

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ



Национальный исследовательский
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Т Р У Д Ы
ПЯТНАДЦАТОЙ
ВСЕРОССИЙСКОЙ КОНФЕРЕНЦИИ
СТУДЕНЧЕСКИХ
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ
ИНКУБАТОРОВ

Томск, 17–19 мая 2018 г.



ТОМСК
«Издательство НТЛ»
2018

Конвертация μ CRL-описания автомата в формат kitidis

Н.В. Кан

Томский государственный университет, г. Томск, Россия

Данная работа посвящена созданию программы, преобразующей описание автомата на некотором языке (языке системы BALM-II [1], языке системы UPPAAL [2], языке μ CRL2 [3] и т.д.) в программу на языке C++, имитирующей поведение автомата, либо преобразующей это описание к формату kitidis [4]. В первом случае полученная программа будет использована в экспериментах, например для задач распознавания автоматов. Во втором – полученное описание автомата в формате kitidis найдёт своё применение в задачах анализа систем, построения композиций, тестов и т.д. с помощью инструментов, входящих в пакет прикладных программ FSMTest.1.0 [4]. Создаваемую программу назвали FSMConvert.

Первым этапом работы над FSMConvert стал обзор и анализ языков описания автоматов. Были рассмотрены следующие языки: язык системы BALM-II, язык системы UPPAAL и язык μ CRL2. В первую очередь было решено реализовать поддержку языка μ CRL2, так как для него есть развитая документация, имеется инструмент mCRL2 [3] для работы с системами, описанными на этом языке, а главное – есть библиотека на языке C++, распространяемая свободно, и ее можно использовать для частичного достижения поставленной цели.

Язык μ CRL2 основан на алгебре взаимодействующих процессов (Algebra of Communicating Processes, ACP). Как и в любой другой алгебре процессов, фундаментальным понятием в μ CRL2 является процесс. Процессы выполняют действия, меняют свои состояния. Их можно композировать специальным образом, для формирования новых процессов. Большие системы обычно состоят из нескольких процессов (или компонентов), работающих параллельно.

Описательные возможности языка μ CRL2 таковы, что с его помощью можно задавать системы, которые не всегда представимы в виде конечных автоматов. В частности, в языке нет явного разделения действий на входные и выходные. Подобное разделение действий необходимо делать специальным образом. Для этого было предложено ввести дополнительный файл, в котором указывается, какие действия являются входными, а какие выходными.

Кроме того, требуется, чтобы описанная система обладала свойством «автоматности»: входные и выходные действия должны обязательно перемежаться (входное действие, выходное действие, входное действие, выходное действие и т.д.). Для проверки того, что представленное описание на языке μCRL2 задаёт именно такую систему, используются результаты работы А.В. Лапутенко [5].

Преобразование μCRL2 -описания некоторой системы в конечный автомат в формате kitidis (с учётом выполнения указанных выше требований) с помощью инструмента FSMConvert происходит следующим образом. Сначала пользователь создаёт описание системы на языке μCRL2 . Далее он разделяет действия системы на входные и выходные и, пользуясь указаниями из работы [5], проверяет, что система является конечным автоматом. Затем, прибегнув к инструментам из набора mCRL2, строит помеченную систему переходов (Labelled Transition System, LTS). И наконец, пользователь запускает инструмент FSMConvert, подавая на вход полученную LTS и файл с указаниями, какие действия считаются входными, а какие выходными. На выходе получается файл с описанием автомата в формате kitidis либо программа на языке C++, в зависимости от желания пользователя.

Алгоритм преобразования заданной LTS в автомат, который реализован в инструменте FSMConvert, таков:

Начальное состояние LTS помещается в очередь состояний для обработки.

На шаге цикла выбирается очередное состояние из очереди. Оно объявляется его текущим. Далее строится дерево достижимости состояний LTS глубины 2, начиная из текущего состояния. Каждая ветка, ведущая из корня дерева в лист, описывает переход конечного автомата из текущего состояния в состояние, хранящееся в листе. Пометки этого пути образуют входо-выходную пару перехода автомата. Этот переход заносится в множество переходов автомата, а листовые вершины помещает в конец очереди состояний для обработки, если это состояние ранее в очереди не встречалось. Цикл заканчивается по исчерпанию очереди состояний для обработки.

Полученный автомат записывается в файл либо по нему строится программа на языке C++ методами, изложенными в работе [6]. Поскольку изначально мы проверили, что заданное μCRL2 -описание является конечным автоматом (количество состояний конечно, действия перемежаются в нужном порядке: вход, выход, вход, выход и т.д.), то описанный выше алгоритм построения автомата по LTS обязательно

закончит работу за конечное время. Во время работы он не встретит двусмысленных ситуаций, а по его завершению получится конечный автомат.

В дальнейшем планируется реализовать подобным же образом поддержку остальных языков.

СПИСОК ЛИТЕРАТУРЫ

1. *Castagnetti G., Piccolo M., Villa T., et al.* Solving Parallel Equations with balm-ii, Technical Report No UCB/EECS-2012-181, Electrical Engineering and Computer Sciences University of California at Berkeley, 2012, <http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-181.pdf>. (дата обращения 21.03.2018).
2. *Behrmann G., David A., Larsen K.* A tutorial on Uppaal // *Lecture Notes in Computer Science*. 2004. V. 3185. P. 200–236.
3. *mCRL2* user manual // URL: http://mcrl2.org/release/user_manual/user.html (дата обращения 27.04.18).
4. *Shabaldina N.* FSMTest-1.0: a manual for researches / N. Shabaldina, M. Gromov // *Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2015)*, Batumi, Georgia, September 26–29, 2015. [S. l.]. 2015. P. 216–219.
5. *Громов М.Л., Ланутенко А.В.* Проверка свойств конечных автоматов с помощью набора инструментов mCRL2 // *Изв. вузов. Физика*. 2015. Т. 58. № 11/2. С. 65–69.
6. *Шалыто А.А.* Автоматное программирование // *Известия Уральского государственного университета*. 2006. № 43. С.181–190.

Кан Надежда Владимировна, студентка; n.kan.08@gmail.com