МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИЗИРОВАННЫХ СИСТЕМ

УДК 004.021; 519.711

В.А. БАШКИН

ВЕРИФИКАЦИЯ НА ОСНОВЕ МОДЕЛЕЙ С ОДНИМ НЕОГРАНИЧЕННЫМ СЧЕТЧИКОМ

Предлагается новый метод доказательства свойств систем, моделируемых при помощи односчетчиковых сетей Петри (параллельных и распределенных процессов, алгоритмов и протоколов). Подобные модели позволяют представлять системы как с конечными, так и с бесконечными множествами состояний. Проверяемые свойства предлагается записывать при помощи формул теморальной логики EF (логики достижимости). Новизна метода состоит в использовании некоторых конструктивных теоретико-числовых свойств бесконечной части одномерных линейных и полулинейных множеств. Представлен алгоритм формальной верификации темпоральных формул логики EF, использующий декомпозицию формулы и вычисления над однопериодическими полулинейными базисами специального вида.

Ключевые слова: *глобальная верификация моделей; темпоральная логика; односчетчиковые сети; полулинейность; достижимость.*

A new method is presented for verification of systems, modeled by one-counter Petri nets (parallel and distributed processes, algorithms and protocols). Petri net models allow to represent both finite- and infinite-state systems. It is proposed to formulate the checked property by EF-formula (where EF is a temporal logic of reachability). The novelty of our approach is based on the application of some specific constructive number-theoretic properties of an infinite part of one-dimensional linear and semilinear sets. We present an algorithm of EF temporal formulae verification, that uses formula decomposition and computations over specific single-periodic semilinear bases.

Keywords: *global model-checking; temporal logic; one-counter nets; semilinearity; reachability.*

ВВЕДЕНИЕ

При разработке достаточно сложных систем важной задачей является строгое доказательство их корректности, то есть соответствия системы формальному описанию её предполагаемых свойств. Такими свойствами могут быть: отсутствие тупиков, невозможность зацикливания, справедливое распределение ресурса, правильная последовательность действий, отсутствие избыточных элементов и т.д. В случае параллельных и распределенных систем возникают дополнительные критические свойства, связанные с возможностью возникновения нескольких независимых потоков вычислений: максимальная и минимальная степени параллелизма, неизбежность синхронизации после распараллеливания и т.д.

Одним из классических способов формализованного анализа систем является верификация моделей (model checking) [5]. Метод состоит в том, что вначале строится математическая модель системы, адекватно отражающая важные аспекты её структуры (например, диаграмму переходов управляющего устройства), а также формулируется математическое утверждение, описывающее нужное свойство (например, отсутствие тупиков). Далее запускается какой-то алгоритм проверки, который выясняет истинность данного утверждения для данной модели.

Для описания проверяемых свойств обычно используют формулы, заданные посредством различных темпоральных логик. Например, свойства трасс достаточно хорошо описываются при помощи логики линейного времени (LTL – Linear Time Logic), свойства систем переходов – при помощи логики ветвящегося времени (CTL –

Computation Time Logic). В данной работе мы рассматриваем свойства графа достижимых состояний системы, формализованные при помощи логики EF (сужения CTL) [5].

В тех случаях, когда модель системы конечна (может быть описана конечным автоматом) и не очень велика, мы простым перебором можем проверить практически все её интересные свойства. Однако если реальная система содержит какие-то неограниченные составляющие (например, целочисленные переменные) или просто достаточно объемна, проверка становится невозможной или слишком трудоемкой. В частности, для моделей систем с двумя счетчиками и проверкой на ноль неразрешимы практически все важнейшие свойства – достижимость заданного состояния, отсутствие тупиков и т.д. (что следует из неразрешимости проблемы останова для универсальных моделей вычисления, к которым относятся и двухсчетчиковые машины Минского).

В связи с этим в настоящее время большое внимание уделяется поиску более узких классов бесконечных моделей со всё ещё достаточно обширными наборами разрешимых свойств, а также разработке новых символьных методов анализа этих свойств. Символьные методы подразумевают использование для задания бесконечных множеств (например, множеств состояний) какого-то конечного (символьного) представления. Так, в регулярной верификации используется представление бесконечных множеств при помощи регулярных выражений. Для представления бесконечных полулинейных множеств возможно использование формул арифметики Пресбургера [4].

Данная работа посвящена исследованию систем, содержащих один потенциально неограниченный ресурс, моделируемый целочисленным неотрицательным счетчиком. Такие системы эквивалентны сетям Петри не более чем с одной неограниченной позицией [2]. Примерами ресурсов могут служить «заявки» или «исполнители» в схемах потоков работ (workflow), «пакеты» или «задержки» в моделях сетевых протоколов, «финансы» в схемах бизнес-процессов и т.д. Проблема локальной верификации систем с одним ресурсом достаточно хорошо изучена ([5,7,9]). Нами рассматривается более общая проблема глобальной верификации, которая может быть сформулирована следующим образом: «При каких точных количествах ресурса (значениях счетчика) заданная система обладает заданным свойством?» Здесь в качестве «свойства» предполагается произвольная формула логики EF. Разработан алгоритм решения данной проблемы, основанный на представлении бесконечных наборов значений счетчика в виде однопериодических полулинейных базисов. Алгоритм состоит из двух этапов: вначале строится конечное символьное дерево достижимости системы, затем на его основе индуктивно по структуре логической формулы строится множество ресурсов, для которых данная формула выполняется.

Работа построена следующим образом. В разделе 2 даны синтаксис и семантика логики EF, показаны примеры свойств, записанных на этом языке. Сформулированы проблемы локальной и глобальной верификации (доказательства свойств). В разделе 3 приведен способ моделирования систем при помощи односчетчиковых сетей. Раздел 4 посвящен описанию новых результатов, касающихся конструктивных особенностей однопериодического представления полулинейных множеств натуральных чисел. В разделе 5 приводятся основанные на этих результатах алгоритм построения символьного дерева достижимости для односчетчиковой сети и алгоритм глобальной символьной верификации темпоральной формулы. В заключении приводятся возможные направления дальнейших работ в данной области.

ФОРМАЛЬНОЕ ОПИСАНИЕ СВОЙСТВ

Пусть LTS = $(S, s_0, \rightarrow, L)$ – *система помеченных переходов*, где S – множество состояний, $s_0 \in S$ – начальное состояние, $\rightarrow \subseteq S\times S$ – отношение переходов (множество дуг), L: $(\rightarrow) \rightarrow \Sigma$ – помечающая функция (здесь и далее Σ – конечный алфавит меток срабатываний).

Переход t=(s,s') меняет состояние системы с s на s'. Таким образом, система помеченных переходов – это предельно абстрактная схема функционирования обладающей состояниями реальной системы. Она может быть и бесконечной, если бесконечно множество состояний S.

Темпоральная логика EF обладает следующим синтаксисом [5,6]:

$$j \ ::= \mathbf{true} \mid \neg j \mid j_1 \wedge j_2 \mid \mathrm{E}\langle a\rangle j \mid \mathrm{EF} j$$

Отношение выполнимости |= для состояния s системы LTS, темпоральной формулы j и метки $a \in \Sigma$ определяется индуктивно:

$s \models \mathbf{true}$;

$s \models \neg j \ \Leftrightarrow \ \neg(s \models j)$;

$s \models j_1 \wedge j_2 \ \Leftrightarrow \ s \models j_1 \ \wedge \ s \models j_2$;

$s \models \mathrm{E}\langle a\rangle j \ \Leftrightarrow \ \exists\, s' \in S : (s,s') \in (\rightarrow) \ \wedge L(s,s')=a \wedge s' \models j$;

$s \models \mathrm{EF} j \ \Leftrightarrow$ для некоторого пути $(s_1,s_2,\ldots)$, где $s=s_1$, $\exists\, i\,{}^3 1 : s_i \models j$.

Стандартным образом через $\wedge$ и $\neg$ определяются остальные логические связки ($\vee$,$\Rightarrow$ и др.). Для удобства записи формул обычно дополнительно определяют двойственные темпоральные операторы: $\mathrm{A}\langle a\rangle j = \neg\mathrm{E}\langle a\rangle\neg j$, $\mathrm{AG} j = \neg\mathrm{EF}\neg j$.

Формула $\mathrm{E}\langle a\rangle j$ означает «может сработать переход с меткой a, переводящий систему в какое-то новое состояние, в котором выполняется формула j ». Двойственная к ней формула $\mathrm{A}\langle a\rangle j$ означает «срабатывание любого перехода с меткой a переводит систему в состояние, в котором выполняется формула j ». Таким образом, оператор E формализует понятие «существования», оператор A – понятие «неизбежности».

Формула $\mathrm{EF} j$ означает «может сработать последовательность переходов, переводящая систему в какое-то новое состояние, в котором выполняется формула j ». Двойственная к ней формула $\mathrm{AG} j$ означает «срабатывание любой последовательности переходов переводит систему в состояние, в котором выполняется формула j ».

Логика EF является расширением логики Хеннесси-Милнера и сужением логики ветвящегося времени CTL [5]. Она позволяет формализовать всевозможные свойства достижимости, например:

– AG EF $\mathrm{A}\langle a\rangle$**true** – в системе всегда существует вероятность наступления события «a» (точнее, из любого достижимого состояния достижимо состояние, при котором может выполниться только переход «a»). Это свойство может служить признаком правильной завершаемости процесса (если «a» — действие, возможное только в финальном состоянии).

– EF AG $\mathrm{E}\langle a\rangle$**true** – система может сработать таким образом, что в ней возникнет единственный постоянно активный переход «a». Например, этот переход может сигнализировать о возникновении переполнения памяти.

– AG ($\mathrm{E}\langle a\rangle$**true** $\Rightarrow$ $\mathrm{E}\langle b\rangle$**true**) – событие «$b$» возможно только в таких состояниях системы, в которых возможно событие «a». Например, подтверждение транзакции возможно только тогда, когда возможен и её откат.

– $\mathrm{A}\langle a\rangle$ EF $\mathrm{E}\langle b\rangle$**true** – если возможно «a», то после его выполнения остается хотя бы одна возможность когда-то в будущем выполнить «b». Например, в качестве «a» может выступать событие принятия рискованного решения, в качестве «b»

– событие получения крупных дивидендов (другими словами, «риск не является бессмысленным»).

– AG A$\langle a \rangle$ AG EF$\langle b \rangle$**true** – если возможно «*a*», то после его выполнения рано или поздно неизбежно наступает «*b*». Например, это может соответствовать правильности написания параллельной программы (если «*a*» – событие распараллеливания, «*b*» – событие синхронизации, то есть соединения распараллеленных ветвей вычисления).

– AG ((EF E$\langle a \rangle$**true**) $\vee$ (E$\langle b \rangle$**true**)) – всегда возможно либо возникновение в будущем «*a*», либо прямо сейчас – «*b*». Например, «*a*» – это правильное завершение процесса, «*b*» – предусмотренный аварийный выход.

Таким образом, темпоральная логика EF очень удобна в качестве языка описания интересующих нас свойств систем. Разумеется, она не является универсальной (в частности, не позволяет формализовать существование бесконечной последовательности срабатываний одного и того же перехода), однако для большинства задач проверки свойств достижимости достаточно выразительна.

Задача (локальной) верификации состоит в определении того, выполняется ли данная формула темпоральной логики в данном состоянии системы.

Задача глобальной верификации состоит в построении конечного эффективного представления множества всех состояний системы, в которых выполняется данная формула темпоральной логики.

Для конечных систем задача глобальной верификации сводится к конечному набору задач локальной верификации, каждая из которых может быть решена перебором. В общем случае систем с бесконечным числом состояний даже задача локальной верификации становится неразрешимой. Следовательно, разумно рассматривать какие-то промежуточные классы систем, например, системы с одним видом неограниченного ресурса.

ФОРМАЛЬНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ С ОДНИМ РЕСУРСОМ

Пусть **Nat** – множество неотрицательных целых чисел.

Односчетчиковой сетью [7] называется набор $N = (Q,T,L)$, где

– Q – конечное множество управляющих состояний,

– $T \subset Q \times Q \times \mathbf{Z}$ – конечное множество переходов,

– $L: T \to \Sigma$ – помечающая функция.

Состояние сети описывается парой (q,c), где $q \in Q$ – текущее управляющее состояние, $c \in \mathbf{Nat}$ – текущее значение счетчика.

Переход $t=(q,q',z)$ *активен* в состоянии (q,c), если $c+z>0$.

Активный переход может *сработать*, переводя сеть в состояние $(q',c+z)$ (обозначается $(q,c) \xrightarrow{t} (q',c+z)$). Изменение счетчика z также будем обозначать $\delta(t)$.

Внешний наблюдатель в момент срабатывания перехода t видит только его метку $L(t)$ (то есть он не может различить срабатывания переходов с одинаковой меткой).

Для конечной последовательности переходов $U = t_1.t_2 \ldots t_{n-1}.t_n$ определим пред- и постусловие (далее $\dot{-}$ обозначает усеченное вычитание до нуля):

$$^{\bullet}U = {}^{\bullet}(t_2 \ldots t_{n-1}.t_n) + |\delta(t_1)| \text{ при } \delta(t_1)<0 \text{ и } {}^{\bullet}U = {}^{\bullet}(t_2 \ldots t_{n-1}.t_n) \dot{-} \delta(t_1) \text{ при } \delta(t_1) \geq 0;$$

$$U^{\bullet} = (t_1.t_2 \ldots t_{n-1})^{\bullet} \dot{-} |\delta(t_n)| \text{ при } \delta(t_n)<0 \text{ и } U^{\bullet} = (t_1.t_2 \ldots t_{n-1})^{\bullet} + \delta(t_n) \text{ при } \delta(t_n) \geq 0.$$

Односчетчиковые сети эквивалентны сетям Петри с не более чем одной неограниченной позицией [2]. Таким образом, эта модель адекватно формализует как последовательные, так и параллельные системы с неограниченным ресурсом. Процесс моделирования состоит в построении схемы конечной части реальной системы, то есть