КИБЕРНЕТИКА

А.Н. ЧЕБОТАРЕВ

УДК 519.713.1

ИСПОЛЬЗОВАНИЕ СОГЛАСОВАНИЯ ЛОГИЧЕСКИХ СПЕЦИФИКАЦИЙ АВТОМАТОВ ПРИ РЕШЕНИИ ИГРОВЫХ ЗАДАЧ

Аннотация. Игровые модели широко используются при решении задач реализуемости, синтеза и верификации реактивных систем. Для решения таких задач в статье рассматривается автоматный подход, основанный на понятии согласованности автоматов или их логических спецификаций. Соответствующие методы существенно отличаются от тех, которые используются в игровом контексте. Показано, как эти методы можно применять для синтеза выигрышной стратегии в игре двух лиц, не связанной с проектированием реактивных систем.

Ключевые слова: язык спецификации L, согласование автоматов, синтез автомата, модель игры, выигрышная стратегия.

ВВЕДЕНИЕ

Игровые модели широко используются при решении задач реализуемости, синтеза и верификации открытых систем, т.е. систем, взаимодействующих со своим окружением. В этих задачах спецификация требований к поведению системы, взаимодействующей со средой, выражается в каком-либо логическом языке, таком как различного рода темпоральные логики, логики первого порядка или ограниченные логики второго порядка. Важной особенностью открытой системы является то, что в процессе ее проектирования никакие ограничения на поведение среды, кроме определяемых ее спецификацией, не должны налагаться. Спецификация открытой системы предполагается такой, что все требования к поведению системы при ее взаимодействии со средой выполняются при любом допустимом поведении среды.

Использование для синтеза открытых (реактивных) систем моделей, описывающих игру между системой и средой, с которой она взаимодействует, было предложено в работах Пнуэли и Рознера [1], Абади, Лэмпорта и Волпера [2], Дилла [3]. Как правило, такие модели определяются двудольным графом и условием выигрыша. Вершины графа (состояния) разбиты на два множества, каждое из которых соответствует одному из игроков, а условие выигрыша в общем случае задает некоторое множество бесконечных последовательностей состояний, порождаемых поочередными ходами игроков. Игры на конечных графах рассматривались в таких задачах проектирования реактивных систем, как синтез контроллера [4], верификация протоколов [5], композиционное проектирование, верификация [6] и др. Построение корректного алгоритма функционирования системы заключается в нахождении выигрышной стратегии в игре, моделирующей эту систему. Такую стратегию можно получить только за счет ограничения выбора очередного хода системы. Выбор очередного хода среды может быть любым в рамках ее возможностей. Спецификации, для которых выигрышные стратегии существуют, называются реализуемыми. Сложность построения такой стратегии зависит от способа задания условия выигрыша и, следовательно, от характера свойств системы, задаваемых этим условием. Так, при задании условия LTL формулой задача построения стратегии полна в классе 2EXPTIME [1]. Поэтому для практического применения игровых моделей ограничиваются более узкими классами свойств. В настоящей работе для спецификации игровых моделей используется логический язык L [7], формулы которого задают свойства безопасности (safety). Формально свойства рассматриваются как подмножества множества Σ^{ω} всех бесконечных слов (сверхслов) в некотором алфавите Σ. Множество $S \subseteq \Sigma^{\omega}$ определяет свойство безопасности тогда и только тогда, когда каждое сверхслово, не принадлежащее S, имеет префикс, любое продолжение которого не принадлежит S. Для спецификации, задающей только свойства безопасности, проверка ее реализуемости сводится к такому преобразованию, в результате которого (если это возможно) получается спецификация, не ограничивающая поведение среды. В [8, 9] эта задача формализуется в терминах согласования спецификации системы со спецификацией среды. Методы ее решения, предложенные в [8, 9] как на уровне спецификаций, так и на уровне соответствующих автоматов, существенно отличаются от современных методов решения игровых задач. В настоящей статье показано, что методы проектирования реактивных алгоритмов, основанные на согласовании спецификаций, могут использоваться для решения игровых задач, не связанных с таким проектированием. На простом, но достаточно иллюстративном, примере игры НИМ показывается специфика использования этих методов для построения выигрышной стратегии в игре двух лиц с нулевой суммой.

СИНТАКСИС И СЕМАНТИКА ЯЗЫКА L

Язык L [7] представляет собой фрагмент логики предикатов первого порядка с одноместными предикатами и фиксированной областью интерпретации, в качестве которой выступает множество **Z** целых чисел (моментов времени). Спецификация в языке L имеет вид формулы $\forall t F(t)$. Здесь $F(t)$ — формула, построенная с помощью логических связок из атомарных формул (атомов) вида $p(t+k)$, где p — одноместный предикатный символ, t — переменная, принимающая значения из множества целых чисел, k — целое число, называемое рангом атома. Разность между максимальным и минимальным значениями рангов атомов, встречающихся в формуле, называется глубиной формулы.

Пусть $\Omega = \{p_1, p_2, \dots, p_m\}$ — множество всех предикатных символов формулы $F(t)$. При определении семантики языка L он рассматривается как формализм для задания множества сверхслов (бесконечных слов) в алфавите двоичных векторов, длина которых равна мощности множества Ω. Обычно символы такого алфавита рассматриваются как отображения вида $\sigma\colon \Omega \to \{0,1\}$. Так как этот алфавит однозначно определяется множеством предикатных символов Ω, обозначим его $\Sigma(\Omega)$.

Пусть Σ — конечный алфавит, **Z** — множество целых чисел, $\mathbf{N}^+ = z \in \mathbf{Z} | z > 0$. Отображения $u\colon \mathbf{Z} \to \Sigma$ и $l\colon \mathbf{N}^+ \to \Sigma$ называются соответственно двусторонним сверхсловом (обозначается $\dots u(-2)\, u(-1)\, u(0)\, u(1)\, u(2)\dots$) и сверхсловом (обозначается $l(1)\, l(2)\dots$) в алфавите Σ. Бесконечный отрезок $u(k+1, \infty)$ двустороннего сверхслова u назовем k-суффиксом этого сверхслова.

Опишем семантику языка L. Формулы языка L интерпретируются на множестве целых чисел. В силу этого, как показано в [7], можно ограничиться рассмотрением формул, у которых максимальный ранг атомов равен 0. Каждой формуле $F = \forall t F(t)$ ставится в соответствие множество моделей для этой формулы, т.е. множество таких интерпретаций, на которых F истинна. Пусть $\Omega = \{p_1, p_2, \ldots, p_m\}$ — множество всех предикатных символов, имеющихся в формуле F (сигнатура формулы). Интерпретация формулы F — это упорядоченный набор определенных на **Z** одноместных предикатов $\pi_1, \ldots, \pi_m$, соответствующих предикатным символам из Ω. Интерпретацию $I = \langle \pi_1, \ldots, \pi_m \rangle$ 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. Если не различать интерпретации и соответствующие им двусторонние сверхслова, то можно говорить об истинности формулы F на двустороннем сверхслове u и значении формулы $F(t)$ в некоторой точке τ двустороннего сверхслова u. Смысл понятия глубины формулы состоит в том, что истинностное значение формулы $F(t)$ глубины r в точке τ интерпретации u определяется отрезком $u(\tau - r, \tau)$ соответствующего двустороннего сверхслова u.

Будем считать, что формула $F = \forall t F(t)$ задает множество 0-суффиксов всех двусторонних сверхслов из $M(F)$. Обозначим это множество сверхслов $W(F)$.

Определим автоматную семантику языка L.

Определение 1. Конечный $X-Y$-автомат Мура A — это пятерка $\langle X, Y, Q, \chi_A, \mu_A \rangle$, где X и Y — соответственно входной и выходной алфавиты, Q — конечное множество состояний, а $\chi_A : Q \times X \to 2^Q$ и $\mu_A : Q \to Y$ — функции соответственно переходов и выходов. Автомат A называется детерминированным, если для любых $x \in X$, $q \in Q$ $|\chi_A(q, x)| \leq 1$, в противном случае он называется недетерминированным.

Определение 2. $X-Y$-автомат $A = \langle X, Y, Q, \chi_A, \mu_A \rangle$ называется квазидетерминированным, если для любых $x \in X, q \in Q$ из $q_1, q_2 \in \chi_A(q, x)$ следует $\mu(q_1) \neq \mu(q_2)$.

Квазидетерминированный $X-Y$-автомат удобно рассматривать как детерминированный частичный автомат без выходов с входным алфавитом $\Sigma = X \times Y$. Такой автомат $A = \langle \Sigma, Q, \delta_A \rangle$, где $\delta_A : Q \times \Sigma \to Q$ — частичная функция переходов, назовем Σ-автоматом.

Определение 3. Σ-автомат $A = \langle \Sigma, Q, \delta_A \rangle$ 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 $q = \delta_A(q_2, \sigma_2)$.

Циклический Σ-автомат можно охарактеризовать в терминах допустимых сверхслов.

Определение 4. Сверхслово $l = \sigma_1 \sigma_2 \ldots$ 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 автомата A, если оно допустимо хотя бы в одном из его состояний.

Обозначим $W(A)$ множество всех сверхслов, допустимых для автомата A. Формула F специфицирует автомат A, если $W(A) = W(F)$.

Пусть $\Omega = \{p_1, p_2, \ldots, p_m\}$ — сигнатура формулы $F(t)$, а r — глубина этой формулы. Последовательность $\sigma_0, \sigma_1, \ldots, \sigma_r$ векторов из $\Sigma(\Omega)$ назовем состоянием глубины r, а множество $Q(r, \Omega)$ всех таких последовательностей — пространством состояний глубины r для формулы $F(t)$. Формулу $F(t)$ будем рассматривать как пропозициональную формулу от переменных $p_1(t), \ldots, p_m(t)$, $p_1(t-1), \ldots, p_m(t-1), \ldots, p_1(t-r), \ldots, p_m(t-r)$. Если компоненты вектора σ_i в состоянии $q = \sigma_0, \sigma_1, \ldots, \sigma_r$ рассматривать как истинностные значения соответствующих атомов ранга $i - r$ при определенном упорядочении множества Ω, то можно говорить о значении формулы $F(t)$ на состоянии q.

На множестве $Q(r,\Omega)$ определим отношение N непосредственного следования так, что за каждым состоянием $q=\sigma_0,\sigma_1,\ldots,\sigma_r$ непосредственно следуют $2^{|\Omega|}$ состояний вида $\sigma_1,\ldots,\sigma_r,\sigma$, где $\sigma\in\Sigma(\Omega)$. Множество всех состояний, которые непосредственно следуют за q, обозначим $N(q)$.

При использовании языка L для спецификации $X-Y$-автоматов предикатные символы ставятся в соответствие входным и выходным двоичным каналам специфицируемого автомата. Поэтому множество предикатных символов Ω разбивается на два класса: входные и выходные, которые обозначаются соответственно U и W. При этом $X=\Sigma(U)$, $Y=\Sigma(W)$ и каждый вектор из $\Sigma(\Omega)$ можно рассматривать как пару $\langle x,y\rangle$, где $x\in X$, $y\in Y$.

Построим вспомогательный Σ-автомат $A'(F)=\langle\Sigma(\Omega),Q',\delta_A\rangle$. Множество состояний Q' — это все те состояния из $Q(r,\Omega)$, на которых $F(t)$ истинна. Пусть состояние $q\in Q'$ имеет вид $\sigma_0,\sigma_1,\ldots,\sigma_r$ и $\sigma\in\Sigma(\Omega)$, тогда $\delta_A(q,\sigma)=\sigma_1,\ldots,\sigma_r,\sigma$. Если таких состояний нет, то значение $\delta_A(q,\sigma)$ не определено. Автомат $A(F)$ определим как максимальный циклический подавтомат автомата $A'(F)$.

Как следует из определения функции переходов, все состояния автомата $A(F)$ вида $\sigma,\sigma_1,\ldots,\sigma_r$, где $\sigma\in\Sigma(\Omega)$, эквивалентны. Автомат $A^*(F)$, состояниями которого являются классы этой эквивалентности, эквивалентен автомату $A(F)$. Таким образом, в качестве состояний автомата, специфицируемого формулой F, можно рассматривать последовательности векторов из $\Sigma(\Omega)$ вида $\sigma_1,\ldots,\sigma_r$. Для автомата $A^*(F)$ состояние $\sigma_0,\sigma_1,\ldots,\sigma_r$ автомата $A(F)$ можно рассматривать как переход из состояния $\sigma_0,\sigma_1,\ldots,\sigma_{r-1}$ в состояние $\sigma_1,\ldots,\sigma_r$ под действием символа σ_r.

ИГРОВАЯ МОДЕЛЬ ОТКРЫТОЙ СИСТЕМЫ

Игра двух лиц осуществляется на размеченном двудольном графе $G=\langle V_1,V_2,\Delta_1,\Delta_2,P,L\rangle$, где V_1,V_2 — непересекающиеся множества вершин соответственно Игрока 1 и Игрока 2; $\Delta_1\colon V_1\to 2^{V_2}$, $\Delta_2\colon V_2\to 2^{V_1}$ — отображения, определяющие множество ребер графа, а именно: (v,v') — ребро, если $v\in V_i$, $v'\in\Delta_i(v)$ $(i\in\{1,2\})$; P — множество двоичных переменных (предикатных символов); $L\colon V_1\cup V_2\to 2^P$ — отображение, размечающее вершины графа G. Каждое ребро графа соответствует ходу игрока, так что $\Delta_i(v)$ определяет множество ходов игрока i, допустимых в позиции v. Такой двудольный граф называется ареной игры. Игра определяется ареной G, начальной вершиной $v_0\in V_1\cup V_2$, указывающей, какой из игроков делает первый ход, и условием выигрыша для одного из игроков (как правило, для Игрока 1). Предполагается, что для всякого $v\in V_1$ $\Delta_1(v)\neq\varnothing$ и для всякого $v\in V_2$ $\Delta_2(v)\neq\varnothing$. Допустимый процесс ρ — это бесконечная последовательность вершин, генерируемая игроками, осуществляющими поочередные ходы, и начинающаяся вершиной v_0. Таким образом, $\rho=v_0v_1v_2\ldots$, где для $n=1,2,3,\ldots$ $v_{2n-1}\in\Delta_1(v_{2n-2})$, $v_{2n}\in\Delta_2(v_{2n-1})$. Поведение игрока определяется его стратегией, ограничивающей возможность выбора очередного хода из множества допустимых ходов. Обычно интересуются стратегией Игрока 1, представляющей собой отображение $\varphi_1\colon v_0\cdot(V_2\cdot V_1)^*\to 2^{V_2}$ такое, что для всякого ρ, заканчивающегося вершиной $v\in V_1$, имеем $\varphi_1(\rho)\subseteq\Delta_1(v)$. Процессом, ограничиваемым стратегией φ_1, называется такой допустимый процесс, что для всякого $v_i\in V_1$ $v_{i+1}\in\varphi_1(v_0\ldots v_i)$. Стратегия, зависящая только от последней вершины процесса, называется позиционной, или не имеющей памяти.

Существует много различных способов задания условия выигрыша, таких как формулы логики LTL [10], условия распознавания для автоматов над бесконечными словами [11], условия четности, использующие раскраску вершин графа [12], и др. В рассматриваемом варианте игры условие выигрыша задается формулой $\forall tf(t)$ языка L с множеством предикатных символов P. Для всякого допустимого процесса $\rho = v_0 v_1 v_2 \ldots$ отображение L определяет бесконечное слово $L(\rho) = L(v_0)L(v_1)L(v_2)\ldots$ в алфавите $\Sigma(P)$. Стратегия φ_1 называется выигрышной, если для любого процесса ρ', ограничиваемого стратегией φ_1, существует такая модель для формулы $\forall tf(t)$, которая имеет 0-суффикс, совпадающий с $L(\rho')$.

СОГЛАСОВАНИЕ АВТОМАТОВ, СПЕЦИФИЦИРОВАННЫХ В ЯЗЫКЕ L

В рамках автоматного подхода игра рассматривается как взаимодействие двух автоматов, определяемое их циклической композицией. Такая композиция соответствует структуре, в которой входы и выходы одного из автоматов соединены соответственно с выходами и входами другого. Построение выигрышной стратегии игрока сводится к согласованию спецификации одного из взаимодействующих автоматов со спецификацией другого [9]. Рассмотрим вкратце процедуру согласования недетерминированного автомата Мура $A = \langle X, Y, Q_A, \chi_A, \mu_A \rangle$ с недетерминированным автоматом Мура $B = \langle Y, X, Q_B, \chi_B, \mu_B \rangle$. Заметим, что понятие согласованности не симметрично: из согласованности автомата A с автоматом B не следует согласованность автомата B с автоматом A.

Пусть $s \in Q_A$, $q \in Q_B$. Если в каждом такте сначала свое состояние изменяет автомат A, а затем автомат B, их взаимодействие определяется соотношениями $s(t) \in \chi_A(s(t-1), \mu_B(q(t-1)))$ и $q(t) \in \chi_B(q(t-1), \mu_A(s(t)))$. Если порядок изменения состояний автоматов противоположный, то соответствующие соотношения имеют вид $s(t) \in \chi_A(s(t-1), \mu_B(q(t)))$ и $q(t) \in \chi_B(q(t-1), \mu_A(s(t-1)))$. Эти соотношения определяют вид спецификаций соответствующих автоматов.

Согласованность автомата A с автоматом B определяется как свойство параллельной циклической композиции соответствующим образом специфицированных автоматов. Параллельная циклическая композиция $X-Y$-автомата A и $Y-X$-автомата B представляет собой максимальный циклический подавтомат прямого произведения соответствующих Σ-автоматов с алфавитом $\Sigma = X \times Y$. Предполагается, что автоматы A и B заданы соответственно спецификациями $\forall tF_A(t)$ и $\forall tF_B(t)$. Автоматы, специфицированные этими формулами, рассматриваются в общем для них пространстве состояний, определяемом сигнатурой и максимальной глубиной формул, специфицирующих автоматы A и B. Заметим, что спецификация параллельной циклической композиции автоматов является конъюнкцией спецификаций этих автоматов. Каждое состояние композиции имеет вид $\langle s, q \rangle$, где $s \in Q_A$, $q \in Q_B$. Так как автоматы и их композиция циклические, состояниям s и q соответствует одно и то же состояние общего для них пространства состояний [9].

Необходимость согласования спецификаций автоматов возникает в силу того, что поведение композиции автоматов должно удовлетворять условию выигрыша, которое может ограничивать поведение автомата B.

Рассмотрим процедуру согласования автомата A с автоматом B. В спецификацию автомата A добавляется формула, задающая условие выигрыша. В полученной таким образом спецификации автомата A определяются те условия, которые ограничивают поведение автомата B. Для этого спецификация представляется в виде пополненного множества дизъюнктов, в котором выделяются дизъюнкты, характеризующие частичность автомата A [9]. Отрицания таких дизъюнктов задают состояния в рассматриваемом пространстве состояний. Для определения, в каких из этих состояний ограничивается поведение автомата B, формула, задающая их, умножается на формулу $F_B(t)$, задающую множество всех состояний автомата B. Формулу $F_B(t)$ удобно представлять в дизъюнктивной нормальной форме. Каждая элементарная конъюнкция в таком представлении соответствует одному или нескольким переходам в автомате B. Полученное произведение определяет множество состояний автомата B, переходы в которые недопустимы. Поэтому в автомате A, рассматриваемом в общем пространстве состояний, удаляются переходы в такие состояния. Это значит, что композиция автоматов A и B не может перейти в соответствующие состояния. Удаление состояний в автомате A осуществляется путем умножения его спецификации на отрицание дизъюнкции формул, задающих удаляемые состояния, т.е. добавлением в спецификацию соответствующей формулы. Такое изменение спецификации может привести к появлению новых ограничений на поведение автомата B. В этом случае описанный процесс будет повторяться до тех пор, пока либо не будет новых ограничений, либо спецификация автомата A станет противоречивой, что выясняется при пополнении спецификации для определения множества дизъюнктов, характеризующих частичность нового автомата A.

В настоящей работе для построения композиции и удаления состояний использовалась система Dual синтеза автоматов по спецификации в языке L.

ПОСТРОЕНИЕ ВЫИГРЫШНОЙ СТРАТЕГИИ В ИГРЕ НИМ

В игре НИМ используется n^2 фишек ($n = 2, 3, 4, \ldots$), расположенных в n рядах. Первый ряд содержит одну фишку, второй — три, i-й ряд — $2i-1$ фишку. Играют два игрока, по очереди удаляющие из какого-либо ряда произвольное (ненулевое) количество фишек. Проигрывает игрок, забирающий последнюю фишку. Бесконечная игра разбивается на партии. Игру может начинать любой из игроков: Игрок 1 или Игрок 2. Очередную партию начинает выигравший игрок. После того как игрок забирает последнюю фишку, игра переходит в начальное состояние. Для упрощения изложения рассмотрим игру при $n = 2$ и двух фишках во втором ряду.

Игра моделируется двумя взаимодействующими автоматами. Один из них (Players) моделирует поведение игроков, а второй (Arena) — арены. Выходные переменные автомата Arena A, R, Q1, Q2 являются входными для автомата Players, а входные переменные автомата Arena F, S1, S2 — выходными для автомата Players. Входным и выходным переменным автоматов соответствуют входные и выходные предикатные символы их спецификаций. Рассмотрим интерпретацию (смысл) этих предикатных символов. Значение символа A в состоянии автомата Arena определяет, какой из игроков делает следующий ход: единичное значение A соответствует ходу Игрока 1, а нулевое — ходу Игрока 2. Очередность ходов игроков определяется формулами $A(t-1) \to \neg A(t)$, $\neg A(t-1) \to A(t)$. Символ R соответствует наличию фишки в первом ряду, а Q1, Q2 — соответственно наличию одной или двух фишек во втором ряду. Символ F соответствует удалению фишки из первого ряда, а S1, S2 — соответственно удалению одной или двух фишек из второго.

Состояния автоматов Players и Arena в течение одного такта изменяются в таком порядке: сначала изменяет состояние автомат Players, затем — автомат Arena. Поэтому спецификация автомата Players удовлетворяет выражению $s(t) \in \chi_A(s(t-1), \mu_B(q(t-1)))$, причем состояние этого автомата зависит только от входного сигнала в предыдущий момент времени. Спецификация автомата Players имеет следующий вид.

Inputs: A, R, Q1, Q2.

Outputs: F, S1, S2.

F(t) ⇒ R(t-1). S1(t) ⇒ (Q1(t-1) | Q2(t-1)). S2(t) ⇒ Q2(t-1).

F(t) | S1(t) | S2(t). ~F(t) | ~S1(t). ~F(t) | ~S2(t). ~S1(t) | ~S2(t).

Эти формулы записаны в соответствии с синтаксисом языка L, реализованного в системе Dual. Здесь и далее символ ~ обозначает отрицание, символ | — дизъюнкцию, а символы $\Rightarrow$ и $\Leftrightarrow$ — соответственно импликацию и эквивалентность. Формулы глубины 0 определяют ограничения на выходной алфавит автомата Players, при которых никакие два действия автомата не могут выполняться одновременно и в каждом такте выполняется одно из них. Символами входного алфавита автомата Players являются состояния автомата Arena, т.е. конституенты единицы от булевых переменных A, R, Q1, Q2. Ходы Игрока 1 соответствуют выходным сигналам автомата Players при поступлении на его вход состояния, в отметке которого имеется символ A, а ходы Игрока 2 определяются состояниями автомата Arena, имеющими отметки с символом ~A. Состояния автомата Arena разобьем на два множества: состояния типа A, определяющие ходы Игрока 1, и состояния типа ~A, определяющие ходы Игрока 2. Таким образом, выходной сигнал автомата Players, соответствующий ходу Игрока 1, переводит автомат Arena в состояние типа ~A, что соответствует следующему ходу Игрока 2.

Спецификация автомата Arena имеет такой вид:

Inputs: F, S1, S2.

Outputs: A, R, Q1, Q2.

~Q1(t) | ~Q2(t). R(t) | Q1(t) | Q2(t). A(t-1) $\Rightarrow$ ~A(t). ~A(t-1) $\Rightarrow$ A(t).

R(t) $\Leftrightarrow$ (R(t-1)&~F(t) | R(t-1)&~Q1(t-1)&~Q2(t-1)&F(t) | ~R(t-1)&Q1(t-1)&S1(t) | ~R(t-1)&Q2(t-1)&S2(t)).

Q1(t) $\Leftrightarrow$ (Q1(t-1)&~S1(t) | Q2(t-1)&S1(t)).

Q2(t) $\Leftrightarrow$ (Q2(t-1)&~S1(t)&~S2(t) | R(t-1)&~Q1(t-1)&~Q2(t-1)&F(t) | ~R(t-1)&Q1(t-1)&S1(t)| ~R(t-1)&Q2(t-1)&S2(t)).

Первые две формулы ограничивают множество возможных состояний автомата.

Начальному состоянию игры соответствует одно из начальных состояний автомата Arena: A(t)&R(t)&~Q1(t)&Q2(t) или ~A(t)&R(t)&~Q1(t)&Q2(t), определяющее, какой из игроков делает первый ход.

Рассматриваемая задача состоит в построении выигрышной стратегии Игрока 1, для чего формулируется условие выигрыша, которому должно удовлетворять поведение композиции взаимодействующих автоматов. Построение выигрышной стратегии (если она существует) состоит в нахождении таких ограничений на переходы в автомате Players, соответствующие ходам Игрока 1, чтобы условия выигрыша выполнялись независимо от поведения Игрока 2, на которое непосредственно влиять нельзя. Условие выигрыша формулируется как недопустимость (после хода Игрока 2) состояний арены, после которых безусловно следует проигрыш Игрока 1. Такие состояния характеризуются наличием единственной фишки и описываются формулами A(t)&R(t)&~Q1(t)&~Q2(t) и A(t)&~R(t)&Q1(t)&~Q2(t). Следовательно, условие выигрыша имеет вид ~A(t-1) $\Rightarrow$ ~(R(t)&~Q1(t)&~Q2(t) | ~R(t)&Q1(t)&~Q2(t)). Добавление этой формулы в спецификацию автомата Players налагает ограничения на поведение арены, а значит, и Игрока 2. Другими словами, в автомате Players запрещаются такие переходы, которые приводят к состояниям арены A(t)&~R(t)&Q1(t)&~Q2(t) или A(t)&R(t)&~Q1(t)&~Q2(t).

Поскольку условие выигрыша характеризует множество недопустимых состояний арены, т.е. состояний A(t)&~R(t)&Q1(t)&~Q2(t) и A(t)&R(t)&~Q1(t)&~Q2(t), в данном случае их не нужно находить, используя дизъюнкты второго рода в пополненном множестве дизъюнктов спецификации автомата Players с добавленным в нее условием выигрыша (как это следует из алгоритма, приведенного в [9]).

Прежде чем рассмотреть более подробно построение выигрышной стратегии Игрока 1, напомним, что состояниями взаимодействующих автоматов и их композиции являются состояния общего для них пространства состояний. При этом все автоматы рассматриваются как автоматы Мура, представленные в этом пространстве состояний. Поскольку глубина формул спецификации равна 1, элементарные конъюнкции, задающие состояния, имеют вид $s_0(t) \& s_1(t-1)$, где $s_1(t-1)$ и $s_0(t)$ — конъюнкции атомов соответственно рангов –1 и 0, образованных предикатными символами сигнатуры спецификации. Если состояние $s_0(t) \& s_1(t-1)$ рассматривать как переход в автомате, то конъюнкция $s_1(t)$ задает состояние автомата, из которого осуществляется переход в состояние $s_0(t)$ под действием входного символа, определяемого входными атомами в состоянии $s_0(t)$.

Спецификация композиции автоматов Players и Arena представляет собой конъюнкцию их спецификаций. Приведённый автомат композиции, полученный в результате синтеза в системе Dual, имеет следующий вид:

(A,R,Q2)	**3:**	S2 $\to$ 0 (R), S1 $\to$ 4 (R, Q1), F $\to$ 8 (Q2).	(R,Q2)	**1:**	$\underline{\text{S2} \to \text{2 (A, R)}}$, S1 $\to$ 5 (A, R, Q1), F $\to$ 9 (A, Q2).
(R,Q1)	**4:**	$\underline{\text{S1} \to \text{2 (A, R)}}$, $\underline{\text{F} \to \text{7 (A, Q1)}}$.	(A,R,Q1)	**5:**	S1 $\to$ 0 (R), F $\to$ 6 (Q1).
(Q2)	**8:**	S2 $\to$ 3 (A, R, Q2), $\underline{\text{S1} \to \text{7 (A, Q1)}}$.	(A,Q2)	**9:**	S2 $\to$ 1 (R, Q2), S1 $\to$ 6 (Q1).
(R)	**0:**	F $\to$ 3 (A, R, Q2).	(A,R)	**2:**	F $\to$ 1 (R, Q2).
(A,Q1)	**7:**	S1 $\to$ 1 (R, Q2).	(Q1)	**6:**	S1 $\to$ 3 (A, R, Q2).

Здесь параллельная композиция представлена в виде $X-Y$-автомата Мура с входным и выходным алфавитами, такими же, как у автомата Arena. Отметки состояний приведены в скобках, где указаны только те переменные, которые входят в отметку без отрицания. Так, например, (R) соответствует отметке (~A&R&~Q1&~Q2). Условие выигрыша ограничивает поведение этого автомата, запрещая в нем подчеркнутые переходы в недопустимые состояния.

Композиция автоматов Players и Arena определяет множество всех допустимых поведений игры. Поэтому для определения множества всех состояний автомата Arena, из которых имеется переход в недопустимые состояния, следует использовать спецификацию приведенного выше автомата композиции.

Построение стратегии Игрока 1 начинается с нахождения всех состояний композиции, из которых имеются переходы в недопустимые состояния. Это состояния **1** , **4** и **8** типа ~A. Переходы в них (соответствующие ходам Игрока 1) должны быть удалены, что приводит к удалению в композиции самих состояний. Рассмотрим, например, последовательность состояний арены A&R&~Q1&Q2, ~A&R&Q1&~Q2, A&~R&Q1&~Q2, начинающуюся начальным состоянием и определяемую последовательной генерацией автоматом Players сигналов ~F& S1&~S2 и F&~S1&~S2. Первый из них соответствует ходу Игрока 1, а второй — ходу Игрока 2. Состояние A&~R&Q1&~Q2 недопустимо, поэтому все переходы в состояние ~A&R&Q1&~Q2, из которого имеется переход в A&~R&Q1&~Q2, должны быть удалены, в частности, необходимо удалить переход из состояния A&R&~Q1&Q2 под действием сигнала ~F&S1&~S2.

Итак, сначала находятся состояния композиции, из которых осуществляется переход в недопустимые состояния. Для рассматриваемого примера в состояние **7**, которому соответствует формула A(t)&~R(t)&Q1(t)&~Q2(t), имеются переходы из состояний **4** и **8**, задаваемых формулами ~A(t)&R(t)&Q1(t)&~Q2(t) и ~A(t)&~R(t)&~Q1(t)&Q2(t). В состояние **2** имеются переходы из состояний **1** и **4**, задаваемых формулами ~A(t)&R(t)&~Q1(t)&Q2(t) и ~A(t)&R(t)&Q1(t)&~Q2(t). Затем определяются все переходы в состояния ~A(t)&R(t)&Q1(t)&~Q2(t), ~A(t) &

~R(t)&~Q1(t)&Q2(t) и ~A(t)&R(t)&~Q1(t)&Q2(t), соответствующие ходам Игрока 1. Так, переход из состояния A(t)&R(t)&~Q1(t)&Q2(t) в состояние ~A(t)&R(t)&Q1(t)&~Q2(t) под действием сигнала S1 (~F(t)&S1(t)&~S2(t)) определяется формулой S1(t)&~S2(t)&A(t-1)&R(t-1)&~Q1(t-1)&Q2(t-1). Кроме того, будут построены переходы F(t)&A(t-1)&R(t-1)&~Q1(t-1)&Q2(t-1), F(t)&A(t-1)&R(t-1)&~Q1(t-1)&~Q2(t-1), S1(t)&A(t-1)&~R(t-1)&Q1(t-1)&~Q2(t-1) и S2(t)&A(t-1)&~R(t-1)&~Q1(t-1)&Q2(t-1).

Добавление в спецификацию автомата Players дизъюнктов

~S1(t) | ~A(t-1) | ~R(t-1) | Q1(t-1) | ~Q2(t-1), ~F(t) | ~A(t-1) | ~R(t-1) | Q1(t-1),

~S1(t) | ~A(t-1) | R(t-1) | ~Q1(t-1) и ~S2(t) | ~A(t-1) | R(t-1) | Q1(t-1) | ~Q2(t-1),

представляющих отрицания формул, задающих указанные переходы, удаляет в нем соответствующие переходы. После добавления этих дизъюнктов из спецификации удаляется условие выигрыша и первый цикл построения выигрышной стратегии Игрока 1 заканчивается.

Если из некоторого состояниия композиции все переходы будут удалены, оно становится тупиковым и при построении композиции также будет удалено. Это означает, что переход в такое состояние в результате хода Игрока 2 отсутствует, т.е. поведение автомата Players, соответствующее Игроку 2, в состоянии, из которого этот переход был определен, ограничивается. Таким образом, после первого цикла работы алгоритма могут появиться новые недопустимые состояния, что влечет очередной цикл. Следовательно, после окончания первого цикла необходимо выяснить, не появились ли новые недопустимые состояния. Для этого синтезируется новая композиция, определяемая преобразованной спецификацией автомата Players и исходной спецификацией автомата Arena.

В первом цикле недопустимые состояния изначально известны. В остальных циклах необходимо определять наличие новых недопустимых состояний. Эти состояния находятся путем сравнения множеств переходов в каждом из оставшихся состояний типа ~A в предыдущей композиции и в новой. Если имеются состояния, количество переходов из которых уменьшилось (уменьшился недетерминизм), то все переходы в такие состояния (из состояний типа A) должны быть удалены.

В рассматриваемом примере (при начальном состоянии A(t)&R(t)&~Q1(t)&Q2(t)) после первого цикла построения стратегии остается одно состояние типа ~A, а именно ~A(t)&R(t)&~Q1(t)&~Q2(t), все переходы из которого сохранены. На этом построение стратегии заканчивается.

Заметим, что состояние типа A удаляется после выполнения очередного цикла по двум причинам: оно становится тупиковым в результате удаления из него всех переходов; оно становится «источником» за счет удаления всех переходов в него в результате удаления состояний типа ~A. Удаление «источника» не приводит к увеличению частичности никакого из оставшихся состояний типа ~A. Так, после удаления всех переходов в состояния ~A(t)&R(t)&Q1(t)&~Q2(t), ~A(t)&~R(t)&~Q1(t)&Q2(t) и ~A(t)&R(t)&~Q1(t)&Q2(t) они становятся «источниками» и в композиции, являющейся циклическим автоматом, отсутствуют. Вместо удаления переходов в эти состояния можно удалить сами состояния. Для этого нужно добавить дизъюнкции A(t) | ~R(t) | ~Q1(t) и A(t) | ~Q2(t). Следовательно, каждый цикл построения стратегии упрощается. Таким образом, построение выигрышной стратегии осуществляется только путем преобразования композиции автоматов.

После очередного цикла построения стратегии нахождение новых недопустимых состояний можно осуществлять следующим образом. В синтезированном автомате композиции анализируются все состояния типа ~A. Каждое состояние определяет количество теоретически возможных переходов из него. Так, например, количество возможных переходов из состояния ~A(t)&R(t)&~Q1(t)&Q2(t) равно трем. Если реальное количество переходов меньше, то это состояние недопустимо, поэтому не нужно выполнять сравнение с предыдущим или исходным вариантом композиции.

ЗАКЛЮЧЕНИЕ

В работе рассмотрен метод нахождения выигрышной стратегии одного из игроков в бесконечной игре двух лиц с нулевой суммой при автоматной интерпретации основных понятий соответствующей модели игры. Автоматная интерпретация понятий «игроки» и «арена» позволила для нахождения выигрышной стратегии использовать методы согласования автоматов, предложенные в [8, 9] для обеспечения корректного поведения взаимодействующих автоматов. Понятие корректности взаимодействия возникает в случае, если один или оба взаимодействующих автомата частичные. При этом корректность определяется требованием, чтобы автоматы никогда не могли оказаться в таких состояниях, когда переход в одном из них под действием сигнала, поступающего из другого, будет не определен.

Обычно в игровой модели реактивной системы один из игроков представляет систему, а другой — среду. Согласование системы со средой состоит в таком преобразовании поведения системы, при котором ее взаимодействие со средой становится корректным и сохраняет заданные свойства. Особенностью рассматриваемой модели игры является то, что оба игрока моделируются одним и тем же автоматом. При этом в процессе согласования следует удалять из проектируемого автомата не переходы в недопустимые состояния типа A, а переходы в состояния, из которых имеются переходы в недопустимые состояния.

Рассмотрена процедура согласования, основанная на синтезе автомата по его спецификации в языке L. В этой процедуре необходимо синтезировать автомат Arena и его композицию с автоматом Players. Построение выигрышной стратегии состоит в преобразовании композиции автоматов путем удаления в автомате Players некоторых переходов, соответствующих ходам Игрока 1. Все эти преобразования можно выполнять на уровне спецификаций, не синтезируя соответствующих автоматов. Однако при этом для нахождения недопустимых состояний необходимо осуществлять проверку непротиворечивости (т.е. пополнение) спецификации автомата Players после каждой ее модификации. Заметим, что пополнять следует только спецификацию автомата Players, а синтезируется композиция автоматов, спецификация которой включает в себя спецификации обоих автоматов. Однако это не сильно влияет на временную сложность синтеза, поскольку она в основном определяется размерностью задачи, т.е. количеством предикатных символов в спецификации и ее глубиной, а эти параметры у спецификаций автоматов и их композиции одни и те же. Какой из вариантов процедуры согласования проще, зависит от используемых алгоритмов синтеза и пополнения множества дизъюнктов, а также от их реализации в виде программ. По-видимому, при реализации процедуры на распределенных (параллельных) компьютерных системах преимущество будет иметь вариант на основе синтеза, поскольку большинство алгоритмов синтеза допускает высокую степень распараллеливания.

Заметим, что в статье рассматривается композиция инициальных автоматов, которая также является инициальным автоматом. Поэтому при определении недопустимых состояний анализируются не все состояния композиции типа ~A, а только те, которые принадлежат ее инициальному подавтомату. Таким образом, методы, основанные на понятии согласованности неинициальных автоматов [8, 9], могут потребовать значительно большего количества вычислений, чем описанный выше способ преобразования инициального автомата.

СПИСОК ЛИТЕРАТУРЫ

1. Pnueli A., Rosner R. On the synthesis of a reactive module // Proc. 16th ACM Symp. on Principles of Programming Languages. — New York: ACM Press, 1989. — P. 179–190.
2. Abadi M., Lamport L., Wolper P. Realizable and unrealizable specifications of reactive systems // Intern. Colloq. on Automata, Languages, and Programming; Lect. Notes Comput. Sci. — 1989. — **372.** — P. 1–17.
3. Dill D.L. Trace theory for automatic hierarchical verification of speed independent circuits // Adv. Res. in VLSI / J. Allen and F.T. Leighton (eds). — Cambridge: MIT Press, 1988. — P. 51–65.
4. Alfaro L. de, Faella M., Majumdar R., Raman V. Code aware resource management // Proc. 5th Intern. ACM Conf. on Embedded Software (EMSOFT 05). — New York: ACM Press, 2005. — P. 191–202.
5. Kremer S., Raskin J.-F. A game-based upsilon erification of non-repudiation and fair exchange protocols // Proc. 12th Intern. Conf. on Concurrency Theory (CONCUR 01); Lect. Notes Comput. Sci. — 2001. — **2154.** — P. 551–565.
6. Alfaro L. de, Henzinger T.A. Interface theories for component-based design // 1st Intern. Workshop on Embedded Software (EMSOFT 01); Lect. Notes Comput. Sci. — 2001. — **2211.** — P. 148–165.
7. Чеботарев А.Н. Регулярная форма спецификации детерминированных автоматов в языке L // Приклад. дискрет. математика. — 2010. — № 4. — С. 64–72.
8. Чеботарев А.Н. Взаимодействие автоматов // Кибернетика и системный анализ. — 1991. — № 6. — С. 17–29.
9. Чеботарев А.Н. Общий метод проверки согласованности взаимодействующих автоматов с конечной памятью // Там же. — 1999. — № 6. — С. 25–37.
10. Lichtenstein O., Pnueli A. Checking that finite state concurrent programs satisfy their linear specification // Proc. 12th ACM Symp. on Principles of Programming Languages. — New York: ACM Press, 1985. — P. 97–107.
11. Thomas W. Automata on infinite objects // Handbook of Theoret. Comput. Sci. / J. van Leeuven (ed.). — 1990. — P. 134–191.
12. Friedmann O., Lange M. Solving parity games in practice // Lect. Notes Comput. Sci. — 2009. — **5799.** — P. 182–196.

Поступила 23.04.2013