Опубликовано в материалах 2-й межвузовской научной конференции по проблемам информатики СПИСОК-2011, с. 373-374.

М. А. Лукин
Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Разработка и верификация многопоточных автоматных программ

Предлагаемый подход объединяет разработку и верификацию многопоточных автоматных [1, 2] программ в одном инструментальном средстве.

В автоматном программировании Автоматная программа состоит из *источников событий* для конечных автоматов, *системы конечных автомато*в и *объектов управления*. Источником событий может быть также и внешняя для программы среда.

Каждый поток управляется автоматом либо системой вложенных автоматов. Для обеспечения синхронизации потоков конечные автоматы отправляют друг другу события (то есть, одни конечные автомат становятся источниками событий для других конечных автоматов).

Верификация программ общего вида (в части построения модели) практически не может быть автоматизирована. Для автоматных программ эта задача решаема. Поэтому в настоящей работе также ставится задача разработки метода автоматической верификации автоматных программ. Наиболее известным верификатором является верификатор *SPIN* [3], который является открытым и бесплатным. Кроме того, *SPIN* был разработан для верификации многопоточных алгоритмов. При верификации [4, 5] на его основе требования к

программе записываются на языке линейной темпоральной логики (*LTL*) [6]. Инверсии каждой *LTL*-формулы может быть сопоставлен автомат *Бюхи* [7]. Собственно верификация состоит в том, что верификатор с целью построения контрпримера (если он имеется) должен «пересечь» модель *Крипке* [4] и автомат *Бюхи*. *М*одель *Крипке* автоматически строится верификатором по модели программы, записанной на языке *Promela*.

Для визуальных автоматных программ, во-первых, построение модели на указанном языке по графам переходов может быть автоматизировано, а, во-вторых, переход от контрпримера на модели к контрпримеру в терминах автоматов также может быть автоматизирован.

Отличительной особенностью предлагаемого метода является возможность верифицировать параллельные программы.

Для поддержки метода было создано инструментальное средство Stater, которое позволяет разрабатывать автоматные программы, включая генерацию кода на разных языках программирования и верификацию. Для инструментального средства был разработан удобный интерфейс, подсвечивающий контрпример. Кроме того, Stater позволяет создавать автоматизированные классы [8], которые удобно встраиваются в уже существующие проекты.

Литература

1. *Шалыто А.А.* Технология автоматного программирования. http://is.ifmo.ru/works/tech_aut_prog/
2. *Туккель Н.И., Шалыто А.А.* Программирование с явным выделением состояний // *Мир ПК*. — 2001. — № 9. — С. 132—138.
3. *SPIN home page*. http://SPINroot.com

4. *Кларк Э., Грамберг О., Пелед Д.* Верификация моделей программ: Model Checking. М.: МЦНМО, 2002.
5. *Лифшиц Ю.* Верификация программ и темпоральные логики. Лекция №3 курса «Современные задачи теоретической информатики». http://download.yandex.ru/class/lifshits/lecture-note03.pdf
6. *Linear temporal logic.* http://en.wikipedia.org/wiki/Linear_temporal_logic
7. *Büchi automaton.* http://en.wikipedia.org/wiki/Büchi_automaton
8. Поликарпова Н.И., Шалыто А.А. Автоматное программирование. СПб.: Питер, 2009.