УДК 519.713.1

А.Н. ЧЕБОТАРЕВ

НЕКОТОРЫЕ ПОДМНОЖЕСТВА МОНАДИЧЕСКОЙ ЛОГИКИ ПЕРВОГО ПОРЯДКА (MFO), ИСПОЛЬЗУЕМЫЕ ДЛЯ СПЕЦИФИКАЦИИ И СИНТЕЗА Σ-АВТОМАТОВ

Аннотация. Рассмотрены два фрагмента, LP и LF, логики первого порядка с ограниченными кванторами, используемые для спецификации трансдьюсеров. Логика LP позволяет характеризовать текущее поведение системы на основе ее поведения в прошлом, а LF — на основе поведения в будущем. Определены два вида семантик для этих логик и исследованы свойства специфицируемых в них автоматов.

Ключевые слова: логики первого порядка, формулы прошедшего времени, формулы будущего времени, автоматная семантика, симметричные формулы.

ВВЕДЕНИЕ

В задачах спецификации, верификации и синтеза реактивных систем существенную роль играют логические языки, т.е. языки, в основе которых лежат различные фрагменты логик одноместных предикатов. Это такие логики, как монадические логики второго порядка, т.е. логики второго порядка с одноместными предикатами, типичным представителем которых является S1S [1] (монадическая логика второго порядка с одним последователем); μ-исчисления [2], основанные на использовании операторов наименьшей и наибольшей неподвижной точки; темпоральные логики с линейным (LTL) и ветвящимся (CTL, CTL*) временем [3] и, наконец, различные логики первого порядка, из которых наиболее популярна логика MFO[<] [4], т.е. логика первого порядка с одноместными предикатами, интерпретируемая на линейных структурах, другими словами, монадическая теория первого порядка линейно упорядоченных множеств. Все перечисленные логики — это разрешимые логики, имеющие различные выразительные возможности (т.е. классы описываемых свойств) и соответственно различную временную сложность разрешающей процедуры, в частности процедуры проверки выполнимости формулы на модели.

Логические языки различным образом используются при верификации и синтезе реактивных систем, хотя в том и в другом случае соответствующая спецификация определяет требования к поведению системы и среды в процессе их взаимодействия. В первом случае проверяется выполнимость спецификации на математической модели системы, в качестве которой, как правило, используется структура Крипке. Во втором случае решается задача построения такой системы, поведение которой при ее взаимодействии со средой будет удовлетворять требованиям спецификации, независимо от возможного поведения среды. Для решения этой задачи обычно используется игровой подход, основанный на рассмотрении бесконечной игры между системой и средой, с которой она взаимодействует. Построение корректного алгоритма функционирования системы заключается в определении выигрышной для нее стратегии. Спецификации, для которых выигрышные стратегии существуют, называются реализуемыми [5].

Языки спецификации, рассматриваемые в настоящей работе, ориентированы на задачи синтеза. Предполагается, что спецификация реактивной системы состоит из двух частей: спецификации требований к поведению проектируемой части системы, моделируемой X/Y-автоматом (трансдьюсером), и информации о возможном поведении среды, представленной в виде спецификации недетерминированного X/Y-автомата. Очевидно, что требования к поведению специ-

фицируемого трансдьюсера, в явной или неявной форме ограничивающие поведение среды, не могут быть реализованы. Отсюда возникает проблема согласования спецификации системы со спецификацией среды, решение которой приведено в [6]. Понятие согласуемости спецификаций системы и среды связано с ограничением свойств, определяемых этими спецификациями, свойствами безопасности (safety) [7], что позволяет решать задачу согласования как на уровне спецификаций, так и на уровне синтезированных автоматов [8]. Таким образом, в рассматриваемом подходе к проектированию реактивных систем проблемы синтеза и согласования спецификаций разделены.

Для решения задач синтеза в [9, 10] предложены весьма ограниченные фрагменты логики MFO[<], позволившие получить достаточно эффективные алгоритмы. В настоящей работе существенно расширены возможности языков спецификации практически без усложнения процедуры синтеза. Рассматриваются два фрагмента логики MFO[<], один из которых позволяет характеризовать текущее поведение системы на основе поведения в прошлом, а другой — на основе поведения в будущем. Определяется автоматная семантика этих логик, и исследуются свойства специфицируемых в них автоматов. Поскольку в статье не рассматриваются вопросы взаимодействия проектируемого автомата и среды, предикатные символы, фигурирующие в спецификации, не разделяются на входные и выходные, что необходимо при спецификации трансдьюсера. Поэтому в качестве автоматной модели, определяющей семантику логической формулы, используется Σ-автомат, т.е. автомат без выходов, входной алфавит которого соответствует произведению входного и выходного алфавитов трансдьюсера. При определении автоматной семантики логического языка он, так же как и автомат, рассматривается как формализм для задания множеств бесконечных слов (сверхслов). Необходимые сведения о множествах бесконечных слов (ω-языках) и используемых автоматных моделях изложены в следующем разделе.

БЕСКОНЕЧНЫЕ СЛОВА И АВТОМАТЫ

Пусть Σ — конечный алфавит, $\mathbf{Z}$ — множество целых чисел, $\mathbf{N}^+ = \{z \in \mathbf{Z} \mid z > 0\}$, $\mathbf{N}^- = \{z \in \mathbf{Z} \mid z \le 0\}$. Отображение r множества $\{1, \dots, n\}$ $(n \ge 0)$ в Σ называется словом длины n в алфавите Σ и обозначается $\sigma_1\sigma_2\dots\sigma_n$, где $\sigma_i = r(i)$ для всех $1 \le i \le n$. Слово длины 0 (пустое слово) обозначается ε. Отображение $u: \mathbf{Z} \to \Sigma$ называется двусторонним сверхсловом ($\mathbf{Z}$-словом) в алфавите Σ и обозначается $\dots\sigma_{-2}\sigma_{-1}\sigma_0\sigma_1\sigma_2\dots$, где $\sigma_i = u(i)$, $i \in \mathbf{Z}$. Отображения $l: \mathbf{N}^+ \to \Sigma$ и $g: \mathbf{N}^- \to \Sigma$ называются соответственно сверхсловом (обозначается $\sigma_1\sigma_2\cdots$, где $\sigma_i = l(i)$, $i \in \mathbf{N}^+$) и обратным сверхсловом (обозначается $\cdots\sigma_{-2}\sigma_{-1}\sigma_0$, где $\sigma_i = g(i)$ для $i \in \mathbf{N}^-$). Множество всех слов в алфавите Σ, включая пустое слово, обозначается Σ^*, множество всех сверхслов — Σ^ω, а множество всех обратных сверхслов — $\Sigma^{-\omega}$. Множество всех двусторонних сверхслов в алфавите Σ обозначим $\Sigma^{\mathbf{Z}}$. На множестве $\Sigma^* \cup \Sigma^{-\omega} \cup \Sigma^\omega$ определим обычным образом (частичную) операцию конкатенации «$\cdot$», которую распространим также на множества слов и сверхслов. Пусть $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^* \cup \Sigma^\omega$, тогда $L_1 \cdot L_2 = \{r \cdot l \mid r \in L_1, l \in L_2\}$.

Отрезок $u(\tau)u(\tau+1)\dots u(\tau+k)$, $k \ge 0$, двустороннего сверхслова u обозначается $u(\tau, \tau+k)$. Бесконечные отрезки $u(-\infty, k)$ и $u(k+1, \infty)$, где $k \in \mathbf{Z}$, назовем соответственно k-префиксом и k-суффиксом двустороннего сверхслова u. Для $n \in \mathbf{N}^+$ n-префиксом сверхслова l называется слово $l(1)\dots l(n)$. Если значение позиции в $\mathbf{Z}$-слове или сверхслове несущественно, то понятие префикса и суффикса определяется следующим образом. Обратное сверхслово g называется префиксом $\mathbf{Z}$-слова u, а сверхслово l — его суффиксом, если $g \cdot l = u$. Слово r называется префиксом сверхслова l, если существует такое сверхслово l_1, что $l = r \cdot l_1$.

Множество Σ^ω будем рассматривать как топологическое пространство с так называемой естественной топологией [11]. Открытыми множествами в этой топологии являются все множества вида $K \cdot \Sigma^\omega$, где $K \subseteq \Sigma^*$. Дополнение открытого множества в Σ^ω называется замкнутым множеством. Замкнутое множество характеризуется следующим утверждением [11].

Утверждение 1. Множество L замкнуто тогда и только тогда, когда каждое сверхслово, не принадлежащее L, имеет конечный префикс, не являющийся префиксом никакого сверхслова из L.

Другими словами, всякое сверхслово, каждый префикс которого есть префикс сверхслова из L, принадлежит L.

Пусть $R \subseteq \Sigma^\omega$. Наименьшее замкнутое множество, включающее R, называется его замыканием и обозначается $\overline{R}$.

Утверждение 2. Сверхслово $l \in \Sigma^\omega$ принадлежит $\overline{R}$ тогда и только тогда, когда каждый префикс сверхслова l есть префикс сверхслова из R.

Симметричным образом вводится топология на $\Sigma^{-\omega}$.

Для описания множеств сверхслов различного типа будем использовать регулярные выражения. Предварительно определим над множествами слов операции конечной итерации «*» и бесконечной итерации. Бесконечная итерация в зависимости от типа результата имеет три модификации: для сверхслов — «$^\omega$», для обратных сверхслов — «$^{-\omega}$» и для **Z**-слов — «$^{\mathbf{Z}}$».

Пусть $R \subseteq \Sigma^*$, операции итерации над R определяются следующим образом: $R^* = \{r_1 r_2 \dots r_n \mid n \geq 0, r_i \in R\}$, заметим, что $n = 0$ соответствует пустому слову;

$$R^\omega = \{r_0 r_1 r_2 \dots \mid \text{для всех } i \geq 0 \ r_i \in R \setminus \varepsilon\};$$

$$R^{-\omega} = \{\dots r_2 r_1 r_0 \mid \text{для всех } i \geq 0 \ r_i \in R \setminus \varepsilon\};$$

$$R^{\mathbf{Z}} = \{\dots r_{-2} r_{-1} r_0 r_1 r_2 \dots \mid \text{для всех } i \in \mathbf{Z} \ r_i \in R \setminus \varepsilon\}.$$

Регулярные выражения для описания регулярных множеств сверхслов, обратных сверхслов и **Z**-слов представляют собой конечные объединения соответственно выражений вида UR^ω, $R^{-\omega}U$ и для **Z**-слов $R_1^{-\omega}UR_2^\omega$ или $R^{\mathbf{Z}}$, где R, R_1, R_2 и U — выражения, построенные из символов алфавита Σ с помощью операций объединения, конкатенации (символ конкатенации будем часто опускать) и конечной итерации.

Важный класс регулярных множеств сверхслов (регулярных ω-языков) составляют беззвездочные регулярные ω-языки. Для их описания определим над множествами слов еще одну операцию: $\lim R$, результат которой представляет собой множество всех таких сверхслов, бесконечное количество префиксов которых содержится в R. Беззвездочные ω-языки — это такие множества сверхслов, которые можно представить выражениями вида $\bigcup_{i=1}^{m} U_i \cdot \lim V_i$, где U_i и V_i — выражения, построенные из конечных подмножеств множества Σ^* с помощью операций конкатенации, объединения, пересечения и дополнения до Σ^*.

Заметим, что многие выражения, использующие $*$, задают беззвездочные языки, поскольку существуют эквивалентные им (но более громоздкие) выражения, не использующие $*$. Например, $(ab)^*$ эквивалентно $(b\varnothing^C \cup \varnothing^C a \cup \cup \varnothing^C (aa \cup bb)\varnothing^C)^C$, где верхний индекс C обозначает дополнение до $\{a, b\}^*$, таким образом, $\varnothing^C = \Sigma^*$.

Рассмотрим теперь некоторые автоматные модели, используемые для определения автоматной семантики логических языков в задачах синтеза реактивных систем. Для этой цели обычно используются автоматы над бесконечными слова-

ми или деревьями, распознающие ω-языки, определяемые формулами спецификации [1]. Из автоматов этого типа здесь ограничимся только автоматом Бюхи как первой автоматной модели, предложенной Бюхи для решения проблемы разрешимости логики S1S.

Автомат Бюхи A определяется пятеркой объектов $< Q, \Sigma, Q_0, \Delta, F >$, где Q — конечное множество состояний, Σ — входной алфавит, $Q_0 \subseteq Q$ — множество начальных состояний, $\Delta \subseteq Q \times \Sigma \times Q$ — отношение переходов, $F \subseteq Q$ — множество финальных состояний. Вычислением в автомате Бюхи на входном сверхслове l называется сверхслово состояний, начинающееся в состоянии из Q_0 и генерируемое автоматом в ответ на сверхслово l. Вычисление называется успешным, если в нем бесконечное число раз встречается хотя бы одно состояние из F. Автомат Бюхи распознает входное сверхслово l, если существует успешное вычисление на этом сверхслове. Множество всех сверхслов в алфавите Σ, распознаваемых автоматом A, называется ω-языком, распознаваемым этим автоматом. Автомат Бюхи называется детерминированным, если Q_0 состоит из одного состояния и Δ представляет собой функцию $\delta: Q \times \Sigma \to Q$. Класс ω-языков, распознаваемых детерминированными автоматами Бюхи, меньше класса ω-языков, распознаваемых недетерминированными автоматами. Поэтому не существует алгоритма детерминизации автоматов в классе автоматов Бюхи. Обычно в результате детерминизации автомата Бюхи получают детерминированный автомат другого типа, например автомат Рабина. Важное значение автоматов Бюхи состоит в том, что класс распознаваемых ими ω-языков совпадает с регулярными ω-языками и языками, определимыми в монадических логиках второго порядка. Поэтому в задачах синтеза обычно логическую спецификацию транслируют в недетерминированный автомат Бюхи.

Для характеризации ω-языков, определимых в логиках первого порядка, используются так называемые бессчетчиковые автоматы Бюхи [12]. Обозначим $L_{q,p}$ множество слов, переводящих автомат из состояния q в состояние p.

Определение 1. Автомат Бюхи $A = < Q, \Sigma, Q_0, \Delta, F >$ называется бессчетчиковым, если для любых $q \in Q$, $r \in \Sigma^*$ и $m \geq 1$ из $r^m \in L_{q,q}$ следует, что $r \in L_{q,q}$.

Перейдем к рассмотрению автоматов-преобразователей (трансдьюсеров), ассоциируемых с формулами монадических логик первого порядка.

Определение 2. Конечный неинициальный X/Y-автомат A — это четверка $< X, Y, Q, \chi_A >$, где X, Y, Q — конечные множества соответственно входных символов, выходных символов и состояний, а $\chi_A : Q \times X \times Y \to 2^Q$ — функция переходов автомата.

Будем X/Y-автомат A называть квазидетерминированным, если для любых $q \in Q$, $x \in X$, $y \in Y$ $|\chi_A(q, x, y)| \leq 1$. Квазидетерминированные X/Y-автоматы удобно рассматривать как детерминированные частичные автоматы без выхода, с входным алфавитом $\Sigma = X \times Y$. Такой автомат $A = < \Sigma, Q, \delta_A >$, где $\delta_A : Q \times \Sigma \to Q$ — частичная функция, называется Σ-автоматом.

Предполагается, что символы алфавита Σ представляют собой двоичные векторы длины m, что соответствует кодированию абстрактных символов наборами значений двоичных переменных из множества $\Omega = \{x_1, \ldots, x_m\}$. Этим переменным соответствуют предикатные символы в логических спецификациях автоматов.

Определение 3. 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 $q_1 = \delta_A(q, \sigma_1)$ и $q = \delta_A(q_2, \sigma_2)$.

В дальнейшем под Σ-автоматом будем понимать циклический Σ-автомат. Такой автомат можно однозначно охарактеризовать в терминах допустимых сверхслов.

Определение 4. Сверхслово $l = \sigma_1 \sigma_2 \ldots$ в алфавите Σ допустимо в состоянии q 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 автомата A, если оно допустимо хотя бы в одном из его состояний. Множество всех сверхслов, допустимых для автомата A, обозначим $W(A)$. Два Σ-автомата, A_1 и A_2, назовем слабо эквивалентными, если $W(A_1) = W(A_2)$.

Определение 5. Обратное сверхслово $\ldots \sigma_{-2}\sigma_{-1}\sigma_0$ в алфавите Σ 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 $i = -1, -2, -3, \ldots$ выполняется $\delta_A(q_i, \sigma_{i+1}) = q_{i+1}$.

Таким образом, с каждым состоянием q циклического Σ-автомата ассоциируются два множества сверхслов: множество $W(q)$ всех сверхслов, допустимых в состоянии q, и множество $P(q)$ всех обратных сверхслов, представимых состоянием q.

Определение 6. Пусть множество Q состояний Σ-автомата A равно $\{q_1, \ldots, q_n\}$. Семейство множеств сверхслов $S(A) = (W_1, \ldots, W_n)$, где $W_i = W(q_i)$ $(i = 1, \ldots, n)$ назовем поведением автомата A.

Два Σ-автомата, A_1 и A_2, с поведениями соответственно $(W_1', \ldots, W_n')$ и $(W_1'', \ldots, W_m'')$ называются строго эквивалентными, если каждое W_i' $(i = 1, \ldots, n)$ содержится среди $W_1'', \ldots, W_m''$ и каждое W_i'' $(i = 1, \ldots, m)$ содержится среди $W_1', \ldots, W_n'$.

Автомату A с поведением $(W_1, \ldots, W_n)$ поставим в соответствие обратный ему автомат A^- с поведением $(P_1^-, \ldots, P_n^-)$, где P_i $(i = 1, \ldots, n)$ — множество обратных сверхслов, представимых состоянием q_i автомата A. Граф автомата, обратного автомату A, получается из графа автомата A путем изменения направления его дуг на обратное.

ЛОГИКИ ПЕРВОГО ПОРЯДКА

Монадические логики первого порядка используются для задания ω-языков, что позволяет установить тесную связь между логиками и автоматами. К наиболее изученным таким логикам принадлежат логика F1S (монадическая логика первого порядка с одним последователем), обычно рассматриваемая как фрагмент первого порядка логики S1S, и логика MFO [<], сигнатура которой содержит двуместный предикатный символ $<$, интерпретируемый как линейный порядок на натуральных числах. Так как функция следования, имеющаяся в сигнатуре логики F1S, выражается в логике первого порядка через отношение $<$, а обратное неверно, то логика MFO [<] более выразительна, чем F1S. В логику MFO [<] часто добавляют функцию следования SUC, что не отражается на ее выразительности. Ниже будет рассмотрена именно такая логика (MFO). Строго говоря, из-за аксиомы индукции для линейного порядка, выражающейся формулой второго порядка, MFO также логика второго порядка, однако общепринято называть ее логикой первого порядка.

Алфавит этой логики состоит из символов следующих видов:

1) множество переменных $V = \{t, t_1, t_2, \ldots\}$;
2) множество одноместных предикатных символов $\Omega = \{P_1, P_2, \ldots\}$;
3) двуместные предикатные символы $=$ и $<$;
4) унарный функциональный символ SUC;
5) логические связки;
6) кванторы $\exists$ и $\forall$;
7) скобки.

Кроме того, дополнительно определяются отношения $t_1 \le t_2$, $t_1 \ge t_2$, а также вводится сокращение $\mathrm{SUC}(t) = t+1$, $\mathrm{SUC}(\mathrm{SUC}(t)) = t+2$ и т.д. Таким образом, термы имеют вид $(t_i + k)$, где $t_i \in V$, $k \in \mathbf{N}$. Атомарные формулы могут быть двух типов: $P_i(\tau)$ или $\tau_1 \rho \tau_2$, где $P_i \in \Omega$, $\rho \in \{=, <, >, \le, \ge\}$, а τ, τ_1, τ_2 — термы. Формулы строятся из атомарных формул с помощью логических связок, кванторов, применяемых к предметным переменным, и, возможно, скобок. Обозначение $\varphi(t_1, \ldots, t_n)$ используется для указания того, что все свободные переменные в формуле содержатся среди $t_1, \ldots, t_n$. Формула, не содержащая свободных пере-

менных, называется замкнутой, или предложением. Интерпретируются формулы на множестве натуральных чисел **N**.

Множество сверхслов в алфавите Σ (ω-язык), задаваемое предложением φ, определяется следующим образом. Между одноместными предикатными символами в формуле φ и символами алфавита Σ устанавливается взаимно однозначное соответствие. Сверхслово $l \in \Sigma^{\omega}$ определяет набор предикатов P_{σ}, $\sigma \in \Sigma$, истинных в тех и только тех его позициях, в которых имеется символ σ. Таким образом, сверхслово l можно рассматривать как интерпретацию формулы φ, при которой каждый предикатный символ, соответствующий символу σ, интерпретируется предикатом P_{σ}. Истинность предложения φ при этой интерпретации будем обозначать $l \models \varphi$. Если символы алфавита Σ закодированы наборами значений двоичных переменных $x_1, \ldots, x_n$, то взаимно однозначное соответствие устанавливается между предикатными символами из Ω и этими переменными. В этом случае сверхслово l определяет набор предикатов P_i, $i = 1, \ldots, n$, истинных в тех и только в тех позициях сверхслова l, в которых i-й элемент набора значений переменных равен 1. Формула φ задает ω-язык $L(\varphi) = \{l \in \Sigma^{\omega} \mid l \models \varphi\}$.

Утверждение 3 [12]**.** Класс ω-языков, задаваемых формулами логики MFO, совпадает с классом беззвездочных регулярных ω-языков и классом языков, распознаваемых бессчетчиковыми автоматами Бюхи.

В задачах спецификации и верификации реактивных систем широкое распространение получили темпоральные логики: LTL, CTL, CTL* и др. Здесь ограничимся рассмотрением только логики LTL. Пропозициональную темпоральную логику LTL можно рассматривать как вариант логики MFO, интерпретируемой на сверхсловах, где вместо кванторов по предметным переменным используются темпоральные операторы. Формулы логики LTL строятся из констант **true**, **false** и пропозициональных переменных $p_1, p_2, \ldots, p_n$ путем применения к ним логических связок, унарных темпоральных операторов **X** (в следующий момент), **F** (когда-нибудь), **G** (всегда) и бинарного оператора **U** (до тех пор, пока). В качестве интерпретаций используются сверхслова в алфавите $\Sigma = \{0,1\}^n$, причем i-й позиции в символе этого алфавита соответствует переменная p_i.

В дальнейшем вместо «формула логики LTL» или «формула логики MFO» будем писать «LTL-формула» или «MFO-формула». Поскольку формулы логик LTL и MFO интерпретируются в одном и том же классе структур, между LTL-формулами и MFO-формулами вида $F(t)$ можно определить эквивалентность так, что φ эквивалентна $F(t)$, если они принимают одно и то же истинностное значение при интерпретации (w, t), где $w \in \Sigma^{\omega}$, а t — позиция в сверхслове w. MFO-формулу, эквивалентную LTL-формуле φ, обозначим $\varphi(t)$.

Семантику LTL-формул зададим с помощью отображения h LTL-формул в эквивалентные им MFO-формулы. Для этого между пропозициональными переменными логики LTL и предикатными символами логики MFO устанавливается взаимно однозначное соответствие так, что переменной p_i соответствует предикатный символ P_i. Отображение h определим следующим образом:

$h(p_i) = P_i(t)$, где P_i принадлежит сигнатуре Ω логики MFO;

$h(\mathbf{X}\varphi) = \varphi(t+1)$, где $\varphi(t+1)$ получается из $\varphi(t)$ путем подстановки $t+1$ вместо t;

$h(\mathbf{F}\varphi) = \exists t_1 (t_1 \geq t)\varphi(t_1)$ (здесь и далее символ конъюнкции опускается);

$h(\mathbf{G}\varphi) = \forall t_1 (t_1 \geq t) \to \varphi(t_1)$;

$h(\varphi_1 \mathbf{U} \varphi_2) = \exists t_1 (t_1 \geq t)\varphi_2(t_1) \forall t_2 (t_1 > t_2 \geq t) \to \varphi_1(t_2)$.

Логические операции над LTL-формулами отображаются в соответствующие логические операции над эквивалентными им MFO-формулами. Так, например, MFO-формула $\varphi(t)$, эквивалентная LTL-формуле $\varphi = \mathbf{G}(p_1 \to \mathbf{XF}p_2)$, имеет вид $\forall t_1 (t_1 \geq t) \to (P_1(t_1) \to (\exists t_2 (t_2 \geq t_1) P_2(t_2 + 1)))$.

LTL-формула φ истинна на сверхслове w (обозначается $w \models \varphi$), если $(w, 0) \models h(\varphi)$. Заметим, что при интерпретации приведенной выше формулы φ на

структурах вида $(w, 0)$, где 0 — начальная позиция сверхслова w, эквивалентная ей MFO-формула приобретает вид $\forall t(P_1(t) \to (\exists t_1(t_1 \geq t)P_2(t_1+1)))$.

Формула φ логики LTL задает ω-язык $L(\varphi) = \{w \models \varphi\}$. Как следует из этого определения, ω-язык, определимый в логике LTL, определим также в логике MFO. Известно, что верно и обратное утверждение [4, 13].

ЛОГИКИ С ОГРАНИЧЕННЫМИ КВАНТОРАМИ

Модели дискретного времени, используемые в большинстве логик, изоморфны натуральным числам, т.е. определяют время, конечное в прошлом и бесконечное в будущем. Однако многие утверждения, возникающие в процессе построения спецификации системы, выражаются более просто и естественно при использовании значений переменных в моменты времени, предшествующие текущему. Осознание удобства рассмотрения поведения системы в прошлом привело к добавлению в темпоральные логики операторов прошедшего времени. При этом область интерпретации осталась бесконечной в одну сторону, что приводит к неоднозначным ситуациям, когда вычисление значения подформулы требует выхода за начальную границу времени [14]. Наиболее простое и естественное решение этой проблемы состоит в рассмотрении времени, бесконечного в обе стороны, т.е. изоморфного множеству целых чисел. Это также существенно упрощает преобразование формул спецификации в процессе синтеза в силу их инвариантности относительно сдвига во времени. В этом случае для описания семантики логических спецификаций используются циклические автоматные модели, функционирование которых не привязано к конкретному моменту времени. В дальнейшем в рассматриваемых фрагментах логики MFO предполагается ее интерпретация на множестве $\mathbf{Z}$ целых чисел.

В [9] рассматривалось очень простое подмножество L логики MFO, формулы которого имеют вид $\forall tF(t)$, где $F(t)$ — формула, не содержащая кванторов и символов числовых отношений. Класс автоматов, специфицируемых в этом языке, ограничен автоматами с конечной памятью. В [10] этот язык расширен конструкцией, содержащей ограниченные кванторы, что позволило специфицировать некоторые автоматы, не обладающие конечной памятью. В настоящей работе рассматриваются фрагменты логики MFO с ограниченными кванторами, обладающие бóльшими выразительными возможностями и практически не усложняющие алгоритм синтеза.

Определение 7. Формула $F(t)$, с единственной свободной переменной t, называется k-ограниченной справа ($k \in \mathbf{Z}$), если для любого $\tau \in \mathbf{Z}$ значения формулы $F(\tau)$ на всех двусторонних сверхсловах, имеющих одинаковые $(\tau + k)$-префиксы, совпадают.

Определение 8. Формула $F(t)$, с единственной свободной переменной t, называется k-ограниченной слева ($k \in \mathbf{Z}$), если для любого $\tau \in \mathbf{Z}$ значения формулы $F(\tau)$ на всех двусторонних сверхсловах, имеющих одинаковые $(\tau + k)$-суффиксы, совпадают.

Формула $F(t)$ ограничена с обеих сторон, если существуют такие $k_1, k_2 \in \mathbf{Z}$, что $k_1 < k_2$ и $F(t)$ k_1-ограничена слева и k_2-ограничена справа. Так, формула $\neg x(t-2)x(t-1)y(t)$ 0-ограничена справа и –3-ограничена слева. Формулы вида $\forall tF(t)$ назовем формулами прошедшего времени (P-формулами), если $F(t)$ ограничена справа, и формулами будущего времени (F-формулами), если $F(t)$ ограничена слева. Если $F(t)$ ограничена с двух сторон, то формулу $\forall tF(t)$ можно трактовать либо как P-формулу, либо как F-формулу.

Рассмотрим фрагмент логики MFO, определяемый следующим образом.

Все замкнутые формулы (предложения) имеют вид $\forall tF(t)$, где $F(t)$ — формула с одной свободной переменной. Таким образом, замкнутые формулы $\forall t_1F_1(t_1) \vee \forall t_2F_2(t_2)$ или $\neg \forall tF(t)$ не принадлежат рассматриваемому фрагменту. Каждая подформула формулы $F(t)$ имеет не более двух свободных переменных. Все кванторные подформулы с одной свободной (в этой подформуле) пере-

менной имеют вид $\exists t_1(t_1 \le t+k)F_1(t_1)$ или $\forall t_1(t_1 \le t+k) \to F_1(t_1)$, а кванторные подформулы с двумя свободными переменными — $\exists t_2(t_1 < t_2 \le t+k)F_2(t_2)$ или $\forall t_2(t_1 < t_2 \le t+k) \to F_2(t_2)$, где $k \in \mathbf{Z}$. Выражения $(t_1 \le t+k)$ и $(t_1 < t_2 \le t+k)$ в этих кванторных подформулах называются ограничениями области действия кванторов, а сами кванторы — ограниченными. Атомарные формулы вида $\tau_1 \rho \tau_2$, где τ_1 и τ_2 — термы, а $\rho \in \{<, >, \le, \ge\}$, встречаются только в ограничениях области действия кванторов. Каждая такая формула равносильна формуле вида $(t_1 \le t_2 + k)$, где $k \in \mathbf{Z}$. В остальном синтаксис этого фрагмента соответствует синтаксису логики MFO.

Язык спецификации, который назовем языком LP, состоит только из предложений рассмотренного фрагмента. Очевидно, что формулы с одной свободной переменной, удовлетворяющие приведенным выше требованиям, ограничены справа. Таким образом, язык LP составляют P-формулы.

Модели (интерпретации) для формулы $F = \forall tF(t)$ будем отождествлять с двусторонними сверхсловами в алфавите Σ, который определяется одноместными предикатными символами, имеющимися в формуле F. При этом, как отмечалось в предыдущем разделе, существуют две возможные интерпретации предикатных символов:

— каждый предикатный символ ассоциируется с символом алфавита Σ;

— предикатные символы соответствуют двоичным переменным, наборы значений которых кодируют символы алфавита Σ.

В первом случае для каждой пары различных предикатных символов P_i и P_j в спецификации явно или неявно присутствует формула $\forall t \neg(P_i(t) \& P_j(t))$.

Множество всех моделей для формулы F обозначим $M(F)$. Кроме того, для описания автоматной семантики формулы F понадобятся еще два множества:

$P(F)$ — множество префиксов всех моделей из $M(F)$,

$W(F)$ — множество суффиксов всех моделей из $M(F)$.

Для формулы $F = \forall tF(t)$ определим две автоматные семантики: детерминированную и недетерминированную. Детерминированная семантика однозначно ставит в соответствие формуле F детерминированный Σ-автомат. Недетерминированная семантика — соответственно недетерминированный Σ-автомат.

Рассмотрим сначала детерминированную семантику.

Каждому обратному сверхслову $g \in P(F)$ поставим в соответствие множество сверхслов $S_g = \{l \in W(F) \mid g \cdot l \in M(F)\}$, т.е. S_g состоит из всех тех сверхслов, конкатенация каждого из которых с префиксом g соответствует модели для F. Назовем такие сверхслова допустимыми продолжениями префикса g. Пусть $S(F) = \{S_g \mid g \in P(F)\}$. Можно показать, что для рассматриваемых классов формул такая совокупность множеств сверхслов конечна. Пусть $S(F) = \{S_1, S_2, \ldots, S_n\}$, тогда формула F специфицирует Σ-автомат $A(F)$ с поведением $(\overline{S}_1, \ldots, \overline{S}_n)$, где $\overline{S}_i$ — замыкание множества S_i.

Иногда удобнее пользоваться несколько иной формулировкой этой семантики. На множестве префиксов $P(F)$ определим отношение эквивалентности так, что два префикса, g_1 и g_2, эквивалентны, если они имеют одно и то же множество допустимых продолжений. Эта эквивалентность разбивает множество $P(F)$ на классы эквивалентности $P_1, P_2, \ldots, P_n$, так что классу P_i соответствует состояние q_i специфицируемого Σ-автомата. Определение функции переходов имеет вид $\delta(q_i, \sigma) = q_j$ тогда и только тогда, когда $P_i\sigma \subseteq P_j$. Из этого определения сразу следует, что функция переходов детерминирована. Действительно, поскольку множества $P_1, P_2, \ldots, P_n$ попарно не пересекаются, приведенному соотношению может удовлетворять только единственное множество P_j.

Рассмотрим пример построения по P-формуле автомата, определяемого детерминированной семантикой.

Пример 1. Пусть $F = \forall tF(t)$, где $F(t) = \exists t_1(t_1 \le t+2)b(t_1)\forall t_2(t_1 < t_2 \le t) \to a(t_2)$ и $\Sigma = \{a, b, c\}$. Можно показать, что множество моделей для формулы F равно $(\Sigma ba^* \cup \Sigma\Sigma ba^* \cup ba^*)^{\mathbf{Z}} \cup (\Sigma ba^* \cup \Sigma\Sigma ba^* \cup ba^*)^{-\omega} a^{\omega}$.

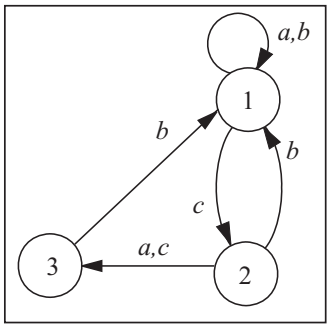

Рис. 1

Приведем разбиение множества префиксов $P(F)$ на классы эквивалентности: P_1 — префиксы заканчиваются словами из $\{aa, ba, b\}$, P_2 — префиксы заканчиваются словами из $\{ac, bc\}$, P_3 — префиксы заканчиваются словами ca, cc.

Очевидно, что эти условия задают разбиение $\{P_1, P_2, P_3\}$ множества $P(F)$. Построим теперь соотношения между классами разбиения, определяющие функцию переходов автомата $A(F)$. Поскольку конкретные множества префиксов из $P(F)$ не рассматриваются, соотношения должны быть такими, что для каждого $\sigma \in \Sigma$ соотношение $P_i\sigma \subseteq P_j$ выполняется тогда и только тогда, когда из того, что $g \in P_i$ есть префикс некоторой модели для F, следует, что $g\sigma$ также префикс модели для F. Это требование обеспечивается путем учета ограничений на допустимые фрагменты двусторонних сверхслов, определяемые множеством моделей $M(F)$. Так, например, $P_3(a \vee c) = \varnothing$, поскольку слова caa, cca, cac и ccc не могут быть фрагментами никакой модели для F, а $P_3 b \subseteq P_1$, так как префиксы из $P_3 b$ имеют окончания cab, ccb и, следовательно, принадлежат только P_1. В результате необходимые соотношения имеют вид $P_1 a \subseteq P_1$, $P_1 b \subseteq P_1$, $P_1 c \subseteq P_2$, $P_2 a \subseteq P_3$, $P_2 b \subseteq P_1$, $P_2 c \subseteq P_3$, $P_3 b \subseteq P_1$. Эквивалентность префиксов, принадлежащих одному и тому же классу разбиения, следует из того, что приведенные соотношения определяют одно и то же множество допустимых продолжений для всех префиксов из каждого класса. Таким образом, имеем автомат с тремя состояниями (рис. 1).

Недетерминированную семантику сформулируем симметрично второму определению детерминированной семантики.

Каждый суффикс $l \in W(F)$ определяет множество $P(l)$ допустимых для него префиксов, т.е. $P(l) = \{g \in P(F) \mid g \cdot l \in M(F)\}$. На множестве суффиксов $W(F)$ определим отношение эквивалентности так, что два суффикса, l_1 и l_2, эквивалентны, если они имеют одно и то же множество допустимых для них префиксов. Эта эквивалентность разбивает множество $W(F)$ на классы эквивалентности $S_1, S_2, \ldots, S_n$, каждому из которых соответствует состояние специфицируемого Σ-автомата. Функция переходов определяется следующим образом: для $\sigma \in \Sigma$ $\delta'(q_i, \sigma) = \{q_{i_j}\}$, где $\{q_{i_j}\}$ — множество всех таких q_{i_j}, для которых выполняется соотношение $\sigma S_{i_j} \subseteq S_j$. Как и раньше, формула F специфицирует Σ-автомат $A'(F)$ с поведением $(\overline{S}_1, \ldots, \overline{S}_n)$, где $\overline{S}_i$ — замыкание множества S_i. Как следует из приведенных выше семантик, детерминированный и недетерминированный автоматы, специфицируемые формулой F, слабо эквивалентны, т.е. имеют одно и то же множество допустимых сверхслов, равное $\overline{W(F)}$.

Покажем, что автомат $A(F)$, специфицируемый формулой F в соответствии с детерминированной семантикой, есть подавтомат циклической детерминизации автомата $A'(F)$, определяемого недетерминированной семантикой.

Сначала уточним понятие циклической детерминизации недетерминированного Σ-автомата A'. Это максимальный циклический подавтомат автомата A, состояниями которого являются все подмножества множества состояний автомата A', а функция переходов определяется следующим образом. Пусть состояние q автомата A имеет вид $\{q'_1, \ldots, q'_k\}$, тогда для $\sigma \in \Sigma$ $\delta(q, \sigma) = \bigcup_{i=1}^{k} \delta'(q'_i, \sigma)$, где δ и δ' — функции переходов соответственно автоматов A и A'.

Пусть $g \in P(F)$, S_g — множество допустимых продолжений для префикса g и $l \in S_g$. Обозначим $[l]$ класс эквивалентности суффикса l, которому соответствует состояние q' автомата $A'(F)$. Любой суффикс из $[l]$ есть допустимое продолжение для g, т.е. $[l] \subseteq S_g$. Если имеется несколько классов эквивалентности суффиксов, таких что g — допустимый префикс для принадлежащих им сверхслов,

то объединение всех таких классов составляет множество S_g. Пусть $S_1, \dots, S_n$ — классы эквивалентности суффиксов $l_1, \dots, l_n \in S_g$ и, следовательно, $S_1, \dots, S_n \subseteq S_g$. Напомним, что S_g — это поведение автомата $A(F)$ в каком-то его состоянии. Отсюда следует, что поведение детерминированного автомата $A(F)$ в произвольном состоянии есть объединение поведений недетерминированного автомата $A'(F)$ в некоторых его состояниях. Таким образом, состояния автомата $A(F)$ можно отождествить с множествами состояний автомата $A'(F)$.

Пусть $g = g_1\sigma$. Покажем, что для каждого класса $S_i \subseteq S_g$ выполняется $\sigma S_i \subseteq S'_i$, где S'_i — класс эквивалентности суффикса σl_i, принадлежащего S_{g_1}. Обозначим $P(l_i)$ и $P(\sigma l_i)$ множества всех префиксов из $P(F)$, имеющих допустимые продолжения соответственно l_i и σl_i. Каждый префикс из $P(\sigma l_i)\sigma$ имеет допустимое продолжение l_i. Отсюда следует, что $P(\sigma l_i)\sigma \subseteq P(l_i)$. По определению $[l_i]$ — множество допустимых продолжений для всех префиксов из $P(l_i)$, а $P(\sigma l_i)\sigma \subseteq P(l_i)$, поэтому $[l_i]$ — допустимые продолжения для $P(\sigma l_i)\sigma$. Таким образом, $\sigma[l_i]$ — допустимые продолжения для $P(\sigma l_i)$ и, следовательно, $\sigma[l_i] \subseteq [\sigma l_i]$.

Пусть в автомате $A(F)$ префиксам g_1 и g соответствуют состояния q_1 и q, а значит, $\delta(q_1, \sigma) = q$. Пусть состоянию q_1 соответствует множество классов эквивалентности суффиксов $S'_1, \dots, S'_m \subseteq S_{g_1}$. Переход из состояния q_1 под действием σ определяется теми S'_j, для которых имеется $S_i \subseteq S_g$, такое что $\sigma S_i \subseteq S'_j$. Таким образом, q составляют те состояния автомата $A'(F)$, в которые из состояний, составляющих q_1, осуществляется переход по сигналу σ, что соответствует определению функции переходов детерминизации автомата $A'(F)$.

Выше было показано, что каждое множество S_g, $g \in P(F)$, — объединение некоторых классов эквивалентности суффиксов из $W(F)$. Однако не для каждого множества этих классов существует S_g, совпадающее с их объединением. Поэтому из приведенного рассуждения следует, что автомат $A(F)$ — подавтомат детерминизации автомата $A'(F)$.

Пример 2. Построим для формулы F из примера 1 автомат, соответствующий недетерминированной семантике. Напомним, что множество $M(F)$ для этой формулы равно $(\Sigma ba^* \cup \Sigma\Sigma ba^* \cup ba^*)^{\mathbf{Z}} \cup (\Sigma ba^* \cup \Sigma\Sigma ba^* \cup ba^*)^{-\omega} a^{\omega}$. Определим разбиение множества суффиксов $W(F)$ на классы эквивалентности в соответствии с их начальными отрезками. Это разбиение состоит из трех классов: S_1, S_2, S_3. Суффиксы из S_1 начинаются словами из $a\bar{b}$ или $c\bar{b}b$, суффиксы из S_2 — словами из $\bar{b}b$, где $\bar{b} = (a \vee c)$, а суффиксы из S_3 начинаются символом b. Разбиение осуществлялось, исходя из таких же требований, как и в примере 1, гарантирующих, что классы разбиения состоят из эквивалентных суффиксов. Более того, это классы эквивалентности, поскольку никакие два класса не имеют одного и того же множества допустимых префиксов. Так, имеется префикс, заканчивающийся символом c, допустимый для S_2 и S_3 и недопустимый для S_1. Префикс, заканчивающийся символами cc, допустим для S_3 и недопустим для S_2. Таким образом, автомат $A'(F)$ имеет три состояния. Множество соотношений между классами разбиения, определяющее функцию переходов автомата $A'(F)$, имеет следующий вид:

$$aS_1 \subseteq S_1,\ bS_1 \subseteq S_3,\ aS_2 \subseteq S_1,\ bS_2 \subseteq S_3,$$
$$cS_2 \subseteq S_1,$$
$$aS_3 \subseteq S_2,\ bS_3 \subseteq S_3,\ cS_3 \subseteq S_2.$$

Соответствующий автомат $A'(F)$ представлен на рис. 2. Несложно убедиться, что циклическая детерминизация этого автомата совпадает с автоматом из примера 1, где состоянию 1

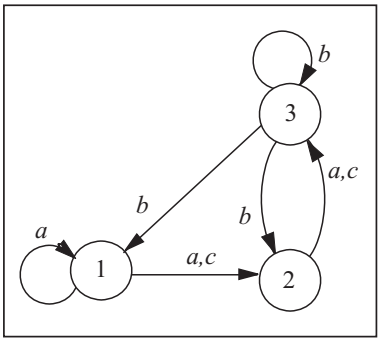

Рис. 2

автомата $A(F)$ соответствует множество всех состояний автомата $A'(F)$, состоянию 2 — множество $\{2, 3\}$, а состоянию 3 — множество $\{3\}$.

Выше рассматривался фрагмент LP логики MFO, множество предложений которого составляют P-формулы. Рассмотрим теперь в некотором смысле симметричный фрагмент LF для представления F-формул. Синтаксис языка LF совпадает с синтаксисом языка LP во всем, кроме вида ограничений области действия кванторов. Так, все кванторные подформулы с одной свободной переменной имеют вид $\exists t_1(t_1 \geq t + k)F_1(t_1)$ или $\forall t_1(t_1 \geq t + k) \to F_1(t_1)$, а кванторные подформулы с двумя свободными переменными — $\exists t_2(t_1 > t_2 \geq t + k)F_2(t_2)$ или $\forall t_2(t_1 > t_2 \geq t + k) \to F_2(t_2)$, где $k \in \mathbf{Z}$. Обе рассмотренные автоматные семантики для языка LP распространяются и на язык LF.

Заметим, что для любой LTL-формулы $\mathbf{G}\varphi$ имеется эквивалентная ей LF-формула $F = \forall th(\varphi)$, где h — определенное ранее отображение LTL-формул в MFO-формулы с одной свободной переменной. Здесь под эквивалентными формулами понимаются формулы, задающие одно и то же множество сверхслов, причем множество сверхслов, задаваемое формулой F, интерпретируемой на множестве $\mathbf{Z}$, равно $W(F)$. Действительно, формула $\mathbf{G}\varphi$ задает множество всех сверхслов, в каждой позиции которых истинна формула φ. Легко видеть, что формула $F = \forall th(\varphi)$ принадлежит языку LF, и $W(F)$ — множество всех сверхслов, обладающих этим свойством. Индукцией по структуре LF-формулы можно показать, что обратное утверждение также верно.

Рассмотрим примеры применения детерминированной и недетерминированной семантик для определения Σ-автоматов, специфицируемых LF-формулами.

Пример 3. Пусть $F = \forall t(a(t) \to \exists t_1(t_1 > t)b(t_1))$, $\Sigma = \{a, b, c\}$. Построим автомат $A(F)$, специфицируемый формулой F согласно детерминированной семантике.

Формула F определяет множество $\mathbf{Z}$-слов, в которых после каждого символа a встречается символ b. Формально это выражается в виде $M(F) = (b \vee c \vee a(a \vee c)^* b)^{\mathbf{Z}}$, т.е. модели построены из отрезков двух типов: не содержащих символа a и отрезков, начинающихся символом a и заканчивающихся символом b.

Разбиение множества префиксов $P(F)$ на классы эквивалентности P_1 и P_2 соответствует двум типам префиксов: 1) префиксы, заканчивающиеся символом b, после которого имеется, возможно пустая, последовательность символов c, или префиксы, не содержащие символа a; 2) префиксы, заканчивающиеся символом a, после которого идет последовательность, не содержащая символа b. Таким образом,

$$P_1 = (b \vee c \vee a(a \vee c)^* b)^{-\omega},\ P_2 = (b \vee c \vee a(a \vee c)^* b)^{-\omega} a(a \vee c)^*,\ P_1 \cap P_2 = \varnothing.$$

Соответствующие этим классам множества допустимых продолжений префиксов имеют вид

$$S_1 = (b \vee c \vee a(a \vee c)^* b)^{\omega},\ S_2 = (a \vee c)^* b(b \vee c \vee a(a \vee c)^* b)^{\omega}.$$

Несложно убедиться в справедливости следующих соотношений между классами P_1 и P_2:

$$P_1(b \vee c) \subseteq P_1, P_1 a \subseteq P_2,\ P_2(a \vee c) \subseteq P_2,\ P_2 b \subseteq P_1.$$

Автомат, определяемый этими соотношениями, изображен на рис. 3.

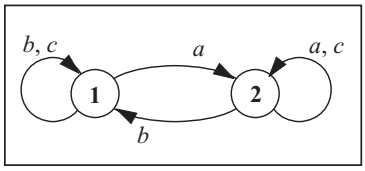

Рис. 3

Множества $W(1)$ и $W(2)$ равны замыканиям соответственно S_1 и S_2.

Заметим, что $S_1 \neq S_2$, хотя их замыкания равны $\overline{S_1} = \overline{S_2} = \Sigma^{\omega}$. В силу этого автомат $A(F)$ не приведенный. Если его рассматривать как автомат Бюхи с финальным состоянием 1, то он будет распознавать ω-язык, задаваемый формулой F.

Пример 4. Построим автомат $A'(F)$, специфицируемый формулой F из предыдущего примера, в соответствии с недетерминированной семантикой.

Покажем, что разбиение $W(F)$ на классы эквивалентности имеет вид

$$S_1 = (b \vee c \vee a(a \vee c)^* b)^* b(b \vee c \vee a(a \vee c)^* b)^\omega, \; S_2 = c^\omega.$$

Множества S_1 и S_2 не пересекаются, так как все сверхслова первого из них содержат символ b. Множество допустимых для S_1 префиксов совпадает с $P(F)$, поскольку все сверхслова из S_1 имеют суффикс, начинающийся символом b, после которого каждый символ a предшествует некоторому символу b. Для S_2, состоящего из одного сверхслова, покажем, что никакое другое сверхслово не имеет такого множества допустимых префиксов, как сверхслово c^ω. Если сверхслово содержит символ b, то для него допустим префикс вида la, где $l \in \Sigma^{-\omega}$, недопустимый для c^ω. Если сверхслово содержит символ a, то оно содержит и символ b, а следовательно, имеет допустимый префикс, недопустимый для c^ω. Остается показать, что $S_1 \cup S_2 = W(F)$. Дополнение множества $\Sigma^* b\Sigma^\omega$ равно $(a \vee c)^\omega$, однако никакой суффикс из $W(F)$ не может содержать a и не содержать b, поэтому дополнение S_1 до $W(F)$ равно c^ω. Легко видеть, что выполняются следующие соотношения: $\Sigma S_1 \subseteq S_1$, $bS_2 \subseteq S_1$, $cS_2 \subseteq S_2$, $aS_2 \not\subset W(F)$, что соответствует недетерминированному автомату $A'(F)$, приведенному на рис. 4.

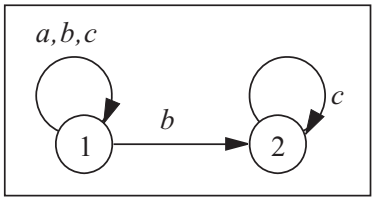

Рис. 4

Детерминизация этого автомата дает автомат, полученный в примере 3.

СИММЕТРИЧНЫЕ ФОРМУЛЫ

Формула языка LP и формула языка LF называются симметричными, если одна из них получается из другой путем замены операций в термах обратными (+ на – и наоборот) и символов числовых отношений симметричными ($<$ на $>$, $\le$ на $\ge$ и наоборот). Формулу, симметричную F, обозначим $\widetilde{F}$. Например, для LP-формулы

$$F = \forall t(a(t+2) \to \exists t_1(t_1 \le t) b(t_1 - 1) \; \forall t_2((t_2 < t_1) \to c(t_2 + 2))$$

симметричная ей LF-формула $\widetilde{F}$ равна

$$\forall t(a(t-2) \to \exists t_1(t_1 \ge t) b(t_1 + 1) \forall t_2((t_2 > t_1) \to c(t_2 - 2)).$$

Для определения соотношений между автоматами, специфицируемыми симметричными формулами, понадобятся дополнительные понятия, касающиеся бесконечных слов.

Пусть **Z**-слово $u = \ldots \sigma_{-2}\sigma_{-1}\sigma_0\sigma_1\sigma_2 \ldots$, где $\sigma_i = u(i)$. Инверсия **Z**-слова u — это такое **Z**-слово u^-, что $u^-(i) = u(-i)$. Аналогично инверсией сверхслова $\sigma_0\sigma_1\sigma_2 \ldots$ является обратное сверхслово $\ldots \sigma_{-2}\sigma_{-1}\sigma_0$, где $\sigma_{-i} = \sigma_i$. Для $W \subseteq \Sigma^\omega$ $(\Sigma^{-\omega})$ обозначим $W^- = \{w^- \mid w \in W\}$.

Утверждение 4. Каждая модель m для формулы F есть инверсия модели для симметричной ей формулы $\widetilde{F}$.

Индукцией по структуре формулы $F(t)$ легко показать, что из $(w, t) \models F(t)$ следует $(w^-, -t) \models \widetilde{F}(t)$, из чего вытекает справедливость утверждения 4.

Рассмотрим теперь, как связаны между собой автоматы, специфицируемые симметричными формулами.

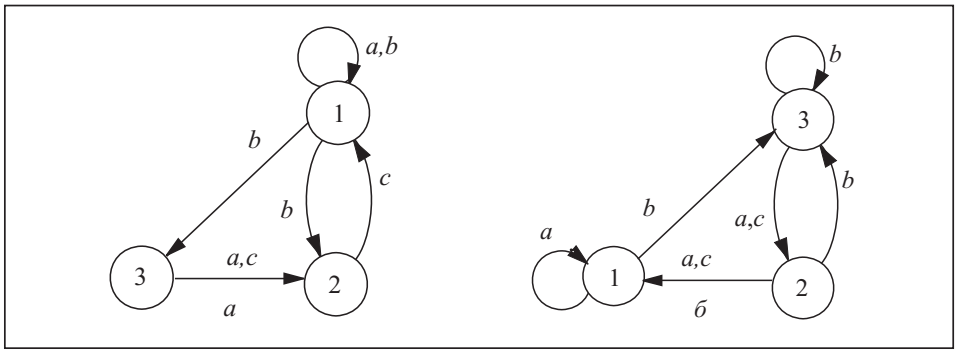

Рис. 5

Каждый префикс модели m для формулы F является инверсией некоторого суффикса модели m^- для симметричной формулы и наоборот. Таким образом, имеют место соотношения $P(F) = W(\widetilde{F})^-$ и $W(F) = P(\widetilde{F})^-$. Разбиению множества $P(F)$ на классы эквивалентности соответствует разбиение $W(\widetilde{F})$ на инверсные им классы эквивалентности. Отсюда следует, что автомат $A'(\widetilde{F})$ обратный автомату $A(F)$, а автомат $A'(F)$ обратный автомату $A(\widetilde{F})$. Таким образом, имея для симметричных формул один из специфицируемых ими автоматов, часто можно построить остальные три автомата, используя операции обращения и детерминизации автоматов. Проиллюстрируем это для симметричных формул

$$F = \forall t \exists t_1 (t_1 \le t+2) b(t_1) \forall t_2 (t_1 < t_2 \le t) \to a(t_2),$$

$$\widetilde{F} = \forall t \exists t_1 (t_1 \ge t-2) b(t_1) \forall t_2 (t_1 > t_2 \ge t) \to a(t_2)$$

из примера 1. Пусть имеем автомат $A(F)$, изображенный на рис. 1. Построим последовательно автоматы $A'(\widetilde{F})$, $A(\widetilde{F})$ и $A'(F)$.

На рис. 5, *а* приведен автомат, обратный автомату $A(F)$ и, следовательно, совпадающий с $A'(\widetilde{F})$, а на рис. 5, *б* — его циклическая детерминизация, совпадающая с $A(\widetilde{F})$. Автомат, обратный этому автомату, совпадает с $A'(F)$, что подтверждается примером 2. Из этого рассмотрения видно, что детерминированные автоматы, специфицируемые симметричными формулами, в общем случае различаются, хотя симметричные формулы, не содержащие кванторов кроме внешнего, автоматно эквивалентны, т.е. специфицируют один и тот же автомат. Более того, автомат $A(\widetilde{F})$ на рис. 5, *б,* специфицируемый LF-формулой, можно специфицировать и LP-формулой, т.е. для LF-формулы $\widetilde{F}$ имеется автоматно эквивалентная ей LP-формула.

ЗАКЛЮЧЕНИЕ

При решении задач синтеза автоматных моделей большое значение имеет выбор языка спецификации, позволяющий специфицировать достаточно широкий класс автоматов определенного вида и обеспечивающий приемлемую сложность процедуры синтеза. Очевидно, что чем проще синтаксис языка спецификации, тем меньше сложность процедуры синтеза. Исходя из этих соображений в [9] в качестве языка спецификации использовалось подмножество L логики MFO, формулы которого имеют вид $\forall t F(t)$, где $F(t)$ — формула, не содержащая кванторов и символов числовых отношений. Класс автоматов, специфицируемых в этом языке, составляют автоматы с конечной памятью. В [10] этот язык расширен конструкцией, содержащей ограниченные кванторы, что позволило специфицировать некоторые автоматы, не обладающие конечной памятью. В качестве области интерпретации обоих этих языков высту-

пает множество целых чисел. Это упрощает как процесс написания спецификации, делая его более естественным, так и преобразование формул спецификации в процессе синтеза в силу их инвариантности относительно сдвига во времени. В настоящей работе рассмотрены два фрагмента, LP и LF, логики MFO с ограниченными кванторами, также интерпретируемые на целых числах. В логике LP текущее состояние системы однозначно определяется ее предшествующим поведением, а в логике LF — будущим поведением. Язык LP расширяет выразительные возможности языка из [10], практически не усложняя алгоритм синтеза. Обычно под выразительными возможностями логического языка понимают класс свойств, выразимых в этом языке. Здесь под выразительными возможностями языка понимается класс специфицируемых в нем автоматов. Язык LF дает альтернативные средства для спецификации автоматов, аналогичные формулам логики LTL вида $\mathbf{G}\varphi$. Это удобно в тех случаях, когда требования, определяемые спецификацией, относятся к будущему, например, «после каждого запроса должна следовать реакция на него». То, что все предложения обоих языков начинаются квантором $\forall t$, обусловлено тем, что формулы спецификации должны описывать требования к поведению автомата в любой момент времени, т.е. в каждом его состоянии. Следует отметить, что сверхсловарная семантика языков не соответствует их автоматной семантике, поэтому формулы, неэквивалентные в соответствии со сверхсловарной семантикой, могут специфицировать один и тот же автомат. В этом смысле выразительные возможености языков используются не в полной мере.

В статье определены взаимозависимости между автоматами, специфицируемыми симметричными формулами. Полученные результаты позволяют использовать один и тот же алгоритм (с незначительной модификацией) для синтеза автоматов, специфицируемых как LP-, так и LF-формулами. При этом может потребоваться использование алгоритма детерминизации автомата, дающего экспоненциальное (относительно количества состояний) увеличение сложности процедуры синтеза.

СПИСОК ЛИТЕРАТУРЫ

1. Thomas W. Automata on infinite objects. Handbook of Theoretical Comput. Sci. J. van Leeuwen (ed.). Amsterdam: Elsevier Science Publishers, 1990. P. 134–191.
2. Stomp F.A., de Rover W.P., Gerth R.T. The μ-calculus as an assertion language for fairness arguments. *Information and Computation*. 1989. Vol. 82. P. 278–322.
3. Emerson E.A. Temporal and modal logic. Handbook of Theoretical Comput. Sci. J. van Leeuwen (ed.). Amsterdam: Elsevier Science Publishers, 1990. P. 996–1071.
4. Schneider K. Verification of reactive systems. Berlin; Heidelberg: Springer, 2004. 606 p.
5. Abadi M., Lamport L., Wolper P. Realizable and unrealizable specifications of reactive systems. *Intern. Colloq. on Automata, Languages, and Programming. Lecture Notes in Computer Science*. 1989. Vol. 372. P. 1–17.
6. Чеботарев А.Н. Согласование спецификаций автоматов, представленных в языке L. *Кибернетика и системный анализ*. 2016. Т. 52, № 3. С. 3–15.
7. Alpern B., Schneider F.B. Defining liveness. *Information Processing Letters*. Vol. 21. 1985. P. 181–185.
8. Чеботарев А.Н. Согласование взаимодействующих автоматов. *Кибернетика и системный анализ*. 2015. Т. 51, № 5. С. 13–25.
9. Чеботарев А.Н. Регулярная форма спецификации детерминированных автоматов в языке L. *Приклад. дискрет. математика*. 2010. № 4. С. 64–72.
10. Чеботарев А.Н. Расширение логического языка спецификации и проблема синтеза. *Кибернетика и системный анализ*. 1996. № 6. С. 11–27.

11. Finkel O. Topological properties of omega context free languages. *Theoretical Computer Science*. 2001. Vol. 262 (1–2). P. 669–697.
12. Diekert V., Gastin P. First-order definable languages. Logic and Automata: History and Perspectives. J. Flum, E. Gradel, T. Wilke (eds.). Amsterdam: Univ. Press, 2007. P. 261–306.
13. Gabbay D., Hodkinson I., Reynolds M. Temporal logic. Oxford: Clarendon Press, 1994. Vol. 1. 668 p.
14. Coen-Porisini A., Pradella M., San Pietro M. A finite-domain semantics for testing temporal logic specifications. *Lecture Notes in Computer Science*. 1998. Vol. 1486. P. 41–54.

Надійшла до редакції 17.02.2017

А.М. Чеботарьов

ДЕЯКІ ПІДМНОЖИНИ МОНАДИЧНОЇ ЛОГІКИ ПЕРШОГО ПОРЯДКУ (MFO), ЩО ВИКОРИСТОВУЮТЬСЯ ДЛЯ СПЕЦИФІКАЦІЇ І СИНТЕЗУ Σ-АВТОМАТІВ

Анотація. Розглянуто два фрагменти, LP і LF, логіки першого порядку з обмеженими кванторами, які використовуються для специфікації транс-д'юсерів. Логіка LP дозволяє характеризувати поточну поведінку системи на основі її минулої поведінки, а LF — на основі майбутньої поведінки. Визначено два види семантик для цих логік та досліджено властивості автоматів, що в них специфікуються.

Ключові слова: логіки першого порядку, формули минулого часу, формули майбутнього часу, автоматна семантика, симетричні формули.

A.N. Chebotarev

SOME SUBSETS OF MONADIC FIRST ORDER LOGIC (MFO) USED FOR SPECIFICATION AND SYNTHESIS OF Σ-AUTOMATA

Abstract. Two fragments, LP and LF, of first-order logic with bounded quantifiers used for specification of transducers are considered. Fragment LP enables the current behaviour of a system to be characterized on the basis of its past behaviour, and LF relies on the future behaviour. Two kinds of semantics are defined for logics LP and LF, and the properties of automata specified in these logics are investigated.

Keywords: first order logics, past formulas, future formulas, automatic semantics, symmetric formulas.

Чеботарев Анатолий Николаевич,
доктор техн. наук, ведущий научный сотрудник Института кибернетики НАН Украины им. В.М. Глушкова, Киев, e-mail: ancheb@gmail.com.