

УДК 681.3+519.1

<https://doi.org/10.33619/2414-2948/53/01>

## АНАЛИЗ ОЦЕНКИ КАЧЕСТВА НАДЕЖНОСТИ ПРОГРАММНОГО ПРОДУКТА НА ОСНОВЕ ТЕОРИИ ГРАФОВ И ЦЕПЕЙ МАРКОВА

©*Ормонова Э. М.*, ORCID: 0000-0002-2476-8722, SPIN-код: 8825-7054, Кыргызско-Узбекский университет, г. Ош, Кыргызстан, [ormonovae@mail.ru](mailto:ormonovae@mail.ru)

## ANALYSIS OF ASSESSING THE RELIABILITY QUALITY OF THE SOFTWARE PRODUCT BASED ON THE GRAPHS THEORY AND MARKOV CHAINS

©*Ormonova E.*, ORCID: 0000-0002-2476-8722, SPIN-code: 8825-7054,  
Kyrgyz-Uzbek University, Osh, Kyrgyzstan, [ormonovae@mail.ru](mailto:ormonovae@mail.ru)

*Аннотация.* В работе для определения качества программного продукта на основе математической модели используются методы теории графов и цепи Маркова. Поскольку современное программирование, разработка структуры данных и создание математической модели программного продукта не может обойтись без теории графов. В качестве объекта исследования применяется язык программирования Pascal ABC. Создана математическая модель программного продукта с помощью теории графов. Разработаны принципы оценки программного качества методами теории графов, а также цепи Маркова. Предложены модели вероятностной оценки качества компьютерных программ с помощью уравнений Колмогорова, составленные на основе теории графов.

*Abstract.* In this paper, we use graph theory and Markov chain methods to determine the quality of a software product based on a mathematical model. Since modern programming, developing a data structure and creating a mathematical model of a software product, we cannot do without graph theory. As an object of research, the Pascal ABC programming language is used. We have created a mathematical model of a software product using graph theory. The principles of software quality assessment by methods of graph theory, as well as Markov chains, are developed. Models of probabilistic assessment of the quality of computer programs using the Kolmogorov equations, based on graph theory.

*Ключевые слова:* теория граф, цепи Маркова, качества программного продукта, дифференциальные уравнения, программирования, таблица инцидентности, уравнения Колмогорова.

*Keywords:* graph theory, Markov chains, software product quality, differential equations, programming, incidence table, Kolmogorov equations.

*Объектами исследования* являются компьютерные программы, а именно язык программирования Pascal ABC. *Цель работы:* целью исследования является определение качества программного продукта с использованием международных и межгосударственных стандартов на основе теории графов. *Методы исследований:* для определения надежности программного средства используется теория графов для составления уравнения Колмогорова.



Задачей исследования является язык программирования Pascal ABC и определение надежности программы с использованием ориентированного графа. Приведена математическая модель программного продукта.

Метод определения надежности компьютерной программы с использованием теории графов путем составления математической модели программного продукта будут применены для определения качества программного продукта.

Основные рекомендации и выводы диссертационного исследования будут иметь прикладной характер и ориентированы на использование при определении менеджмента качества компьютерной программы, используемые для различных прикладных задач.

В частности, теория графов для определения качества программного продукта и математическая модель программного продукта, составленная в виде уравнения Колмогорова, применяется для тестирования программного обеспечения вычислительных и автоматизированных систем.

*Экономическая эффективность.* Составление математической модели программного продукта является наиболее трудной задачей. Существуют различные модели для определения математической модели программного продукта. Поэтому теория графов, использованная нами, для определения качества компьютерной программы Pascal ABC является наиболее экономически эффективным методом.

При определении качества программного обеспечения (программного продукта) актуальной является задача эффективного прогнозирования надежности программного средства и определение вероятностного распределения ошибок в программе [1–2]. Поэтому для определения отдельных показателей качества программного продукта применяются стохастические методы, теории графов и марковские цепи, позволяющие найти количественные значения параметра качества программы [3].

В связи с этим в последние годы теории графов и цепи Маркова (теория массового обслуживания) стали активно применяться в программировании одновременно с использованием ЭВМ в силу удобного выражения задач обработки информации на теоретико-графовом языке. Модель программы в виде управляющего графа, модель арифметического выражения в виде ориентированного дерева, синтаксические деревья, деревья сортировки, сети Петри и другие теоретико-графовые конструкции внесли свой существенный вклад в развитие программирования и его автоматизации. Появление суперкомпьютеров и сетей и возникшая при этом проблема эффективной организации параллельных и распределенных вычислений над информационными массивами большого объема подтвердили тенденцию использования графов как наиболее эффективного средства автоматизации программирования.

В настоящее время эти теории востребованы во многих областях: в химии, информатике, программировании, экономике, логистике, схемотехнике и пр. В математике, например, граф — это абстрактное представление множества объектов и связей между ними. Графом называют пару  $(V, E)$  где  $V$  — это множество вершин, а  $E$  — множество пар, каждая из которых представляет собой связь (эти пары называют ребрами).

Граф может быть ориентированным или неориентированным. В ориентированном графе связи являются направленными (то есть пары в  $E$  являются упорядоченными, например пары  $(a, b)$  и  $(b, a)$  — это две разные связи). В свою очередь в неориентированном графе, связи ненаправленные, и поэтому если существует связь  $(a, b)$  то значит что существует связь  $(b, a)$ .

Сущность теории графов и цепи Маркова заключается в том, что рассматриваемая система переходит последовательно из одного состояния (цикла) в другое, находясь в любой

момент времени только в одном состоянии (положении). Такие переходы называются шагом (циклом) процесса. При этом для анализа случайных процессов с дискретными состояниями удобно пользоваться графом состояний.

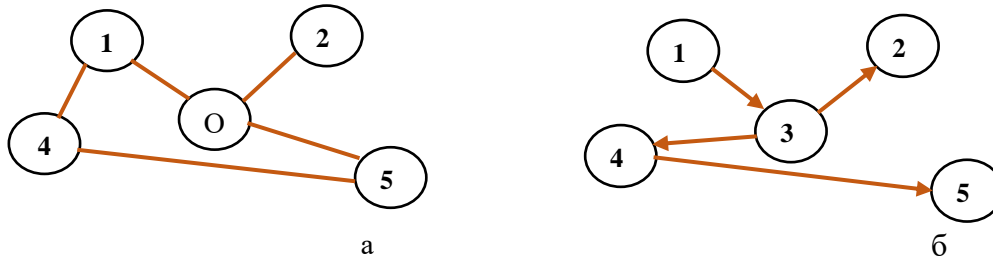


Рисунок 1. а — неориентированный граф б — ориентированный граф [4].

Задачей исследования является определение надежности программного продукта с использованием ориентированного графа (используя язык программирования Pascal ABC).

Рассмотрим пример Pascal ABC, где требуется определить, является ли заданное трехзначное число палиндромом (палиндромом читается одинаково слева направо и справа налево, например, палиндромами являются числа 121, 282, слова «шалаш», «наган»).

```

Program palindrom;
Uses Crt;
var x: Integer;
Begin
Clrscr;
Write ('Введите целое число: ');
readln(x);
If x mod 10 = x div 160 then Write ('Введенное число является палиндромом')
else Write ('Введенное число не является палиндромом');
readln;
End.
    
```

Составляем граф программы, Рисунок 2.

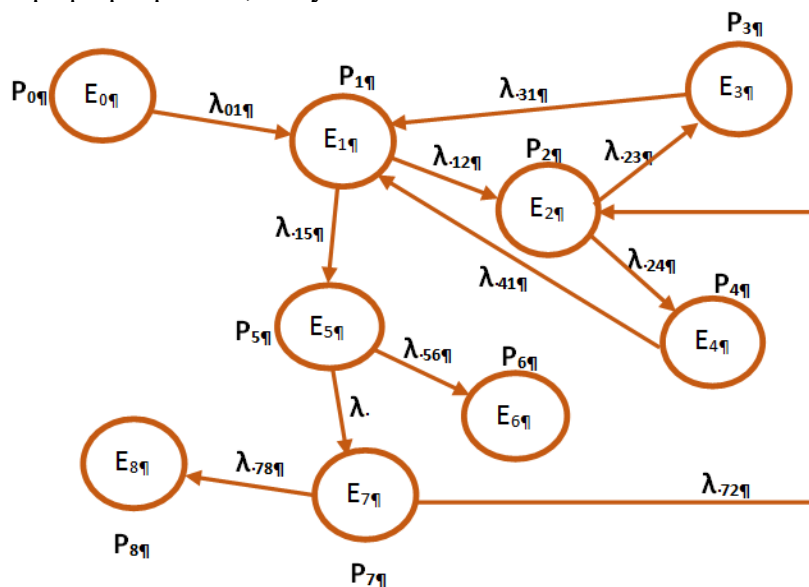


Рисунок 2. Граф программы, где  $E_0$  — запуск программы;  $E_1$  — ввод данных;  $E_2$  — контроль вводимых данных;  $E_3$  — обнаружение ошибок и ее исправление;  $E_4$  — пользователь не обнаружил ошибку;  $E_5$  — программа выполняет свою функцию;  $E_6$  — программный отказ и восстановление программы;  $E_7$  — контроль результатов, выход данных;  $E_8$  — завершение работы программы.

Используя полученный граф, составим таблицу инцидентности.

Таблица.

ТАБЛИЦА ИНЦИДЕНТНОСТИ

	0	1	2	3	4	5	6	7	8
P <sub>0</sub>	-1	0	0	0	0	0	0	0	0
P <sub>1</sub>	1	0	-1	1	1	-1	0	0	0
P <sub>2</sub>	0	1	0	-1	-1	0	0	-1	0
P <sub>3</sub>	0	-1	1	0	0	0	0	0	0
P <sub>4</sub>	0	-1	1	0	0	0	0	0	0
P <sub>5</sub>	0	1	0	0	0	0	-1	-1	0
P <sub>6</sub>	0	0	0	0	0	1	0	0	0
P <sub>7</sub>	0	0	1	0	0	1	0	0	-1
P <sub>8</sub>	0	0	0	0	0	0	0	1	0

Здесь P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub> ... P<sub>8</sub> — вероятности переходов из одного состояния в другое, а +λ коэффициенты переходов. Матрица инцидентности для определения качества программного продукта является инструментом качественного анализа программного продукта (<https://habr.com/ru/post/65367/>).

Для аналитического определения параметров качества программ нужно составить уравнения Колмогорова. Используя общее правило составления уравнений Колмогорова для нашего графа (цепи Маркова) получим следующие дифференциальные уравнения.

Уравнение Колмогорова для данного графа:

$$\begin{aligned} \frac{dP_0}{dt} &= -\lambda_{01}P_0 \\ \frac{dP_1}{dt} &= -\lambda_{12}P_1 - \lambda_{15}P_1 + \lambda_{31}P_3 + \lambda_{41}P_4 + \lambda_{01}P_0 = -(\lambda_{12} + \lambda_{15})P_1 + \lambda_{31}P_3 + \lambda_{41}P_4 + \lambda_{01}P_0 \\ \frac{dP_2}{dt} &= -\lambda_{24}P_2 - \lambda_{23}P_2 + \lambda_{72}P_7 + \lambda_{12}P_1 = -(\lambda_{24} + \lambda_{23})P_2 + \lambda_{72}P_7 + \lambda_{12}P_1 \\ \frac{dP_3}{dt} &= \lambda_{23}P_2 - \lambda_{31}P_3 \\ \frac{dP_4}{dt} &= \lambda_{24}P_2 - \lambda_{41}P_4 \\ \frac{dP_5}{dt} &= \lambda_{65}P_6 + \lambda_{15}P_1 - \lambda_{56}P_5 - \lambda_{57}P_5 = -(\lambda_{56} + \lambda_{57})P_5 + \lambda_{65}P_6 + \lambda_{15}P_1 \\ \frac{dP_6}{dt} &= \lambda_{65}P_6 - \lambda_{65}P_6 \\ \frac{dP_7}{dt} &= \lambda_{57}P_5 - \lambda_{78}P_7 - \lambda_{72}P_7 = -(\lambda_{78} + \lambda_{72})P_7 + \lambda_{57}P_5 \\ \frac{dP_8}{dt} &= \lambda_{78}P_7 \end{aligned} \tag{1}$$

Таким образом, получили системы из 9 линейных дифференциальных уравнений с неизвестными функциями P<sub>0</sub>, P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>, P<sub>7</sub>, P<sub>8</sub>.

Для них выполняется следующее нормировочное условие:

$$P_0 + P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 = 1 \quad (2)$$

Чтобы решить системы уравнений необходимо задать начальные условия. Если вероятности  $P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8$  постоянны, то их производные равны нулю, т. е. чтобы найти вероятности нужно все левые части в уравнениях (1) положить равными нулю.

Тогда получим:

$$\begin{aligned} -\lambda_{01}P_0 &= 0 \\ \lambda_{31}P_3 + \lambda_{41}P_4 + \lambda_{01}P_0 - (\lambda_{12} + \lambda_{15})P_1 &= 0 \\ \lambda_{72}P_7 + \lambda_{12}P_1 - (\lambda_{24} + \lambda_{23})P_2 &= 0 \\ \lambda_{23}P_2 - \lambda_{31}P_3 &= 0 \\ \lambda_{24}P_2 - \lambda_{41}P_4 &= 0 \\ \lambda_{65}P_6 + \lambda_{15}P_1 - (\lambda_{56} + \lambda_{57})P_5 &= 0 \\ \lambda_{65}P_6 - \lambda_{65}P_6 &= 0 \\ \lambda_{57}P_5 - (\lambda_{78} + \lambda_{72})P_7 &= 0 \\ \lambda_{78}P_7 &= 0 \end{aligned} \quad (3)$$

Для решения системы уравнений (3) воспользуемся нормировочным условием (2).

Таким образом, моделирование качества программного продукта (ПП) с помощью математического аппарата теории графов и Марковских цепей эффективно решает практические задачи в области прогнозирования качества ПП, так как цепи Маркова позволяет изучать поведения систем с дискретным множеством состояний.

#### Выводы:

1. Разработаны принципы оценки качества программного методами теории графов, а также цепи Маркова.
2. Предложены модели вероятностной оценки качества компьютерных программ с помощью уравнений Колмогорова, составленные на основе теории графов.

#### Список литературы:

1. Назаров А. А., Терпугов А. Ф. Теория массового обслуживания. Томск, 2004. 225 с.
2. Гнеденко Б. В. Введение в теорию массового обслуживания. М.: КомКнига, 2005. 397 с.
3. Климкина Н. Л., Гриценко Е. М. Программный комплекс статистического анализа вероятностных процессов на основе цепей Маркова. Саратов: СГТУ. Сер.: Наука и образование, 2007.
4. Whittaker J. A., Thomason M. G. A Markov chain model for statistical software testing // IEEE Transactions on Software engineering. 1994. V. 20. №10. P. 812-824. <https://doi.org/10.1109/32.328991>
5. Sayre K. D., Poore J. Improved techniques for software testing based on Markov chain usage models. University of Tennessee, Knoxville, 1999.

*References:*

1. Nazarov, A. A., & Terpugov, A. F. (2004). *Teoriya massovogo obsluzhivaniya*. Tomsk.
2. Gnedenko, B. V. (2005). *Vvedenie v teoriyu massovogo obsluzhivaniya*. Moscow.
3. Klimkina, N. L., & Gritsenko, E. M. (2007). Programmnyi kompleks statisticheskogo analiza veroyatnostnykh protsessov na osnove tsepei Markova. Saratov.
4. Whittaker, J. A., & Thomason, M. G. (1994). A Markov chain model for statistical software testing. *IEEE Transactions on Software engineering*, 20(10), 812-824. <https://doi.org/10.1109/32.328991>
5. Sayre, K. D., & Poore, J. (1999). *Improved techniques for software testing based on Markov chain usage models* (Doctoral dissertation, University of Tennessee, Knoxville).

*Работа поступила  
в редакцию 02.03.2020 г.*

*Принята к публикации  
07.03.2020 г.*

*Ссылка для цитирования:*

Ормонова Э. М. Анализ оценки качества надежности программного продукта на основе теории графов и цепей Маркова // Бюллетень науки и практики. 2020. Т. 6. №4. С. 12-17. <https://doi.org/10.33619/2414-2948/53/01>

*Cite as (APA):*

Ormonova, E. (2020). Analysis of Assessing the Reliability Quality of the Software Product Based on the Graphs Theory and Markov Chains. *Bulletin of Science and Practice*, 6(4), 12-17. <https://doi.org/10.33619/2414-2948/53/01> (in Russian).

