ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
ОБНИНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ АТОМНОЙ ЭНЕРГЕТИКИ (ИАТЭ)

Факультет кибернетики

С.Ю. Цыкунова, Е.Н. Алонцева

# МЕТОДЫ СЕТЕВОГО МОДЕЛИРОВАНИЯ

Лабораторный практикум по курсу
«Теоретические основы
автоматизированного управления»

*Рекомендовано к изданию*
*Редакционно-издательским советом университета*

Обнинск 2007

**УДК 681.5.01(076.5)**

**Цыкунова С.Ю., Алонцева Е.Н.** Методы сетевого моделирования: Лабораторный практикум по курсу «Теоретические основы автоматизированного управления». – Обнинск: ИАТЭ, 2007. – 40 с.

В пособии изложены основы двух методов сетевого моделирования: метода сетевого графика планирования работ и метода построения моделей систем с помощью сетей Петри. Приведены теоретические сведения, изложен порядок выполнения и рассмотрены примеры лабораторных работ.

Для студентов специальности 230102 «Автоматизированные системы обработки информации и управления».

Илл. 10, табл. 6, библиограф. 6 назв.

Рецензенты: д.т.н., проф. В.А. Острейковский,
к.ф.-м.н. В.А. Чепурко

Темплан 2007, поз. 18

## Введение

При выполнении сложного проекта (например, создание автоматизированной системы управления) на каждом из этапов могут приниматься ошибочные решения, которые при накоплении и усилении на каждой последующей стадии выполнения проекта, могут привести к необходимости пересмотра всего проекта. Для предотвращения таких ситуаций используются методы моделирования будущей системы и методы управления проектами, одним из которых является метод сетевого планирования. В основе метода лежит формальное графическое представление комплекса взаимосвязанных работ в виде сетевого графика.

Сетевое планирование – это средство руководителей, технических администраторов, экономистов, менеджеров и инженеров для осуществления контроля и переработки планов, ресурсов и технических результатов, направленное на выполнение в заданное время определенного комплекса работ. Построение, анализ и расчет параметров сетевых графиков позволяют оптимизировать процесс разработки проекта по различным критериям (например, по структуре, времени, стоимости и т.п.).

Методы сетевого моделирования применяются также для анализа и синтеза информационных и управляющих систем. Это объясняется сочетанием простоты графового представления достаточно сложных моделей и возможности их аналитического исследования. Среди сетевых моделей известны сети Петри.

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, которая может использоваться для изучения функционирования системы и выработки предложений по ее усовершенствованию.

При написании пособия использован опыт, полученный авторами во время проведения лабораторных работ по курсу «Теоретические основы автоматизированного управления» в течение нескольких лет. Пособие предназначено для студентов специальности 230102 «Автоматизированные системы обработки информации и управления».

# Работа №1. Расчет и оптимизация сетевых графиков

Цель работы: усвоение студентами методов сокращения срока выполнения проекта путем оптимальной расстановки трудовых ресурсов.

*Область применения сетевых графиков*

Многие крупные проекты, такие как строительство здания, создание предприятия, внедрение автоматизированной системы бухгалтерского учета и т.д., можно разбить на большое количество различных операций (работ). Некоторые из этих операций могут выполняться одновременно, другие – только последовательно: одна операция после окончания другой. Например, при создании автоматизированной системы управления предприятием совместимы во времени тестирование создаваемой системы и обучение персонала, но окончание этапа внедрения системы возможно только после завершения перечисленных работ.

Задачи планирования работ по осуществлению некоторого проекта состоят:

- в определении времени окончания проекта, а также сроков выполнения отдельных работ, образующих проект;
- в определении резервов времени для выполнения отдельных работ;
- в определении работ, задержка в выполнении которых ведет к задержке выполнения всего проекта;
- в управлении ресурсами.

*Основные понятия*

Сетевой график – это графическое изображение сетевой модели комплекса взаимосвязанных работ, направленных на достижение определенной заранее намеченной цели. Математически сетевая модель представляет собой ориентированный (направленный) линейный граф без петель и контуров, состоящий из конечного множества элементов: вершин и попарно соединяющих их дуг $(i, j)$.

Основными логическими элементами, участвующими в построении сетевого графика, являются работа и событие. Работа – это трудовой процесс, для выполнения которого требуются затраты времени и ресурсов. Такая работа называется действительной. Также в сетевом планировании могут встречаться работа-ожидание и работа-зависимость. Работа-ожидание – это процесс, требующий только затраты времени. Работа-зависимость или фиктивная работа – это логическая связь между работами, не предполагающая затрат времени и ресурсов, но указывающая, что начало выполнения некоторых работ возможно только после завершения других работ. На сетевом графике действительные работы и ожидания изображают сплошными линиями, а фиктивные работы – пунктирными.

Событие представляет собой факт окончания всех предшествующих данному событию работ и возможность начала других работ, непосредственно следующих за данным событием. Событие наступает в момент времени, соответствующий завершению всех работ, входящих в него. В сетевых графиках события изображают обычно кружком с цифровым индексом внутри, обозначающим порядковый номер события. Последнее событие означает завершение всего комплекса работ или проекта.

События являются связующими звеньями между работами. Каждая работа сетевого графика определяется двумя событиями: предшествующим, которое указывает на начало работы и обозначается *i*, и последующим, указывающим на окончание работы и обозначаемым *j*. Работа обозначается индексом $(i, j)$, а продолжительность ее выполнения (временная оценка $t(i, j)$) выражается в единицах времени.

В сетевом графике существуют два особых события:

– исходное, соответствующее началу выполнения работ и обозначаемое *0*;

– завершающее, отображающее достижение намеченной цели данного комплекса работ и обозначаемое последним порядковым номером *n*.

В сетевом планировании важное значение имеет путь сетевого графика – это любая непрерывная последовательность взаимосвязанных событий и работ сетевого графика по направлению стрелок.

Различают два типа пути сетевого графика: путь между событиями $L(i, k)$, т.е. путь, соединяющий два события *i* и *k*, и полный путь сетевого графика $L(0,n)$. Если известна продолжительность каждой работы сетевого графика $t(i,j)$, то длина любого пути определяется суммой продолжительностей составляющих его работ, т.е.

$$t(L)=\sum_{i,j\in L} t(i,j).$$

Полный путь сетевого графика с наибольшей продолжительностью называется критическим. Длительность критического пути $t_{кр}$ – это время, необходимое для завершения всего комплекса работ

$$t_{кр}=\max_{L\in G}\{t[L]\},$$

где *G* – граф, представляющий собой сетевой график.

*Параметры сетевого графика*

Для каждого события $i, i\in\{0,n\}$ сетевого графика можно вычислить ранний срок наступления события $T_p(i)$, поздний срок наступления события $T_n(i)$ и резерв времени события $P(i)$.

Ранний срок свершения события $T_p(i)$ – это такой момент времени, раньше которого событие не может произойти; $T_p(i)$ равен продолжительности максимального из путей от *0* до *i*-го события

$$T_p(i)=\max\{t[L(0-i)]\}=\max\{T_p(i-1)+t(i-1,i)\},$$

где $L(0-i)$ – путь от *0*-го до *i*-го события; $T_p(i-1)$ – ранний срок наступления предыдущего события; $t(i-1,i)$ – продолжительность работы между событиями $(i-1)$ и *i*.

Раньше этого срока событие наступить не может, но возможно наступление его позже, без увеличения общей длительности выполнения всего комплекса работ сетевого графика. Чтобы вычислить насколько позже может наступить событие, вычисляют поздний срок наступления события $T_n(i)$

$$T_n(i)=t_{кр}-\max\{t[L(i-n)]\}=\min\{T_n(i+1)-t(i,i+1)\},$$

где $L(i-n)$ – путь от *i*-го до *n*-го события; $T_n(i+1)$ – поздний срок наступления следующего за *i* события; $t(i,i+1)$ – продолжительность работы между событиями *I* и $(i+1)$.

При этом $T_n(0)=T_p(0)=0$ и $T_n(n)=T_p(n)=t_{кр}$.

Очевидно, что для событий, принадлежащих критическому пути $T_n(i)=T_p(i)$. Все события сетевого графика, не принадлежащие критическому пути, обладают резервом времени события

$$P(i)=T_n(i)-T_p(i).$$

Для каждой работы $(i,j)$ с продолжительностью $t(i,j)$ можно определить следующие параметры:

1) ранний возможный срок начала работы $t_{рн}(i,j)=T_p(i)$;

2) поздний допустимый срок начала $t_{пн}(i,j)=T_n(j)-t(i,j)$;

3) ранний возможный срок окончания $t_{ро}(i,j)=T_p(i)+t(i,j)$;

4) поздний допустимый срок окончания $t_{по}(i,j)=T_n(j)$;

5) полный резерв времени работы $P_n(i,j)=T_n(j)-T_p(i)-t(i,j)$;

6) свободный резерв времени работы

$$P_c(i,j)=T_p(j)-T_n(i)-t(i,j).$$

Полные резервы времени для работ некоторого пути $L_i$ не могут быть использованы одновременно для всех работ. Полное или частичное использование этого резерва на какой-либо работе приводит к уменьшению резерва времени связанных с этой работой событий и, следовательно, к снижению полного резерва времени других работ этого пути.

Свободный резерв $P_c(i,j)$ показывает, насколько можно увеличивать продолжительность $t(i,j)$ работы $(i,j)$, не влияя на любые другие работы сети. Свободный резерв может принимать отрицательное значение. Формально отрицательное значение $P_c(i,j)$ свидетельствует об отсутствии свободного резерва. Фактически это означает, что если данная работа будет начата в свой поздний срок, то последующее событие не сможет произойти в свой ранний срок.

Для работ критического пути полный и свободный резервы времени равны нулю.

*Понятие оптимального сетевого графика*

В данный момент нет формализованной (а тем более автоматизированной) процедуры построения оптимального сетевого графика. Поэтому перед специалистом возникает задача оптимизации уже существующей сети. Можно выделить два пути оптимизации сетевого графика. Первый – оптимизация по структуре. Анализ взаимосвязи между работами и событиями позволяет выявить возможность изменения последовательности работ или выполнения каких-то работ параллельно по времени. Второй путь – оптимизация по времени, т.е. существует возможность сокращения планируемого срока разработки путем уменьшения продолжительности критического пути при перераспределении ресурсов, имеющихся у работ с большими резервами времени, или при привлечении дополнительных ресурсов.

Об оптимальности сетевого графика по времени можно судить по следующим признакам:

- свободные резервы времени работ близки к 0;
- полные резервы времени работ имеют небольшие значения;
- длительности параллельных работ и путей близки по значениям;
- полные пути сетевого графика близки по длительности.

По последнему признаку можно ввести формальный критерий оптимальности – коэффициент напряженности *К*

$$K = \frac{t_{\min}}{t_{кр}},$$

где $t_{\min}$ – длительность наикратчайшего пути; $t_{кр}$ – длительность критического пути.

Наикратчайший путь сетевого графика – это полный путь сетевого графика с наименьшей продолжительностью.

Очевидно, что чем ближе коэффициент *К* к единице, тем более оптимальным можно считать сетевой график.

*Порядок выполнения работы и содержание отчета*

Пусть в виде сетевого графика представлен план выполнения проекта по разработке и внедрению АСУП.

Каждому студенту выдается сетевой график, где над дугами указаны объемы работ в единицах (*d*), а под дугами – число людей (*r*). Производительность труда (*e*) принимается одинаковая для всех работ. Длительность работы рассчитывается как

$$t(i, j) = \frac{d}{r \cdot e} \ .$$

Для упрощения оптимизации сетевого графика считаем все работы действительными, одинаковыми по трудоемкости и равными по квалификации. Каждый специалист, занятый в проекте, может участвовать в любой работе.

*Порядок выполнения работы*

1. Расчет сетевого графика:

– изучить основные понятия и параметры сетевого графика;

– рассчитать параметры сетевого графика и представить их в виде таблицы;

– ответить на контрольные вопросы.

2. Оптимизация сетевого графика:

– предложить алгоритм оптимизации сетевого графика;

– написать программу для оптимизации сетевого графика;

– представить основные шаги оптимизации и конечный результат в виде таблицы.

*Содержание отчета*

1. Цель работы и исходные данные.

2. Краткое изложение методики расчета параметров сетевого графика.

3. Результаты расчета параметров сетевого графика в виде таблицы.

4. Сетевой график с рассчитанными параметрами.

5. Краткое изложение алгоритма оптимизации сетевого графика.

6. Представление основных шагов оптимизации и конечного результата в виде таблицы.

7. Вывод (доказательство) об оптимальности полученного сетевого графика.

*Контрольные вопросы*

1. Что такое сетевой график?
2. Для каких задач используются сетевые графики?
3. Что такое работа-ожидание и фиктивная работа, в чем их предназначение?
4. Как определяется срок выполнения проекта?
5. Что такое критический путь сетевого графика?
6. Какими параметрами характеризуются события сети? Как они вычисляются и что означают с точки зрения выполнения проекта?
7. Какими параметрами характеризуются работы сети? Как они вычисляются и что означают с точки зрения выполнения проекта?
8. Как вычисляются резервы времени работ?
9. Чем отличаются полный и свободный резервы времени работ?
10. Какие показатели характеризуют оптимальность сетевого графика?
11. Признаки оптимального по длительности сетевого графика.

*Пример расчета сетевого графика и алгоритма оптимизации*

Пусть дан сетевой график (рис.1). Необходимо расставить трудовые ресурсы таким образом, чтобы время выполнения проекта максимально сократилось.

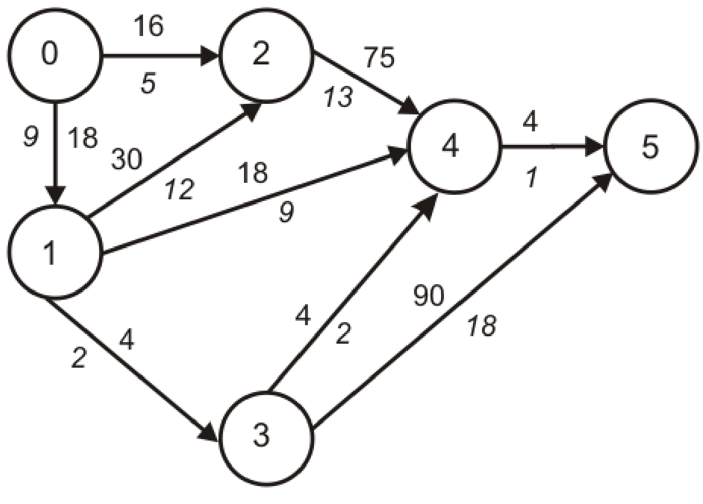



Рис.1. Пример варианта сетевого графика.

Рассчитаем параметры сетевого графика и представим их в табл.1-3.

Критический путь данного сетевого графика проходит через работы (0,1)-(1,2)-(2,4)-(4,5). Срок выполнения проекта равен 2,85 ед. времени. Из табл.2 видно, что работы (0,2), (1,4), (1,3), (3,4) и (3,5) имеют полные резервы времени, но на работах (1,3) и (3,4) свободные резервы равны нулю. Это значит, что если увеличить время выполнения этих работ, то они могут стать критическими. Следовательно, сократить количество занятых людей можно только на участках, где есть свободный резерв времени, т.е. на работах (0,2), (1,4) и (3,5). Чтобы максимально сократить время критического пути нужно увеличить количество людей, работающих на самом напряженном участке. По табл.2 видно, что самый напряженный участок на данный момент это работа (2,4).

Таблица 1

Параметры событий сетевого графика

| Событие, $i$ | $T_p(i)$ | $T_n(i)$ | $P(i)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0,4 | 0,4 | 0 |
| 2 | 0,9 | 0,9 | 0 |
| 3 | 0,8 | 1,65 | 0,85 |
| 4 | 2,05 | 2,05 | 0 |
| 5 | 2,85 | 2,85 | 0 |

Таблица 2

Параметры работ сетевого графика

| Работа | $d$ , ед. | $r$ , чел. | $t(i,j)$ | $P_n(i,j)$ | $P_c(i,j)$ |
|---|---|---|---|---|---|
| (0,1) | 18 | 9 | 0,40 | 0 | 0 |
| (0,2) | 16 | 5 | 0,64 | 0,26 | 0,26 |
| (1,2) | 30 | 12 | 0,50 | 0 | 0 |
| (2,4) | 75 | 13 | 1,16 | 0 | 0 |
| (4,5) | 4 | 1 | 0,80 | 0 | 0 |
| (1,4) | 18 | 9 | 0,40 | 1,25 | 1,25 |
| (1,3) | 4 | 2 | 0,40 | 0,85 | 0 |
| (3,4) | 4 | 2 | 0,40 | 0,85 | 0 |
| (3,5) | 90 | 18 | 1 | 1,05 | 0,2 |

Таблица 3

Параметры путей сетевого графика

| $L(0,n)$ | $T$, ед. |
|---|---|
| (0,2)-(2,4)-(4,5) | 2,59 |
| (0,1)-(1,2)-(2,4)-(4,5) | 2,85 |
| (0,1)-(1,4)-(4,5) | 1,60 |
| (0,1)-(1,3)-(3,4)-(4,5) | 2,00 |
| (0,1)-(1,3)-(3,5) | 1,80 |
| $T_{кр}$ | 2,85 |
| $t_{\min}$ | 1,6 |
| $K$ | 0,56 |

Пусть на каждом шаге оптимизации можно переставить только одного человека. Перестановку будем совершать с работы с наибольшим свободным резервом времени на работу критического пути с наибольшей длительностью. Перестановку будем производить до тех пор, пока длительность критического пути уменьшается.

Для данного сетевого графика понадобилось 10 шагов. Значения параметров после оптимизации приведены в табл. 4-6.

Таблица 4

Параметры событий сетевого графика после оптимизации

| Событие, $i$ | $T_p(i)$ | $T_n(i)$ | $P(i)$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0,40 | 0,40 | 0 |
| 2 | 0,90 | 0,92 | 0,02 |
| 3 | 0,80 | 0,80 | 0 |
| 4 | 1,60 | 1,60 | 0 |
| 5 | 2,00 | 2,00 | 0 |

Таблица 5

Параметры работ сетевого графика после оптимизации

| Работа | $d$ , ед. | $r$ , чел. | $t(i,j)$ | $P_n(i,j)$ | $P_c(i,j)$ |
|---|---|---|---|---|---|
| (0,1) | 18 | 9 | 0,4 | 0 | 0 |
| (0,2) | 16 | 4 | 0,8 | 0,12 | 0,1 |
| (1,2) | 30 | 12 | 0,5 | 0,02 | 0 |
| (2,4) | 75 | 22 | 0,68 | 0,02 | 0 |
| (4,5) | 4 | 2 | 0,4 | 0 | 0 |
| (1,4) | 18 | 3 | 1,2 | 0 | 0 |
| (1,3) | 4 | 2 | 0,4 | 0 | 0 |
| (3,4) | 4 | 1 | 0,8 | 0 | 0 |
| (3,5) | 90 | 16 | 1,13 | 0,08 | 0,08 |

Таблица 6

Параметры путей сетевого графика после оптимизации

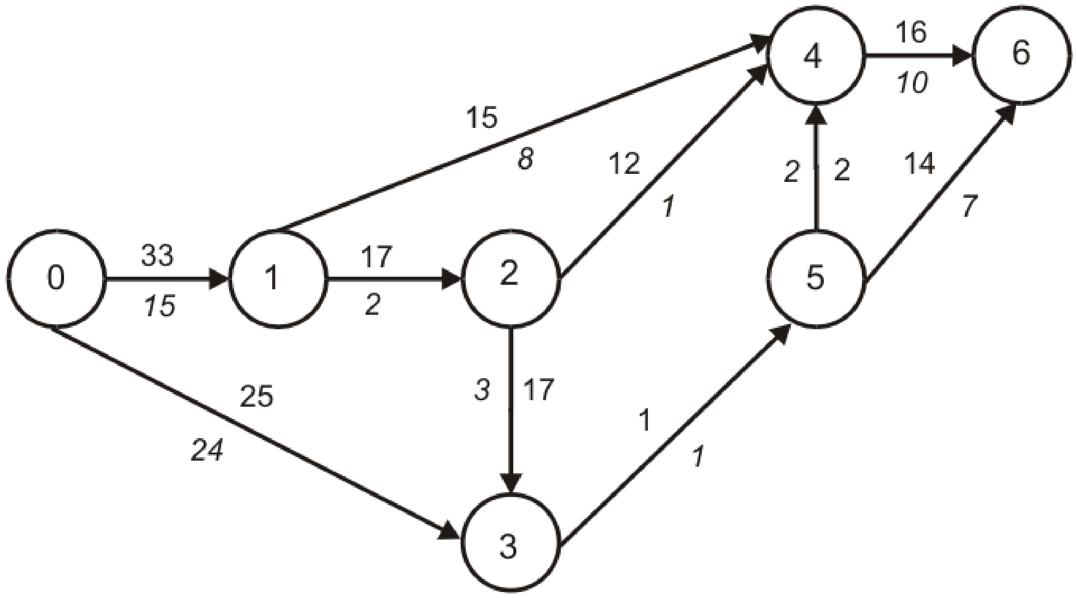
| $L(0,n)$ | $T$ , ед. |
|---|---|
| (0,2)-(2,4)-(4,5) | 1,88 |
| (0,1)-(1,2)-(2,4)-(4,5) | 1,98 |
| (0,1)-(1,4)-(4,5) | 2 |
| (0,1)-(1,3)-(3,4)-(4,5) | 2 |
| (0,1)-(1,3)-(3,5) | 1,93 |
| $T_{кр}$ | 2 |
| $t_{\min}$ | 1,88 |
| $K$ | 0,94 |

Проверим, соответствуют ли полученные значения параметров признакам оптимальности сетевого графика. Свободные резервы времени работ имеют малые значения (0,1 ед. времени), полные резервы времени работ значительно сокращены (с 1,25 до 0,12). Длительности полных путей сетевого графика близки по значениям, т.к. коэффициент напряженности близок к единице.

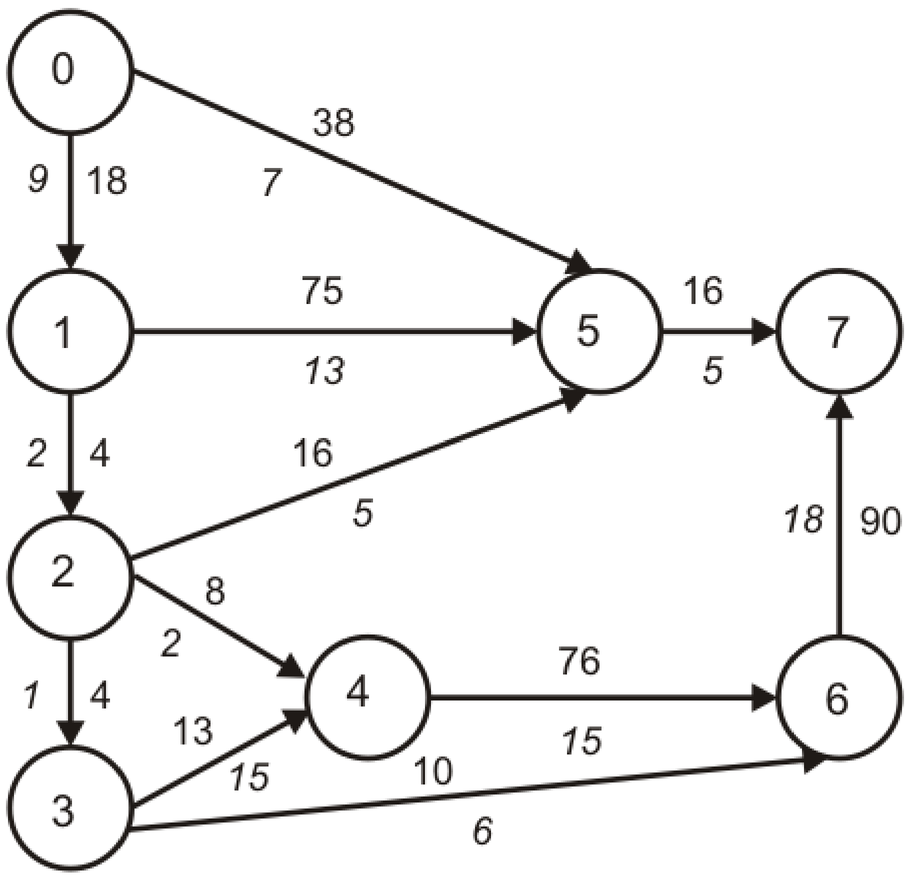
На основе анализа параметров можно сделать вывод, что сетевой график стал оптимальным по длительности. Время выполнения проекта сократилось на 30%.

*Варианты заданий*

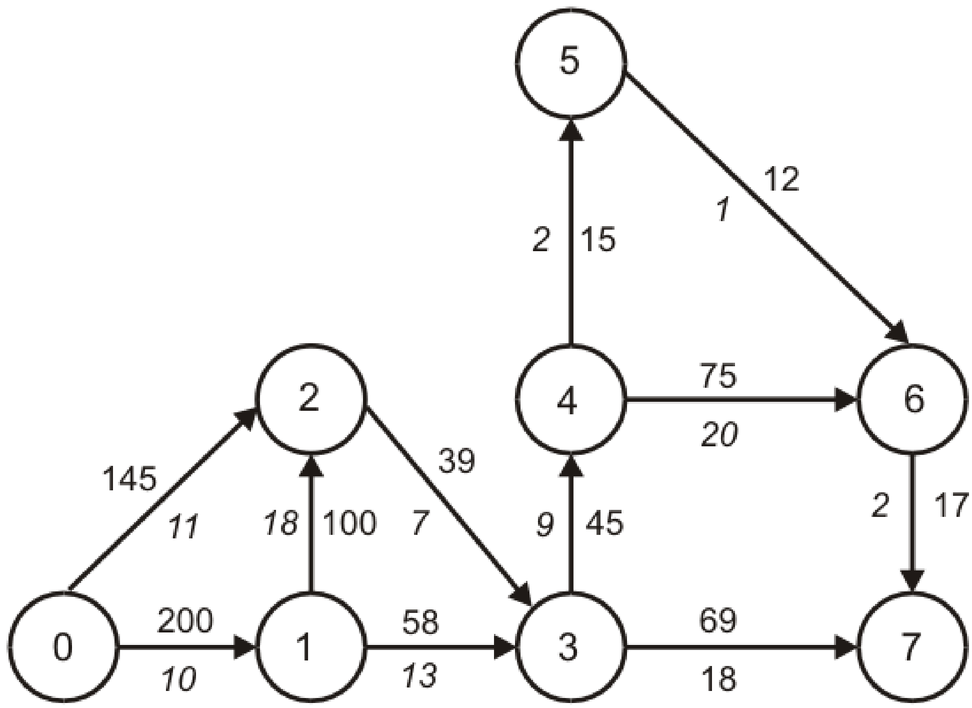
Вариант 1



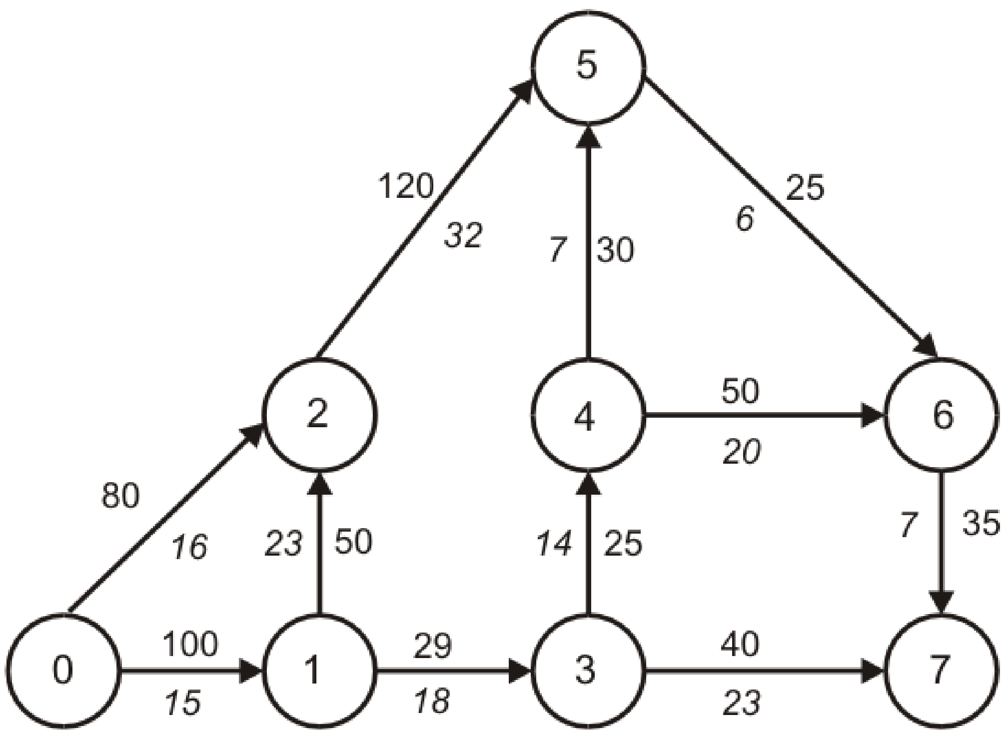

Вариант 2



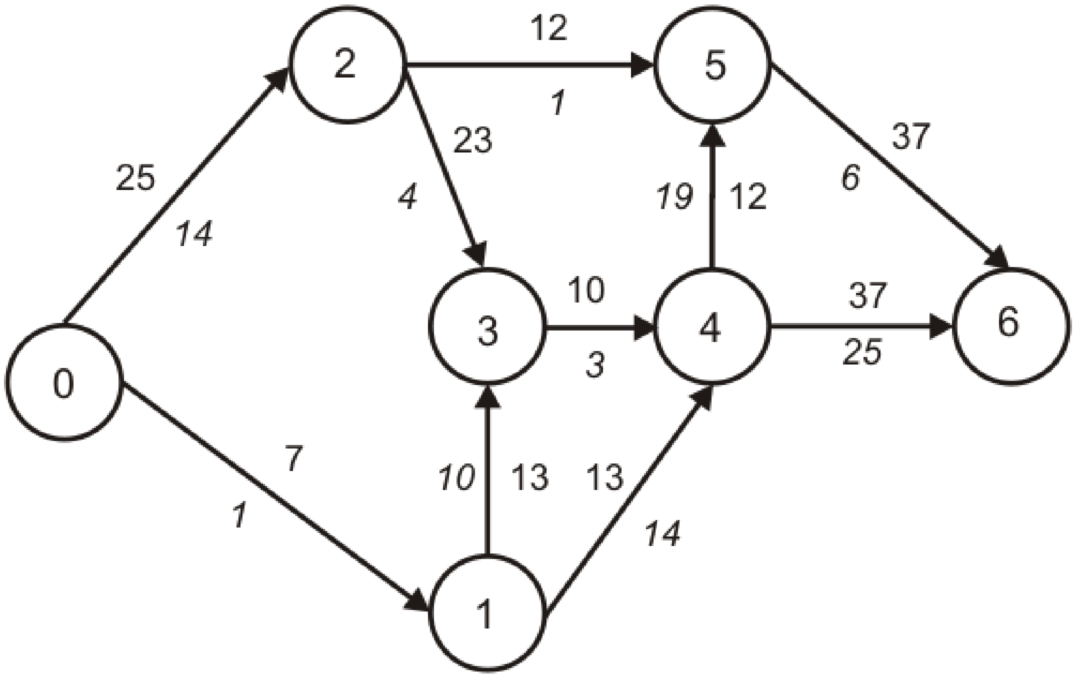
Вариант 3



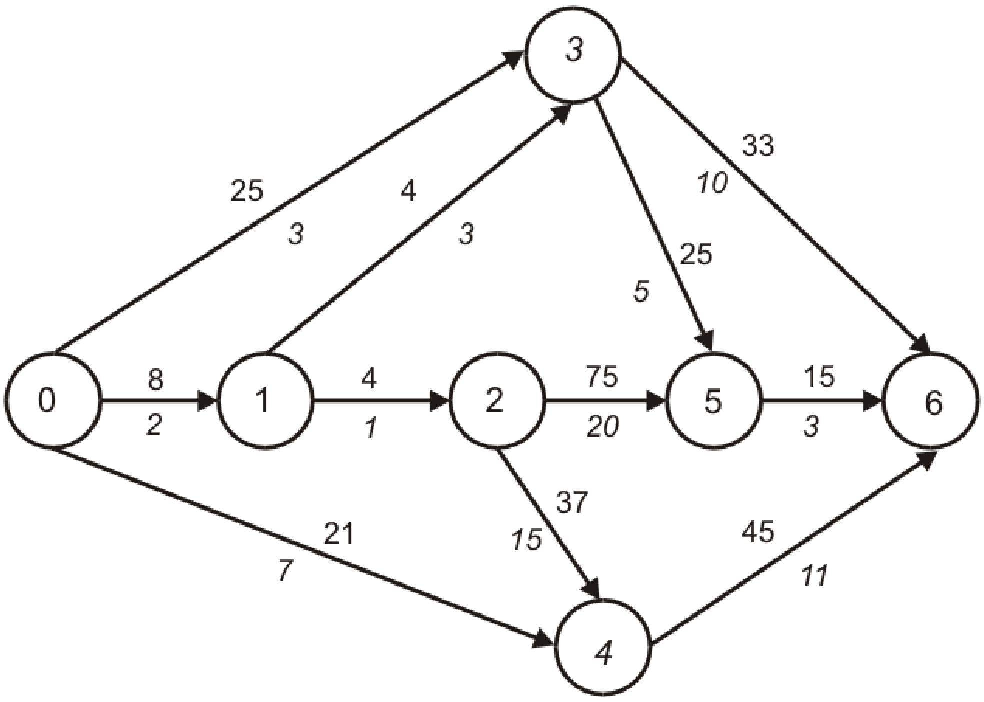

Вариант 4



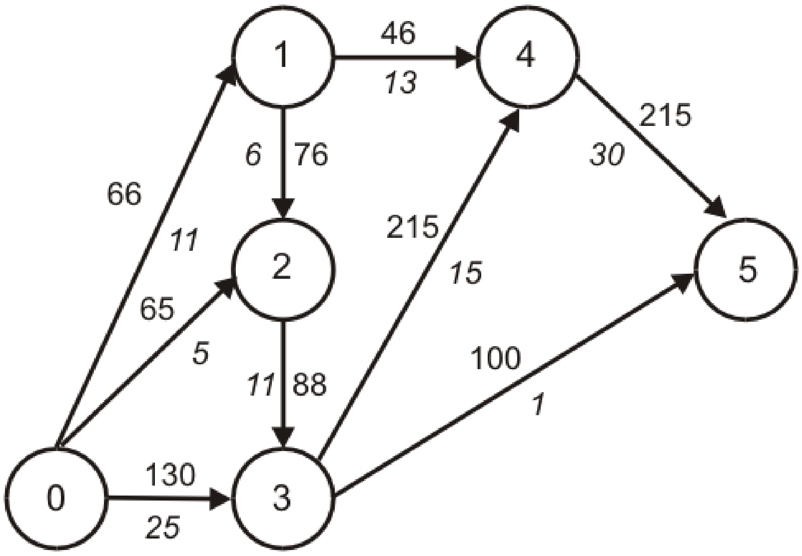
Вариант 5



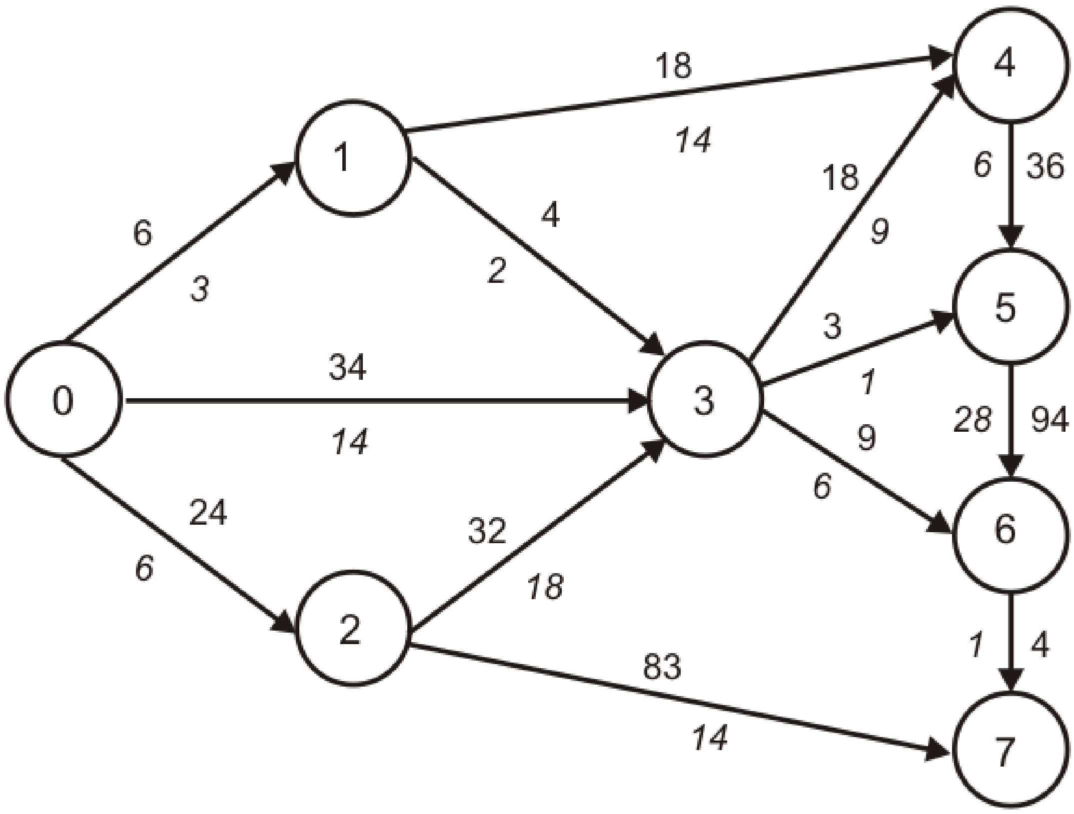

Вариант 6



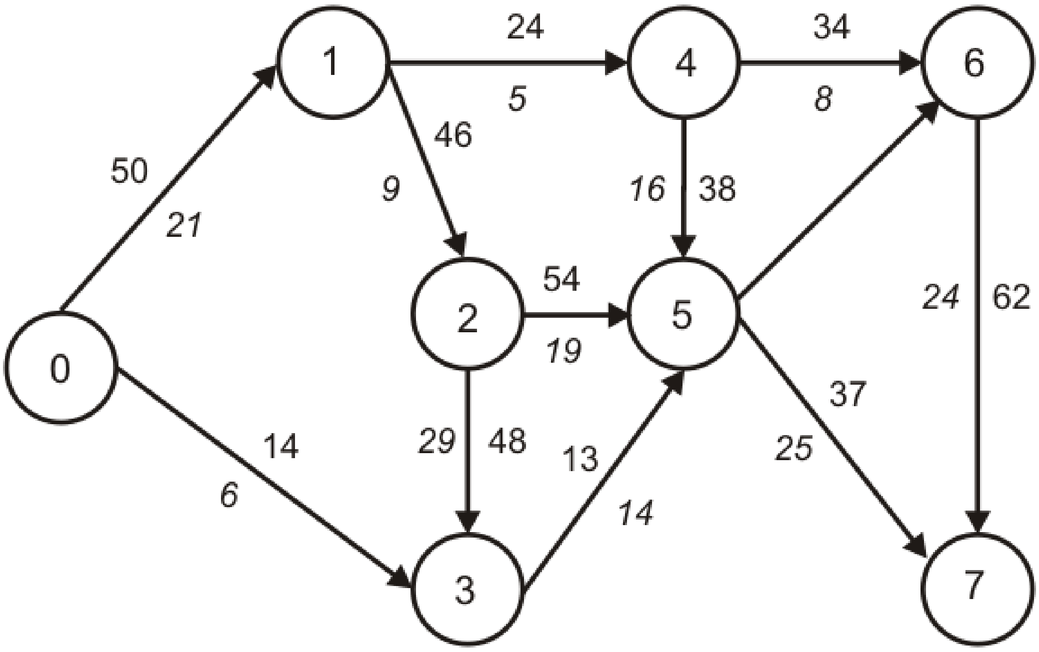
Вариант 7



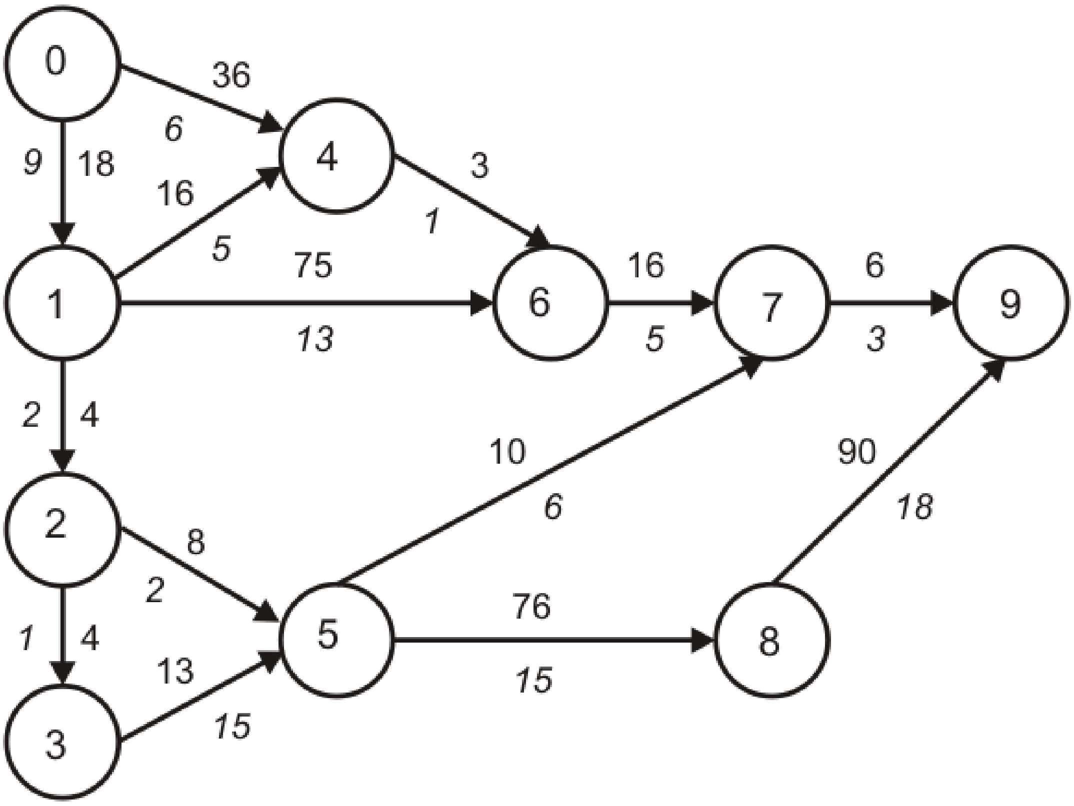

Вариант 8



Вариант 9




Вариант 10



Литература

1. Мамиконов А.Г. Основы построения АСУ: Учебник для вузов. – М.: Высшая школа, 1981. – 248 с.

2. Бахвалов Л.А., Мамиконов А.Г., Попов В.В. и др. Автоматизированные системы управления. Лабораторный практикум: Учеб. пособие для студентов вузов по спец. АСУ / Под ред. А.Г. Мамиконова. – М.: Высшая школа, 1985. – 95 с.

3. Исследование операций (в 2-х т.). Т. 2. Модели и применения / Под ред. Дж. Моудера, С. Элмаграби. – М.: Мир, 1981. – 677 с.

4. Архипова Н.И., Кульба В.В., Косяченко С.А., Чанхиева Ф.Ю. Исследование систем управления: Учеб. пособие для вузов. – М.: Издательство ПРИОР, 2002. – 384 с.

# Работа №2. Моделирование и анализ динамики системы (сети Петри)

Цель работы: усвоение студентами метода моделирования и анализа динамики функционирования системы на примере сети Петри.

## *Область применения сетей Петри*

Теория сетей Петри появилась на рубеже 60-70-х годов XX-го века как один из методов моделирования и анализа дискретных параллельных и распределенных систем. Сеть названа по имени немецкого математика К.А. Петри, который занимался сетями разного вида, близкими к современному понятию сетей Петри. Модель системы в виде сети Петри позволяет через причинно-следственные связи наглядно описать структурные особенности функционирования системы.

Моделирование будущей или существующей системы с помощью сети Петри позволяет ответить на ряд вопросов, которые обычно возникают при конструировании системы:

– выполняет ли система функции, для которых предназначена?

– функционирует ли она эффективно?

– могут ли в ней возникнуть ошибки и аварийные ситуации?

– имеются ли в ней потенциально узкие места?

– можно ли упростить систему или заменить ее отдельные компоненты на более совершенные?

– можно ли из данных систем сконструировать более сложную?

– и многие другие.

На данный момент теория сетей Петри содержит большое количество различных методов и средств анализа систем и применяется во многих отраслях вычислительной техники.

## *Основные понятия*

Компоненты системы и их действия представляются абстрактными *событиями*. Например, событием может быть исполнение оператора программы, выполнение операции на конвейерной линии или завершение этапа проекта. Событие может произойти один раз, повториться многократно, порождая конкретные *действия* (реализации события) или не произойти ни разу. Совокупность действий образует *процесс*, порождаемый этой системой.

Реальная система функционирует во времени, события происходят в некоторые моменты времени и длятся некоторое время. Однако в сетях Петри понятие времени носит качественный характер и отражает причинно-следственные связи между событиями в системе и состояние системы, т.е. Сети Петри являются асинхронными.

Взаимодействие событий в асинхронных системах имеет динамическую структуру. Взаимодействия описываются более просто, если указывать не связи между событиями, а ситуации, при которых это событие может реализоваться. При этом глобальные ситуации в системе формируются с помощью локальных операций, называемых *условиями* реализации событий. Условие имеет емкость: условие не выполнено (емкость равна 0), условие выполнено (емкость равна 1), условие выполнено с n-кратным запасом (емкость равна n). Сочетания условий соответствуют локальным состояниям системы. Условие соответствует таким ситуациям в моделируемой системе, как наличие данного для операции в программе, наличие детали на конвейере и т.п. Определенные сочетания условий позволяют реализоваться некоторому событию, а реализация события, в свою очередь, изменяет некоторые условия.

В сетях Петри события и условия представлены абстрактными символами двух непересекающихся алфавитов, называемых соответственно множеством *переходов T* и множеством *мест* (позиций) *P*. Графически места в сети Петри отображаются кружками, а переходы узкими прямоугольниками. Связи между местами и переходами изображаются в виде *направленных дуг*. Места, из которых ведут дуги на данный переход, называются *входными*. Места, на которые ведут дуги из данного перехода, называются *выходными*. На рис. 2.1*а* места $p_1$ и $p_2$ являются входными для перехода $t_1$, а место $p_3$ – выходным. Выполнение условия изображается *разметкой* соответствующего места с помощью помещения в кружок определенного числа *маркеров* (меток, фишек). Маркеры обозначаются точками. Каждое место может содержать от 0 до n маркеров (где n – целое число). Число n также обозначает емкость условия. Если маркеров много, то в кружок ставится число. Число маркеров обычно соответствует количеству единиц ресурса. Сеть Петри с разметкой называется маркированной сетью. На рис. 2.1*б* представлена маркированная сеть, где места $p_1$ и $p_2$ содержат по одному маркеру.

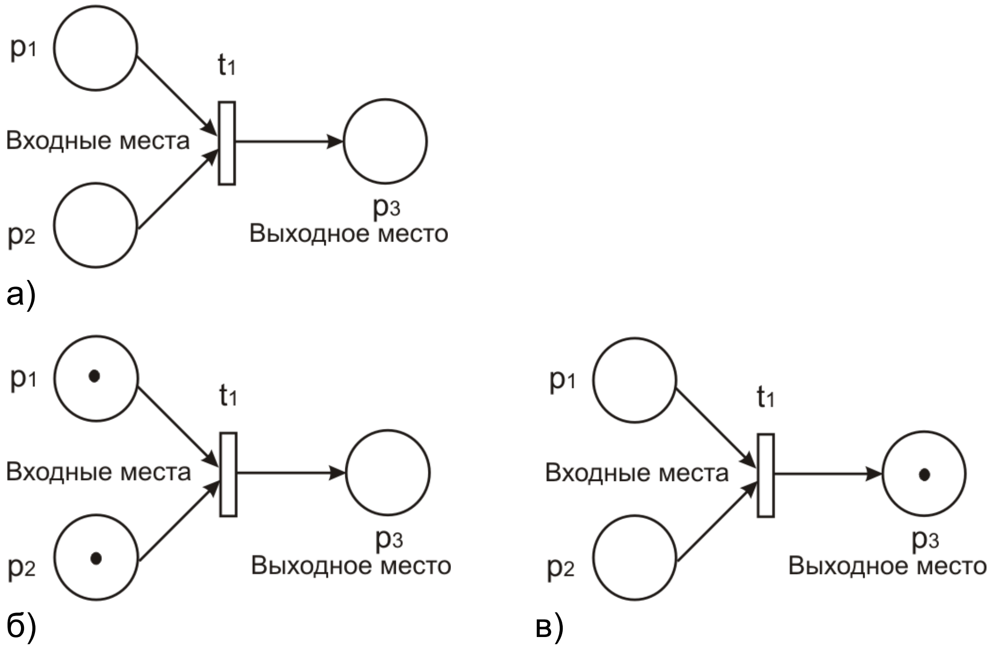
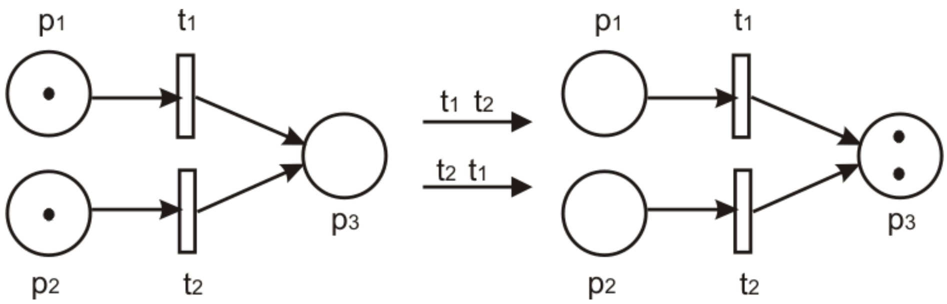



Рис.2.1. Пример сети Петри

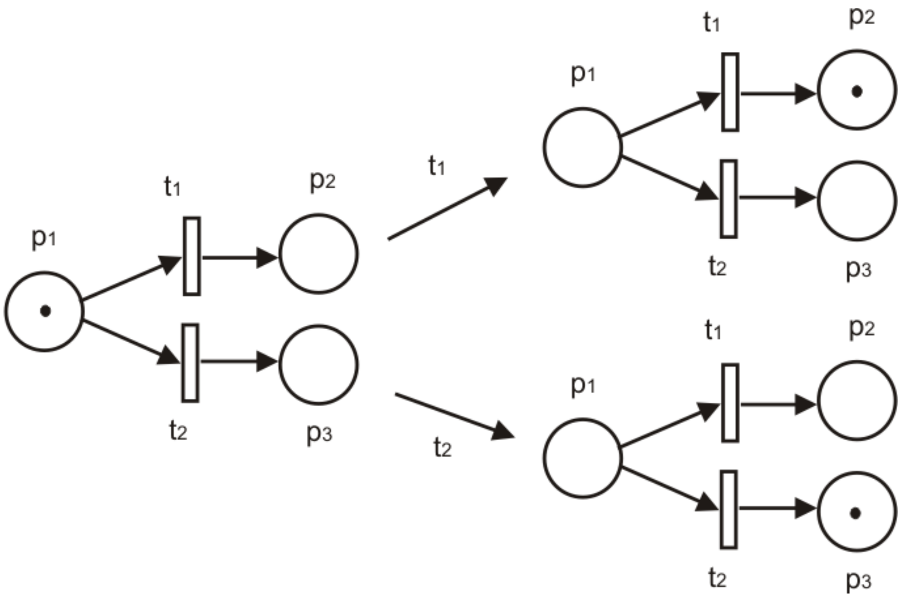
Динамика поведения моделируемой системы описывается множеством возможных последовательностей реализации событий, т.е. множеством срабатываний переходов. Срабатывание перехода – неделимое действие, изменяющее разметку его входных и выходных мест по следующему правилу: из каждого входного места изымается по маркеру и в каждое выходное место добавляется по одному маркеру. На рис. 2.1*в* показана сеть после срабатывания перехода $t_1$.

Различные варианты срабатывания переходов в зависимости от разметки показаны на рис.2.2. Если два и более перехода могут сработать, и они не имеют общих входных мест, то их срабатывания являются независимыми действиями, осуществляемыми в любой последовательности или параллельно (рис.2.2*а*).

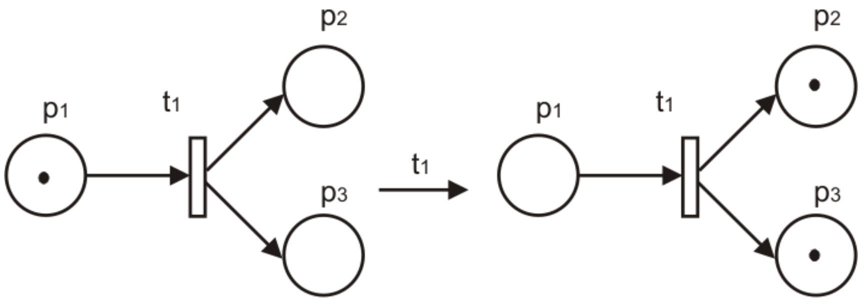
Если несколько переходов могут сработать и имеют общее входное место, то срабатывает только один, любой из них (рис. 2.2*б*). При этом, сработав, переход может лишить возможности сработать другие переходы. Таким способом в сети моделируется конфликт между событиями, когда реализация одного события исключает возможность реализации других. Такой переход назовем переходом типа «или» или альтернативным.



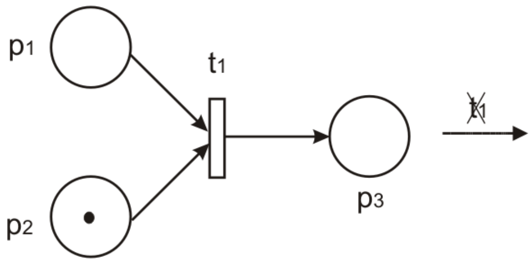

а) независимые срабатывания




б) альтернативные (конфликтные) срабатывания




в) переход типа «и»




г) срабатывание невозможно (тупиковая ситуация)

Рис. 2.2. Примеры срабатывания переходов в сети Петри

Для описания параллельных процессов используется переход типа «и», который означает параллельное использование ресурса. Например, один сотрудник числится в двух подразделениях («и там, и там»). С точки зрения отдела кадров это соответствует варианту на рис. 2.2*в*. Но в настоящий момент времени он может находиться или в первом или во втором подразделениях (рис. 2.2*б*).

Сеть останавливается, если ни один переход не может сработать. Переход не может сработать, если условия реализации события, представленного переходом, не выполняются. Такое состояние сети называется тупиковой ситуацией. В сети на рис.2.2*г* для срабатывания перехода не хватает маркера на месте $p_1$, условия перехода не выполнены и возникает тупиковая ситуация. Например, для выполнения работы нужно два сотрудника, а на месте находится только один.

В сети Петри есть понятие *кратности дуг*. Кратность дуги означает, сколько маркеров могут перейти по этой дуге за одно срабатывание перехода. Во всех рассмотренных ранее примерах при срабатывании перехода по каждой дуге переходил один маркер, т.е. все дуги имеют кратность 1. Такая сеть называется ординарной. Если нужно переместить сразу два (или более) маркера, то рисуют или две (или более) дуги, или одну дугу с пометкой кратности над ней (рис.2.3).

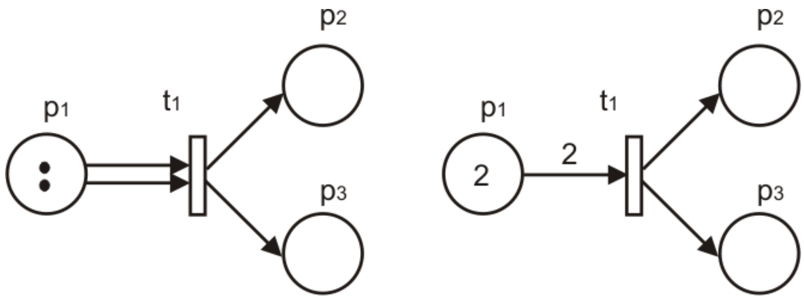



Рис.2.3. Пример обозначения кратности дуг и маркеров

### *Определение сети Петри*

Введем формальное определение сети Петри.

Сетью называют тройку ($P$, $T$, $F$), где $P$ – множество мест, $T$ – множество переходов, $F \subseteq P \times T \cup T \times P$ – отношение инцидентности и для ($P$, $T$, $F$) выполнены следующие условия:

– $P \cap T = \varnothing$ (множество мест и переходов не пересекаются);

– $(F \neq \varnothing) \wedge (\forall x \in P \cup T, \exists y P \cup T : xFy \vee yFx)$ (т.е. любой элемент сети инцидентен хотя бы одному элементу другого типа);

– если для произвольного элемента сети $x \in X$ обозначить через ${}^{*}x$ множество его входных элементов $\{y \mid yFx\}$, а через $x^{*}$ – множество его выходных элементов $\{y \mid xFy\}$, то $\forall p_1, p_2 \in P : ({}^{*}p_1 = {}^{*}p_2) \wedge (p_1^{*} = p_2^{*}) \Rightarrow (p_1 = p_2)$ (т.е. сеть не содержит пары мест, которые инцидентны одному и тому же множеству переходов).

$xFy$ означает, что из вершины *x* к вершине *y* ведет дуга.

Сеть Петри представляет собой двудольный ориентированный мультиграф *Np* = (*P*, *T*, *F, W, $M_0$*), где (*P*, *T*, *F*) – конечная сеть, $W : F \rightarrow N \setminus \{0\}$ – функция кратности дуг, $M_0 : P \rightarrow N$ – функция первоначальной разметки сети Петри, *N* – множество всех натуральных чисел (включая 0).

Функция $M_0$ сопоставляет каждому $p \in P$ некоторое число (маркеров) $M_0(p)$. В графическом представлении сети разметка места *p* изображается точками, или при большом количестве числом $M_0(p)$ (рис.2.3).

Функционирование сети Петри описывается формально с помощью множества последовательностей срабатываний переходов и множества достижимых в сети разметок.

Разметка сети *Np* – это функция $M : P \rightarrow N$. Если места в сети упорядочены, т.е. $P = (p_1,..., p_n)$, то разметку $M$ сети (в том числе и $M_0$) можно задать как вектор чисел $M = (m_1,..., m_n)$ такой, что для любого *i*, $1 \leq i \leq n$, $m_i = M(p_i)$.

На основе отношения инцидентности $F \subseteq P \times T \cup T \times P$ и функции кратности дуг $W : F \rightarrow N \setminus \{0\}$ можно ввести функцию инцидентности $F : P \times T \cup T \times P \rightarrow N$, которая определяется как

$$F(x, y) = \begin{cases} n, Fy \wedge (W(x, y) = n) \\ 0, \neg(xFy). \end{cases}$$

Если места в сети упорядочены, то можно каждому переходу *t* сопоставить два численных вектора ${}^{*}F(t)$ и $F^{*}(t)$ длиной *n*, где $n = |P|$:

${}^{*}F(t) = (b_1,...,b_n)$, где $b_i = F(p_i, t)$,

$F^{*}(t) = (b_1,...,b_n)$, где $b_i = F(t, p_i)$.

Множество разметок $M'$, достижимых в сети *Np* от разметки $M$, обозначим через $R(N, M)$. Множеством достижимых разметок в сети *Np* называют множество $R(N) = R(N, M_0)$, т.е. множество всех разметок, достижимых в сети от начальной разметки $M_0$.

Разметка $M \in R(N)$ называется тупиковой, если в сети не существует ни одного перехода, который может сработать при этой разметке.

Можно представить возможные изменения разметок сети *Np*, происходящие в результате срабатывания ее переходов в виде графа разметок – ориентированного графа, множество вершин которого образовано множеством $R(N)$ достижимых в *Np* разметок. Из вершины $M$ в вершину $M'$ ведет дуга, помеченная символом перехода *t*. Еще такой граф называют деревом достижимости.

Любой конечный граф разметок сети (покрывающее дерево) может быть построен по определенным правилам.

1. Первоначально предполагается, что дерево содержит единственную вершину-корень ($M_0$) и не имеет дуг.

2. Пусть $M$ – вершина дерева, вершина $M$ объявляется листом, если

а) ни один из переходов сети не может сработать при разметке $M$;

б) на пути из корня дерева в вершину $M$ существует вершина $M'$ такая, что $M = M'$;

в) на пути из корня дерева в вершину $M$ существует вершина $M'$ такая, что $M < M'$. Значение соответствующей координаты p в $M$ заменяется на $\omega$ и вершина $M$ объявляется листом.

3. Если вершина $M$ не лист, то $M$ внутренняя вершина дерева. Для каждого перехода *t*, такого что $M \geq^* F(t)$, в дерево добавляется новая вершина $M'$ и дуга, ведущая из $M$ в $M'$, помеченная символом *t*.

Символ $\omega$ в графе разметки (случай 2а) появляется тогда, когда при срабатывании перехода *t* на месте *p* происходит бесконечное накопление маркеров или когда в изначальной модели на месте *p* содержится неограниченное число каких-либо ресурсов (маркеров).

Например, рассмотрим сеть Петри на рис.2.5*б*. На месте $p_4$ происходит бесконечное накопление маркеров при срабатывании перехода $t_2$. Тогда множество достижимых разметок $R(N)$ бесконечно и граф разметок не может быть построен. Чтобы построить граф разметок вводится символ $\omega$. Конечный граф разметок для этой сети представлен на рис.2.4.

$$(1,0,0,2) \xrightarrow{t_1} (0,1,1,2) \xrightarrow{t_2} (1,0,0,3) \xrightarrow{t_1} (0,1,1,3) \xrightarrow{t_2} (1,0,0,\omega)$$

Рис. 2.4. Пример построения графа разметок (для сети на рис.2.5*б*)

При этом над $\omega$ справедливы следующие операции (*а* – целое неотрицательное число) $\omega - a = \omega; \omega + a = \omega; \omega + \omega = \omega$.

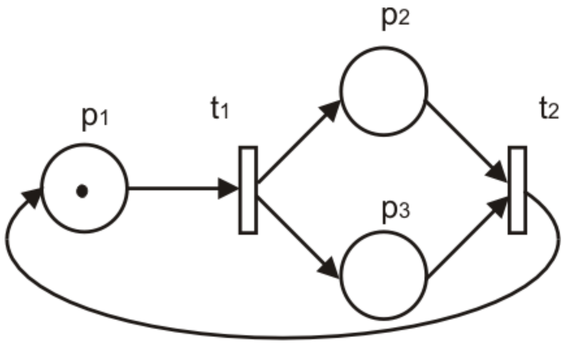
### *Основные свойства сети Петри*

Среди свойств сети Петри выделяют ограниченность, безопасность, сохранение, живость и устойчивость.

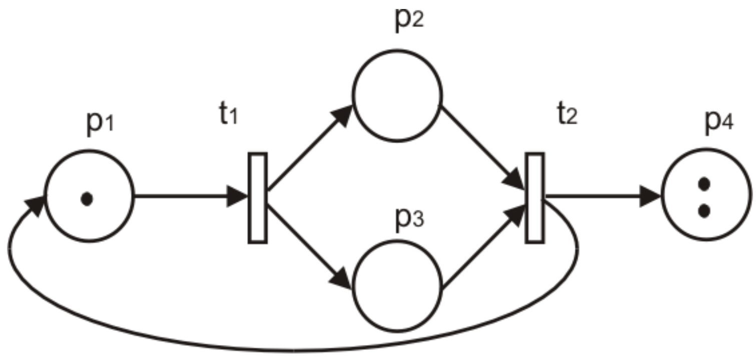
Место *p* в сети Петри *Np* = (*P*, *T*, *F, W, M₀*) называется ограниченным, если существует число *n* такое, что для любой достижимой в сети разметки $M$ справедливо неравенство $M(p) \leq n$. Сеть *Np* называется *ограниченной* сетью, если любое ее место ограничено. Множество достижимых разметок $R(N)$ ограничено, если и только если *Np* – ограниченная сеть. На рис.2.4 приведены примеры ограниченной (рис. 2.5*а*) и неограниченной (2.5*б*) сетей Петри.

Место *p* в сети Петри *Np* называется безопасным, если $\forall M \in R(N) : M(p) \leq 1$. Сеть Петри *безопасна*, если все ее места безопасны. Сеть Петри на рис.2.5*а* безопасна, т.к. любая достижимая разметка в сети представляет собой вектор из нулей и единиц.

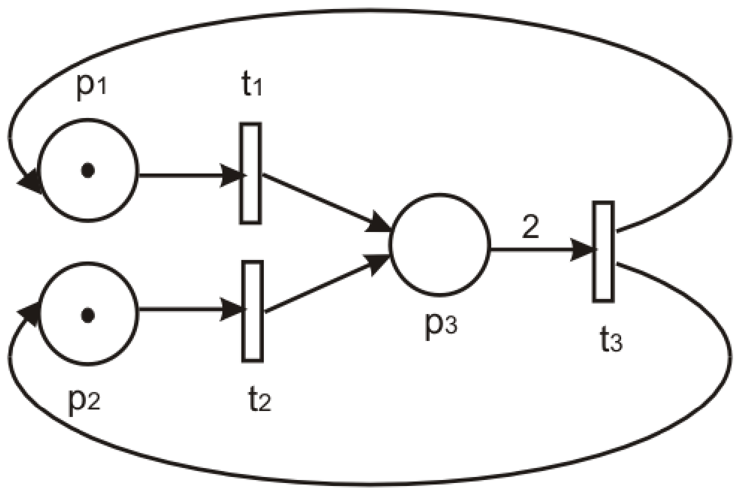
На рис.2.5*б* сеть не является безопасной, т.к. в начальной разметке $M_0(p_4) = 2$ .




а) ограниченная и безопасная сеть Петри




б) неограниченная и небезопасная сеть Петри




в) сохраняющая потенциально живая устойчивая сеть Петри

Рис. 2.5. Пример сети Петри с определенными свойствами

Ограниченность и безопасность характеризуют емкость условий: в дискретной информационной системе, моделируемой соответствующими сетями, можно ограничить емкость накопителей, необходимых для хранения условий накопления событий.

Родственным понятиям ограниченности и безопасности является понятие *консервативной* (сохраняющей) сети, в которой сумма маркеров во всех ее местах остается постоянной при работе сети, т.е. $\forall M_1 M_2 \in R(N): \sum_{p \in P} M_1(p) = \sum_{p \in P} M_2(p)$. В консервативной сети каждый переход консервативен, его срабатывание не меняет число маркеров в сети.

Переход *t* в сети Петри *Np* называется потенциально живым при разметке $M \in R(N)$, если $\exists M' \in R(N, M): M' \geq^{*} F(t)$, т.е. существует от разметки $M$ достижимая разметка $M'$, при которой переход *t* может сработать. Если существует разметка, при которой переход *t* может сработать, то он является потенциально живым в сети. Переход *t* мертвый, если он не может сработать при любой достижимой в сети разметке. Переход *t* является живым, если он может сработать при любой достижимой в сети разметке.

Сеть называется *живой*, если все ее переходы живы. Сеть мертва, если в ней есть тупиковая разметка. Сеть потенциально живая, если все ее переходы потенциально живы при любой разметке.

Переход *t* называется устойчивым в сети *Np*, если $\forall t' \in T \setminus \{t\}, \forall M \in R(N): (M \geq^{*} F(t)) \wedge (M \geq^{*} F(t')) \Rightarrow (M \geq^{*} F(t) +^{*} F(t'))$, (если переход *t* может сработать), то никакой другой переход, сработав, не может лишить его этой возможности. Сеть *Np* устойчива, если все ее переходы устойчивы.

На рис. 2.5*в* приведена сохраняющая потенциально живая устойчивая сеть Петри.

### *Пример построения и анализа сети Петри*

Построим модель работы научной лаборатории. В лаборатории работают три сотрудника. Каждый сотрудник может приходить и уходить с работы в любое время. В лаборатории сотрудники обсуждают разработки и выполняют проекты. Для разработки проекта нужно как минимум два сотрудника. На рис. 2.6 приведена модель работы лаборатории в виде сети Петри.

Сеть $N_P$ состоит из трех мест $p_1$, $p_2$ и $p_3$ и трех переходов $t_1$, $t_2$, $t_3$. Два маркера на месте $p_1$ и один маркер на месте $p_2$ означают сотрудников лаборатории. Место $p_1$ задает локальное состояние, соответствующее тому, что сотрудник находится дома. Позиция $p_2$ означает нахождение на работе. Маркеры на месте $p_3$ будут соответствовать разработанным сотрудниками лаборатории проектам.

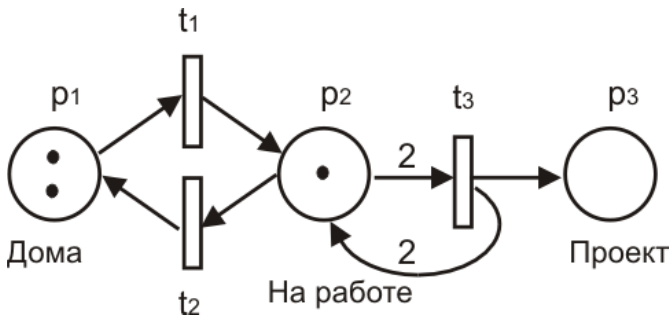



Рис.2.6. Модель работы научной лаборатории

Таким образом, изображенная на рис. 2.6 разметка сети $N_P$ задает состояние модели, в котором два сотрудника находятся дома, один – на работе, и еще ни один проект не разработан.

Для разметки сети $N_P$ активными являются переходы $t_1$и $t_2$. Срабатывание первого перехода состоит в «перемещении» одного маркера из места $p_1$ в $p_2$ и содержательно соответствует приходу одного из сотрудников на работу. Срабатывание перехода $t_2$ «перемещает» маркер из позиции $p_2$ в позицию $p_1$ и соответствует уходу одного сотрудника с работы домой. Переход $t_3$ не является активным в данной разметке. Для его срабатывания необходимо наличие в позиции $p_3$ по крайней мере двух маркеров.

После срабатывания перехода $t_1$ (прихода на работу еще одного сотрудника) переход $t_3$ становится активным, т.е. два сотрудника могут создать проект. После срабатывания $t_3$ в месте $p_3$ появляется один маркер (количество созданных проектов), а в позиции $p_2$ попрежнему остаются два маркера (сотрудника).

Проведем формальное описание полученной сети Петри. Множество мест сети $P=(p_1,p_2,p_3)$ и множество переходов $T=(t_1,t_2,t_3)$. Функция инцидентности равна

$F=\{(p_1,t_1,p_2),(p_2,t_2,p_1),(p_2,t_3,p_2+p_3)\}$, где

$^*F=\{(p_1,t_1),(p_2,t_2),(p_2,t_3)\}$, $F^*=\{(t_1,p_2),(t_2,p_1),(t_3,p_2+p_3)\}$.

Начальная разметка сети $M_0=(2,1,0)$.

Нарисуем граф разметок для данной сети (рис.2.7). Граф содержит $\omega$, т.к. на месте $p_3$ происходит увеличение маркеров на 1 при каждом срабатывании перехода срабатывании $t_3$.

$$
\begin{array}{cccccccc}
 & & & (2,1,0) & & & & \\
t_2\downarrow & & & & & \downarrow t_1 & & \\
(3,0,0) & & & & & (1,2,0) & & \\
t_1\downarrow & & \downarrow t_1 & & & \downarrow t_3 & & \downarrow t_2 \\
(2,1,0) & & (0,3,0) & & & (1,2,1) & & (2,1,0) \\
 & t_3\downarrow & t_2\downarrow & & \downarrow t_2 & \downarrow t_3 & \downarrow t_1 & \\
 & (0,3,1) & (1,2,0) & & (2,1,\omega) & (1,2,\omega) & (0,3,0) & \\
t_3\downarrow & \downarrow t_2 & & & & & & \\
(0,3,\omega) & (1,2,1) & & & & & &
\end{array}
$$

Рис. 2.7. Граф разметок для сети на рис. 2.6

Проанализируем свойства сети Петри. Начальная разметка сети равна $M_0 = (2,1,0)$, т.е. $M_0(p_1) = 2$, следовательно, сеть не является безопасной. На месте $p_3$ происходит неограниченное накопление маркеров (возможно такое состояние, когда $M(p_3) > n$) и в графе разметки присутствует символ $\omega$.Следовательно, сеть не является ограниченной; т.к. при каждом срабатывании перехода $t_3$ число маркеров в сети увеличивается на 1, то данная сеть не является сохраняющей. Все переходы сети являются потенциально живыми, в сети нет тупиковых ситуаций, следовательно, сеть потенциально жива.

### *Виды сетей Петри*

Сети Петри можно разделить на обыкновенные и сети высокого уровня. Также среди сетей Петри высокого уровня выделяют сети с охраной (с приоритетами), раскрашенные сети, вложенные (иерархические) сети, вложенные сети с охраной.

Сети Петри, в которых маркеры одинаковы, являются равными для условий и не отличаются друг от друга, называют *обыкновенными сетями Петри*. Обыкновенные сети Петри являются удобным средством моделирования и анализа различных параллельных и распределенных систем. Однако при моделировании реальных систем возникают проблемы, связанные с большим размером получающихся сетей.

Для сокращения размеров сети и ее большей наглядности в начале 80-х годов XX века были определены *сети Петри высокого уровня*, которые характеризуются следующими особенностями:

– разметка сети задается с помощью индивидуальных, т. е. различимых между собой маркеров;

– переходы могут срабатывать в различных режимах, удаляя маркеры из одних позиций и добавляя их в другие, при этом единственным априорным ограничением является требование локальности, т. е. в любом режиме переход может удалять маркеры только из своих входных позиций и добавлять только в выходные позиции.

Таким образом, маркеры в сетях Петри высокого уровня могут быть разных типов, и разметка сопоставляет позиции сети не просто количество, как в обыкновенных сетях Петри, а мультимножество маркеров. Число различных индивидуальных маркеров, используемых в разметках сети, предполагается конечным. Изменяется также вид пометок, приписанных дугам сети. Вместо натуральных чисел, задающих кратность дуг в обыкновенных сетях Петри, в сетях Петри высокого уровня дугам приписываются выражения, содержащие переменные. Различные режимы срабатывания переходов задаются различными означиваниями переменных в этих выражениях, при этом переменной в качестве значения приписывается маркер некоторого типа, а значением выражения является мультимножество маркеров.

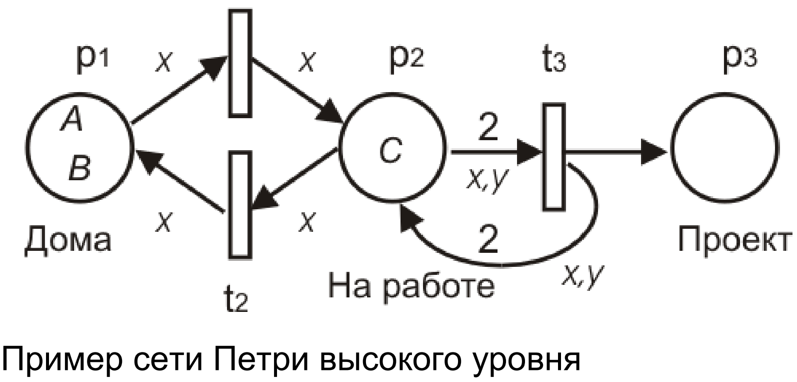



Рис. 2.8. Пример сети Петри высокого уровня

На рис. 2.8 приведен пример сети Петри высокого уровня, который является уточнением примера, представленного на рис. 2.6. В этой сети каждый сотрудник имеет индивидуальный маркер *A*, *B* и *C*. Каждый из сотрудников, как и в предыдущем примере, может находиться дома или на работе. Переменная *x* в цикле перехода из дома на работу и с работы домой может принимать любое из значений *A*, *B* и *C*. Три возможных означивания переменной *x* задают три режима срабатывания для каждого из переходов $t_1$ и $t_2$. Таким образом, один переход $t_1$ в этой сети высокого уровня представляет три перехода – по одному для каждого из участников *A*, *B* и *C*. В начальной разметке сети сотрудники *A* и *B* находятся дома, а *C* – на работе. После срабатывания перехода при означивании $x:=A$ сотрудники *A* и *C* будут находиться на работе и могут провести обсуждение, что приведет к выработке проекта, т.е. переменным *x* и *y* присвоится значения *A* и *C*, и переход $t_3$ сработает.

Для задания условий срабатывания перехода в сетях Петри высокого уровня переходу может быть приписано логическое выражение, называемое *охраной* перехода. На рис.2.9 приведена модификация сети, в которой переходу $t_2$ приписана охрана $x \neq A$. Это означает, что переход $t_2$ может сработать только при означивании *x*, отличном от *A*. Содержательно это моделирует ситуацию, когда *A* является «трудоголиком» и вообще не уходит домой. В этом примере также выражение *y* на дуге от $p_3$ к $t_3$ заменено на выражение *A*. Теперь выработка проекта не может проходить без участия сотрудника *A*.

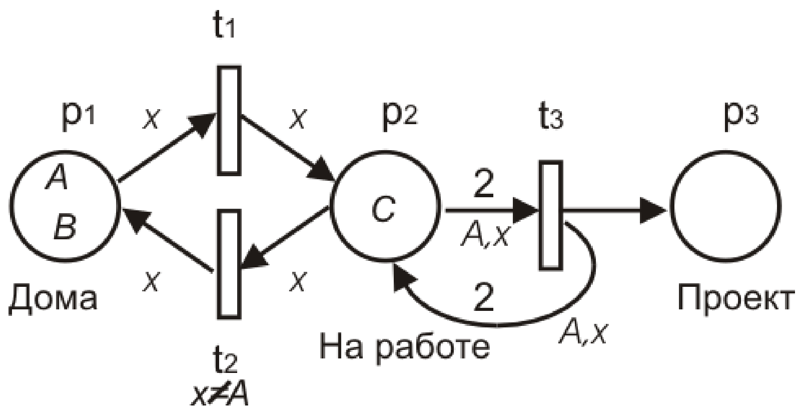



Рис. 2.9. Пример сети Петри высокого уровня с охраной

Характерным примером сетей Петри высокого уровня являются *раскрашенные сети* Йенсена. Раскрашенные сети Петри основаны на языке с типами, которые называются цветами. Маркерам в раскрашенных сетях приписаны различные цвета (типы). Места также имеют цвет, при этом место может содержать маркеры только приписанного ей цвета.

Известно, что сети Петри высокого уровня, в частности, раскрашенные сети Йенсена при условии конечности набора цветов и числа индивидуальных маркеров каждого цвета моделируются обыкновенными сетями Петри, т. е. по сети высокого уровня можно построить эквивалентную ей по поведению обыкновенную сеть Петри. Но при такой «развертке» сети высокого уровня в обыкновенную сеть размер сети может существенно увеличиться.

Во вложенных сетях маркеры в местах сети сами могут быть сетями Петри. Вложенная сеть состоит из системной сети и элементных сетей, представляющих сетевые маркеры. Например, элементной сетью можно смоделировать состояние сотрудника лаборатории (уставший или отдохнувший). А системная сеть будет практически неотличима от сети на рис.2.8, кроме того, что маркеры в позициях $p_1$ и $p_2$ будут являться экземплярами элементной сети.

Если нужно, чтобы сотрудник приходил на работу только в активном состоянии, то на переход $t_1$ системной сети накладывается соответствующее условие – охрана.

С помощью охраны можно накладывать условие и на структуру сетевого маркера, и на ее разметку. Определение вложенной сети Петри предполагает конечное число элементных сетей, определяющих сетевую структуру сетевых маркеров. Поэтому условие на структуру сетевых маркеров существенно не отличается от условий, налагаемых на маркеры в сетях Петри высокого уровня.

В отличие от сетей высокого уровня, вложенные сети строго выразительнее обыкновенных сетей Петри, но все же слабее по выразительности, чем универсальные вычислительные модели (например, машины Тьюринга).

## *Порядок выполнения работы и содержание отчета*

*Порядок выполнения работы*

1. Изучить теорию и правила построения сетей Петри.
2. Построить сеть Петри.
3. Формально описать сеть Петри.
4. Проанализировать полученную сеть Петри.

5. Провести анализ модели.
6. Ответить на контрольные вопросы и решить тестовую задачу.

*Содержание отчета*

1. Цель работы и текст задания.
2. Описание модели системы и схема сети Петри.
3. Формальное описание и граф разметок сети Петри.
4. Свойства полученной сети Петри.
5. Вывод о достоинствах и недостатках полученной модели.

*Контрольные вопросы*

1. Для чего и где применяются сети Петри?
2. Определение сети Петри?
3. Какая сеть называется ординарной?
4. Правила построения графа разметок.
5. Для чего и в каких случаях вводится символ $\omega$?
6. Перечислите основные свойства сетей Петри.
7. Перечислите основные виды переходов.
8. Какие виды сетей Петри вы знаете?
9. Что такое охрана?
10. Как строятся вложенные сети Петри?

*Варианты заданий*

Вариант 1. Описать с помощью сети Петри процессы поиска и устранения неисправностей в некоторой технической системе, состоящей из 5 однотипных блоков; в запасе имеется один исправный блок (в холодном резерве); известны статистические данные об интенсивностях возникновения отказов и длительностях таких операций, как поиск неисправностей, замена и ремонт отказавшего блока. Отказ запасного блока не рассматривается.

Вариант 2. Описать с помощью сети Петри работу и процессы возникновения неисправностей в некоторой технической системе, состоящей из 5 однотипных блоков и двух блоков в горячем резерве. Известны статистические данные об интенсивностях возникновения отказов. Система работает при трех исправных блоках.

Вариант 3. Описать с помощью сети Петри последовательную обработку запросов сервером. Сервер находится в состоянии ожидания до тех пор, пока от пользователей не поступят запросы, которые он обрабатывает и отправляет результат такой обработки пользователям.

Сервер одновременно может обрабатывать 3 типа запроса от пользователей. Запросы поступают не одновременно, делятся на три типа и их обработка содержит разное количество операций (1 тип – 3 операции, 2 тип – 2 операции, 3 тип – 4 операции) и занимает разное количество времени. Потенциальное количество пользователей неограниченно. Кроме того, ведется статистика обработки каждого типа запросов.

Вариант 4. Описать с помощью сети Петри работу небольшой Управляющей компании (УК) с жителями.

От жителей района в УК поступают заявления, секретарь делит заявления на 3 категории. Первая категория – заявления на перерасчет – отправляет в бухгалтерию, где заявление обрабатывается, формируется квитанция и отправляет жителю. Вторая категория – жалобы на сантехников – сортирует по трем фамилиям (т.к. в УК три сантехника). Как только число жалоб на какого-либо сантехника достигает 5, секретарь пишет служебную записку и отправляет ее директору. Директор рассматривает жалобу и принимает решение лишить сантехника премии или уволить. Эта информация передается сантехнику. В случае увольнения берется на работу новый сантехник. Заявления третьей категории – разное – отправляются директору на рассмотрение, он принимает решение и по почте отправляет ответ жителю.

Вариант 5. Описать с помощью сети Петри процесс проведения лабораторных занятий в компьютерном классе.

В классе 8 компьютеров, в подгруппе 11 студентов. За одним компьютером может сидеть не более одного студента. Студенты занимают компьютеры и делают задание. Студент обращается к преподавателю, преподаватель подходит к студенту. После этого студент продолжает самостоятельную работу или освобождает компьютер и уходит с занятия. Занятие заканчивается, когда все студенты закончат работу за компьютерами.

Вариант 6. Описать с помощью сети Петри процесс проведения лабораторных занятий в компьютерном классе.

В классе 10 компьютеров, в подгруппе 13 студентов. За одним компьютером может сидеть не более одного студента. Студенты занимают компьютеры и делают задание. Студент обращается к преподавателю, преподаватель подходит к студенту. После этого студент продолжает самостоятельную работу или освобождает компьютер и уходит с занятия. Преподаватель за занятие может подойти к каждому студенту не более одного раза. Занятие заканчивается, когда преподаватель подойдет ко всем студентам.

Вариант 7. Для разработки АСУ магазина требуется оптимизировать работу отдела снабжения.

Описать с помощью сети Петри работу отдела снабжения, который имеет 3 поставщика товаров, каждый поставщик предлагает 5 видов товаров. После закупки товар поступает на один из двух складов (каждый склад может принять не более 4 видов товара). Со склада товар поступает на прилавки магазина. Требуется предложить такую сеть Петри, которая позволила бы синхронизировать независимые поставки товара.

Вариант 8. Описать с помощью сети Петри систему массового обслуживания, которая состоит из 5 ед. оборудования и одной базы данных. Для экономии технических средств оборудование расположено по кругу. Чтобы обработать заявку, поступившую на пункт обработки, необходимо две одновременно свободные единицы оборудования, находящиеся рядом (справа и слева). После обслуживания заявки, оборудование записывает значение в базу данных, затем заявка освобождает оборудование и выходит из системы массового обслуживания.

Вариант 9. Описать с помощью сети Петри работу материально-технического отдела снабжения, который имеет 3 машины, 3 бригады грузчиков и 2 погрузчика. Для загрузки машины необходимо 2 бригады грузчиков и 1 погрузчик. Требуется построить такую сеть Петри, которая позволит синхронизировать независимые действия грузчиков. В сети не должно быть взаимных блокировок, например, когда каждая бригада грузчиков займет три разных машины. Необходимо обосновать наличие двух бригад и погрузчиков в отделе.

Вариант 10. Описать с помощью сети Петри работу двух групп пользователей на единственной рабочей станции. Рабочая станция состоит из трёх однотипных блоков, в запасе имеется два исправных блока; известны статистические данные об интенсивностях возникновения отказов и длительностях замены и ремонта отказавшего блока.

Литература

1. Котов В.Е. Сети Петри. – М.: Наука. Главная редакция физико-математической литературы, 1984. – 160 с.

2. Ломазова И. А. Вложенные сети Петри и моделирование распределенных систем // Труды международной конференции «Программные системы: теория и приложения», ИПС РАН, г. Переславль-Залесский, май 2004. – Т.1 – М.: Физматлит, 2004. – с. 337-352.

## Содержание

Редактор О.Ю. Волошенко
Компьютерная верстка Е.Н. Алонцева

ЛР №020713 от 27.04.1998

Подписано к печати 19.11.2007 Формат бум. 60×84/16

Печать ризограф. Бумага MB Печ. л. 2,5
Заказ № Тираж 50 экз. Цена договорная

Отдел множительной техники ИАТЭ
249035, г.Обнинск, Студгородок, 1