10 СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УДК 681.142.2

ГРАФО-АНАЛИТИЧЕСКИЕ МОДЕЛИ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ В САПР

А.Г. Зыков, А.В. Безруков, О.Ф. Немолочнов, В.И. Поляков, А.В. Андронов

Рассматривается применение графо-аналитических моделей для верификации вычислительных процессов в рамках САПР, приводится пример построения комплексного кубического покрытия по графо-аналитической модели ациклического вычислительного процесса, формулируется задача синтеза тестовых наборов на основе кубического покрытия.

Ключевые слова: графо-аналитические модели, верификация вычислительных процессов, кубические покрытия, тестирование.

Введение

Проблема анализа качества аппаратного и программного обеспечения становится сегодня все более острой, особенно по мере расширения использования информационных технологий и нанотехнологий в приборостроении. Экспоненциальный рост сложности аппаратного и программного обеспечения вычислительных процессов порождает повышенные требования к бездефектному проектированию. Известны примеры, как дорого обходятся ошибки, допущенные на различных этапах проектирования аппаратуры, поэтому все современные САПР обязательно снабжаются методологическими, программными и инструментальными средствами анализа разрабатываемого изделия на всех этапах автоматизированного проектирования. Не менее актуальными являются проблемы, связанные с обеспечением проектирования надежных программ. Однако возможности средств верификации сегодня заметно отстают от возможностей систем проектирования и технологии изготовления, поэтому разработка машинно-ориентированных методов верификации аппаратно-программных компонентов вычислительных процессов является актуальной [1]. Одним из направлений формализации верификации вычислительных процессов (ВВП) является использование их описания с помощью графо-аналитических моделей (ГАМ), на основании которого строится комплексное кубическое покрытие и синтезируются тестовые наборы [2].

Ставится задача нахождения допустимых значений переменных в системах неравенств-отношений, определяющих условия-предикаты вычислительных процессов программ.

Построение комплексного покрытия ациклического вычислительного процесса

В работе [3] предложен метод верификации вычислительных процессов на основе алгебро-топологического подхода. Рассмотрим работу метода на примере ГАМ простого интервального процесса (рис. 1).

Описание ГАМ можно готовить и редактировать в текстовом редакторе. В итоге получаем xml-описание ГАМ следующего вида (приведена часть описания):

```
<GAM name=»multiswitch»>
      <!—Уникальное в рамках пространства имен имя модели -->
      <meta>
            <!—Краткое описание ГАМ -->
            <description>Демонстрационная ГАМ для вычислительного процесса с мно-
жественным ветвлением</description>
            </meta>
            <attributes>
                  <!—Атрибуты модели -->
                  <vars>
                  <!—Внешние переменные для ГАМ -->
                        <in>
                        <!—Входные параметры для ГАМ -->
                              <var name=»x»>
                              <!—Уникальное имя в рамках ГАМ -->
                                    <type>int</type>
                                    <!—тип данных -->
                                    <default>0</default>
                                    <!—Значение по-умолчанию -->
                              </var>
                        </in>
                        <out>
                              <var name=″y″>
                                    <!—index определяет порядок переменных -->
                                    <type>float</type>
```

```
                        <default>0</default>
                    </var>
                </out>
        </vars>
        <startUid>1</startUid>
        <!—Начальный элемент ГАМ -->
</attributes>
<nodes>
        <!—Вершины ГАМ -->
        <cv uid=”1”>
        <!—Определение вершины. -->
                <input>In</input>
                <!—Ссылка на предыдущую вершину -->
                <trueOutput>4</trueOutput>
                <!—Ссылка на следующую вершину -->
                <falseOutput>2</falseOutput>
                <condition> x &gt; 5 </condition>
                <!—Условие -->
        </cv>
        <cv uid=”2”>
                <input>1</input>
                <trueOutput>5</trueOutput>
                <falseOutput>3</falseOutput>
                <condition> x &gt; 0</condition>
        </cv>
```

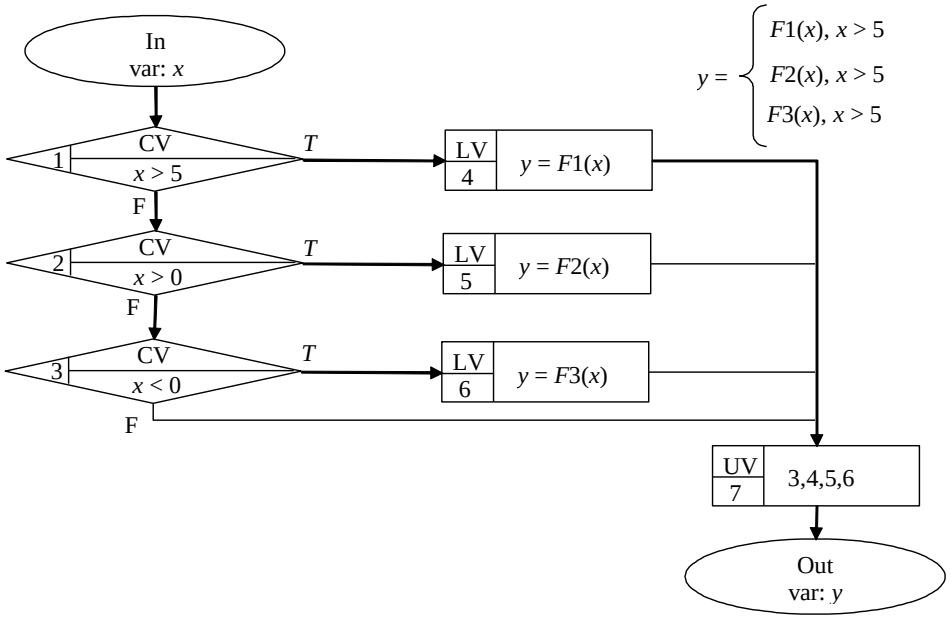

Рис. 1. ГАМ вычислительного процесса для интервальной функции

Далее xml-описание ГАМ необходимо преобразовать в машинно-ориентированный вид. Для машинной обработки были разработаны специальные списочные структуры, представляющие ГАМ. Элементы этих структур изображены на рис. 2. После преобразования происходит поиск всех возможных кубов покрытия по алгоритму, представленному на рис. 3. Результат построения покрытия приведен в таблице (символ × обозначает произвольное значение из области определения предиката или переменной), где **Y** – предыдущее значение, а **Y'** – последующее значение переменной *y*.

На следующем этапе происходит определение значений входных переменных для того, чтобы процесс выполнялся по определенной ветви. Для этого, в общем случае, нужно найти хотя бы одно решение системы неравенств, задаваемой логическими выражениями в условных вершинах ГАМ. В общем виде такая система выглядит следующим образом:

$$\begin{cases} F_0(a_{01}, a_{02}, a_{03}, \ldots, a_{0i}) \\ F_1\ (a_{11}, a_{12}, a_{13}, \ldots, a_{1i}) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ F_k(a_{k1}, a_{k2}, a_{k3}, \ldots, a_{ki}) \end{cases},$$

где F_k – логическое выражение для соответствующей условной вершины с номером k на проверяемой ветви, a_{ki} – значение переменной a_i на этом этапе.

В рассматриваемой ГАМ для пути 1 – 4 – 7 (номера вершин) система примет вид

$\{x > 5$.

Для пути 1 – 2 – 5 – 7 получим

$$\begin{cases} x > 0 \\ x \le 5. \end{cases}$$

Аналогично определяются системы отношений-неравенств и для остальных кубов покрытия.

Условная вершина:

Тип: CV	Входы	Переменные УП	Выход	
	z_k	$r_1, r_2, \ldots, r_i$	z_j^T	z_j^F
Имя или номер вершины	AD_k	$AD_1, AD_2, \ldots, AD_i$ / FR УП	AD_j^T	AD_j^F

Линейная вершина:

Тип: LV	Входы	Переменные УП	Выход
	z_k	$r_1, r_2, \ldots, r_i$	z_j
Имя или номер вершины	AD_k	$AD_1, AD_2, \ldots, AD_i$	AD_j

Объединяющая вершина:

Тип: UV	Входы	Выход
	$z_1, z_2, \ldots, z_k$	z_j
Имя или номер вершины	AD_k	AD_j

Рис. 2. Элементы списка машинного описания ГАМ

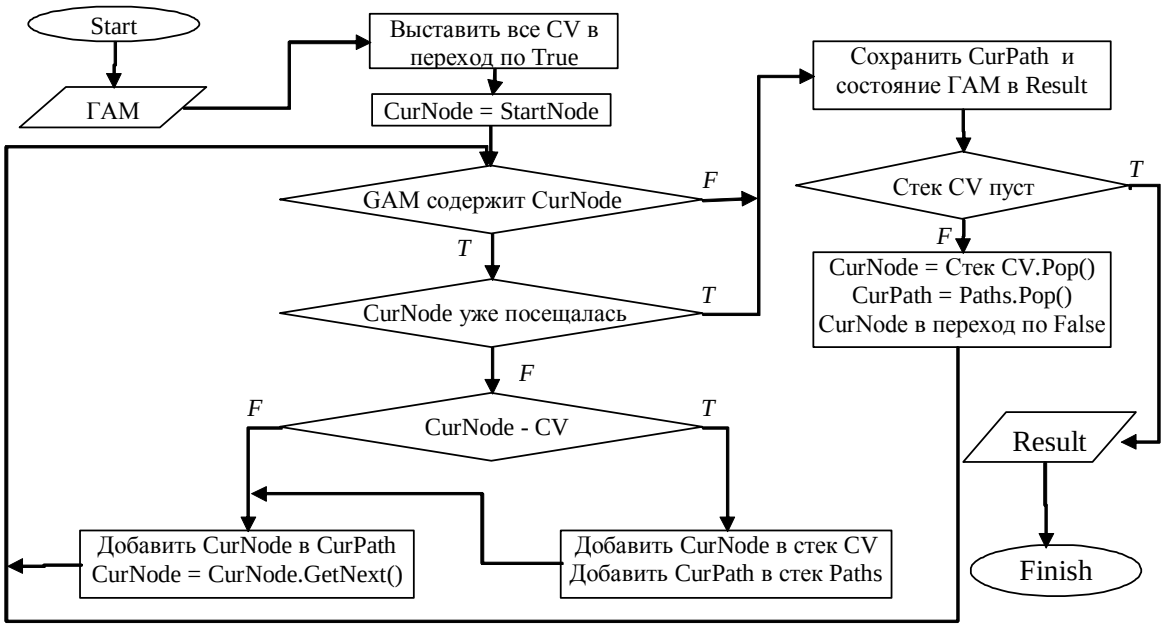

Рис. 3. Алгоритм построения кубического покрытия ГАМ

CV1	CV2	CV3	Y	Y’
0	**0**	**0**	×	×
0	**0**	**1**	×	$F3(x)$
0	**1**	×	×	$F2(x)$
1	×	×	×	$F1(x)$

Таблица. Кубическое покрытие ГАМ рис. 1

Для определения значений переменных, участвующих в условных выражениях, сформулируем следующие правила.

1. Достаточно найти только одно решение для совокупности переменных условия-предиката, нет необходимости решать систему со всей математической строгостью.
2. Не все переменные будут участвовать в каждом выражении, а многие будут участвовать в виде константы, что упрощает решение системы.
3. Во многих случаях систему можно будет разделить на части, так как нередко вершины не связаны по данным.
4. Часть ветвей программы будут в принципе недостижимы ни при каких ситуациях (don't care), некоторые из этих случаев можно определять без решения системы в целом.

На последнем этапе с помощью синтезированных тестов проводится тестирование программного обеспечения (ПО), на основании которого делается вывод о соответствии или несоответствии программы и ее спецификации.

Для реализации и исследования предложенного метода синтеза контролирующих тестов и дальнейших научно-исследовательских работ в области верификации вычислительных процессов разработана исследовательская САПР верификации вычислительных процессов (ИСАПР ВВП).

Общая структура исследовательской САПР ВВП

Итерационно-рекурсивная модель вычислительного процесса программ, предложенная в [4], является теоретической основой для разработки такой САПР, а в работе [5] сформулированы основные цели, которые преследуются при ее создании. С учетом этих положений предлагается следующая структура ИСАПР (рис. 4).

Ядро системы представляет собой набор системных и пользовательских плагинов, взаимодействующих друг с другом через единый интерфейс. Взаимодействие с хранилищем ведется через отдельный (общий для САПР) интерфейс, что позволяет унифицировать способ хранения и доступа к данным и применить любую доступную систему управления базами данных. Система для выполнения своей учебной функции также должна предоставлять три типа пользовательских интерфейса.

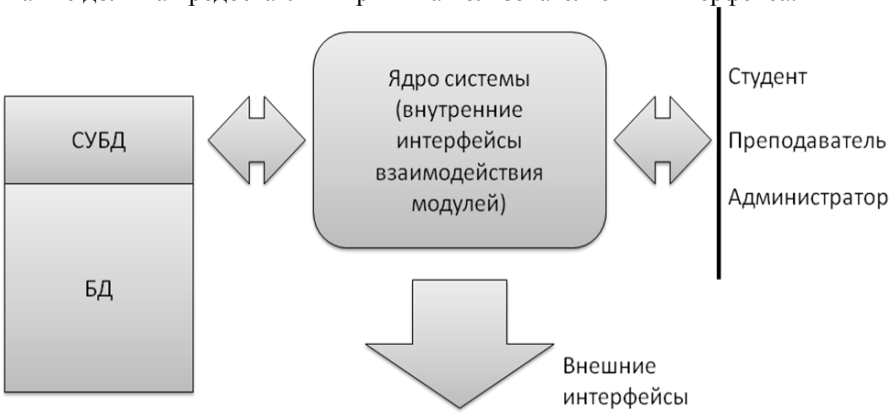

Рис. 4. Структурная схема ИСАПР ВВП

Под внешними интерфейсами подразумеваются способы обмена данными с информационными системами университета для более полной интеграции ИСАПР в учебный процесс.

Система разрабатывается как достаточно универсальная учебно-исследовательская среда, без привязки к определенному типу задач, но, так как в рамках работы рассматривается применение графо-аналитических моделей для верификации ПО, то к реализации предлагается следующая схема взаимодействия модулей (рис. 5).

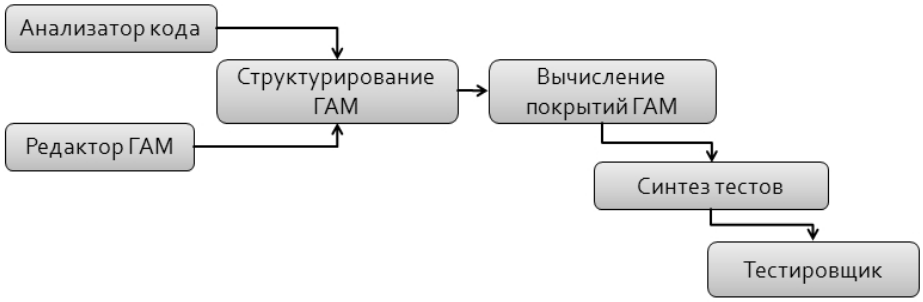

Рис. 5. Модули учебно-исследовательской САПР верификации на основе ГАМ

Данное разделение на модули, очевидно, вытекает из схемы применения метода, представленной в [2]. Назначение модулей системы верификации следующее:

- модуль «Анализатор кода» восстанавливает ГАМ на основе скомпилированной программы;
- модуль «Редактор ГАМ» позволяет строить модели вручную на основе спецификации программы;
- модуль «Структурирование ГАМ» упрощает полученные модели, выделяя в машинном представлении ГАМ структурные единицы (что особенно важно для моделей, полученных от анализатора) [6];
- назначение модулей «Вычисление покрытий ГАМ» [7] и «Синтез тестов» очевидно из их названий;
- модуль «Тестировщик» создает необходимое для тестирования окружение и выводит результаты тестирования.

«Редактор ГАМ» работает только при непосредственном участии пользователя, остальные модули могут использоваться как в автоматическом, так и полуавтоматическом режиме.

Заключение

Использование предложенной схемы верификации на основе графо-аналитических моделей позволяет наглядно (на уровне логической структуры и, как следствие, интерфейса) продемонстрировать весь процесс верификации, а предложенная структура определяет проектирование и разработку исследовательской САПР верификации вычислительных процессов, позволяет сделать разработку отдельных модулей во многом независимой.

Литература

1. Bruce Wile, John C. Goss. Wolfgang Roesner. Comprehensive Functional Verification: The Complete Indastry Cycle. – San Francisco, 2005. – 676 с.
2. Зыков А.Г., Немолочнов О.Ф., Поляков В.И., Безруков А.В., Кузьмин В.В. Графо-аналитические модели как средство верификации вычислительных процессов // Труды Международного конгресса по интеллектуальным системам и информационным технологиям. Научное издание в 4-х томах. – М.: Физматлит, 2010. – Т. 2. – 400 с.
3. Немолочнов О.Ф., Зыков А.Г., Лаздин А.В., Поляков В.И. Верификация в исследовательских, учебных и промышленных системах // Научно-технический вестник СПб ГИТМО (ТУ). – 2003. – Вып. 11. – С. 146–151.
4. Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Осовецкий Л.Г., Сидоров А.В., Кулагин В.С. Итерационно-рекурсивная модель вычислительных процессов программ // Изв. вузов. Приборостроение. – 2005. – Т. 48. – № 12. – С. 14–20.
5. Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Петров К.В. Учебно-исследовательская САПР верификации и тестирования вычислительных процессов программ // Научно-технический вестник СПбГУ ИТМО. – 2006. – Вып. 32. – С. 127–128.
6. Немолочнов О.Ф., Зыков А.Г., Поляков В.И., Сидоров А.В. Структурирование программ и вычислительных процессов на множество линейных и условных вершин // Научно-технический вестник СПбГУ ИТМО. – 2005. – Вып. 19. – С. 207–212.
7. Немолочнов О.Ф., Зыков А.Г., Поляков В.И. Кубические покрытия логических условий вычислительных процессов и программ // Научно-технический вестник СПбГУ ИТМО. – 2004. – Вып. 14.– С. 225–233.

Зыков Анатолий Геннадьевич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, zykov_a_g@mail.ru

Безруков Александр Владимирович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, versus1945@list.ru

Немолочнов Олег Фомич – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, зав. кафедрой, nemolochnov_o_f@mail.ru

Поляков Владимир Иванович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кандидат технических наук, доцент, v_i_polyakov@mail.ru

Андронов Алексей Викторович – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, starsabre@mail.ru