

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информационные системы в строительстве»

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ
В НАУЧНЫХ ИССЛЕДОВАНИЯХ

Методические указания к лабораторным работам

Ростов-на-Дону
ДГТУ
2021

УДК 004.8(075.8)

Составители: А.А. Ляпин

Информационные системы и технологии в научных исследованиях: метод. указания к лабораторным работам. – Ростов-на-Дону : Донской гос. техн. ун-т, 2021. – 18 с.

Методические указания к лабораторным работам содержат краткую теорию по изучаемым вопросам, примеры применения информационных систем в научных исследованиях, и индивидуальные задания.

Предназначены для обучающихся магистратуры по направлению подготовки 09.04.02 «Информационные системы и технологии» очного и заочного обучения.

УДК 004.8(075.8)

Печатается по решению редакционно-издательского совета
Донского государственного технического университета

Ответственный за выпуск зав. кафедрой «Информационные системы
в строительстве» д-р физ.-мат. наук, профессор А.А. Ляпин

В печать __.__.2021 г.
Формат 60×84/16. Объем 1,0 усл. п. л.
Тираж 60 экз. Заказ № ____

Издательский центр ДГТУ
Адрес университета и полиграфического предприятия:
344000, г. Ростов-на-Дону, пл. Гагарина, 1

© Донской государственный
технический университет, 2021

ОГЛАВЛЕНИЕ

Лабораторная работа № 1.	Моделирование динамики линейного осциллятора	4
Лабораторная работа № 2.	Размножение бактерий	10
Лабораторная работа № 3.	Динамика популяции типа «хищник – жертва».....	11
Лабораторная работа № 4.	Имитационное моделирование в среде SciLab	12
Лабораторная работа № 5.	Моделирование асинхронных процессов с использованием сетей Петри	14
Лабораторная работа № 6.	Матричные методы анализа сетей Петри	18
Лабораторная работа № 7.	Исследование сетей Петри на основе дерева достижимости	20
Лабораторная работа № 8.	Анализ сетей Петри с использованием инвариантов	22
Литература	23

Лабораторная работа № 1. Моделирование динамики линейного осциллятора

Цель работы: изучить методы исследования поведения динамических систем на примере решения задачи Коши для линейного дифференциального уравнения 2 порядка с использованием систем MathCAD и Matlab.

Рассмотрим движение из положения равновесия груза на пружинке под действием внешнего импульсного усилия $f(t)$:

$$m \frac{dx(t)}{dt^2} + kx(t) = f(t), \quad (1)$$

здесь $x(t)$ – переменная, описывающая смещение груза,

t – время,

m – масса груза,

k – жесткость пружины.

Уравнение (1) есть линейное неоднородное дифференциальное уравнение второго порядка с постоянными коэффициентами.

В качестве начальных условий примем $x(0) = 0$, $\frac{dx}{dt}(0) = 0$.

Для заданных коэффициентов уравнения и вида финитной на отрезке $[0, \tau]$ функции $f(t)$ требуется:

1. Определить характер изменения смещения груза на отрезке $[0, T]$.
2. Построить фазовый портрет движения: зависимость скорости движения груза от его смещения.
3. Определить примерное значение коэффициента динамичности системы «груз на пружине», равного отношению максимальной амплитуды смещения груза к амплитуде его установившихся колебаний.
4. Определить влияние изменения параметров системы m, k, τ на характер поведения исследуемой функции $x(t)$.
5. Ввести демпфирующий элемент в уравнение (1): добавить в левой части уравнения слагаемое вида $\eta \frac{dx(t)}{dt}$, коэффициент демпфирования η принять равным 0.1. Определить его влияние на поведение системы.

Варианты заданий

№	m	k	τ	T	$f(t)$
0	1	4	3	10	1
1	2	10	10	30	$\cos(t)$
2	2	10	10	30	$\cos(\sqrt{5} t)$
3	2	5	4	15	$1/(t+1)$
4	3	1	5	15	$\sin(t) \cdot \exp(-0.2 t)$
5	4	1	3	10	$t \cdot \exp(-0.3 t)$
6	2	2	4	15	$\cos(2t)/(t+2)$
7	1	4	2	10	$\ln(1+t^2)$
8	3	6	3	20	$t/(t^2+1)$
9	2	1	2	30	$t \cdot \cos(t+2)$

Номер варианта соответствует последней цифре в зачетке.

Задание выполнить в системе MathCAD, а также по выбору в системе MatLab или Scilab с использованием численных расчетов и модуля имитационного моделирования Simulink (Xcos).

Пример выполнения работы в системе MathCAD.

Зададим параметры системы:

$$\underline{m} := 0.5 \quad k := 2 \quad \tau := 3 \quad \underline{T} := 10$$

Определим импульс усилия:

$$f(t) := \begin{cases} \sqrt{t} & \text{if } 0 \leq t \leq \tau \\ 0 & \text{otherwise} \end{cases}$$

Определим правую часть дифференциального уравнения с его приведением к системе в нормальной форме

$$D(t, x) := \begin{pmatrix} x_1 \\ \frac{f(t) - k \cdot x_0}{m} \end{pmatrix}$$

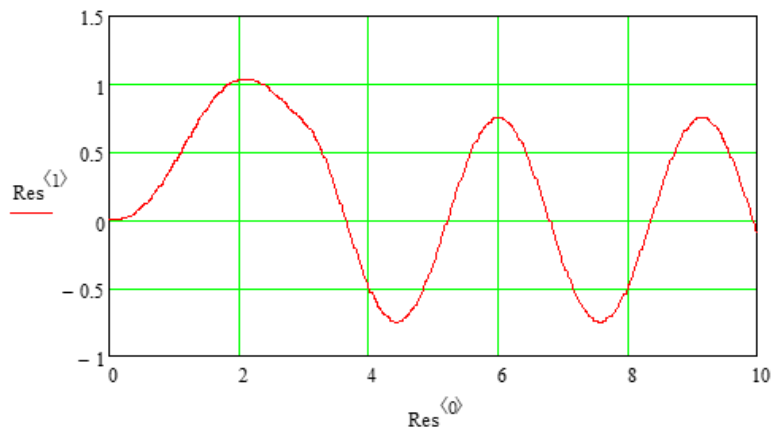
Зададим начальные условия для дифференциального уравнения

$$x := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

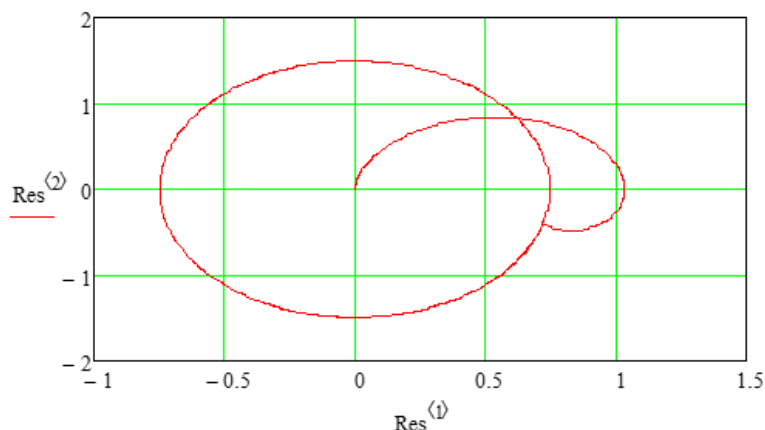
Решим уравнение методом Рунге-Кутты с постоянным шагом на интервале $[0, T]$ и 1000 точками интегрирования

$$\text{Res} := \text{rkfixed}(x, 0, T, 1000, D)$$

Построим график перемещений груза



Построим фазовый портрет системы



Сделаем вывод, что характер движения динамической системы соответствует центру.

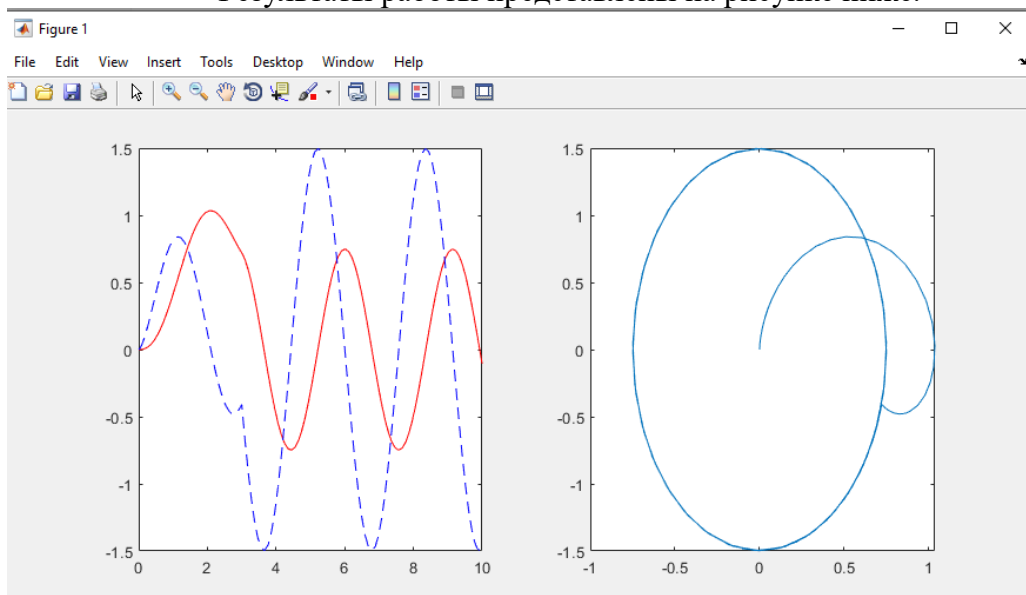
Пример выполнения работы в системе Matlab.

А) Использование численного алгоритма на основе метода Рунге-Кутты.

Текст скрипта в системе Matlab решения поставленной задачи.

```
% Определяем интервал интегрирования ОДУ
T=10;
t_int=[0 T];
% Задаем вектор начальных условий
init_cond=[0;0];
% Вызываем метод Рунге-Кутты 4-го порядка решения уравнения.
% Функция F определяет правую часть уравнения, записанного
% в виде системы ДУ в нормальной форме
[t,X]=ode45(@F, t_int, init_cond);
% Выводим графики временных зависимостей
subplot(1,2,1)
plot(t,X(:,1),'r')
hold on
subplot(1,2,1)
plot(t,X(:,2),'b--')
subplot(1,2,2)
plot(X(:,1),X(:,2))
% Определяем правую часть системы
function D = F(t,X)
    k=2;
    m=0.5;
    tau=3;
    if (t<=tau)
        f = sqrt(t);
    else
        f = 0;
    end
    dX1 = X(2);
    dX2 = (f-k*X(1))/m;
    D = [dX1;dX2];
End
```

Результаты работы представлены на рисунке ниже.



Б) Исследование задачи в среде имитационного моделирования Simulink системы Matlab

Представим дифференциальное уравнение $m \frac{dx(t)}{dt^2} + kx(t) = f(t)$ в разрешенном относительно старшей производной виде:

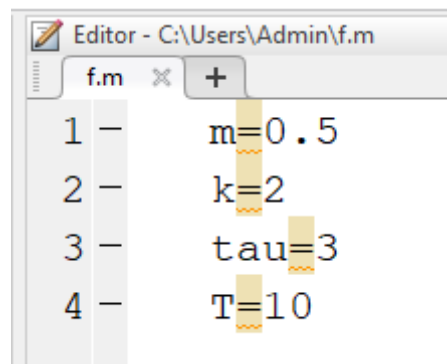
$$\frac{dx(t)}{dt^2} = \frac{f(t)}{m} - \frac{kx(t)}{m}.$$

Тогда, с помощью оператора интегрирования, применяемого дважды, можно получить интегральное уравнение:

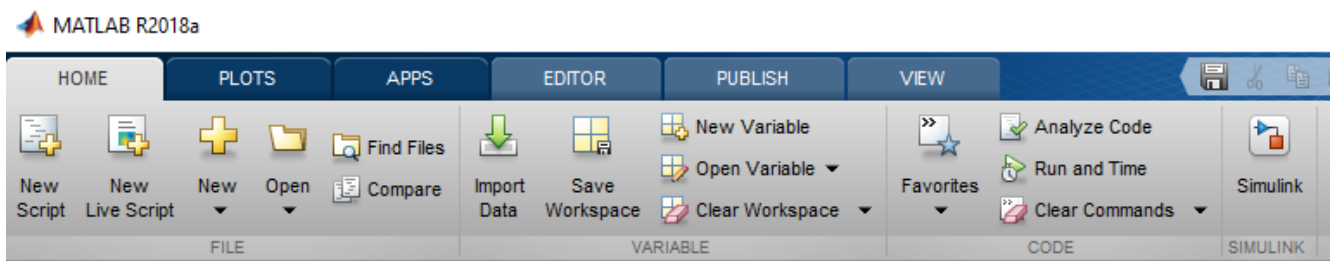
$$x(t) = \int_0^t dt \int_0^t dt \left(\frac{f(t)}{m} - \frac{kx(t)}{m} \right). \quad (2)$$

Для уравнения (2) можно составить имитационную модель с использованием блоков системы Simulink Matlab.

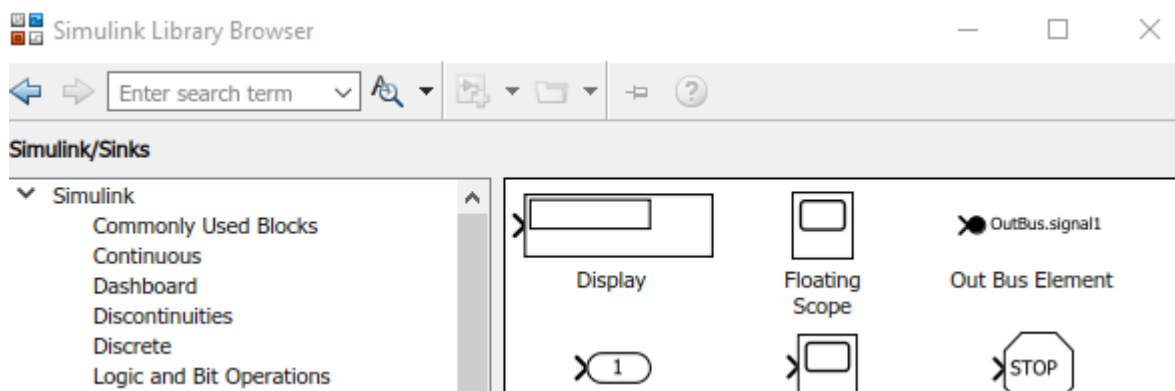
Для этого в командной строке или редакторе системы Matlab зададим начальные значения переменных задачи.



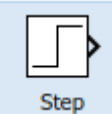
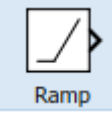
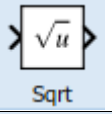
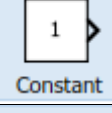


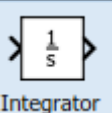
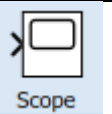
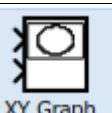
Вызовем модуль Simulink



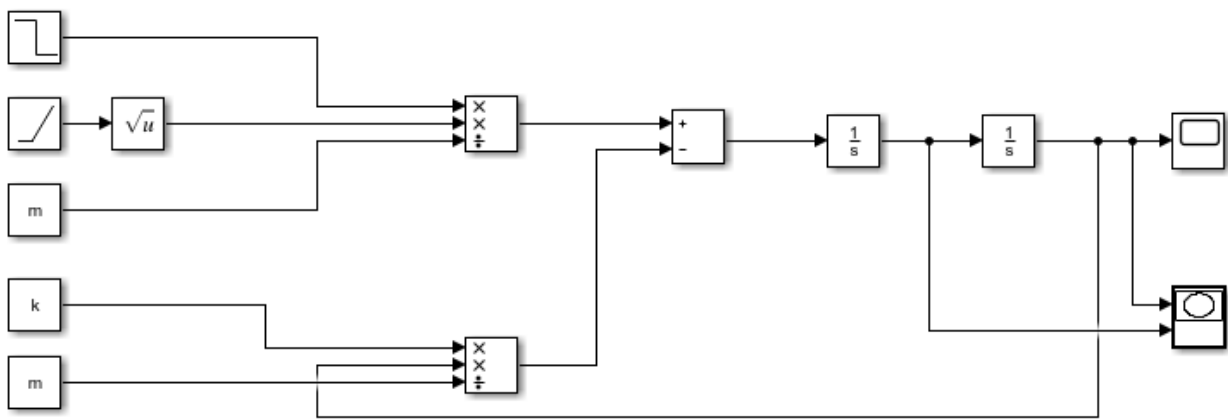
Отобразим на экране библиотеку графических блоков симуляции процессов (Simulink Library Browser)



Характеристики использованных блоков и их назначение приведены в таблице ниже:

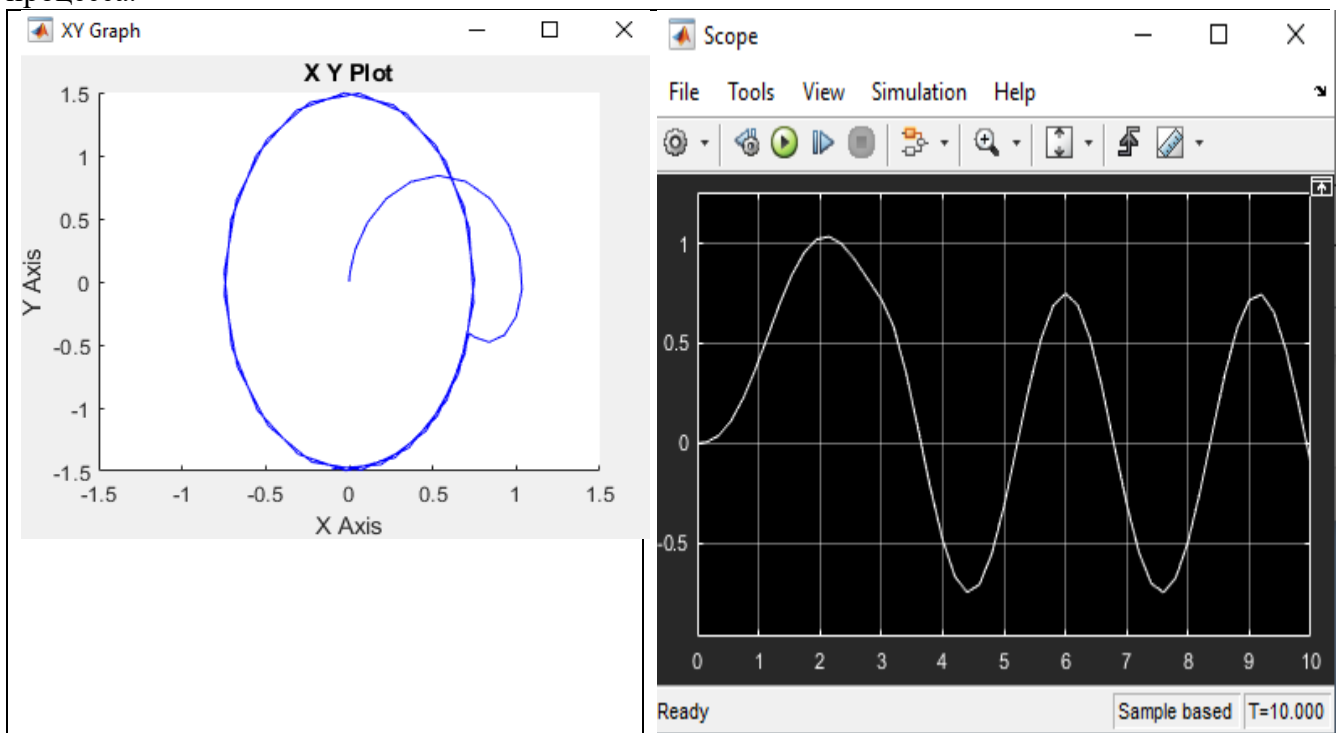
Блок	Группа	Название	Назначение	Параметры
 Step	Sources	Step	Пороговая единичная функция	Step time - время действия импульса $\tau=3$, Initial value = 1, Final value = 0
 Ramp	Sources	Ramp	Полулинейная функция: $Y=t$ при $t>0$	нет
 Sqrt	Math Operations	Sqrt	Квадратный корень	нет
 Constant	Sources	Constant		Constant value = m или k
 Product	Math Operations	Product	Произведение или частное двух и более величин	Number of inputs: **/ - соответствует последовательности знаков операций между входами блока
 Add	Math Operations	Add	Сумма или разность двух или более величин	List of signals: +- - соответствует последовательности знаков операций между входами блока
 Integrator	Continuous	Integrator	Оператор интегрирования по времени	Initial condition: начальные условия по перемещению или скорости, равные 0 – для нашей задачи
 Scope	Sinks	Scope	Блок построения графика характеристики от времени $x(t)$	нет
 XY Graph	Sinks	XY Graph	Блок построения фазовых характеристик $y(x)$	нет

И создадим основную схему процесса решения задачи Коши для дифференциального уравнения.



Запустим процесс имитационного моделирования кнопкой Run.

В итоге получим график зависимости перемещения груза на пружине и фазовый портрет процесса.



Лабораторная работа № 2. Размножение бактерий

Известно, что некоторые бактерии размножаются пропорционально их наличному количеству, но в то же время вырабатывают яд, истребляющий их пропорционально количеству яда и количеству бактерий. Скорость выработки яда также пропорциональна наличному количеству бактерий. Коэффициенты пропорциональности соответственно равны k, k_1 и k_2 .

Определить закон изменения количества бактерий и яда, если известно, что в начальный момент времени количество бактерий было равно a , а яд отсутствовал. Показать, что число бактерий сначала возрастает до некоторого наибольшего значения M , а затем убывает до нуля. Построить фазовый портрет нелинейной динамической системы. Определить характер устойчивости поведения системы.

Указания к решению

Обозначив количество яда через x , количество бактерий y и выполнив условия задачи, получим систему дифференциальных уравнений

$$\begin{aligned}\frac{dy}{dt} &= ky - k_1xy \\ \frac{dx}{dt} &= k_2y\end{aligned},$$

где $\frac{dy}{dt}, \frac{dx}{dt}$ - скорости размножения бактерий и выработки яда соответственно, k, k_1, k_2 - коэффициенты пропорциональности.

Начальные условия имеют вид: $x(0) = 0, y(0) = a$.

Варианты заданий

N	a	k	k_1	k_2
0	1	1	1	1
1	2	2	1	2
2	1	2	1	0.5
3	2	1	2	1
4	1	1	2	2
5	2	2	2	0.5
6	1	2	1	1
7	2	1	1	2
8	1	1	2	0.5
9	2	2	2	1

Определить по графику процесса момент времени, когда число бактерий достигнет наибольшего значения M , а также само значение M .

Решение задачи осуществить одним из способов, приведенных в лабораторной работе 1.

Лабораторная работа № 3. Динамика популяции типа «хищник – жертва»

Динамика популяций типа хищник - жертва описывается системой двух нелинейных дифференциальных уравнений (уравнения Лотки -Вольтерры)

$$\begin{cases} \frac{dX}{dt} = k_1AX - k_2XY \\ \frac{dY}{dt} = k_2XY - k_3BY \end{cases} \quad (1)$$

Неизвестная функция X описывает популяцию жертвы, а Y - популяцию хищника. Величина A характеризует пищу жертвы, а коэффициент k_1 устанавливает зависимость между $\frac{dX}{dt}$ и X , коэффициент k_2 устанавливает прямо пропорциональную зависимость между скоростями изменения популяций и численностью популяций хищник - жертва, k_3 и B характеризуют скорости смертности хищника.

Начальные условия задачи (численность популяций жертвы и хищника в начальный момент времени) определяются соотношениями:

$$X(0) = x_0, \quad Y(0) = y_0.$$

Варианты заданий

- Исследовать решение уравнений (1) при постоянных k_1, k_2, k_3 и функциях времени $A(t), B(t)$, задаваемых таблицей.

N	k_1	k_2	k_3	$A(t)$	$B(t)$	x_0	y_0
0	5	1	3	$1+0.8\sin t$	t^2	2	2
1	5	2	5	1	$1/(1+t)$	2	1
2	3	3	5	$1/(1+t)$	1	2	2
3	2	3	4	t	T	3	4
4	3	4	3	t	\sqrt{t}	2	1
5	4	1	4	$1/(3+t)$	$\ln(1+t)$	4	1
6	5	3	3	$0.75+0.25\sin t$	1	2	3
7	4	4	4	1	$\sqrt{t+5}$	4	2
8	3	2	3	$0.8+0.2\cos(t)$	$\sqrt{t+1}$	2	3
9	2	1	1	3	0.5	4	3

- Построить графики $X(t), Y(t)$ решений, а также фазовые траектории $y(x)$. Определить характер устойчивости поведения системы.

Решение задачи осуществить одним из способов, приведенных в лабораторной работе 1.

Лабораторная работа № 4. Имитационное моделирование в среде SciLab

Xcos (Scilab Connected Object Simulator) – составная часть пакета Scilab, обеспечивающая возможность визуального моделирования динамических систем.

Scilab является свободно распространяемым аналогом пакета Matlab, а среда имитационного моделирования Xcos – аналогом Simulink.

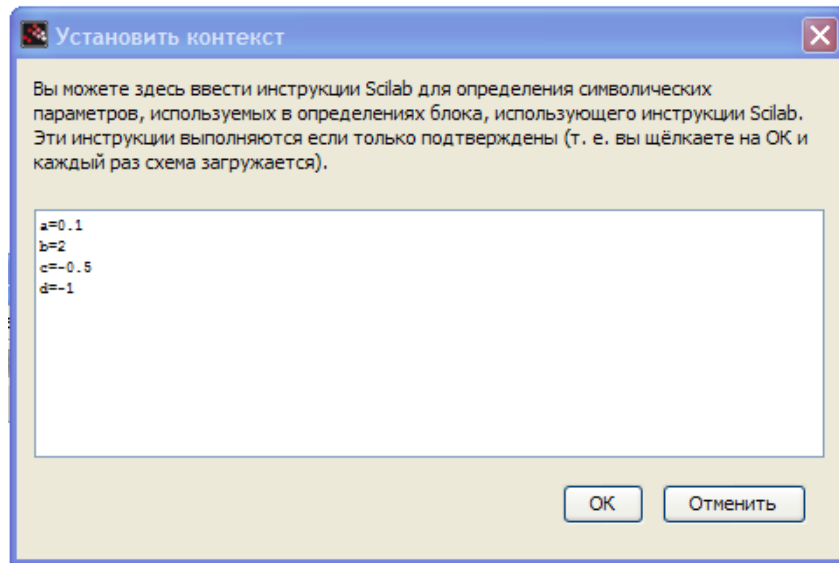
Моделируемые системы могут быть как непрерывными, так и дискретными.

В качестве примера рассмотрим решение дифференциального уравнения третьего порядка с постоянными коэффициентами в среде Xcos:

$$y''' + ay'' + by' + cy + d = 0.$$

Начальные условия являются нулевыми: $y''(0) = y'(0) = y(0) = 0$.

1. Через меню *Моделирование>Установить контекст* введем параметры уравнения:





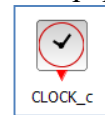
2. Приведем уравнение к форме:

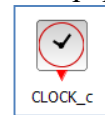
$$y''' = -ay'' - by' - cy - d.$$

Для отыскания неизвестной $y(t)$ необходимо трехкратное интегрирование производной, входящей в данное уравнение. Получающиеся после каждого интегрирования результаты умножаются на соответствующий коэффициент и суммируются. Начальные условия задаются в свойствах блоков интегрирования. Для вывода решения $y(t)$ и построения фазовой кривой $y'(y)$ используются блоки CSCOPE и CSCOPXY из палитры «Регистрирующие устройства».

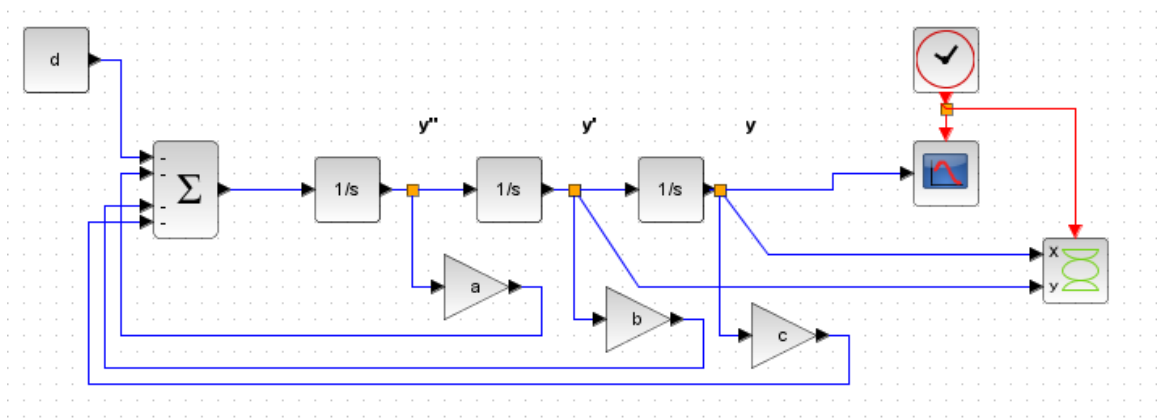


Сумматор  и блок умножения  расположены в палитре «Математические операции». Блок интегрирования – в палитре «Системы с непрерывным временем».



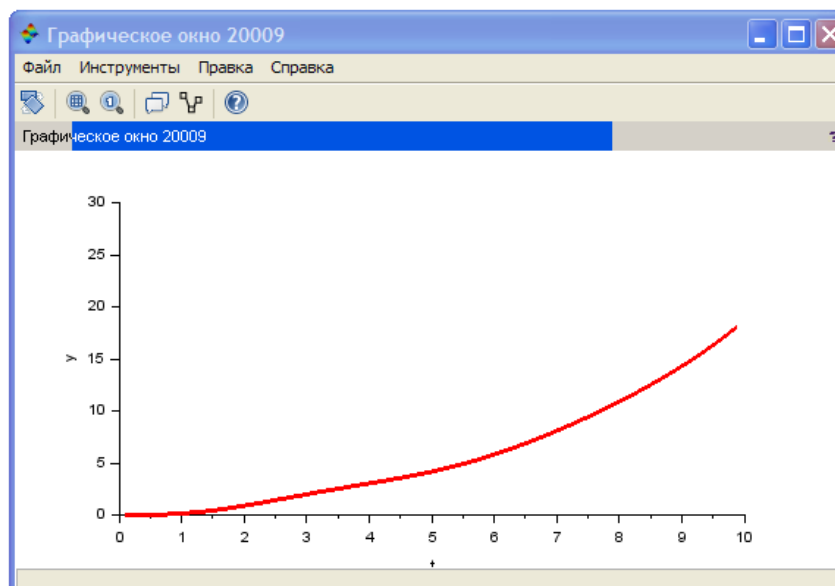
Для инициализации динамического процесса применяется таймер  из палитры «Обработка событий».

В итоге получим общую схему имитационного расчета поведения динамической системы:



3. Настройка параметров моделирования осуществляется через меню *Моделирование>Установка*. В частности зададим конечное время интегрирования равным 10.

4. Графический результат помещается в отдельные графические окна. Через меню «Правка» в них можно настроить параметры осей, кривой, подписи и т.д.



Задание

1. Исследовать динамику процесса, описываемую дифференциальным уравнением третьего порядка.

Лабораторная работа № 5. Моделирование асинхронных процессов с использованием сетей Петри

Сети Петри используются для моделирования асинхронных систем, функционирующих как совокупность параллельных взаимодействующих процессов. Анализ сетей Петри позволяет получить информацию о структуре и динамическом поведении моделируемой системы.

Описание сети Петри $S=(P, T, I, O, M_0)$ предполагает задание множеств: P – позиций, T – переходов, I – входящих дуг для переходов, O – исходящих дуг и M_0 – начальной маркировки позиций, определяющей количество меток в них $m(p_i)$.

Анализ сетей Петри заключается в изучении их основных свойств: безопасности, ограниченности, сохранении, активности, достижимости и покрываемости.

Безопасность

Позиция $p_i \in P$ сети $S=(P, T, I, O, M_0)$ является безопасной, если $m(p_i) \leq 1$ для любой $M \in R(C, M_0)$, где R – множество маркировок сети S , достижимых из начальной маркировки M_0 . Сеть Петри безопасна, если безопасна каждая ее позиция.

Ограниченность

Соответствующая позиция является K -безопасной или K -ограниченной, если количество меток в ней не может превысить целое число K .

Позиция $p_i \in P$ сети $S=(P, T, I, O, M_0)$ является K -безопасной, если $m(p_i) \leq K$ для всех $M \in R(C, M_0)$.

Позиция называется ограниченной, если она K -безопасна для некоторого K .

Сеть Петри ограничена, если все ее позиции ограничены.

Сохранение

Сеть Петри называется строго сохраняющей, если для всех $M \in R(C, M_0)$

$$\sum_{p_i \in P} m(p_i) = \sum_{p_i \in P} m_0(p_i)$$

Сеть Петри должна сохранять ресурсы, которые она моделирует. Однако не всегда имеется однозначное соответствие между меткой и количеством или числом ресурсов.

Активность (живость)

Тупик в сети Петри – это переход (или множество переходов), которые не могут быть запущены. Переход называется активным, если он не заблокирован, то есть может быть запущен. Сеть активна, если в ней всегда существует хотя бы один активный переход

Достижимость и покрываемость

Задача достижимости заключается в определении для маркировки M_0 возможности достижения маркировки $M \in R(C, M_0)$ путем срабатывания некоторого количества переходов.

Задача покрываемости: для сети S с начальной маркировкой M_0 и маркировки M определить, существует ли такая достижимая маркировка $M' \in R(C, M_0)$, что $M' \geq M$.

Для описания совместной работы параллельных процессов рассмотрим основные классические типы взаимодействия и синхронизации процессов.

Задача о взаимном исключении

Данная задача возникает в условиях, когда несколько процессов разделяют общую переменную. Каждый процесс может вначале считать значение переменной, затем вычислить новое значение и записать его на то же место.

Процесс 1

Процесс 2

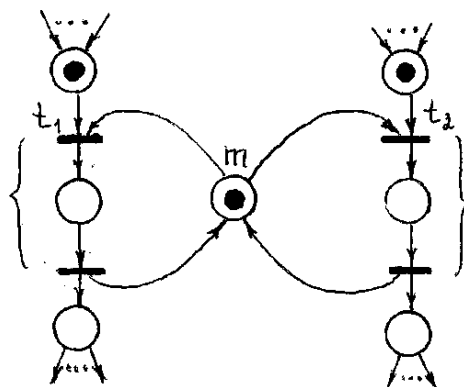


Рисунок 1 – Задача о взаимном исключении

Задача о производителе/потребителе

В этой задаче разделяемым объектом является буфер. Процесс производитель порождает компоненту информации и размещает ее в буфер. Процесс потребитель ждет пока компонента информации разместится в буфере, после этого он может ее удалить из буфера и использовать. Как правило используется буфер ограниченной емкости, то есть имеется возможность разместить в нем не более n компонентов данных.

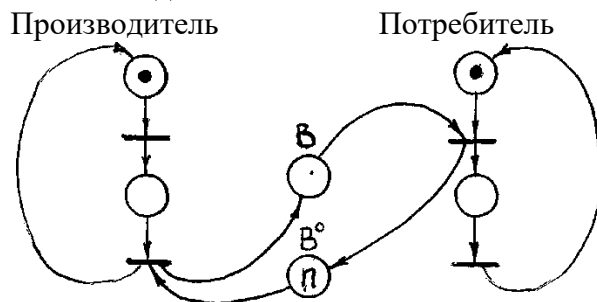


Рисунок 2 – Задача о производителе/потребителе

Задача об обедающих мудрецах

Мудрецы сидят за круглым столом и каждый из них может пребывать в одном из двух состояний: думает или обедает. На столе блюда китайской кухни, а между мудрецами лежит по одной палочке. Для приема пищи мудрец должен взять палочку слева и палочку справа. В этом случае соседи обедающего мудреца могут только думать и ждать, когда освободятся палочки, чтобы приступить к обеду. На рис.3 сеть Петри представлено взаимодействие трех процессов (трех мудрецов), состояния которых отражены позициями O_i и D_i ; (O_i – обедает, D_i – думает). Позиции p_i представляют палочки для еды (разделяемые ресурсы).

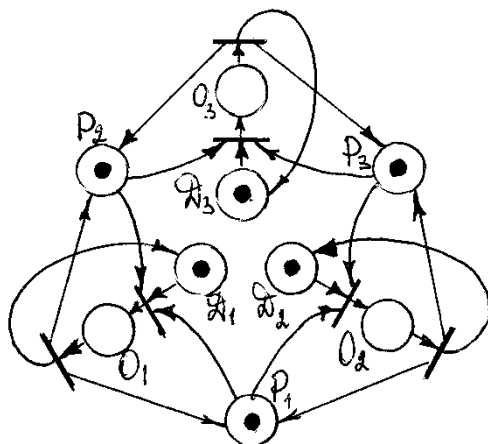


Рисунок 3 – Задача об обедающих мудрецах

Задача о чтении/записи

Рассматривается взаимодействие процессов двух типов: процессы чтения и процессы записи. Все процессы совместно используют общий файл или переменную или элемент данных. Процессы чтения не изменяют объект, а процессы записи изменяют. Поэтому процессы записи должны взаимно исключать все другие процессы чтения и записи, в то время как несколько процессов чтения могут иметь доступ к разделяемым данным одновременно. Взаимодействие процессов необходимо организовывать так, чтобы не могла возникнуть тупиковая ситуация и был обеспечен механизм взаимного исключения со стороны процессов записи.

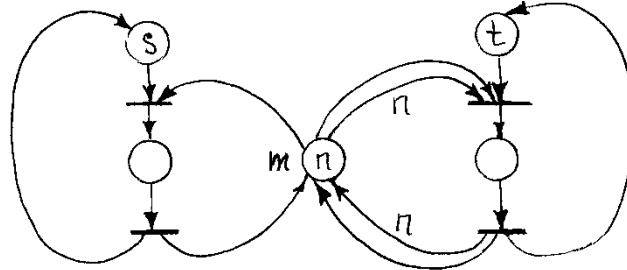


Рисунок 4 – Задача о чтении/записи

P – и V – операции над семафорами

Одним из распространенных механизмов синхронизации процессов являются P - и V - операции над семафорами. Семафор – это переменная, которая может принимать только неотрицательные целые значения. V - операция увеличивает значение на единицу, а P-операция уменьшает его на единицу. P-операция может выполняться только в том случае, когда полученное значение семафора остается неотрицательным. Таким образом, если значение семафора равно 0, то P-операция должна ждать, пока какой-нибудь другой процесс не выполнит V -операцию. P- и V -операции определены как примитивные, то есть никакая другая операция не может изменять значения семафора одновременно с ними.

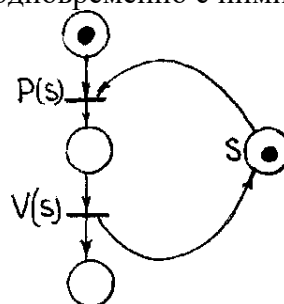
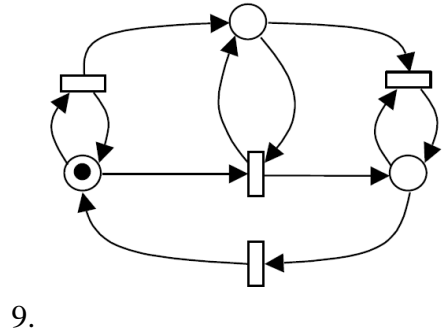
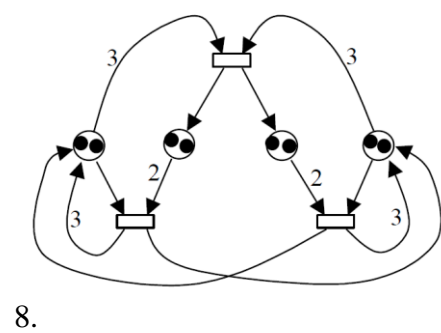
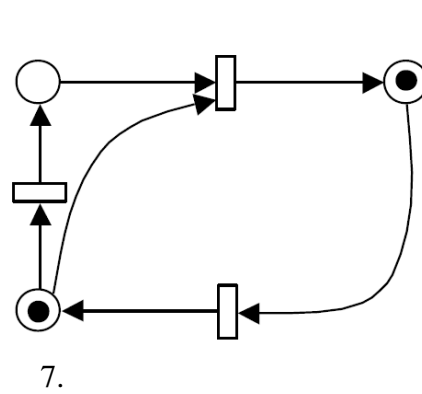
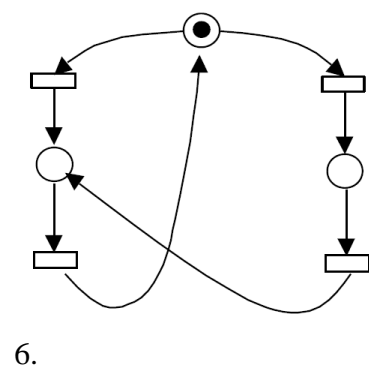
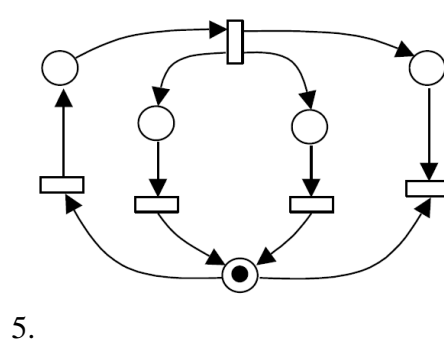
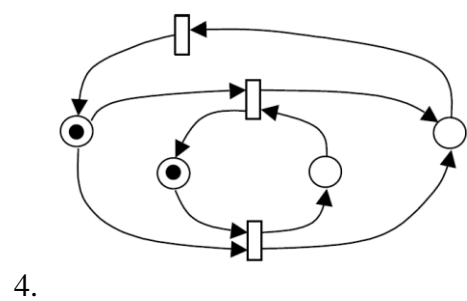
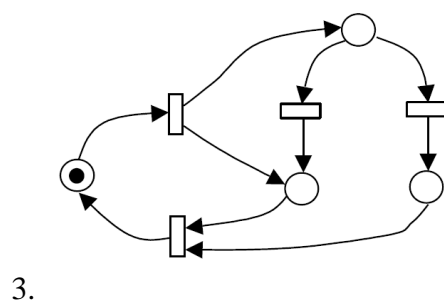
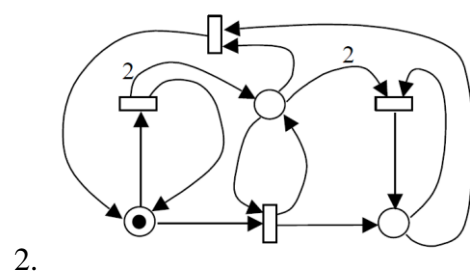
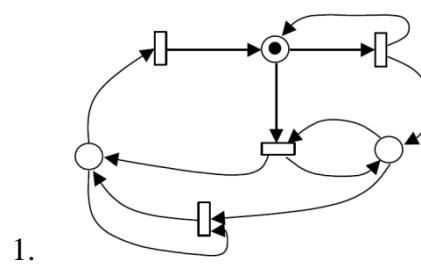
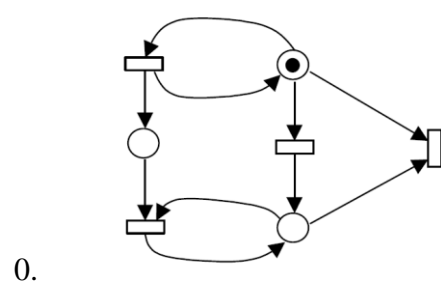


Рисунок 5 – P – и V – операции над семафорами

Задание

1. С использованием программ `cpntools` и `tina` провести моделирование классических задач для сетей Петри. Исследовать возможные свойства сетей.
2. По вариантам индивидуальных заданий изучить основные свойства предлагаемых сетей.

Варианты заданий



Лабораторная работа № 6. Матричные методы анализа сетей Петри

Матричный подход основывается на представлении сети двумя матрицами A^- и A^+ , представляющими входную и выходную функции сети. Каждая матрица имеет m строк (по одной на переход) и n столбцов (по одному на позицию).

Определим $A^-(j,i)=K(P_i,I(t_j))$, а $A^+(j,i)=K(P_i,O(t_j))$.

A^- определяет входы в переходы, A^+ - выходы. K – кратность позиции по входам и выходам.

Введём m – вектор-строку $e(j)$, содержащую нули везде, за исключением j -й компоненты. Переход t_j представляется m - вектором $e(j)$.

Тогда переход t_j в маркировке M разрешён, если $M \geq e(j)A^-$, а результат запуска перехода t_j в маркировке M записывается как:

$$\delta(M,t_j)=M - e(j) A^- + e(j) A^+ = M + e(j)(- A^- + A^+) = M + e(j)A,$$

где $A = A^+ - A^-$ - составная матрица изменений.

Тогда для последовательности запусков переходов $G = t_{j1}, t_{j2}, \dots, t_{jk}$ имеем

$$\begin{aligned} \delta(M,G) &= \delta(M, t_{j1}, t_{j2}, \dots, t_{jk}) = M + e(j_1)A + e(j_2)A + \dots + e(j_k)A = \\ &= M + [e(j_1) + e(j_2) + \dots + e(j_k)]A = M + f(G)A \end{aligned}$$

Вектор $f(G) = e(j_1) + e(j_2) + \dots + e(j_k)$ называется вектором запуска последовательности $t_{j1}, t_{j2}, \dots, t_{jk}$. i -й элемент вектора $f(G)$, $f(G)_i$ - это число запусков перехода t_j в последовательности $t_{j1}, t_{j2}, \dots, t_{jk}$. Вектор запусков, следовательно, является вектором с неотрицательными целыми компонентами.

Рассмотрим задачу сохранения при векторе-столбце взвешивания меток W , все n компонент которого равны 1. Если M_0 - начальная маркировка, а M' - произвольная достижимая маркировка, необходимо, чтобы $M_0W = M'W$. Тогда существует последовательность запусков переходов G , которая переводит сеть из M_0 в M' . Поэтому

$$M' = \delta(M_0, G) = M_0 + f(G)A$$

Следовательно, $M_0W = M'W = (M_0 + f(G)A)W = M_0W + f(G)AW$, поэтому $f(G)AW = 0$.

Поскольку это должно быть верно для всех $f(G)$, имеем $AW = 0$.

Если между меткой и количеством ресурса существует неоднозначное соответствие, то W может иметь положительные, необязательно единичные компоненты и

сеть Петри является сохраняющей тогда и только тогда, когда существует такой положительный вектор W , что $AW = 0$. Это обеспечивает простой алгоритм проверки сохранения, а также позволяет получать вектор взвешивания W для сохраняющей сети.

Матричная теория является инструментом для решения проблемы достижимости. Пусть маркировка M' достижима из M_0 , тогда существует последовательность (возможно пустая) запусков переходов G , которая приводит из M_0 к M' . Это означает, что $f(G)$ является неотрицательным целым решением следующего матричного уравнения для вектор-строки X :

$$M' = M_0 + XA \quad (1)$$

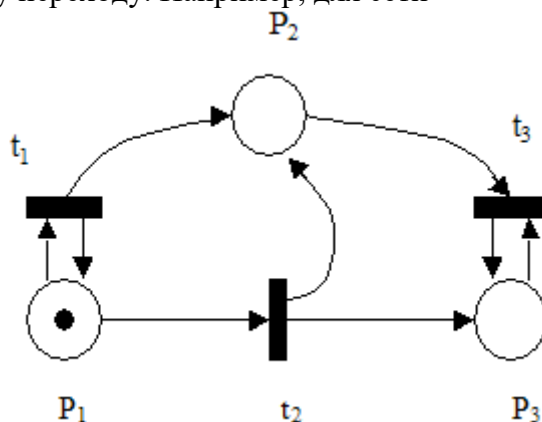
Следовательно, если M' достижима из M_0 , тогда уравнение (1) имеет решение в неотрицательных целых; если уравнение не имеет решения, тогда M' недостижима из M_0 . При этом компоненты X_j соответствуют количеству срабатываний j -го перехода при движении от разметки M_0 к M' .

Задание

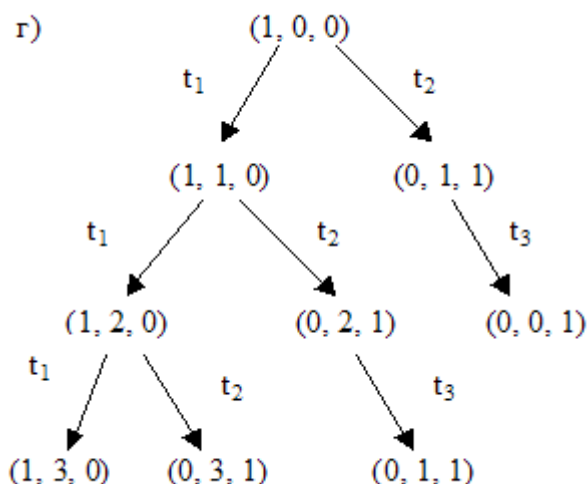
Для вариантов заданий из предыдущей лабораторной работы исследовать свойства сохраняемости и достижимости.

Лабораторная работа № 7. Исследование сетей Петри на основе дерева достижимости

Дерево достижимости представляется множеством состояний сети. Вершинами дерева являются маркировки. Корень дерева – начальная маркировка. Дуги дерева соответствуют некоторому срабатываемому переходу. Например, для сети



четыре первых уровня дерева достижимости имеют вид



Чтобы избежать бесконечного разрастания дерева его приводят к конечному виду на основе следующего алгоритма.

Каждая вершина i дерева связывается с расширенной маркировкой $M(i)$. В этой маркировке число меток в позиции может быть либо неотрицательным целым, либо неограниченно расти и обозначаться символом ω . Каждая вершина классифицируется как *граничная* или *терминальная* или *дублирующая* или *внутренняя*.

Граничными являются вершины, которые не обработаны алгоритмом. Алгоритм превратит их в терминальные, дублирующие, внутренние.

Алгоритм начинает с начальной маркировки M_0 , принимая ее в качестве граничной вершины.

Пусть x – граничная вершина.

1. Если в дереве имеется другая вершина y , не являющаяся граничной, и с ней связана также маркировка $M(x)=M(y)$, то вершина x – дублирующая.

2. Если для маркировки $M(x)$ ни один из переходов не разрешен для всех $t_j \in T$, то x – терминальная вершина.

3. Для всякого перехода $t_j \in T$, разрешенного в $M(x)$, необходимо создать новую вершину z дерева. Маркировка $M(z)$ определяется для каждой позиции P_i следующим образом:

а) если $M(x)_i = \omega$, то $M(z)_i = \omega$

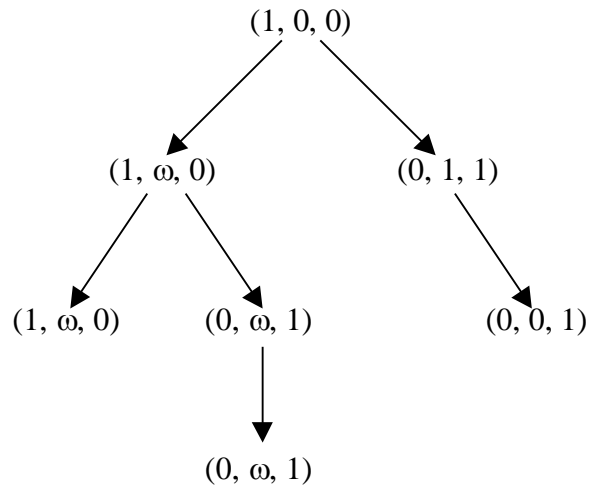
б) если на пути от корневой вершины к x существует вершина y с $M(y) < \delta(M(x), t_j)$ и $M(y)_i < \delta(M(x), t_j)_i$, то $M(z)_i = \omega$, δ -функция следующего состояния.

в) в противном случае $M(z)_i = \delta(M(x), t_j)_i$,

Функция $\delta(M(x), t_j)$ не определена, если t_j не разрешен в маркировке $M(x)$. Если t_j разрешен, то $\delta(M(x), t_j) = M'(x)$, где $M'(x)$ – маркировка, полученная после срабатывания t_j . Дуга, помеченная t_j , направлена от вершины x к вершине z . Вершина x переопределяется как внутренняя, вершина z становится граничной.

4. Когда все вершины дерева – терминальные, дублирующие или внутренние, алгоритм останавливается.

Для нашего примера конечное дерево достижимости имеет вид:



Задание

1. Сформулировать зависимость основных свойств сети Петри от вида конечного дерева достижимости и типа его вершин.

2. Исследовать свойства сети по вариантам индивидуальных заданий. Сравнить с результатами предыдущей лабораторной работы.

Лабораторная работа № 8. Анализ сетей Петри с использованием инвариантов

Цель работы – провести анализ сети Петри с использованием Р- и t- инвариантов.

Целочисленный вектор-столбец $Y = \{Y_i\}$, $i=1,2,\dots,n$, являющийся решением линейной системы

$$AY = 0, \quad (2)$$

где A – матрица изменений разметки сети, называется *Р-инвариантом* или *Р-циклом*. Р-инвариант характеризует некоторое свойство распределения меток по позициям сети.

Умножая уравнение (1) на Y справа, получим $M'Y = M_0Y = \text{const}$ – свойства инвариантности распределения меток.

Если все компоненты Р-цикла неотрицательны, его называют Р-цепью. полная Р-цепь – это Р-цепь, все компоненты которой положительны.

Сеть Петри инвариантна, если для неё существует полная Р-цепь. Полная Р-цепь включает в себя все позиции сети. Доказывается, что *инвариантная* сеть Петри является *ограниченной*.

Целочисленный вектор-столбец X называется *t-инвариантом*, если он является решением линейной системы алгебраических уравнений

$$A^T X = 0 \quad (3)$$

Из уравнения (1) при выполнении (3) следует, что при некоторой ненулевой последовательности X срабатываний переходов сеть возвращается к исходной маркировке:

$$M' = M_0,$$

то есть сеть устойчива.

Таким образом, если сеть имеет полную t-цепь (все элементы вектора X *положительны*), то она жива при любой начальной маркировке, поскольку в цепи маркирований от M_0 до $M' = M_0$ срабатывают все переходы.

По устойчивости процессов и значению t-инварианта можно определить живость сети: если сеть жива и ограничена, то она инвариантна и устойчива.

Задание

1. Исследовать свойства живости и ограниченности сети по вариантам индивидуальных заданий.

Литература

1. Лазарев Ю. Моделирование процессов и систем в MatLab / Ю. Лазарев. – СПб.: Питер, 2005. – 512 с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем / Дж. Питерсон. – М.: Мир, 1984. – 264 с.
3. Зайцев Д.А. Сети Петри и моделирование систем / Д.А. Зайцев. Методические указания к практическим занятиям и лабораторным работам. Украинская национальная академия связи им. О. С. Попова, 2006.
4. Алексеев Е.Р. Scilab: Решение инженерных и математических задач / Е.Р. Алексеев, О.В. Чеснокова, Е.А. Рудченко. – М. : ALT Linux ; БИНОМ. Лаборатория знаний, 2008. – 260 с.