

ПРИМЕНЕНИЕ МЕТРИК ХОЛСТЕДА ДЛЯ КОЛИЧЕСТВЕННОГО ОЦЕНИВАНИЯ ХАРАКТЕРИСТИК ПРОГРАММ ЭВМ

А. В. АВЕРЬЯНОВ, И. Н. КОШЕЛЬ, В. В. КУЗНЕЦОВ

*Военно-космическая академия им. А.Ф. Можайского, 197198, Санкт-Петербург, Россия
E-mail: Aver957@mail.ru*

Использованы метрики Холстеда для статистического выражения алгоритмов на языках программирования в мнемосокодах и машинных кодах. Объектом исследования являются три программы в мнемосокодах и машинных кодах, реализующих разветвляющийся алгоритм, алгоритм, связанный с обработкой массива, и алгоритм, использующий подпрограмму. Установлено, что оценки длины программ существенно отличаются от значений, полученных в результате непосредственного анализа текстов программ. Доказано, что при переходе от программирования в мнемосокодах к машинным кодам при реализации алгоритмов значительно увеличивается объем программ. Чем ниже уровень используемого языка программирования, тем больше объем программы, что в результате приводит к увеличению количества потенциальных ошибок и снижению надежности функционирования программного обеспечения.

Ключевые слова: метрики Холстеда, оператор, операнд, длина, словарь программы, уровень языка программирования, объем программы, надежность программного обеспечения

Основным средством определения количественных показателей надежности программного обеспечения (ПО) ЭВМ являются модели надежности, представляющие собой математические функции. Модели предназначены для оценки и предсказания значений различных метрик надежности ПО на всех этапах его жизненного цикла, а также для решения ряда других задач, возникающих в процессе разработки программ [1]. Существующие эмпирические модели надежности ПО основаны на анализе структурных особенностей комплексов программ и определяют зависимость показателей надежности от числа межмодульных связей, количества циклов в модулях, отношения числа прямолинейных участков программы к числу точек повторения и т.д. [2].

Один из наиболее распространенных подходов к прогнозированию надежности программ основывается на анализе их текста. Этот подход впервые был изложен в книге М. Х. Холстеда „Начала науки о программах“ [3], предложенные им характеристики известны как „метрики Холстеда“. Они относятся к эмпирическим моделям, которые можно построить на этапе проектирования ПО, когда уже осуществлена разбивка на модули и известна структура программ.

Согласно Холстеду, свойства любого алгоритма (или программы для ЭВМ), которые могут быть вычислены или измерены, включают следующие метрические характеристики [3]:

η_1 — количество отдельных операторов;

η_2 — количество отдельных операндов;

N_1 — общее число всех операторов, появляющихся в данной программе;

N_2 — общее число всех операндов, появляющихся в данной программе.

Через первичные метрические характеристики выражаются словарь

$$\eta = \eta_1 + \eta_2 \quad (1)$$

и длина программы ЭВМ

$$N = N_1 + N_2. \quad (2)$$

Холстед предложил оценивать длину программы по формуле

$$\dot{N} = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2. \quad (3)$$

В формуле (3) левой части \dot{N} присвоен символ „ $\dot{\cdot}$ “ для того, чтобы отличать вычисленную длину от значения N (2), полученного в результате непосредственного анализа программы. Соотношение (3) рассматривается как приближенное. Очевидно, что $N \neq \dot{N}$. Ответ на вопрос, насколько \dot{N} дает точную оценку длины алгоритма в зависимости от использованного в нем словаря η (1), имеет важное практическое значение.

Проверку уравнения длины проведем с использованием трех программ в мнемосокодах учебной ЭВМ, реализующих разветвляющийся алгоритм, алгоритм, связанный с обработкой массива, и алгоритм, использующий подпрограмму [4, 5]. Эти программы применялись при анализе программных моделей ЭВМ с использованием принципа Парето [6], а также для количественного оценивания частоты использования машинных команд ЭВМ [7].

Выбор указанных выше алгоритмов обосновывается еще и тем, что проверку уравнения (3) необходимо проводить прежде всего для небольших программ, чтобы неизвестные эффекты модульности не замаскировали основное проверяемое уравнение [3].

Мнемосокоды трех исследуемых программ представлены в табл. 1. Первая программа реализует алгоритм вычисления среднего арифметического максимальных элементов трех заданных массивов чисел, вторая — алгоритм вычисления функции, третья — алгоритм вычисления суммы десяти элементов массива.

Таблица 1

Номер команды	1. Программа с подпрограммой MAX		2. Программирование разветвляющегося процесса		3. Программирование цикла с переадресацией (обработка массива)	
	Мнемосокоды	Машинные коды	Мнемосокоды	Машинные коды	Мнемосокоды	Машинные коды
1	RD #85	21 1 085	IN	01 0 000	RD #40	21 1 040
2	WR R1	32 0 001	WR 30	22 0 030	WR 30	22 0 030
3	RD #14	21 1 014	SUB #16	24 1 016	RD #10	21 1 010
4	WR R2	32 0 002	JS 010	13 0 010	WR 31	22 0 031
5	CALL M	19 0 022	RD 30	21 0 030	RD #0	21 1 000
6	WR R6	32 0 006	SUB #11	24 1 011	WR 32	22 0 032
7	RD #100	22 1 100	WR 31	22 0 031	M1: RD 32	21 0 032
8	WR R1	32 0 001	MUL 31	25 0 031	ADD @30	23 2 030
9	RD #4	21 1 004	SUB #125	24 1 125	WR 32	22 0 032
10	WR R2	32 0 002	JMP 020	10 0 020	RD 30	21 0 030
11	CALL M	19 0 022	RD 30	21 0 030	ADD #1	23 1 001
12	WR R7	32 0 007	MUL 30	25 0 030	WR 30	22 0 030
13	RD #110	21 1 110	WR 31	22 0 031	RD 31	21 0 031
14	WR R1	32 1 001	RD 30	21 0 030	SUB #1	24 1 001
15	RD #9	21 1 009	MUL #72	25 1 072	WR 31	22 0 031
16	WR R2	32 0 002	ADD 31	23 0 031	JNZ M1	12 0 006
17	CALL M	19 0 022	ADI 10640	43 0 000	RD 32	21 0 032
				10 6 400		
18	ADD R7	33 0 007	DIVI 100168	46 0 000	OUT	02 0 000
				10 0 168		
19	ADD R6	33 0 006	OUT	02 0 000	HLT	09 0 000
20	DIV #3	26 1 003	HLT	09 0 000		
21	OUT	02 0 000				
22	HLT	09 0 000				
23	RD @R1	31 4 001				
24	WR R3	32 0 003				
25	L2: RD @R1+	31 5 001				
26	WR R4	32 0 004				

Продолжение табл. 1

Номер команды	1. Программа с подпрограммой MAX		2. Программирование разветвляющегося процесса		3. Программирование цикла с переадресацией (обработка массива)	
	Мнемокоды	Машинные коды	Мнемокоды	Машинные коды	Мнемокоды	Машинные коды
27	SUB R3	34 0 003				
28	RD R3	31 0 003				
29	JS L1	13 0 031				
30	MOV R3,R4	30 0 034				
31	L1: JRNZ R2,L2	17 0 025				
32	RD R3	31 0 003				
33	RET	08 0 000				

Мнемокоды подпрограммы MAX — поиска максимального элемента массива с соответствующими отдельным командам комментариями — приведены в табл. 2.

Таблица 2

Команда	Примечания
M: RD @R1 WR R3	Загрузка первого элемента массива в регистр R3
L2: RD @R1+ WR R4 SUB R3 JS L1	Чтение элемента массива и модификация адреса Сравнение и замена, если R3 < R4
L1: JRNZ R2, L2	Цикл
RD R3	Чтение результата в аккумулятор
RET	Возврат в основную программу

Все управляющие структуры в программах, например JS..., JNZ..., JRNZ..., классифицируются как операторы. Метки M:, L1: и L2 — не переменные и не константы, поэтому они не являются операндами, а относятся к составным частям операторов CALL M, JS L1, JRNZ R2, L2.

Первичные метрические характеристики η_1 , η_2 , N_1 , N_2 , словарь η , длина N и приближенное значение оценки длины \dot{N} для трех исследуемых программ в мнемокодах приведены в табл. 3.

Таблица 3

Программа в мнемокодах	η_1	η_2	η	N_1	N_2	N	\dot{N}	$ N - \dot{N} $	V , бит
Программирование разветвляющегося процесса	12	8	20	20	9	29	67	38	125
Программирование цикла с переадресацией	7	13	20	20	15	35	68	33	151
Программа с подпрограммой	12	32	44	52	41	93	203	110	508

Представленные в табл. 3 результаты показывают, что оценки \dot{N} длины программ существенно отличаются от значений N , полученных в результате непосредственного анализа текста программ. Это противоречит утверждению, что „ \dot{N} дает довольно точную оценку длины алгоритма в виде простой зависимости от использованного в нем словаря“ [3].

Объяснение этого противоречия заключается в следующем:

— приведенное в [3] обоснование справедливости соотношения (3) соответствует программам, написанным на языках высокого уровня Алгол, Фортран, ПЛ/1;

— соотношение (3) справедливо прежде всего для длинных программ: например, для 120 программ, написанных на ПЛ/1, с общим числом операторов, превышающим 100 000, коэффициент корреляции между N и \dot{N} составляет 0,98; таким образом, для больших программ корреляция между N и \dot{N} достаточно велика [8].

Важной метрической характеристикой алгоритма является его размер. Соответствующая метрическая характеристика размера при любой реализации (программы) алгоритма называется объемом V программы, выраженным в битах:

$$V = N \log_2 \eta.$$

Для рассматриваемых трех алгоритмов объемы программ приведены в табл. 3.

Если конкретный алгоритм переводится с одного языка на другой, то объем программы меняется. Действительный объем V увеличивается или уменьшается в зависимости от развитости используемых языков программирования. Например, при переводе алгоритма с Фортрана в машинный код некой конкретной ЭВМ объем программы увеличивается. Для 8-разрядных микропроцессоров машинные коды команд имеют длину 1, 2 или 3 байта. Первый байт указывает код операции, а последующие — ее операнды (непосредственные значения, ссылки на регистры или адреса памяти) [9, 10]. Выполнение программы начинается с первой машинной инструкции (команды) и осуществляется последовательно от инструкции к инструкции (естественный порядок выборки команд).

Первичные метрические характеристики, а также параметры N , \dot{N} и V для трех исследуемых программ в машинных кодах приведены в табл. 4, а машинные коды программ — в табл. 1. Необходимо отметить, что в модели учебной ЭВМ используется десятичная система счисления для кодирования команд и представления данных.

Таблица 4

Программа в машинных кодах	η_1	η_2	η	N_1	N_2	N	\dot{N}	$ N - \dot{N} $	V , бит
Программирование разветвляющегося процесса	13	11	24	20	22	42	86	44	193
Программирование цикла с переадресацией	9	17	26	20	108	128	98	30	602
Программа с подпрограммой	15	42	57	48	52	100	285	185	583

Анализ полученных результатов показывает, что при переходе от программирования в мнемосокодах к машинным кодам при реализации алгоритмов значительно увеличивается объем V программ (при программировании цикла с переадресацией почти в четыре раза). Объем V является одной из важных для практики метрик Холстеда, так как определяет число потенциальных (переданных) ошибок, введенных в программу. Если выразить число ошибок в виде соотношения [3]

$$B = V/3000,$$

то получим значения B , приведенные в табл. 5.

Таблица 5

Программа	Программа в мнемосокодах	Программа в машинных кодах
Программирование разветвляющегося процесса	0,042	0,064
Программирование цикла с переадресацией	0,050	0,201
Программа с подпрограммой	0,170	0,194

На основании вышеизложенного и полученных оценок метрических характеристик можно сделать следующие выводы.

1. Для небольших программ в мнемосокодах объемом несколько сотен бит оценки \dot{N} их длины могут существенно отличаться от значений N , полученных в результате непосредственного анализа текста программ. Поэтому для таких программ целесообразно использовать метрики, выражающие их словарь и длину, полученные в результате прямого подсчета количества отдельных операторов и операндов, а также общего числа всех операторов и операндов.

2. Переход при программировании от мнемочкодов к машинным кодам (языку низшего уровня) приводит к значительному росту объемов программ, реализующих исследуемые алгоритмы решения задач.

3. Число потенциальных ошибок, введенных в программу при программировании, является важнейшей характеристикой, определяющей надежность функционирования ПО ЭВМ. Чем ниже уровень используемого языка программирования, тем больше объем программы, что в результате приводит к увеличению количества потенциальных ошибок и уменьшению надежности программ.

СПИСОК ЛИТЕРАТУРЫ

1. Мороз Г. Б. Пуассоновские модели роста надежности программного обеспечения и их приложение. Аналитический обзор // Управляющие системы и машины. 1996. № 1—2. С. 69—84.
2. Эксплуатация средств вычислительной техники. Ч. 1. Аппаратные средства вычислительной техники: Учебник / Под ред. Г. В. Крэмеза. СПб: ВКА им. А. Ф. Можайского, 2013. 505 с.
3. Холстед М. Х. Начала науки о программах / Пер. с англ. М.: Финансы и статистика, 1981. 128 с.
4. Жмакин А. П. Архитектура ЭВМ. СПб: БХВ-Петербург, 2010. 352 с.
5. Басыров А. Г. Организация ЭВМ и систем: практикум. СПб: ВКА им. А. Ф. Можайского, 2012. 83 с.
6. Аверьянов А. В., Белая Т. И., Молчанов О. Е. Анализ программных моделей учебных ЭВМ с использованием принципа Парето // Естественные и технические науки. 2016. № 6. С. 160—164.
7. Аверьянов А. В., Калюжный А. В. Применение анализа Парето для количественного оценивания частоты использования машинных команд ЭВМ // Изв. вузов. Приборостроение. 2019. Т. 62, № 2. С. 101—105.
8. Надежность автоматизированных систем: практикум / Сост.: К. А. Эсаулов, А. В. Аверьянов, В. В. Кузнецов. СПб: ВКА им. А. Ф. Можайского, 2014. 71 с.
9. Микропроцессорные средства вычислительной техники. Ч. 1. Микропроцессоры / Под общ. ред. О. Е. Молчанова. Л.: ВИКИ им. А.Ф. Можайского, 1990. 168 с.
10. Абель П. Ассемблер. Язык и программирование для IBM PC: Пер. с англ. Киев: НТИ, 2003. 734 с.

Сведения об авторах

- Алексей Васильевич Аверьянов** — канд. техн. наук, доцент; ВКА им. А. Ф. Можайского, кафедра информационно-вычислительных систем и сетей; E-mail: Aver957@mail.ru
- Игорь Николаевич Кошель** — ВКА им. А. Ф. Можайского, кафедра информационно-вычислительных систем и сетей; начальник факультета; E-mail: kin1470@mail.ru
- Вадим Викторович Кузнецов** — канд. техн. наук, доцент; ВКА им. А. Ф. Можайского, кафедра информационно-вычислительных систем и сетей; E-mail: vadimkuznetsov@mail.ru

Поступила в редакцию
03.06.19 г.

Ссылка для цитирования: Аверьянов А. В., Кошель И. Н., Кузнецов В. В. Применение метрик Холстеда для количественного оценивания характеристик программ ЭВМ // Изв. вузов. Приборостроение. 2019. Т. 62, № 11. С. 970—975.

APPLICATION OF HALSTEAD METRICS FOR QUANTITATIVE ESTIMATION OF COMPUTER PROGRAM CHARACTERISTICS

A. V. Averyanov, I. N. Koshel, V. V. Kuznetsov

A. F. Mozhaisky Military Space Academy, 197198, St. Petersburg, Russia
E-mail: Aver957@mail.ru

Halstead metrics are applied to statistical characterization of algorithms in programming languages in mnemonic and machine codes. Three programs in mnemonic codes and machine codes are chosen as the studied objects; the programs implement a branching algorithm, an algorithm associated with array processing, and an algorithm using a subroutine, correspondingly. The derived estimates of the programs length are established to differ significantly from the values obtained as a result of direct analysis of program texts. It is proved that the transition from programming in mnemonic codes to machine codes in the implementation of algorithms significantly increases the program volume. The lower the level of the programming language used, the larger the volume of the program, which as a result leads to an increase in the number of potential errors and a decrease in the reliability of software functioning.

Keywords: Halstead metrics, operator, operand, length, program dictionary, programming language level, program value, software reliability

REFERENCES

1. Moroz G.B. *Upravlyayushchiye sistemy i mashiny*, 1996, no. 1–2, pp. 69–84. (in Russ.)
2. Kremez G.V., ed., *Eksploatatsiya sredstv vychislitel'noy tekhniki. Chast' 1. Apparatsnyye sredstva vychislitel'noy tekhniki* (Operation of Computer Equipment. Part 1. Computer Hardware), St. Petersburg, 2013, 505 p. (in Russ.)
3. Halstead M.H. *Elements of Software Science*, NY, Elsevier, 1977, 127 p.
4. Zhmakin A.P. *Arkhitektura EVM* (Computer Architecture), St. Petersburg, 2010, 352 p. (in Russ.)
5. Basyrov A.G. *Organizatsiya EVM i sistem: praktikum* (Organization of Computers and Systems: Workshop), St. Petersburg, 2012, 83 p. (in Russ.)
6. Aver'yanov A.V., Belaya T.I., Molchanov O.E. *Estestvennye i tekhnicheskiye nauki*, 2016, no. 6, pp. 160–164. (in Russ.)
7. Aver'yanov A.V., Kalyuzhnyy A.V. *Journal of Instrument Engineering*, 2019, no. 2(62), pp. 101–105. (in Russ.)
8. Esaulov K.A., Aver'yanov A.V., Kuznetsov V.V. *Nadezhnost' avtomatizirovannykh sistem: praktikum* (Reliability of Automated Systems: Workshop), St. Petersburg, 2014, 71 p. (in Russ.)
9. Molchanov O.E., ed., *Mikroprotsessornyye sredstva vychislitel'noy tekhniki. Chast' 1. Mikroprotsessory* (Microprocessor-Based Computing Equipment. Part 1. Microprocessors), Leningrad, 1990, 168 p. (in Russ.)
10. Abel P. *IBM PC Assembler Language and Programming*, Prentice Hall, 1987.

Data on authors

- Alexey V. Averyanov** — PhD, Associate Professor; A. F. Mozhaisky Military Space Academy, Department of Information Systems and Networks; E-mail: Aver957@mail.ru
- Igor N. Koshel** — A. F. Mozhaisky Military Space Academy, Department of Information Systems and Networks; Head of the Faculty; E-mail: kin1470@mail.ru
- Vadim V. Kuznetsov** — PhD, Associate Professor; A. F. Mozhaisky Military Space Academy, Department of Information Systems and Networks; E-mail: vadimkuznetsov@mail.ru

For citation: Averyanov A. V., Koshel I. N., Kuznetsov V. V. Application of Halstead metrics for quantitative estimation of computer program characteristics. *Journal of Instrument Engineering*. 2019. Vol. 62, N 11. P. 970–975 (in Russian).

DOI: 10.17586/0021-3454-2019-62-11-970-975