О ВЕРИФИКАЦИИ ПРОГРАММ СО СЛОЖНЫМ ПОВЕДЕНИЕМ

В.И. Ульянцев, А.А. Шалыто

Ульянцев Владимир Игоревич
Шалыто Анатолий Абрамович

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Санкт-Петербург
Тел.: +7 (904) 646-64-02
E-mail: ulyantsev@rain.ifmo.ru

Секция – Информационные технологии

Программные комплексы, требующие высокого уровня надежности, обычно представляют собой системы со сложным поведением [1] (поведение зависит от предыстории). Цена ошибки в таких системах может быть очень высока [2]. Одним из подходов к созданию надежного программного обеспечения (ПО) является автоматное программирование [1]. В рамках парадигмы автоматного программирования ключевыми компонентами ПО являются конечные автоматы.

Среди преимуществ конечных автоматов можно отметить наглядность их представления в графической форме и возможность их *формальной верификации* [3], которое может дополнять тестирование, применяемое обычно при отладке программы. Однако, как заметил Э. Дейкстра, если при тестировании ошибки в программе не найдены, это еще не значит, что их там нет. Покажем на примере «простой» системы со сложным поведением, что не только тестирование не гарантирует отсутствие ошибок, но и верификация, проведенная после тестирования.

В [1] авторами вручную был построен автомат управления часами с будильником, приведенный на рисунке. Данный автомат со сложным поведением является «простым», так как содержит три управляющих состояния. Переходы автомата зависят от четырех событий (A, H, M, T), двух переменных (x_1, x_2) и содержат семь выходных воздействий ($z_1 - z_7$).

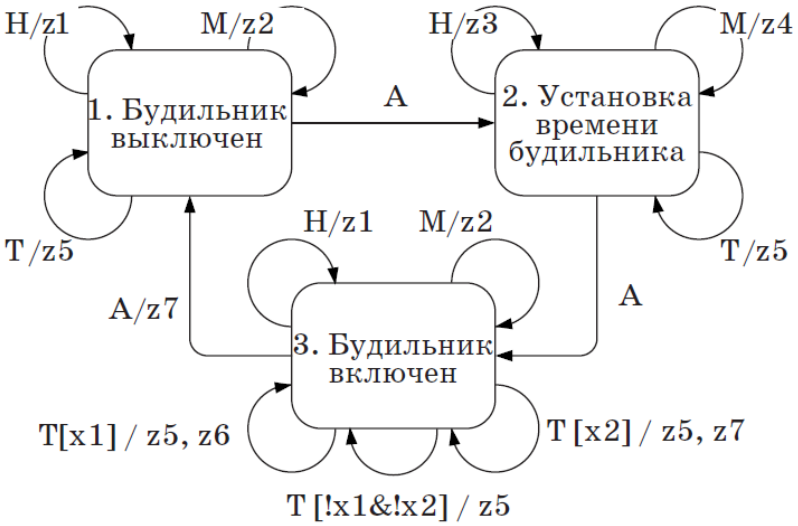

Рисунок. Автомат управления часами с будильником

Ни авторы, ни читатели при прочтении [1] не обнаружили некорректности поведения приведенного автомата. После этого, была выполнена проверка корректности

рассмотренного автомата при помощи тестов. В работе [4] приведены 38 тестов, которые могут быть использованы для тестирования этого автомата (автомат «простой», а тестов уже много). «Корректность» этого набора тестов подтверждается тем, что в данной работе по этим тестам с помощью генетического алгоритма был построен автомат, изоморфный приведенному на рисунке.

Однако, генерация автомата только по набору тестов не всегда может обеспечить желаемое его поведение. Поэтому, в работе [5] было предложено использовать генетические алгоритмы совместно с верификацией. При этом кроме указанных выше тестов записываются темпоральные свойства (например, *LTL*-формулы), которым должен соответствовать искомый автомат, а в ходе его генерации с помощью генетического алгоритма эти свойства проверяются. Если автомат удается построить, то он гарантированно удовлетворяет и заданным тестам, и темпоральным свойствам.

В работе [5] автомат управления часами с будильником строился по тем же 38 тестам, а для верификации в ходе работы генетического алгоритма использовалось 11 темпоральных свойств. Казалось бы, для автомата всего с тремя состояниями и 14 переходами используется достаточно «мощная» система проверки. Однако и ее оказалось недостаточно.

Как и набор тестов, так и набор темпоральных свойств может быть неполным, и бывают случаи, когда при их расширении может быть обнаружена некорректность поведения формально построенного автомата. В работе [6] приведены пять темпоральных свойств, отличных от указанных выше. При этом на четырех из них автомат ведет себя корректно, но, неожиданно для авторов, пятое свойство «после того, как был включен звук будильника, он когда-нибудь будет выключен» не выполнилось – существует такая последовательность событий, при которой будильник будет звучать всегда. Поясним это. Если будильник звонит, то можно нажать кнопку *M* два или более раз. Тем самым будильник будет звонить до тех пор, пока не будет нажата кнопка *A* или пока не произойдет выходное воздействие *z*7, чего может не произойти никогда.

Для того, чтобы обеспечить выполнение этого свойства, необходимо переходы по событиям *M* и *H* из третьего состояния «Будильник включен» дополнить выходным воздействием *z*7. Авторы надеются, что после этого автомат, наконец, будет вести себя корректно.

Из изложенного выше следует, что при сложном поведении даже простые автоматы требуют сложной проверки.

Источники

1. Поликарпова Н. И., Шалыто А. А. Автоматное программирование. СПб.: Питер, 2009. http://is.ifmo.ru/books/_book.pdf.
2. Риган П., Хемилтон С. NASA: миссия надежна // Открытые системы. 2004. № 3. С. 12–17. http://www.osp.ru/text/302/184060.html.
3. Вельдер С. Э., Лукин М. А., Шалыто А. А., Яминов Б. Р. Верификация автоматных программ. СПб.: Наука. 2011. – 241 с. http://is.ifmo.ru/verification/velder_verification_posobie_nauka.pdf.
4. Царев Ф. Н. Методы построения конечных автоматов на основе эволюционных алгоритмов. Диссертация на соискание ученой степени кандидата технических наук. НИУ ИТМО. 2012. http://is.ifmo.ru/disser/tsarev_disser.pdf.
5. Егоров К. В., Царев Ф. Н., Шалыто А. А. Применение генетического программирования для построения автоматов управления системами со сложным поведением на основе обучающих примеров и спецификации // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2010. № 5 (69), с. 81–86.
6. Ульянцев В. И. Отчет о верификации программы управления часами с будильником. НИУ ИТМО. 2012. http://is.ifmo.ru/verification/2013/alarm_clock_verification.pdf