КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

A.N. Chebotarev

TRANSFORMATION OF RESTRICTIONS TO THE BEHAVIOR OF OPERATIONAL PART OF A REACTIVE ALGORITHM INTO THE RESTRICTIONS TO THE BEHAVIOR OF ITS CONTROL PART

An approach to the development of a logical specification of the control part of a reactive algorithm is proposed based on the procedure of checking the compatibility of interacting automata.
Key words: specification, language L, compatibility, automata composition.

Запропоновано підхід до проектування специфікації керуючої частини реактивного алгоритму, що базується на використанні процедури перевірки узгодженості взаємодіючих автоматів.
Ключові слова: специфікація, мова L, узгодженість, композиція автоматів.

Предложен подход к проектированию спецификации управляющей части реактивного алгоритма, основанный на использовании процедуры проверки согласованности взаимодействующих автоматов.
Ключевые слова: спецификация, язык L, согласованность, композиция автоматов.

УДК 519.713.1

А.Н. ЧЕБОТАРЕВ

ПРЕОБРАЗОВАНИЕ ОГРАНИЧЕНИЙ НА ПОВЕДЕНИЕ ОПЕРАЦИОННОЙ ЧАСТИ РЕАКТИВНОГО АЛГОРИТМА В ОГРАНИЧЕНИЯ НА ПОВЕДЕНИЕ ЕГО УПРАВЛЯЮЩЕЙ ЧАСТИ

Введение. Формальная спецификация реактивного алгоритма – это совокупность утверждений, определяющих требования к его поведению. Эти утверждения формулируются в виде формул формального языка, для чего обычно используются логические языки, такие как темпоральные логики, исчисление предикатов первого порядка и прочее. В этой работе в качестве такого языка рассматривается простой логический язык L [1]. Спецификация требований состоит из двух частей: спецификации управляющей части алгоритма и спецификации операционной части. В качестве математической модели, определяющей семантику каждой из частей спецификации, используется конечный автомат. Поскольку эти модели определяют поведение реактивного алгоритма, работающего потенциально бесконечно, рассматриваются автоматы над бесконечными входными последовательностями (сверхсловами). Автоматы, специфицирующие управляющую и операционную части алгоритма, будем называть соответственно управляющим и операционным. Они взаимодействуют друг с другом, образуя структуру, в которой входы и выходы одного из них соединены соответственно с выходами и входами другого. Характер этого взаимодействия определяется понятием композиции автоматов. Существенное влияние на возможное поведение такой композиции оказывает частичность одного или обоих автоматов. В случае частичности взаимодействующих автоматов возникает

проблема обеспечения корректности их совместного поведения, т. е. исключения такой ситуации, при которой в процессе функционирования автоматов переход в одном из них под действием входного сигнала, поступающего из другого, будет не определен. Обеспечение корректности функционирования композиции автоматов налагает на поведение управляющего автомата соответствующие ограничения, которые должны быть отображены в его спецификации.

Операционная часть алгоритма реализуется в виде достаточно регулярной структуры (программной или аппаратурной), поэтому построение спецификации и реализация этой части представляют собой существенно более простые задачи по сравнению с аналогичными задачами для управляющей части. В проектировании реактивного алгоритма наиболее сложной задачей является построение спецификации требований к функционированию управляющей части алгоритма, чему и посвящена настоящая работа.

Поскольку построение спецификации – это неформальный процесс, осуществляемый человеком, то спецификация может содержать ошибки, устранение которых после синтеза и на последующих этапах проектирования превращается в очень сложную задачу. Потому необходимо так организовать процесс построения спецификации, чтобы уменьшить вероятность появления в ней ошибок. Для этого предлагается при написании спецификации управляющей части учитывать не все аспекты поведения операционной части с тем, чтобы затем спецификацию формально преобразовать к виду, учитывающему эти аспекты. При этом проектируемая спецификация управляющей части алгоритма пополняется путем добавления к ней новых утверждений, формируемых автоматически. Это упрощает написание спецификации и, соответственно, уменьшает вероятность допущения ошибки. В настоящей работе предлагается способ такого формального преобразования спецификации, который будем называть распространением ограничений из спецификации операционной части на спецификацию управляющей. В основе такого преобразования лежит метод проверки согласованности спецификаций взаимодействующих автоматов [2, 3].

Язык спецификации L. Язык L [1] представляет собой логику 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 $\forall tF(t)$, где $F(t)$ – формула с единственной переменной t, построенная с помощью логических связок из атомарных формул (атомов) вида $p(t + k)$, где p – одноместный предикатный символ, t – переменная, принимающая значения из множества **Z**, а k – целое число, называемое *рангом атома*. Разность между максимальным и минимальным рангами атомов в формуле $F(t)$ называется ее *глубиной*. Поскольку $F(t)$ интерпретируется на множестве целых чисел, то для произвольного целого k имеет место эквивалентность $\forall tF(t) \Leftrightarrow \forall tF(t + k)$, где $F(t + k)$ обозначает формулу, получающуюся из $F(t)$ путем прибавления k к рангам всех ее атомов. Таким образом, можно ограничиться рассмотрением формул $F(t)$, у которых максимальный ранг атомов равен 0 (нормализованные вправо формулы).

При определении семантики языка L он рассматривается как формализм для задания множеств бесконечных слов (сверхслов) в алфавите двоичных векторов, длина которых равна числу различных предикатных символов в формуле.

Пусть Σ – конечный алфавит, $\mathbf{Z}$ – множество целых чисел, $\mathbf{N}^+ = \{z \in \mathbf{Z} \mid z > 0\}$. Отображения u: $\mathbf{Z} \to \Sigma$ и l: $\mathbf{N}^+ \to \Sigma$ называются соответственно *двусторонним сверхсловом* (обозначается $\ldots u\,(-2)\,u\,(-1)\,u\,(0)\,u\,(1)\,u\,(2)\ldots$), и *сверхсловом* (обозначается $l\,(1)\,l\,(2)\ldots$) в алфавите Σ. Для двустороннего сверхслова u и $n \in \mathbf{Z}$ определим *n-суффикс* $u\,(n{+}1, \infty)$ как сверхслово $u\,(n{+}1)\,u\,(n{+}2)\ldots$.

Пусть $\Omega = \{p_1, p_2, \ldots, p_q\}$ – множество всех предикатных символов, которые встречаются в формуле $F(t)$ (сигнатура формулы). Область интерпретации языка L есть множество $\mathbf{Z}$, и не интерпретированы только предикатные символы. Таким образом, интерпретация формулы $\forall t F(t)$ – это набор $< \pi_1, \pi_2, \ldots, \pi_q >$ определенных на $\mathbf{Z}$ одноместных предикатов, которые соответствуют всем предикатным символам из множества Ω. Каждый такой предикат π_i можно рассматривать как двустороннее сверхслово в алфавите $\{0, 1\}$, а набор из q таких предикатов – как двустороннее сверхслово в алфавите $\Sigma(\Omega) = \{0, 1\}^q$. Мы не будем различать интерпретации и соответствующие двусторонние сверхслова в алфавите $\Sigma(\Omega)$, поэтому будем говорить об истинности и ложности формулы $\forall t F(t)$ на двустороннем сверхслове u. Интерпретация, при которой формула $\forall t F(t)$ истинна, называется *моделью* для этой формулы. С каждой формулой $F = \forall t F(t)$ ассоциируется множество $M(F)$ всех моделей для нее. Таким образом, формула $F = \forall t F(t)$ определяет множество сверхслов в алфавите $\Sigma(\Omega)$, которое будем обозначать $W(F)$, а именно, множество 0-суффиксов всех двусторонних сверхслов из $M(F)$. При использовании языка L для спецификации 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 U и W соответственно. Определим входной алфавит X и выходной алфавит Y как множества всех двоичных векторов длины $|U|$ и $|W|$ соответственно. Каждый вектор из $\Sigma(\Omega)$ можно рассматривать как пару $< x, y >$, где $x \in X$, $y \in Y$, поэтому для сверхслов в алфавите $\Sigma(\Omega)$ будем использовать обозначение $< x_1, y_1 > < x_2, y_2 > \ldots$.

Определим автоматную семантику языка L.

Определение 1. Конечный X–Y-автомат Мура A – это пятерка $< X, Y, Q, \chi, \mu >$, где X и Y – входной и выходной алфавиты соответственно, Q – конечное множество состояний, а χ: $Q \times X \to 2^Q$ и μ: $Q \to Y$ – соответственно функции переходов и выходов. Автомат A называется *детерминированным*, если для любых $x \in X$, $q \in Q$ $\quad |\chi(q, x)| \le 1$; в противном случае он называется *недетерминированным*.

Определение 2. X–Y-автомат $A = < X, Y, Q, \chi, \mu >$ называется *циклическим*, если для каждого $q \in Q$ существуют такие $q_1, q_2 \in Q$ и $x_1, x_2 \in X$, что $q_1 \in \chi(q, x_1)$ и $q \in \chi(q_2, x_2)$.

Циклический X–Y-автомат можно охарактеризовать в терминах допустимых сверхслов.

Определение 3. Вход-выходное сверхслово $l = < x_1, y_1 > < x_2, y_2 > \ldots$ *допустимо в состоянии* q автомата A, если существует такое сверхслово состояний $q_0 q_1 q_2 \ldots$, где $q_0 = q$, что для каждого $i = 0, 1, 2, \ldots$ $q_{i+1} \in \chi(q_i, x_{i+1})$, а $\mu(q_{i+1}) = y_{i+1}$. Вход-выходное сверхслово l *допустимо для автомата* A, если оно допустимо хотя бы в одном из его состояний.

Обозначим $W(A)$ множество всех вход-выходных сверхслов, допустимых для автомата A. Автомат A *удовлетворяет* спецификации F, если $W(A) = W(F)$. Рассмотрим способ построения автомата $A(F)$, удовлетворяющего спецификации $F = \forall t F(t)$ [1].

Пусть $\Omega = \{p_1, p_2, \ldots, p_q\}$ – сигнатура формулы $F(t)$, а r – глубина этой формулы. Последовательность $s_0, s_1, \ldots, s_r$ векторов из $\Sigma(\Omega)$ назовем *состоянием* глубины r, а множество $Q(r, \Omega)$ всех таких последовательностей – *пространством состояний* глубины r для формулы $F(t)$. Наровне с обозначением $s_0, \ldots, s_r$ для состояния глубины r будем использовать обозначение $< x_0, y_0 >, \ldots, < x_r, y_r >$. Формулу $F(t)$ будем рассматривать как пропозициональную формулу от переменных $p_1(t), \ldots, p_q(t), p_1(t–1), \ldots, p_q(t–1), \ldots, p_1(t–r), \ldots, p_q(t–r)$. Если компоненты вектора s_i в состоянии $q = s_0, s_1, \ldots, s_r$ рассматривать как истинностные значения соответствующих атомов ранга $i – r$ при некотором упорядочении множества Ω, то можно говорить о значении формулы $F(t)$ на состоянии q. Таким образом, любая формула $f(t)$ глубины r и сигнатуры Ω задает подмножество пространства состояний $Q(r, \Omega)$, а именно, множество тех состояний, на которых она истинна. Будем это множество обозначать $Q(f(t))$.

Проверка непротиворечивости спецификаций. Для проверки непротиворечивости спецификации используются резолюционные методы логического вывода [4, 5], эффективность которых удалось существенно повысить за счет учета специфики предметной области.

Пусть $F(t)$ задана в виде к.н.ф., все элементарные дизъюнкции которой нормализованы вправо. Как обычно, элементарную дизъюнкцию литер будем называть *дизъюнктом*. Дизъюнкт, не содержащий литер, называется *пустым*. К.н.ф. формулы $F(t)$ будем задавать в виде множества дизъюнктов.

Пусть c_1 и c_2 – нормализованные вправо дизъюнкты, $p(t)$ – атом нулевого ранга и $c_1 = c'_1 \vee p(t)$, а $c_2 = c'_2 \vee \neg p(t)$. Дизъюнкт $c = c'_1 \vee c'_2$ называется *R-резольвентой* дизъюнктов c_1 и c_2 по атому $p(t)$. Операция получения R-резольвенты двух дизъюнктов называется *R-резольвированием.*

R-выводом дизъюнкта c из множества дизъюнктов C называется такая конечная последовательность дизъюнктов $c_1, ..., c_k$, что $c_k = c$ и каждый дизъюнкт c_i ($i = 1, ..., k$) либо принадлежит C, либо является R-резольвентой двух предшествующих дизъюнктов, либо есть результат нормализации вправо дизъюнкта c_{i-1}.

Справедливо следующее утверждение [4]. Множество C нормализованных

вправо дизъюнктов противоречиво тогда и только тогда, когда существует *R*-вывод пустого дизъюнкта из *C*.

Дизъюнкт $c_1(t)$ *поглощает* дизъюнкт $c_2(t)$, если существует такое $k \in \mathbf{Z}$, что все литеры из $c_1(t+k)$ содержатся среди литер дизъюнкта $c_2(t)$. Описанная выше процедура может быть усовершенствована за счет удаления в процессе ее выполнения дизъюнктов, поглощаемых другими дизъюнктами, преобразуемого множества. Усовершенствованную таким образом процедуру будем называть *R-пополнением* множества дизъюнктов. Таким образом, формула $\forall tF(t)$, заданная множеством *C* нормализованных вправо дизъюнктов, невыполнима тогда и только тогда, когда в процессе *R*-пополнения *C* будет получен пустой дизъюнкт. Множество дизъюнктов называется *R-полным*, если процедура *R*-пополнения его не изменяет.

В работе [5] описано усовершенствование рассмотренного метода, связанное с разбиением множества дизъюнктов на несколько классов. Нам понадобится вариант этого метода, соответствующий разбиению дизъюнктов на два класса: *дизъюнкты первого рода*, содержащие выходные литеры нулевого ранга и *дизъюнкты второго рода*, не содержащие выходных литер нулевого ранга. Заметим, что здесь понятия входные и выходные литеры определены относительно управляющего автомата. *R*-резольвирование допускается только между дизъюнктами одного и того же класса. Правило *R*-резолюции с указанным ограничением на его применение называется *S-резолюцией*. Как и любой другой метод резолюций, этот метод заключается в применении к множеству дизъюнктов правила *S*-резолюции до тех пор, пока не произойдет насыщение, т. е. либо будет получен пустой дизъюнкт, либо каждая *S*-резольвента будет поглощаться одним из дизъюнктов, имеющихся в полученном множестве. Этот метод позволяет существенно сократить количество порождаемых резольвент по сравнению с методом *R*-резолюции.

Модель взаимодействия автоматов. В соответствии с автоматной семантикой реактивного алгоритма рассмотрение взаимодействия его или его части с соответствующей средой сводится к рассмотрению взаимодействия двух циклических автоматов. Взаимодействующие автоматы, один из которых моделирует поведение управляющей части алгоритма, а другой – поведение среды, с которой он взаимодействует, образуют структуру, в которой входы и выходы одного из автоматов соединены соответственно с выходами и входами другого. Пусть такая структура имеет вид, показанный на рис. 1, где $A = < X, Y, Q_A, \chi_A, \mu_A >$ – автомат Мура, соответствующий управляющей части алгоритма, а $B = < Y, X, Q_B, \chi_B, \mu_B >$ соответствует операционной части алгоритма.

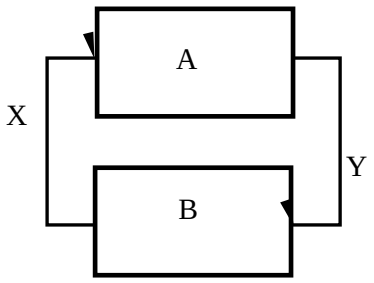

РИС. 1. Композиция автоматов

В случае, когда взаимодействующие автоматы образуют циклическую структуру, су-

щественное значение имеет разнесение во времени процессов изменения входных и выходных сигналов автоматов. Поэтому определение соответствующей композиции должно явно или неявно отражать невозможность порочного цикла в процессе изменения сигналов.

Пусть $q(t)$ обозначает значение состояния автомата A в момент времени t, $s(t)$ – значение состояния автомата B, а $x(t)$ и $y(t)$ – значения входов соответственно автоматов A и B в тот же момент времени. Рассматриваемый способ организации взаимодействия автоматов Мура A и B может быть описан соотношениями

$$q(t) \in \chi_A(q(t-1), x(t)),\ y(t) = \mu_A(q(t)),$$

$$s(t) \in \chi_B(s(t-1), y(t-1)),\ x(t) = \mu_B(s(t)).$$

Предполагается, что спецификации записываются таким образом, чтобы специфицируемые автоматы удовлетворяли этим соотношениям.

Рассматриваемое здесь понятие композиции ориентировано на аппаратурную реализацию, когда взаимодействие между управляющей и операционной частями алгоритма осуществляется с помощью передачи сигналов, а не сообщений.

Согласование автоматов. Существенное влияние на возможное поведение описанной выше структуры оказывает частичность одного или обоих автоматов. В случае частичности взаимодействующих автоматов возникает проблема обеспечения корректности их совместного функционирования. Это требование формализуется в виде понятия согласованности одного автомата с другим [2], которое определяется как свойство автономного автомата, называемого *параллельной циклической композицией* автоматов A и B.

Определение 4. Параллельная циклическая композиция автоматов $A = < X, Y, Q_A, \chi_A, \mu_A >$ и $B = < Y, X, Q_B, \chi_B, \mu_B >$ представляет собой автономный недетерминированный автомат $C = < Z, Q_C, \chi_C, \mu_C >$, где $Z = Y \times X$, $Q_C = Q_B \times Q_A$, а функция переходов χ_C и функция выходов μ_C определяются следующим образом: для любых $q, q' \in Q_A$ и $s, s' \in Q_B$ $(s'. q') \in \chi_C(s, q)$ тогда и только тогда, когда $q' \in \chi_A(q, \mu_B(s'))$, $s' \in \chi_B(s, \mu_A(q'))$; $\mu_C(s, q) = (\mu_B(s), \mu_A(q))$.

Рассмотрим алгоритм согласования автомата A с автоматом B при задании формул $F_A(t)$ и $F_B(t)$ в виде множеств дизъюнктов C_A и C_B соответственно [3].

Если спецификация автомата задана множеством дизъюнктов C, соответствующие ей автоматы в пространстве состояний $Q(r, \Omega)$ будем обозначать $A'(r, C)$ и $A(r, C)$. Поскольку множество Ω предикатных символов в дизъюнктах из C_A и C_B одно и то же, то, полагая его одинаково упорядоченным для обеих спецификаций, в качестве специфицируемых автоматов будем рассматривать автоматы $A = A(r, C_A)$ и $B = A(r, C_B)$, построенные в одном и том же пространстве состояний $Q(r, \Omega)$, ранг r которого равен максимальному из рангов дизъюнктов, принадлежащих C_A и C_B.

В работе [2] показано, что состояние $q \in Q(r, \Omega)$ принадлежит циклической композиции автоматов A и B только тогда, когда $F_A(t)$ и $F_B(t)$ истинны на q. Таким образом, формула $F_C = \forall t F_A(t)\ \&\ F_B(t)$, которой соответствует множество дизъюнктов $C_A \cup C_B$, специфицирует параллельную циклическую композицию автоматов A и B.

Состояние $q \in Q_C$, рассматриваемое в пространстве состояний $Q(r, \Omega)$, *некорректно*, если оно имеет область перехода, на которой $F_A(t)$ ложна, а $F_B(t)$ истинна. Согласование автомата A с автоматом B осуществляется путем удаления в автомате A соответствующих состояний.

Пусть автоматы $A = A(r, C_A)$ и $B = A(r, C_B)$ специфицированы S-полными множествами дизъюнктов C_A и C_B, соответственно. Необходимым условием частичности автомата, специфицированного S-полным множеством дизъюнктов C, является наличие в C дизъюнктов второго рода. Действительно, пусть q – состояние автомата, специфицированного формулой $\forall t F(t)$, соответствующей множеству дизъюнктов C. Состояние q называется *частичным*, если существует область перехода $N(q, x)$, на которой формула $F(t)$ ложна. Если дизъюнкт c принадлежит C, то формула $F(t)$ ложна на множестве состояний, задаваемом конъюнкцией литер (конъюнктом) $\neg c$. Очевидно, что любой области перехода соответствует конъюнкт, отрицание которого является дизъюнктом второго рода. Таким образом, наличие в C_A дизъюнкта второго рода свидетельствует о том, что среди состояний, задаваемых C_A, имеются частичные состояния.

Проверка согласованности автомата A с автоматом B начинается с того, что все дизъюнкты первого рода из C_B добавляются к C_A, после чего полученное множество дизъюнктов S-пополняется. Если пополненное множество дизъюнктов, которое обозначим C'_A, не содержит дизъюнктов второго рода, то специфицируемый им автомат согласован с автоматом B. Если C'_A содержит дизъюнкты второго рода, то начинается процесс согласования. Для каждого такого дизъюнкта $c(t)$ прежде всего необходимо выяснить, на каких областях перехода, содержащихся в $Q(\neg c(t))$ истинна формула $F_B(t)$.

Это делается путем логического перемножения $\neg c(t)$ и $F_B(t)$. Если $\neg c(t)\ \&\ F_B(t) \equiv 0$, то переходим к рассмотрению следующего дизъюнкта второго рода. Если конъюнкция $F_c(t) = \neg c(t)\ \&\ F_B(t)$ не равна тождественно нулю, то она задает множество состояний, принадлежащих всем тем областям перехода из $Q(\neg c\ (t))$, на которых $F_B(t)$ истинна. Состояния, для которых эти области являются областями перехода, необходимо удалить из автомата A. Чтобы получить формулу, задающую множество удаляемых состояний, следует в произвольной д.н.ф. формулы $F_c(t)$ удалить все литеры нулевого ранга, а затем ранги всех оставшихся литер увеличить на 1.

Удалению из автомата $A(r, C)$ множества состояний, задаваемых формулой $f(t)$, соответствует добавление к C множества дизъюнктов, определяемого произвольной к.н.ф. формулы $\neg f(t)$. Если некоторые из дизъюнктов этой к.н.ф. не нормализованы вправо, то их следует нормализовать.

После того как описанная процедура удаления состояний будет выполнена для всех имевшихся в C'_A дизъюнктов второго рода, в результате чего к множеству C'_A, возможно, будут добавлены новые дизъюнкты, полученное множество дизъюнктов *S*-пополняется. Если после пополнения появятся новые дизъюнкты второго рода, то процесс повторяется до тех пор, пока при очередном *S*-пополнении либо будет получен пустой дизъюнкт, что говорит о несогласованности автомата *A* с автоматом *B*, либо все вновь полученные дизъюнкты второго рода будут поглощаться ранее имевшимися дизъюнктами. На этом процесс удаления состояний заканчивается. Теперь полученное множество дизъюнктов (вместе с дизъюнктами второго рода из C_B) необходимо проверить на непротиворечивость. Как уже говорилось, это удобно сделать методом *S*-резольвирования. В случае непротиворечивости полученного множества дизъюнктов автомат *A* согласован с автоматом *B*. Противоречивость же указанного множества дизъюнктов свидетельствует о невозможности согласовать автомат *A* с автоматом *B*.

Распространение ограничений из операционной части алгоритма на управляющую. Построение спецификации управляющей части алгоритма осуществляется исходя из требований, предъявляемых к функционированию композиции управляющего и операционного автоматов. Часто такие требования можно сформулировать в виде ограничений, налагаемых на допустимое поведение операционного автомата. Поскольку спецификация операционной части алгоритма предполагается фиксированной, эти требования могут быть удовлетворены только за счет подходящей спецификации управляющей части. Рассматриваемая задача состоит в том, чтобы сформировать такие утверждения спецификации управляющей части алгоритма, чтобы композиция управляющего и операционного автоматов удовлетворяла заданным требованиям и взаимодействие этих автоматов было корректным. При этом используется описанная выше процедура согласования спецификаций, с помощью которой проектируемая спецификация управляющей части алгоритма пополняется путем добавления к ней новых утверждений, формируемых автоматически. Таким образом осуществляется распространение ограничений из операционной части на управляющую. Это упрощает написание спецификации и, соответственно, уменьшает вероятность допущения ошибки.

Охарактеризуем вид ограничений в спецификации операционной части алгоритма, которые могут быть формально распространены на спецификацию управляющей части. Добавление к спецификации операционной части формул, выражающих эти ограничения, делает ее частичной. Таким образом, распространение ограничений возможно только в том случае, когда пополненное множество дизъюнктов спецификации операционной части содержит дизъюнкты первого рода.

Следует заметить, что при интерпретации приведенного метода согласования спецификаций управляющей и операционной частей алгоритма на уровне автоматов вместо автомата $A(r, C_A)$ рассматривается автомат $A'(r, C_A)$. Результа-

ты согласования этих автоматов с автоматом $A(r, C_B)$ совпадают. При этом состояния автомата $A'(r, C_A)$, не принадлежащие циклическому автомату $A(r, C_A)$, будут удалены этой процедурой, а не процедурой выделения циклического подавтомата из $A'(r, C_A)$.

Заключение. В работе предложен подход к проектированию логической спецификации управляющей части реактивного алгоритма, основанный на использовании процедуры проверки согласованности спецификаций взаимодействующих автоматов. Основная идея состоит в том, чтобы упростить написание спецификации, снизив тем самым вероятность допущения в ней ошибки. Упрощение исходной спецификации достигается за счет того, что при написании спецификации управляющей части алгоритма учитываются не все факторы, ограничивающие его поведение.

Процесс согласования спецификации управляющей части с операционной так преобразует спецификацию управляющей части, что взаимодействие этих частей становится корректным. Процедура проверки согласованности спецификаций используется в методологии доказательного проектирования [6] как верификационная процедура, в предложенном подходе она используется как процедура проектирования.

1. *Чеботарев А.Н.* Об одном подходе к функциональной спецификации автоматных систем // Кибернетика и системный анализ. – 1993. – № 3. – С. 31 – 42.
2. *Чеботарев А.Н.* Взаимодействие автоматов // Кибернетика. – 1991. – № 6. – С. 17 – 29.
3. *Чеботарев А.Н.* Общий метод проверки согласованности взаимодействующих автоматов с конечной памятью // Кибернетика и системный анализ. – 1999. – № 6. – С. 25 – 37.
4. *Чеботарев А.Н.* Проверка непротиворечивости простых спецификаций автоматных систем // Там же. – 1993. – № 3. – С. 3 – 11.
5. *Чеботарев А.Н.* Метод раздельного резольвирования для проверки выполнимости формул языка *L* // Там же. – 1998. – № 6. – С. 13 – 20.
6. *Чеботарев А.Н., Головинский А.Л.* Доказательное проектирование алгоритмов функционирования реактивных систем // Искусственный интеллект. – 2008. – № 3. – С. 771 – 780.

Получено 04.10.2013