

УДК 519.767.2

Дата подачи статьи: 23.03.15

DOI: 10.15827/0236-235X.110.048-054

МЕТОД ОГРАНИЧЕНИЙ ВЕРИФИЦИРУЕМЫХ МОДЕЛЕЙ*Ю.П. Кораблин, д.т.н., профессор, y.p.k@mail.ru;**А.А. Шипов, аспирант, a-j-a-1@yandex.ru**(Российский государственный социальный университет,
ул. Вильгельма Пика, 4, г. Москва, 129256, Россия)*

Программные системы с каждым днем становятся все более сложными и комплексными, поэтому необходимо наличие таких инструментов, которые позволяли бы относительно легко выполнять проверку их работы на соответствие заданным спецификациям, особенно, когда речь идет о больших и распределенных программных системах. В данной статье предлагается новая методика, представляющая собой отдельный этап процесса верификации систем подобного рода и позволяющая сделать верификацию в целом более эффективной. Поскольку методика – это отдельный этап верификации для метода Model Checking или метода проверки на моделях и может быть с легкостью автоматизирована, ее внедрение в уже существующую технологию не является сложной задачей.

В основе работы метода лежит принцип, согласно которому модели всех систем содержат избыточные вычислительные последовательности, их анализ при проверке конкретных свойств системы необязателен и не оказывает влияния на конечный результат. Исключение данных вычислительных последовательностей из исходной модели перед верификацией заданных свойств и является основной задачей метода.

В статье дано подробное описание предлагаемого метода, а также выполнена оценка его эффективности. Теоретический материал подкреплён рядом наглядных примеров работы метода, в частности, примером для распределенной программной системы.

Авторы полагают, что использование данного метода позволит существенно сократить число анализируемых при верификации вычислительных последовательностей для модели заданной системы и тем самым повысить производительность этого процесса.

Ключевые слова: верификация, модель Крипке, автомат Бюхи, формула временной логики, LTL, CTL.

В последнее время широкое распространение в области верификации получил метод Model Checking [1–6], или метод проверки на моделях. Суть его – в верификации свойств системы, представленных в виде формул временных логик LTL или CTL [7, 8]. При этом верифицируемая модель может быть задана явно (путем перечисления всех состояний и соединяющих их ребер) или неявно (путем задания булевых функций, определяющих отношение переходов и множество начальных состояний). Однако при верификации больших информационных систем, в том числе и распределенных программных, зачастую приходится иметь дело с очень большим числом состояний вычислительного процесса для их моделей. Например, для программы, в рамках которой одновременно выполняются 10 потоков, причем каждый из них может находиться в одном из 5 состояний, общее число состояний вычислительного процесса может достигать $5^{10} = 9765625$, что может стать непреодолимым препятствием на пути верификации тех или иных ее свойств. Кроме того, в модели могут явно присутствовать вычисления, переход в которые никогда не встретится на практике.

Сегодня существуют два основных подхода к выполнению верификации данных систем:

- построение более абстрактной, менее детализированной модели на базе исходной;
- выполнение верификации модели на ее сжатом, приведенном к некоторому сжатому виду представлении.

Предлагаемый в данной статье метод ограничений верифицируемых моделей позволяет сфор-

мулировать совершенно иной подход, направленный на сокращение тех состояний модели, которые невыполнимы на практике или избыточны, а также на явное уменьшение числа вычислительных последовательностей до того минимума, которого будет достаточно для выполнения верификации тех или иных свойств. Например, для построенной модели конкретного авиатехнического средства требуется выполнить верификацию систем руления и механизмов реверсной работы двигателей.

Очевидно, что для решения поставленной задачи совершенно не требуется проверка тех вычислительных последовательностей, переход в которые объективно может наступить лишь во время пребывания судна в воздушном пространстве. Поэтому логично сначала произвести редукцию избыточных вычислительных последовательностей для заданной модели и лишь затем выполнять верификацию ее свойств.

**Ограничение
верифицируемых моделей**

Рассмотрим предлагаемый метод ограничения верифицируемых моделей на примере работы кухонной микроволновой печи. Через c , m , w , f обозначим предикаты, определяющие состояния нашей модели: c – дверца печи закрыта, m – задан режим работы печи, w – печь работает, f – в печи присутствует еда. На рисунке 1 изображена модель печи, представленная в виде структуры Крипке.

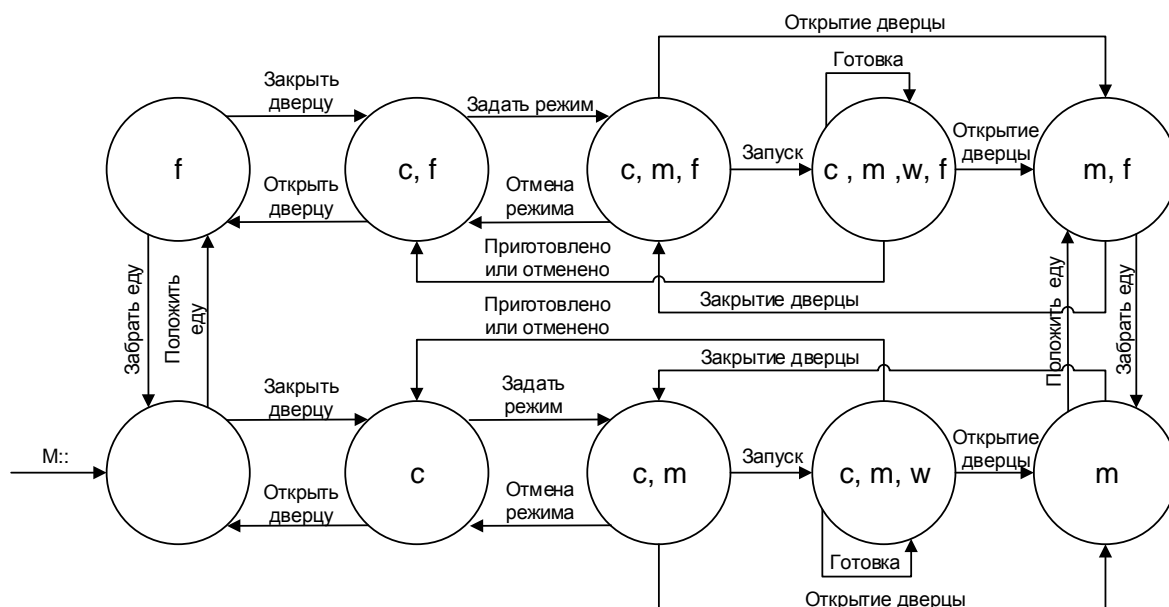


Рис. 1. Модель кухонной микроволновой печи

Fig. 1. A model of kitchen microwave oven

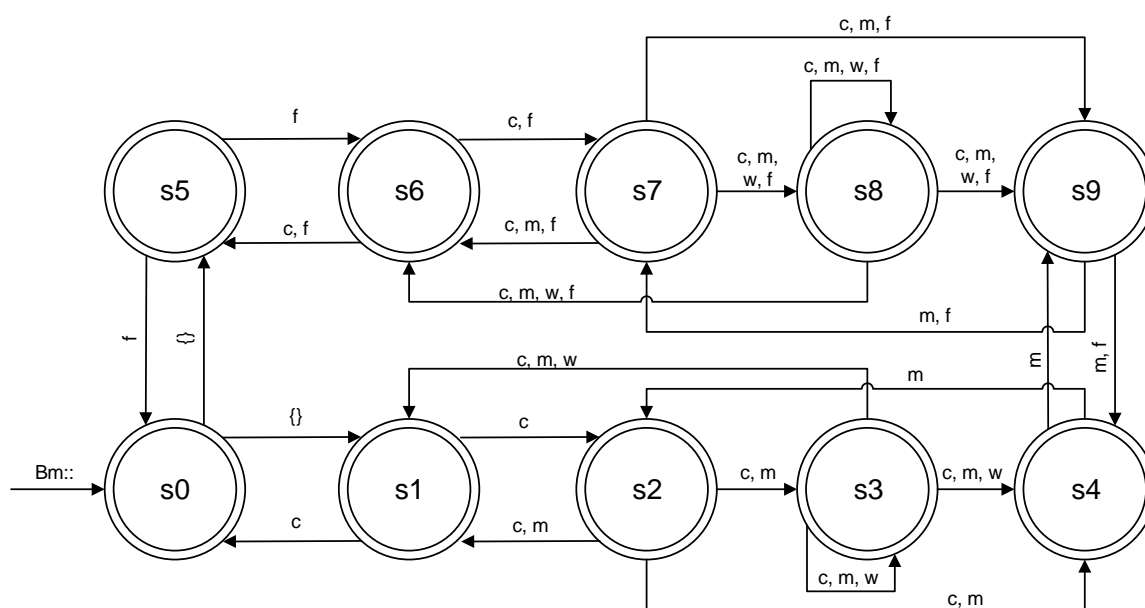


Рис. 2. Автомат Бюхи Вт для кухонной микроволновой печи

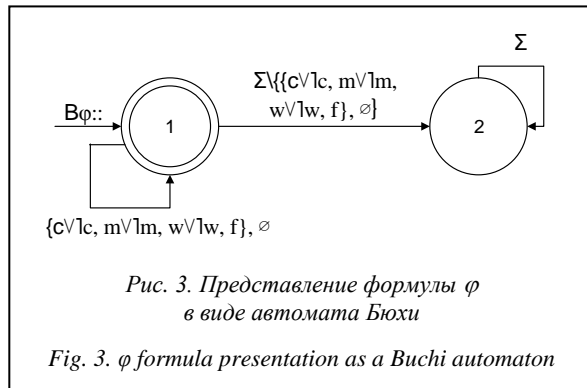
Fig. 2. Bm Buchi automaton for a kitchen microwave oven

Для наложения ограничений необходимо преобразовать данную модель к автомату Бюхи [9], который представлен на рисунке 2.

Ограничением, накладываемым на эту модель, будет следующее условие: при работе с печью в ней обязательно должна находиться еда или печью никто не пользуется. Представим это ограни-

чение в виде формулы временной логики LTL: $\varphi = G(f \vee \emptyset) \leftarrow$ «Всегда либо в печи присутствует еда, либо дверца открыта, режим не задан, печь не работает и еда в печи отсутствует»; $\emptyset \models l_c \wedge l_m \wedge l_w \wedge l_f$. Для использования данного высказывания в качестве ограничивающего условия необходимо также преобразовать его в автомат Бюхи. Через Σ

обозначим входной алфавит этого автомата над множеством 2^{AP} , где $\Sigma = \{\emptyset, \{c\}, \{m\}, \{w\}, \{f\}, \{c, m\}, \{c, w\}, \dots, \{c, m, w, f\}\}$, $AP = \{c, m, w, f\}$ [1]. Результат представлен на рисунке 3.



Двойной окружностью в автоматах Бюхи обозначается допускающее состояние, то есть такое состояние, через которое автомат должен будет проходить бесконечное число раз, а одинарной окружностью обозначаются недопускающие состояния, то есть такие состояния, пребывание автомата в которых бесконечно долго свидетельствует о недопустимости входного слова ω [1].

Построив синхронную композицию автоматов Bm и $B\phi$ в соответствии с описанным в [1] методом, получим автомат Бюхи $Bm \otimes B\phi$, в котором будет соблюдаться заданное нами условие. На рисунке 4 представлена соответствующая синхронная композиция автоматов Бюхи.

В качестве результирующего автомата, удовлетворяющего заданному ограничению, для которого будет выполняться верификация проверяемых свойств, примем автомат Бюхи $R(Bm \otimes B\phi)$, где R – функция редукции, исключающая сначала все изолированные и недопускающие состояния, а затем все допускающие состояния, из которых не существует ни одного перехода. Изолированными

будем называть такие состояния, в которые не существует ни одного пути из начального состояния. На рисунке 5 изображен данный автомат.

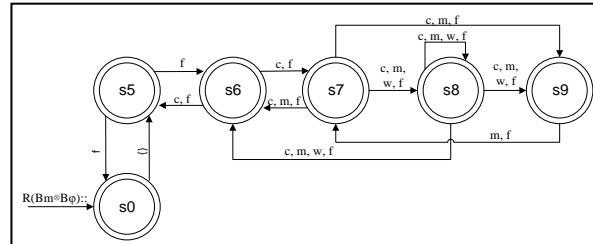


Fig. 5. $R(Bm \otimes B\phi)$ Buchi automaton

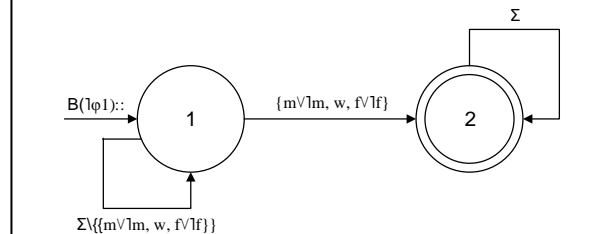
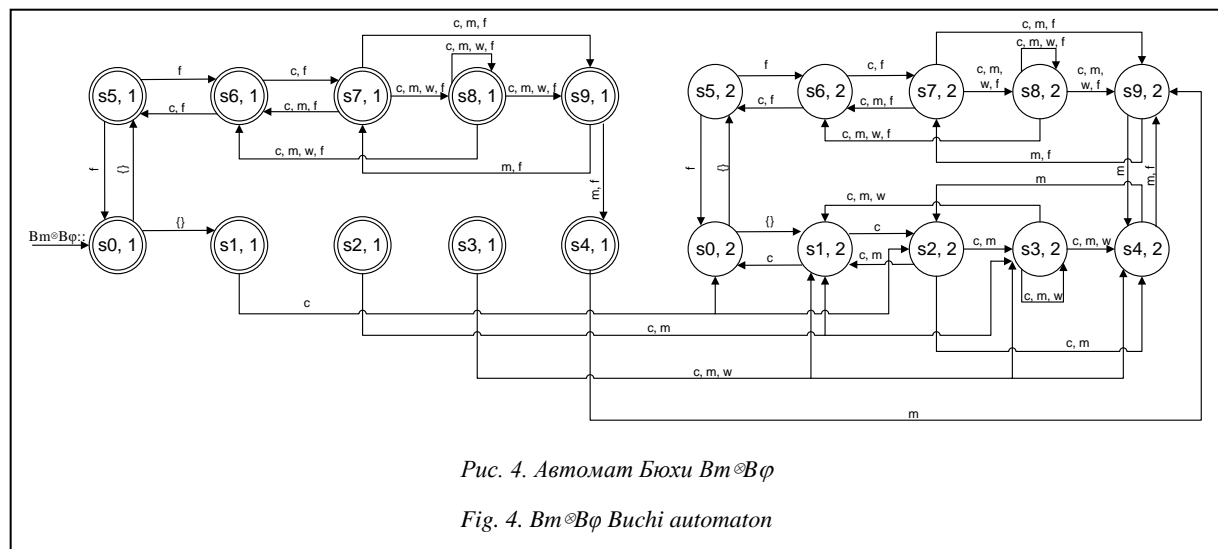
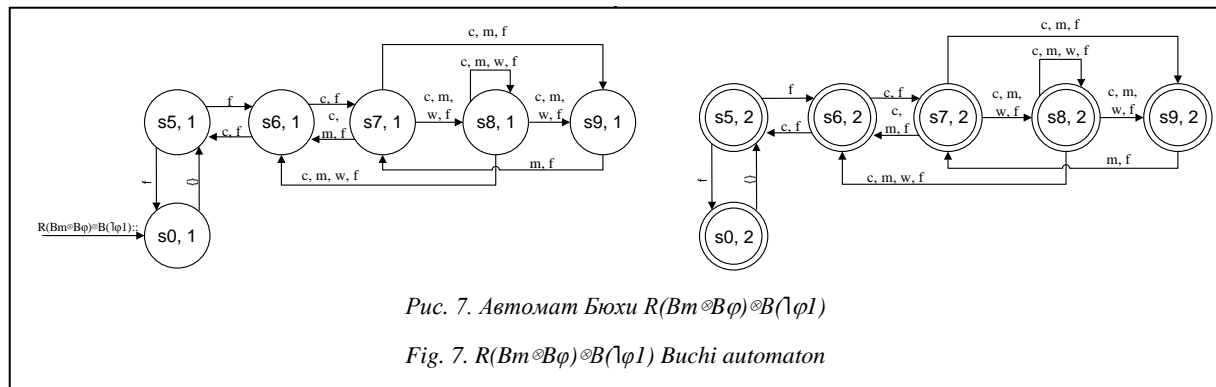


Fig. 6. $\neg \phi 1$ formula presentation as a Buchi automaton

Как видно из рисунка 5, область вычислений нашей модели сократилась на четыре состояния по сравнению с исходной (рис. 2).

Теперь проверим, возможна ли работа печи с открытой дверцей. Для проверки данного свойства введем следующую формулу временной логики LTL: $\phi 1 = G(w \rightarrow c)$ – «Всегда, если печь работает, то значит, дверца закрыта». Возьмем отрицание данной формулы $\neg \phi 1 = \neg(G(w \rightarrow c)) = F(w \wedge \neg c)$ и построим для нее соответствующий автомат Бюхи (рис. 6).





Этот автомат переходит в допускающее состояние только в случае возникновения ошибки, то есть тогда, когда микроволновая печь будет работать с открытой дверцей. Таким образом, построив синхронную композицию данного автомата и проверяемого, можно установить, способна ли система перейти в ошибочное состояние и, если да, то в каком случае. Композиция автоматов $R(Bm \otimes B\varphi)$ и $B(\wedge \varphi 1)$ представлена на рисунке 7.

Нетрудно понять, что из начального состояния данного автомата невозможно попасть ни в одно из допускающих состояний. Это означает, что формула $\wedge \varphi 1$ никогда не будет истинной, а потому можно утверждать, что микроволновая печь будет работать только с закрытой дверцей.

Ограничение моделей распределенных программных систем

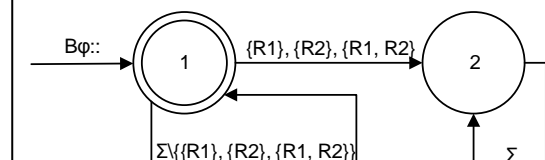
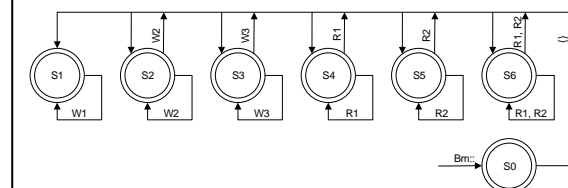
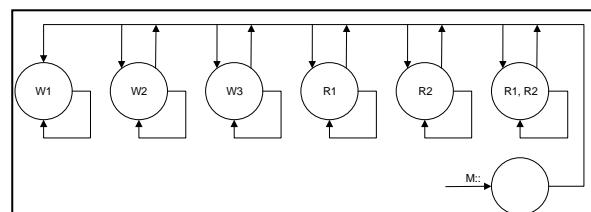
Рассмотрим работу данного метода на примере распределенной программной системы, реализованной на основе модели «читатели-писатели». В этом случае система будет состоять из трех потоков «писателей», осуществляющих запись в разделяемый ресурс, и двух потоков «читателей», имеющих возможность производить чтение из данного ресурса. Условимся, что производить запись одновременно может лишь один из «писателей» и право на чтение имеет любой из «читателей», пока не осуществляется запись. Также введем следующее условие: как только какой-либо «писатель» захочет начать записывать, процесс чтения для всех «читателей» автоматически прекращается. Данная модель может быть, например, задана структурой Крипке, представленной на рисунке 8.

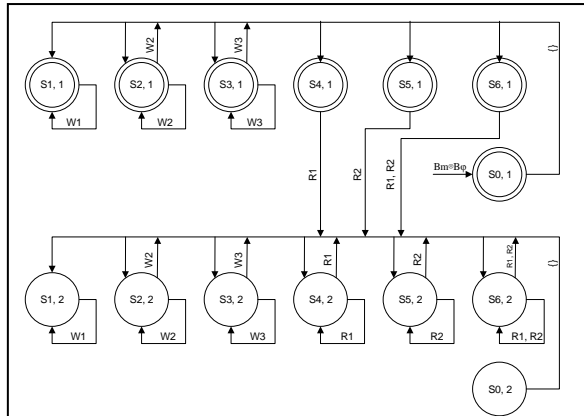
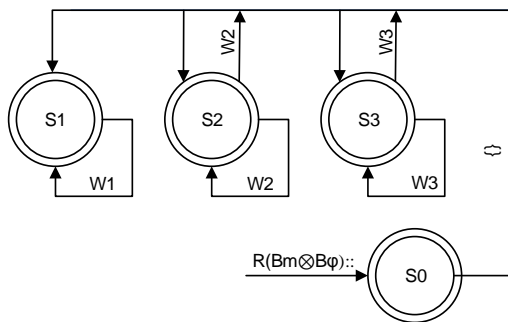
Построим на основе данной модели соответствующий автомат Бюхи (рис. 9) и реализуем для него предложенный выше метод ограничений верифицируемых моделей. После этого проверим, имеется ли в системе возможность возникновения тупиковых ситуаций, то есть таких ситуаций, при которых разделяемый ресурс будет безвозвратно заблокирован работой одного из «писателей».

Введем для рассматриваемой системы ограничение «Всегда осуществляется запись» с целью

оставить лишь потоки «писателей», так как только они способны забрать ресурс в бессрочное пользование в рамках заданных нами условий. Представим это ограничение в виде формулы временной логики LTL: $\varphi = G(W1 \vee W2 \vee W3)$ и построим для нее соответствующий автомат Бюхи (рис. 10).

В этом случае входной алфавит Σ данного автомата над множеством 2^{AP} примет вид: $\Sigma = \{\emptyset, \{W1\}, \{W2\}, \{W3\}, \{R1\}, \{R2\}, \{W1, W2\}, \dots, \{W1, W2, W3, R1, R2\}\}$, $AP = \{W1, W2, W3, R1, R2\}$.



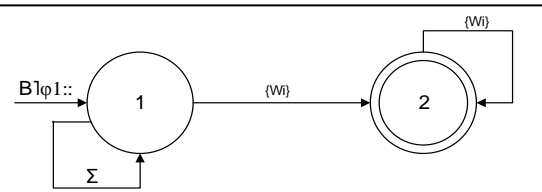
Рис. 11. Автомат Бюхи $Bm \otimes B\phi$ Fig. 11. $Bm \otimes B\phi$ Buchi automatonРис. 12. Автомат Бюхи $R(Bm \otimes B\phi)$ Fig. 12. $R(Bm \otimes B\phi)$ Buchi automaton

Синхронная композиция автоматов $Bm \otimes B\phi$, отвечающая заданному нами условию, представлена на рисунке 11.

Окончательный результат ограничения нашей модели после применения функции редукции R к композиции $Bm \otimes B\phi$ можно видеть на рисунке 12.

Теперь на имеющейся ограниченной вычислительной последовательности модели выполним проверку на существование тупиков. Для этого введем следующую LTL-формулу: $\phi_1 = GF(1W_i)$ – «Всегда в будущем процесс записи для любого из потоков будет обязательно завершен». Возьмем отрицание этой формулы $\neg\phi_1 = FG(W_i)$ и попытаемся доказать, что в будущем хотя бы раз встретится такой «писатель», который никогда не освободит разделяемый ресурс. Автомат Бюхи для формулы $\neg\phi_1$ представлен на рисунке 13.

Как видно из этого рисунка, данный автомат перейдет в допускающее состояние только в том случае, если существует переход, помеченный $\{W_i\}$, и допускающее состояние всегда будет переходить само в себя толь-

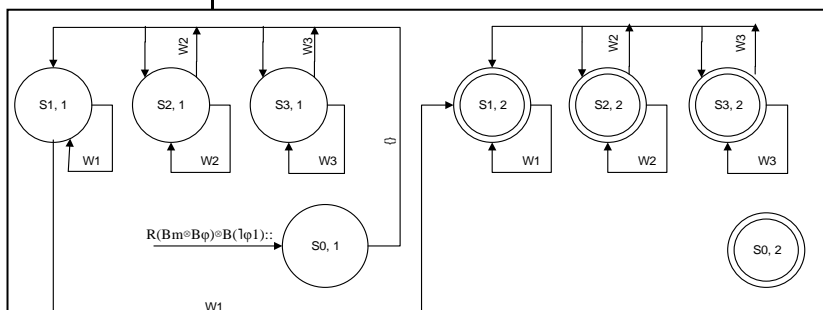
Рис. 13. Представление формулы $\neg\phi_1$ в виде автомата БюхиFig. 13. $\neg\phi_1$ formula presentation as a Buchi automaton

ко по переходу, помеченному тем же $\{W_i\}$. Композиция автоматов $R(Bm \otimes B\phi)$ и $B(\neg\phi_1)$ представлена на рисунке 14.

В этой композиции существует переход в допускающее состояние ($S1, 2$), что свидетельствует о том, что формула $\neg\phi_1$ выполняется и в системе может возникнуть тупиковая ситуация в случае работы первого «писателя». Из состояний ($S2, 1$) и ($S3, 1$) нет переходов в допускающие, так как, если, например, перейти по переходу $\{W_2\}$ из состояния ($S2, 1$) в ($S2, 2$), то в соответствии с контрольным автоматом на рисунке 13 был бы возможен переход дальше только по переходам, помеченным $\{W_2\}$. Однако из состояния ($S2, 2$) существует переход, например, в ($S3, 2$), из которого, в свою очередь, существует переход, помеченный $\{W_3\}$, что явно противоречит контрольному автомату.

Эффективность метода ограничения моделей

В общем виде работу метода ограничения верифицируемых моделей можно описать следующим образом. По заданной модели M строится автомат Бюхи Bm , допускающий все вычислительные последовательности, которые возможны в M . По заданной LTL-формуле ϕ строится ограничивающий автомат Бюхи, допускающий лишь те вычислительные последовательности, на которых выполняется ϕ . После этого строится синхронная композиция автоматов $Bm \otimes B\phi$, на которой после применения к ней редуцирующей функции R осу-

Рис. 14. Автомат Бюхи $R(Bm \otimes B\phi) \otimes B(\neg\phi_1)$ Fig. 14. $R(Bm \otimes B\phi) \otimes B(\neg\phi_1)$ Buchi automaton

ществляется последующая верификация требуемых свойств по алгоритму, описанному в [1].

Произведем оценку вышеописанного метода ограничений верифицируемых моделей с целью демонстрации его практической эффективности.

Пусть дана модель некоторой системы $M(n)$, где n – число ее состояний для автомата Бюхи. И пусть также дано некоторое множество проверяемых условий $\Psi\{S_1(k_1), S_2(k_2), \dots, S_m(k_m)\}$, где $S_i(k_i)$ – контрольный автомат проверяемого условия от k_i состояний автомата Бюхи. Для выполнения проверки очередного условия на модели требуется построить синхронную композицию контрольного автомата проверяемого условия и автомата имеющейся модели, как описано в [1, 10]. Таким образом, для проверки всего множества условий Ψ на модели M потребуется построить m композиций: $M \times S_1 \& M \times S_2 \& \dots \& M \times S_m$. Суммарное число операций данного процесса будет равно $op = nk_1 + nk_2 + \dots + nk_m = \sum_{(1 \leq i \leq m)} (n \cdot k_i) = n \sum_{(1 \leq i \leq m)} k_i$. Теперь введем модель $M'(n')$, которая является сокращенной версией модели $M(n)$ после применения к ней функции редукции R и удовлетворяет условию $n' < n$. Для такой модели число операций при проверке множества условий Ψ будет равно $op' = n'k_1 + n'k_2 + \dots + n'k_m = \sum_{(1 \leq i \leq m)} (n' \cdot k_i) = n' \sum_{(1 \leq i \leq m)} k_i$. Таким образом, неравенство $op' < op$ также имеет место. Отметим, что в Ψ входят лишь те условия, которые можно проверить на ограниченной модели.

Покажем, что существует такая ситуация, когда имеет место неравенство $op' + r < op$, где $r = nz$, n – исходное число состояний модели M , z – число состояний усекающего условия для автомата Бюхи. Представим это неравенство в виде конечного числа операций $(n' \sum_{(1 \leq i \leq m)} k_i) + (nz) < n \sum_{(1 \leq i \leq m)} k_i$. Заменяв операцию суммирования в обеих частях символом μ (суммарное число состояний для всех условий из множества Ψ) и разделив их на $(n\mu)$, получим $n'/n + z/\mu < 1$. Выполнение данного неравенства гарантирует ускорение при верификации модели с использованием описанного метода. Для оценки ускорения может быть использована следующая формула: $A = 1/(n'/n + z/\mu)$, где A – уско-

рение. Поскольку для эффективной работы метода требуется, чтобы знаменатель был меньше 1 и n' , как правило, в ограниченной модели всегда меньше n , определяющей является дробь z/μ и требуется контролировать, чтобы выполнялось неравенство $z/\mu < 1 - n'/n$, иначе использование вышеописанного метода неэффективно.

Проверим эффективность метода для вышеописанного примера с микроволновой печью. Исходное число состояний модели $n=10$ (рис. 2), число состояний модели после ограничения $n'=6$ (рис. 5). Число состояний усекающего условия $z=2$ (рис. 3), суммарное число состояний всех контрольных автоматов $\mu=2$ (рис. 6). Таким образом, ускорение $A=1/(6/10 + 2/2) = 0,625$, что свидетельствует о неэффективности применения данного метода в конкретном случае. Однако попробуем ввести еще 2 некоторых дополнительных условия проверки для этой системы, к примеру: «готовка когда-нибудь начнется» и «после закрытия дверцы задается режим готовки». Контрольные автоматы Бюхи для этих дополнительных условий будут состоять из 2 состояний каждый, и, таким образом, в конечном итоге при введении этих 2 дополнительных условий $\mu=6$, а итоговое ускорение $A=1/(6/10 + 2/6) = 1,07143$.

Поскольку ускорение в 0,07 не является показателем, возьмем некоторую абстрактную ситуацию. Пусть дана некоторая модель, состоящая из 10^{12} состояний. После применения к ней данного метода она сократилась до 10^{10} состояний. В качестве проверяемых условий будут даны 4 условия, описанные в [1]: свойства достижимости, безопасности, живости и справедливости. Суммарное число всех состояний автоматов Бюхи для этих свойств будет равно 8. Усекающий автомат будет состоять из 2 состояний. Таким образом, получится следующее ускорение: $A=1/(10^{10}/10^{12} + 2/8) = 3,846$. Если же проверяемых условий будет хотя бы 10, а суммарное число состояний всех автоматов Бюхи будет равно 20, получим следующее ускорение: $A=1/(10^{10}/10^{12} + 2/20) = 9,09$.

В таблице приведена зависимость ускорения A от значений n'/n и z/μ .

Зависимость ускорения от состояния модели

Acceleration depending on the model state

n'/n	z/μ										
	1/1	1/2	1/4	1/8	1/16	1/32	1/64	1/128	1/256	1/512	1/1024
1/1	0,5	0,667	0,8	0,889	0,941	0,97	0,985	0,992	0,996	0,998	0,999
1/2	0,667	1	1,333	1,6	1,778	1,882	1,939	1,969	1,984	1,992	1,996
1/4	0,8	1,333	2	2,667	3,2	3,556	3,765	3,879	3,938	3,969	3,984
1/8	0,889	1,6	2,667	4	5,333	6,4	7,111	7,529	7,758	7,877	7,938
1/16	0,941	1,778	3,2	5,333	8	10,67	12,8	14,22	15,06	15,52	15,75
1/32	0,97	1,882	3,556	6,4	10,67	16	21,33	25,6	28,44	30,12	31,03
1/64	0,985	1,939	3,765	7,111	12,8	21,33	32	42,67	51,2	56,89	60,24
1/128	0,992	1,969	3,879	7,529	14,22	25,6	42,67	64	85,33	102,4	113,8
1/256	0,996	1,984	3,938	7,758	15,06	28,44	51,2	85,33	128	170,7	204,8
1/512	0,998	1,992	3,969	7,877	15,52	30,12	56,89	102,4	170,7	256	341,3
1/1024	0,999	1,996	3,984	7,938	15,75	31,03	60,24	113,8	204,8	341,3	512

В заключение следует отметить, что целью разработки новой методики является необходимость сокращения количества проверяемых вычислительных последовательностей моделей больших и распределенных программных систем. В результате учета специфики задачи верификации данных систем был разработан описанный метод ограничений верифицируемых моделей, позволяющий выполнять верификацию информационных систем с учетом их характерных особенностей и полностью решающий поставленную задачу. Основная идея метода базируется на том, что он позволяет заметно сократить число вычислительных последовательностей системы, которые либо являются избыточными для верификации того или иного ее свойства, либо на них это свойство объективно не может выполняться в рамках реальных условий, позволяя тем самым упростить и ускорить процесс верификации.

На основании результатов анализа работы описанной методики можно сделать вывод, что ее применение эффективно лишь в тех случаях, когда суммарное число состояний всех контрольных автоматов Бюхи для проверяемых условий на модели удовлетворяет некоторому минимальному

значению, которое может быть вычислено согласно представленной ранее формуле.

Литература

1. Карпов Ю.Г. Model Checking. Верификация параллельных и распределенных программных систем. СПб: БХВ-Петербург, 2010. 552 с.
2. Суад Алагич, Майкл Арбиб. Проектирование корректных структурированных программ. М.: Радио и связь, 1979. 292 с.
3. Кораблин Ю.П., Косакян М.Л. Анализ моделей программ на языке асинхронных функциональных схем средствами темпоральной логики линейного времени // Программные продукты и системы. 2014. № 2 (106). С. 5–10.
4. Кораблин Ю.П. Семантика языков программирования. М.: Изд-во МЭИ, 1992. 100 с.
5. Ábrahám E., Havelund K. (Eds). Tools and Algorithms for the Construction and Analysis of Systems. Springer; 2014 edition (March 24, 2014), 652 p.
6. Bérard B., Bidoit M., Finkel A., Laroussinie F., Petit A., Petrucci L., Schnoebelen P. Systems and Software Verification. Model Checking Techniques and Tools. Springer, 2001, 190 p.
7. Kröger, Fred, Merz, Stephan. Temporal Logic and State Systems. Springer; 2008 ed. (March 27, 2008), 436 p.
8. Manna Z., Pnueli A. The Temporal Logic of Reactive and Concurrent Systems: Specification, 1992 edition (December 18, 1991), 427 p.
9. Khousainov B., Nerode A. Automata Theory and its Applications. Boston–Basel–Berlin, Birkhäuser Publ., 2001, 432 p.
10. Велдер С.А., Лукин М.А., Шалута А.А., Яминов Б.Р. Верификация автоматных программ. СПб: Наука, 2011. 244 с.

DOI: 10.15827/0236-235X.109.048-054

Received 23.03.15

THE LIMITATION METHOD FOR VERIFIABLE MODELS

Korablin Yu.P., Dr.Sc. (Engineering), Professor, y.p.k@mail.ru;

Shipov A.A., Postgraduate Student, a-j-a-1@yandex.ru

(Russian State Social University, Vilgelma Pika St. 4, Moscow, 129256, Russian Federation)

Abstract. Nowadays software systems become more and more complex and sophisticated, so we need to have some useful instruments to check if they fit target specifications, especially if we are talking about large and distributed software systems. The article demonstrates a new method which is the separate stage of verification process of information systems. It is intended to make verification more effective. Since the method is a separate verification step for Model Checking method and can be automated easily, its implementation to an existing technology is not a difficult task.

The method is based on the principle saying that the model of all systems contains redundant computational sequences. They are not necessary to be analyzed during the properties verification process and it does not affect the final result. The main goal of this method is to exclude these computational sequences from the original model before certain properties verification.

The article provides a detailed description of the proposed method and its efficiency estimation. The theoretical material of the article includes illustrative examples of the method, in particular for distributed software system.

The authors believe that using of this method can significantly reduce the number of analyzed sequences in verification process for a given system computational model, and thereby improve the performance of the process.

Keywords: verification, Kripke structure, Buchi automaton, temporal logic formula, LTL, CTL.

References

1. Karpov Yu.G. Model Checking. Verifikatsiya parallelnykh i raspredelennykh programmnykh sistem [Model Checking. Verification of parallel and distributed software systems]. St. Petersburg, BHV-Peterburg Publ., 2010, 552 p.
2. Alagich S., Arbib M. Proektirovanie korrektnykh strukturirovannykh program [Valid Structured Programs Design]. Moscow, Radio i svyaz Publ., 1979, 292 p.
3. Korablin Yu.P., Kosakyan M.L. Analysis of the program use the language of asynchronous functional schemes models using linear-time temporal logic. Programmnye produkty i sistemy [Software & Systems]. 2014, no. 2 (106), pp. 5–10 (in Russ.).
4. Korablin Yu.P. Semantika yazykov programmirovaniya [Programming Language Semantics]. Moscow, MEI Publ., 1992, 100 p.
5. Ábrahám E., Havelund K. (Eds.). Tools and Algorithms for the Construction and Analysis of Systems. Springer Publ., 2014, 652 p.
6. Bérard B., Bidoit M., Finkel A., Laroussinie F., Petit A., Petrucci L., Schnoebelen P. Systems and Software Verification. Model Checking Techniques and Tools. Springer Publ., 2001, 190 p.
7. Kröger F., Merz S. Temporal Logic and State Systems. Springer Publ., 2008, 436 p.
8. Manna Z., Pnueli A. The Temporal Logic of Reactive and Concurrent Systems: Specification. 1992, 427 p.
9. Khousainov B., Nerode A. Automata Theory and its Applications. Boston–Basel–Berlin, Birkhäuser Publ., 2001, 432 p.
10. Velder S.A., Lukin M.A., Shaluta A.A., Yaminov B.R. Verifikatsiya avtomatnykh program [Verification of automata-based program]. St. Petersburg, Nauka Publ., 2011, 244 p.