КИБЕРНЕТИКА

А.Н. ЧЕБОТАРЕВ

УДК 519.713.1

О КЛАССЕ ФОРМУЛ ЯЗЫКА L^*, СПЕЦИФИЦИРУЮЩИХ АВТОМАТЫ С КОНЕЧНОЙ ПАМЯТЬЮ

Ключевые слова: *язык спецификации* L^*, $\exists$-*формула, двустороннее сверхслово,* Σ-*автомат, автомат с конечной памятью, k-разделимость.*

ВВЕДЕНИЕ

Автоматы с конечной памятью [1, 2] играют важную роль в теории и практике проектирования дискретных систем. С одной стороны, эти автоматы обладают такими свойствами, которые позволяют упростить методы их анализа, преобразования и реализации, с другой — практически всякая реализация автомата, то ли в виде аппаратурного модуля, то ли в виде программы, может рассматриваться как автомат с конечной памятью. Более того, любой автомат, не обладающий конечной памятью, за счет расширения его алфавита можно преобразовать в автомат с конечной памятью, сохраняя его функциональность, т.е. выполняемую им функцию. Логические языки спецификации автоматов, не обладающих конечной памятью, гораздо богаче языков спецификации автоматов с конечной памятью. Примером языка, позволяющего специфицировать только автоматы с конечной памятью, является достаточно простой язык L [3]. Использование этого языка позволяет значительно упростить процедуры проектирования, т.е. перехода от декларативной спецификации требований к функционированию автомата к его императивному (процедурному) представлению. Язык L^* является расширением языка L за счет введения дополнительных выразительных средств, необходимых для спецификации автоматов, не обладающих конечной памятью. Заметим, что ни язык L^*, ни какой-либо другой язык логики предикатов первого порядка не дают возможности специфицировать произвольные автоматы. Тем не менее язык L^* позволяет специфицировать достаточно широкий класс автоматов, представляющих интерес с практической точки зрения. Расширение языка L привело к существенному усложнению методов синтеза, основанных на теореме о спецификации [4]. Описанная в [4] версия языка L^* строилась таким образом, чтобы формулы языка могли быть эквивалентно преобразованы к виду, удовлетворяющему условиям теоремы о спецификации. Здесь рассматривается несколько расширенный вариант этого языка, выходящий за рамки требований теоремы о спецификации, в результате чего в некоторых случаях имеющиеся процедуры синтеза могут давать неверный результат. Кроме того, большинство методов проектирования, таких как проверка непротиворечивости спецификаций, их детерминизация, проверка реализуемости открытой системы, верификация и другие, разработаны для спецификаций в языке L. Поэтому большое значение имеет возможность преобразования спецификации из языка L^* в язык L. Учитывая, что

© А.Н. Чеботарев, 2010

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. Для обоснования способа перехода к языку L необходимо охарактеризовать класс формул языка L^*, специфицирующих автоматы с конечной памятью. Очевидно, что все формулы языка L принадлежат этому классу, однако существенный интерес представляют формулы, специфицирующие автоматы с конечной памятью и не принадлежащие языку L. В настоящей работе вводятся необходимые понятия, в терминах которых характеризуется класс формул, специфицирующих автоматы с конечной памятью. Такая характеризация позволяет обосновать переход от спецификации в языке L^* к спецификации в языке L.

ЦИКЛИЧЕСКИЕ Σ-АВТОМАТЫ С КОНЕЧНОЙ ПАМЯТЬЮ

Языки L и L^* используются для спецификации и проектирования реактивных систем, в частности автоматов над бесконечными словами (сверхсловами). Основным объектом спецификации и синтеза является частичный, неинициальный, детерминированный автомат без выходов $A = <\Sigma, Q, \delta_A>$, где Σ — конечный входной алфавит, Q — конечное множество состояний, δ_A: $Q \times \Sigma \to Q$ — частичная функция переходов. Такой автомат назовем Σ-автоматом.

Σ-автомат $A = <\Sigma, Q, \delta_A>$ называется циклическим, если для каждого $q \in Q$ существуют такие $\sigma_1, \sigma_2 \in \Sigma$ и $q_1, q_2 \in Q$, что $q_1 \in \delta_A(q, \sigma_1)$ и $q \in \delta_A(q_2, \sigma_2)$.

Поведение циклического Σ-автомата удобно описывать в терминах множеств слов и сверхслов над алфавитом Σ, поэтому приведем основные определения, связанные с этими понятиями.

Пусть Σ — конечный алфавит, **Z —** множество целых чисел, $\mathbf{N}^+ = \{z \in \mathbf{Z} \mid z > 0\}$ и $\mathbf{N}^- = \{z \in \mathbf{Z} \mid z \le 0\}$. Отображение r множества $\{1, \ldots, n\}$ ($n \in \mathbf{N}^+$) в Σ называется словом длины n в алфавите Σ и обозначается $r = r(1)r(2)\ldots r(n)$. Отображения u: $\mathbf{Z} \to \Sigma$, l: $\mathbf{N}^+ \to \Sigma$ и g: $\mathbf{N}^- \to \Sigma$ 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 сверхловом (обозначается $\ldots g(-2)g(-1)g(0)$) в алфавите Σ. Множество всех слов в алфавите Σ обозначается Σ^*. Множества всех сверхслов и обратных сверхслов в алфавите Σ будем обозначать соответственно Σ^ω и $\Sigma^{-\omega}$. 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. Для $n \in \mathbf{N}^+$ n-префиксом сверхслова l называется слово $l(1)\ldots l(n)$ и n-суффиксом обратного сверхслова g — слово $g(1 - n)\ldots g(0)$.

Сверхслово $l = \sigma_1\sigma_2 \ldots$ в алфавите Σ допустимо в состоянии q Σ-автомата A, если существует такое сверхслово состояний $q_0q_1q_2\ldots$, где $q_0 = q$, что для любого $i = 0, 1, 2, \ldots$ $q_{i+1} = \delta_A(q_i, \sigma_{i+1})$.

Обратное сверхслово в алфавите Σ $\ldots\sigma_{-1}\sigma_0$ представимо состоянием q Σ-автомата A, если существует такое обратное сверхслово состояний $\ldots q_{-2}q_{-1}q_0$, где $q_0 = q$, что для любого $i = -1, -2, \ldots$ $\delta_A(q_i, \sigma_{i+1}) = q_{i+1}$.

Таким образом, с каждым состоянием q_i циклического Σ-автомата ассоциируются два множества сверхслов: множество $S(q_i)$ всех сверхслов, допустимых в состоянии q_i, и множество $P(q_i)$ всех обратных сверхслов, представимых состоянием q_i. Аналогично, для произвольного $k \in \mathbf{N}^+$ рассмотрим множество $S^k(q_i)$ всех слов длины k, допустимых в состоянии q_i, и множество $P^k(q_i)$ всех слов длины k, представимых состоянием q_i.

Состояния q_i и q_j называются эквивалентными, если $S(q_i) = S(q_j)$. Автомат, не имеющий эквивалентных состояний, называется приведенным.

Пусть множество Q состояний Σ-автомата A равно $\{q_1,\ldots, q_n\}$. Семейство множеств сверхслов $(S(q_1), \ldots, S(q_n))$ назовем поведением автомата A.

Два автомата, A_1 и A_2, с поведениями соответственно $(S'_1, \ldots, S'_n)$ и $(S''_1, \ldots, S''_m)$ называются эквивалентными, если каждое S'_i $(i = 1,\ldots, n)$ содержится среди $S''_1, \ldots, S''_m$ и каждое S''_i $(i = 1,\ldots, m)$ содержится среди $S'_1, \ldots, S'_n$.

Пусть $A = <\Sigma, Q, \delta_A>$ и $q_1, q_2 \in Q$, $\sigma \in \Sigma$. Тройку $<q_1, \sigma, q_2>$, такую что $\delta_A(q_1, \sigma) = q_2$, назовем переходом в Σ-автомате A из состояния q_1 в состояние q_2. Будем говорить, что входное слово $r = \sigma_1\ldots\sigma_n$ переводит состояние q' в состояние q'', если для каждого $i = 1, \ldots, n$ в автомате A имеется переход $<q_i, \sigma_i, q_{i+1}>$ и $q' = q_1$, $q''= q_{n+1}$.

Σ-автомат A называется автоматом с конечной памятью, если существует такое натуральное k, что для любого входного слова длины k все состояния автомата A, в которых оно допустимо, переводятся этим словом в эквивалентные состояния. Минимальное такое k называется глубиной памяти автомата.

Утверждение 1. Приведенный циклический Σ-автомат A с множеством состояний Q обладает конечной памятью глубины не более k тогда и только тогда, когда для любых $q_i, q_j \in Q$ $(q_i \neq q_j)$ $P^k(q_i) \cap P^k(q_j) = \varnothing$.

Доказательство. Необходимость. Допустим противное, т.е. что существуют такие состояния $q_i, q_j \in Q$, что $P^k(q_i) \cap P^k(q_j) \neq \varnothing$. Тогда существуют состояния, которые одним и тем же словом длины k переводятся в неэквивалентные состояния (в силу приведенности автомата), что противоречит определению автомата с конечной памятью глубины k.

Достаточность. Если множества слов длины k, представимых состояниями автомата A, не пересекаются, то каждое слово длины k, допустимое для автомата A, содержится только в одном из $P^k(q_i)$ $(q_i \in Q)$. Отсюда следует, что каждое слово длины k переводит все состояния автомата A, в которых оно допустимо, в одно и то же состояние, из чего, в свою очередь, вытекает, что автомат A обладает конечной памятью глубины не более k. Конец доказательства.

ЯЗЫКИ СПЕЦИФИКАЦИИ L И L*

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 формулы вида $\exists t_i\, (t_i \leq t + k_1) \& F_1(t_i) \& \forall t_j ((t_i + k_2 \leq t_j \leq t + k_3) \to F_2(t_j))$, 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 эквивалентного преобразования формул $F(t)$ такого вида часто используется равносильность [5]

$$F(t) \Leftrightarrow F(t - 1) \& h(t) \vee g(t), \tag{1}$$

где $F(t - 1)$ обозначает формулу, полученную из $F(t)$ путем замены t на $t - 1$, $h(t) = F_2(t + k_3)$, а $g(t) = (F_1(t + k_1) \vee \ldots \vee F_1(t - k_2 + k_3 + 1)$, если $k_3 < k_1 + k_2$, или $F_1(t + k_1)$ & $F_2(t + k_3)$ & … & $F_2(t + k_1 + k_2)$, если $k_3 \geq k_1 + k_2$). Правую часть равносильности (1) назовем 1-разверткой $\exists$-формулы $F(t)$.

Каждой замкнутой формуле F ставится в соответствие множество $M(F)$ 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 $\Sigma(\Omega)$, где $\Omega = \{p_1, \ldots, p_m\}$. Алфавит $\Sigma(\Omega)$ представляет собой множество всех двоичных векторов длины m. Символы алфавита $\Sigma(\Omega)$ иногда удобно рассматривать как отображения $\sigma: \Omega \to \{0,1\}$. Пусть $\Omega_1 \subseteq \Omega$, проекцией символа $\sigma \in \Sigma(\Omega)$ на Ω_1 будем называть ограничение отображения σ на Ω_1. Понятие проекции символа на Ω_1 естественным образом распространяется на слова и сверхслова, так что проекция сверхслова в алфавите $\Sigma(\Omega)$ на Ω_1 есть сверхслово в алфавите $\Sigma(\Omega_1)$.

В дальнейшем не будем различать интерпретации и соответствующие им двусторонние сверхслова, поэтому можно говорить об истинностном значении формулы F на двустороннем сверхслове u и значении формулы $F(t)$ в некоторой позиции τ двустороннего сверхслова u.

Пусть $P(F)$ — множество 0-префиксов всех моделей из $M(F)$. Каждому $g \in P(F)$ поставим в соответствие множество сверхслов $S(g) = \{l \in \Sigma^\omega \mid g \cdot l \in M(F)\}$. Другими словами, $S(g)$ состоит из всех тех сверхслов, конкатенация каждого из которых с 0-префиксом g соответствует модели для F. Такие сверхслова назовем допустимыми продолжениями 0-префикса g. Таким образом получим совокупность множеств сверхслов $\mathrm{S}(F) = \{S(g) \mid g \in P(F)\}$. Основная идея, позволяющая использовать формулу F для спецификации автомата, состоит в том, что совокупность множеств сверхслов $\mathrm{S}(F)$ рассматривается как поведение некоторого Σ-автомата.

Пусть $S \subseteq \Sigma^\omega$ и $r \in \Sigma^*$. Левым частным множества сверхслов S по слову r (обозначается $r\backslash S$) называется множество всех таких сверхслов l, что $r \cdot l \in S$, т.е. $r\backslash S = \{l \mid r \cdot l \in S\}$. Таким образом, если $r\backslash S = S_1$, то $r \cdot S_1 \subseteq S$.

В [4] показано, что Σ-автомат $A(F) = <\Sigma, Q_A, \delta>$, поведение которого совпадает с $\mathrm{S}(F) = \{S_1, \ldots, S_n\}$, может быть определен следующим образом: $\Sigma = \Sigma(\Omega)$, $Q_A = \{q_1, \ldots, q_n\}$ и для $q_i \in Q_A$, $\sigma \in \Sigma$ значение $\delta(q_i, \sigma)$ определено и равно q_j тогда и только тогда, когда $\sigma\backslash S_i = S_j$, если $\sigma\backslash S_i = \varnothing$, то значение $\delta(q_i, \sigma)$ не определено. Если функцию переходов автомата $A(F)$ определить как $\delta^*: Q_A \times \Sigma^* \to Q_A$, то для $q_i \in Q_A$, $r \in \Sigma^*$ $\delta^*(q_i, r) = q_j$ тогда и только тогда, когда $r\backslash S_i = S_j$. Заметим, что если $r\backslash S_i = S_j$ и $g \in P_i$, то $g \cdot r \in P_j$, и наоборот, если $g \in P_i$ и $g \cdot r \in P_j$, то $r\backslash S_i = S_j$.

Два 0-префикса, $g_1, g_2 \in P(F)$, назовем эквивалентными, если множества всех допустимых продолжений для них совпадают, т.е. $S(g_1) = S(g_2)$. Таким образом, совокупности множеств $\mathrm{S}(F) = \{S_1, \ldots, S_n\}$ соответствует разбиение множества $P(F)$ на классы эквивалентности $P_1, \ldots, P_n$.

Утверждение 2. Множество k-суффиксов ($k \in \mathbf{N}^+$) всех обратных сверхслов из класса P_j совпадает с множеством всех слов длины k, представимых состоянием q_j автомата $A(F)$.

Доказательство. 1. Пусть слово r — суффикс обратного сверхслова $g \in P_j$, т.е. $g = g_1 \cdot r$, и $g_1 \in P_i$. Тогда $r\backslash S_i = S_j$, т.е. $\delta^*(q_i, r) = q_j$. Таким образом, слово r представимо состоянием q_j.

2. Пусть r представимо состоянием q_j, т.е. существует такое состояние q_i, что $\delta^*(q_i, r) = q_j$. Как следует из определения функции переходов δ^* автомата $A(F)$, $r\backslash S_i = S_j$. Пусть $g \in P_i$. Тогда $g \cdot r \in P_j$, т.е. r — суффикс обратного сверхслова, принадлежащего P_j. Конец доказательства.

Хотя это утверждение справедливо для слов произвольной длины, оно не справедливо для сверхслов. Множество P_j может не совпадать с множеством всех обратных сверхслов, представимых состоянием q_j.

Непересекающиеся множества P_i и P_j обратных сверхслов в одном и том же алфавите называются k-разделимыми, если существует такое $k \in \mathbf{N}^+$, что множества k-суффиксов всех обратных сверхслов из P_i и всех обратных сверхслов из P_j не пересекаются. Два непересекающихся множества обратных сверхслов называются конечно неразделимыми, если не существует такого $k \in \mathbf{N}^+$, что они k-разделимы.

Теорема 1. Формула $F = \forall t F(t)$ языка L^* специфицирует автомат с конечной памятью тогда и только тогда, когда существует такое $k \in \mathbf{N}^+$, что классы эквивалентности обратных сверхслов из $P(F)$ попарно k-разделимы.

Справедливость этой теоремы следует непосредственно из утверждений 1 и 2.

Формула $F(t)$, с единственной свободной переменной t, называется k-ограниченной ($k \in \mathbf{Z}$), если для любого $\tau \in \mathbf{Z}$ значения формулы $F(\tau - k)$ на всех двусторонних сверхсловах, имеющих одинаковые τ-префиксы, совпадают. Смысл данного определения состоит в том, что при вычислении истинностного значения k-ограниченной формулы $F(t)$ в позиции τ интерпретации u не нужно рассматривать значения предикатов для $t > \tau + k$. Примером 0-ограниченной формулы $F(t)$ может служить формула $\exists t_1(t_1 \le t) \& y(t_1) \& \forall t_2((t_1 < t_2 \le t) \to \neg x(t_2))$. Будем рассматривать 0-ограниченные формулы как способ задания множеств обратных сверхслов, а именно, будем полагать, что формула $F(t)$ задает множество 0-префиксов всех тех двусторонних сверхслов, на которых истинна $F(0)$. Заметим, что множества обратных сверхслов, задаваемые $\exists$-формулой и ее отрицанием, конечно неразделимы. Таким образом, необходимым условием того, что автомат, специфицируемый формулой F, не обладает конечной памятью, является наличие в ней $\exists$-подформул. Однако наличие в спецификации $\exists$-подформулы не всегда приводит к тому, что специфицируемый автомат не обладает конечной памятью. Ниже покажем, что любую спецификацию $\forall t F(t)$ в языке L^* за счет введения дополнительных предикатных символов можно преобразовать в спецификацию $\forall t F_z(t)$ в этом же языке, специфицирующую автомат с конечной памятью.

Формула $F_z(t)$ получается из формулы $F(t)$ следующим образом. Все имеющиеся в $F(t)$ $\exists$-подформулы вида $\varphi_i(t_i)$ ($i = 1, 2, \ldots, n$) последовательно, начиная с подформул, имеющих максимальную глубину вложения в другие $\exists$-подформулы, заменяются атомами вида $z_i(t_i + r_i)$, где r_i — ранг заменяемой $\exists$-подформулы [5], а z_i — новый предикатный символ, отсутствующий в текущей формуле. Полученную формулу обозначим $f_z(t)$. С каждым введенным предикатом $z_i(t)$, соответствующим формуле $\varphi_i(t_i)$, ассоциируется формула $z_i(t) \leftrightarrow \varphi_i(t - r_i)$, где $\varphi_i(t - r_i)$ получается из $\varphi_i(t_i)$ заменой t_i на $t - r_i$. Формула $F_z(t)$ имеет вид $f_z(t) \mathop{\&}\limits_{i=1}^{n} (z_i(t) \leftrightarrow \varphi_i(t - r_i))$.

Утверждение 3. Проекция любой модели для формулы $\forall t F_z(t)$ на сигнатуру Ω формулы $F(t)$ является моделью для формулы $\forall t F(t)$, и наоборот, для любой модели u формулы $\forall t F(t)$ существует модель для формулы $\forall t F_z(t)$, проекция которой на Ω совпадает с u.

Доказательство. Пусть u_1 — модель для формулы $\forall t F_z(t)$. Покажем, что формула $F(t)$ истинна в произвольной позиции τ проекции u_1 на Ω. Поскольку u_1 — модель для формулы $\forall t F_z(t)$, то $f_z(t)$ истинна в позиции τ и для каждого $\tau \in \mathbf{Z}$ значение $z_i(\tau + r_i)$ совпадает со значением $\varphi_i(\tau)$. Формула $F(t)$ получается, если в $f_z(t)$ вместо всех вхождений атомов вида $z_i(t_i + r_i)$ подставить правые части эквивалентностей для $z_i(t)$, заменив в них t на $t_i + r_i$. Отсюда следует, что $F(t)$ истинна в позиции τ проекции u_1 на Ω. Обратно, имея модель u для формулы $\forall t F(t)$, легко построить соответствующую модель для $\forall t F_z(t)$, положив для каждого

$\tau \in \mathbf{Z}$ и $i = 1, 2, \ldots, n$ $z_i(\tau) = \varphi_i(\tau - r_i)$. Остальные предикатные символы в каждой позиции τ имеют те же значения, что и в модели u. Конец доказательства.

Пусть 1-развертка $\exists$-формулы $\varphi(t)$ имеет вид $\varphi(t-1)\&h(t) \vee g(t)$, где формулы $h(t)$ и $g(t)$ не содержат вхождений $\exists$-формул.

Утверждение 4. Всякая модель для формулы $F = \forall t(z(t) \leftrightarrow \varphi(t))$, где z — предикатный символ, не содержащийся в $\varphi(t)$, является моделью для формулы $F_1 = \forall t(z(t) \leftrightarrow (z(t-1)\&h(t) \vee g(t)))$.

Доказательство. Пусть u — модель для формулы F, тогда в любой позиции $\tau \in \mathbf{Z}$ интерпретации u значения $z(\tau)$ и $\varphi(\tau)$ совпадают. Из этого следует, что формулы $\forall t(z(t) \leftrightarrow (\varphi(t-1)\&h(t) \vee g(t)))$ и $F_1 = \forall t(z(t) \leftrightarrow (z(t-1)\&h(t) \vee g(t)))$ в любой позиции интерпретации u принимают одно и то же значение. Таким образом, в силу равносильности (1) u — модель для формулы F_1. Конец доказательства.

Из этого утверждения следует, что всякая модель для формулы $F_z = \forall t f_z(t) \mathop{\&}\limits_{i=1}^{n} (z_i(t) \leftrightarrow \varphi_i(t))$ является моделью для формулы $f_z = \forall t f_z(t) \mathop{\&}\limits_{i=1}^{n} (z_i(t) \leftrightarrow (z_i(t-1)\&h_i(t) \vee g_i(t)))$, где $h_i(t)$ и $g_i(t)$ определяются 1-развертками соответствующих формул $\varphi_i(t)$.

Пусть g — 0-префикс модели для формулы $F_z = \forall t F_z(t) = \forall t f_z(t) \mathop{\&}\limits_{i=1}^{n} (z_i(t) \leftrightarrow \varphi_i(t))$ и 1-развертки для формул $\varphi_i(t)$ имеют соответственно вид $\varphi_i(t-1)\&h_i(t) \vee g_i(t)$ $(i = 1, 2, \ldots, n)$.

Утверждение 5. Множества всех допустимых продолжений для g, определяемых формулой F_z и формулой $f_z = \forall t f(t) = \forall t f_z(t) \mathop{\&}\limits_{i=1}^{n} (z_i(t) \leftrightarrow (z_i(t-1)\&h_i(t) \vee g_i(t)))$, совпадают.

Доказательство. Пусть l — допустимое продолжение для g, определяемое формулой f_z, покажем, что оно допустимо и для формулы F_z, т.е. что в любой его позиции τ истинна формула $F_z(t)$. Доказательство проведем индукцией по длине префикса сверхслова l, соответствующего позиции τ этого сверхслова. Базис: $\tau = 0$. Префикс нулевой длины соответствует самой правой позиции обратного сверхслова g. Поэтому значения $f(t)$ и $F_z(t)$ в этой позиции совпадают. Покажем теперь, что из совпадения значений $f(t)$ и $F_z(t)$ в позиции τ следует совпадение их значений в позиции $\tau + 1$. Для этого достаточно показать, что $z_i(\tau+1) = \varphi_i(\tau+1)$ для всех $i = 1, 2, \ldots, n$. Из истинности $F_z(t)$ в позиции τ следует, что $z_i(\tau) = \varphi_i(\tau)$ $(i = 1, 2, \ldots, n)$, поэтому из $\varphi_i(\tau+1) = (\varphi_i(\tau)\&h_i(\tau+1) \vee g_i(\tau+1))$ и $z_i(\tau+1) = (z_i(\tau)\&h_i(\tau+1) \vee g_i(\tau+1))$ следует, что $z_i(\tau+1) = \varphi_i(\tau+1)$ $(i = 1, 2, \ldots, n)$. Таким образом, значения $f(t)$ и $F_z(t)$ в позиции $\tau + 1$ совпадают.

Обратное утверждение, т.е. что допустимое для формулы F_z продолжение обратного сверхслова g совпадает с допустимым его продолжением для формулы f_z, следует из того, что множество моделей для F_z содержится в множестве моделей для f_z. Конец доказательства.

Теорема 2. Формула $F_z = \forall t f_z(t) \mathop{\&}\limits_{i=1}^{n} (z_i(t) \leftrightarrow \varphi_i(t))$, полученная из формулы языка L^* описанным выше способом, специфицирует автомат с конечной памятью.

Доказательство. Из утверждений 4 и 5 следует, что $S(F_z) \subseteq S(f_z)$, т.е. автомат, специфицируемый формулой F_z, является подавтоматом автомата, специфицируемого формулой f_z. Поскольку f_z — формула языка L, специфицируемый ею автомат обладает конечной памятью [6]. Любой его подавтомат также обладает конечной памятью. Конец доказательства.

ЗАКЛЮЧЕНИЕ

Известно, что любой (конечный) автомат, не обладающий конечной памятью, можно преобразовать в автомат с конечной памятью путем добавления переменных, определяющих структуру выходного алфавита, и коррекции функции выходов таким образом, чтобы сохранялось поведение соответствующего Σ-автомата относительно исходного алфавита. В настоящей статье такое преобразование осуществляется регулярным образом на уровне спецификации требований к функционированию автомата. Спецификация в языке L^* за счет введения дополнительных предикатных символов преобразуется в спецификацию в этом же языке, но специфицирующую автомат с конечной памятью. При этом проекция всех моделей преобразованной спецификации на исходную сигнатуру совпадает с множеством всех моделей для исходной спецификации. В статье доказывается, что преобразованная спецификация обладает всеми необходимыми свойствами.

Поскольку класс автоматов с конечной памятью совпадает с классом автоматов, специфицируемых в языке L, для преобразованной спецификации в языке L^* существует эквивалентная ей спецификация в языке L. При наличии способа получения такой спецификации синтез требуемого автомата может быть осуществлен средствами, разработанными для языка L. Таким способом может быть решено несколько проблем. Во-первых, устраняется необходимость проверки состояний синтезированного автомата на фиктивность, во-вторых, расширяются возможности применения методов синтеза к спецификациям, не удовлетворяющим теореме о спецификации [4]. В связи с этим в настоящей работе рассматривается расширенный вариант языка L^*, выходящий за рамки требований теоремы о спецификации, на которой основаны все методы синтеза. Расширение языка L^* состоит в том, что в качестве подформул, входящих в $\exists$-формулы языка, могут использоваться не только $\exists$-формулы и формулы языка L, но и произвольные формулы языка L^*.

СПИСОК ЛИТЕРАТУРЫ

1. Гилл А. Введение в теорию конечных автоматов. — М.: Наука, 1966. — 227 с.
2. Брауэр В. Введение в теорию конечных автоматов. — М.: Радио и связь, 1987. — 392 с.
3. Чеботарев А.Н. Об одном подходе к функциональной спецификации автоматных систем. I // Кибернетика и системный анализ. — 1993. — № 3. — С. 31–42.
4. Чеботарев А.Н. Расширение логического языка спецификации и проблема синтеза // Там же. — 1996. — № 6. — С. 11–27.
5. Чеботарев А.Н. Синтез процедурного представления автомата, специфицированного в логическом языке L^*. I // Там же. — 1997. — № 4. — С. 60–74.
6. Чеботарев А.Н. Об одном подходе к функциональной спецификации автоматных систем. III // Там же. — 1993. — № 6. — С. 3–14.

Поступила 14.02.2009