

СЕКЦИЯ I. МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ МАШИН И КОМПЬЮТЕРНЫХ СЕТЕЙ

АНАЛИЗ И ВЕРИФИКАЦИЯ ИЕРАРХИЧЕСКИХ ВРЕМЕННЫХ ТИПИЗИРОВАННЫХ СЕТЕЙ ПЕТРИ

Г.А. Белошапко, В.А. Непомнящий

*Институт систем информатики им. А.П. Ершова СО РАН
ordcoder@gmail.com*

Введение

Раскрашенные сети Петри (РСП) широко используются в качестве моделей распределенных систем. Для них разработана известная система анализа CPN Tools [1]. В ИСИ СО РАН разработана система верификации раскрашенных сетей Петри [6], использующая известную систему SPIN [2]. Также разработана модификация раскрашенных сетей Петри, названная иерархическими временными типизированными сетями Петри (ИВТ-сетями) [3,4]. ИВТ-сети использовались с целью анализа и верификации распределенных систем, представленных на языке SDL [3,4]. Естественно, возникает задача применения к ИВТ-сетям мощных средств верификации и анализа, разработанных для РСП. Для этого наряду с системой анализа ИВТ-сетей можно использовать транслятор из ИВТ-сетей в РСП.

Работа частично поддержана грантом РФФИ № 14-07-00401.

1. Сети Петри высокого уровня

Раскрашенные сети Петри являются расширением ординарных сетей Петри. В отличие от ординарных сетей Петри, где все фишки считаются неотличимыми друг от друга, каждая фишка в раскрашенной сети обладает индивидуальностью — значением некоторого типа, которое называется цветом. Неиерархическая раскрашенная сеть Петри состоит из трех частей: структуры сети, деклараций и пометок сети. Иерархическая сеть представляет собой множество неиерархических сетей, функционирующих как единое целое.

Декларации состоят из описания множеств цветов (типов) и объявлений переменных, каждая из которых принимает значения из некоторого множества цветов. Декларации также могут содержать определение операций и функций на языке ML.

Пометка сети приписывается месту, переходу либо дуге и определяет правила распределения и перемещения фишек по сети. Каждое место имеет три разных типа пометок: имя места, множество цветов и инициализирующее выражение.

При функционировании раскрашенной сети возможность срабатывания перехода зависит не только от наличия фишек во входных местах перехода, но и от их значений. Чтобы определить срабатывание перехода, необходимо определить значения переменных перехода. При этом все вхождения одной и той же переменной замещаются одним и тем же значением. Набор значений переменных, при которых выполнена спусковая функция перехода, называется связыванием. Значение выражения на входной дуге определяет, сколько и каких фишек должно содержаться в соответствующем входном месте перехода, чтобы переход мог сработать. Выражения на выходных дугах определяют, сколько и каких фишек будет помещено в выходные места перехода, когда он работает.

Для представления времени вводится понятие глобальных часов. Значение часов представляет текущее время в модели. Начальное значение часов задается при описании сети. Множества цветов (все или часть) получают признак *timed*.

Опишем ИВТ-сети Петри. Дуги в ИВТ-сетях не имеют пометок, а вместо них переходы помечаются выражениями на языке Python, которые определяют результирующую разметку при каждом срабатывании переходов. В ИВТ-сетях присутствуют места-очереди, способные хранить неограниченное число фишек. Новая фишка в таком месте помещается в конец очереди и остается недоступной до тех пор, пока из места не будут извлечены все фишки, поступившие до нее.

В ИВТ-сетях различают два вида мест, обычные и многослойные. Многослойное место помечено специальным типом *layer*, представляющим пару из номера слоя (целого числа) и значения любого допустимого типа. Многослойное место, которое не является очередью, всегда содержит не более одной фишки, принадлежащей одному слою. Многослойное место-очередь содержит не более одной очереди в каждом слое.

ИВТ-сети являются квазибезопасными в том смысле, что каждое место может иметь не более одной фишки, принадлежащей определенному слою.

В ИВТ-сетях временной механизм связан с переходами: каждому переходу сопоставлена пара неотрицательных чисел d_{\min} и d_{\max} . Если t — момент времени, в который переход стал возможен, то его срабатывание произойдет в некоторый момент времени из интервала $[t + d_{\min}, t + d_{\max}]$, если условие срабатывания не будет нарушено.

2. Свойства сетей ИВТ-сетей Петри

Перечислим некоторые базовые свойства ИВТ-сетей. Для них рассматриваются только разметки, достижимые из начальной разметки.

Достижимость (*reachability*). Разметка Q называется достижимой из разметки P , если существует допустимая последовательность срабатываний переходов, переводящая сеть из разметки P в разметку Q .

Ограниченность (*boundedness*). Для конкретного места сети определяется нижняя L и верхняя границы U количества фишек в нем. Это означает, что во время функционирования сети данное место всегда содержит не менее L и не более U фишек. Аналогично определяется нижняя Lm и верхняя Um границы места в виде мультимножеств фишек, содержащихся в нем. То есть во время функционирования сети множество фишек в данном месте является надмножеством Lm и подмножеством Um .

Домашняя разметка (*home marking*). Разметка сети называется домашней, если она достижима из любой разметки. Аналогично, множество разметок называется домашним, если из любой разметки сети достижима хотя бы одна разметка из множества.

Живость (*liveness*). Разметка называется тупиком, если в ней нет возможных переходов. Переход называется мертвым, если он никогда не сможет сработать. Переход называется живым, если он никогда не может стать мертвым.

Справедливость (*fairness*). Свойство справедливости определено, если у сети есть последовательность бесконечных срабатываний (ПБС). Переход называется частичным, если он встречается бесконечно часто в любой ПБС. Переход называется честным, если он встречается бесконечно часто в любой ПБС, в которой он становится допустимым бесконечно часто. Переход называется справедливым, если он встречается бесконечно часто в любой ПБС, в которой он остается допустимым с некоторого момента.

3. Метод трансляции ИВТ-сетей в РСП

Рассмотрим разработанный метод трансляции ИВТ-сетей в РСП. Транслятор строит результирующую сеть в несколько этапов. На каждом этапе набор конструкций, специфичных для ИВТ-сетей, преобразуется в конструкции раскрашенных сетей Петри,

симулирующих их работу. Общую схему алгоритма трансляции можно представить в следующем виде:

- по файлу с ИВТ-сетью в формате PNML построить внутреннее представление сети;
- преобразовать места-очереди в места типа список;
- преобразовать многослойные места в места с типом кортеж из двух элементов;
- транслировать временные конструкции;
- сгенерировать ML-код и декларации, соответствующие выражениям на языке Python;
- перевести полученную сеть в формат, совместимый с CPN Tools.

Опишем подробнее реализацию транслятора и форматы данных. Транслятор представляет собой исполняемую программу на языке Python 2. Транслятор состоит из нескольких модулей, отвечающих за различные функции, такие как связь мест и мест-ссылок на подстраницах, генерация деклараций, поиск состояний ИВТ-сети. Одной из причин выбора Python как языка программирования является наличие встроенной библиотеки для построения синтаксического дерева Python кода, которая используется для разбора деклараций в ИВТ-сетях.

Входным файлом является файл с ИВТ-сетью в формате PNML (основанном на XML). Выходным файлом является файл с РСП в формате системы CPN Tools. Этот формат тоже основан на XML. Транслятор работает с сетью во внутреннем представлении, которое является графом с вершинами типов страница, место, переход и дуга. Ребра соединяют страницы со всеми находящимися на них местами, переходами и дугами, а также дуги с инцидентными местами и переходами.

Места-очереди транслируются в места типа список. Первый элемент списка соответствует началу очереди, последний — концу. Заметим, что место-очередь преобразуется в одно место.

Транслятор преобразует многослойное место типа X в место типа $(\text{product int} * X)$. Фишке в слое n со значением x многослойного места будет соответствовать фишка со значением (n, x) в РСП. Заметим, что многослойное место преобразуется в одно место, а количество дуг не увеличивается.

В ИВТ-сетях используется временная модель Мерлина, в которой время определяется локально для каждого перехода, то есть готовый переход может сработать в течение определенного интервала времени. В РСП используется глобальное время, а фишки несут временной штамп, который показывает по истечению какого момента времени они готовы к использованию. Известно, что данные модели эквивалентны.

При трансляции перехода с точечными временным интервалом вводится вспомогательное место T_{delay} , отсчитывающее задержку между готовностью перехода и возможностью его срабатывания.

Для моделирования перехода с временным интервалом $[a, b]$, показанном на рис. 1, вводится вспомогательное место T_{delay} и переход T_{wait} . T_{wait} служит как альтернатива переходу T . Каждый раз, когда может сработать переход T , вместо него может сработать переход T_{wait} .

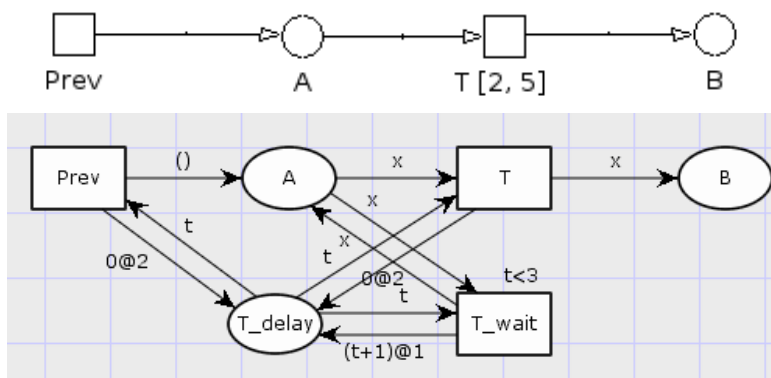


Рис. 1. Пример трансляции временной конструкции

При трансляции переходов с временной задержкой появляется не более одного дополнительного места и перехода. Таким образом суммарное количество мест и переходов увеличивается не более, чем в 3 раза.

4. Анализ ИВТ-сетей Петри

Для анализа свойств ИВТ-сетей можно использовать трансляцию их в раскрашенные сети Петри. Основные сложности, возникающие при таком подходе состоят в том, что результаты анализа получаются для раскрашенной сети Петри, и необходим «обратный ход», чтобы перенести эти результаты на исходную сеть.

Поэтому возникает задача проверки набора простых свойств ИВТ-сетей без трансляции в РСП. Для этого разработаны алгоритмы проверки свойств ИВТ-сетей Петри. Набор свойств был выбран аналогично тому набору свойств, которые можно проверить с помощью системы CPN Tools у раскрашенных сетей Петри, а именно свойства достижимости, ограниченности, живости, а также свойство, является ли разметка домашней.

Реализована программа на языке Python, осуществляющая проверку данных свойств. Общая схема работы алгоритма анализа:

- по файлу с ИВТ-сетью в формате PNML построить внутреннее представление сети;
- построить граф достижимости разметок поиском в глубину из начальной разметки;
- выделить компоненты сильной связности графа достижимости разметок;
- проверить достижимость разметок;
- проверить ограниченность мест;
- проверить живость разметок и переходов;
- определить множество домашних разметок.

5. Пример

С помощью транслятора была успешно преобразована ИВТ-сеть, моделирующая протокол PAR [5] в РСП.

ИВТ-сеть, моделирующая протокол PAR, содержит 78 мест и 75 переходов на 9 страницах. Сеть содержит 3 фишки в месте User_Sender. Эти фишки моделируют сообщения, передаваемые протоколом. Полученные сообщения моделируются фишками в месте Receiver_User.

С помощью разработанной системы трансляции были успешно верифицированы несколько свойств протокола:

1. Безопасность. В любом достижимом состоянии в каждом месте одновременно находится не более одной фишки.

2. Невозможность потери сообщений. В любом достижимом состоянии последовательность полученных сообщений является префиксом последовательности отправленных сообщений.
3. Невозможность вставки сообщений. Невозможно состояние, в котором получено больше сообщений, чем отправлено.

Заключение

В данной работе описан метод трансляции ИВТ-сетей в РСР, на базе которого реализована система анализа и верификации ИВТ-сетей, использующая средства анализа и верификации РСР CPN Tools [1] и CPN Ver [6]. Применение этого метода проиллюстрировано примером верификации протокола PAR. В дальнейшем предполагается применение разработанных средств для анализа и верификации различных коммуникационных протоколов.

Литература

1. Jensen K., Kristensen L. M. Coloured Petri Nets. Modelling and Validation of Concurrent Systems. Springer-Verlag, 2009.
2. Holzmann G.J. The SPIN model checker. Primer and Reference Manual. Addison-Wesley, 2004.
3. Непомнящий В.А., Аргиров В.С., Белоглазов Д.М., Быстров А.В., Четвертаков Е.А., Чурина Т.Г. Моделирование и верификация коммуникационных протоколов, представленных на языке SDL, с помощью сетей Петри высокого уровня // Программирование. 2008, № 6, с. 35–49.
4. Чурина Т.Г., Аргиров В.С. Моделирование спецификаций языка SDL с помощью модифицированных ИВТ-сетей. Препринт ИСИ СО РАН. 2005. № 124. 62 с.
5. Таненбаум Э. Компьютерные сети. СПб.: Питер, 2003.
6. Стененко А.А., Непомнящий В.А. Верификация раскрашенных сетей Петри методом проверки моделей. Препринт ИСИ СО РАН. 2015. № 178. 29 с.

ИССЛЕДОВАНИЕ ГИБРИДНЫХ ЭВРИСТИЧЕСКИХ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

Е.А. Боронихина, В.А. Сибирякова

Томский государственный университет
lin_san@mail.ru, val349@mail.ru

Задача коммивояжера — важная задача транспортной логистики, отрасли, занимающейся планированием транспортных перевозок. Задача состоит в определении кратчайшего гамильтонова цикла в графе. Существует несколько частных случаев задачи коммивояжера: геометрическая (планарная или евклидова), треугольная, симметричная и асимметричная задачи, а также задача с несколькими коммивояжерами — определение нескольких циклов в графе, суммарная стоимость которых будет минимальной.

Задача относится к классу NP-полных и является трансвычислительной. Чтобы гарантировать существование решения, наложим на задачу следующее ограничение: будем считать входной граф полностью связным. В случае если между некоторыми городами не существует сообщения, введем дополнительные ребра с большей длиной. Если оптимальный маршрут для данного графа существует, введенные ребра никогда не попадут в решение.

Выделяют два типа решения задачи коммивояжера: точные и эвристические [1]. К точным методам относятся алгоритм полного перебора (АПП) и метод ветвей и границ (МВГ). На поиск решения требуются большие временные затраты, но результаты, полученные данными методами, имеют 100% точность.

Эвристические методы будем разбивать на подгруппы: жадные методы — алгоритм ближайшего соседа (БС) и рекурсивного перебора (АРП), эволюционные — генетические (ГА) и метод имитации отжига (МИО), а так же поведенческие (роевые) алгоритмы — муравьиный (АМК), имитации иммунной системы (ИИС) и интеллектуальных