КИБЕРНЕТИКА

А.Н. ЧЕБОТАРЕВ

УДК 519.713.1

ВЕРИФИКАЦИЯ СПЕЦИФИКАЦИЙ В ЯЗЫКЕ L ОТНОСИТЕЛЬНО ТЕМПОРАЛЬНЫХ СВОЙСТВ, НЕ ВЫРАЗИМЫХ В ЭТОМ ЯЗЫКЕ

Ключевые слова: *верификация, реактивный алгоритм, язык спецификации L, темпоральные свойства, язык GR(1), сверхслова, открытая система.*

ВВЕДЕНИЕ

Одной из важных задач проектирования реактивных алгоритмов является обеспечение правильности исходной спецификации алгоритма, а точнее, требований к его функционированию. Это может быть достигнуто только путем формальной верификации необходимых свойств алгоритма. Задача верификации состоит в том, чтобы показать, что модель верифицируемой системы обладает заданным свойством корректности ее поведения. Верификация предполагает наличие формальной модели алгоритма и языка для задания проверяемых свойств. В качестве модели алгоритма рассматривается его спецификация в логическом языке. При этом специфицируемый алгоритм представляется в виде двух частей: управляющей и операционной, взаимодействующих между собой и с внешней средой. Логическая спецификация алгоритма определяет описание его управляющей части и тех аспектов операционной части и внешней среды, которые относятся к взаимодействию управляющей и операционной частей между собой и с внешней средой. Таким образом, логическая спецификация алгоритма состоит из трех частей: спецификаций управляющей части, операционной части и внешней среды. В качестве математической модели каждой части спецификации рассматривается конечный автомат. Составляющие части алгоритма удобно специфицировать как неинициальные системы, отдельно задавая начальное условие, определяющее начальные состояния соответствующих автоматов.

В настоящей работе предполагается, что для спецификации автоматов используется достаточно простой логический язык L [1]. Простота этого языка обусловлена необходимостью иметь эффективные процедуры синтеза императивного представления алгоритма, исходя из его декларативной спецификации. В качестве языков для описания проверяемых свойств широко применяются темпоральные логики. В этом случае для верификации наибольшее распространение получил теоретико-автоматный подход, при котором осуществляется проверка на пустоту языка, представимого произведением автоматных моделей для верифицируемого алгоритма и отрицания формулы, выражающей необходимое свойство. Недостатком такого подхода является высокая сложность используемых процедур, ограничивающая область его применения. В статье рассматриваются свойства, представимые формулами из несколько расширенного класса GR(1) [2] темпоральной логики с линейным временем LTL [3]. Для этого случая предлагается более

простой подход к верификации темпоральных свойств алгоритма, не требующий построения автоматных моделей. Хотя рассматриваемые свойства не выразимы в языке L, предлагаемый подход ограничивается эффективными средствами проверки непротиворечивости формул языка L [4, 5].

ЯЗЫК L СПЕЦИФИКАЦИИ РЕАКТИВНЫХ АЛГОРИТМОВ

Спецификация в языке L имеет вид формулы $\forall tF(t)$, интерпретируемой на множестве $\mathbf{Z}$ целых чисел. Формула $F(t)$ строится с помощью логических связок из атомарных формул (атомов) вида $p(t - k)$, где p — одноместный предикатный символ, t — переменная, принимающая значения из множества целых чисел, рассматриваемого как множество моментов дискретного времени, а k — натуральное число, называемое рангом атома. Разность между максимальным и минимальным значениями рангов атомов, встречающихся в формуле, называется ее глубиной.

Описание семантики языка L основано на рассмотрении его как формализма для задания множеств сверхслов в алфавите двоичных векторов, длина которых равна количеству различных предикатных символов языка. Следует отметить две наиболее распространенные трактовки символов этого алфавита. При первой трактовке символы алфавита представляют собой наборы значений двоичных переменных, соответствующих предикатным символам языка, в предположении, что они линейно упорядочены некоторым образом. При второй трактовке символами алфавита являются всевозможные подмножества множества Ω всех предикатных символов языка. Очевидно, что между алфавитами, трактуемыми первым и вторым способами, имеется взаимно однозначное соответствие. Поскольку такой алфавит однозначно определяется множеством Ω предикатных символов, в случаях, когда эту связь необходимо явно указать, будем его обозначать как $\Sigma(\Omega)$.

Пусть Σ — конечный алфавит, $\mathbf{Z}$ — множество целых чисел, а $\mathbf{N}^+ = \{z \in \mathbf{Z} \mid z > 0\}$. Отображения u: $\mathbf{Z} \to \Sigma$ и l: $\mathbf{N}^+ \to \Sigma$ называются соответственно двусторонним сверхсловом (обозначается $\ldots u(-2)u(-1)u(0)u(1)u(2)\ldots$) и сверхсловом (обозначается $l(1)l(2)\ldots$) в алфавите Σ . Отрезок $u(\tau)u(\tau+1)\ldots u(\tau+k)$ двустороннего сверхслова u обозначается $u(\tau, \tau + k)$. Бесконечный отрезок $u(k+1, \infty)$ назовем k-суффиксом двустороннего сверхслова u. Если значение k не существенно, то будем говорить об w-суффиксе. Множество всех сверхслов в алфавите S обозначается Σ^{ω}, а двустороннее сверхслово (сверхслово), представляющее собой бесконечное повторение одного и того же непустого слова r, обозначим $r^{\mathbf{Z}}$ (r^{ω}).

Каждой формуле $F = \forall tF(t)$ языка L ставится в соответствие множество всех моделей для этой формулы, т.е. множество таких интерпретаций, на которых F истинна. Пусть $\Omega = \{p_1, \ldots, p_m\}$ — множество всех предикатных символов, встречающихся в формуле F (сигнатура формулы F). Интерпретация формулы F — это упорядоченный набор определенных на $\mathbf{Z}$ одноместных предикатов $\pi_1, \ldots, \pi_m$, соответствующих предикатным символам из Ω. Пусть Σ — множество всех двоичных векторов длины m, тогда интерпретацию $I = <\pi_1, \ldots, \pi_m>$ можно представить в виде двустороннего сверхслова в алфавите Σ , а множество всех моделей для F — в виде множества $M(F)$ двусторонних сверхслов в этом алфавите. В дальнейшем не будем различать интерпретации и соответствующие им двусторонние сверхслова, поэтому будем говорить об истинности формулы F на двустороннем сверхслове $u \in \Sigma^{\mathbf{Z}}$ и значении формулы $F(t)$ в некоторой позиции τ двустороннего сверхслова u, понимая под этим значение формулы $F(\tau)$ в интерпретации u. Сверхсловарная семантика формулы $F = \forall tF(t)$ определяется множеством сверхслов $W(F)$, представляющим собой множество всех ω-суффиксов двусторонних сверхслов из $M(F)$.

При интерпретации формул вида $\forall tF(t)$ на множестве целых чисел для любого $k \in \mathbf{Z}$ справедлива эквивалентность $\forall tF(t) \Leftrightarrow \forall tF(t+k)$, где $F(t+k)$ обозначает фор-

мулу, полученную из $F(t)$ путем добавления k к рангам всех ее атомов (сдвиг на k). Таким образом, можно ограничиться рассмотрением формул, у которых максимальный ранг атомов равен 0. Такие формулы будем называть нормализованными вправо. Смысл понятия глубины формулы состоит в том, что истинностное значение нормализованной вправо формулы $F(t)$ глубины r в позиции τ интерпретации u определяется отрезком $u(\tau–r, \tau)$ соответствующего двустороннего сверхслова u.

СПОСОБЫ ПРОВЕРКИ НЕПРОТИВОРЕЧИВОСТИ СПЕЦИФИКАЦИЙ В ЯЗЫКЕ L

Для описания рассматриваемых здесь способов проверки непротиворечивости формулы $\forall tF(t)$ языка L удобно воспользоваться понятием пространства состояний, ассоциируемого с этой формулой [4]. Пусть Ω — сигнатура формулы $F(t)$, а r — ее глубина. Последовательность $s_0, s_1, \ldots, s_r$ векторов из $\Sigma(\Omega)$ назовем состоянием глубины r, а множество $Q(r, \Omega)$ всех таких последовательностей — пространством состояний глубины r для формулы $F(t)$. На множестве $Q(r, \Omega)$ определим отношение N непосредственного следования так, что за каждым состоянием $q = s_0, s_1, \ldots, s_r$ непосредственно следуют $2^{|\Omega|}$ состояний вида $s_1, \ldots, s_r, s$, где $s \in \Sigma(\Omega)$. Отношение, обратное N, обозначим P и будем называть отношением непосредственного предшествования. Очевидно, что состоянию $s_0, s_1, \ldots, s_r$ непосредственно предшествуют $2^{|\Omega|}$ состояний вида $s, s_0, \ldots, s_{r-1}$, где $s \in \Sigma(\Omega)$. Пусть $Q_1 \subseteq Q(r, \Omega)$. Обозначим $N(Q_1)$ множество всех состояний, непосредственно следующих за состояниями из Q_1, а $P(Q_1)$ — аналогичное множество для отношения P. Если компоненты вектора s_i в состоянии $q = s_0, s_1, \ldots, s_r$ рассматривать как истинностные значения соответствующих атомов ранга $i - r$ при некотором упорядочении множества Ω, то можно говорить о значении формулы $F(t)$ на состоянии q. Формулу $F(t)$ будем рассматривать как способ задания множества $Q(F(t))$ состояний из $Q(r, \Omega)$, а именно, тех состояний, на которых она истинна. Ограничения отношений N и P на множество $Q(F(t))$ обозначим соответственно N_F и P_F.

Пусть $G(F)$ — граф отношения N_F. Граф $G = <V, E>$, где V — множество вершин, а E — множество дуг графа, назовем циклическим, если для каждой его вершины q существуют дуги (q_1, q) и (q, q_2), принадлежащие E. Спецификация $F = \forall tF(t)$ непротиворечива тогда и только тогда, когда граф $G(F)$ имеет непустой циклический подграф [4]. Множество состояний (вершин) максимального такого подграфа $G^*(F)$ графа $G(F)$ назовем ядром множества $Q(F(t))$.

Понятие состояния можно определить как отрезок длины $r + 1$ сверхслова, соответствующего интерпретации формулы глубины r. Если формула глубины r истинна на состоянии q, соответствующем отрезку $u(\tau, \tau + r)$ интерпретации u, то она истинна в позиции $\tau + r$ этой интерпретации. Каждая модель u для формулы $F = \forall tF(t)$ однозначно определяет такое двустороннее сверхслово состояний $\ldots q_{-2}, q_{-1}, q_0, q_1, q_2, \ldots$, что для каждого $i \in \mathbf{Z}$ $q_{i+1} \in N(q_i)$, и наоборот, каждое такое двустороннее сверхслово состояний, принадлежащих $Q(F(t))$, однозначно определяет модель для формулы F.

Утверждение 1. Пусть Q^* — ядро множества состояний $Q(F(t))$, тогда для каждого состояния q из Q^* существует модель u для формулы $\forall tF(t)$, содержащая отрезок, совпадающий с состоянием q.

Пусть Q' и Q'' — такие подмножества множества $Q(F(t))$, что $P_F(Q') = Q'$ и $N_F(Q'') = Q''$. Для проверки непустоты графа $G^*(F)$ достаточно построить одно из множеств — Q' или Q''. Таким образом, эта проверка может быть выполнена одним из следующих алгоритмов. Первый алгоритм состоит в построении последовательности множеств состояний: $Q_0 = Q(F(t))$, $Q_1 = Q_0 \cap N(Q_0)$, …, $Q_{i+1} = Q_i \cap N(Q_i)$, … до тех пор, пока для некоторого k не выполнится $Q_{k-1} = Q_k$. Если при этом $Q_k \neq \varnothing$,

то множество вершин графа $G^*(F)$ также не пусто, и наоборот, $Q_k = \varnothing$ свидетельствует о его пустоте. Второй алгоритм использует отношение P и аналогичным образом строит последовательность: $Q_0 = Q(F(t))$, $Q_1 = Q_0 \cap P(Q_0)$, …, $Q_{i+1} = Q_i \cap P(Q_i)$, … . Условия окончания и результат определяются так же, как и для первого алгоритма. Результаты применения данных алгоритмов к множеству состояний Q обозначим соответственно $N^*(Q)$ и $P^*(Q)$. Рассмотрим некоторые свойства этих множеств.

Утверждение 2. Справедливо равенство $N^*(P^*(Q)) = P^*(N^*(Q)) = Q^*$, где Q^* — ядро множества состояний Q.

Утверждение 3. Каждое состояние множества $N^*(Q)$ достижимо из некоторого состояния этого множества, т.е. из $q \in N^*(Q)$ следует $P(q) \neq \varnothing$.

Утверждение 4. Из каждого состояния множества $P^*(Q)$ достижимо некоторое состояние этого множества, другими словами, из каждого состояния множества $P^*(Q)$ достижимо его ядро.

Лемма 1. Пусть $Q_1 \subseteq Q$, тогда $N^*(Q_1) \subseteq N^*(Q)$ и $P^*(Q_1) \subseteq P^*(Q)$.

Справедливость этой леммы вытекает из следующих очевидных равносильностей: $N(Q_1 \cup Q_2) = N(Q_1) \cup N(Q_2)$ и $P(Q_1 \cup Q_2) = P(Q_1) \cup P(Q_2)$.

Утверждение 5. Пусть $Q_1 \subseteq N^*(Q)$, тогда $P^*(Q_1) \subseteq Q^*$, где Q^* — ядро множества Q.

Это утверждение следует на основании леммы 1 из $P^*(N^*(Q)) = Q^*$ и $Q_1 \subseteq \subseteq N^*(Q)$.

Обозначим $P(F(t))$ формулу, задающую множество всех тех состояний из $Q(r, \Omega)$, которые непосредственно предшествуют состояниям из множества $Q(F(t))$, а $N(F(t))$ — формулу, задающую множество всех состояний, непосредственно следующих за состояниями из этого множества. Как показано в [4], для $F(t)$, заданной в д.н.ф., $P(F(t))$ получается, если в каждой элементарной конъюнкции, входящей в $F(t)$, удалить все литеры нулевого ранга и полученную д.н.ф. сдвинуть на 1 вправо, т.е. увеличить ранги всех атомов на 1. Аналогично, $N(F(t))$ получается, если в каждой элементарной конъюнкции, входящей в д.н.ф. формулы $F(t)$, удалить все литеры минимального ранга (т.е. ранга $-r$, если таковые имеются) и полученную д.н.ф. сдвинуть на 1 влево. Пересечению множеств состояний соответствует конъюнкция задающих их формул. Таким образом, алгоритмы проверки непротиворечивости формулы $\forall t F(t)$ состоят в итеративном выполнении операции $F(t)\&P(F(t))$ или $F(t)\&N(F(t))$ до стабилизации формулы. Результаты применения этих алгоритмов к формуле $F(t)$ обозначим соответственно $P^*(F(t))$ и $N^*(F(t))$.

При задании формулы $F(t)$ в виде к.н.ф. первый алгоритм сводится к методу R-резолюции [4], для которого получены различные усовершенствования [6, 7].

СВОЙСТВА СПЕЦИФИКАЦИЙ И СПОСОБЫ ИХ ЗАДАНИЯ

Всякая спецификация реактивного алгоритма характеризует совокупность его свойств, выраженных в виде множества сверхслов в алфавите, определяемом спецификацией. Пусть Σ — конечный алфавит. Свойством над Σ назовем произвольное подмножество множества Σ^ω. Так, формула F языка L сигнатуры Ω задает свойство, определяемое множеством сверхслов $W(F)$ в алфавите $\Sigma(\Omega)$. Спецификация F, с которой ассоциируется алфавит Σ, обладает свойством $S \subseteq \Sigma^\omega$ тогда и только тогда, когда $W(F) \subseteq S$. Будем говорить, что модель формулы F обладает свойством S, если все ее w-суффиксы содержатся в S.

Под верификацией спецификации реактивного алгоритма понимается автоматическая проверка наличия у нее требуемых свойств. Свойства, выразимые в языке L, нет необходимости проверять, поскольку соответствующие им формулы должны быть включены в спецификацию алгоритма и полученный в результате синтеза алгоритм будет заведомо обладать этими свойствами. Поэтому при верификации спе-

цификаций в языке L проверяются свойства, не выразимые в этом языке. Для задания таких свойств широкое распространение получили темпоральные логики. В настоящей работе рассматриваются свойства, задаваемые формулами темпоральной логики с линейным временем (LTL), принадлежащими классу GR(1) [2].

Формулы рассматриваемой темпоральной логики строятся из символов атомарных высказываний, принадлежащих множеству $\Omega = \{p_1, p_2, \dots, p_m\}$, с помощью пропозициональных связок и унарных темпоральных операторов **X**, **G**, **F**. В качестве области интерпретации выступает множество всех сверхслов в алфавите $\Sigma(\Omega)$. Здесь символы алфавита удобно трактовать как подмножества множества W. Для определения семантики формулы используется понятие ее истинностного значения в позиции i сверхслова, соответствующего интерпретации $\sigma \in (\Sigma(\Omega))^{\omega}$.

Приведем индуктивное определение истинности формулы φ в позиции i сверхслова $\sigma = \sigma_1\sigma_2\sigma_3\dots$.

Формула p, где $p \in \Omega$, истинна тогда и только тогда, когда $p \in \sigma_i$.

Формула **X**φ истинна тогда и только тогда, когда φ истинна в позиции $i + 1$ сверхслова σ.

Формула **G**φ истинна тогда и только тогда, когда φ истинна во всех позициях $j \geq i$ сверхслова σ.

Формула **F**φ истинна тогда и только тогда, когда существует такое $j \geq i$, что формула j истинна в позиции j сверхслова σ.

Значение истинности формул $\neg\varphi_1$, $\varphi_1 \vee \varphi_2$, $\varphi_1 \& \varphi_2$ определяется обычным образом.

Формула φ истинна в интерпретации σ, если она истинна в позиции 1 соответствующего сверхслова. Таким образом, каждая формула φ задает множество $W(\varphi)$ сверхслов в алфавите $\Sigma(\Omega)$, т.е. множество всех тех сверхслов, на которых она истинна.

Формулы из класса GR(1) имеют вид $(\mathbf{GF}\varphi_1 \,\&\, \dots \,\&\, \mathbf{GF}\varphi_m) \rightarrow (\mathbf{GF}\psi_1 \,\&\, \dots \,\&\, \mathbf{GF}\psi_n)$, где формулы φ_i ($i = 1, \dots, m$) č ψ_j ($j = 1, \dots, n$) построены из атомарных высказываний с помощью пропозициональных связок и темпорального оператора **X**.

Для верификации спецификаций в языке L удобно иметь один и тот же язык для представления спецификаций и их свойств. В качестве такого языка будем использовать язык L_{max}, формулы которого интерпретируются на множестве **Z** целых чисел. Это язык логики предикатов первого порядка с одноместными предикатами из множества $\Omega = \{p_1, \dots, p_m\}$ и одним двуместным предикатом ≤, интерпретируемым как отношение линейного порядка на **Z**. Множество термов языка имеет вид $T = \{(t_i + k) \mid t_i \in V, k \in \mathbf{Z}\}$, ćääå $V = \{t, t_1, t_2, \dots\}$ — множество предметных переменных. Имеется два типа атомарных формул: формулы вида $p(\tau)$, где $p \in \Omega$, ą $\tau \in T$, и формулы вида $(\tau_1 \leq \tau_2)$, где $\tau_1, \tau_2 \in T$. Так же, как и для языка L, с каждой формулой F языка L_{max} ассоциируется множество $W(F)$ сверхслов в алфавите $\Sigma(\Omega)$.

Определим отображение ξ формул языка GR(1) в формулы L_{max}:

$\xi(p) = p(t)$, где $p \in \Omega$;

$\xi(\mathbf{X}\varphi) = F(t+1)$, где $F(t) = x(j)$, например, $\xi(\mathbf{X}p) = p(t+1)$, ą $\xi(\mathbf{XXX}p) = p(t+3)$;

$\xi(\neg\varphi) = \neg(\xi(\varphi))$;

если * — произвольная двуместная пропозициональная связка, то $\xi(\varphi_1 * \varphi_2) = \xi(\varphi_1) * \xi(\varphi_2)$;

$\xi(\mathbf{GF}\varphi_1) = \forall t \exists t_1 (t_1 \geq t) \& F_1(t_1)$, а $\xi(\mathbf{FG}\varphi_1) = \exists t \forall t_1 ((t_1 \geq t) \rightarrow F_1(t_1))$, где $F_1(t) = \xi(\varphi_1)$.

Можно показать, что если $\varphi \in$ GR(1), ņī $W(\varphi) = W(\xi(\varphi))$, т.е. формулы φ č ξ(φ) определяют одно и то же свойство. Это позволяет при верификации вместо формул языка GR(1) использовать соответствующие формулы языка L_{max}, интерпретируемые на множестве целых чисел.

В заключение раздела рассмотрим формулу $\mathbf{G}(\varphi_1 \to \mathbf{F}\varphi_2)$, где φ_1 и φ_2 не содержат операторов **G** и **F**. Эта формула не принадлежит классу GR(1), однако выражает важный класс свойств, которые часто необходимо проверять при верификации спецификаций алгоритмов. Формула языка L_{max}, задающая это же свойство, имеет вид $\forall t(F_1(t) \to \exists t_1(t_1 \geq t)\&F_2(t_1))$, где $F_1(t) = \xi(\varphi_1)$, а $F_2(t) = \xi(\varphi_2)$. Далее будет показано, что предлагаемый подход может использоваться для проверки и такого рода свойств.

ПРОВЕРКА СВОЙСТВ БЕЗ УЧЕТА ИНФОРМАЦИИ О СРЕДЕ

Наиболее простое свойство φ, выразимое в GR(1) и не выразимое в языке L, имеет вид $\mathbf{GF}\varphi_1$. В языке L_{max} ему соответствует формула $\forall t \exists t_1(t_1 \geq t)\&F_1(t_1)$, где $F_1(t)$ не содержит кванторов. Всякое сверхслово, принадлежащее этому свойству, имеет бесконечно много позиций, в которых истинна формула $F_1(t)$. Таким образом, спецификация $F = \forall t F(t)$ не обладает этим свойством тогда и только тогда, когда для нее существует такая модель *u,* что, начиная с некоторой позиции τ, формула $\neg F_1(t)$ истинна на *u* для всех $t \geq \tau$. Пусть глубина формулы $F(t)$ равна *r*, тогда в интерпретации *u* найдутся такие две позиции τ_i и τ_j $(\tau < \tau_i < \tau_j)$, что $u(\tau_i - r,\ \tau_i) = u(\tau_j - r,\ \tau_j)$. Двустороннее сверхслово $(u(\tau_i + 1,\ \tau_j))^{\mathbf{Z}}$ также является моделью для *F*. Эта модель обладает свойством $\forall t \neg F_1(t)$, поэтому из того, что существует модель, не обладающая свойством φ, следует, что существует модель, обладающая свойством $\forall t \neg F_1(t)$. В то же время, если существует модель, обладающая свойством $\forall t \neg F_1(t)$, то очевидно, что спецификация *F* не обладает свойством φ. Итак, спецификация *F* не обладает свойством φ тогда и только тогда, когда формула $\forall t F(t)\&\neg F_1(t)$ непротиворечива. Поскольку это формула языка L, ее непротиворечивость легко проверяется, например, методом *R*-резолюции [4].

Как уже отмечалось, важное значение имеет свойство, выражаемое формулой $\varphi = \forall t(F_1(t) \to \exists t_1(t_1 \geq t)\&F_2(t_1))$, где $F_1(t)$ и $F_2(t_1)$ — формулы языка L, не содержащие кванторов.

Сначала рассмотрим ситуацию, в которой $F_1(t)$ сохраняет истинное значение вплоть до момента t_1, когда становится истинной формула $F_2(t)$. Более точно эта ситуация выражается формулой $\forall t(F_1(t)\&\neg F_2(t)\&\neg F_2(t+1) \to F_1(t+1))$. В этом случае свойство φ может быть заменено свойством $\varphi' = \forall t \exists t_1(t_1 \geq t)\&(\neg F_1(t_1) \vee F_2(t_1))$, утверждающим, что формула $F_1(t)\&\neg F_2(t)$ не сохраняет истинное значение бесконечно долго. Таким образом, задача сводится к рассмотренной выше проверке свойства вида $\mathbf{GF}\varphi_1$, т.е. к проверке противоречивости формулы $\forall t F(t)\&F_1(t)\&\neg F_2(t)$.

Рассмотрим теперь случай, когда $F_1(t)$ не сохраняет истинное значение до момента t_1. Спецификация *F* не обладает свойством φ тогда и только тогда, когда для нее существует модель с ω-суффиксом, в начальной позиции которого истинна $F_1(t)$, а во всех остальных позициях, включая начальную, ложна $F_2(t)$. Определить наличие такой модели можно следующим образом. Как описано ранее, строим формулу $N^*(F(t))$ и умножаем ее на $\neg F_2(t)$. Затем для полученной формулы $F'(t) = N^*(F(t))\&\neg F_2(t)$ строим формулу $P^*(F'(t))$ и умножаем ее на $F_1(t)$. Спецификация $\forall t F(t)$ не обладает свойством φ тогда и только тогда, когда это произведение не равно тождественно 0. Покажем справедливость этого утверждения. Пусть Q' — множество состояний, задаваемое формулой $P^*(F'(t))$.

Если конъюнкция $P^*(F'(t))\&F_1(t)$ не равна тождественно 0, то Q' содержит состояние *q*, на котором истинна формула $F_1(t)\&\neg F_2(t)$. Из утверждений 1 и 5 следует, что для спецификации *F* существует модель *u*, в некоторой позиции τ которой истинна формула $F_1(t)\&\neg F_2(t)$. Из утверждения 4 следует, что из состояния *q* достижимо ядро множества Q'. Таким образом, для спецификации *F*

существует модель, в позиции τ которой истинна формула $F_1(t)\&\neg F_2(t)$, а во всех остальных позициях ложна формула $F_2(t)$. Это означает, что спецификация F не обладает свойством φ.

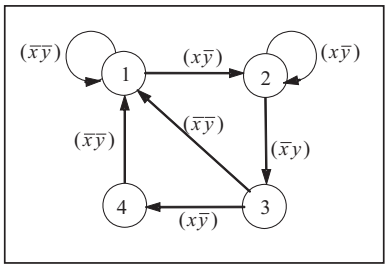

Рис. 1

Пусть состояние q, на котором истинна формула $F_1(t)$, не принадлежит множеству Q', т.е. $P^*(F'(t))\&F_1(t) = 0$. Тогда либо на этом состоянии истинна формула $F_2(t)$, либо из него не достижимо ядро множества Q', т.е. любая модель, содержащая отрезок, соответствующий состоянию q, не имеет ω-суффикса, в начальной позиции которой истинна формула $F_1(t)\&\neg F_2(t)$, а во всех остальных позициях ложна формула $F_2(t)$. Таким образом, если конъюнкция $P^*(F'(t))\&F_1(t)$ равна тождественно 0, то спецификация обладает свойством φ.

Пример. Формула $F(t)$ в спецификации $F = \forall tF(t)$ имеет вид

$$\neg y(t-2)x(t-1)\neg y(t-1)(x(t)\neg y(t)\vee\neg x(t)y(t)) \vee$$
$$\vee \neg x(t-1)\neg y(t) \vee y(t-2)\neg y(t-1)\neg x(t)\neg y(t),$$

а проверяемое свойство φ задано формулой $\forall t(x(t) \to \exists t_1(t_1 \geq t)y(t_1))$. Автомат, специфицируемый формулой F, приведен на рис. 1.

Процесс верификации выглядит следующим образом.

В соответствии с приведенным алгоритмом строим сначала формулу $N^*(F(t))$:

$$N(F(t)) = x(t-2)\neg y(t-2)(x(t-1)\neg y(t-1)\vee\neg x(t-1)y(t-1))\vee \neg x(t-2)\neg y(t-1)\vee$$
$$\vee \neg y(t-2)\neg x(t-1)\neg y(t-1);$$
$$F^1(t) = N(F(t)) \,\&\, F(t) = \neg x(t-2)\neg x(t-1)\neg y(t-1)\neg y(t)\vee$$
$$\vee \neg y(t-2)x(t-1)\neg y(t-1)(x(t)\neg y(t) \vee \neg x(t)y(t))\vee x(t-2)\neg y(t-2)\neg x(t-1)\neg y(t) \vee$$
$$\vee \neg x(t-2)y(t-2)\neg y(t-1)\neg x(t)\neg y(t);$$
$$N(F^1(t)) = x(t-2)\neg y(t-2)(x(t-1)\neg y(t-1) \vee \neg x(t-1)y(t-1))\vee\neg x(t-2)\neg y(t-1) \vee$$
$$\vee \neg y(t-2)\neg x(t-1)\neg y(t-1).$$

Поскольку $N(F^1(t)) = N(F(t))$, на этом процесс построения $N^*(F(t))$ заканчивается и $N^*(F(t)) = F^1(t)$. Умножение $F^1(t)$ на $\neg y(t)$ дает следующую формулу $F'(t)$:

$$F'(t) = F^1(t)\&\neg y(t) = \neg x(t-2)\neg x(t-1)\neg y(t-1)\neg y(t)\vee\neg y(t-2)x(t-1)\neg y(t-1)x(t)\neg y(t)\vee$$
$$\vee x(t-2)\neg y(t-2)\neg x(t-1)\neg y(t) \vee \neg x(t-2)y(t-2)\neg y(t-1)\neg x(t)\neg y(t).$$

Далее строим формулу $P^*(F'(t))$:

$$P(F'(t)) = \neg x(t-1)\neg x(t)\neg y(t) \vee \neg y(t-1)x(t)\neg y(t)\vee x(t-1)\neg y(t-1)\neg x(t)\vee$$
$$\vee \neg x(t-1)y(t-1)\neg y(t).$$

Несложно убедиться, что формула $P(F'(t))\&F'(t)$ эквивалентна $F'(t)$, поэтому процесс построения $P^*(F'(t))$ заканчивается и $P^*(F'(t)) = F'(t)$.

Произведение $F'(t)$ на $x(t)$ не равно тождественно 0, и, следовательно, верифицируемая спецификация F не обладает свойством φ. Конец примера.

Верифицируемый алгоритм обладает свойством $\varphi = \varphi_1\&\varphi_2$ тогда и только тогда, когда он обладает свойством φ_1 и обладает свойством φ_2, поэтому проверка свойства φ сводится к двум независимым проверкам свойства φ_1 и свойства φ_2.

ВЕРИФИКАЦИЯ ОТКРЫТЫХ СИСТЕМ

При верификации открытых систем [8] достаточно ограничиться рассмотрением двух компонентов такой системы: собственно алгоритма и его среды. При этом чаще всего в качестве верифицируемого алгоритма выступает спецификация его управляющей части, а в качестве среды — объединенная спецификация операционной части и внешней среды. Проверяемые свойства, как правило, имеют вид $\varphi^e \to \varphi^s$, где φ^e — свойство среды, а φ^s — свойство верифицируемого алгоритма. Формулам φ^e и φ^s в языке GR(1) соответствуют конечные конъюнкции формул языка L и формул вида $\forall t \exists t_1 (t_1 \geq t) \& F_1(t_1)$. Рассмотрим сначала случай, когда формулы φ^e и φ^s имеют соответственно вид $\forall t \exists t_1 (t_1 \geq t) \& F_1(t_1)$ и $\forall t \exists t_1 (t_1 \geq t) \& F_2(t_1)$. Спецификация не обладает свойством $\varphi^e \to \varphi^s$ тогда и только тогда, когда она имеет хотя бы одну модель, не обладающую этим свойством, т.е. обладающую свойством $\varphi^e \& \neg \varphi^s$. Вообще говоря, утверждения «модель не обладает свойством φ» и «модель обладает свойством $\neg\varphi$» не равносильны. Свойство S называется суффиксно замкнутым, если из того, что сверхслово принадлежит S, следует, что любой его ω-суффикс также принадлежит S. Учитывая, что свойства, выразимые формулами $\mathbf{GF}\varphi$ и $\mathbf{FG}\neg\varphi$, суффиксно замкнуты, несложно показать, что свойства, выразимые формулами языка GR(1), и их дополнения суффиксно замкнуты. Для таких свойств модель не обладает свойством φ тогда и только тогда, когда она обладает свойством $\neg\varphi$. Таким образом, спецификация не обладает свойством $\varphi^e \to \varphi^s$ тогда и только тогда, когда для нее существует модель с ω-суффиксом, обладающим следующими свойствами: а) во всех его позициях истинна формула $\neg F_2(t)$; б) имеется бесконечное количество позиций, в которых истинна формула $F_1(t)$. Проверка существования такой модели осуществляется следующим образом. Пусть Q — множество состояний, задаваемое формулой $F(t) \& \neg F_2(t)$. Выделим в Q все состояния, на которых истинна $F_1(t)$, что на уровне формул соответствует формуле $F(t) \& \neg F_2(t) \& F_1(t)$. Если эта формула тождественно равна нулю, то процесс верификации заканчивается с положительным результатом. В противном случае необходимо определить, содержит ли множество состояний Q_1, задаваемое рассматриваемой формулой, состояние, достижимое из себя. Если Q_1 содержит такое состояние, то оно принадлежит ядру множества Q, а это означает, что для верифицируемой спецификации существует модель, являющаяся моделью для формулы $\forall t \neg F_2(t)$ и имеющая ω-суффикс с бесконечным количеством позиций, в которых истинна формула $F_1(t)$. Если такой модели не существует, то никакая модель не обладает свойством $\varphi^e \& \neg \varphi^s$ и, следовательно, спецификация обладает проверяемым свойством. Для проверки наличия в множестве Q_1 такого состояния строим множество всех тех состояний из Q, которые достижимы (в смысле отношения N) из Q_1, и берем пересечение этого множества с Q_1. Возможны три ситуации:

1) если пересечение пусто, то Q_1 не содержит состояния, достижимого из себя;

2) если пересечение совпадает с Q_1, то Q_1 содержит такое состояние;

3) если пересечение не пусто и равно $Q_2 \subseteq Q_1$, где Q_2 не совпадает с Q_1, то переходим к проверке наличия в Q_2 состояния, достижимого из себя, и т.д.

Проверка свойства завершается при возникновении первой или второй ситуации.

Осталось уточнить процедуру построения множества всех тех состояний из Q, которые достижимы из заданного множества $Q_1 \subseteq Q$. Обозначим это множество $N^+(Q_1)$. В основе процедуры лежит итеративное применение операции $N(Q_i)$. Сначала получаем множество $Q_2 = N(Q_1) \cap Q$, затем для $i = 2, 3, \ldots$ строим последовательность $Q_{i+1} = (N(Q_i) \cap Q) \cup Q_i$, пока не получим $Q_{i+1} = Q_i = N^+(Q_1)$. На уровне формул последняя операция имеет вид $F_{i+1}(t) = N(F_i(t)) \& F(t) \vee F_i(t)$.

Возможны модификации этого алгоритма, например, его можно начинать с проверки свойства φ^s, т.е. с проверки противоречивости формулы $\forall t F(t) \& \neg F_2(t)$. Если спецификация обладает этим свойством, то проверка заканчивается. Если проверяемая формула непротиворечива, формула, полученная в результате проверки, умножается на $F_1(t)$ и дальше процесс протекает так, как описано ранее.

Рассмотрим свойство $\varphi_1 \to \varphi_2 \& \varphi_3$. Поскольку $(\varphi_1 \to \varphi_2 \& \varphi_3) \Leftrightarrow (\varphi_1 \to \varphi_2) \& (\varphi_1 \to \to \varphi_3)$, его проверка сводится к проверке двух свойств рассмотренного ранее вида.

Пусть теперь проверяемое свойство имеет вид $\varphi_1 \& \varphi_2 \to \varphi_3$, где $\varphi_i = \forall t \exists t_1 (t_1 \geq \geq t) \& F_i(t_1)$ $(i = 1, 2, 3)$. Спецификация обладает этим свойством тогда и только тогда, когда все модели, имеющие ω-суффикс, во всех позициях которого истинна формула $\neg F_3(t)$, имеют либо ω-суффикс, во всех позициях которого истинна $\neg F_1(t)$, либо ω-суффикс, во всех позициях которого истинна $\neg F_2(t)$. Таким образом, спецификация не обладает свойством $\varphi_1 \& \varphi_2 \to \varphi_3$ тогда и только тогда, когда в множестве состояний, задаваемых формулой $F(t) \& \neg F_3(t)$, имеются два состояния — q_1 и q_2, на которых истинны соответственно формулы $F_1(t)$ и $F_2(t)$ и такие, что q_2 достижимо из q_1, а q_1 достижимо из q_2. Пусть Q — множество состояний, задаваемое формулой $F(t) \& \neg F_3(t)$, а Q_{10} и Q_{20} — множества всех тех состояний из Q, на которых истинны соответственно $F_1(t)$ и $F_2(t)$. Строим множество всех тех состояний из Q, которые достижимы из Q_{10} $(N^+(Q_{10}))$, и берем его пересечения с Q_{10} и Q_{20}. В результате получим множества $Q_{11} \subseteq Q_{10}$ и $Q_{21} \subseteq Q_{20}$. Затем строим множество $N^+(Q_{21})$ и берем его пересечения с Q_{11} и Q_{21}, что дает множества $Q_{12} \subseteq \subseteq Q_{10}$ и $Q_{22} \subseteq Q_{20}$, и т.д. В итоге будут получены две последовательности: $Q_{10} \supseteq \supseteq Q_{11} \supseteq Q_{12} \supseteq \dots$ и $Q_{20} \supseteq Q_{21} \supseteq Q_{22} \supseteq \dots$. Процесс заканчивается, когда хотя бы одно из множеств этих последовательностей будет пусто либо в каждой последовательности два смежных множества будут равны. В первом случае спецификация обладает свойством $\varphi_1 \& \varphi_2 \to \varphi_3$, во втором — не обладает. Очевидно, что приведенный алгоритм легко распространяется на проверку свойства, задаваемого импликацией с $k > 2$ сомножителями в левой части. Обозначим их $\varphi_0, \varphi_1, \dots, \varphi_{k-1}$, а множества состояний из Q, на которых истинны соответственно формулы $F_0(t), F_1(t), \dots, F_{k-1}(t)$, — как $Q_{00}, Q_{10}, \dots, Q_{(k-1)0}$. Процесс дальнейшего вычисления состоит в построении для $j = 0, 1, \dots, k-1$ последовательностей $Q_{j0} \supseteq Q_{j1} \supseteq Q_{j2} \supseteq \dots$. На i-й итерации вычисляются очередные k членов этих последовательностей в соответствии с формулой $Q_{j(i+1)} = N^+(Q_{(i \bmod k)i}) \cap \cap Q_{ji}$ $(j = 0, 1, \dots, k-1)$. Условия окончания процесса те же, что и выше.

Вычисление всех состояний, достижимых из заданного множества состояний, представляет собой незначительную модификацию проверки непротиворечивости, основанной на итеративном вычислении $N(F(t))$ для формулы $F(t)$ языка L. Таким образом, легко видеть, что проверка свойства, заданного формулой языка GR(1), сводится к проверкам непротиворечивости формул языка L.

ЗАКЛЮЧЕНИЕ

Рассмотрены методы верификации спецификаций реактивных алгоритмов относительно свойств, выразимых формулами из класса GR(1) темпоральной логики LTL. Следует подчеркнуть, что речь идет о верификации декларативных спецификаций, представленных в виде формул логического языка, а не императивных представлений алгоритмов в терминах состояний и отношений переходов. Процесс верификации сводится к преобразованию формул языка L, рассматриваемых как пропозициональные формулы, в качестве переменных которых выступают атомы языка L. Для этой цели используются средства для проверки непротиворечивости спецификаций в языке L и некоторые другие средства проектирования алгоритмов, рассматриваемых как открытые системы.

Хотя в статье речь идет о формулах из класса GR(1), нетрудно заметить, что на самом деле рассматривается более широкий класс темпоральных формул. Так, в подформулах φ_i формул вида $\mathbf{GF}\varphi_i$ допускается использование оператора $\mathbf{X}$, а в качестве множителей, составляющих формулу φ^s, могут выступать также формулы вида $\mathbf{G}(\varphi_1 \to \mathbf{F}\varphi_2)$. Возможны и другие расширения класса GR(1), однако все они ограничены формулами, задающими суффиксно замкнутые свойства, поскольку только для таких формул существуют формулы языка L_{max}, задающие те же свойства.

Отметим еще одну особенность предложенного подхода к верификации, состоящую в том, что предлагается не универсальный алгоритм, пригодный для любых формул из класса GR(1), а набор алгоритмов, используемых для базовых формул. Верификация сложной формулы сводится к последовательности процедур, верифицирующих свойства, соответствующие структурным составляющим формулы общего вида. Можно построить универсальный алгоритм проверки свойств, задаваемых формулами из класса GR(1), не зависящий от вида этих формул, однако в этом случае процедуры верификации не ограничатся булевскими преобразованиями формул языка L.

В настоящей статье рассмотрены методы верификации неинициальных спецификаций, хотя возможна ситуация, когда неинициальная спецификация не обладает требуемым свойством, которым обладает инициальная система, получаемая на заключительном этапе проектирования алгоритма. Поэтому представляют интерес методы верификации инициальной системы, определяемой неинициальной спецификацией и начальным условием, но рассмотрение этой проблемы выходит за рамки настоящей статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Чеботарев А.Н. Об одном подходе к функциональной спецификации автоматных систем // Кибернетика и системный анализ. — 1993. — № 3. — С. 31–42.
2. Piterman N., Pnueli A., Sa'ar Y. Synthesis of reactive(1) designs // Proc. Conf. on Verification, Model Checking and Abstract Interpretation. — 2006. — P. 364–380.
3. Lichtenstein O., Pnueli A. Checking that finite state concurrent programs satisfy their linear specification // Proc. 12th Symp. on Principles of Programming Languages (New York, ACM). — 1985. — P. 97–107.
4. Чеботарев А.Н. Проверка непротиворечивости простых спецификаций автоматных систем // Кибернетика и системный анализ. — 1994. — № 3. — С. 3–11.
5. Чеботарев А.Н. Проверка непротиворечивости формул языка L, представленных в дизъюнктивной нормальной форме. I // Там же. — 2005. — № 4. — С. 22–28.
6. Чеботарев А.Н. Метод раздельного резольвирования для проверки выполнимости формул языка L // Там же. — 1998. — № 6. — С. 13–20.
7. Кривой С.Л., Чеботарев А.Н. Усовершенствованный метод проверки выполнимости множества дизъюнктов в языке L // Таврич. вестн. информатики и математики. — 2006. — № 1. — С. 7–13.
8. Pnueli A., Rosner R. On the synthesis of a reactive module // Proc. 16th Ann. Symp. on Principles of Programming Languages (New York, ACM). — 1989. — P. 179–190.

Поступила 12.01.2009