

Горбаченко И.М. Создание тестирующих систем с помощью автоматного программирования. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей IX Междунар. научно-техн. конф. – Пенза: ПДЗ, 2009. – С. 98-101.

СОЗДАНИЕ Тестирующих СИСТЕМ С ПОМОЩЬЮ АВТОМАТНОГО ПРОГРАММИРОВАНИЯ

И.М. Горбаченко

Сибирский государственный технологический университет,
г. Красноярск, Россия

Предлагается использовать автоматное программирование для создания тестирующих систем. Суть автоматного программирования заключена в выявлении набора состояний описываемого процесса, условий перехода между этими состояниями и выявлении действий, которые должны выполняться в том или ином состоянии.

Gorbachenko I.M. The development of testing systems with the help of finite state programming.

The use of finite state programming for the development of testing systems is suggested. The essence of finite state programming consists on revealing of the conditions of a described process, terms of transition between these states and revealing the actions which must be performed in a proper condition.

На современном этапе развития техники все большее применение находят средства компьютерного тестирования для получения оценки результатов обучения. Поэтому моделирование процесса тестирования является актуальной задачей, поскольку позволяет адекватно оценить как сам процесс, так и тестовые задания, которые служат для проверки знаний.

Для моделирования могут применяться разные методы и методологии: сети Петри, методологии IDEF0, ARIS Toolset и др. В том числе могут применяться приемы автоматного программирования.

Суть автоматного программирования заключена в выявлении набора состояний описываемого процесса, условий перехода между этими состояниями и выявлении действий, которые должны выполняться в том или ином состоянии. В автоматном программировании автомат (иногда называют конечным автоматом) реагирует на входное воздействие не только сменой состояния, но и формированием определенных значений на выходах.

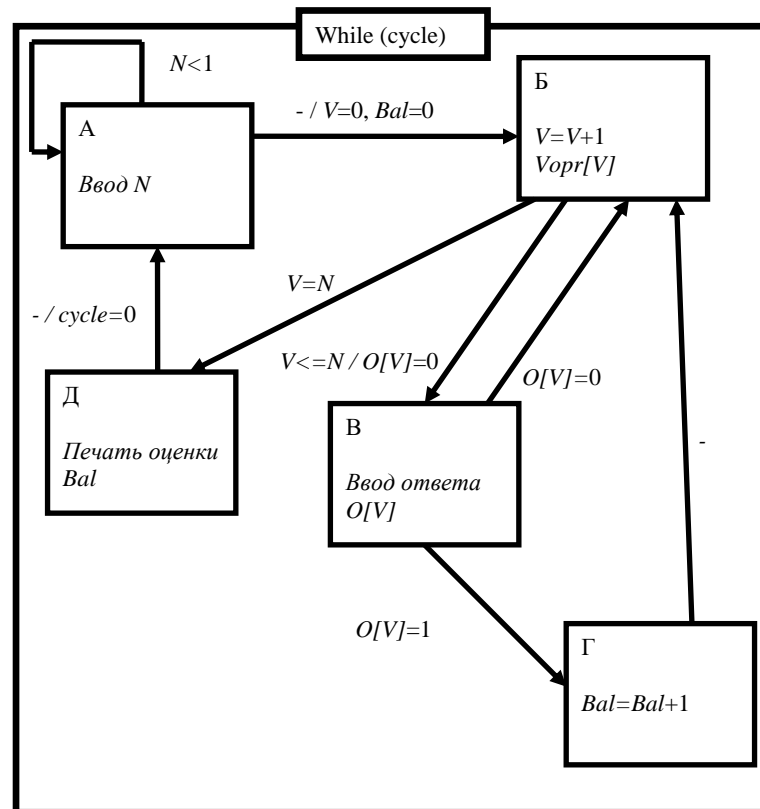
Для описания таких процессов могут применяться автоматы Мура или Мили [1].

При тестировании можно выделить следующие состояния:

- начало тестирования (состояние А),
- выбор вопроса (состояние Б),
- ответ на вопрос (состояние В),
- проверка ответа на вопрос (состояние Г),
- выставление оценки (состояние Д).

При таком распределении начальной вершиной будет являться состояние «начало тестирования», выходом – «выставление оценки». В остальных вершинах будут располагаться распознаватели и соединения.

Графическое представление автомата, описывающего процесс тестирования, приведено на рисунке.



Автомат, отображающий процесс тестирования

Процесс тестирования циклический, так как он повторяется с некоторой периодичностью. Повторение тестирования может быть через разный промежуток времени – через месяц, семестр, учебный год и т.д. Поэтому в обозначение процесса тестирования введено While (от одного из возможных описаний цикла на реальном языке программирования, например, СИ, Паскаль, Ассемблер).

При начале тестирования надо определить число тестовых заданий N (процесс А), при этом число заданий должно быть больше 0. Об этой проверке свидетельствует возвратная стрелка на процесс А.

После определения числа необходимых заданий выполняются следующие действия: число заданных тестовых заданий V определяется равным 0, и оценка Bal (выраженная числом правильных ответов) тоже равна 0.

Затем задается очередной вопрос $Vopr[V]$ и число заданных тестовых заданий V увеличивается на 1 (процесс Б). Если число заданных тестовых заданий V не достигло требуемого числа N (т.е. $V < N$), то выполняется действие: ответ на очередное тестовое задание $O[V]$ приравнивается к 0 (т.е. ответ на тестовое задание считается неверным или на тестовое задание нет ответа), и идет процесс ввода ответа на очередное задание (процесс В).

Если $O[V]=1$ (т.е. ответ на тестовое задание правильный), то число правильных ответов увеличивается на 1 (процесс Г). После этого происходит переход к процессу Б (т.е. определению очередного тестового задания).

Если $O[V]=0$ (т.е. ответ на тестовое задание неправильный), то происходит переход к процессу Б (т.е. определению очередного тестового задания).

Если число заданных тестовых заданий V достигло требуемого числа N (т.е. $V=N$), то идет процесс вывода оценки, т.е. значения переменной Bal (процесс Д). Затем идет обнуление значений переменных ($cycle=0$), и тестирование можно выполнять следующий раз.

Подобным образом можно описать практически любой процесс, который повторяется периодически. При этом каждое состояние может, в свою очередь, представлять автомат.

Особенностью предлагаемого подхода к описанию циклически повторяющихся процессов является то, что повышается централизация логики за счет устранения ее в обработчиках событий и формирования системы взаимосвязанных автоматов, которые вызываются из обработчиков [2].

Кроме того, пошаговое выполнение действий при функционировании автомата позволяют проводить протоколирование работы автомата. Создание протокола при этом может быть произведено автоматически и использовано для решения задач большой размерности при сложной логике процесса.

Запись протокола облегчает программирование поведения таких динамических систем, причем программирование может вестись на любом языке. В частности, программы можно писать на языке С.

При таком подходе автомат описывается статически: его структура закодирована в тексте программы. Это свойство обеспечивает требуемое быстродействие работы программы. Однако остается другая проблема: экспоненциальный рост описания автомата с увеличением числа входных переменных. Тем не менее применение автоматного подхода к проектированию систем (в том числе программ) позволит повысить их качество.

Библиографический список

1. Непейвода Н.Н. Стили и методы программирования : курс лекций : учеб. пособие для студентов вузов, обучающихся в области информ. технологий. – М. : Интернет-ун-т информ. технологий, 2005. – 320 с. (Серия «Основы информационных технологий»).
2. Шалыто А. Технология автоматного программирования [Электрон. ресурс] / режим доступа: <http://www.is.ifmo.ru>