УДК 519.713.1

А.Н. ЧЕБОТАРЕВ

ПРЕОБРАЗОВАНИЕ СПЕЦИФИКАЦИИ АВТОМАТА В ЯЗЫКЕ L^* В АВТОМАТНО ЭКВИВАЛЕНТНУЮ СПЕЦИФИКАЦИЮ В ЯЗЫКЕ L

Ключевые слова: *язык спецификации* L^*, *∃-формула, двустороннее сверхслово, Σ-автомат, элиминация кванторов, автоматная эквивалентность спецификаций.*

ВВЕДЕНИЕ

В основе используемого подхода к доказательному проектированию реактивных алгоритмов лежит спецификация функциональных требований к алгоритму в языке логики первого порядка с одноместными предикатами и формальный переход от спецификации к процедурному представлению алгоритма. Одна из основных проблем, связанных со спецификацией свойств алгоритма, состоит в разработке подходящего языка спецификации. При этом приходится искать компромисс между выразительными возможностями языка и сложностью алгоритмов проектирования. Для разрешения этого противоречия используются два уровня языка: язык исходной спецификации L^* [1], с достаточными для практических нужд выразительными возможностями и обеспечивающий удобство записи функциональных требований к алгоритму, и язык L [2], обладающий сравнительно ограниченными выразительными возможностями, но эффективно обрабатываемый процедурами синтеза. Возможности спецификации в языке L ограничены автоматами с конечной памятью [3], что позволило разработать для спецификаций в этом языке эффективные алгоритмы синтеза, проверки непротиворечивости и верификации темпоральных свойств, а также методы проектирования открытых систем, к которым относятся реактивные алгоритмы. Поэтому язык L^* используется (когда это необходимо) для начальной спецификации, которая затем преобразуется в спецификацию в языке L. В [4] рассмотрен способ перехода к спецификации в языке L, однако автомат, синтезированный по полученной таким образом спецификации, может иметь дополнительные, так называемые фиктивные состояния. Для устранения фиктивных состояний используется процедура проверки некоторых состояний автомата на фиктивность [5]. Такая процедура довольно сложна и, в отличие от всех остальных процедур проектирования алгоритма, использует процедурное представление автомата в виде множества состояний и функций переходов и выходов, а не формулы языка спецификации.

В настоящей статье предлагается преобразование исходной спецификации в спецификацию в языке L, специфицирующую автомат, в некотором смысле эквивалентный автомату, специфицируемому исходной спецификацией в языке L^*. Поскольку языки L^* и L обладают различными выразительными возможностями, такой переход осуществляется за счет введения дополнительных предикатных переменных.

ОСНОВНЫЕ ПОНЯТИЯ

Языки L^* и L построены на основе соответствующих фрагментов логики предикатов первого порядка с одноместными предикатами, определенными на множестве моментов дискретного времени, в качестве которого выступает множество $\mathbf{Z}$ целых чисел. Спецификация в обоих языках имеет вид формулы $\forall t F(t)$, где $F(t)$ — формула с одной свободной переменной t. В языке L формула $F(t)$ строится с помощью логических связок из атомов вида $p(t+k)$, где p — одноместный предикатный символ, t — переменная, принимающая значения из множества $\mathbf{Z}$, а k — целочисленная константа (сдвиг во времени). Язык L^* отличается от языка L тем, что при построении формулы $F(t)$ наряду с атомарными формулами используются еще формулы вида $\exists t_i (t_i \le t + k_1) \& F_1(t_i) \& \forall t_j ((t_i + k_2 \le \le t_j \le t + k_3) \to F_2(t_j))$, где $k_1, k_2, k_3 \in \mathbf{Z}$, $F_2(t_j)$ — формула языка L, а $F_1(t_i)$ — формула языка L^*. Такие формулы будем называть $\exists$-формулами. Для формулы $F(t)$ языка L^* определим понятие ранга (обозначается $\#(F(t))$) следующим образом:

1) $\#(p(t+k)) = k$ (ранг атома);

2) если $F(t) = \exists t_1 (t_1 \le t + k_1) \& F_1(t_1) \& \forall t_2 ((t_1 + k_2 \le t_2 \le t + k_3) \to F_2(t_2))$, то $\#(F(t)) = \max(r_1 + k_1, r_2 + k_3)$, где $r_1 = \#(F_1(t_1))$, $r_2 = \#(F_2(t_2))$;

3) если $F(t)$ построена из $F_i(t)$, $i = 1, \ldots, m$, с помощью логических связок, то $\#(F(t)) = \max(\#(F_1(t)), \ldots, \#(F_m(t)))$.

При определении автоматной семантики языков спецификации эти языки и автоматы рассматриваются как формализмы для задания множеств сверхслов (бесконечных слов) в алфавите $\Sigma(\Omega)$, где $\Omega = \{p_1, \ldots, p_m\}$ — набор предикатных символов спецификации. Алфавит $\Sigma(\Omega)$ представляет собой множество всех двоичных векторов длины m. Символы алфавита $\Sigma(\Omega)$ иногда удобно рассматривать как отображения вида $\sigma\colon \Omega \to \{0,1\}$. Пусть $\Omega_1 \subseteq \Omega$, проекцией символа $\sigma \in \Sigma(\Omega)$ на Ω_1 назовем ограничение отображения σ на Ω_1.

Поскольку в качестве связующего звена между формулами и специфицируемыми ими автоматами выступают множества сверхслов, определим необходимые понятия, касающиеся сверхслов и автоматов.

Пусть Σ — конечный алфавит, $\mathbf{Z}$ — множество целых чисел, $\mathbf{N}^+ = \{z \in \mathbf{Z} | z > 0\}$, $N^- = \{z \in \mathbf{Z} | z \le 0\}$. Отображения $u\colon \mathbf{Z} \to \Sigma$, $l\colon \mathbf{N}^+ \to \Sigma$ и $g\colon \mathbf{N}^- \to \Sigma$ называются соответственно двусторонним сверхсловом (обозначается $\ldots u(-2)u(-1)u(0)u(1)u(2)\ldots$), сверхсловом (обозначается $l(1)l(2)\ldots$) и обратным сверхсловом (обозначается $\ldots g(-2)g(-1)g(0))$ в алфавите Σ. Отрезок $u(\tau)u(\tau+1)\ldots u(\tau+k)$ двустороннего сверхслова u обозначается $u(\tau, \tau+k)$. Бесконечные отрезки $u(-\infty, k)$ и $u(k+1, \infty)$ назовем соответственно k-префиксом и k-суффиксом двустороннего сверхслова u. Если значение k не существенно, то будем говорить об ω-префиксе и ω-суффиксе.

Автоматная семантика языков описывается в терминах Σ-автоматов [6]. Конечный неинициальный Σ-автомат A — это тройка $< \Sigma, Q, \delta_A >$, где Σ, Q — конечные множества соответственно входных символов и состояний, а $\delta_A : Q \times \Sigma \to Q$ — частичная функция переходов автомата.

Сверхслово $l = \sigma_1 \sigma_2 \ldots$ в алфавите Σ допустимо в состоянии q Σ-автомата A, если существует такое сверхслово состояний $q_0 q_1 q_2 \ldots$, где $q_0 = q$, что для любого $i = 0, 1, 2, \ldots$ $q_{i+1} = \delta_A(q_i, \sigma_{i+1})$. Сверхслово l допустимо для Σ-автомата A, если оно допустимо хотя бы в одном из его состояний.

Каждой замкнутой формуле F ставится в соответствие множество 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). Интерпретация формулы F — это упорядоченный набор определенных на $\mathbf{Z}$ одноместных предикатов $\pi_1, \ldots, \pi_m$, соответствующих предикатным символам из Ω. Интерпретацию $I = < \pi_1, \ldots, \pi_m >$ можно представить в виде двустороннего сверхслова в алфавите $\Sigma(\Omega)$, а множество всех моделей для F — в виде множества $M(F)$ двусторонних сверхслов в этом алфавите. В дальнейшем интерпретации рассматриваются как двусторонние сверхслова, поэтому будем говорить об истинностном значении формулы F на двустороннем сверхслове u и значении формулы $F(t)$ в некоторой его позиции τ. Разность между максимальным и минимальным значениями рангов атомов, встречающихся в формуле языка L, называется глубиной формулы. Смысл понятия глубины формулы состоит в том, что истинностное значение формулы $F(t)$ глубины r в позиции τ интерпретации u определяется отрезком $u(\tau - r, \tau)$ соответствующего двустороннего сверхслова u.

Будем полагать, что формула $F=\forall t F(t)$ задает множество сверхслов $W(F)$, совпадающее с множеством всех ω-суффиксов двусторонних сверхслов из $M(F)$. Замкнутой формуле F языка L ставится в соответствие Σ-автомат A, для которого множество допустимых сверхслов совпадает с множеством $W(F)$ сверхслов, задаваемых формулой F. Класс Σ-автоматов, специфицируемых формулами языка L, совпадает с классом детерминированных циклических Σ-автоматов с конечной памятью [7]. Формулы $F_1=\forall t F_1(t)$ и $F_2=\forall t F_2(t)$ одного и того же или различных языков, специфицирующие Σ-автоматы с конечной памятью, назовем автоматно эквивалентными, если $W(F_1)=W(F_2)$.

Для описания преобразования спецификации из языка L^* в язык L требуется следующая равносильность [4].

Пусть $F(t)=\exists t_1(t_1 \le t+k_1)\& F_1(t_1)\& \forall t_2((t_1+k_2 \le t_2 \le t+k_3)\to F_2(t_2))$, где $k_1,k_2,k_3 \in \mathbf{Z}$, тогда

$$\begin{aligned}
&F(t)\Leftrightarrow F(t-1)\& F_2(t+k_3)\vee\\
&\vee(F_1(t+k_1)\vee\ldots\vee F_1(t-k_2+k_3+1),\ \text{если } k_3<k_1+k_2,\\
&\vee F_1(t+k_1)\& F_2(t+k_3)\&\ldots\& F_2(t+k_1+k_2),\ \text{если } k_3\ge k_1+k_2).
\end{aligned} \tag{1}$$

Здесь $F(t+k)$ обозначает формулу, полученную из $F(t)$ путем замены t на $t+k$.

Правую часть равносильности (1) назовем 1-разверткой $\exists$-формулы $F(t)$. Ее k-развертка получается в результате итеративного применения равносильности (1) k раз. Пусть 1-развертка $\exists$-формулы $F(t)$ имеет вид $F(t-1)\& h(t)\vee g(t)$, тогда ее k-развертка равна $F(t-k)\& h(t-k+1)\&\ldots\& h(t)\vee \alpha_{k-1}\vee\alpha_{k-2}\vee\ldots\vee\alpha_0$, где $\alpha_0=g(t)$, а α_i $(i=1,\ldots,k-1)$ равно $g(t-i)\& h(t-i+1)\&\ldots\& h(t)$.

ПЕРЕХОД ОТ ЯЗЫКА L* К ЯЗЫКУ L

Пусть замкнутые формулы F_1 и F_2 имеют соответственно сигнатуры Ω_1 и Ω_2, а Ω — непустое подмножество множества $\Omega_1\cap\Omega_2$. Формулы F_1 и F_2 назовем эквивалентными относительно сигнатуры Ω, если множества проекций всех моделей (двусторонних сверхслов) из $M(F_1)$ и $M(F_2)$ на Ω совпадают. Аналогично, формулы F_1 и F_2 автоматно эквивалентны относительно сигнатуры Ω, если множество проекций на Ω всех сверхслов из $W(F_1)$ совпадает с множеством проекций на Ω всех сверхслов из $W(F_2)$.

Основная идея состоит в том, чтобы исходную спецификацию S в языке L^* преобразовать в такую, эквивалентную относительно ее сигнатуры, спецификацию S_1 в языке L^*, которая специфицирует Σ-автомат с конечной памятью. В этом случае существует спецификация в языке L, автоматно эквивалентная S_1, т.е. такая спецификация F, что $W(S_1)=W(F)$. Преобразование S в S_1 осуществляется путем введения дополнительных предикатных символов для предикатов, удовлетворяющих $\exists$-подформулам спецификации S.

В [4] описан переход от языка L^* к языку L, называемый элиминацией кванторов, однако получаемая при этом спецификация может не быть автоматно эквивалентна исходной спецификации S относительно ее сигнатуры. Показано, что автомат, специфицируемый спецификацией S, эквивалентен в некотором смысле подавтомату автомата, специфицируемого построенной спецификацией в языке L. Отсюда следует необходимость выполнения довольно сложной процедуры проверки некоторых состояний синтезированного автомата на фиктивность.

В предлагаемом подходе к построению спецификации в языке L также используется процедура элиминации кванторов, однако полученная спецификация преобразуется в спецификацию, автоматно эквивалентную исходной спецификации относительно ее сигнатуры.

Процедура элиминации кванторов в формуле $F(t)$ состоит из двух этапов. На первом этапе формула $F(t)$ преобразуется путем введения дополнительных предикатных символов. Пусть $\varphi_i(t)$ — максимальная $\exists$-подформула формулы $F(t)$, т.е.

подформула, не содержащаяся ни в какой другой $\exists$-подформуле. Каждая такая $\exists$-подформула заменяется атомом вида $z_i(t+r_i)$, где r_i — ранг заменяемой $\exists$-подформулы, а z_i — предикатный символ, отсутствующий в формуле $F(t)$, и в спецификацию добавляется эквивалентность $z_i(t) \leftrightarrow \varphi_i(t-r_i)$. Если $\exists$-формулы в правых частях эквивалентностей содержат отличные от них $\exists$-подформулы, то с ними поступают так же, как и с $\exists$-подформулами исходной формулы. Они заменяются соответствующими обозначениями новых предикатов, и добавляются эквивалентности вида $z_j(t) \leftrightarrow \varphi_j(t)$. Это осуществляется до тех пор, пока ни одна из $\exists$-формул $\varphi_j(t)$ не будет содержать вхождений $\exists$-подформул, отличных от нее. В результате получим спецификацию $S_1 = \forall t F_z(t)$. Как показано в [8], эта спецификация эквивалентна исходной спецификации относительно ее сигнатуры и специфицирует автомат с конечной памятью. На втором этапе в каждой эквивалентности $z_i(t) \leftrightarrow \varphi_i(t-r_i)$ $\exists$-формула $\varphi_i(t-r_i)$ заменяется правой частью равносильности (1), где обозначение формулы $\varphi_i(t-r_i-1)$ заменяется на $z_i(t-1)$. Это дает спецификацию $S_2 = \forall t\, f_z(t)$ в языке L. Второй этап приводит к неэквивалентности преобразования, выражающейся в том, что полученная формула S_2 языка L имеет дополнительные модели по сравнению с формулой $\forall t F_z(t)$, а синтезированный по ней автомат может иметь дополнительные (фиктивные) состояния.

Рассмотрим пример элиминации кванторов.

Пример 1. Пусть $F(t) = \exists t_1 (t_1 \le t-1) \neg x(t_1) y(t_1) \forall t_2 ((t_1 + 2 \le t_2 \le t-1) \to$ $\to \exists t_3 (t_3 \le t_2 - 1) \neg y(t_3)) \vee \neg y(t)$. В этой формуле имеется единственная максимальная $\exists$-подформула, ранг которой равен – 1. Заменив ее атомом $z_1(t-1)$ и добавив соответствующую эквивалентность, получим $(z_1(t-1) \vee \neg y(t)) \& (z_1(t) \leftrightarrow (\exists t_1 (t_1 \le t) \&$ $\& \neg x(t_1) y(t_1) \& \forall t_2 ((t_1 + 2 \le t_2 \le t) \to \exists t_3 (t_3 \le t_2 - 1) \neg y(t_3))))$. В правой части добавленной эквивалентности имеется $\exists$-подформула ранга – 1, которую заменяем атомом $z_2(t-1)$. Добавляемая эквивалентность для $z_2(t)$ имеет вид $z_2(t) \leftrightarrow$ $\leftrightarrow (\exists t_3 (t_3 \le t_2) \neg y(t_3))$. Таким образом, $F_z(t) = (z_1(t-1) \vee \neg y(t)) \& (z_1(t) \leftrightarrow (\exists t_1$ $(t_1 \le t) \neg x(t_1) y(t_1) \forall t_2 ((t_1 + 2 \le t_2 \le t) \to z_2(t-1))) \& (z_2(t) \leftrightarrow \exists t_3 (t_3 \le t_2) \neg y(t_3))$. Заменив правые части эквивалентностей соответствующими 1-развертками, в которых обозначения $\exists$-подформул заменены соответственно атомами $z_1(t-1)$ и $z_2(t-1)$, получим

$$S_2 = \forall t (z_1(t-1) \vee \neg y(t)) \& (z_1(t) \leftrightarrow (z_1(t-1) z_2(t-1) \vee$$
$$\vee \neg x(t-1) y(t-1) \vee \neg x(t) y(t))) \& (z_2(t) \leftrightarrow (z_2(t-1) \vee \neg y(t))).$$

Конец примера.

Рассмотрим формулы $F_1 = \forall t (z(t) \leftrightarrow \exists t_1 (t_1 \le t) g(t_1) \forall t_2 ((t_1 < t_2 \le t) \to h(t_2)))$ и $F_2 = \forall t (z(t) \leftrightarrow (z(t-1) h(t) \vee g(t)))$, где $g(t)$ и $h(t)$ — формулы языка L, и охарактеризуем классы моделей для этих формул. Несложно убедиться в справедливости следующих утверждений.

Утверждение 1. Все модели как для F_2, так и для F_1 удовлетворяют формуле $\forall t (z(t) \to (g(t) \vee h(t)))$, т.е. являются моделями для этой формулы.

Утверждение 2. Всякая модель для F_2, удовлетворяющая формуле $\forall t \exists t_1 (t_1 \le t) g(t_1)$, есть модель для F_1.

Утверждение 3. Всякая модель для F_2, удовлетворяющая формуле $\forall t \exists t_1 (t_1 \le t) \neg z(t_1)$, есть модель для F_1.

Покажем, например, справедливость утверждения 2.

Пусть u — модель для F_2, удовлетворяющая формуле $\forall t \exists t_1 (t_1 \le t) g(t_1)$, и τ — произвольная позиция в u. Покажем, что из истинности формулы $z(t) \leftrightarrow (z(t-1) h(t) \vee g(t))$ в позиции τ следует истинность формулы $z(\tau) \leftrightarrow \varphi_1(\tau)$, где $\varphi_1(\tau) = \exists t_1 (t_1 \le \tau) g(t_1) \forall t_2 ((t_1 < t_2 \le \tau) \to h(t_2))$. Поскольку u удовлетворяет формуле $\forall t \exists t_1 (t_1 \le t) g(t_1)$, существует $\tau_1 \le \tau$ такое, что $g(\tau_1)$ истинна на u. Пусть τ_1 — ближайшее к τ значение, для которого $g(\tau_1)$ истинна на u, т.е. для всех $\tau_1 < t \le \tau$ $g(t)$ ложна. Рассмотрим два случая.

1. Если $\tau_1 = \tau$, то $z(\tau)$ истинна на u. Очевидно, что $\varphi_1(t)$ также истинна в позиции τ, а значит, $z(\tau) \leftrightarrow \varphi_1(\tau)$ истинна на u.

2. Пусть $\tau_1 < \tau$. Рассмотрим сначала случай, когда $z(\tau)$ истинна на u. Тогда в силу истинности $z(\tau-1)h(\tau) \vee g(\tau)$ истинны $z(\tau-1)$ и $h(\tau)$. Следовательно, для всех $\tau_1 < t \le \tau$ истинна $h(t)$. В этом случае $\varphi_1(\tau)$ также истинна, а значит, $z(\tau) \leftrightarrow \varphi_1(\tau)$ истинна на u. Если $z(\tau)$ ложна, то либо существует $\tau_1 < t \le \tau$ такое, что $h(t)$ ложна, в силу чего $\varphi_1(\tau)$ ложна и $z(\tau) \leftrightarrow \varphi_1(\tau)$ истинна на u, либо для всех $\tau_1 \le t \le \tau$ $z(t)$ ложна, что противоречит истинности $g(\tau_1)$. Таким образом, во всех случаях из истинности $z(\tau) \leftrightarrow (z(\tau-1)h(\tau) \vee g(\tau))$ следует истинность формулы $z(\tau) \leftrightarrow \varphi_1(\tau)$, из чего вытекает, что u — модель для F_1. Конец доказательства.

Если интерпретация не обладает свойством из утверждения 2, т.е. не удовлетворяет формуле $\forall t \exists t_1 (t_1 \le t) g(t_1)$, то она имеет ω-префикс, во всех позициях которого истинна $\neg g(t)$. Если интерпретация не обладает свойством из утверждения 3, то она имеет ω-префикс, во всех позициях которого истинна $z(t)$. Отсюда следует, что если интерпретация не обладает ни одним из этих свойств, то она имеет ω-префикс, во всех позициях которого истинна $\neg g(t) z(t)$, т.е. она удовлетворяет формуле $\exists t \forall t_1 ((t_1 \le t) \to \neg g(t_1) z(t_1))$. Если существует модель для F_2, удовлетворяющая этой формуле, то в силу утверждения 1 она удовлетворяет формуле

$$\exists t \, \forall t_1 ((t_1 \le t) \to \neg g(t_1) h(t_1) z(t_1)). \tag{2}$$

Очевидно, что не существует моделей для F_1, удовлетворяющих этой формуле, т.е. всякая модель для F_2, имеющая ω-префикс, в каждой позиции которого истинна формула $\neg g(t) h(t) z(t)$, не является моделью для F_1. В то же время для F_2 существуют такие модели, например модель, удовлетворяющая формуле $\forall t (\neg g(t) h(t) z(t))$. Таким образом, справедливо следующее утверждение.

Утверждение 4. Все модели для F_2, удовлетворяющие формуле (2), и только они, не являются моделями для F_1.

Отсюда следует также, что все модели для F_1 содержатся среди моделей для F_2.

Выше рассматривалась $\exists$-формула со значениями $k_1, k_3 = 0$ и $k_2 = 1$. Утверждение, аналогичное утверждению 4, справедливо и для $\exists$-формул с любыми значениями k_i $(i = 1, 2, 3)$. Это следует из того факта, что для каждой формулы вида $F(t) = \exists t_1 (t_1 \le t + k_1) \& F_1(t_1) \& \forall t_2 ((t_1 + k_2 \le t_2 \le t + k_3) \to F_2(t_2))$, где k_1, k_2, $k_3 \in \mathbf{Z}$, существует эквивалентная ей формула вида $\exists t_1 (t_1 \le t) \& f_1(t_1) \& \forall t_2 ((t_1 < t_2 \le t) \to f_2(t_2))$. Пусть 1-развертка формулы $F(t)$ имеет вид $F(t-1) \& h(t) \vee g(t)$. Покажем, что формула $f(t) = \exists t_1 (t_1 \le t) \& g(t_1) \& \forall t_2 ((t_1 < t_2 \le t) \to h(t_2))$ эквивалентна формуле $F(t)$, т.е. что из истинности $F(t)$ в позиции τ произвольной интерпретации u следует истинность $f(\tau)$ в этой интерпретации и наоборот. Формула $F(t)$ истинна в позиции τ интерпретации u тогда и только тогда, когда существует такое $k \ge 0$, что $F_1(\tau + k_1 - k)$ истинна в интерпретации u и для всех t, удовлетворяющих $\tau + k_1 - k + k_2 \le t \le \tau + k_3$, истинна формула $F_2(t)$. Используя понятие k-развертки, это утверждение можно переформулировать следующим образом.

Утверждение 5. $\exists$-формула $F(t)$ истинна в позиции τ двустороннего сверхслова u тогда и только тогда, когда существует такое $k \ge 1$, что формула α_k в $(k+1)$-развертке формулы $F(t)$ истинна в позиции τ этого сверхслова.

Поскольку развертки формул $F(t)$ и $f(t)$ совпадают с точностью до обозначения формул, то и члены α_k в их $(k+1)$-развертках также совпадают. Таким образом, для любой интерпретации из истинности одной из этих формул в позиции τ следует истинность другой и наоборот, что свидетельствует об их эквивалентности.

Далее рассмотрим, как для спецификации S_1 в языке L^* построить такую спецификацию F в языке L, чтобы $W(S_1) = W(F)$.

АВТОМАТНАЯ ЭКВИВАЛЕНТНОСТЬ СПЕЦИФИКАЦИЙ

Пусть S_1 и S_2 — спецификации соответственно в языке L^* и L такие, что $M(S_1) \subseteq M(S_2)$, и G — множество всех моделей для S_2, не являющихся моделями для S_1. Все ω-суффиксы каждой модели для спецификации содержатся в множестве сверхслов, задаваемых этой спецификацией, поэтому спецификация S_2 автоматно эквивалентна спецификации S_1 тогда и только тогда, когда множество всех ω-суффиксов моделей из G содержится в $W(S_1)$.

Для описания способа построения спецификации F, автоматно эквивалентной спецификации S_1, удобно воспользоваться понятием пространства состояний, ассоциируемого с формулой вида $\forall t F(t)$ [7]. Пусть $F = \forall t F(t)$ — формула языка L глубины r, с сигнатурой $\Omega = \{p_1, \ldots, p_m\}$. Формулу $F(t)$ будем рассматривать как пропозициональную формулу с пропозициональными переменными $p_1(t), \ldots, p_m(t)$, $p_1(t-1), \ldots, p_m(t-1), \ldots, p_1(t-r), \ldots, p_m(t-r)$. Если $F_1(t)$ и $F_2(t)$ — логически эквивалентные формулы, очевидно, что формулы $\forall t F_1(t)$ и $\forall t F_2(t)$ задают одно и то же множество сверхслов. Последовательность $\sigma_0, \sigma_1, \ldots, \sigma_r$ символов алфавита $\Sigma(\Omega)$ 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 состоянием $q = \sigma_0, \sigma_1, \ldots, \sigma_r$ непосредственно следуют 2^m состояний вида $\sigma_1, \ldots, \sigma_r, \sigma$, где $\sigma \in \Sigma(\Omega)$. Отношение, обратное N, обозначим P и назовем отношением непосредственного предшествования. Очевидно, что состоянию $\sigma_0, \sigma_1, \ldots, \sigma_r$ непосредственно предшествуют 2^m состояний вида $\sigma, \sigma_0, \sigma_1, \ldots, \sigma_{r-1}$, где $\sigma \in \Sigma(\Omega)$. 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 σ_i в состоянии $q = \sigma_0, \sigma_1, \ldots, \sigma_r$ рассматривать как истинностные значения соответствующих атомов ранга $i - r$, то можно говорить о значении формулы $F(t)$ на состоянии q.

Пусть $F(t)$ имеет вид $(w_1(t-1) \& \neg u(t) \vee \neg w_2(t-1) \& u(t)) \to w_1(t)$ и $\Omega = < u, w_1, w_2 >$. Для этой формулы $r = 1$. Вычислим значение $F(t)$ на состоянии $\sigma_0, \sigma_1 = < 101 >, < 011 >$. При этом атомы ранга 0 принимают значения из σ_1, а атомы ранга – 1 — из σ_0. Таким образом, $w_1(t) = 1, u(t) = 0$, $w_1(t-1) = 0$ и $w_2(t-1) = 1$, следовательно, $F(t)$ на состоянии σ_0, σ_1 принимает значение 1.

Формулу $F(t)$ будем рассматривать как представление множества $Q(F(t))$ состояний из $Q(r, \Omega)$, а именно, тех состояний, на которых она истинна.

Итак, состояние из $Q(r, \Omega)$ представляет собой отрезок длины $r+1$ двустороннего сверхслова. Таким образом, любой интерпретации для формулы F соответствует двустороннее сверхслово состояний $\ldots q_{-2} q_{-1} q_0 q_1 q_2 q_3 \ldots$, такое, что $q_{i+1} \in N(q_i)$ для любого $i \in \mathbf{Z}$. Пусть u — интерпретация для формулы $F = \forall t F(t)$, а $Q(u)$ — множество всех состояний, встречающихся в двустороннем сверхслове состояний, соответствующем u.

Утверждение 6. Интерпретация u является моделью для формулы F тогда и только тогда, когда $Q(u) \subseteq Q(F(t))$.

Пусть $S_1 = \forall t F_z(t)$ и $S_2 = \forall t f_z(t)$ — спецификации, полученные соответственно на первом и втором этапах элиминации кванторов. Способ построения спецификации, автоматно эквивалентной S_1, рассмотрим сначала для случая, когда $F_z(t)$ содержит только одну $\exists$-подформулу. При этом $M(S_1)$ не содержит только те модели из $M(S_2)$, которые удовлетворяют формуле вида

$$\exists \tau \, \forall t_1 ((t_1 \le \tau) \to \varphi(t_1)). \qquad (3)$$

Заметим, что если модель обладает свойством (3), то все состояния, встречающиеся в обратном сверхслове состояний, соответствующем ее τ-префиксу (τ удовлетворяет формуле (3)), принадлежат множеству $Q(\varphi(t))$.

Рассмотрим следующее преобразование формулы $f_z(t)$.

Пусть D — множество состояний из $Q(r, \Omega)$, задаваемое формулой $f_z(t)$. Формула $f_z(t) \& \neg\varphi(t)$ задает множество $D_0 = Q(f_z(t) \& \neg\varphi(t))$ всех состояний из D, не принадлежащих $Q(\varphi(t))$. Построим множество состояний всех таких двусторонних сверхслов состояний, соответствующих моделям для S_2, которые имеют ω-префикс с бесконечным количеством вхождений состояний из D_0.

Для этого построим максимальное подмножество D^* состояний из D_0, достижимых (в смысле транзитивного замыкания отношения N) из состояний этого подмножества. Таким образом, если D^* не пусто, каждое его состояние достижимо из некоторого состояния из D^*. Для произвольного множества $D_1 \subseteq D$ обозначим $N^+(D_1)$ множество всех состояний из D, достижимых из D_1. Построим множество $N^+(D^*)$, а формулу, задающую это множество состояний, обозначим $F(t)$. Для множества моделей из $M(\forall t F(t))$ справедливы следующие утверждения.

Утверждение 7. Все модели из $M(S_1)$ содержатся среди $M(\forall t F(t))$.

Действительно, $M(S_1)$ — это все модели из $M(\forall t\, f_z(t))$, не обладающие свойством (3), т.е. не имеющие ω-префикса, для которого соответствующее обратное сверхслово состояний содержит состояния только из $Q(\varphi(t))$. Таким образом, всякое обратное сверхслово состояний, соответствующее ω-префиксу модели из $M(S_1)$, имеет бесконечное количество позиций с состояниями из D_0 и, следовательно, все состояния двустороннего сверхслова состояний, соответствующего любой модели из $M(S_1)$, достижимы из содержащихся среди них состояний из D_0. Из этого следует, что все состояния такого сверхслова содержатся во множестве $N^+(D^*)$, а в силу утверждения 6 модели из $M(S_1)$ содержатся среди $M(\forall t F(t))$.

Утверждение 8. Для любого ω-суффикса модели из $M(\forall t F(t))$, не принадлежащей $M(S_1)$, существует модель из $M(S_1)$ с таким же ω-суффиксом.

Поскольку каждое состояние из D^* достижимо из некоторого состояния из D^*, в силу конечности этого множества оно содержит одно или несколько состояний, достижимых из себя, и все состояния из $N^+(D^*)$ достижимы из этих состояний. Пусть модель $u \in M(\forall t F(t))$ не принадлежит $M(S_1)$. Рассмотрим произвольный ее ω-суффикс l, которому соответствует сверхслово состояний $q_1 q_2 q_3 \ldots$ Поскольку q_1 принадлежит $N^+(D^*)$, существует состояние из D^*, достижимое из себя, из которого достижимо q_1. Из этого следует, что существует модель для S_1 с ω-суффиксом l и ω-префиксом, для которого соответствующее ему обратное сверхслово состояний содержит бесконечно много позиций с состояниями из D^*.

Таким образом, формула $F = \forall t F(t)$ автоматно эквивалентна формуле S_1.

Рассмотрим теперь случай, когда спецификация S_1 содержит две $\exists$-подформулы. Предлагаемый способ построения спецификации, автоматно эквивалентной S_1, легко обобщить на формулу S_1, содержащую $k > 2$ $\exists$-подформул.

Множество $M(S_1)$ не содержит только те модели из $M(S_2)$, которые удовлетворяют по крайней мере одной из формул — $\exists \tau \forall t_1((t_1 \le \tau) \to \varphi_1(t_1))$ или $\exists \tau \forall t_1((t_1 \le \tau) \to \varphi_2(t_1))$, где $\varphi_1(t)$ и $\varphi_2(t)$ определяются видом соответствующих $\exists$-подформул в формуле S_1. Пусть D_1 — множество состояний, задаваемое формулой $f_z(t) \& \neg\varphi_1(t)$, а D_2 — формулой $f_z(t) \& \neg\varphi_2(t)$. Построим множество состояний всех таких двусторонних сверхслов состояний, соответствующих моделям для S_2, которые имеют ω-префикс с бесконечным количеством вхождений как состояний из D_1, так и состояний из D_2. Для этого определим все такие пары состояний (q_i, q_j), где $q_i \in D_1$, $q_j \in D_2$, что q_j достижимо из q_i, а q_i достижимо из q_j. Затем, если это множество не пусто, построим множество всех состояний из D, которые достижимы из состояний этих пар. Формула $F(t)$, задающая полученное множество, представляет собой результат требуемого преобразования формулы $f_z(t)$.

Сначала покажем, что все модели из $M(S_1)$ содержатся среди $M(\forall t F(t))$.

Действительно, $M(S_1)$ — это все модели из $M(\forall t\, f_z(t))$, не имеющие ω-префикса, для которого соответствующее обратное сверхслово состояний содержит состояния только из $Q(\varphi_1(t))$ или только из $Q(\varphi_2(t))$. Таким образом, всякое обратное сверхслово состояний, соответствующее ω-префиксу модели из $M(S_1)$, имеет бесконечное количество позиций как с состояниями из D_1, так и с состояниями из D_2. Поэтому каждое состояние из D_1, содержащееся в таком обратном сверхслове состояний, достижимо из имеющегося в нем состояния из D_2 и наоборот. Следовательно, все состояния двустороннего сверхслова состояний, соответствующего любой модели из $M(S_1)$, достижимы из имеющихся среди них состояний из D_1 и D_2. Из этого вытекает, что все состояния такого сверхслова принадлежат множеству $Q(F(t))$, а в силу утверждения 6 модели из $M(S_1)$ содержатся среди $M(\forall t F(t))$. Аналогично тому, как это сделано выше, можно показать, что для любого ω-суффикса модели из $M(\forall t F(t))$, не принадлежащей $M(S_1)$, существует модель из $M(S_1)$ с таким же ω-суффиксом.

ПРЕОБРАЗОВАНИЕ СПЕЦИФИКАЦИИ

В основе описанных выше процедур преобразования формулы $f_z(t)$ лежат операции построения множества состояний из $Q = Q(F(t))$, непосредственно следующих за заданным множеством состояний $Q_1 \subseteq Q$, и операция построения множества всех состояний из Q, достижимых из заданного множества состояний. Рассмотрим, как эти операции выполняются на уровне преобразований формул, задающих соответствующие множества состояний. Множество состояний из $Q = Q(F(t))$, непосредственно следующих за состояниями из $Q_1 = Q(F_1(t))$, равно $N(Q_1) \cap Q$. На уровне формул эта операция выглядит как $N(F_1(t)) \& F(t)$, где формула $N(F_1(t))$ задает множество всех тех состояний из $Q(r, \Omega)$, которые непосредственно следуют за состояниями из множества, задаваемого формулой $F_1(t)$. Для формулы $F(t)$ глубины r, заданной в д.н.ф., $N(F(t))$ получается, если в каждой элементарной конъюнкции формулы $F(t)$ удалить все литеры минимального ранга (т.е. ранга $-r$) и полученную д.н.ф. сдвинуть на 1 влево [9].

Построение формулы, задающей множество всех тех состояний из $Q(F(t))$, которые достижимы из его подмножества $Q_0 = Q(F_0(t))$, осуществляется следующим образом. Сначала строится формула $F_1(t) = N(F_0(t)) \& F(t)$, затем, начиная с $i = 1$, итеративно вычисляются $F_{i+1}(t) = N(F_i(t)) \& F(t) \vee F_i(t)$ до стабилизации формулы.

При наличии только одной $\exists$-подформулы в исходной спецификации преобразование формулы $f_z(t)$ сводится к построению максимального подмножества $D^* \subseteq D_0$, все состояния которого достижимы из D^*. При этом для $i = 0, 1, 2, \ldots$ строится последовательность множеств $D_{i+1} = N^+(D_i) \cap D_i$ до тех пор, пока для некоторого j не будет получено $D_{j+1} = D_j$. Если $D_j = \varnothing$, то $M(S_2)$ не содержит моделей из $M(S_1)$ и, следовательно, спецификация S_1 противоречива; если $D_j \neq \varnothing$, то $D_j = D^*$.

При наличии двух $\exists$-подформул в исходной спецификации преобразование формулы $f_z(t)$ сводится к построению пары максимальных подмножеств $Q_1 \subseteq D_1$ и $Q_2 \subseteq D_2$ таких, что все состояния из Q_1 достижимы из Q_2, а все состояния из Q_2 достижимы из Q_1. Нетрудно показать, что такие подмножества содержат по крайней мере одну пару состояний $q_1 \in Q_1$ и $q_2 \in Q_2$, которые достижимы одно из другого.

Пусть Q — множество состояний, задаваемое формулой $f_z(t)$, а $Q_{10} = D_1$ и $Q_{20} = D_2$. 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 $Q_{12} \subseteq Q_{11}$ и $Q_{22} \subseteq Q_{21}$, и т.д. В результате такого процесса получаются две последовательности: $Q_{10} \supseteq Q_{11} \supseteq Q_{12} \supseteq \ldots$ и $Q_{20} \supseteq Q_{21} \supseteq Q_{22} \supseteq \ldots$ Процесс заканчивается, когда хотя бы одно из множеств этих последовательностей станет пустым, что свидетельствует о противоречивости исходной спецификации, либо в каждой последовательности три последних множества будут равны между собой. Формула, задающая множество всех тех состояний из Q, которые достижимы из состояний последнего множества любой последовательности, представляет собой результат требуемого преобразования формулы $f_z(t)$. Очевидно, что приведенный алгоритм легко распространяется на случай наличия $k>2$ $\exists$-подформул в исходной спецификации. Обозначим $Q_{10},\ldots,Q_{k0}$ множества состояний, задаваемые соответственно формулами $f_z(t)\&\neg\varphi_1(t),\ldots$ $\ldots, f_z(t)\&\neg\varphi_k(t)$. Процесс вычисления состоит в построении для $j=1,\ldots,k$ последовательностей $Q_{j0}\supseteq Q_{j1}\supseteq Q_{j2}\supseteq\ldots$ На *i*-й итерации $(i=1,2,\ldots)$ вычисляются очередные k членов этих последовательностей в соответствии с формулой $Q_{ji}=N^+(Q_{s(i)i})\cap Q_{j(i-1)}$, где $s(i)=(i-1)\operatorname{mod}k+1$, $j=1,\ldots,k$. Окончание процесса определяется равенством последних $k+1$ членов каждой последовательности.

Пример 2. Исходная спецификация S имеет вид $\forall t(\exists t_1(t_1\le t)\neg x(t_1)\vee$ $\vee\, y(t))\&(x(t-1)\vee\neg x(t))$. Рассмотрим последовательность ее преобразований в автоматно эквивалентную относительно ее сигнатуры спецификацию в языке L.

Спецификация S_1 представляет собой формулу $\forall t S_z(t)=\forall t(z(t)\vee y(t))\&$ $\&(x(t-1)\vee\neg x(t))\&(z(t)\leftrightarrow\exists t_1(t_1\le t)\neg x(t_1))$. Формула $f_z(t)$ в соответствующей спецификации S_2 имеет вид $(z(t)\vee y(t))\&(x(t-1)\vee\neg x(t))\&(z(t)\leftrightarrow(z(t-1)\vee$ $\vee\neg x(t)))=z(t-1)x(t-1)z(t)\vee\neg z(t-1)x(t-1)x(t)y(t)\neg z(t)\vee\neg x(t)z(t)$. Формула $\varphi(t)=$ $=x(t)z(t)$ характеризует ω-префиксы моделей из $M(S_2)$, отсутствующих в $M(S_1)$. Формула $D_0(t)$, задающая множество состояний $D_0=Q(f_z(t)\&\neg\varphi(t))$, равна $f_z(t)\&\neg\varphi(t)=\neg z(t-1)x(t-1)x(t)y(t)\neg z(t)\vee\neg x(t)z(t)$. Построим теперь формулу $D^*(t)$, задающую множество состояний D^*. При этом одновременно будет построена формула $F(t)$, задающая $N^+(D^*)$.

Сначала строим формулу $D_1(t)$:

$$N(D_0(t))=\neg x(t-1)z(t-1)\vee x(t-1)y(t-1)\neg z(t-1);$$
$$F_1(t)=N(D_0(t))\& f_z(t)=\neg x(t-1)z(t-1)\neg x(t)z(t)\vee$$
$$\vee x(t-1)y(t-1)\neg z(t-1)\neg x(t)z(t)\vee x(t-1)y(t-1)\neg z(t-1)x(t)y(t)\neg z(t).$$

Поскольку $N(F_1(t))=N(D_0(t))$, имеем $F_2(t)=N(F_1(t))\& f_z(t)\vee F_1(t)=F_1(t)$ и, следовательно, $N^+(D_0(t))=F_1(t)$. Таким образом, $D_1(t)=N^+(D_0(t))\& D_0(t)=$ $=\neg x(t-1)z(t-1)\neg x(t)z(t)\vee x(t-1)y(t-1)\neg z(t-1)\neg x(t)z(t)\vee x(t-1)y(t-1)\neg z(t-1)\&$ $\& x(t)y(t)\neg z(t)$.

При построении $D_2(t)$ получим $N(D_1(t))=\neg x(t-1)z(t-1)\vee x(t-1)y(t-1)$ $\neg z(t-1)=N(D_0(t))$, поэтому $N^+(D_1(t))=N^+(D_0(t))=F_1(t)$. Таким образом, $D_2(t)=N^+(D_1(t))\& D_1(t)=D_1(t)$, следовательно, $D^*(t)=D_1(t)$, а значит, $F(t)=N^+(D_1(t))=F_1(t)$.

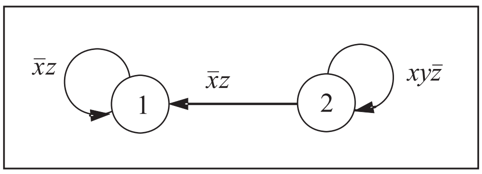

Рис. 1

Граф автомата, специфицируемого формулой $\forall tF(t)$, приведен на рис. 1.

Заметим, что автомат, синтезированный по формуле S_2, полученной в результате элиминации кванторов, имеет три состояния, из которых одно фиктивное.

ЗАКЛЮЧЕНИЕ

В используемом подходе к доказательному проектированию реактивных алгоритмов языком исходной спецификации является логический язык L* с достаточными для практических задач выразительными возможностями. Однако большинство методов проектирования, таких как проверка непротиворечивости спецификаций, их детерминизация, проверка реализуемости открытой системы, верификация и другие, разработаны для спецификаций в более простом языке L, что позволило получить приемлемую эффективность этих методов. Поэтому важное значение имеет преобразование спецификации из языка L* в язык L.

Учитывая, что в языке L могут быть специфицированы только автоматы с конечной памятью, такое преобразование предполагает переход от спецификации автомата, не обладающего конечной памятью, к спецификации соответствующего автомата с конечной памятью. В настоящей работе предложен метод преобразования спецификации в языке L^* в спецификацию в языке L, автоматно эквивалентную исходной спецификации относительно ее сигнатуры. Такое преобразование осуществляется в три этапа. На первом этапе спецификация в языке L^* за счет введения дополнительных предикатных символов преобразуется в эквивалентную относительно ее сигнатуры спецификацию в этом же языке, но специфицирующую автомат с конечной памятью. На втором этапе полученная спецификация автомата с конечной памятью преобразуется в спецификацию в языке L. Это простое синтаксическое преобразование дает спецификацию, автоматно не эквивалентную преобразуемой спецификации в языке L^*. На третьем этапе спецификация в языке L преобразуется в спецификацию в этом же языке, автоматно эквивалентную спецификации автомата с конечной памятью в языке L^*.

Таким путем решается несколько проблем: устраняется необходимость проверки состояний синтезированного автомата на фиктивность; проверка непротиворечивости спецификации в языке L^* сводится к проверке непротиворечивости спецификации в языке L, для которой разработаны эффективные методы; появляется возможность применения методов синтеза к спецификациям, не удовлетворяющим теореме о спецификации [4]. В связи с этим в статье рассмотрен расширенный вариант языка L^*, выходящий за рамки требований теоремы о спецификации, на которой были основаны все методы синтеза, использовавшиеся для языка L^*. Расширение языка L^* состоит в том, что в качестве подформул, входящих в $\exists$-формулы языка, могут быть не только $\exists$-формулы и формулы языка L, но и произвольные формулы языка L^*.

СПИСОК ЛИТЕРАТУРЫ

1. Чеботарев А.Н. Расширение логического языка спецификации и проблема синтеза // Кибернетика и системный анализ. — 1996. — № 6. — С. 11–27.
2. Чеботарев А.Н. Об одном подходе к функциональной спецификации автоматных систем // Там же. — 1993. — № 3. — С. 31–42.
3. Гилл А. Введение в теорию конечных автоматов. — М.: Наука, 1966. — 227 с.
4. Чеботарев А.Н. Синтез процедурного представления автомата, специфицированного в логическом языке L^*. I // Кибернетика и системный анализ. — 1997. — № 4. — С. 60–74.
5. Чеботарев А.Н. Синтез процедурного представления автомата, специфицированного в логическом языке L^*. II // Там же. — 1997. — № 6. — С. 115–127.
6. Чеботарев А.Н. Синтез алгоритма по его логической спецификации // Управляющие системы и машины. — 2004. — № 5. — С. 53–60.
7. Чеботарев А.Н. Синтез недетерминированного автомата по его логической спецификации. I // Кибернетика и системный анализ. — 1995. — № 5. — С. 3–15.
8. Чеботарев А.Н. О классе формул языка L^*, специфицирующих автоматы с конечной памятью // Там же. — 2010. — № 1. — С. 3–9.
9. Чеботарев А.Н., Куривчак О.И. Аппроксимация множеств сверхслов формулами языка L // Там же. — 2007. — № 6. — С. 18–26.

Поступила 22.04.2009