

Опубликовано в тезисах конференции «Software Engineering Conference in Russia (SECR-2005)» .– М.: РУССОФТ, 2005.– с. 12–14.

*Корнеев Г.А., Шалыто А.А.*

## **Построение системы конечных автоматов по программному коду**

Санкт-Петербургский государственный университет информационных технологий, механики и оптики, Санкт-Петербург  
[kgeorgiy@rain.ifmo.ru](mailto:kgeorgiy@rain.ifmo.ru), [shalyto@mail.ifmo.ru](mailto:shalyto@mail.ifmo.ru)

В рамках развития автоматного программирования [1] встает вопрос о формальных методах построения конечных автоматов или их систем по программам на императивных языках программирования. В работе [2] был предложен метод преобразования программ, не содержащих рекурсии. В работе [3] этот метод был развит, что позволило преобразовывать программы с явной рекурсией.

Отметим, что в области аппаратного обеспечения эта задача рассматривалась в 70-х годах и ее решение для одной процедуры приведено в работе [4].

Во всех указанных работах рассматривались программы, состоящие из одной процедуры и не рассматривалась задача трассировки (пошагового исполнения) в обратном направлении, что является важным в некоторых приложениях [5].

В работе [6] при участии авторов был предложен метод преобразования программы из произвольного числа рекурсивных процедур в систему взаимодействующих автоматов. Получаемая система автоматов обеспечивает трассировку исходной программы, как в прямом, так и в обратном направлении.

В настоящей работе предлагается основанный на результатах статьи [6] формальный метод построения такой системы автоматов по программе. Для системы автоматов, построенной в результате применения предлагаемого метода, доказываемся, что трассировка в обоих направлениях осуществляется корректно.

Предлагаемый метод позволяет преобразовывать программы, состоящие из блочных операторов, операторов цикла с предусловием, полного и укороченного ветвления, присваивания и вызова процедуры. Язык поддерживающий такой набор конструкций является достаточно богатым для описания произвольной программы [7] и позволяет описывать рекурсивные программы.

В предлагаемом методе по программе строится система взаимодействующих автоматов, содержащая по два автомата для каждой процедуры: *прямой* (для трассировки вперед) и *обратный* (для

трассировки назад). Автоматы, построенные по одной процедуре, имеют общее множество состояний, но разные функции переходов [6].

Автоматы строятся рекурсивно на основе дерева вывода преобразуемой программы. При преобразовании вершины дерева вывода программы во фрагменты автоматов, к уже построенным фрагментам добавляется не более двух состояний и четырех переходов. Таким образом, суммарное количество состояний и переходов в полученной системе автоматов линейно по количеству вершин в дереве вывода, а, следовательно, и по числу операторов в исходной программе.

Для обеспечения трассировки назад создаваемая система автоматов использует стек, в котором сохраняются данные, требующиеся для реализации обратной трассировки.

Для систем конечных автоматов, построенных в соответствии с предлагаемым методом доказываются следующие свойства:

- *адекватность* — система автоматов выполняет те же действия, что и исходная программа;
- *обратимость* — при выполнении действий обратного автомата будут восстановлены исходные значения всех переменных, при условии, что в точке входа они были такими же, как и в точке выхода при прямом проходе;
- *полнота* — для каждого состояния, не являющегося конечным, при любых значениях переменных условие на одном из переходов должно быть истинно;
- *непротиворечивость* — условия на переходах из одного состояния не могут быть истинными одновременно;
- *отсутствие недостижимых состояний* — любое состояние может быть достигнуто по переходам из начального состояния.

При этом состояние называется достижимым, если оно начальное или в него существует переход из достижимого состояния. Таким образом, проверяется только принципиальная возможность достижения состояний, а условия на переходах фактически игнорируются.

Доказательство *адекватности* и *обратимости* позволяет утверждать, что трассировка в обоих направлениях осуществляется корректно. *Полнота* и *непротиворечивость* обеспечивают детерминированность построенных автоматов. Данные свойства доказываются посредством структурной индукции на дереве вывода преобразуемой программы.

Предложенный метод является математической основой создания пакета *Vizi*, предназначенного для создания визуализаторов алгоритмов, используемых при обучении основам программирования и дискретной математики, опубликованные на сайте <http://is.ifmo.ru>.

Работа выполнена в рамках совместной лаборатории “СПбГУ ИТМО — Vorland” и при поддержке РФФИ по гранту №05-07-90011.

1. *Шалыто А. А.* SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998.
2. *Шалыто А.А., Туккель Н. И.* Преобразование итеративных алгоритмов в автоматные // Программирование. 2002. № 5. с.12-26.
3. *Туккель Н. И., Шалыто А. А., Шамгунов Н. Н.* Реализация рекурсивных алгоритмов на основе автоматного подхода // Телекоммуникации и информатизация образования. 2002. № 5.
4. *Баранов С. И.* Синтез микропрограммных автоматов (граф-схемы и автоматы). Л.: Энергия, 1979.
5. *Казаков М. А., Столяр С. Е.* Визуализаторы алгоритмов как элемент технологии преподавания дискретной математики и программирования // Международная научно-методическая конференция “Телематика-2000”. СПб.: 2000. с.189-191.
6. *Казаков М. А., Корнеев Г. А., Шалыто А. А.* Метод построения логики работы визуализатора алгоритмов на основе конечных автоматов // Телекоммуникации и информатизация образования. 2003. №6. с. 27-58.
7. *Грис Д.* Наука программирования. М.: Мир, 1984.