Национальный технический университет
«Киевский политехнический институт»
Факультет информатики и вычислительной техники
Кафедра Вычислительной Техники

Расчётно-графическая работа

по дисциплине:
Компьютерное моделирование
на тему:
«Исследование временных процессов в стохастических сетях Петри»

Выполнил:
Студент 3-го курса
Факультета ФИВТ
Группы ИО-93
Свинарчук Сергей
Вариант: 9316

Киев, 2011

Содержание.

I. Исследование моделей информационных потоков.
II. Задание на работу.
III. Режим графического редактора.
IV. Построение марковского графа и дерева достижимости.
V. Моделирование ССП.
VI. Моделирование log-сервера и сервера приложений.
VII. Моделирование сервера безопасности.
VIII. Приложение.
8.1. Классы графических элементов.
8.2. Классы графического редактора.
8.3. Классы для представления ССП и марковских графов.
8.4. Классы для генерирования случайных потоков.
8.5. Классы для моделирования ССП.
8.6. Классы мейнфрейма.

I. Исследование моделей информационных потоков.

Общие теоретические сведения. В информационных потоках мы оперируем таким понятием как τ_i[сек.] – время между случайными событиями в информационном потоке. $M\tau = 1/\lambda$ - ср. значение для τ_i, где λ[событий в сек.] $\approx 0.01 \div 100$ - интенсивность потока. Коэффициент вариации потока $g = \frac{D\tau}{M\tau^2}$ является главной характеристикой потока, поскольку зная коэффициент вариации мы можем определить какой это поток.

Поток с равномерным распределением

При моделировании случайных величин особое место занимает непрерывное распределение, называющееся равномерным. Функция плотности вероятности случайной величины, имеющей равномерное распределение на интервале (a, b), имеет вид:

$$\mathbf{f(x)} = \begin{cases} \dfrac{1}{\mathbf{b-a}}, \mathbf{a \le x \le b} \\ 0, \quad \mathbf{x < a, x > b} \end{cases} \qquad (2.1)$$

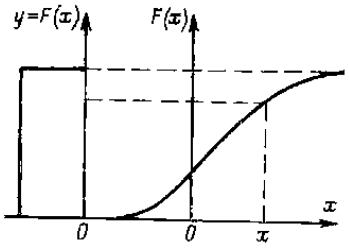


Рис.1. Принцип преобразования равномерного распределения в заданное

Большинство способов моделирования случайных величин основано на использовании псевдослучайных равномерно распределенных в интервале (0,1) последовательностей чисел. При этом в основу положена следующая известная в математической статистике теорема**:** если случайная величина *X* имеет плотность распределения *f(x),* то распределение случайной величины *Y=F(x)* является равномерным в интервале (0, 1)

Таким образом, задаваясь функцией распределения *F(x),* можно выбирать случайное значение *Y* из равномерного распределения в интервале *(0,1)* и определять значение аргумента, для которого *F(x)=Y* (рис.2.1). Полученная таким образом случайная величина *X* будет иметь заданную функцию распределения *F(x).* Эта операция может быть представлена аналитически следующим выражением:

$$\int_{-\infty}^{\mathbf{x_i}} \mathbf{f(x)dx} = \mathbf{y_i} \qquad (2.2)$$

согласно которому определяется значение x_i, соответствующее значению функции распределения, равному y_i. Для некоторых частных законов распределения уравнение (2.2) удается решить непосредственно, в других случаях прибегают к приближенным способам решения, в частности к аппроксимации подынтегральной функции полиномами, интегрируемыми в квадратурах.

Моделирование дискретных случайных величин с известным распределением вероятностей в общем случае производится следующим образом. Предположим, что заданы численные значения вероятностей $p_1, p_2, ..., p_n$ для независимых событий $A_1, A_2, ..., A_n$, составляющих полную группу. В соответствии с этим интервал (0,1) разбивается на ***n*** отрезков так, чтобы длина *i-го* отрезка равнялась вероятности p_i. Производя выборку случайных чисел ξ_i из равномерного распределения в интервале (0,1), будем определять, на какой отрезок попадает число ξ_i. Попадание числа на *i-й* отрезок фиксируется как факт свершения события A_i.

Другой способ моделирования дискретных величин состоит в формировании интервалов между моментами наступления соседних событий. При этом задача сводится к уже описанному выше случаю моделирования непрерывной случайной величины.

Математическое ожидание равномерного закона равно

$$Mx = \frac{\sum Wi}{N}$$

Дисперсия равномерного закона равна

$$Dx = \frac{(W_i - Mx)^2}{N}$$

Поток вычисляется по формуле:

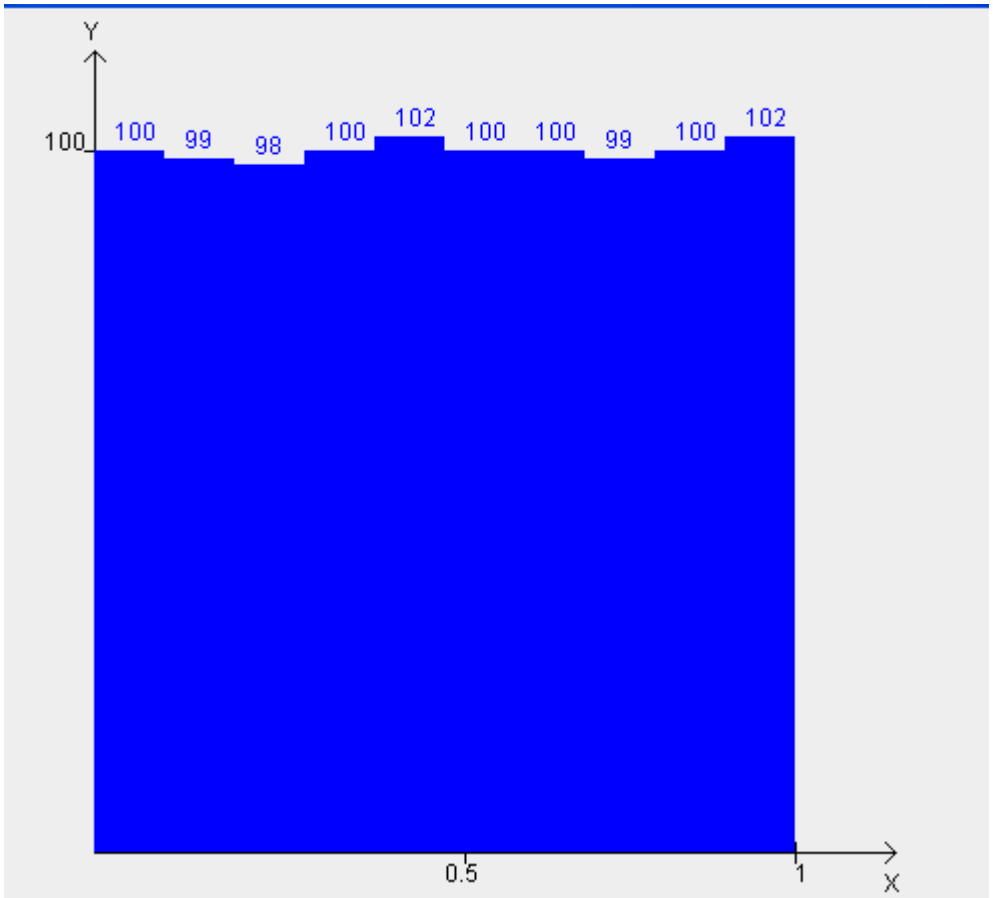
$$W_{i+1} = (W_i \cdot A + C) / D$$

где W_0 - случайное число от 0 до 1, $W_0 = 12$.

Таблица лучших результатов:

№	A	C	D	W0	Мат. Ожидание	Дисперсия
1	60	2392591	2147483647	12	0.5123%	0.234
2	1	2133377	2147483647	12	0.481%	0.3123

Результат при A=60; **C**=2133377; **D**=2147483647**:**



Поток с экспоненциальным распределением (Поток Пуассона)

Экспоненциальное распределение. Непрерывная случайная функция, распределенная по экспоненциальному закону, имеет функцию распределения *F(x)* и плотность распределения *f(x)* вида:

$$\mathbf{F(x)} = \begin{cases} 0 & \mathbf{x} < 0; \\ 1 - \mathbf{e}^{-\lambda \mathbf{x}}, & \mathbf{x} \geq 0; \end{cases} \qquad (2.3)$$

$$\mathbf{f(x)} = \begin{cases} 0 & \mathbf{x} < 0; \\ \lambda \mathbf{e}^{-\lambda \mathbf{x}}, & \mathbf{x} \geq 0; \end{cases} \qquad (2.4)$$

Значения математического ожидания и дисперсии для экспоненциального закона распределения равны соответственно $1/\lambda$ и $1/\lambda^2$.

Экспоненциальное распределение занимает в теории массового обслуживания особое место. Это связано с тем, что распределение многих величин, таких, например, как интервалы между наступлениями соседних событий, времена обслуживания и др., во многих задачах массового обслуживания принимаются имеющими экспоненциальное распределение. Хотя реальные распределения (в частности распределение времени обслуживания) часто существенно отличаются от экспоненциального, использование последнего позволяет намного упростить решение задач. Кроме того, использование экспоненциального распределения позволяет получить во многих случаях оценку для более напряженного режима обслуживания.

Для моделирования случайных величин x_i ,имеющих экспоненциальное распределение, можно воспользоваться непосредственным решением уравнения (2.2). Действительно, с учетом (2.4) получаем:

$$\lambda \int_0^{\mathbf{x_i}} \mathbf{e}^{-\lambda \mathbf{t}} \mathbf{dt} = \xi_{\mathbf{i}} \xi \qquad (2.5)$$

После интегрирования имеем:

$$1 - \mathbf{e}^{-\lambda \mathbf{x_i}} = \xi_{\mathbf{i}} \qquad (2.6)$$

Поскольку случайная величина $(1-\xi_i)$ имеет также равномерное распределение в интервале (0,1) окончательно получаем:

$$\mathbf{x_i = -(1/\lambda)\ln \xi_i = -\theta \ln \xi_i} \qquad (2.7)$$

где альтернативной параметризацией является параметр масштаба $\theta=1/\lambda$.

Поток Пуассона

<u>Определение.</u> Если поток событий стационарен, ординарен и без последействий, то такой поток называется **простейшим (пуассоновским)** потоком.
Пусть T – промежуток времени между двумя произвольными соседними событиями в простейшем потоке. Найдем функцию распределения:

$$F(t) = P(T < t)$$

В соответствии с законом распределения Пуассона, получаем:

$$F(t) = 1 - e^{-\lambda t} \qquad f(t) = \lambda e^{-\lambda t}$$

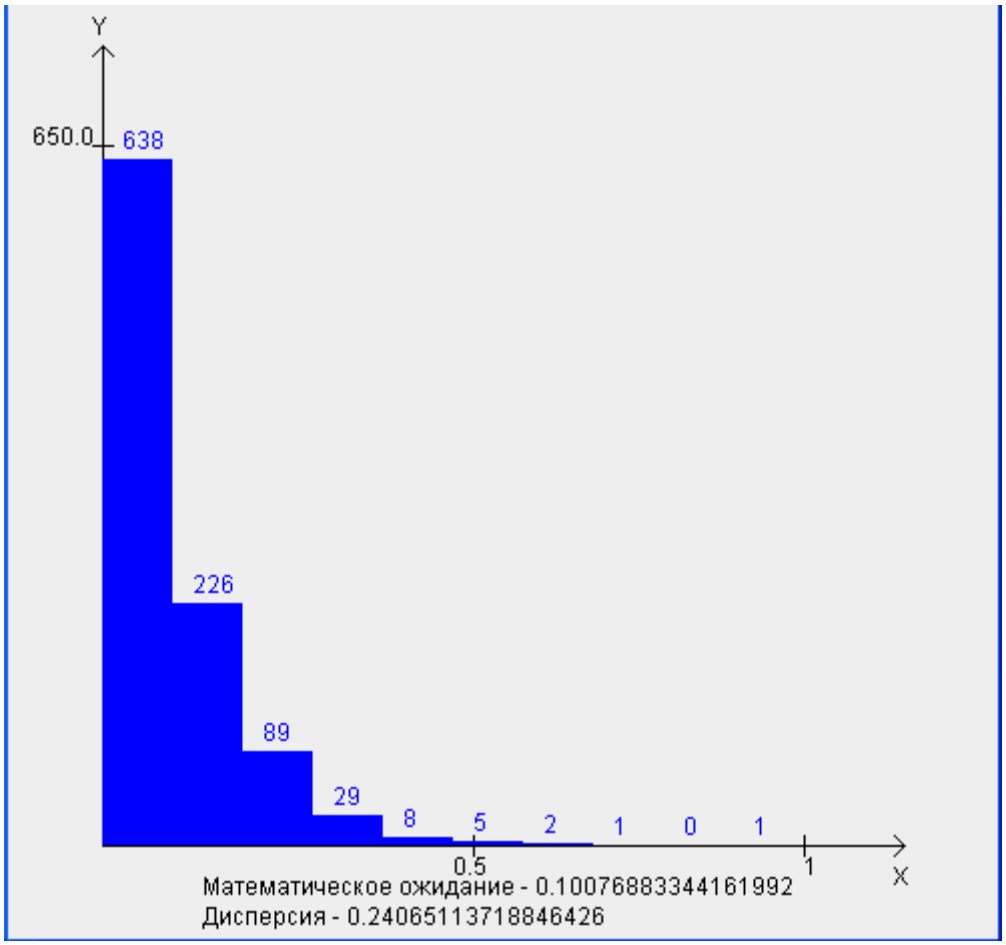
Математическое ожидание, дисперсия и среднее квадратическое отклонение этой величины соответственно равны:

$$m_t = \frac{1}{\lambda}; \; D_t = \frac{1}{\lambda^2}; \quad \sigma_t = \frac{1}{\lambda};$$

Лучшие результаты :

A	C	D	λ	Математическое ожидание	Дисперсия	Математическое ожидание (теор.)	Дисперсия (теор.)
65	37465	2147483647	10	0,0992	0,2419	0,1	0,01
81	14249	2147483647	10	0,1	0,240	0,1	0,01
1001	42741	2147483647	12	0,0813	0,254	0,083	0,0069
6021	10589	98745254	11	0,0886	0,247	0,09	0,0082
16801	78589	987459852	13	0,076	0,260	0,076	0,0059
16801	91241	987459852	10	0,098	0,241	0,1	0,01
4401	194257	78954216	14	0,064	0,2557	0,071	0,0051

Результат при A = 81, C = 14249, D = 2147483647, λ = 10:



Поток Эрланга

Потоки Эрланга также являются потоками с ограниченным последействием. Они образуются просеиванием простейшего потока.

Суть этого просеивания состоит в следующем. Если изобразить на временной оси простейший поток, поставив в соответствие каждому событию некоторую точку, и выбросить из потока каждую вторую точку, то получим **поток Эрланга первого порядка.** Оставив каждую третью точку и выбросив две промежуточные, получаем **поток Эрланга второго порядка** и т.д.

<u>Определение.</u> Потоком Эрланга *k* – порядка называется поток, получаемый из простейшего, если сохранить в простейшем потоке каждую (k + 1) – ю точку, а остальные выбросить.

Очевидно, что простейший поток может рассматриваться как поток Эрланга нулевого порядка.

Пусть имеется простейший поток с интервалами T_1, T_2, … между событиями. Величина T – промежуток времени между двумя соседними событиями в потоке Эрланга k – го порядка.

Очевидно, что $T = \sum_{i=1}^{k+1} T_i$. Так как первоначальный поток – простейший, то случайные величины T_1, T_2, … распределены по показательному закону:

$$f(t) = \lambda e^{-\lambda t};$$

Вероятность первого события равна λdt, а второго - $P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}$. Эти события должны осуществиться совместно, значит, их вероятности надо перемножить.

$$f_k(t)dt = \frac{\lambda(\lambda t)^k}{k!} e^{-\lambda t} dt$$

$$f_k(t) = \frac{\lambda(\lambda t)^k}{k!} e^{-\lambda t}$$

Полученный закон распределения называется **законом распределением Эрланга *k*- го порядка**.

При $k = 0$ получаем показательный закон распределения.

Математическое ожидание, дисперсия и среднее квадратическое отклонение для распределения Эрланга находятся по формулам:

$$m_k = \frac{k+1}{\lambda}; \qquad D_k = \frac{k+1}{\lambda^2}; \qquad \sigma_k = \frac{\sqrt{k+1}}{\lambda};$$

Плотность потока Эрланга равна

$$\Lambda_k = \frac{\lambda}{k+1};$$

Для промежутка времени между двумя соседними событиями в потоке T рассмотрим нормированную величину $\widetilde{T} = \dfrac{T}{k+1}$. Такой поток будет называться **нормированным потоком Эрланга**.

Закон распределения для такого потока будет иметь вид:

$$\tau_i = \frac{1}{k * \lambda} \sum_j^k \ln(W_{i-j}), \text{ k=1/g}$$

Математическое ожидание и дисперсия будут равны:

$$\widetilde{m}_k = m_0 = \frac{1}{\lambda}; \qquad \widetilde{D}_k = \frac{D_k}{(k+1)^2} = \frac{1}{\lambda^2 (k+1)};$$

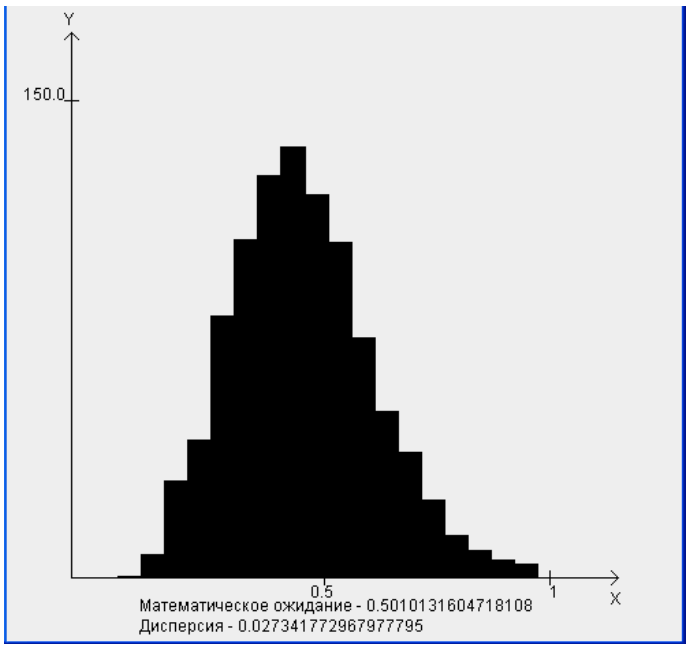
Получается, что неограниченном увеличении k нормированный поток Эрланга приближается к регулярному потоку с постоянными интервалами, равными $\dfrac{1}{\lambda}$.

Изменение порядка нормированного потока Эрланга позволяет получить различную степень последействия. Последействие возрастает с увеличением k.

Лучшие результаты:

A	C	D	λ	k	Mx	Dx	Mx (теор.)	Dx (теор.)	ΔMx	ΔDx
65	37465	2147483647	11	2	0,4250106	0,041322	0.5	0,041322	0,00106	0,027911
81	14249	2147483647	10	3	0,32925	0,012506	0,3333	0,003333	0,00405	0,009173
1001	42741	2147483647	10	2	0,501013	0,027341	0,5	0,005	0,001013	0,012341
6021	10589	98745254	11	2	0.45086	0.020689	0.5	0.004132	0,04914	0,016557
16801	78589	987459852	13	3	0.255541	0.007294	0.3333	0.001972	0,077759	0,005322
16801	91241	987459852	10	2	0.62258	0.18705	0.5	0.005	0,12258	0,18205
4401	194257	78954216	14	4	0.19978	0.003985	0.25	0.0012755	0,05022	0,0027095

Результат при A = 1001, C = 42741, D = 2147483647, λ = 10, K=2:



Гиперэкспоненциальный поток

Иногда исследование распределения интервалов времени между событиями потока даёт оценку коэффициента вариации существенно больше 1. В этом случае использование модели потока Пуассона опять таки неприемлимо, поскольку это внесёт существенные искажения в результаты моделирования. В данном случае необходимо использовать поток частых событий. Одним из вариантов такого потока является гиперэкспоненциальный поток.

Для упрощения теоретических расчётов в работе используется упрощённая модель гиперэкспоненциального потока с непрерывной фазой. Фаза определяет вероятность обслуживания в более медленном канале рассматриваемой двухканальной системы.

Гиперэкспоненциальный поток применяется тогда, когда в системе имеется несколько каналов обработки событий, которые работают с разной скоростью и вероятность попадания на каждый из которых задана. Логично, что при наличии только одного такого канала поток становится Пуссоновским.

$$F(\tau) = 1 - \sum_{i=1}^{k} a_i e^{-\alpha_i \tau}$$

$$f(\tau) = \sum_{i=1}^{k} \alpha_i a_i e^{-\alpha_i \tau}$$

$$\tau_i = -\frac{1}{\alpha} \ln W_2$$

Несмотря на простоту описания этой модели получить аналитические оценки основных характеристик очень сложно

Наиболее удобным вариантом модели этого потока является упрощенная модель гиперэкспоненциального потока с непрерывной фазой φ.

φ - определяет физически вероятность обслуживания в медленном канале двухканальной системы.

Интегральный закон распределения для такой модели

$$F(\tau) = 1 - \varphi e^{-2\varphi\alpha\tau} + 2(1-\varphi) e^{-2(1-\varphi)\alpha\tau}$$

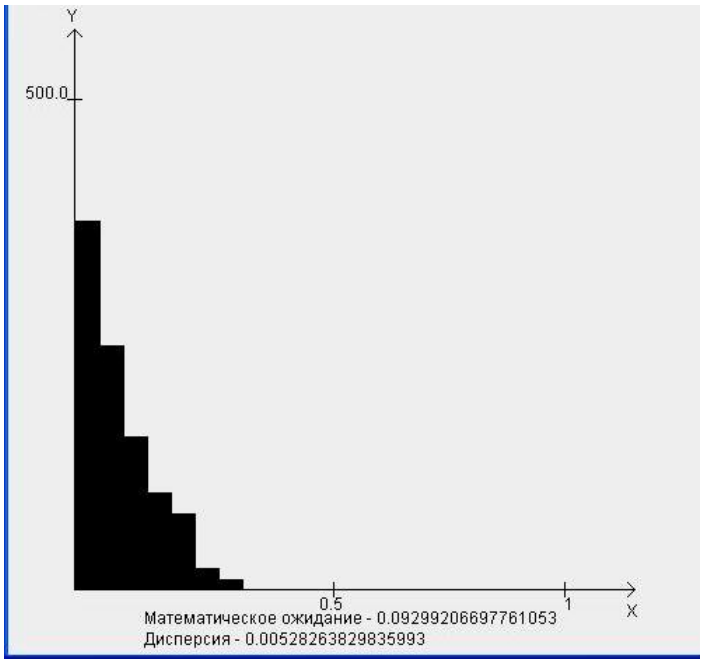
Дифференциальный закон распределения данной модели

$$f(\tau) = 2\varphi^2 \alpha e^{-2\varphi\alpha\tau} + 2(1-\varphi)^2 \alpha e^{-2(1-\varphi)\alpha\tau}$$

Лучшие результаты:

Параметры					Гиперэкспонен.		Экспонен.		Теорет.	
A	C	D	λ	G	Мат. ожид.	Дисп.	Мат. ожид.	Дисп.	Мат. ожид. (теор.)	Дисп. (теор.)
65	37465	2147483647	10	1,5	0,0857	0,0044	0,0992	0,2419	0,1	0,01
81	14249	2147483647	10	1,5	0,0864	0,0047	0,1	0,240	0,1	0,01
1001	42741	2147483647	12	1,5	0,0729	0,0034	0,0813	0,254	0,083	0,0069
6021	10589	98745254	11	1,5	0,0792	0,0037	0,0886	0,247	0,09	0,0082
16801	78589	987459852	13	1,5	0,0667	0,0027	0,076	0,260	0,076	0,0059
16801	91241	987459852	10	1,5	0,1009	0,0156	0,098	0,241	0,1	0,01
4401	194257	78954216	14	1,5	0,0661	0,0026	0,064	0,2557	0,071	0,0051

Результат при A = 4401, C = 194257, D = 78954216, λ = 14, G = 1.5:



II. Задание на работу.

Цель работы: Разработать имитационную модель организации выполнения заданий в компьютерной системе реального времени с многозадачной и многопользовательской операционной системой на основе интерпретации развития вычислительных процессов на языке асинхронных стохастических сетей Петри.

Спецвопрос: Построить дерево достижимости стохастической сети Петри и на основе формальных преобразований перейти к анализу структуры и параметров случайного процесса, характеризующего поведение системы ситуационного управления при многократном выполнении однотипных заданий, с последующей количественной оценкой вероятностей переходов и вероятностей состояний в однородной марковской цепи.

Теория: Общая структура управляющей компьютерной системы, связанной с объектом и группой терминалов операторов представлены на рис. А.1.

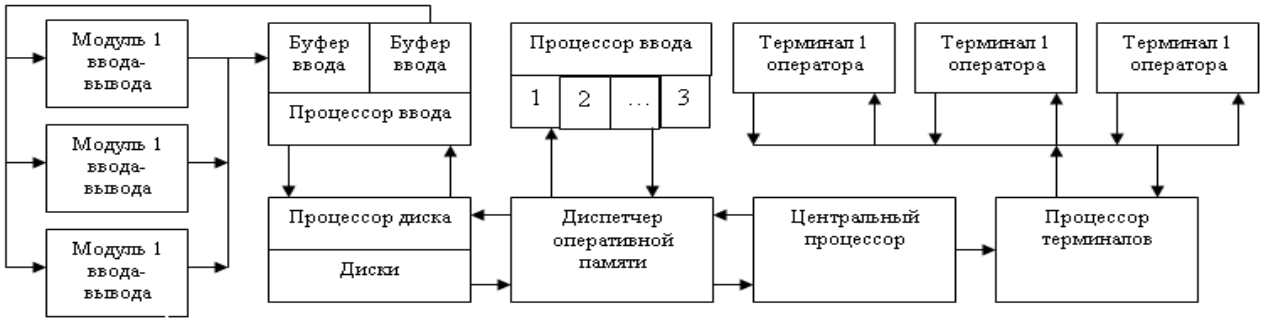


Рис. А.1. Структура исследуемой системы поддержки принятия решений группой операторов – лиц принимающих решения (ЛПР)

Прикладные программы (файлы) пользователей занимают те или иные ресурсы системы в соответствии с многократно циклически выполняемыми этапами заданий, определяемых по варианту (табл. А.3). Среди вариантов структур, включающих отдельные ресурсы системы для выполнения типовых заданий, выделяются следующие:

Структура ИЭ – измерительного эксперимента, включает: средство ввода-вывода, связанные с объектом на основе процессора ввода-вывода (ПВВ) и его буферной памяти: буфера вывода (Бвыв) и буфера ввода (БВв), а также процессора диска (ПД), на который записываются данные измерений (Бвыв-ПВВ-БВв-ПД).

Структура ПЭ – планируемого эксперимента, в котором дополнительно используется диспетчер памяти (ДП) и соответствующие разделы оперативной памяти (Бвыв-ПВВ-БВв-ПД-ДП).

Структура АО (аналитической обработки), в которой активно используется процессор диска, диспетчер памяти и выделенные разделы оперативной памяти, а также центральный процессор (ЦП), т.е. структура определяемая как (ПД-ДП-ЦП).

Структура ПР (принятия решений), включающая все средства аналитической обработки и процессор терминалов (ПТ) для связи с операторами (ПД-ДП-ЦП-ПТ).

Структура ПК (проведения конференций), включающая процессор терминалов, центральный процессор и диспетчер памяти, и обеспечивающая интерактивный режим взаимодействия операторов (ДП-ЦП-ПТ).

Неиспользуемые ресурсы исключаются из структуры системы в модели. Начальное размещение файлов на устройствах выбирается по варианту.

Одним из этапов формирования имитационной модели технической системы является создание формальной структуры системы, например, как системы массового обслуживания (СМО). Принимая очереди заданий простейшими, составим модель рассматриваемой системы как системы массового обслуживания следующего вида (рис. А.2).

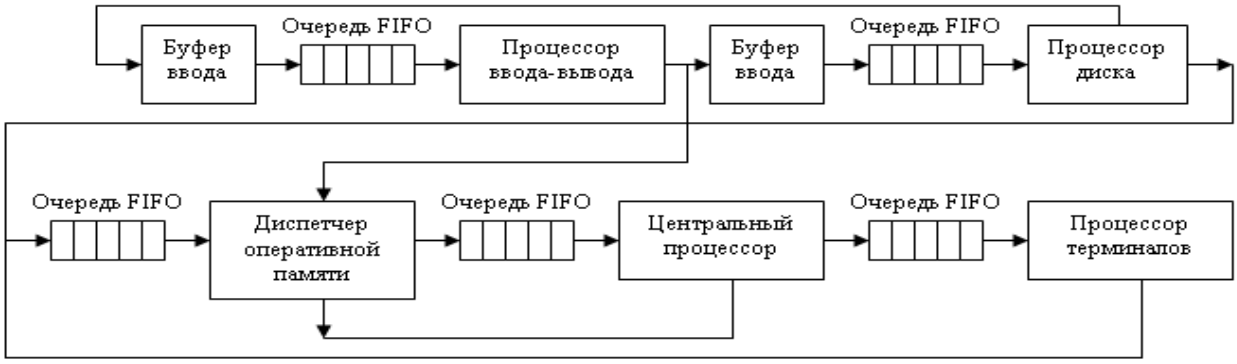


Рис. А.2. Формальная структура исходной системы в виде системы массового обслуживания потоков заданий ЛПР и их файлов-программ

Наиболее эффективным способом исследования такой не простой модели СМО является имитационное моделирование случайного процесса, который порождается в много приборной СМО с обратными связями (сети с очередями) и различными характеристиками потоков входных и обслуженных заданий при различной длительности обслуживания в каждом приборе. Среди средств описания имитационных моделей формальных дискретных систем выделяется мат. аппарат временных стохастических сетей Петри (ССП), которые относятся к семантическим сетям и отличаются высокой информативностью и простотой выполнения моделей.

На рис. А.3 представлена модель формальной системы массового обслуживания на языке временных стохастических сетей Петри.

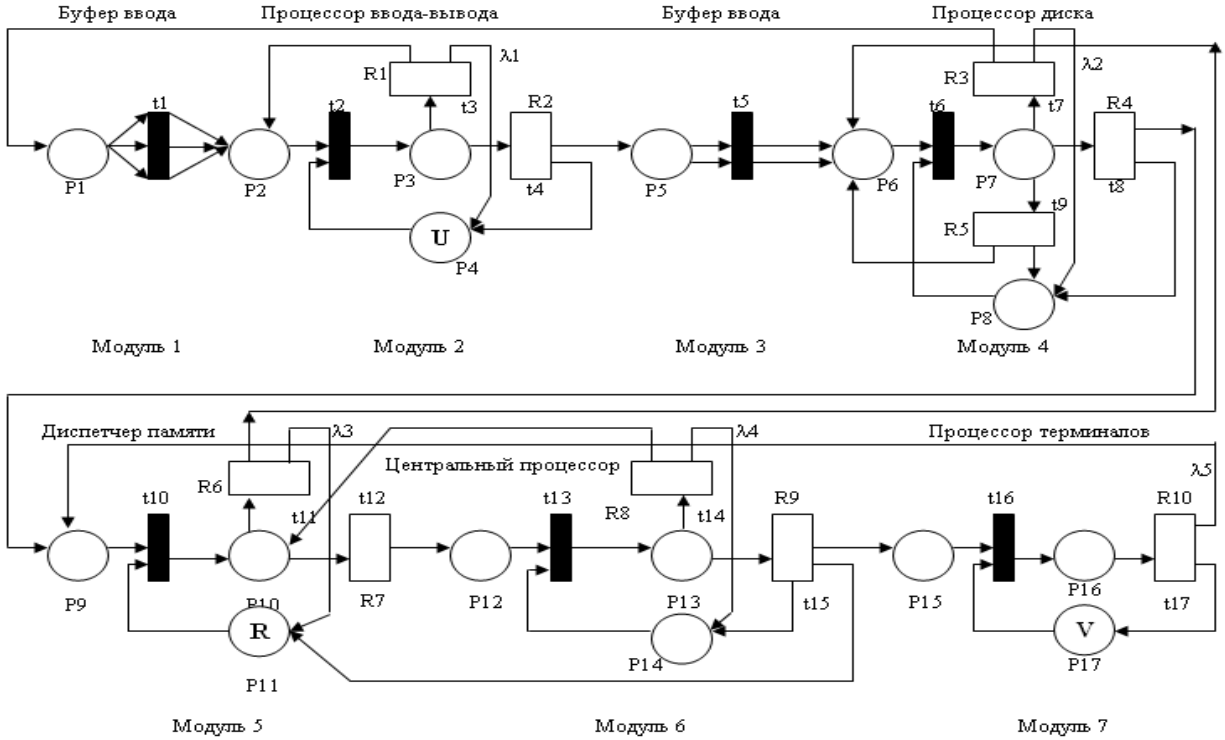


Рис. А.3. Обобщенная структура Petri-nets модели исследуемой системы как системы массового обслуживания

Для описания работы модели на основе сетей Петри достаточно задать две основных таблицы (табл. А.1 и табл. А.2) назначения фишек и интерпретации событий.

Для исследования в качестве исходной используется обобщенная Petri-nets модель системы, которую необходимо сократить в соответствии с заданной структурой и многократно выполняемыми этапами заданий и используемыми ресурсами (по вариантам из табл.А.3 и табл. А.4). Варианты структурной S-системы и соответствующей Petri-nets модели выбираются по таблице А.3: система ИЭ – измерительного эксперимента, система ПЭ – планируемого

эксперимента, система АО – аналитической обработки, система ПР – принятия решений, система ПК – проведения конференций. В задании начальной маркировки сети табл. А.3:

- U – общее количество используемых модулей ввода-вывода информации в связях с объектом;
- R – количество разделов оперативной памяти для реализации многозадачного режима обработки информации центральным процессором;
- V – количество терминалов пользователей, которые участвуют в принятии решений;
- c – количество файлов-заданий, обслуживаемых процессором ввода-вывода в начальный момент времени – T_0 моделирования;
- d – количество файлов-заданий, обслуживаемых диспетчером памяти в начальный момент времени – T_0 моделирования;
- e - количество файлов-заданий, обслуживаемых процессором терминалов в начальный момент времени – T_0 моделирования;
- L – количество (входных и выходных дуг первого перехода) файлов-заданий накапливаемых в буфере вывода для последующей обработке выводными устройствами.

Для формирования случайных значений интервалов между событиями запуска-срабатывания временных переходах Petri-nets модели используются генераторы событий информационных потоков с тем или иным распределением в зависимости от коэффициента вариации g, которые выбираются по вариантам в таблице А.4 на основе заданных интенсивностей $\lambda_1\lambda_2\lambda_3\lambda_4\lambda_5$ информационных потоков обслуживания запросов в соответствующих приборах:

- λ_1 – процессором ввода-вывода (с коэффициентом вариации в потоке g = 1/5);
- λ_2 – процессором диска (с коэффициентом вариации в потоке g = 1);
- λ_3 – диспетчером памяти (с коэффициентом вариации в потоке g = 1/3);
- λ_4 – центральным процессором (с коэффициентом вариации в потоке g = 8);
- λ_5 – процессором терминалов (с коэффициентом вариации в потоке g = 1);

Для завершения моделирования должно быть определено общее время моделирования T_{mod}, которое необходимо задавать произвольно перед проведением модельного эксперимента, как и времена промежуточных отчетов по результатам моделирования.

Отчет: Согласно варианту выписываем свои параметры модели:

Номер зачётной книги 9316. Свинарчук Сергей Владиславович.

Таблица А.3. Параметры структуры имитационной модели системы управления

Вариант по 2-м последним цифрам	**Вариант структуры**	**Внешних устройств**	**Разделов памяти**	**Терминалов операторов**	**Размещение файлов и обьем буфера вывода (дуг)**			
	S	**U**	**R**	**V**	**c**	**d**	**e**	**l**
16	ПР	-	4	2	-	3	1	-

Таблица А.4. Параметры потоков обслуживания задач управления объектом

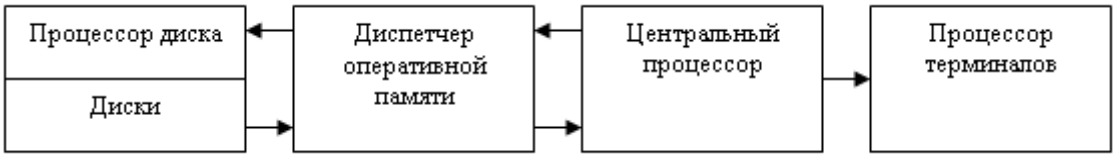
Вариант по 1-ой букве отчества	**Процессора ввода-вывода**	**Процессора диска**	**Диспетчера памяти**	**Центрального процессора**	**Процессора терминалов**
	λ1 (1/с)	**λ2(1/с)**	**λ3 (1/мс)**	**λ4 (1/мс)**	**λ5 (1/с)**
В	0,04	80	10	50	2

Таблица А.7. Параметры структуры имитационной модели

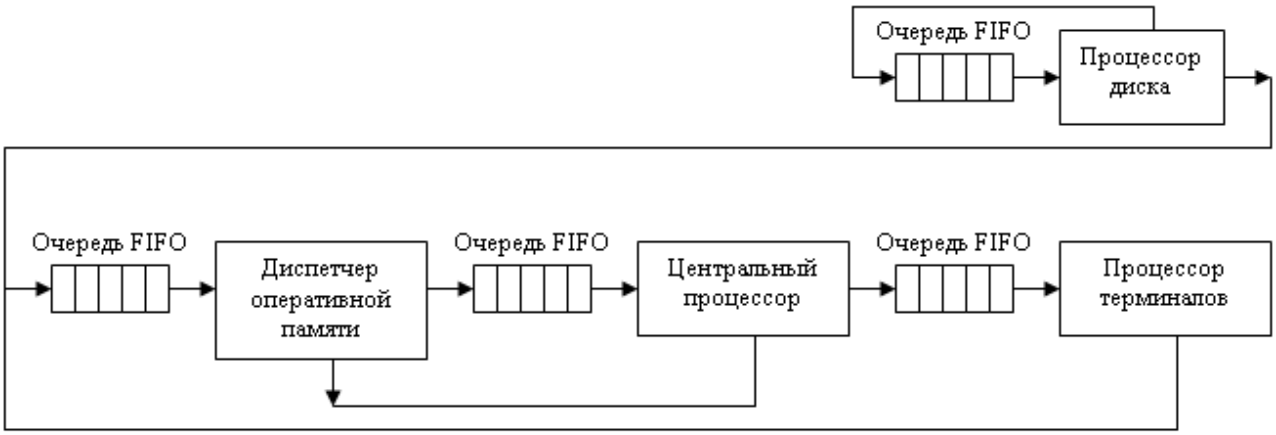
Вариант по 1-ой букве фамилии	**Вариант структур**	**Запросов сессии**	**Количество путей**	**Зарег. пользователей**	**Количество ключей**	**Время расчета Кл**
	A	**U**	**S**	**N**	**K**	**Тк (мс)**
С	4	4	3	1	2	0,05

Таблица А.8. Параметры потоков обслуживания запросов

Вариант по 1-ой букве имени	В окне приложения	Маршрутизатор	Трафик в сети	Log-сервер	Сервер приложения
	λ_1(1/c)	λ_2(1/c)	λ_3(1/c)	λ_4(1/c)	λ_5(1/c)
С	0,9	2	8	12	4

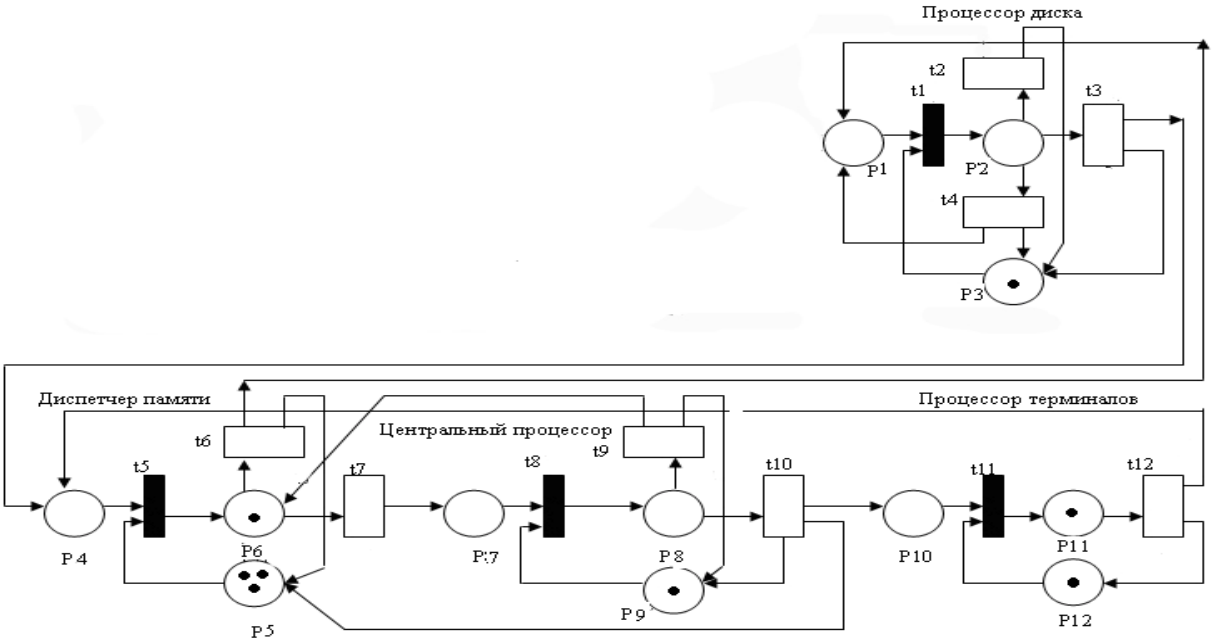


Структура исследуемой системы поддержки принятия решений группой операторов – лиц принимающих решения (ЛПР) для данного варианта



Формальная структура системы в виде системы массового обслуживания потоков заданий ЛПР и их файлов-программ для данного варианта

Изобразим согласно заданию структуру Petri-nets модели для системы аналитической обработки:



А теперь для описания работы модели на основе сетей Петри зададим две основных таблицы (табл. А.1 и табл. А.2) назначения фишек и интерпретации событий:

Таблица А.1. Назначение фишек в маркировке позиций сети Петри

Позиция сети	Назначение фишек в маркировке сети	Начальное значение
P1	Файлы в очереди на обслуживание процессором диска (ПД)	0
P2	Процессор диска обслуживает файлы-задания	0
P3	Процессор диска свободен	1
P4	Файлы в очереди на обслуживание диспетчером памяти (ДП)	0
P5	Файлы в разделах оперативной памяти	d = 3
P6	Свободные разделы оперативной памяти	R – d = 1
P7	Файлы-задания в очереди на обслуживание центральным процессором	0
P8	Файлы-задания обслуживаются центральным процессором	0
P9	Центральный процессор свободен	1
P10	Файлы-результаты в очереди на обслуживание терминальным процессором (ТП), для выдачи на терминалы	0
P11	Файлы-результаты анализируются операторами-лицами, принимающими решения (ЛПР)	e=1
P12	Свободные терминалы операторов	v-e=1

Таблица А.2. Интерпретации событий и времена задержек между событиями: запуск-срабатывание переходов

Переходы сети	**Содержание событий запуск-срабатывание перехода**	**Времена задержек**
T1	Запуск-срабатывание – файл поступает из очереди на обслуживание процессором диска	0
T2 R2=0.3	Запуск-начало обслуживания файла процессором диска срабатывание – конец перезаписи с диска в буфер вывода	Случайное 1/λ2=25 мс
T3 R3=0.6	Запуск-начало обслуживания файла процессором диска срабатывание – конец перезаписи с диска в очередь ДП	Случайное 1/λ2=25 мс
T4 R4=0.1	Запуск-начало обслуживания файла процессором диска срабатывание – конец перезаписи с диска на диск	Случайное 1/λ2=25 мс
T5	Запуск-срабатывание – файл поступает из очереди на обслуживание диспетчером оперативной памяти	0
T6 R6=0.1	Запуск-начало обслуживания файла диспетчером памяти срабатывание – конец записи файла из раздела на диск	Случайное 1/λ3=0.1 мс
T7 R7=0.9	Запуск-начало обслуживания файла процессором диска срабатывание – конец записи файла в очередь к ЦП	Случайное 1/λ3=0.1 мс
T8	Запуск-срабатывание – файл поступает из очереди на обслуживания центральным процессором	0
T9 R8=0.8	Запуск – начало обслуживания файла ЦП Срабатывание – конец записи результатов в раздел ОП	Случайное 1/λ4=0.02 мс
T10 R9=0.2	Запуск – начало обслуживания файла ЦП Срабатывание – конец передачи результатов в очередь ТП	Случайное 1/λ4=0.02 мс
T11	Запуск-срабатывание – файл поступает из очереди на обслуживания терминальным процессором	0
T12	Запуск – начало обслуживания файла ТП и оператором Срабатывание – конец принятия решения и передача файла в очередь для записи в раздел ОП	Случайное 1/λ5=0,5 мс

Вектор начальных значений: M^0 = (0, 0, 1, 0, 3, 1, 0, 0, 1, 0, 1, 1)

Матрица входов

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
1	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	1	0

Матрица выходов

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1

Результирующая матрица

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
-1	1	-1	0	0	0	0	0	0	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0
0	-1	1	1	0	0	0	0	0	0	0	0
1	-1	1	0	0	0	0	0	0	0	0	0
0	0	0	-1	1	-1	0	0	0	0	0	0
1	0	0	0	-1	1	0	0	0	0	0	0
0	0	0	0	-1	0	1	0	0	0	0	0
0	0	0	0	0	0	-1	1	-1	0	0	0
0	0	0	0	1	0	0	-1	1	0	0	0
0	0	0	0	0	1	0	-1	1	1	0	0
0	0	0	0	0	0	0	0	0	-1	1	-1
0	0	0	1	0	0	0	0	0	0	-1	1

III.Построение графического редактора.

На данном этапе проектирования было разработано программное обеспечение, позволяющие моделировать стохастические сети Петри. Особенностью данного продукта является то, что исходными данными является сама модель стохастической сети Петри, а не матрицы входов и выходов.

Для рисования фигур был разработан абстрактный класс FigureNet, который предоставляет абстрактные методы (edit – задание параметров фигуры, isIn – проверка была ли выделена фигура мышкой, getIntersection – получения координат фигуры)для работы с разными фигурами, а также набор статистических методом(getName – получения имени фигуры, setName – установка имени фигуры, show- показать фигуру, hide – скрыть фигуру, moveTo – передвинуть фигуру).

Для рисования вершин графов используется класс PositionNet, который содержит в себе метод setCountFishek, чтоб установить количество фишек.

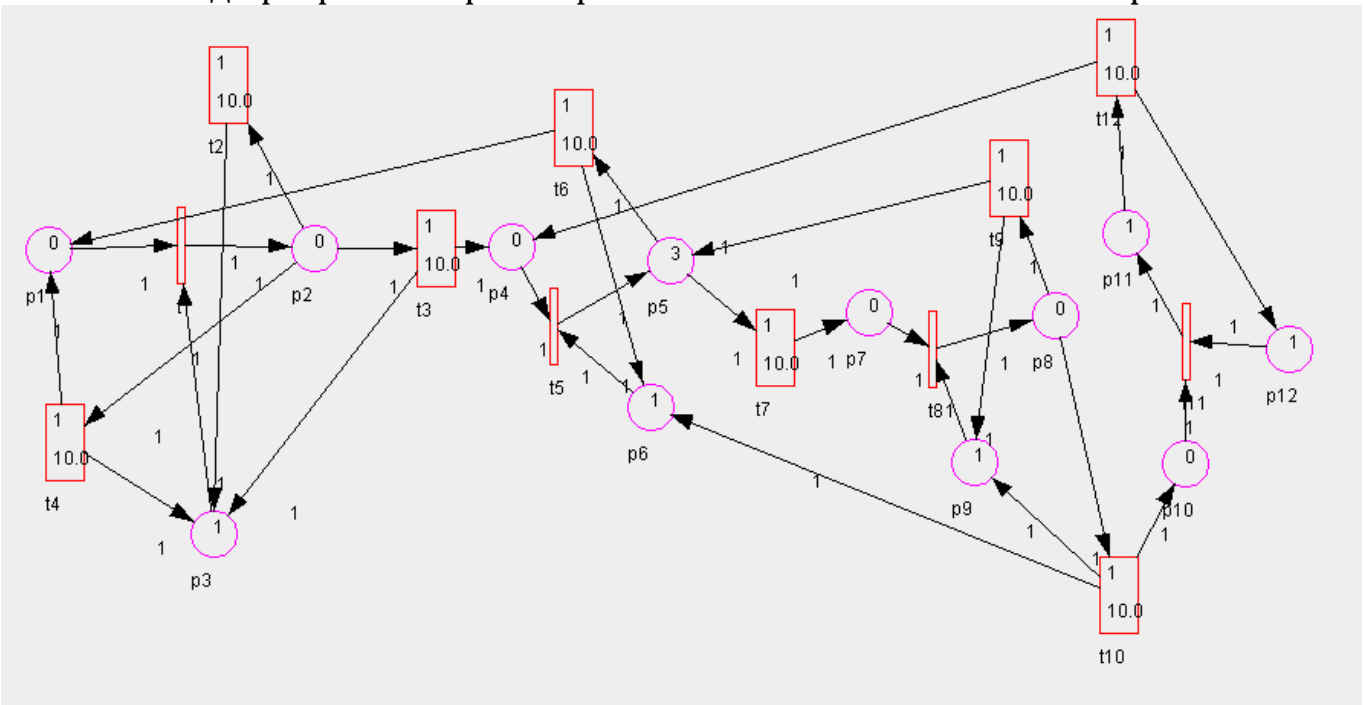
Для рисования временных вершин используется класс TimeTransitionNet, в котором есть методы которые позволяют установить коэффициент ковариации и интенсивность потока.

Для рисования мгновенных вершин используется класс TransitionNet. И для рисования линий мы используем класс LineNet, который содержит метод setCount для установки количетва линий между двумя обьектами.

Краткий алгоритм работы с программой:

В начале пользователь может добавить на рабочую область элементы – «состояния» и элементы «переходы». После этого надо установить связи между элементами. Третьим шагом является задание начальной маркировки стохастической сети Петри. После этого надо задать параметры переходов и описать конфликтные ситуации, если таковые имеются. После этого программа готова к моделированию.

Внешний вид программы при набранной стохастической сети Петри:



После компиляции граф преобразуется в ряд таблиц:

Переходы сети

Табличный редактор сетей Петри.

Переходы сети | Ключи распределения | Результирующие функции переходов | Маркировка сети | Функции входов переходов | Функции выходов переходов

Имя	Временной	Коэфф. ковари...	Интенсивность
t1	false		
t2	true	1	10.0
t3	true	1	10.0
t4	true	1	10.0
t5	false		
t6	true	1	10.0
t7	true	1	10.0
t8	false		
t9	true	1	10.0
t10	true	1	10.0
t11	false		
t12	true	1	10.0

Функции входов переходов

Табличный редактор сетей Петри.

Переходы сети | Ключи распределения | Результирующие функции переходов | Маркировка сети | Функции входов переходов | Функции выходов переходов

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
1	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	0	0	0	0	1	0

Функции выходов переходов

Табличный редактор сетей Петри.

Переходы сети | Ключи распределения | Результирующие функции переходов | Маркировка сети | Функции входов переходов | Функции выходов переходов

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	1

Результирующие функции переходов

Табличный редактор сетей Петри.

Переходы сети | Ключи распределения | Результирующие функции переходов | Маркировка сети | Функции входов переходов | Функции выходов переходов

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
-1	1	-1	0	0	0	0	0	0	0	0	0
0	-1	1	0	0	0	0	0	0	0	0	0
0	-1	1	1	0	0	0	0	0	0	0	0
1	-1	1	0	0	0	0	0	0	0	0	0
0	0	0	-1	1	-1	0	0	0	0	0	0
1	0	0	0	-1	1	0	0	0	0	0	0
0	0	0	0	-1	0	1	0	0	0	0	0
0	0	0	0	0	0	-1	1	-1	0	0	0
0	0	0	0	1	0	0	-1	1	0	0	0
0	0	0	0	0	1	0	-1	1	1	0	0
0	0	0	0	0	0	0	0	0	-1	1	-1
0	0	0	1	0	0	0	0	0	0	-1	1

Маркировка сети

Табличный редактор сетей Петри.

Переходы сети | Ключи распределения | Результирующие функции переходов | Маркировка сети | Функции входов переходов | Функции выходов переходов

Имя	Количество фишек
p1	0
p2	0
p3	1
p4	0
p5	3
p6	1
p7	0
p8	0
p9	1
p10	0
p11	1
p12	1

IV. Построение марковского графа и дерева достижимости.

Дерево достижимости ССП – это граф, вершинами которого являются реальные состояния (маркировки) сети, которые могут быть достигнуты из каждого очередного реального состояния последовательными и независимыми запусками всех разрешенных переходов в соответствующей граничной маркировке M_x, начиная с начальной M_0. Запуски срабатывания мгновенных переходов, если они разрешены, отмечаются, как единое событие вмести с запуском-срабатыванием предыдущего временного перехода.

В целом ряде моделей взаимодействующих процессов дерево достижимости оказывается бесконечным. Анализ процедуры построение дерева позволяет выявить несколько факторов, позволяющих сократить (свернуть) дерево достижимости сети Петри.

Во-первых: появляются вершины, в которых нет разрешенных переходов. Это так называемые пассивные M_T, а соответствующие вершины называются терминальными или конечными.

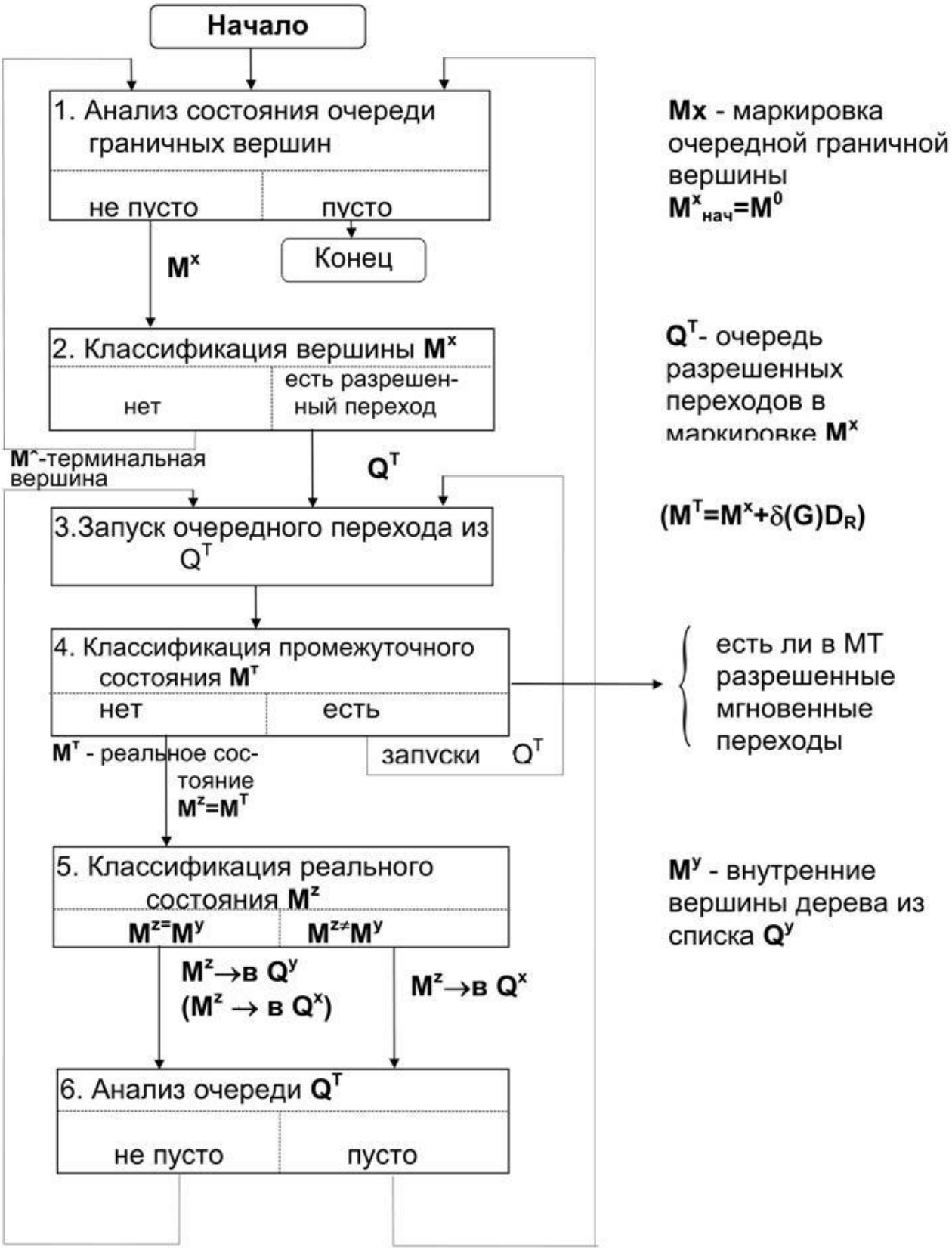
Во-вторых: некоторые формируемые вершины соответствуют маркировкам M_z, полностью совпадающим (эквивалентным) уже имеющимся маркировкам M_y внутренних вершин, т.е. $M_z=M_y$. Такие вершины M_z называются дублирующими по отношению к внутренним вершинам.

В-третьих: получаемые новые вершины могут иметь маркировки M_z, обладающие свойством покрываемости по отношению к внутренним вершинам M_y, которое определяется как $M_z > M_y$. Это соответствует случаю неограниченного “размножения” фишек в одной или в нескольких позициях сети. Тогда вводиться специальный параметр w (например, 99) как элемент вектора маркировки для тех позиций P_i, для которых $m_z(p_i)>m_y(p_i)$. Тогда $m_z(p_i) = w$. При последующих запусках-срабатываниях к w не прибавляются фишки и не вычитаются, полагая, что w =бесконечности.

Алгоритм построения дерева достижимости ССП включает в себя классификатор вершин, который переводит все исследуемые вершины M_z в четыре вида:

1. Граничные
2. Внутренние
3. Дублирующие
4. Терминальные

Алгоритм построения дерева достижимости



Построим д для данной ССП:

Частота переходов между состояниями | Дерево достижимости

Состояния сети | Переходы между состояниями

- M0 (0, 0, 1, 0, 3, 1, 0, 0, 1, 0, 1, 1) inner
  - T0(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1) M1(0, 0, 1, 0, 4, 0, 0, 0, 1, 0, 0, 2) inner
    - T312(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0) M52(0, 0, 1, 0, 3, 0, 0, 1, 0, 0, 0, 2) inner
      - T183(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0) M0(0, 0, 1, 0, 3, 1, 0, 0, 1, 0, 1, 1) inner
      - T184(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0) M1(0, 0, 1, 0, 4, 0, 0, 0, 1, 0, 0, 2) inner
      - T185(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0) M53(0, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0, 2) inner
        - T187(0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0) M2(0, 0, 1, 0, 2, 1, 0, 1, 0, 0, 1, 1) inner
          - T307(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1) M52(0, 0, 1, 0, 3, 0, 0, 1, 0, 0, 0, 2) inner
          - T308(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0) M64(0, 0, 1, 0, 2, 2, 0, 0, 1, 0, 2, 0) inner
            - T230(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1) M0(0, 0, 1, 0, 3, 1, 0, 0, 1, 0, 1, 1) inner
            - T231(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0) M62(0, 0, 1, 0, 1, 2, 0, 1, 0, 0, 2, 0) inner
              - T216(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1) M2(0, 0, 1, 0, 2, 1, 0, 1, 0, 0, 1, 1) inner
              - T217(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0) M63(0, 0, 1, 0, 1, 3, 0, 0, 1, 1, 2, 0) inner
                - T233(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1) M64(0, 0, 1, 0, 2, 2, 0, 0, 1, 0, 2, 0) inner
                - T234(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1) M64(0, 0, 1, 0, 2, 2, 0, 0, 1, 0, 2, 0) inner
                - T235(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0) M66(0, 0, 1, 0, 0, 3, 0, 1, 0, 1, 2, 0) inner
                  - T224(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1) M62(0, 0, 1, 0, 1, 2, 0, 1, 0, 0, 2, 0) inner
                  - T225(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1) M62(0, 0, 1, 0, 1, 2, 0, 1, 0, 0, 2, 0) inner
                  - T226(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0) M67(0, 0, 1, 0, 0, 4, 0, 0, 1, 2, 2, 0) inner
                  - T227(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0) M63(0, 0, 1, 0, 1, 3, 0, 0, 1, 1, 2, 0) inner
                - T236(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M60(0, 1, 0, 0, 0, 4, 0, 0, 1, 1, 2, 0) inner
              - T218(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0) M64(0, 0, 1, 0, 2, 2, 0, 0, 1, 0, 2, 0) inner
              - T219(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0) M65(0, 0, 1, 0, 0, 2, 1, 1, 0, 0, 2, 0) inner
              - T220(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M59(0, 1, 0, 0, 0, 3, 0, 1, 0, 0, 2, 0) inner
            - T232(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M61(0, 1, 0, 0, 1, 3, 0, 0, 1, 0, 2, 0) inner
          - T309(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0) M0(0, 0, 1, 0, 3, 1, 0, 0, 1, 0, 1, 1) inner
          - T310(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0) M55(0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1) inner
          - T311(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M5(0, 1, 0, 0, 1, 2, 0, 1, 0, 0, 1, 1) inner
        - T188(0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0) M2(0, 0, 1, 0, 2, 1, 0, 1, 0, 0, 1, 1) inner
        - T189(0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0) M52(0, 0, 1, 0, 3, 0, 0, 1, 0, 0, 0, 2) inner
        - T190(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0) M54(0, 0, 1, 0, 1, 0, 2, 1, 0, 0, 0, 2) inner
        - T191(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M51(0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 2) inner
      - T186(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M50(0, 1, 0, 0, 2, 1, 0, 1, 0, 0, 0, 2) inner
    - T313(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M4(0, 1, 0, 0, 3, 1, 0, 0, 1, 0, 0, 2) inner
  - T1(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0) M2(0, 0, 1, 0, 2, 1, 0, 1, 0, 0, 1, 1) inner
  - T2(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0) M3(0, 1, 0, 0, 2, 2, 0, 0, 1, 0, 1, 1) inner

Из дерева достижимости с помощью «склеивания» одноименных вершин построим Марковский граф.
Таблица вершин графа, которая включает в себя поля:

- Имя вершины
- Маркировка вершины
- Тип вершины
- Количество попаданий в вершину
- Суммарное время пребывания в маркировке

$$T_s = \sum_{i=1}^{n} (T - T_0)$$

где n – количество попаданий в данную маркировку
T – время ухода из данной маркировки
T0 – время прихода в данную маркировку

- Суммарное время возвращения в маркировке

$$T_v = \sum_{i=1}^{n} (T_0 - T_l)$$

где n – количество попаданий в данную маркировку
T0 – время прихода в данную маркировку
Tl – последнее время ухода из маркировки

- Последнее время ухода из маркировки

Таблица марковского графа

Частота переходов между состояниями | Дерево достижимости

Состояния сети | Переходы между состояниями

Имя	Код	Нотация	Количе...	Суммарное время пр...	Суммарное время во...	Последнее время ухо...
M0	(0, 1, 0, 0, 2, 0, 0, ...	inner	1	0.008908287	0.0	0.008908287
M1	(0, 1, 0, 0, 1, 0, 0, ...	inner	1	0.020708019	0.008908287	0.029616306
M2	(1, 1, 0, 0, 1, 1, 0, ...	inner	1	0.108756594	0.08253484	0.19129144
M3	(0, 0, 1, 1, 2, 0, 0, ...	inner	0	0.0	0.0	0.0
M4	(0, 0, 1, 0, 2, 0, 0, ...	inner	3	0.38382685	0.9616407	1.3454676
M5	(0, 0, 1, 0, 1, 0, 0, ...	inner	2	0.028126836	1.2400867	1.2682135
M6	(0, 1, 0, 0, 1, 1, 0, ...	inner	7	0.21678138	1.150569	1.3673501
M7	(0, 1, 0, 0, 0, 1, 0, ...	inner	3	0.057032347	0.849972	0.90700436
M8	(1, 1, 0, 0, 0, 2, 0, ...	inner	0	0.49029124	0.3224237	0.81271505
M9	(0, 0, 1, 0, 1, 1, 0, ...	inner	2	0.20228684	1.3722343	1.5745213
M10	(0, 0, 1, 0, 0, 1, 0, ...	inner	4	0.004884362	1.5097065	1.5145909
M11	(0, 1, 0, 0, 0, 2, 0, ...	inner	1	0.025387883	1.5745213	1.5999092
M12	(0, 0, 1, 0, 0, 2, 0, ...	terminal	0	0.0	0.0	0.0
M13	(0, 0, 1, 0, 0, 0, 1, ...	inner	0	0.0	0.0	0.0
M14	(0, 1, 0, 0, 0, 0, 1, ...	inner	0	0.0	0.0	0.0
M15	(1, 1, 0, 0, 0, 1, 0, ...	covering	2	0.05291854	0.029616304	0.08253484
M16	(0, 0, 1, 1, 1, 0, 0, ...	covering	0	0.0	0.0	0.0
M17	(0, 0, 1, 1, 0, 0, 1, ...	covering	0	0.0	0.0	0.0

Переходы между состояниями

Частота переходов между состояниями | Дерево достижимости

Состояния сети | Переходы между состояниями

Имя	Из	В	Цепочка запусков переходов
T0	M0	M1	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T1	M0	M2	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T2	M0	M3	(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
T3	M3	M4	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T4	M3	M5	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T5	M3	M6	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
T6	M3	M3	(1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
T7	M3	M0	(0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0)
T8	M3	M7	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
T9	M7	M8	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T10	M7	M9	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T11	M7	M10	(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
T12	M10	M11	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T13	M10	M12	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T14	M10	M13	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
T15	M10	M10	(1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
T16	M10	M7	(0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0)
T17	M10	M14	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
T18	M14	M15	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T19	M14	M16	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T20	M14	M17	(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
T21	M17	M18	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T22	M17	M17	(1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
T23	M17	M14	(0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0)
T24	M17	M19	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
T25	M19	M20	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
T26	M20	M21	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T27	M20	M22	(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)

V. Моделирование ССП.

В процессе моделирования реализуется основной режим работы имитационной модели – выполнение сетей Петри.

Особенностью режима выполнения ССП является то, что при обработке событий запуски временных переходов планируются конкретные времена будущих событий срабатывания этих переходов. Это второй принцип управления имитационными моделями формальных дискретных систем.

Для планирования интервалов времени между событиями «срабатывания переходов» (обслуживания заявок) используются имитационные модели информационных потоков обслуживания с конкретными заданиями по вариантам характеристиками интенсивности потока и коэффициента вариации. Асинхронное развитие процессов реализуется на основе принципа самодостаточности: если переход разрешен, то он запускается и одновременно планируется время его срабатывания.

Исключение составляют конфликтные переходы. Для разрешения конфликтов в ССП могут использоваться различные средства, например, приоритеты переходов, семафоры и т.п. Однако наиболее удобным механизмом разрешения конфликтов для ССП являются ключи распределения – R_i, которые становятся в соответствие конфликтующим переходам и отражают вероятности срабатывания каждого из них в конфликте. В режиме моделирования процессов в случае конфликта производится розыгрыш с учетом вероятностей R_i и запускается только один переход.

Наша программа работает по тактам, значить время работы программы зависит от продолжительности такта. Время работы такта генерируется генератором случайных чисел с нормальным распределением.

Запустим программу в работу и получим следующие результаты:

Управляющий список переходов

- переход
- время запуска
-время срабатывания
-запущен в данный момент
-количество запусков
-суммарное время работы перехода

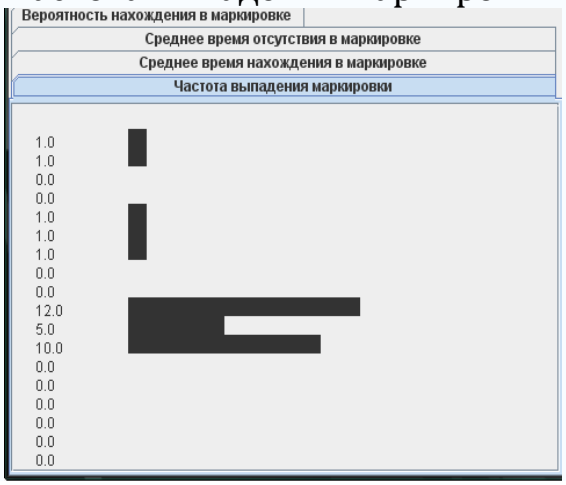
$$T_{sp} = \sum_{i=1}^{n} T_{zp} - T_{vsp}$$

где Tzp – время запуска перехода
Tvsp – время срабатывания перехода
n – количество запусков перехода

Управляющий список | Блокирование переходов

Переход	Время запуска	Время срабат...	Запущен	Количество за...	Суммарное вр...
t1	3.3066745	3.3066745	false	11	0.0
t2	3.3066745	3.5721085	false	3	0.31891966
t3	2.8445098	2.9775093	false	6	0.98268294
t4	2.8378284	2.8445098	false	3	0.32298017
t5	2.9775093	2.9775093	false	6	0.0
t6	2.9775093	3.3066745	false	8	0.66364
t7	1.9974207	2.0227854	false	6	0.75268614
t8	2.0227854	2.0227854	false	6	0.0
t9	2.0227854	2.1280346	false	6	0.7296021

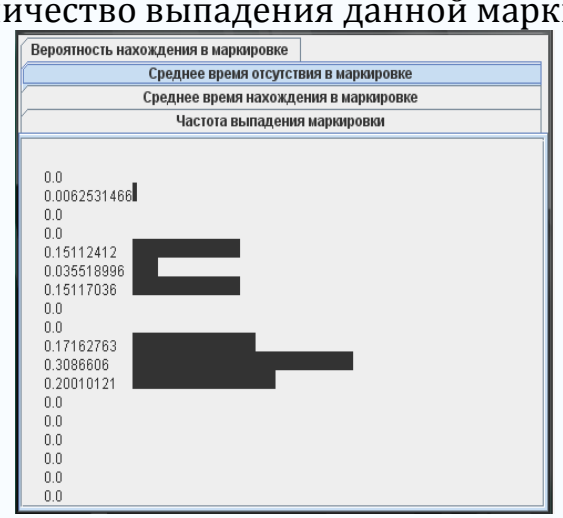
Частота выпадения маркировки



Среднее время отсутствия в маркировке

$$T_{os} = \frac{T_v}{k}$$

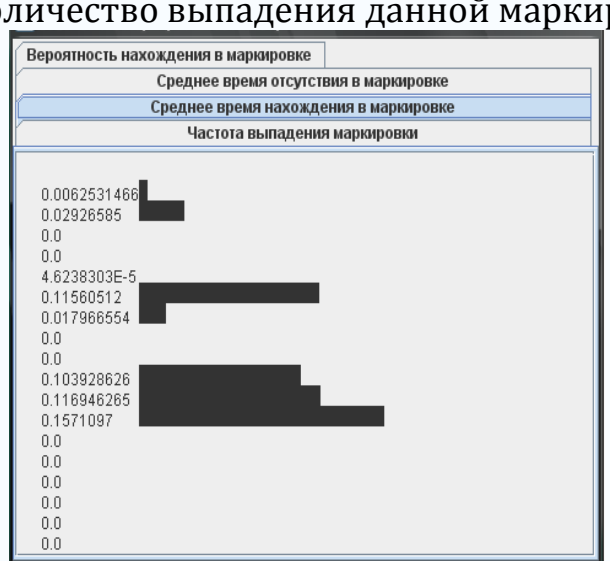
где Tv – суммарное время возвращения в маркировку
к – количество выпадения данной маркировки



Среднее время нахождения в маркировке

$$T_{vs} = \frac{T_s}{k}$$

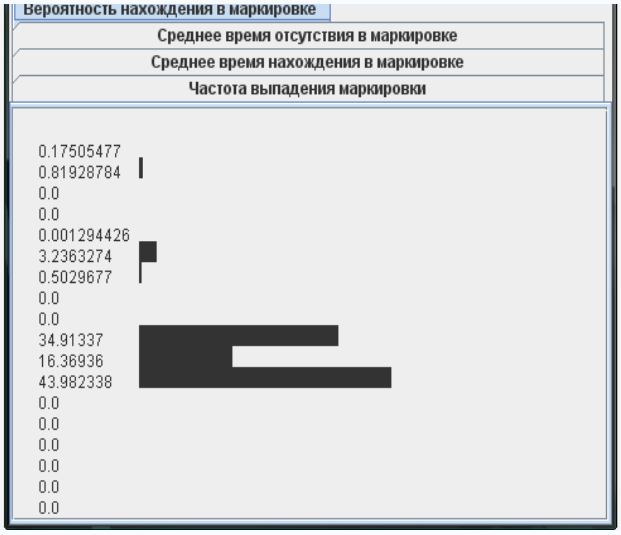
где Ts – суммарное время пребывания в маркировке
к – количество выпадения данной маркировки



Вероятность нахождения в маркировке

$$P = \frac{T_s}{T}$$

где Ts – суммарное время пребывания в данной маркировке
T – общее время работы системы



Также в результате моделирования получаем следующие характеристики:
- Среднее время простоя перехода

$$T_{pp} = \frac{T - T_{sp}}{k}$$

где T – время работы системы
Tsp – суммарное время работы перехода
к – количество запуска перехода

- Среднее время работы перехода

$$T_{rp} = \frac{T_{sp}}{k}$$

где к – количество запусков перехода
Tsp – суммарное время работы перехода

- Загрузка переходов

$$P_p = \frac{T_{sp}}{T}$$

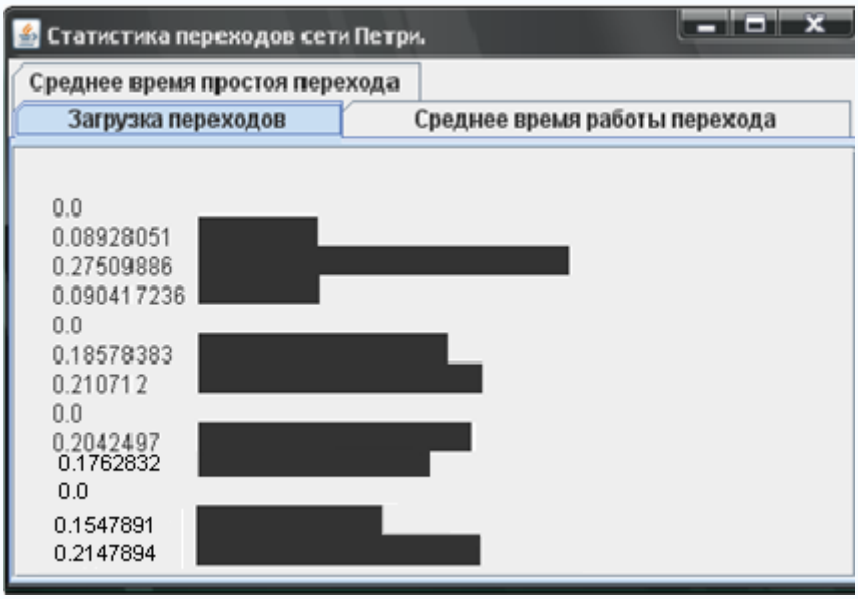
где T – время работы системы
Tsp – суммарное время работы перехода

Выводы.

1. Мы исследовали марковский граф и дерево достижимости данной ССП. Последняя не имеет терминальных вершин, поэтому вычислительная система, построенная на ее основе, может стабильно работать, без остановки, как угодно долго.

2. В ходе моделирования ССП были вычислены следующие характеристики:

   - Среднее время работы каждого прибора.
   - Среднее время простоя.
   - Процент загрузки оборудования.



Результатов моделирования видно, что все соответствующие временным переходам приборы в системе загружены на 15-30%. Это значит, что система с данной конфигурацией оборудования нерациональна. Для исправления этого недостатка необходимо изменить конфигурацию соединений в ССП, либо увеличить количество фишек, либо же изменить соотношения интенсивностей временных переходов.

VI. Моделирование лог-сервера и сервера приложений.

Структура ССП для моделирования лог-сервера и сервера приложений показана на рис. А. 6.

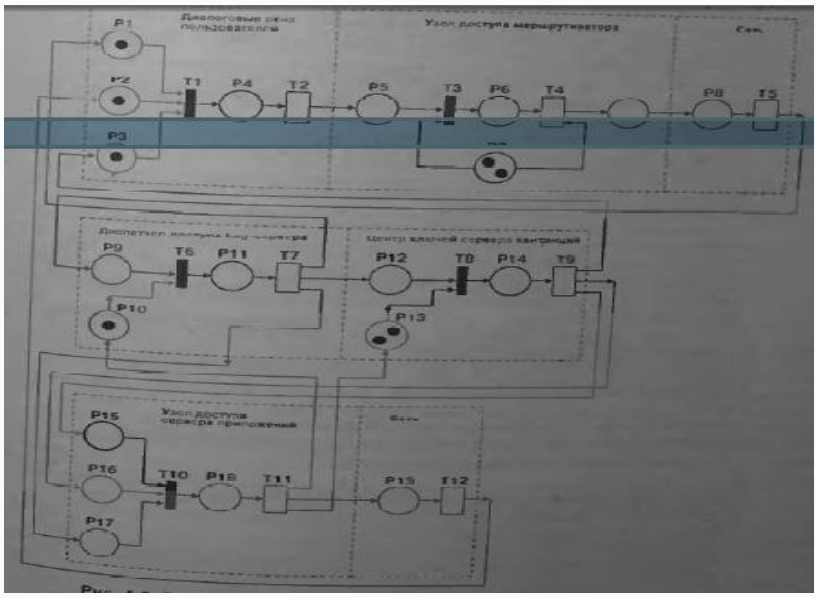


Рис. А. 6. ССП - модель удаленного взаимодействия пользователя с сервером приложений.

Таблица А.5. Назначение фишек в маркировке позиций в ССП – модели распределенной системы на базе лог-сервера и сервера приложений.

Позиция сети	**Назначение фишки в маркировке сети**	**Начальное значение**
P1	Логин пользователя, а в дальнейшем право доступа к серверу приложений в каждом сеансе.	1
P2	Запросы пользователя к серверу для информационно-аналитической поддержки сессии принятия решений	U=1
P3	Пароль пользователя, а в дальнейшем сеансовые ключи шифрования	1
P4	Обработка запроса и предоставление доступа к окну приложения, формирование запроса к серверу	0
P5	Очередь запросов к узлу доступа маршрутизатора сети	0
P6	Обработка запроса на передачу сообщений по сети	0
P7	Количество свободных путей передачи сообщений в сети	S=4
P8	Транспортировка по сети и сборка пакетов сообщений в процессе удаленного взаимодействия	0
P9	Запрос на предоставление прав доступа к серверу приложений	0
P10	Количество пользователей, которым могут быть предоставлены права доступа в конкретном сеансе	N=2
P11	Определение лог - сервером прав доступа по конкретному запросу	0
P12	Запрос на предоставление сеансового ключа шифрования для работы с сервером приложений	0
P13	Количество сеансовых ключей, которые могут быть предоставлены пользователям в конкретном сеансе	K=4
P14	Формирование сервером квитанций сеансового ключа в соответствии с правами доступа	0

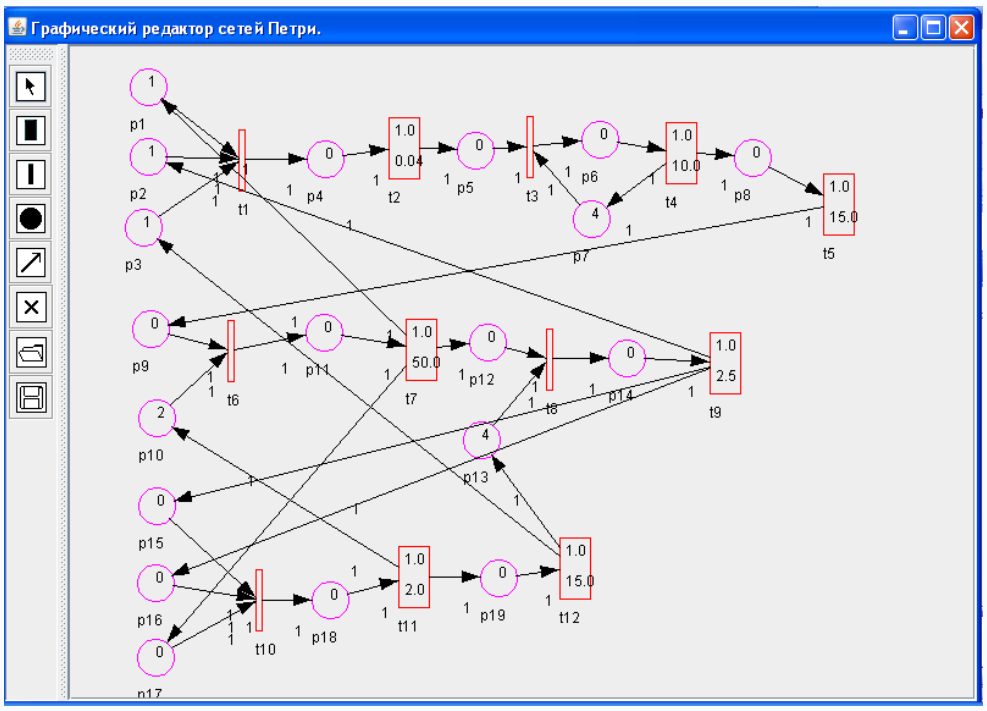
P15	Право доступа к серверу приложений	0
P16	Удаленный запрос пользователя к серверу приложений	0
P17	Сеансовый ключ шифрования сигнатуры ответа сервера приложений удаленному пользователю	0
P18	Обработка запроса, поиск и анализ соответствующих данных, формирование сигнатуры сообщения	0
P19	Транспортировка пакетов по сети и сборка сообщений в процессе удаленного взаимодействия	0

Таблица А.6. Назначение переходов в ССП – модели распределенной системы на базе лог – сервера и сервера приложений.

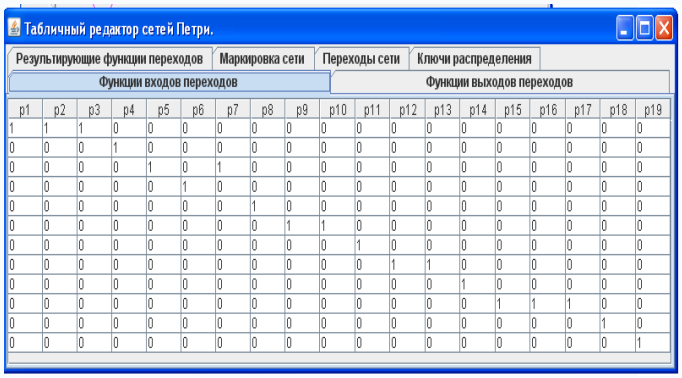
Переход в сети	**События запуск и срабатывание перехода**	**Время задержки**
T1	Запуск/срабатывание – открытие для конкретного пользователя диалогового окна приложения	0
T2	Запуск – начало работы с окном приложения Срабатывание – формирование запроса в сеть	$\lambda 1=0.04$
T3	Запуск/срабатывание – прием запроса на передачу его по свободному маршруту	0
T4	Запуск – начало работы по выбору и прокладке маршрута Срабатывание – передача запроса в сеть	$\lambda 2=10$
T5	Запуск – начало передачи запроса в сеть Срабатывание – завершение передачи запроса в сеть	$\lambda 3=15$
T6	Запуск/срабатывание – прием запроса для анализа прав доступа к серверу приложений	0
T7	Запуск – начало анализа Срабатывание – рассылка конкретных прав доступа пользователя в сеансе: 1 – в окно приложения пользователя, 2 – передача запроса для формирования ключа шифрования, 3 – на сервер	$\lambda 4=50$
T8	Запуск/срабатывание – прием запроса для формирования сеансового ключа шифрования	0
T9	Запуск – начало расчета ключа Срабатывание – рассылка конкретного сеансового ключа: 1 – пользователю для декодирования информации, 2 – передача запроса серверу приложений (ключ зашифрован паролем), 3 – серверу приложений.	$Tk=0.4$
T10	Запуск/срабатывание – прием запроса для поиска информации в БД и последующей обработки ее сервером приложений	0
T11	Запуск – начало работы информационно-аналитического приложения Срабатывание – передача пользователю сообщения – ответа на запрос и сигнатуры сообщения, а также возврат прав доступа и ключа системе защиты	$\mu_{пр}=2$
T12	Запуск – начало передачи запроса в сеть Срабатывание – завершение передачи запроса в сеть	$\lambda 3=15$

Моделирование.

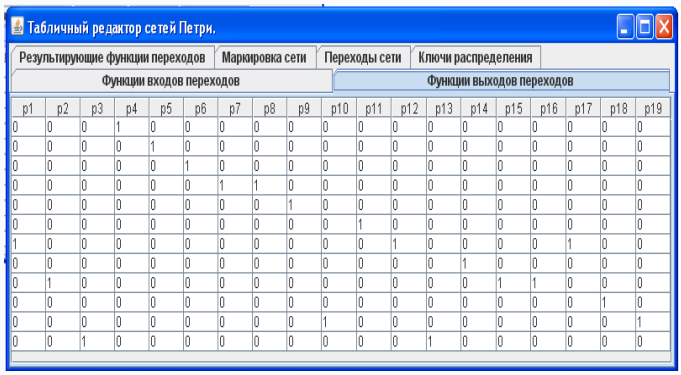
ССП в графическом виде:



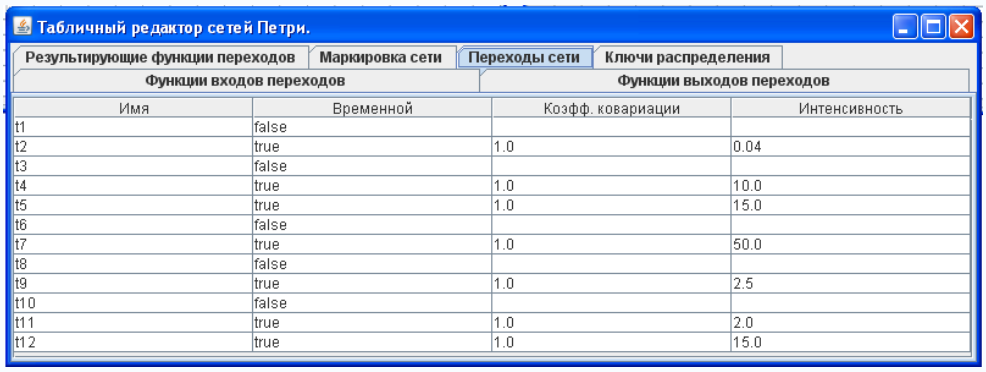
Откомпилируем в табличное представление:



p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

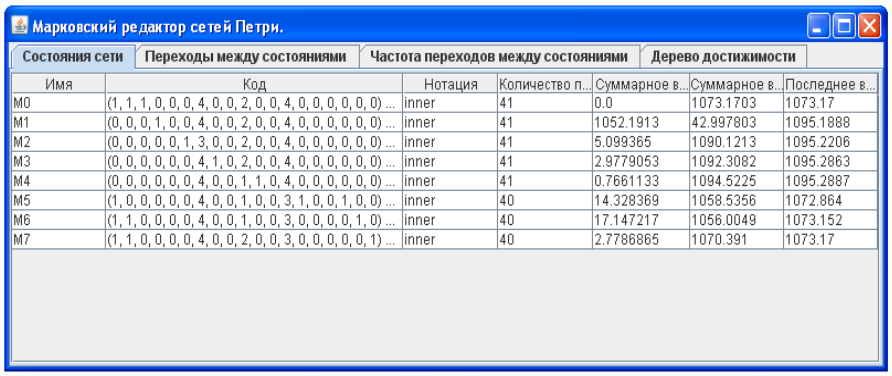


p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0



Имя	Временной	Коэфф. ковариации	Интенсивность
t1	false		
t2	true	1.0	0.04
t3	false		
t4	true	1.0	10.0
t5	true	1.0	15.0
t6	false		
t7	true	1.0	50.0
t8	false		
t9	true	1.0	2.5
t10	false		
t11	true	1.0	2.0
t12	true	1.0	15.0

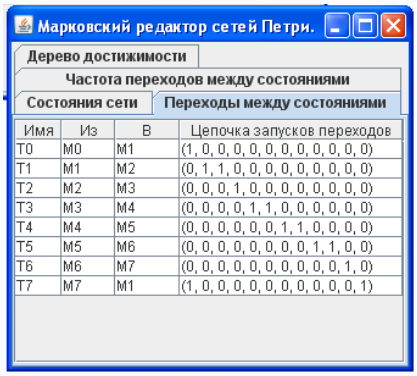
Получим марковский граф:



Марковский редактор сетей Петри.

Состояния сети | Переходы между состояниями | Частота переходов между состояниями | Дерево достижимости

Имя	Код	Нотация	Количество п...	Суммарное в...	Суммарное в...	Последнее в...
M0	(1, 1, 1, 0, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	0.0	1073.1703	1073.17
M1	(0, 0, 0, 1, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	1052.1913	42.997803	1095.1888
M2	(0, 0, 0, 0, 0, 1, 3, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	5.099365	1090.1213	1095.2206
M3	(0, 0, 0, 0, 0, 0, 4, 1, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	2.9779053	1092.3082	1095.2863
M4	(0, 0, 0, 0, 0, 0, 4, 0, 0, 1, 1, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	0.7661133	1094.5225	1095.2887
M5	(1, 0, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 1, 0, 0, 1, 0, 0) ...	inner	40	14.328369	1058.5356	1072.864
M6	(1, 1, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 1, 0) ...	inner	40	17.147217	1056.0049	1073.152
M7	(1, 1, 0, 0, 0, 0, 4, 0, 0, 2, 0, 0, 3, 0, 0, 0, 0, 0, 1) ...	inner	40	2.7786865	1070.391	1073.17

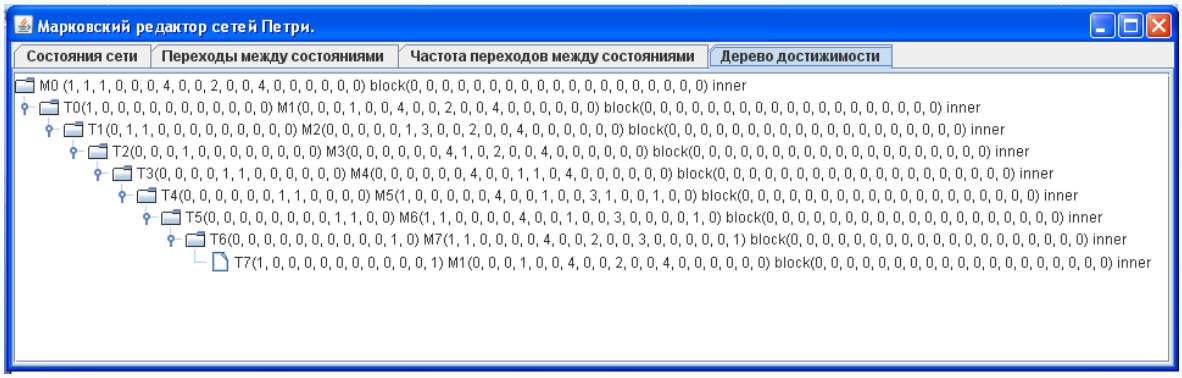


Марковский редактор сетей Петри.

Дерево достижимости | Частота переходов между состояниями | Состояния сети | Переходы между состояниями

Имя	Из	В	Цепочка запусков переходов
T0	M0	M1	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
T1	M1	M2	(0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
T2	M2	M3	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
T3	M3	M4	(0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0)
T4	M4	M5	(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
T5	M5	M6	(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0)
T6	M6	M7	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)
T7	M7	M1	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)

Дерево достижимости:



Марковский редактор сетей Петри.

Состояния сети | Переходы между состояниями | Частота переходов между состояниями | Дерево достижимости

M0 (1, 1, 1, 0, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T0(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) M1(0, 0, 0, 1, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T1(0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0) M2(0, 0, 0, 0, 0, 1, 3, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T2(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0) M3(0, 0, 0, 0, 0, 0, 4, 1, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T3(0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0) M4(0, 0, 0, 0, 0, 0, 4, 0, 0, 1, 1, 0, 4, 0, 0, 0, 0, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

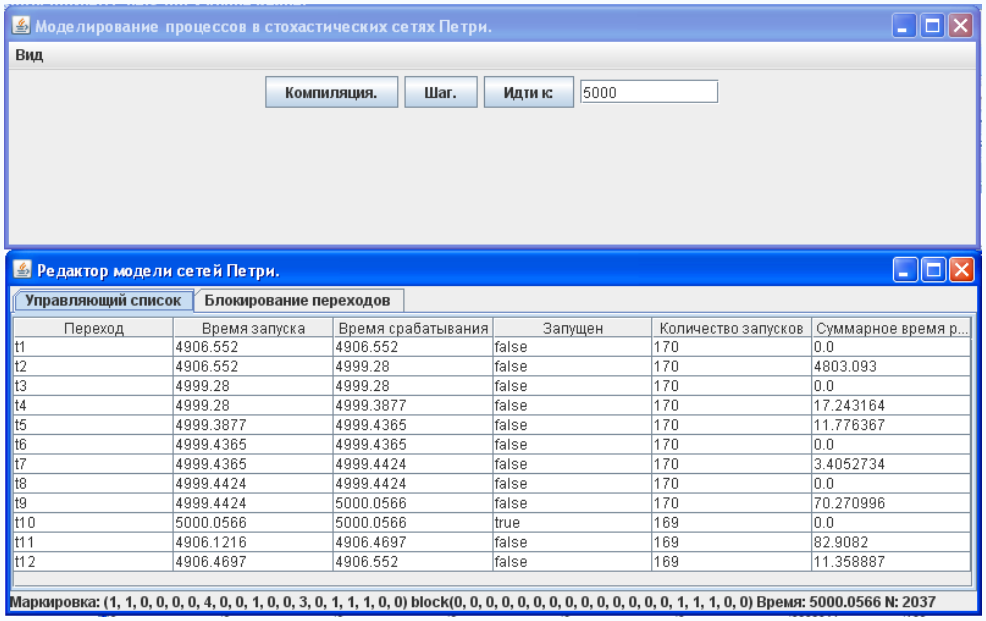
T4(0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0) M5(1, 0, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 1, 0, 0, 1, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T5(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0) M6(1, 1, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 1, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T6(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0) M7(1, 1, 0, 0, 0, 0, 4, 0, 0, 2, 0, 0, 3, 0, 0, 0, 0, 0, 1) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

T7(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1) M1(0, 0, 0, 1, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) inner

Запустим программу в работу.



Моделирование процессов в стохастических сетях Петри.

Вид

Компиляция. | Шаг. | Идти к: 5000

Редактор модели сетей Петри.

Управляющий список | Блокирование переходов

Переход	Время запуска	Время срабатывания	Запущен	Количество запусков	Суммарное время р...
t1	4906.552	4906.552	false	170	0.0
t2	4906.552	4999.28	false	170	4803.093
t3	4999.28	4999.28	false	170	0.0
t4	4999.28	4999.3877	false	170	17.243164
t5	4999.3877	4999.4365	false	170	11.776367
t6	4999.4365	4999.4365	false	170	0.0
t7	4999.4365	4999.4424	false	170	3.4052734
t8	4999.4424	4999.4424	false	170	0.0
t9	4999.4424	5000.0566	false	170	70.270996
t10	5000.0566	5000.0566	true	169	0.0
t11	4906.1216	4906.4697	false	169	82.9082
t12	4906.4697	4906.552	false	169	11.358887

Маркировка: (1, 1, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 0, 1, 1, 1, 0, 0) block(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0) Время: 5000.0566 N: 2037

Получим такие результаты:

Марковский редактор сетей Петри.

Состояния сети | Переходы между состояниями | Частота переходов между состояниями | Дерево достижимости

Имя	Код	Нотация	Количество п...	Суммарное в...	Суммарное в...	Последнее в...
M0	(1, 1, 1, 0, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	0.0	1073.1703	1073.17
M1	(0, 0, 0, 1, 0, 0, 4, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	1052.1913	42.997803	1095.1888
M2	(0, 0, 0, 0, 0, 1, 3, 0, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	5.099365	1090.1213	1095.2206
M3	(0, 0, 0, 0, 0, 0, 4, 1, 0, 2, 0, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	2.9779053	1092.3082	1095.2863
M4	(0, 0, 0, 0, 0, 0, 4, 0, 0, 1, 1, 0, 4, 0, 0, 0, 0, 0, 0) ...	inner	41	0.7661133	1094.5225	1095.2887
M5	(1, 0, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 1, 0, 0, 1, 0, 0) ...	inner	40	14.328369	1058.5356	1072.864
M6	(1, 1, 0, 0, 0, 0, 4, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 1, 0) ...	inner	40	17.147217	1056.0049	1073.152
M7	(1, 1, 0, 0, 0, 0, 4, 0, 0, 2, 0, 0, 3, 0, 0, 0, 0, 0, 1) ...	inner	40	2.7786865	1070.391	1073.17

Марковский редактор сетей Петри.

Состояния сети | Переходы между состояниями | Частота переходов между состояниями | Дерево достижимости

M0	M1	M2	M3	M4	M5	M6	M7
0	170	0	0	0	0	0	0
0	487101855	170	0	0	0	0	0
0	0	1855420	170	0	0	0	0
0	0	0	1191772	170	0	0	0
0	0	0	0	334472	170	0	0
0	0	0	0	0	6971184	169	0
0	0	0	0	0	0	8303944	169
169	0	0	0	0	0	0	1123515

Построим гистограммы вероятностей состояний и процента загрузки оборудования:

Статистика маркировок сети Петри.

Среднее время отсутствия в маркировке | Вероятность нахождения в маркировке

Частота выпадения маркировки | Среднее время нахождения в маркировке

0.0
96.06077
0.34485936
0.23552468
0.0681047
1.4054041
1.6581453
0.22717516

Статистика переходов сети Петри.

Загрузка переходов | Среднее время работы перехода | Среднее время простоя перехода

0.0
0.96060765
0.0
0.0034485937
0.0023552468
0.0
6.81047E-4
0.0
0.01405404
0.0
0.016581453
0.0022717516

Выводы.

1. Мы исследовали марковский граф и дерево достижимости данной ССП. Последняя не имеет терминальных вершин, поэтому вычислительная система, построенная на ее основе, может стабильно работать, без непредвиденного останова, как угодно долго.
2. Из листинга видно, что около 95% всего времени работает переход T1, в то время как остальные простаивают. Для того чтобы обеспечить равномерную загрузку всех временных переходов сети, необходимо подбирать приборы с равными интенсивностями выходных потоков. При большой нагрузке на сервер его пропускная способность определяется по самому медленному узлу в цепи.

VII. Моделирование сервера безопасности.

Структура ССП для моделирования сервера безопасности показана на рис. А.9.

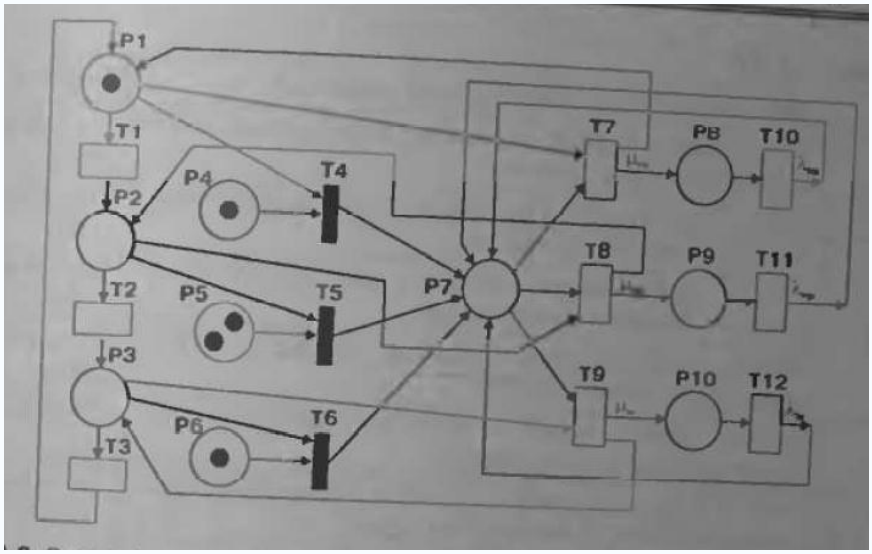


Рис. А.9. ССП – модель подсистемы аудита и исправления ошибок ПО на разных этапах его жизненного цикла.

Таблица А. 9. Назначение фишек в маркировке ССП.

Позиция сети	**Назначение фишки в маркировке сети**	**Начальное значение**
P1	Признак этапа проектирования ПО	1
P2	Признак этапа кодирования ПО	0
P3	Признак этапа модификации ПО	0
P4	Начальное количество ошибок в проекте	$N_{0\omega}=1$
P5	Дополнительные ошибки в исходном коде	$N_{0\omega}1=2$
P6	Дополнительные ошибки при модификации	$N_{0\omega}2=1$
P7	Динамика количества ошибок в ПО	0
P8	Количество исправленных ошибок в проекте	0
P9	Количество исправленных ошибок в программе	0
P10	Количество исправленных ошибок в модели	0

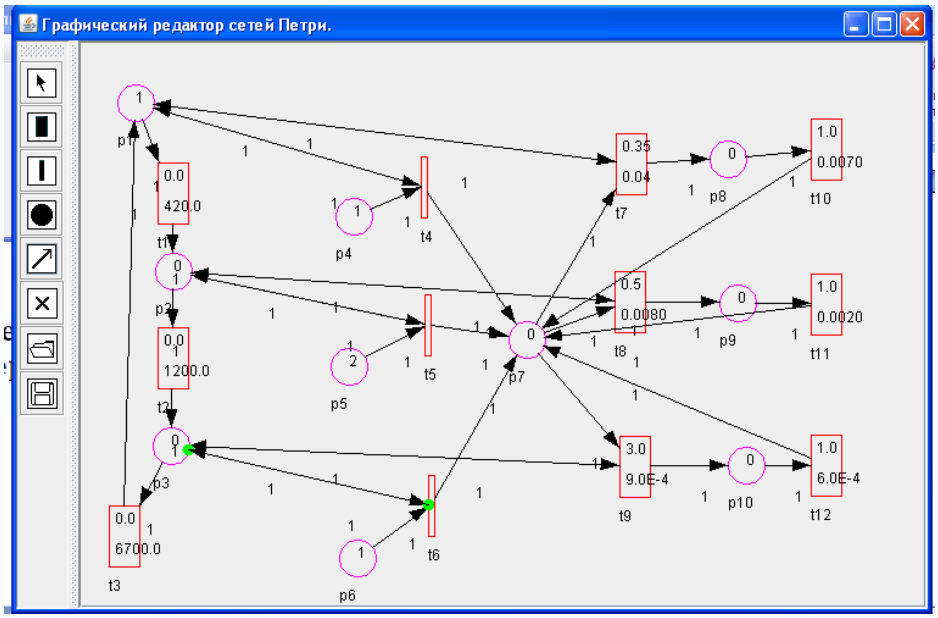
Таблица А.10. Интерпретация событий и времен задержек между событиями запуск – срабатывание переходов.

Переход в сети	**События запуск и срабатывание перехода**	**Времена задержки**
T1	Запуск – начало моделирования Срабатывание – конец проектирования	$T_{пк}=420$
T2	Запуск – начало кодирования Срабатывание – конец программирования	$T_{пр}=1200$
T3	Запуск – начало эксплуатации Срабатывание – конец модификации	$T_{м}=6700$
T4	Запуск/срабатывание – передача планируемой группы ошибок проекта	0
T5	Запуск/срабатывание – передача планируемой группы ошибок программы	0
T6	Запуск/срабатывание – передача планируемой группы ошибок модификации	0
T7	Запуск – начало исправления очередной ошибки проекта	$\mu_{пк}=0.04$

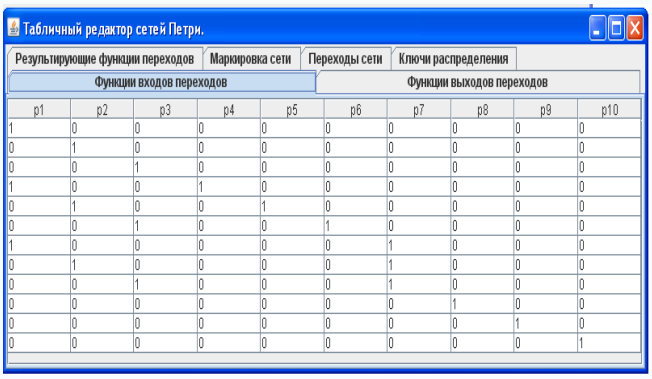
	Срабатывание – конец исправления очередной ошибки проекта	g=0.35
T8	Запуск – начало исправления очередной ошибки проекта Срабатывание – конец исправления очередной ошибки программ	$\mu_{пр}$=0.008 g=0.5
T9	Запуск – начало исправления очередной ошибки проекта Срабатывание – конец исправления очередной ошибки модификации	$\mu_{м}$=0.0009 g=3
T10	Запуск – начало внесения очередной ошибки в проект Срабатывание – конец внесения очередной ошибки в проект	$\lambda_{пк}$=0.007
T11	Запуск – начало внесения очередной ошибки в проект Срабатывание – конец внесения очередной ошибки в программу	$\lambda_{пр}$=0.002
T12	Запуск – начало внесения очередной ошибки в проект Срабатывание – конец внесения очередной ошибки при модификации	$\lambda_{м}$=0.0006

Моделирование.

Наберем данную модель в редакторе:



Откомпилируем сеть в таблицы:

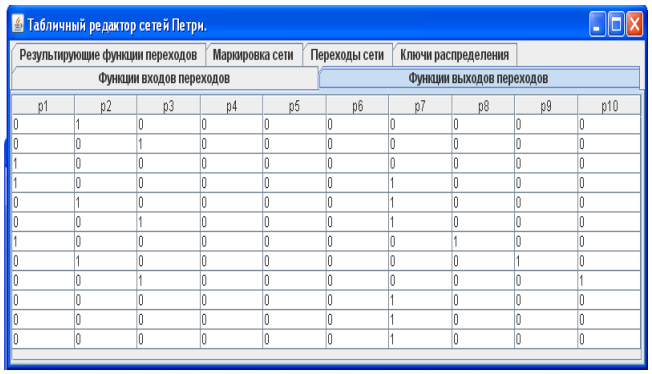


Табличный редактор сетей Петри.

Результирующие функции переходов | Маркировка сети | Переходы сети | Ключи распределения

Функции входов переходов | Функции выходов переходов

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
1	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

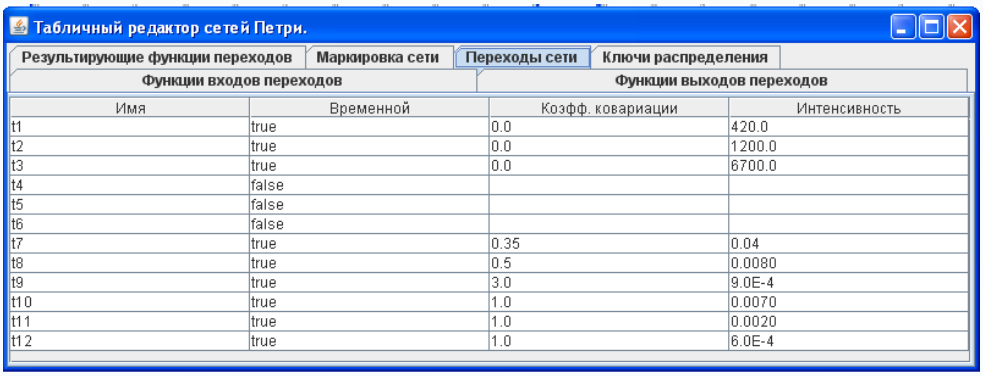


Табличный редактор сетей Петри.

Результирующие функции переходов | Маркировка сети | Переходы сети | Ключи распределения

Функции входов переходов | Функции выходов переходов

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0



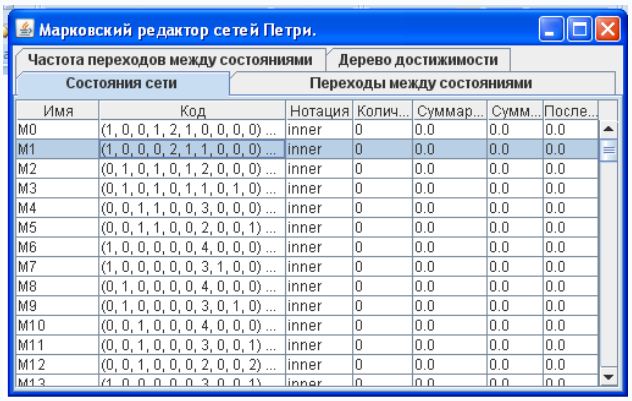
Табличный редактор сетей Петри.

Результирующие функции переходов | Маркировка сети | Переходы сети | Ключи распределения

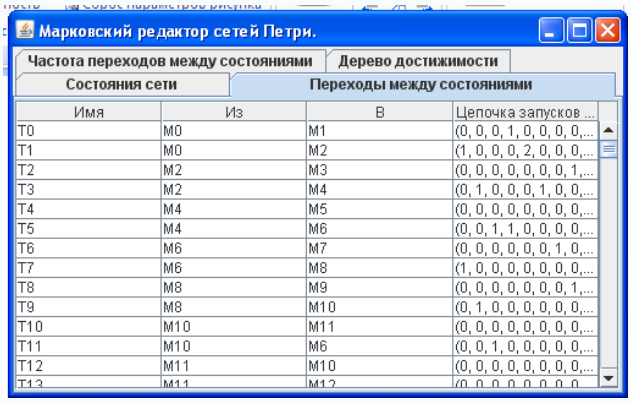
Функции входов переходов | Функции выходов переходов

Имя	Временной	Коэфф. ковариации	Интенсивность
t1	true	0.0	420.0
t2	true	0.0	1200.0
t3	true	0.0	6700.0
t4	false		
t5	false		
t6	false		
t7	true	0.35	0.04
t8	true	0.5	0.0080
t9	true	3.0	9.0E-4
t10	true	1.0	0.0070
t11	true	1.0	0.0020
t12	true	1.0	6.0E-4

Построим марковский граф:

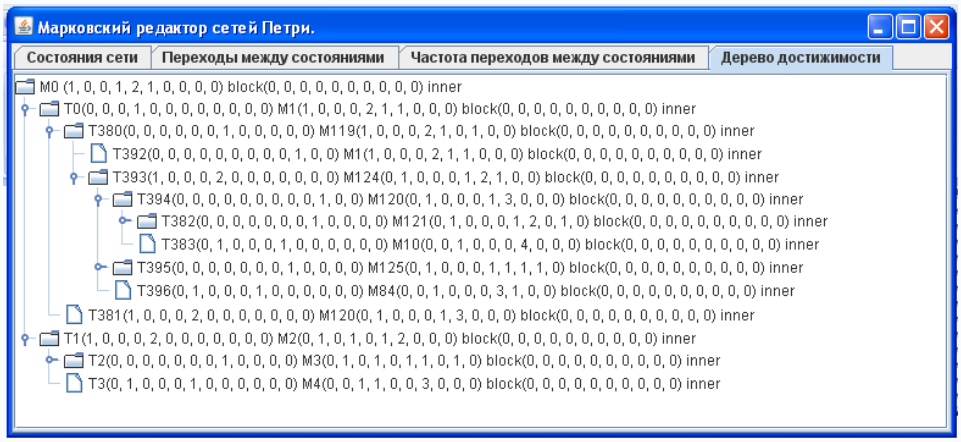


Имя	Код	Нотация	Колич...	Суммар...	Сумм...	После...
M0	(1, 0, 0, 1, 2, 1, 0, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M1	(1, 0, 0, 0, 2, 1, 1, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M2	(0, 1, 0, 1, 0, 1, 2, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M3	(0, 1, 0, 1, 0, 1, 1, 0, 1, 0) ...	inner	0	0.0	0.0	0.0
M4	(0, 0, 1, 1, 0, 0, 3, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M5	(0, 0, 1, 1, 0, 0, 2, 0, 0, 1) ...	inner	0	0.0	0.0	0.0
M6	(1, 0, 0, 0, 0, 0, 4, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M7	(1, 0, 0, 0, 0, 0, 3, 1, 0, 0) ...	inner	0	0.0	0.0	0.0
M8	(0, 1, 0, 0, 0, 0, 4, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M9	(0, 1, 0, 0, 0, 0, 3, 0, 1, 0) ...	inner	0	0.0	0.0	0.0
M10	(0, 0, 1, 0, 0, 0, 4, 0, 0, 0) ...	inner	0	0.0	0.0	0.0
M11	(0, 0, 1, 0, 0, 0, 3, 0, 0, 1) ...	inner	0	0.0	0.0	0.0
M12	(0, 0, 1, 0, 0, 0, 2, 0, 0, 2) ...	inner	0	0.0	0.0	0.0
M13	(1, 0, 0, 0, 0, 0, 3, 0, 0, 1) ...	inner	0	0.0	0.0	0.0

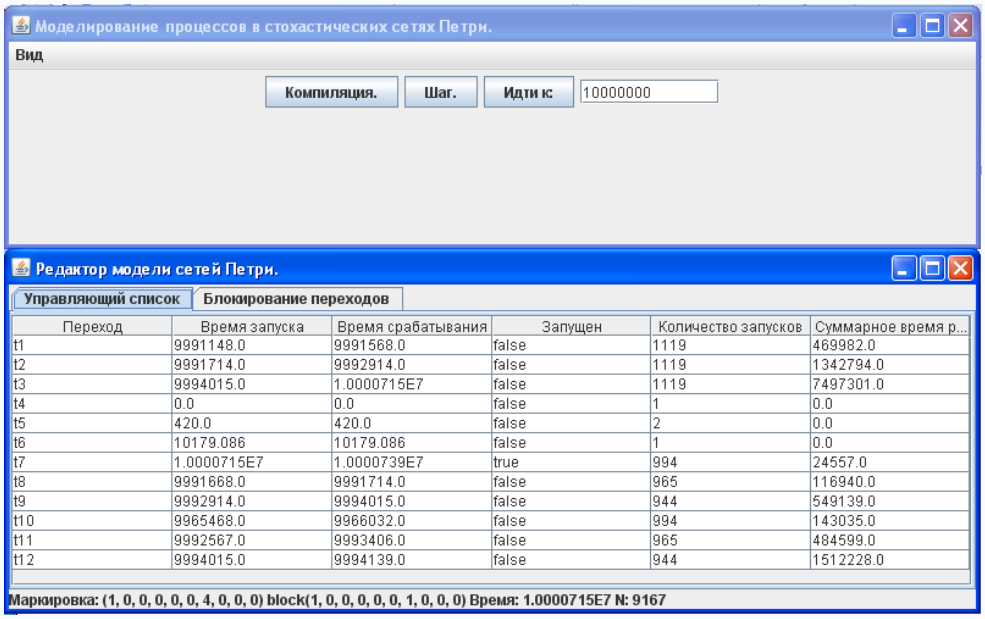


Имя	Из	В	Цепочка запусков ...
T0	M0	M1	(0, 0, 0, 1, 0, 0, 0, 0,...
T1	M0	M2	(1, 0, 0, 0, 2, 0, 0, 0,...
T2	M2	M3	(0, 0, 0, 0, 0, 0, 0, 1,...
T3	M2	M4	(0, 1, 0, 0, 0, 1, 0, 0,...
T4	M4	M5	(0, 0, 0, 0, 0, 0, 0, 0,...
T5	M4	M6	(0, 0, 1, 1, 0, 0, 0, 0,...
T6	M6	M7	(0, 0, 0, 0, 0, 0, 1, 0,...
T7	M6	M8	(1, 0, 0, 0, 0, 0, 0, 0,...
T8	M8	M9	(0, 0, 0, 0, 0, 0, 0, 1,...
T9	M8	M10	(0, 1, 0, 0, 0, 0, 0, 0,...
T10	M10	M11	(0, 0, 0, 0, 0, 0, 0, 0,...
T11	M10	M6	(0, 0, 1, 0, 0, 0, 0, 0,...
T12	M11	M10	(0, 0, 0, 0, 0, 0, 0, 0,...
T13	M11	M12	(0, 0, 0, 0, 0, 0, 0, 0

Построим дерево достижимости:



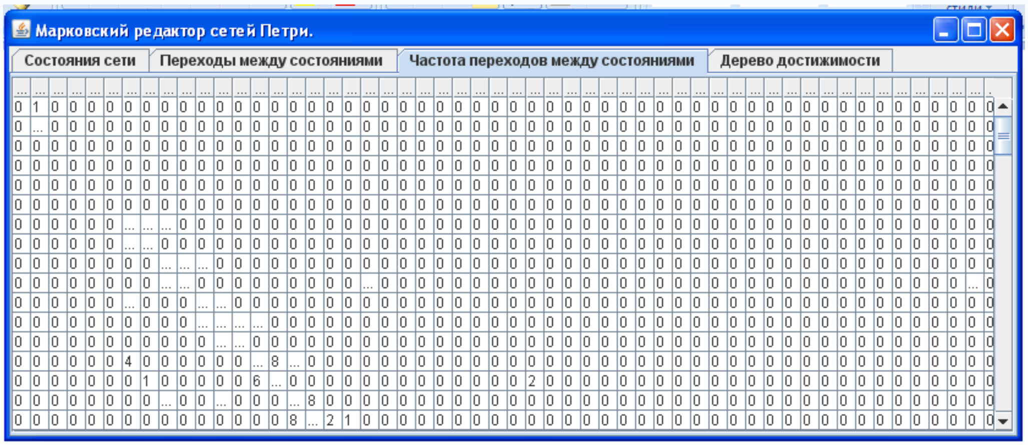
Запустим программу в работу:



Переход	Время запуска	Время срабатывания	Запущен	Количество запусков	Суммарное время р...
t1	9991148.0	9991568.0	false	1119	469982.0
t2	9991714.0	9992914.0	false	1119	1342794.0
t3	9994015.0	1.0000715E7	false	1119	7497301.0
t4	0.0	0.0	false	1	0.0
t5	420.0	420.0	false	2	0.0
t6	10179.086	10179.086	false	1	0.0
t7	1.0000715E7	1.0000739E7	true	994	24557.0
t8	9991668.0	9991714.0	false	965	116940.0
t9	9992914.0	9994015.0	false	944	549139.0
t10	9965468.0	9966032.0	false	994	143035.0
t11	9992567.0	9993406.0	false	965	484599.0
t12	9994015.0	9994139.0	false	944	1512228.0

Результаты моделирования:

Имя	Код	Нотация	Количество п...	Суммарное вр...	Суммарное ...	Последнее ...
M0	(1, 0, 0, 1, 2, 1, 0, 0, 0, 0) block(0, 0, ...	inner	1	0.0	0.0	0.0
M1	(1, 0, 0, 0, 2, 1, 1, 0, 0, 0) block(0, 0, ...	inner	1	420.0	0.0	420.0
M2	(0, 1, 0, 1, 0, 1, 2, 0, 0, 0) block(0, 0, ...	inner	0	0.0	0.0	0.0
M3	(0, 1, 0, 1, 0, 1, 1, 0, 1, 0) block(0, 0, ...	inner	0	0.0	0.0	0.0
M4	(0, 0, 1, 1, 0, 0, 3, 0, 0, 0) block(0, 0, ...	inner	0	0.0	0.0	0.0
M5	(0, 0, 1, 1, 0, 0, 2, 0, 0, 1) block(0, 0, ...	inner	0	0.0	0.0	0.0
M6	(1, 0, 0, 0, 0, 0, 4, 0, 0, 0) block(0, 0, ...	inner	1563	362456.0	9629104.0	9991568.0
M7	(1, 0, 0, 0, 0, 0, 3, 1, 0, 0) block(0, 0, ...	inner	813	68055.0	9897798.0	9965864.0
M8	(0, 1, 0, 0, 0, 0, 4, 0, 0, 0) block(0, 0, ...	inner	1513	1021902.0	8969700.0	9991599.0
M9	(0, 1, 0, 0, 0, 0, 3, 0, 1, 0) block(0, 0, ...	inner	773	231421.0	9761504.0	9992914.0
M10	(0, 0, 1, 0, 0, 0, 4, 0, 0, 0) block(0, 0, ...	inner	1619	6499925.0	3500796.0	1.0000715E7
M11	(0, 0, 1, 0, 0, 0, 3, 0, 0, 1) block(0, 0, ...	inner	807	895095.0	9099043.0	9994139.0
M12	(0, 0, 1, 0, 0, 0, 2, 0, 0, 2) block(0, 0, ...	inner	330	370263.0	9591385.0	9961644.0
M13	(1, 0, 0, 0, 0, 0, 3, 0, 0, 1) block(0, 0, ...	inner	30	6502.0	9666381.0	9672882.0
M14	(1, 0, 0, 0, 0, 0, 2, 1, 0, 1) block(0, 0, ...	inner	11	1866.0	8839978.0	8841845.0
M15	(0, 1, 0, 0, 0, 0, 3, 0, 0, 1) block(0, 0, ...	inner	31	15162.0	9440008.0	9455175.0
M16	(0, 1, 0, 0, 0, 0, 2, 0, 1, 1) block(0, 0, ...	inner	11	2503.5	6592367.5	6594871.5



...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	...	...	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	...	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	...	...	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...	0
0	0	0	0	0	0	...	0	0	0	...	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	...	...	...	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	...	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	4	0	0	0	0	0	0	...	8	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	6	...	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	...	0	0	...	0	0	0	...	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	...	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0