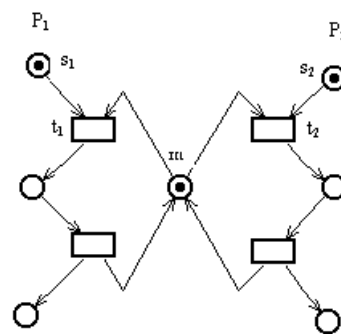
Например, возможна следующая последовательность:

1. Процесс  $P_1$  считывает значение  $x$  из разделяемого объекта;
  2. Процесс  $P_2$  считывает значение  $x$  из разделяемого объекта;
  3. Процесс  $P_1$  вычисляет новое значение  $x'=f(x)$ ;
  4. Процесс  $P_2$  вычисляет новое значение  $x''=g(x)$ ;
  5. Процесс  $P_1$  записывает  $x'$  в разделяемый объект;
  6. Процесс  $P_2$  записывает  $x''$  в разделяемый объект, уничтожая значение  $x$
- Результат вычисления процесса  $P_1$  потерян.

Для исключения подобных проблем используется метод взаимного исключения, основанный на понятии *критическая секция*. Критическая секция – это участок кода процесса, на котором он осуществляет доступ к разделяемому объекту данных. Прежде, чем выполнить свою критическую секцию, процесс

ждёт, пока другой процесс не закончит выполнение собственной критической секции (если такое выполнение имеет место). Затем он входит в критическую секцию и блокирует доступ для любого другого процесса к своей критической секции. После выполнения процессом критической секции деблокируется доступ для других процессов к разделяемому объекту данных.

Следующая ниже сеть Петри моделирует механизм взаимного исключения для двух процессов  $P_1$  и  $P_2$ . Она легко обобщается на произвольное число



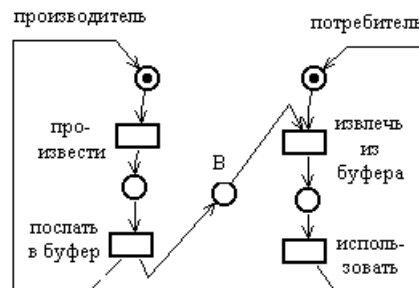
процессов.

Позиция  $m$  представляет условие «критическая секция свободна», разрешающее вход в критическую секцию. Попытка процесса  $P_1$  ( $P_2$ ) войти в критическую секцию осуществляется после помещения фишки в его позицию  $s_1$  ( $s_2$ ). Такая попытка может увенчаться успехом, если в позиции  $m$  содержится фишка. Если оба процесса пытаются войти в критическую секцию одновременно, то переходы  $t_1$  и  $t_2$  вступят в конфликт, и только один из них сможет запуститься. Запуск  $t_1$  запретит запуск перехода  $t_2$ , вынуждая процесс  $P_2$  ждать, пока процесс  $P_1$  выйдет из своей критической секции, и возвратит фишку обратно в позицию  $m$ .

#### 3.3.4. Задача о производителе/потребителе.

В задаче о производителе/потребителе также присутствует разделяемый объект – буфер, посредством которого реализуется взаимодействие через асинхронную передачу сообщений. Процесс-производитель создает сообщения, которые помещаются в буфер. Потребитель ждет, пока сообщение не будет помещено в буфер, извлекает его оттуда и использует. Такое взаимодействие

может быть промоделировано следующей ниже сетью Петри.



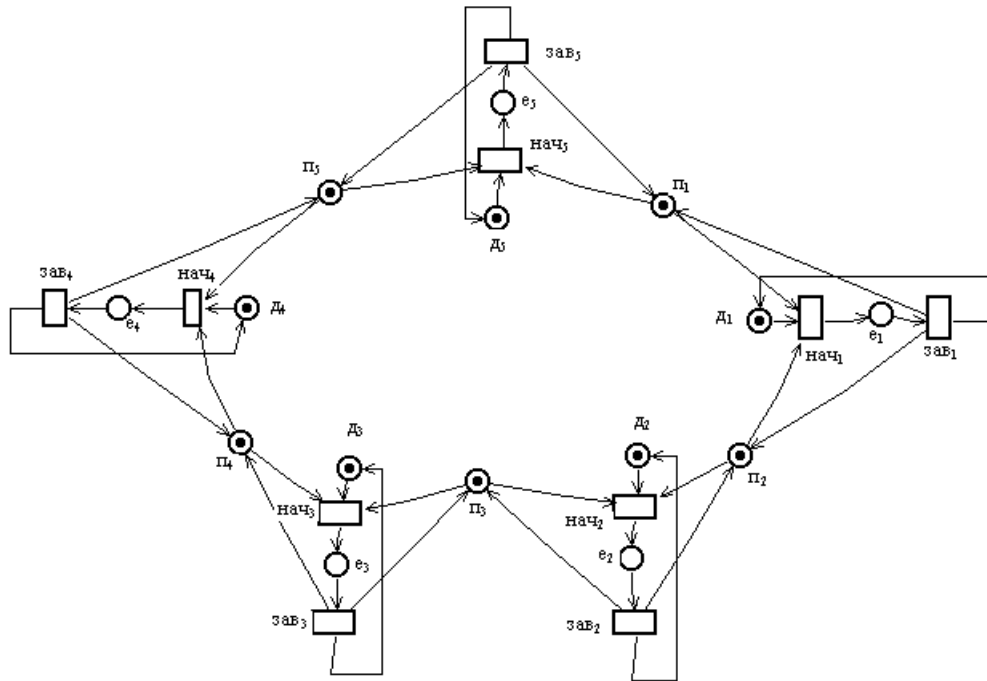
Позиция *В* представляет буфер, каждая фишка соответствует сообщению, которое произведено, но еще не использовано.

### 3.3.5. Задача об обедающих мудрецах.

Задача об обедающих мудрецах была предложена Дейкстрой и связана с пятью мудрецами, которые попеременно то думали, то ели. Мудрецы сидят за круглым столом, на котором много блюд китайской кухни. Между соседями лежит одна палочка для еды. Для приема китайской пищи необходимо две палочки. Поэтому каждый мудрец должен сначала взять палочку слева и палочку справа, а затем приступить к еде. Возможна ситуация, в которой каждый мудрец возьмёт палочку слева, а затем будет ждать, когда освободится палочка с правой стороны. Так они будут ждать, пока не умрут от голода. Тем самым, это состояние системы «обедающие мудрецы» является *тупиковым*.

Проблема тупика в этой системе может быть решена путём следующей модификации её правил поведения: Пусть мудрец при переходе из состояния размышления в состояние приёма пищи захватывает, не по очереди, а одновременно обе палочки (слева и справа), если они свободны. Следующая ниже сеть Петри моделирует такую модифицированную систему обедающих мудрецов, свободную от тупиков.

В этой сети Петри позиция  $\pi_i$ ,  $i \in \{1, 2, 3, 4, 5\}$ , представляет условие « $i$ -тая палочка свободна». В начальной маркировке каждая из этих позиций имеет фишку. Каждому мудрецу  $i \in \{1, 2, 3, 4, 5\}$  соответствует две позиции: позиция  $d_i$  – представляющая условие « $i$ -тый мудрец думает»; и позиция  $e_i$  – представляющая условие « $i$ -тый мудрец ест». В начальной маркировке каждая



позиция  $d_i$  содержит фишку, а каждая позиция  $e_i$  пуста.

Каждому мудрецу  $i \in \{1, 2, 3, 4, 5\}$  также соответствует два перехода: **переход  $nach_i$**  – представляющий событие «начало приёма пищи  $i$ -тым мудрецом»; и переход  **$zav_i$**  – представляющий событие «завершение приёма пищи  $i$ -тым мудрецом».

#### 4. АНАЛИЗ СЕТЕЙ ПЕТРИ

Моделирование систем сетями Петри, прежде всего, обусловлено необходимостью проведения глубокого исследования их поведения. Для проведения такого исследования необходимы методы анализа свойств самих сетей Петри. Этот подход предполагает сведения исследования свойств реальной системы к анализу определённых свойств моделирующей сети Петри.

## 4.1. Свойства сетей Петри.

### Активность.

*Тупик* в сети Петри — это переход (или множество переходов), которые не могут быть запущены. В связи с понятием тупика определим для сети Петри  $N$  с начальной маркировкой  $\mu$  следующие уровни активности переходов:

**Уровень 0:** Переход  $t$  обладает активностью уровня 0 и называется *мёртвым*, если он никогда не может быть запущен.

**Уровень 1:** Переход  $t$  обладает активностью уровня 1 и называется *потенциально живым*, если существует такая  $\mu' \in R(N, \mu)$ , что  $t$  разрешён в  $\mu'$ .

**Уровень 2:** Переход  $t$ , обладает активностью уровня 2 и называется *живым*, если для всякой  $\mu' \in R(N, \mu)$  переход  $t$  является потенциально живым для сети Петри  $N$  с начальной маркировкой  $\mu'$ .

Сеть Петри называется *живой*, если все её переходы являются живыми.

### Достижимость и покрываемость.

**Задача достижимости:** Для данной сети Петри с маркировкой  $\mu$  и маркировки  $\mu'$  определить:  $\mu' \in R(N, \mu)$ ?

**Задача покрываемости.** Для данной сети Петри  $N$  с начальной маркировкой  $\mu$  и маркировки  $\mu'$  определить, существует ли такая достижимая маркировка  $\mu'' \in R(N, \mu)$ , что  $\mu'' \geq \mu'$ .

(Отношение  $\mu'' \geq \mu'$  истинно, если каждый элемент маркировки  $\mu''$  не меньше соответствующего элемента маркировки  $\mu'$ .)

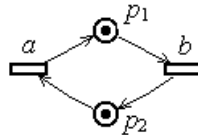
## 4.2. Методы анализа.

Особый интерес вызывают методы анализа свойств сетей Петри, которые обеспечивают автоматический анализ моделируемых систем. Рассмотрим метод анализа сетей Петри, который основан на использовании *дерева достижимости*.

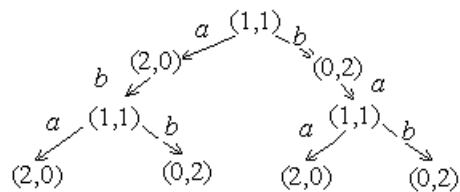
### Дерево достижимости.

Дерево достижимости представляет все достижимые маркировки сети Петри, а также – все возможные последовательности запусков её переходов.

Частичное дерево достижимости маркированной сети Петри. Сеть Петри имеет вид:



Частичное дерево достижимости для трёх шагов построения имеет вид:



Для сети Петри с бесконечным множеством достижимых маркировок дерево достижимости является бесконечным. Сеть Петри с конечным множеством достижимых маркировок также может иметь бесконечное дерево достижимости. Для превращения бесконечного дерева в полезный инструмент анализа строится его конечное представление. При построении конечного дерева достижимости для обозначения бесконечного множества значений маркировки позиции используется символ  $\omega$ .

## **ЛИТЕРАТУРА**

1. Дж. Питерсон. Теория сетей Петри и моделирование систем. М. Мир, 1984.
2. Котов В.Е. Сети Петри. М.: Наука, 1984.