

УДК 004.05+004.415.5

В.Л. ПЕТРИК, Ю.С. МАНЖОС

Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Украина

МОДЕЛЬ ДЕФЕКТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Предлагается модель дефектов программного обеспечения, основанная на целочисленном дескрипторном отображении. Программный код рассмотрен как упорядоченное множество программных конструкций, каждая из которых с некоторой вероятностью может содержать остаточный программный дефект. Модель дефектов учитывает адресные операции, операции перехода и позволяет на основе статистических характеристик кода оценить долю остаточных дефектов, связанных с нарушением семантических инвариантов, и эффективность метода, основанного на контроле целочисленных семантических дескрипторов.

Ключевые слова: семантический контроль, семантический инвариант, целочисленный семантический дескриптор, остаточный программный дефект.

Введение

Сложность программного обеспечения (ПО) современных информационно-управляющих систем (ИУС) обуславливает значительную ресурсоемкость верификации, основанной на классических методах. Это вынуждает искать новые формальные методы доказательства корректности ПО.

Постановка задачи. Одним из методов доказательства корректности является семантический контроль (СК), основанный на контроле физических размерностей программных переменных.

Однако СК дает возможность доказывать лишь частичную корректность программного обеспечения, проявляющуюся в сохранении семантических инвариантов [1, 2].

Известные модели дефектов позволяют оценить достоверность контроля, выполняемого при статическом анализе (СА) исходного кода. Применение целочисленного дескрипторного отображения (ЦДО) благодаря снижению ресурсоемкости сделало возможным использование СК в реальном времени для диагностики и контроля корректности вычислительных процессов (ВП) [3 – 5].

В связи с этим возникла необходимость построения новой модели остаточных дефектов (ОД) программного обеспечения, учитывающей адресные операции, операции перехода и позволяющей на основе статистических характеристик программного кода и ЦДО, оценить достоверность контроля ВП и эффективность метода, основанного на контроле целочисленных семантических дескрипторов (ЦСД).

Решению этой задачи посвящена данная статья.

1. Построение модели дефектов программного обеспечения

Рассмотрим программу как упорядоченное множество программных конструкций, каждая из которых с некоторой вероятностью может содержать ОД, обнаружение которых возможно с использованием контроля преобразований ЦСД операндов и результатов операций. Под ОД будем понимать дефекты, не обнаруживаемые компилятором и редактором связей.

Все программные конструкции можно разделить на операции (выполняемые команды) и операнды. Вероятности их появления в коде определяются арностью (средним количеством операндов) соответствующих операций и функций.

Операции подразделяются на: унарные – минус, abs, sign, sin, cos...; бинарные – сложение, вычитание, умножение, деление, возведение в степень...; n-арные – системные функции, требующие более двух параметров. Примем, что средняя арность равна Z.

Вероятности появления различных команд определяются на основе СА исходного кода или динамического анализа программы посредством формирования статического или динамического операционных спектров [1] и спектров данных. Динамический спектр существенно изменяется для различных режимов работы и отличается от статического.

Обозначим далее:

I – элемент программной конструкции; C – команда (операция); O – операнд; N – константа; A – арифметическая команда; L – логическая команда; R – команда отношения; J – команда перехода; F – вы-

зов функции; T, T^{-1} – прямая и обратная тригонометрические функции; M – математическая функция; S – системная функция; K – скалярная переменная; Q – адресуемый операнд; $+$ – сложение; $-$ – вычитание; $*$ – умножение; $/$ – деление;

P_O, P_C – вероятности операндов и команд в коде;

P_A, P_R, P_L, P_J, P_F – вероятности появления операций арифметических, отношения, логических, перехода, вызовов функций;

$P_+, P_-, P_*, P_/\$ – вероятности появления сложения, вычитания, умножения, деления;

$P_T, P_{T^{-1}}, P_M, P_S$ – вероятности появления вызовов прямых, обратных тригонометрических функций, математических функций, вызовов системных функций.

Сами операнды подразделяются на константы (числа), скалярные переменные и адресуемые переменные (элементы массивов и операнды, адресуемые указателями).

Вероятности появления различных классов операндов также определяются на основе статического или динамического анализа.

Обозначим далее: P_v – вероятность появления переменных; P_N – вероятность появления констант; P_Q – вероятность появления адресуемых переменных (элементов массивов); P_K – вероятность появления скалярных переменных.

Представим программный код в виде модели, часть элементов которой может быть искажена, т.е. содержать ОД, часть которых может быть обнаружена (рис. 1).

Элементы, которые могут содержать ОД, обозначены волнистой чертой. Элементы, не содержащие ОД, обозначены без надчеркивания. Элементы, содержащие ОД, надчеркнуты прямой линией.

Вероятность существования ОД

$$P_{def} = \sum_i \tilde{P}_i \bar{P}_i, \quad (1)$$

где i – тип элемента программной конструкции; \tilde{P}_i – вероятность появления элемента программной конструкции i -го типа, который может содержать ОД; \bar{P}_i – вероятность существования ОД в программной конструкции i -го типа.

Элементы, дефектность которых может быть обнаружена посредством контроля ЦСД, обозначены знаком «^». Тогда вероятность обнаружения ОД i -го типа программной конструкции \hat{P}_i .

Элементы, дефектность которых не может быть распознана, отмечены на графе двойным надчеркиванием. Вероятность их нераспознавания – $\bar{\bar{P}}_i$.

Следует отметить, что для каждого из типов программных элементов события распознавания и нераспознавания, при условии существования ОД, образуют полные группы.

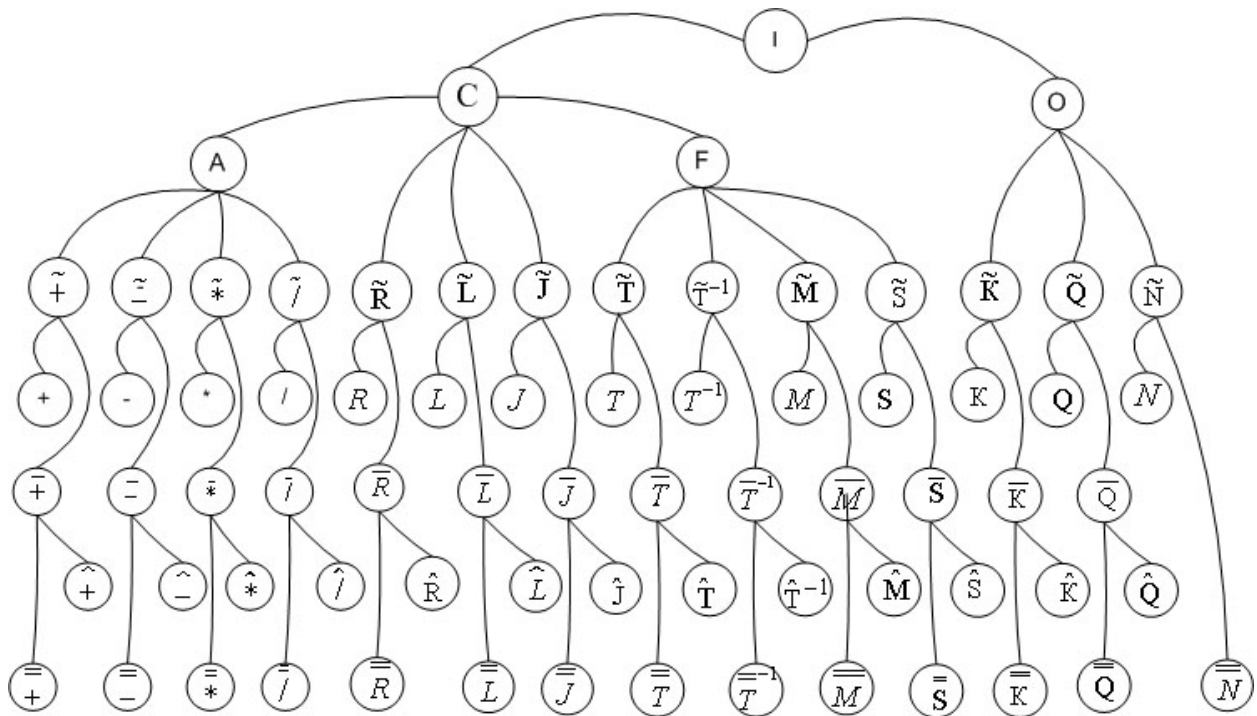


Рис. 1. Модель дефектов программного обеспечения

2. Анализ эффективности метода и достоверности контроля

В условиях принятых ограничений оценим для всех типов элементов программных конструкций абсолютные вероятности \hat{P}_i и $\bar{\bar{P}}_i$.

Построив декартово произведение арифметических операций, получим как всевозможные искажения результата, так и отсутствие ОД элементарных арифметических операций, приведенные в табл. 1. В первом столбце обозначены исходные арифметические операции, которые могут искажаться, результатом чего будет: ОК – отсутствие ОД; Е, F – нераспознаваемый и распознаваемый ОД.

Для операций отношения $\{<, \leq, >, \geq, \neq, =\}$, имеющих суммарную вероятность появления \tilde{P}_R , ОД не может быть распознан, поэтому $\hat{P}_R = 0$, а $\bar{\bar{P}}_R = \bar{P}_R \tilde{P}_R P_C$.

Таблица 1

Возможные исходы искажений арифметических операций

Операция	Результат искажения				Вероятности обнаружения и необнаружения ОД при искажении операций	
	+	-	*	/	\hat{P}_i (i = +, -, *, /)	$\bar{\bar{P}}_i$ (i = +, -, *, /)
+	OK	E	F	F	$\frac{2\bar{P}_+ \tilde{P}_+}{3} P_A P_C$	$\frac{\bar{P}_+ \tilde{P}_+}{3} P_A P_C$
-	E	OK	F	F	$\frac{2\bar{P}_- \tilde{P}_-}{3} P_A P_C$	$\frac{\bar{P}_- \tilde{P}_-}{3} P_A P_C$
*	F	F	OK	F	$\bar{P}_* \tilde{P}_* P_A P_C$	0
/	F	F	F	OK	$\bar{P}_/ \tilde{P}_/ P_A P_C$	0

Для логических операций, имеющих суммарную вероятность появления \tilde{P}_L , ОД, связанный с применением некорректной операции из множества {AND, OR, NOT, XOR}, не может быть распознан, поэтому $\hat{P}_L = 0$, а $\bar{\bar{P}}_L = \bar{P}_L \tilde{P}_L P_C$.

Для операций перехода, имеющих суммарную вероятность появления \tilde{P}_J , ОД, связанный с некорректным адресом перехода, не может быть распознан, поэтому $\hat{P}_J = 0$, а $\bar{\bar{P}}_J = \bar{P}_J \tilde{P}_J P_C$.

Для вызовов прямых тригонометрических функций, имеющих суммарную вероятность появления в программе \tilde{P}_T , ОД, проявляющийся в вызове неверных функций sin, cos, tg, не может быть распознан, поэтому $\hat{P}_T = 0$, а $\bar{\bar{P}}_T = \bar{P}_T \tilde{P}_T P_F P_C$.

Для вызовов обратных тригонометрических функций, имеющих суммарную вероятность появления \tilde{P}_{T-1} , ОД, проявляющийся в вызове неверных функций arcsin, arccos, arctg, не может быть распознан, поэтому $\hat{P}_{T-1} = 0$, а $\bar{\bar{P}}_{T-1} = \bar{P}_{T-1} \tilde{P}_{T-1} P_F P_C$.

Для вызовов иных математических функций, имеющих суммарную вероятность появления \tilde{P}_M , ОД, проявляющийся в вызове неверных функций abs, sign, log, exp, pow, не может быть распознан, поэтому $\hat{P}_M = 0$, а $\bar{\bar{P}}_M = \bar{P}_M \tilde{P}_M P_F P_C$.

Для вызовов системных функций, обеспечивающих работу с памятью, файлами, строками (всего 10 функций), имеющих суммарную вероятность появления \tilde{P}_S , ОД, проявляющийся в вызове неверных функций, не может быть распознан, поэтому $\hat{P}_S = 0$, а $\bar{\bar{P}}_S = \bar{P}_S \tilde{P}_S P_F P_C$.

Для констант, имеющих суммарную вероятность появления \tilde{P}_N , ОД не может быть распознан, поэтому $\hat{P}_N = 0$, а $\bar{\bar{P}}_N = \bar{P}_N \tilde{P}_N P_O$.

Для скалярных переменных, имеющих суммарную вероятность появления \tilde{P}_K , а также неверных операндов, адресуемых индексами массивов или указателями, имеющих суммарную вероятность появления \tilde{P}_Q , ОД могут быть распознаны в случае, если ЦСД имеет значение, отличающееся от необходимого. Искомые вероятности определяются спектром данных, т.е. распределением данных по различным значениям ЦСД, при этом $\sum_{j=1}^{N_D} P_{D_j} = 1$,

где N_D – количество различных ЦСД; P_{D_j} – вероятность появления данных, имеющих j-й ЦСД.

Поэтому вероятность распознавания некорректных использований программных переменных

$$\hat{P}_K = \sum_{j=1}^{N_D} (1 - P_{D_j}) P_{D_j} \bar{P}_K \tilde{P}_K P_O \quad \text{примет значение}$$

$$\hat{P}_K = \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \bar{P}_K \tilde{P}_K P_O, \text{ а вероятность необнару-$$

$$\text{жения дефекта } \bar{\bar{P}}_K = \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) \bar{P}_K \tilde{P}_K P_O.$$

В случае, если исполняемая программа хранится в ПЗУ или ППЗУ, то вероятность распознавания некорректных адресуемых операндов примет значение

$$\hat{P}_Q = \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \bar{P}_Q \tilde{P}_Q P_O, \text{ а вероятность необ-}$$

наружения ОД адресуемых операндов

$$\bar{\bar{P}}_Q = \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) \bar{P}_Q \tilde{P}_Q P_O.$$

В случае же если программа и код хранятся в ОЗУ, то вероятность распознавания некорректных адресуемых операндов примет значение

$$\hat{P}_Q = \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \bar{P}_Q \tilde{P}_Q P_{O\mu},$$

а вероятность необнаружения ОД адресуемых операндов –

$$\bar{\bar{P}}_Q = \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) \bar{P}_Q \tilde{P}_Q P_{O\mu}.$$

Величина μ равна

$$\mu = \frac{W_D}{W_S + W_D}, \quad (2)$$

где W_S , W_D – объемы исполняемого кода и программных данных соответственно.

Найдем эффективность метода целочисленного дескрипторного контроля (ЦДК) семантической корректности ПО как условную вероятность обнаружения ОД при условии их существования. Это позволит оценить достоверность ЦДК как вероятность существования семантических ОД в случае их необнаружения:

$$\eta = \frac{\sum_i \hat{P}_i}{\sum_i \hat{P}_i + \sum_i \bar{\bar{P}}_i} = \frac{\hat{P}}{\hat{P} + \bar{\bar{P}}}, \quad (3)$$

где i – тип элемента программной конструкции; \hat{P}_i и $\bar{\bar{P}}_i$ – абсолютные вероятности соответственно распознавания и нераспознавания ОД; \hat{P} и $\bar{\bar{P}}$ – суммарные вероятности обнаружения и необнаружения.

Вычислим вероятности \hat{P} и $\bar{\bar{P}}$.

$$\hat{P} = P_A P_C \left(\frac{2\bar{P}_+ \tilde{P}_+}{3} + \frac{2\bar{P}_- \tilde{P}_-}{3} + \bar{P}_* \tilde{P}_* + \bar{P}_\gamma \tilde{P}_\gamma \right) + P_O \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\bar{P}_K \tilde{P}_K + \bar{P}_Q \tilde{P}_Q P_{O\mu} \right);$$

$$\begin{aligned} \bar{\bar{P}} = & P_A P_C \left(\frac{\bar{P}_+ \tilde{P}_+}{3} + \frac{\bar{P}_- \tilde{P}_-}{3} \right) + P_C (\bar{P}_L \tilde{P}_L + \bar{P}_J \tilde{P}_J + \bar{P}_R \tilde{P}_R + \\ & + \bar{P}_T \tilde{P}_T P_F + \bar{P}_{T-1} \tilde{P}_{T-1} P_F + \bar{P}_M \tilde{P}_M P_F + \bar{P}_S \tilde{P}_S P_F) + \\ & + P_O \left(\bar{P}_N \tilde{P}_N + \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\bar{P}_K \tilde{P}_K + \bar{P}_Q \tilde{P}_Q \mu \right) \right), \end{aligned}$$

где для программ, хранящихся в ПЗУ, $\mu = 1$, а для программ, хранящихся в ОЗУ, μ соответствует (2).

На основании (1) и приняв, что $\bar{P}_i = \text{const}$, т.е. вероятность существования ОД не зависит от типа элемента, а определяется только частотой появления элементов данного типа в коде (операционным спектром и спектром данных), и обозначив $\bar{P} = \bar{P}_i$, получим, что $P_{\text{def}} = \bar{P} \sum_i \tilde{P}_i$. Тогда

$$\begin{aligned} \hat{P} = & P_A P_C \bar{P} \left(\frac{2\tilde{P}_+}{3} + \frac{2\tilde{P}_-}{3} + \tilde{P}_* + \tilde{P}_\gamma \right) + \\ & + P_O \bar{P} \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\tilde{P}_K + \tilde{P}_Q \mu \right); \\ \bar{\bar{P}} = & P_A P_C \bar{P} \left(\frac{\tilde{P}_+}{3} + \frac{\tilde{P}_-}{3} \right) + P_C \bar{P} (\tilde{P}_L + \tilde{P}_J + \tilde{P}_R + \\ & + P_F (\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S)) + \\ & + P_O \bar{P} \left(\tilde{P}_N + \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\tilde{P}_K + \tilde{P}_Q \mu \right) \right). \end{aligned}$$

В связи с тем, что для определения эффективности метода нам необходимо найти значение выражения (3), то множитель \bar{P} можно опустить в связи с его дальнейшим сокращением:

$$\begin{aligned} \hat{P}' = & P_A P_C \left(\frac{2\tilde{P}_+}{3} + \frac{2\tilde{P}_-}{3} + \tilde{P}_* + \tilde{P}_\gamma \right) + \\ & + P_O \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\tilde{P}_K + \tilde{P}_Q \mu \right); \end{aligned} \quad (4)$$

$$\begin{aligned} \bar{\bar{P}}' = & P_A P_C \left(\frac{\tilde{P}_+}{3} + \frac{\tilde{P}_-}{3} \right) + P_C (\tilde{P}_L + \tilde{P}_J + \tilde{P}_R + \\ & + P_F (\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S)) + \\ & + P_O \left(\tilde{P}_N + \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\tilde{P}_K + \tilde{P}_Q \mu \right) \right). \end{aligned} \quad (5)$$

Так как появление операций (команд) и операндов представляет полную группу событий, то

$$P_C + P_O = 1. \quad (6)$$

На основании средней арности Z выражение (6) преобразуется в $P_C + ZP_C = 1$. Отсюда

$$P_C = \frac{1}{Z+1}, \quad P_O = \frac{Z}{Z+1} \quad (7)$$

После подстановки (7) в (4) – (5) и умножения их на $(Z+1)$, получим:

$$\begin{aligned} \hat{P}'' = & P_A \left(\frac{2\tilde{P}_+}{3} + \frac{2\tilde{P}_-}{3} + \tilde{P}_* + \tilde{P}_\gamma \right) + \\ & + Z \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) \left(\tilde{P}_K + \tilde{P}_Q \mu \right); \end{aligned} \quad (8)$$

$$\begin{aligned} \overline{P}'' = P_A \left(\frac{\tilde{P}_+}{3} + \frac{\tilde{P}_-}{3} \right) + (\tilde{P}_L + \tilde{P}_J + \tilde{P}_R + \\ + P_F(\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S)) + \\ + Z \left(\tilde{P}_N + \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) (\tilde{P}_K + \tilde{P}_Q \mu) \right). \end{aligned} \quad (9)$$

Откуда

$$\begin{aligned} \hat{P}'' + P'' = P_A (\tilde{P}_+ + \tilde{P}_- + \tilde{P}_* + \tilde{P}_/) + \\ + Z \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) (\tilde{P}_K + \tilde{P}_Q \mu) + \\ + \tilde{P}_R + \tilde{P}_L + \tilde{P}_J + P_F(\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S) + \\ + Z \left(\tilde{P}_N + \left(\sum_{j=1}^{N_D} P_{D_j}^2 \right) (\tilde{P}_K + \tilde{P}_Q \mu) \right). \end{aligned}$$

На основании того, что $\tilde{P}_+ + \tilde{P}_- + \tilde{P}_* + \tilde{P}_/ = 1$ (полная группа событий арифметических операций):

$$\begin{aligned} \hat{P}'' + P'' = Z(\tilde{P}_K + \tilde{P}_Q \mu + \tilde{P}_N) + \tilde{P}_R + \\ + \tilde{P}_A + \tilde{P}_L + \tilde{P}_J + P_F(\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S). \end{aligned} \quad (10)$$

Для полной группы событий операндов $\tilde{P}_K + \tilde{P}_Q + \tilde{P}_N = 1$. Введя множитель $(\mu + 1 - \mu)$, получим $\tilde{P}_K + \tilde{P}_Q(\mu + 1 - \mu) + \tilde{P}_N = 1$ или

$$\tilde{P}_K + \tilde{P}_Q \mu + \tilde{P}_N = 1 - \tilde{P}_Q(1 - \mu). \quad (11)$$

Подставив (11) в (10) и учитывая, что если программа хранится в ПЗУ или ППЗУ (характерно для большинства систем критических применений), то $\mu \approx 1$, получим:

$$\begin{aligned} \hat{P}'' + P'' = Z + \tilde{P}_R + \tilde{P}_A + \tilde{P}_L + \tilde{P}_J + \\ + P_F(\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S). \end{aligned}$$

На основании того, что выполняется $\tilde{P}_R + \tilde{P}_A + \tilde{P}_L + \tilde{P}_J + \tilde{P}_F = 1$ (полная группа событий для команд) и $\tilde{P}_T + \tilde{P}_{T-1} + \tilde{P}_M + \tilde{P}_S = 1$:

$$\hat{P}'' + P'' = Z + 1 - P_F + P_F = Z + 1. \quad (12)$$

Из выражения (8) на основании полной группы событий арифметических операций, получим:

$$\hat{P}'' = P_A \left(1 - \frac{\tilde{P}_+}{3} - \frac{\tilde{P}_-}{3} \right) + Z \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right) (\tilde{P}_K + \tilde{P}_Q \mu).$$

Так как

$$\tilde{P}_K + \tilde{P}_Q + \tilde{P}_N = 1$$

и, принимая во внимание, что по результатам СА $\tilde{P}_N \approx 0$, а $\mu = 1$, получим

$$\hat{P}'' = P_A \left(1 - \frac{\tilde{P}_+}{3} - \frac{\tilde{P}_-}{3} \right) + Z \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right). \quad (13)$$

Найдем $\eta = \frac{\hat{P}''}{\hat{P}'' + P''}$, подставив (12) и (13):

$$\eta = \frac{P_A \left(1 - \frac{\tilde{P}_+}{3} - \frac{\tilde{P}_-}{3} \right) + Z \left(1 - \sum_{j=1}^{N_D} P_{D_j}^2 \right)}{Z + P_A \tilde{P}_R + 1}.$$

При условии равномерного распределения данных по семантическому спектру $P_{D_j} = P_D = 1/N_D$, эффективность контроля составляет:

$$\eta = \frac{P_A \left(1 - \frac{\tilde{P}_+}{3} - \frac{\tilde{P}_-}{3} \right) + Z \left(1 - \frac{1}{N_D} \right)}{Z + 1}$$

В случае большого объема адресуемых данных, т.е. $N_D \rightarrow \infty$, получим эффективность метода

$$\eta = \frac{P_A \left(1 - \frac{\tilde{P}_+}{3} - \frac{\tilde{P}_-}{3} \right) + Z}{Z + 1}$$

при ограничениях:

$$\begin{aligned} 0 \leq \tilde{P}_+ \leq 1; 0 \leq \tilde{P}_- \leq 1; 0 \leq \tilde{P}_A \leq 1; \\ \tilde{P}_+ + \tilde{P}_- \leq 1; Z = 1 \dots N. \end{aligned}$$

Принимая $P_A = 0,50$ – доля арифметических команд среди всех команд; $P_R = 0,05$ – доля операций отношения; $Z = 2,1$ – арность операций; $P_+ = 0,39$ – доля сложений в арифметических командах; $P_- = 0,22$ – доля вычитаний среди арифметических команд, получим $\eta = 0,8$. Полученное значение является нижним порогом эффективности метода ЦДК семантической корректности ПО.

Средняя эффективность метода может быть вычислена для вероятностей, лежащих в пределах $0 \dots 1$ и арности операций в диапазоне $1 \dots 3$, при условии, что сумма условных вероятностей элементарных операций не превышает 1.

Средняя эффективность метода составит:

$$\bar{\eta} = \int_0^1 \int_0^1 \int_0^1 \int_0^3 \frac{P_A \left(1 - \frac{\tilde{P}_+}{3} - \frac{\tilde{P}_-}{3} \right) + Z}{1 + Z} dP_A d\tilde{P}_+ d\tilde{P}_- dZ = 0,75.$$

Заключение

Анализ предложенной модели дефектов программного обеспечения показал, что метод контроля корректности вычислительных процессов, основанный на анализе значений ЦСД, имеет среднюю эффективность 0,75. Это означает, что три четверти всех семантических дефектов могут быть обнаруже-

ны в реальном времени при выполнении задач, связанных с управлением сложными техническими объектами, что позволит значительно повысить надежность ИУС.

Направление дальнейших исследований должно быть направлено на повышение достоверности корректности вычислительных процессов, обеспечиваемой повышением диагностики посредством использования нескольких инвариантов ПО.

Литература

1. Харченко В.С. Статистический анализ программного обеспечения системы управления космическим аппаратом и оценка проверяющей способности семантического контроля / В.С. Харченко, Ю.С. Манжос, В.Л. Петрик // *Технология приборостроения*. – 2002. – № 2. – С. 52-59.

2. Манжос Ю.С. Оценка эффективности независимой верификации программного обеспечения / Ю.С. Манжос // *Авиационно-космическая техника и технология*. – 2004. – № 7. – С. 210-214.

3. Петрик В.Л. Экспертиза программного обеспечения информационно-управляющих систем с использованием дескрипторного семантического пространства / В.Л. Петрик // *Радиоэлектронні і комп'ютерні системи*. – 2007. – №2 (21). – С. 29-35.

4. Петрик В.Л. Целочисленное семантическое отображение / В.Л. Петрик // *Радиоэлектронні і комп'ютерні системи*. – 2007. – №3 (22). – С. 73-81.

5. Манжос Ю.С. Дескрипторный контроль программного обеспечения критического применения в реальном времени / Ю.С. Манжос, В.Л. Петрик // *Збірник наукових праць Харківського університету Повітряних Сил*. – 2008. – №1 (16). – С. 90-93.

Поступила в редакцию 2.06.2008

Рецензент: д-р техн. наук, проф., зав. кафедрой программной инженерии И.Б. Туркин, Харьковский национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

МОДЕЛЬ ДЕФЕКТІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В.Л. Петрик, Ю.С. Манжос

Запропонована модель дефектів програмного забезпечення, яка заснована на цілочисельному дескрипторному відображенні. Програмний код розглянутий як впорядкована множина програмних конструкцій, кожна з яких з деякою вірогідністю може містити залишковий програмний дефект. Модель дефектів враховує адресні операції, операції переходу і дозволяє на основі статистичних характеристик коду оцінити долю залишкових дефектів, пов'язаних з порушенням семантичних інваріантів, і ефективність методу, заснованого на контролі цілочисельних семантичних дескрипторів.

Ключові слова: семантичний контроль, семантичний інваріант, цілочисельний семантичний дескриптор, залишковий програмний дефект.

MODEL OF SOFTWARE DEFECTS

V.L. Petrik, Y.S. Manzhos

The model of software defects, based on the integer semantic descriptor imagination, is proposed. A software is considered as an ordered set of the software elementary items. Each items can have a software a latent software defect. The proposed model of software defects takes into account address operations, jump operations and allows on the basis of statistical descriptions of code to estimate the part of the remaining defects related to violation of semantic invariants, and efficiency of the method based on the check of integer semantic descriptors.

Key words: semantic control, semantic invariant, integer semantic descriptor, remaining program defect.

Петрик Валерия Леонидовна – старший преподаватель кафедры экономико-математического моделирования Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина, e-mail: Petrikvl@mail.ru.

Манжос Юрий Семенович – канд. техн. наук, старший преподаватель кафедры программной инженерии Национального аэрокосмического университета им. Н.Е. Жуковского «ХАИ», Харьков, Украина.