УДК 519.68

*А.А. Летичевский*

# АЛГЕБРАИЧЕСКАЯ ТЕОРИЯ ВЗАИМОДЕЙСТВИЯ И КИБЕР-ФИЗИЧЕСКИЕ СИСТЕМЫ

## Введение

Термин «кибер-физическая система» (cyber-physical system — cps) стал популярным сравнительно недавно. Он использовался в контексте четвертой промышленной революции наряду с такими популярными терминами, как «интернет вещей», «облачные вычисления» и др. Кибер-физические системы — это системы, в которых компьютерные программы взаимодействуют с физическими объектами. Такие системы использовались и раньше (встроенные системы, системы управления промышленными объектами и т. п.). В данном случае речь идет о значительно более сложных системах, использующих современные технологии, в том числе интернет и средства его взаимодействия с пользователями [1].

Разработка современных кибер-физических систем активизировала теоретические исследования в области моделирования и верификации таких систем [2, 3]. Основные математические модели, которые используются в настоящее время для исследования кибер-физических систем, — это временны*е* и гибридные автоматы [4–6] и их разновидности. Также отмечается, что существующие модели недостаточны для преодоления проблем, возникающих в процессе разработки современных кибер-физических систем [1]. К таким проблемам относятся, в частности, проблема масштабируемости (scalability), т.е. возможность работы с распределенными системами, содержащими большое число разнообразных компонент, средства описания параллельных взаимодействующих процессов, проблема сложности алгоритмов верификации и тестирования, а также ряд других более специфических проблем. Относительно полный анализ современного состояния теории кибер-физических систем представлен в [7].

В настоящей статье предлагается новая математическая модель для кибер-физических систем, построенная на базе алгебраической теории взаимодействия и технологии инсерционного моделирования. Она отличается более высоким уровнем абстракции по сравнению с другими моделями и, следовательно, позволяет описывать более широкий класс систем. В то же время этот уровень абстракции сохраняет возможность строить алгоритмы анализа, синтеза и верификации, а также адаптировать существующие средства для работы с кибер-физическими системами.

В основе новой модели лежит понятие *полугрупповой транзиционной системы*, которое позволяет, в частности, отвлекаться от деталей, связанных с непрерывным течением времени, или от явления интерливинга, присущего функционированию многоагентных систем. Далее мы показываем, как строить полугрупповые атрибутные среды и их функции погружения. Затем обсуждается структура среды моделирования и основные задачи, которые необходимо решать для построения системы моделирования. Основная идея этих рассмотрений состоит в демонстрации возможности перенесения основных приемов технологии инсерционного моделирования на область кибер-физических систем.

## Алгебраическая теория взаимодействия

Общая теория информационных взаимодействий начинается с нейронных сетей Мак-Каллока Питса (МР43). Теория нейронных сетей привела к появлению теории абстрактных автоматов, которая позволяет изучать поведение и взаимодействие эволюционирующих систем независимо от их структуры. Первоначально теория автоматов развивалась как теория конечных автоматов, и алгебра Клини–Глушкова [8, 9] служила основным средством описания их поведения. В дальнейшем теория автоматов концентрировалась на исследовании вопросов анализа и синтеза, изучении обобщений конечных автоматов и на вопросах сложности. Сети из автоматов исследовались в прикладных областях, связанных с проектированием электронных схем компьютеров. Взаимодействие в явной и общей алгебраической форме появилось только в 70-х годах прошлого века как теория взаимодействующих информационных процессов. Она включает CCS (исчисление взаимодействующих процессов) [10, 11], π-исчисление Милнера [12], CSP (взаимодействующие последовательные процессы) Хоара [13], ACP (алгебра взаимодействующих процессов) [14] и много других ответвлений этих базовых теорий. Одновременно начали развиваться модели параллельных вычислений под влиянием практических запросов параллельного программирования. Наиболее абстрактные модели — это сети Петри [15], модели акторов (актеров) [16], а также широко распространенные идеи объектно-ориентированного (параллельного) программирования. Они занимают промежуточное место между сетевыми моделями (нейронные и автоматные сети, схемы потоков данных) и моделями теории процессов.

**Инсерционное моделирование** сформировалось в 1990-е годы как одно из направлений общей теории взаимодействия. Первоначальное название — модель взаимодействия агентов и сред. Основные понятия инсерционного моделирования (среда, агенты, функция погружения) введены в работах [17–19], опубликованных в 90-х годах. Идейным прототипом инсерционного моделирования следует считать модель взаимодействующих управляющего и операционного автоматов, предложенную В.М. Глушковым еще в 60-х годах прошлого века [20, 21] для описания структур вычислительных машин, а также ее развитие в теории дискретных преобразователей 70-х годов. В этих моделях система представляется в виде композиции двух автоматов — управляющего и информационного. Управляющий автомат играет роль агента, а информационный — роль среды, в которую погружен этот агент. Распределенные модели макроконвейерных параллельных вычислений, исследованные в 1980-е годы [22], еще больше приблизились к современной модели взаимодействия агентов и сред. В этих моделях процессы, соответствующие параллельно работающим процессорам, можно рассматривать как агенты, взаимодействующие в среде распределенных структур данных.

Модели, исследуемые в теории процессов, можно эквивалентным образом представить в виде композиции среды и погруженных в нее агентов. С начала 2000-х инсерционное моделирование становится инструментом разработки прикладных систем, верификации требований и спецификаций распределенных взаимодействующих процессов [23–27]. Система VRS, разработанная в Украине по заказу фирмы Моторола с участием сотрудников Института кибернетики им. В.М. Глушкова, применяется для верификации требований и спецификаций в области телекоммуникационных систем, встроенных систем и систем реального времени. Новая система инсерционного моделирования IMS [28], разработанная в Институте кибернетики, существенно расширяет область применения инсерционного моделирования. В настоящее время ведутся работы по адаптации этой системы к задачам исследования кибер-физических систем.

## Полугрупповые транзиционные системы

**Размеченные транзиционные системы**. Базовое понятие классической теории взаимодействия — *размеченная транзиционная система*. Она определяется множеством состояний $S$, множеством действий $A$ (в зависимости от приложений употребляются также термины *сигналы*, *метки*, *обозреваемые символы*, *операторы*, *сообщения*) и отношением переходов $G \subseteq S \times A \times S$. Высказывание $(s, a, s') \in G$ так же, как и сама тройка $(s,a,s')$ (переход системы), обозначается выражением $s \xrightarrow{a} s'$, а конечная или бесконечная последовательность сопряженных переходов вида

$$s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \ldots \xrightarrow{a_n} s_n \xrightarrow{a_{n+1}} \ldots$$

называется *историей* функционирования системы. Два перехода *сопряжены*, если первый из них оканчивается в состоянии, в котором начинается второй. Последовательность действий $a_1 a_2 \ldots a_n \ldots$ называется трассой, порожденной данной историей. Именно трассы служат основой построения инвариантов эквивалентности состояний транзиционных систем. *Трассовая эквивалентность* состояний определяется как равенство множеств трасс, порождаемых историями, которые начинаются в этих состояниях, а *бисимуляционная эквивалентность* — как эквивалентность деревьев трасс, которые растут из этих состояний.

Трассы в дискретных моделях рассматриваются как элементы свободной полугруппы слов $A^*$, порожденной множеством действий. Единица этой полугруппы (пустое слово) может использоваться для разметки так называемых «скрытых» переходов.

Модели систем, построенные на размеченных транзиционных системах, будем называть *дискретными моделями*. Обычно они привязываются к дискретному времени, хотя множества состояний и действий могут быть бесконечными и даже континуальными. В то же время при рассмотрении физических (технических) систем могут использоваться вещественные параметры, которые изменяются непрерывным образом и которые желательно включать в модель системы на соответствующих уровнях абстракции. В этом случае история может включать в себя непрерывные компоненты, которые не могут быть однозначно разделены на дискретную последовательность переходов и действий. Для того чтобы охватить эти обстоятельства, предлагается новый тип транзиционных систем.

**Полугрупповые транзиционные системы**. Так же, как и размеченная транзиционная система, *полугрупповая транзиционная система* состоит из трех компонент $< S, H, G >$, где $S$ — пространство состояний, $H$ — полугруппа трасс (наблюдаемых процессов полугрупповой транзиционной системы), $G$ — параметризованное отношение переходов. В отличие от размеченной транзиционной системы отношение переходов параметризуется временными параметрами:

$$G = \{g_t \subseteq S \times H \times S \mid t \in T \subseteq \boldsymbol{R}\}.$$

Здесь $T$ — множество моментов времени, которое предполагается аддитивной подполугруппой полугруппы $\boldsymbol{R}$ вещественных чисел по сложению.

Будем использовать следующие обозначения:

$$s \xrightarrow{h}_t s' \Leftrightarrow (s, h, s') \in g_t,$$

$$s \xrightarrow{h} s' \Leftrightarrow \exists (t \in T)(s, h, s') \in g_t.$$

Параметр $t$ перехода $s \xrightarrow{h}_t s'$ называется его длительностью. Не исключаются переходы длительности 0. Компоненты полугрупповой транзиционной системы должны удовлетворять следующим аксиомам:

$A1.\ t, t' \in T \Rightarrow t + t' \in T.$

$A2.\ \forall (s \in S)\exists (h \in H, s' \in S)(s \xrightarrow{h} s').$

$A3.\ s \xrightarrow{h}_t s' \xrightarrow{h'}_{t'} s'' \Rightarrow s \xrightarrow{hh'}_{t+t''} s''.$

$A4.\ s \xrightarrow{h}_{t+t'} s', t,\ t' \in T \Rightarrow \exists (h', h'', s'')(h = h'h'',\ s \xrightarrow{h'}_t s'' \xrightarrow{h''}_{t'} s').$

Аксиома $A1$ называется *аксиомой полугруппы*. Она выражает тот факт, что $T$ — полугруппа, а время направлено в сторону увеличения. В частности, если полугруппа $T$ имеет хотя бы один положительный элемент, то она бесконечна «вправо», т.е. имеет сколь угодно большие значения. С другой стороны, если в $T$ есть хотя бы один отрицательный элемент, она бесконечна также и «влево», т.е. имеет сколь угодно большие по модулю отрицательные значения. Примерами полугрупп моментов времени, которые используются в приложениях, могут служить следующие подгруппы:

— всех вещественных чисел (обычно рассматривается как группа);

— всех натуральных чисел (целых положительных);

— неотрицательных целых чисел;

— **Z** всех целых рациональных чисел (положительных, отрицательных, а также 0);

— всех вещественных чисел вида $\delta x$, где $\delta > 0, x \in \mathbf{Z}$;

— всех неотрицательных вещественных чисел, больших некоторого $\delta > 0$.

Использование этой полугруппы позволяет избегать парадоксов Зенона, связанных с делением конечных интервалов на бесконечное множество отрезков.

Аксиома $A2$ называется *аксиомой продолжения*. Из этой аксиомы следует, что любая история в полугрупповой системе может быть продолжена. Таким образом, вместо тупиковых состояний в полугрупповой системе могут быть неподвижные точки.

Аксиома $A3$ называется *аксиомой свертывания*. Она показывает, что любую конечную историю можно свернуть до одного перехода. В этой аксиоме предполагается, что все переменные связаны неявно квантором общности. Такое предположение будем делать и в дальнейшем для свободных переменных в формулах, если не указано противное.

Аксиома $A4$ называется *аксиомой развертывания*. Она показывает, что любой достаточно длинный переход можно развернуть в историю из нескольких переходов. Эту аксиому также можно использовать для восстановления движения в прошлом из движений в настоящем (включая движение в обратном направлении, если полугруппа $T$ содержит отрицательные числа). Мы не исключаем недетерминированность системы как при движении вперед, так и при движении в обратном направлении. Поэтому разложение перехода большой длительности на переходы меньшей длительности может быть неоднозначным.

Полугрупповая система порождает множество историй, конечную или бесконечную последовательность попарно сопряженных переходов вида

$$s_1 \xrightarrow{h_1}_{t_1} s_2 \xrightarrow{h_2}_{t_2} \ldots \xrightarrow{h_n}_{t_n} s_n \xrightarrow{h_{n+1}}_{t_{n+1}} \ldots,$$

а каждая конечная история длины $n$ порождает трассу $h_1 \ldots h_n$, элемент полугруппы трасс.

Множество параметризированных отношений переходов образует коммутативную полугруппу, если умножение определить с помощью формулы

$$(s, h, s' \in g_t g_{t'} \Leftrightarrow \exists (h', h'', s'')(h = h'h'',\ s'' \xrightarrow{h'}_t s'' \xrightarrow{h''}_t s').$$

**Теорема 1**. Для всех $t, t' \in T \Rightarrow g_t g_{t'} = g_{t+t'} = g_{t'} g_t$.

Доказывается как следствие из аксиом свертывания и развертывания. Действительно, из $(s, h, s') \in g_t g_{t'}$ следует посылка аксиомы свертывания и включение $g_t g_{t'} \subseteq g_{t+t'}$. Пусть теперь $(s, h, s') \in g_{t+t'}$. Тогда по аксиоме развертывания и определению произведения имеем $(s, h, s') \in g_t g_{t'}$. Первое равенство доказано, второе следует из коммутативности полугруппы $T(t + t' = t' + t)$.

Таким образом, полугруппа отношений переходов есть коммутативная полугруппа. Если полугруппа отношений переходов имеет неприводимую систему образующих, то ее можно построить путем определения переходов, принадлежащих образующим этого отношения, а затем замкнуть путем добавления произведений.

**Примеры полугрупповых систем**. Рассмотрим некоторые важные примеры полугрупповых транзиционных систем.

**Размеченные транзиционные системы.** Пусть $S$ — размеченная транзиционная система со множеством действий $Y$ и отношением переходов $s \xrightarrow{y} s'$. Положим $T = N = \{1, 2, ...\}$, в качестве полугруппы трасс $H$ возьмем свободную полугруппу, порожденную множеством $Y$, и определим переход за единицу времени полугрупповой системы $< S, H, G >$ соотношением $s \xrightarrow{h}_1 s' \Leftrightarrow h \in Y, s \xrightarrow{h} s'$. Переходы произвольной длительности определим рекурсивно соотношением

$$s \xrightarrow{h}_{n+1} s' \Rightarrow \exists (z \in Y, h' \in H, s'' \in S)(h = zh',\ s \xrightarrow{z}_1,\ s'' \xrightarrow{h'} ns').$$

Множество $G$ однозначно определяется отношением переходов $g_1$, которое является единственным образующим полугруппы параметризированных отношений переходов. Именно это отношение обычно и берут в качестве определения отношения переходов размеченной транзиционной системы**.** Для моделирования скрытых переходов нужно взять свободную полугруппу с единицей, и использовать эту единицу для разметки скрытых переходов.

**Фазовые потоки.** Для описания фазовых потоков, используемых в теории динамических систем, следует абстрагироваться от полугруппы трасс. Поэтому, чтобы не терять общности, полагаем $H = \{\varepsilon\}$ (единичная полугруппа) и трассы на переходах не пишем. Отношение переходов предполагается функциональным, т.е. отношение переходов — это преобразование множества состояний:

$$T = \boldsymbol{R},\ g_t : S \to S,\ g_0(s) = s,$$

$$s \xrightarrow{\varepsilon}_t s' \Leftrightarrow s \to_t s',\ s' = g_t(s).$$

**Теорема 2**. Полугруппа отношений переходов для фазового потока — коммутативная группа преобразований множества $S$, а отношения переходов взаимно-однозначны.

Единицей группы переходов служит преобразование $g_0$, произведение определяется равенствами $g_t g_{t'} = g_{t+t'}$, обратный элемент $g_t^{-1} = g_{-t}$. Частными случаями фазовых потоков являются полугрупповые транзиционные системы с многомерными евклидовыми пространствами $\boldsymbol{R}^n$ в качестве множества состояний, переходами, определяемыми с помощью групп диффеоморфизмов или системами обыкновенных дифференциальных уравнений, разрешенных относительно первых производных.

## Полугрупповые гибридные автоматы

Основное определение гибридного автомата принадлежит Т. Хенцингеру [5]. В литературе используются различные модификации. В качестве альтернативного определения гибридного автомата рассмотрим понятие полугрупповой системы, достаточно близкое к оригинальному определению Хенцингера. Назовем эту систему *полугрупповым гибридным автоматом*.

Полугрупповой гибридный автомат представляет собой параллельную композицию двух транзиционных систем. Эта композиция будет рассматриваться как полугрупповая транзиционная система. Одна из систем композиции — обычная размеченная система со множеством действий (событий по терминологии Хенцингера) $\Sigma$. Она играет роль управляющей системы, другая — непрерывная динамическая система (управляемая компонента), обладающая несколькими режимами функционирования. Переключение режимов функционирования определяется управляющей системой.

Полугруппа моментов времени $T = \boldsymbol{R}_{\geq 0}$ — множество неотрицательных вещественных чисел, состояния всей системы $\boldsymbol{S} \subseteq V \times S$, где $V$ — множество состояний управляющей компоненты, $S = D^2$, $D = \boldsymbol{R}^X$, $X = \{x_1, x_2, ..., x_m\}$ — множество вещественных переменных, определяющих фазовое пространство непрерывной части гибридного автомата, полугруппа $H$ — свободная полугруппа с единицей $\varepsilon$, порожденная множеством событий $\Sigma$. Пространство $\boldsymbol{R}^X$ используется вместо традиционного $\boldsymbol{R}^m$, во-первых, чтобы подчеркнуть, что порядок на множестве переменных $X$ не играет роли, и, во-вторых, чтобы использовать эти переменные в логических формулах в качестве атрибутов для определения свойств областей фазового пространства. Состояния управляющей компоненты определяют режимы функционирования непрерывной компоненты, ее переходы $v \xrightarrow{\sigma} v'$ соответствуют переключениям режимов от $v$ к $v'$.

Состояния непрерывной компоненты представляют собой пары $s = (u, \dot{u}) \in D^2$. Первый элемент пары определяет положение $u$ компоненты в фазовом пространстве, второй — скорость в точке $u$. Переходы за время $t$ непрерывной компоненты гибридного автомата определяются условием: $(v, s) \xrightarrow{\varepsilon}_t (v, s') \Leftrightarrow$ существует дифференцируемая функция $\varphi : [0, t] \to D$ такая, что

$$(\varphi(0), \dot{\varphi}(0)) = s, (\varphi(t), \dot{\varphi}(t)) = s', \forall (0 < \tau < t)(\varphi(\tau) \in inv(v) \wedge (\varphi(\tau), \dot{\varphi}(\tau)) \in flow(v),$$

где inv и flow — теоретико-множественные функции Хенцингера, определенные на состояниях управляющей системы:

$$inv : V \to 2^{\boldsymbol{D}}, \quad flow : V \to 2^{\boldsymbol{D}^2}.$$

Класс всех дифференцируемых функций, определенных на интервале $[0, t]$ и удовлетворяющих вышеуказанному условию (1), обозначим $F_{v,t}$ и будем называть их свидетелями перехода $(v, s) \xrightarrow{\varepsilon}_t (v, s')$. Заметим, что переход $(v, s) \xrightarrow{\varepsilon}_0 (v, s)$ логически следует из определения переходов управляемой компоненты.

Переходы управляющей компоненты в составе гибридного автомата определяют смену режимов. Они осуществляются за нулевое время и определяются следующим условием:

$$(v, s) \xrightarrow{\sigma}_0 (v', s') \Leftrightarrow v \xrightarrow{\sigma} v' \wedge (s, s') \in jump(\sigma),$$

где $jump : \Sigma \to 2^{S^2}$ — еще одна теоретико-множественная функция Хенцингера.

Переходы непрерывной компоненты и переходы смены режимов будем рассматривать как переходы, порождающие параметризированное отношение переходов полугруппового автомата. Теперь строим это отношение индукцией по длине трасс, полагая

$$g_t = \bigcup_{n=0}^{\infty} g_t^{(n)},$$

где

$$g_t^{(0)} = \{(v, s) \xrightarrow{\varepsilon} t(v, s')\} \mid v \in V, s \in S\},$$

$$g_t^{(1)} = \{(v, s) \xrightarrow{\sigma} t(v', s') \mid (v, s) \xrightarrow{\varepsilon} t'(v, s'') \xrightarrow{\sigma} 0(v', s''') \xrightarrow{\varepsilon} t''(v', s'), t' + t'' = t\},$$

$$g_t^{(n+1)} = g_t^{(n)} \cup \{(v, s) \xrightarrow{h\sigma} t(v', s') \mid (v, s) \xrightarrow{h} t''(v'', s'') \xrightarrow{\sigma} t'(v', s'), t' + t'' = t\}, n \geq 1.$$

Переменные $t', t'', s'', s'''$, фигурирующие в условиях, определяющих свойства переходов, предполагаются связанными квантором существования. Теперь все компоненты полугрупповой системы определены и для доказательства того, что система соответствует определению полугрупповой системы, осталось показать истинность ее аксиом.

В соответствии с определением параметризированного отношения переходов любой переход системы можно представить в виде конечной истории, в которой чередуются переходы непрерывной компоненты, имеющие длительность, отличную от нуля, и переходы смены режимов, имеющие нулевую длительность. Начинаться и кончаться эта история может как переходом непрерывной компоненты, так и переходом смены режима. Такие представления назовем *стандартными разложениями* переходов гибридного автомата. Докажем аксиомы гибридного автомата, рассматриваемого как полугрупповая система.

Аксиома продолжения следует из того, что $(v, s) \xrightarrow{\varepsilon} 0(v, s)$.

Для доказательства аксиомы свертывания для истории

$$(v, s) \xrightarrow{h} t(v', s') \xrightarrow{h'} t'(v'', s'')$$

следует рассмотреть четыре случая стандартных разложений первого и второго переходов и провести доказательство индукцией по сумме длин историй для этих переходов.

1. Первая история кончается, а вторая начинается сменой режимов. Тогда эти смены можно представить в виде

$$(v_1', s_1') \xrightarrow{\sigma'} 0(v', s') \xrightarrow{\sigma''} 0(v_1'', s_1'') \Rightarrow (v_1', s_1') \xrightarrow{\sigma'\sigma''} 0(v_1'', s_1'').$$

Выполнив соответствующую подстановку, сократим длину первой истории.

2. Первая история кончается сменой режимов, а вторая начинается переходом непрерывной компоненты:

$$(v_1', s_1') \xrightarrow{\sigma'} 0(v', s') \xrightarrow{\varepsilon} t''(v_1'', s_1'') \Rightarrow (v_1', s_1') \xrightarrow{\sigma'} t''(v_1'', s_1'').$$

Соответствующая подстановка сократит длину второй истории.

3. Первая история кончается переходом непрерывной компоненты, а вторая начинается сменой режимов. Рассматривается аналогично предыдущему случаю. Сокращается длина первой истории.

4. Первая история кончается, а вторая начинается переходом непрерывной компоненты:

$$(v_1', s_1') \xrightarrow{\varepsilon} {}_{t_1}(v_1', s_2') \xrightarrow{\varepsilon} {}_{t_2}(v_1'', s_1'') \Rightarrow (v_1', s_1') \xrightarrow{\varepsilon} {}_{t_1+t_2}(v_1'', s_1'').$$

Действительно, пусть свидетель первого перехода — функция $\varphi$, а второго — функция $\psi$. Тогда функция $\xi = (\varphi; \psi)$ — свидетель композиции двух переходов. Функция $\xi$ определена на интервале $[0, t_1 + t_2]$ так, что

$$\xi(\tau) = \varphi(\tau), 0 \le \tau \le t_1, \xi(\tau) = \psi(\tau - t_1), t_1 < \tau \le t_1 + t_2.$$

Оператор (( );( )) склеивает графики двух функций и называется последовательной графической их композицией. Базис индукции получается, если каждая из историй состоит из одного перехода.

Для доказательства аксиомы развертывания полезна следующая лемма.

**Лемма**. Для любых $t, t' \in T, (v, s) \xrightarrow{\varepsilon} {}_{t+t'}(v, s') \Rightarrow \exists s''((v, s) \xrightarrow{\varepsilon} {}_{t}(v, s'') \xrightarrow{\varepsilon} {}_{t'}(v, s'))$.

Переход $(v, s) \xrightarrow{\varepsilon} {}_{t+t'}(v, s')$ означает существование функции $\varphi \in F_{v, t+t'}$. Но эта же функция будет свидетелем перехода $(v, s) \xrightarrow{\varepsilon} {}_{t}(v, \varphi(t))$, а функция $\psi$, определенная на интервале $[0, t']$ так, что $\psi(\tau) = \varphi(t + \tau), 0 \le \tau \le t' - t$, является свидетелем перехода $(v, \varphi(t)) \xrightarrow{\varepsilon} {}_{t'}(v, s')$. Таким образом, следствие в утверждении леммы будет истинным при $s'' = (\varphi(t), \dot{\varphi}(t))$. Лемма доказана. Он позволяет разделить любой переход непрерывной компоненты в последовательность из двух переходов в любой момент времени протекания исходного перехода.

Для доказательства аксиомы развертывания рассмотрим стандартное разложение перехода $(v, s) \xrightarrow{h} {}_{t+t'}(v', s'), t, t' > 0$. Стандартное разложение — это конечная история. Префикс истории — это последовательность переходов, которые стоят в начале этой истории. Пусть $t''$ — минимальная длительность префикса рассматриваемого разложения, большая или равная $t$. Сворачивая префикс длительности $t''$ и оставшуюся часть истории в один переход, получим разложение

$$(v, s) \xrightarrow{h'} {}_{t''}(v'', s'') \xrightarrow{h''} {}_{t+t'-t''}(v', s').$$

Возможны два случая: $t'' = t, t'' > t$. В первом случае искомое разложение получено. Во втором случае в силу минимальности $t''$ получим, что в интервале $[t, t'']$ смена режимов не происходила. Поэтому, используя лемму 1, получаем разложение

$$(v, s) \xrightarrow{h'} {}_{t}(v'', s''') \xrightarrow{\varepsilon} {}_{t''-t}(v'', s'') \xrightarrow{h''} {}_{t+t'-t''}(v', s').$$

Наконец, свернув переходы после $t$, получим разложение

$$(v, s) \xrightarrow{h'} {}_{t}(v'', s''') \xrightarrow{h''} {}_{t'}(v', s'),$$

что и требовалось доказать.

Таким образом, рассмотренная конструкция гибридного автомата действительно есть полугрупповая система.

**Сравнение с автоматом Хенцингера**. Полугрупповой гибридный автомат отличается от автомата Хенцингера в нескольких аспектах. Во-первых, в качестве областей значений функций *inv*, *flow* и *jump* у Хенцингера используются предикаты, у нас — множества, на которых эти предикаты истинны. Это различие несущественно.

Более существенно различие в выборе состояний непрерывной компоненты. У Хенцингера это точка фазового пространства, в полугрупповом автомате состояние включает в себя скорость, что позволяет более свободно манипулировать историями функционирования гибридного автомата. В частности, при моделировании многоагентных систем реального времени часто возникает вопрос о синхронизации модельного времени. Аксиомы свертывания и развертывания позволяют решать эту задачу более просто по сравнению с автоматом Хенцингера.

В работе [5] рассматривается два варианта гибридного автомата. Один вариант связывается со свойством безопасности (safety semantics), другой — со свойством жизненности (liveness semantics). Первый вариант ближе всего подходит к понятию полугруппового гибридного автомата. Во втором варианте множество трасс сокращается путем исключения бесконечных трасс, имеющих конечную длительность (эффект парадоксов Зенона). Для полугрупповых систем это свойство можно обеспечить более конструктивным образом, поэтому опускаем эти рассмотрения.

**Полугруппы трасс.** Рассмотрим некоторые классы полугрупп трасс, которые важны для приложений.

**Свободные полугруппы**. Традиционно используются для построения трасс при моделировании программных систем. Символы образующих такой полугруппы связываются с операторами, которые выполняются программой. При этом мы абстрагируемся как от времени, которое требуется для выполнения оператора, так и от содержания самого оператора. Свободная полугруппа с единичным элементом (моноид) может использоваться для моделирования трасс со скрытыми переходами. При моделировании кибер-физических систем переходы, размеченные образующими свободной полугруппы (дискретные переходы), обычно чередуются с переходами в непрерывном времени.

**Частично коммутативные полугруппы**. Это такие полугруппы, часть образующих которых удовлетворяет соотношению перестановочности. Перестановочные символы обычно соответствуют операторам, не связанным информационно. Использование соотношений перестановочности позволяет преобразовывать модели параллельных программ для уменьшения интерливинга (значительно уменьшается число трасс, которые необходимо рассматривать для решения проблемы достижимости). Технику оптимизации интерливинга можно перенести и на модели кибер-физических систем.

**Полугруппы операторов присваивания**. Последовательность операторов присваивания эквивалентна одному параллельному присваиванию. В сочетании с условными присваиваниями используются для сворачивания трасс при доказательствах свойств программ. Дискретные абстракции трасс в непрерывном времени эквивалентны условным присваиваниям. Поэтому такие полугруппы могут использоваться и для доказательства свойств кибер-физических систем.

**Полугруппы с непрерывным временем**. Элементы таких полугрупп проще всего порождать с помощью функций $h:[0, L(h)) \to Y)$, заданных на полуоткрытых временных интервалах $[a, b) = \{t \in T \mid a <= t < b\}$. Умножение в таких полугруппах задается соотношениями

$$(h * h')(t) = h(t), t < L(h),$$
$$(h * h')(t) = h'(t - L(h)), L(h) \le t < L(h) + L(h').$$

Полугруппы с непрерывным временем удобно, в частности, использовать для моделирования нейронных сетей с непрерывными сигналами. Другой пример — среди вещественных переменных гибридного автомата можно выделить множество наблюдаемых переменных и рассмотреть полугруппу трасс, порожденную фрагментами непрерывного изменения этих переменных на полуоткрытых интервалах.

## Эквивалентность и алгебра поведений

Модели реальных систем используются для исследования их поведений. Поведение системы связано с некоторой эквивалентностью. Именно две системы эквивалентны, если они имеют одно и то же поведение. Алгебра поведений характеризует поведения систем и позволяет производить вычисления, необходимые для порождения трасс и доказательства свойств систем.

**Эквивалентность.** Полугрупповая система может рассматриваться как обычная размеченная транзиционная система, действия которой есть элементы полугруппы $H$, а переходы размечены временными параметрами так, что выполняются аксиомы полугрупповой системы. Если длительности переходов перенести в действия, т.е. рассматривать в качестве действий пары $(h,t), h \in G, t \in T,$ то получим дискретную систему, а эквивалентность состояний дискретной системы можно перенести на исходную полугрупповую систему. Понятия трассовой и бисимуляционной эквивалентности, а также алгебра поведений определены для дискретных систем. Остается перенести эти понятия на полугрупповые системы.

Рассмотрим эту ситуацию более подробно. Состояния полугрупповых транзиционных систем так же, как и в дискретном случае, рассматриваются с точностью до трассовой или бисимуляционной эквивалентности.

**Трассовая эквивалентность**. Сначала рассмотрим трассовую эквивалентность. Функционирование полугрупповой системы так же, как и в обычном случае характеризуется историями, т.е. конечными или бесконечными последовательностями сопряженных переходов. Пусть $s_0 \xrightarrow{h_1} t_1 \ldots \xrightarrow{h_n} t_n\ s_n$ — конечная история функционирования полугрупповой системы. Нормированной трассой, порожденной этой историей, назовем пару $(h_1 \ldots h_n, t_1 + \ldots + t_n)$.

Пусть $L(s)$ — множество всех нормированных трасс, порожденных историями, которые начинаются в состоянии $s$. Состояния s и $s'$ называются *трассово эквивалентными*, если $L(s) = L(s')$ . Любую конечную историю полугрупповой системы можно свернуть в один переход, длительность которого равна сумме длительностей переходов этой истории. Пусть $g_t(s) = \{h \mid \exists s'(s \xrightarrow{h}_t s')\}$ — множество переходов длительности $t$, которые начинаются в состоянии $s$.

**Теорема 3.** Имеет место следующая эквивалентность:

$$L(s) = L(s') \Leftrightarrow \forall (t \in T)(g_t(s) = g_t(s')).$$

Формула $\forall (t \in T)(g_t(s) = g_t(s')),$ смысл которой состоит в том, что все полугрупповые разметки переходов одинаковой длительности для двух состояний совпадают, представляет собой необходимое условие трассовой эквивалентности.

Докажем достаточность. Пусть $(h,t) \in L(s).$ Если свернуть историю, порождающую эту трассу, получим историю $(s \xrightarrow{h}_t s')$ длины 1, трасса которой должна принадлежать множеству $L(s')$.

Теорема доказана.

Таким образом, рассматривая состояния полугрупповой системы с точностью до трассовой эквивалентности, абстрагируемся от состояний. Условие трассовой эквивалентности теоремы 3 позволяет абстрагироваться и от представления элементов полугруппы трасс на историях с сохранением временных характеристик трасс.

**Бисимуляционная эквивалентность** слабее трассовой и определяется более тонким образом. Именно, бинарное отношение $R$ на множестве состояний полугрупповой системы назовем *отношением бисимуляции* (bisimulation), если для любой пары $(s, s')$ состояний имеют место следующие утверждения:

$$1)\ (s, s') \in R \wedge s \xrightarrow{h}_t r \Rightarrow \exists r'((r, r') \in R \wedge s' \xrightarrow{h}_t r'),$$
$$2)\ (s, s') \in R \wedge s' \xrightarrow{h}_t r' \Rightarrow \exists r((r, r') \in R \wedge s \xrightarrow{h}_t r).$$

Состояния $s$ и $s'$ полугрупповой системы называются *бисимуляционно эквивалентными* (bisimilar), если существует отношение бисимуляции $R$ такое, что $(s, s') \in R$.

Обычно в системах выделяется множество начальных состояний, и в этом случае система называется *инициальной*. Когда речь идет об инициальной системе, в определение отношения бисимуляции нужно добавить еще требование: если $(s, s') \in R$ и одно из этих состояний является начальным, то и другое тоже.

Эквивалентность систем (трассовая или бисимуляционная), как правило, определяется эквивалентностью их состояний. Для инициальных систем две системы объявляются эквивалентными, если каждое начальное состояние одной из них эквивалентно некоторому начальному состоянию другой. Различие между трассовой и бисимуляционной эквивалентностью проявляется только для случая недетерминированных систем. Полугрупповая система называется детерминированной, если она имеет только одно начальное состояние и из $s \xrightarrow{h}_t s'$ и $s \xrightarrow{h}_t s''$ следует, что $s'$ и $s''$ эквивалентны. Две детерминированные системы бисимуляционно эквивалентны тогда и только тогда, когда они трассово эквивалентны.

Поведение размеченной транзиционной системы является инвариантом эквивалентности состояний. Для конструктивного определения поведений строится алгебра поведений и поведения рассматриваются как решения систем уравнений вида $x = F(x)$ в этой алгебре ($x = (x_1, x_2, ...)$ допускается и бесконечное множество уравнений и неизвестных).

**Полугрупповая алгебра поведений** для полугрупповых транзиционных систем строится так же, как и алгебра поведений для их дискретных моделей: двухосновная алгебра, основное множество — поведения, другое множество — действия. Две операции: префиксинг $a.u$ и недетерминированный выбор $u + v$ ($a$ — действие, $u$ и $v$ — поведения), две константы 0 (тупиковое поведение) и $\perp$ (неопределенное поведение). На алгебре поведений определяется частичный порядок с наименьшим элементом $\perp$. Недетерминированный выбор — это ассоциативная, коммутативная и идемпотентная операция. Единственным отличием полугрупповой алгебры поведений является то, что на множестве действий определена полугрупповая операция. В связи с этим в полугрупповой алгебре поведений возникает еще одно тождество:

$$a.b.u = (ab).u.$$

Для вычисления поведений нужна полная алгебра поведений (каждое направленное множество имеет наименьшую верхнюю грань). Конструкция полной полугрупповой алгебры поведений повторяет конструкцию, описанную в [29] для размеченных транзиционных систем. Теперь можно решать уравнения в алгебре поведений и определить поведение системы в заданном состоянии.

Пусть задана некоторая полугрупповая система. Сопоставим каждому состоянию $s$ поведение $beh(s)$ системы в этом состоянии. Поведения — это элементы полугрупповой алгебры поведений. Эти поведения удовлетворяют системе уравнений:

$$beh(s) = \sum_{t \in T} \sum_{s \xrightarrow{h}_t s'} (h, t).beh(s').$$

Это бесконечная система, однако при достаточно конструктивном задании переходов она может использоваться для последовательного порождения историй и трасс.

Для полугрупповой алгебры поведений может быть доказана теорема о связи поведений с бисимуляционной эквивалентностью состояний.

**Теорема 4.** Два состояния полугрупповой системы бисимуляционно эквивалентны, если и только если их поведения совпадают.

Доказательство практически повторяет доказательство соответствующей теоремы для обычной алгебры поведений, оно здесь не приводится. Теорема 4 позволяет нормализовать транзиционные системы путем представления их состояний в виде выражений алгебры поведений.

Для решения некоторых задач вместо бисимуляционной эквивалентности лучше использовать более сильную трассовую эквивалентность. Важнейший пример такой задачи — задача распознавания достижимости некоторого свойства состояний транзиционной системы. Для трассовой эквивалентности в качестве алгебры поведений используется алгебра, которая получается из алгебры поведений добавлением тождества дистрибутивности:

$$a.(u + v) = a.u + a.v.$$

Если заменить операцию префиксинга последовательной композицией поведений, то получим хорошо известную алгебру Клини. Выражения этой алгебры определяют множества трасс, порождаемых состояниями транзиционных систем.

Алгебру поведений обычно обогащают, добавляя новые операции, которые определяются с помощью систем уравнений. К стандартным обогащениям относятся последовательная и параллельная композиции.

### Атрибутные полугрупповые среды

Полугрупповые транзиционные системы так же, как и временны*е* или гибридные автоматы, удобно использовать для моделирования простейших систем, состоящих из небольшого числа управляемых (непрерывных) и управляющих (дискретных) компонент. Для моделирования более сложных распределенных многоуровневых систем с большим числом взаимодействующих компонент необходимо использовать дополнительные средства, позволяющие моделировать структурные свойства таких систем. В технологии инсерционного моделирования такими средствами являются средства описания взаимодействия агентов и сред.

**Атрибутная полугрупповая среда**. Строится так же, как и атрибутная среда в инсерционном моделировании. Среда и агенты суть полугрупповые транзиционные системы, а их композиция определяется с помощью функции погружения. Описание среды представляет собой набор функциональных символов (сигнатура) многосортного языка исчисления предикатов, который используется для описания свойств системы (базовый логический язык). В приложениях инсерционного моделирования к верификации софтверных систем используется язык первого порядка, возможно, дополненный некоторыми модальностями темпоральной логики и lambda-выражениями для определения функционалов высших порядков. При использовании объектно-ориентированного программирования (ООП) описание среды реализуется описанием класса (для многоуровневых сред это может быть иерархия классов разных типов), а различные типы агентов, погруженных в соответствующие среды, определяются объектами своих классов (классов агентов).

Функциональные символы делятся на *интерпретированные* и *неинтерпретированные*. Интерпретированные символы не меняют свою интерпретацию в течение функционирования системы (например, арифметические операции и функционалы, функции, определенные с помощью компьютерных программ (методов при использовании объектно-ориентированного программирования) и т.п.). Неинтерпретированные символы, которые называются *атрибутами*, изменяют свою интерпретацию с течением времени. Они определяют состояние системы. При моделировании кибер-физических систем удобно различать непрерывные и дискретные атрибуты. Непрерывные атрибуты меняют свои значения непрерывно с течением времени, дискретные — только в моменты выполнения действий, которые эти значения изменяют. В остальные моменты времени они сохраняют значения, полученные при последнем изменении. Простые атрибуты (функциональные символы арности 0) меняют свои значения целиком, функции меняют значения в конечном множестве точек для дискретных атрибутов или областей (для непрерывных функциональных атрибутов).

Для описания функции погружения (взаимодействие агентов и сред) используются *локальные описания* (по другой терминологии — *базовые протоколы*).

Общая форма локального описания остается традиционной:

$$B = \forall x(\alpha(x) \rightarrow < P(x) > \beta(x)).$$

Здесь $x$ — список переменных простых типов (параметры локального описания), $\alpha(x)$ и $\beta(x)$ — формулы базового языка (пред- и постусловия), причем $\beta(x)$ может содержать операторы присваивания и другие операторы, равносильные операторам вызова методов или процедур. Эти операторы рассматриваются как формулы темпоральной логики, которые связывают значения атрибутных выражений после выполнения оператора с их значениями до выполнения оператора. К ним, в частности, относятся операторы, которые изменяют законы эволюции непрерывных атрибутов. Процесс $P(x)$ определяет наблюдаемый переход системы, как элемент полугруппы трасс, и длительность этого перехода.

Локальное описание системы определяет ее свойство, которое неформально можно описать следующим образом. Пусть система, представленная в виде композиции атрибутной полугрупповой среды и погруженных в нее агентов, находится в состоянии $s$. Тогда если для некоторого набора значений параметров $x$ на состоянии $s$ истинно предусловие $\alpha(x)$, то по истечении времени, определяемого процессом $P(x)$, система перейдет в новое состояние $s'$, на котором будет истинным условие $\beta(x)$. При этом переходе будет наблюдаем элемент полугруппы трасс атрибутной среды, определяемый процессом $P(x)$.

Локальные описания можно рассматривать двояким образом. С одной стороны, это формула, которая определяет некоторое свойство системы. С другой стороны, локальное свойство определяет множество разметок переходов системы для некоторого класса ее состояний или недетерминированный оператор на множестве состояний среды. Когда локальное описание используется в таком качестве, называем его *базовым протоколом*.

Базовые протоколы используются для порождения историй и трасс атрибутной среды. Возможны два подхода для решения этой задачи. Первый называется *конкретным*, второй — *символьным* моделированием. В первом случае имеем дело с конкретными состояниями атрибутной среды, т.е. такими состояниями, в которых все атрибуты, необходимые для вычисления предусловия, заданы точно. Новое состояние также должно быть конкретным. Переход вычисляется для кон-

кретного набора значений параметров. Для вычисления перехода должны быть заданы алгоритмы решения двух задач: вычисление предусловия и выбор конкретных значений атрибутов среды, таким образом, чтобы постусловие было истинным на новом состоянии среды. Переход возможен только при истинном значении предусловия и существовании состояния, на котором истинно постусловие.

При символьном моделировании состояние среды определяется свойствами этой среды, выраженными на базовом логическом языке. Условием применимости перехода является выполнимость конъюнкции формулы текущего состояния среды и предусловия локального описания. Для выполнения перехода используется *предикатный трансформер* — преобразование формулы исходного состояния в формулу, определяющую возможные новые состояния. Подробные описания предикатных трансформеров для дискретных сред и достаточно широкого класса формул представлены в [27]. Для полугрупповых атрибутных сред описания предикатных трансформеров предполагается представить в дальнейших публикациях.

**Пример.** Для иллюстрации основных понятий атрибутной полугрупповой среды рассмотрим простейший пример. Это слегка модифицированная модель термостата из статьи [5]. Среда состоит из дискретного атрибута l:enum (on, off) перечислимого типа, непрерывного атрибута x:real, интерпретированных функций символьного типа

Fon: symb – > symb, Foff: symb – > symb,

вещественных констант delta 0 < = delta1 и оператора z: = evolution (z0, F)while(u).

Атрибут l представляет два режима работы термостата: on (включен) и off (выключен). Режимами управляет контроллер, агент, погруженный в среду термостата, [delta 0, delta 1] — интервал переключения контроллера. Атрибут x представляет непрерывно изменяющееся состояние термостата (температуру) как функцию времени. Если y — символ вещественного переменного, то Fon(y) и Foff(y) суть алгебраические выражения, которые определяют функции от y. Они используются для определения эволюции термостата с помощью оператора z: = evolution(z 0, F(z)) while(u(z)). Семантика этого оператора определяется следующим образом. Пусть f: T – > R — решение уравнения $\dot{z} = F(z)$ с начальным условием z(0) = z0. Тогда закон эволюционирования атрибута z меняется от предыдущего на новый закон z(t) = f(t). При этом новый закон определен на максимальном интервале, на котором выполняется условие u(z). В примере Хенцингера Fon(y)=5 – 0,1x, Foff= ˜ – 0,1x.

Функция погружения контроллера в среду термостата определяется следующими локальными описаниями:

```
Launchon: Forall (t, tau) (
t = current time,
delta < = tau < = delta1 & x < 19
– >< after tau (l: = on) >
l: = on, x: = evolution (x(t + tau), Fon(x)) while(x < = 22)
);
Launch off: Forall(t, tau) (
t = current time,
delta < = tau < = delta1 & x > 21
– >< after tau (l: = off) >
l: = off, x: = evolution(x(t + tau), Foff(x)) while(x > = 18)
)
```

В описании этого примера использован шрифт и синтаксис ввода информации в систему IMS.

Непрерывный атрибут current time управляется средой и обычно обозначает время активации базового протокола, который обращается к этому атрибуту. Отсчитывается от начала вызова первого протокола, с которого начинается генерация текущей трассы. Для данного примера это может быть и относительное время, отсчитываемое от предыдущего переключения режима работы термостата. Момент времени t+tau служит моментом окончания предыдущего перехода среды и началом нового перехода со сменой режима.

Таким образом, если в начальном состоянии термостата $x(0)<=18.1$, а закон его эволюции определяется уравнением $\dot{x} = 0$, то история функционирования термостата будет иметь вид

$$s_1 \xrightarrow{\varepsilon} tau_0\ s_1 \xrightarrow{l:=on} 0\ s_1 \xrightarrow{\varepsilon} t_1\ s_1' \xrightarrow{\varepsilon} tau_1\ s_2 \xrightarrow{l:=off} 0$$
$$s_2 \xrightarrow{\varepsilon} t_2\ s_2' \xrightarrow{\varepsilon} tau_2\ s_3 \xrightarrow{l:=on} 0\ s_3 \ldots$$

С помощью аксиомы свертывания эту историю можно преобразовать в

$$s_1 \xrightarrow{l:=on} tau_0\ s_1 \xrightarrow{l:=off} t_1+tau_1\ s_2 \xrightarrow{l:=on} t_2+tau_2\ s_3 \xrightarrow{l:=off} t_3+tau_3\ s_4 \ldots$$

Приведенные базовые протоколы могут служить как для конкретного, так и для символьного моделирования. Конкретное моделирование может служить в основном для теоретических исследований. К недетерминизму приводит уже, скажем, выбор момента времени переключения режимов. Важную роль играет точность вычислений, ошибки округления, устойчивость решений и другие факторы. Поэтому при моделировании кибер-физических систем предпочтением пользуются символьные методы вычислений.

В этом примере недетерминизм проявляется только в выборе задержки tau при выполнении переключения режимов. При конкретном моделировании достаточно рассмотреть нижнее, верхнее и среднее значения tau. При символьном моделировании в формуле состояния будет фигурировать только одно неравенство для tau. Для остальных непрерывных величин current time и x также будут справедливы ограничения, которые следуют из ограничений для tau.

**Моделирование полугрупповой среды**. Основная задача моделирования состоит в порождении трасс, удовлетворяющих заданным условиям, по модели системы. Эти условия могут определять свойства заключительных состояний трасс (целевые состояния, goal states), их максимальные длительности, условия продолжения или отсечения трасс, условия безопасности (safety) и т.п.

Рассмотрим атрибутную среду, заданную системой локальных описаний. Для такой среды удобно сначала построить ее абстракцию, рассматривая базовые протоколы как действия. Полученная модель называется крупношаговой моделью. Полугруппа трасс для крупношаговой модели — это свободная полугруппа с единицей, порожденная базовыми протоколами. Базовые протоколы рассматриваются как неделимые сущности, параллельные композиции базовых протоколов выполняются по правилу интерливинга. Недетерминизм остается только в выборе следующего протокола для исполнения. Каждая трасса крупношаговой модели после ее завершения превращается в мелкошаговую трассу путем раскрытия процессов, содержащихся в базовых протоколах, и преобразовании одного перехода в последовательность переходов, совершаемых при выполнении действий процессов базовых протоколов. Затем строится мелкошаговая трасса, которая преобразуется в соответствии с тождествами соответствующей полугруппы (обычно это частично-коммутативная полугруппа). Семантика мелкошагового моделирования для дискретных систем представлена в [25].

Состояние среды в общем случае имеет вид $E[u_1, u_2, ...]$, где $E$ — неразложимое состояние среды, $u_1, u_2, ...$, — состояния агентов, погруженных в среду. Неразложимость состояния среды означает, что ее нельзя представить в виде нетривиальной композиции среды и агентов, погруженных в нее. При символьном моделировании состояние среды представляет собой формулу базового логического языка, определяющую ограничения на атрибуты среды и атрибуты агентов для многоуровневой структуры системы. Будем предполагать, что все непрерывные атрибуты суть атрибуты среды и они изменяют свое состояние непрерывно в соответствии с установленными для них законами эволюции, а изменение этих законов выполняется с помощью операторов типа evolution. Эти операторы могут задавать закон эволюции явно или неявно в виде уравнений или в виде ограничений, которым должны удовлетворять эти законы.

Первая задача, возникающая при моделировании среды, состоит в выборе протокола, который будет выполняться первым. С каждым базовым протоколом связаны атрибутные выражения, которые этот протокол использует и которые он изменяет. В случае дискретного времени в каждый момент проверяется применимость всех базовых протоколов и выбираются протоколы, которые могут быть выполнены в этот момент времени. В случае непрерывного времени применимость протоколов в идеале должна проверяться непрерывно. Это можно обеспечить путем приближенных оценок, проверяя условия применимости с некоторым малым шагом по времени. В силу того, что вычисления с вещественными числами выполняются приближенно, результаты моделирования будут зависеть от временного кванта и, уменьшая его, можно получать все более точные результаты, хотя при этом будет увеличиваться количество трасс и соответственно сложность вычислений.

Более эффективная проверка может быть выполнена путем решения задачи вычисления или оценки минимального времени, когда предусловие рассматриваемого протокола может быть выполнимо на заданном состоянии среды в это время. Время применимости протокола зависит только от законов эволюции непрерывных атрибутов, которые используются или изменяются в данном протоколе.

Другой метод состоит в том, чтобы встроить проверку определенных условий в законы эволюции непрерывных величин. В моменты, когда выполняются эти условия, включаются проверки всех предусловий, которые могут быть выполнимы. Реализовать этот метод можно путем декомпозиции закона эволюции на отрезки, вставив на границах этих отрезков передачу сообщения среде о необходимости проверки применимости некоторых протоколов. В приведенной выше модели термостата оператор эволюции можно заменить более сложным оператором. Для протокола launchon это может быть

x: = evolution ((x (t+tau), Fon(x)) while(x>21); check point; continue while(x < = 22)).

Получив сообщение checkpoint, среда включает протокол launch on, если l = off, или launch off, если l = on.

Наконец, уменьшить количество проверок можно путем введения отношения следования на множестве базовых протоколов и введения двухуровневого управления моделью. Используемые методы двухуровневой генерации трасс для дискретных систем представлены в [27]. Они без труда переносятся на киберфизические системы.

**Верификация моделей.** Модель может выражать требования к системе и тогда мы рассматриваем задачу верификации требований. Нас могут интересовать полнота и непротиворечивость требований, сохранение свойств безопасности и достижимость состояний, удовлетворяющих определенным требованиям. Эти же

вопросы могут интересовать нас и в случае, когда модель построена по уже существующей системе и тогда нарушение определенных свойств модели может означать их нарушение и в системе. Особый случай представляют модели, выраженные в виде программ в языках высокого уровня. Для разных классов моделей методы верификации могут быть разными. Рассмотрим два подхода (статический и динамический) в том виде, как они применяются в системе IMS.

**Статическая верификация**. Обычно статическая верификация состоит в том, что по модели строятся некоторые логические формулы и общезначимость таких формул означает, что нужные свойства будут выполняться и для любой правильной реализации модели. Начало статическим методам положено работами Хоара и Флойда [30, 31], современные реализации [32] используют метод контрактного программирования. К статическим методам относят также абстрактную интерпретацию [33].

В системе IMS требования формализуются в виде локальных описаний. Две главные задачи верификации, которые рассматриваются в IMS, — это проверка свойств безопасности (safety) и проверка достижимости свойств. Для формулировки свойств состояний среды с погруженными в нее агентами используются формулы базового языка модели (формулы многосортного исчисления предикатов первого порядка). Свойство безопасности — это свойство, которое сохраняется на протяжении всего времени функционирования системы, например, некоторые величины непрерывных компонент должны удовлетворять определенным неравенствам, уравнениям и т.д. Для выполнения свойства безопасности достаточно, чтобы оно сохранялось каждым базовым протоколом. Точнее, пусть $B = \forall x(\alpha(x) \rightarrow < P(x) > \beta(x))$ — базовый протокол, а $\gamma$ — свойство безопасности. Тогда должно выполняться условие $\forall x(\gamma \wedge \alpha(x) \rightarrow pr(\gamma \wedge \alpha(x), \beta(x)))$, где $pr$ — предикатный трансформер, о котором говорилось выше. Условие сохранения является достаточным, но не необходимым условием безопасности. Просто одно из состояний, в котором нарушается условие сохранения, может быть недостижимым.

Статические методы могут применяться также для доказательства недостижимости некоторых условий. Условие недостижимо, если его отрицание является условием безопасности. Если не удается доказать безопасность или недостижимость, то приходится прибегать к динамическим методам.

**Динамическая верификация** состоит в выполнении модели путем конкретного или символьного моделирования (генерация трасс). В процессе выполнения модели проверяются условия безопасности и целевые условия. В случае, когда модель имеет конечное число состояний, полная верификация может быть достигнута исчерпывающим прохождением всех состояний. Эта техника используется в области проверки моделей [34] с языком темпоральной логики в качестве языка спецификаций. Основная область применения — верификация автоматных моделей технических устройств или конечных моделей программ, использующих достаточно высокий уровень абстракции. Для динамической верификации моделей с бесконечным множеством состояний невозможно осуществить исчерпывающее прохождение всех состояний, но, порождая все трассы ограниченной длины, можно найти нарушение условий безопасности или достижение целевых состояний. В IMS динамическая верификация выполняется путем символьного моделирования для моделей, заданных в виде систем базовых протоколов с использованием различных средств сокращения пространства поиска (количества генерируемых трасс и состояний). Различные уровни абстракции достигаются выбором формул для начальных состояний.

**Заключение**

Основной результат статьи состоит в рассмотрении новой модели кибер-физических систем, которая обобщает известные модели типа гибридных и временны*х* автоматов. После соответствующей адаптации эта модель может быть положена в основу дальнейшего развития систем инсерционного моделирования и их использования для разработки алгоритмов моделирования, верификации и тестирования многоагентных кибер-физических систем с многомерными непрерывными компонентами. В остальном статья носит обзорный характер, обсуждаются вопросы применения методов моделирования и верификации, накопленных в области программирования, к разработке кибер-физических систем.

*О.А. Летичевський*

## АЛГЕБРАЇЧНА ТЕОРІЯ ВЗАИМОДІЇ І КІБЕР-ФІЗИЧНІ СИСТЕМИ

Розглянуто нову модель кібер-фізичних систем, яка узагальнює відомі моделі типу гібридних та часових автоматів. Обговорюються питання застосування методів моделювання та верифікації, накопичених в області програмування, до розробки кібер-фізичних систем.

*A.A. Letichevsky*

## ALGEBRAIC INTERACTION THEORY AND CYBER-PHYSICAL SYSTEMS

A new model of cyber-physical systems is considered. The model generalizes known models like hybrid and time automata. The application of modeling and verification methods accumulated in the field of programming to the development of cyber-physical systems is discussed.

1. *Khaitan S.K, McCalley J.D.* Design techniques and applications of cyber physical systems : survey // IEEE Systems Journal. — 2014.
2. *Lee E.A. and Seshia S.A.* Introduction to embedded systems, a cyber-physical systems approach. — Cambridge : MIT Press, 2017. — 564 p.
3. *Alur R.* Principles of cyber-physical systems. — Cambridge : MIT Press, 2015.
4. *Raskin J.F.* An introduction to hybrid automata // in Hristu-Varsakelis / Dimitrios ed. // Handbook of Networked and Embedded Control Systems. — 2005. — P. 491–517.
5. *Henzinger T.A.* The theory of hybrid automata in Inan, M. Kemal and Kurshan R. P. ed. // Verification of digital and hybrid systems. — Berlin; Heidelberg : Springer, 2005. — P. 265–292.
6. *Bengtsson J. and Yi W.* Timed automata: semantics, algorithms and tools, in Desel J., Reisig W. and Rozenberg G. // Lectures on Concurrency and Petri Nets: Advances in Petri Nets. — Berlin; Heidelberg : Springer, 2004. — P. 87–124.
7. *Летичевский А.А., Летичевский А.А. мл., Скобелев. В.Г., Волков В.В.* Кибер-физические системы // Кибернетика и системный анализ. — 2017 — № 6. — С. 3–19.
8. *Kleene, Stephen C.* Representation of events in nerve nets and finite automata, in Shannon, Claude E.; McCarthy, John // Automata Studies. — Princeton University Press. — 1956. — P. 3–42.
9. *Глушков В.М.*, Об одном алгоритме синтеза абстрактных автоматов // Украинский математический журнал. — 1960 — **12**, № 2. — С. 147–156.
10. *Milner R.A.* Calculus of communicating systems // Lecture Notes in Computer Science. — N.Y. : Springer-Verlag, 1980. — **92.**
11. *Milner R.* Communication and concurrency. — N.Y. : Prentice Hall, 1989.
12. *Milner R.* The polyadic π-calculus: a tutorial. : Tech. Rep. ECS—LFCS—91—180. Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, UK. — 1991.

13. *Hoare C.A.R.* Communicating sequential processes. — New Jersey : Prentice Hall, 1985.
14. *Bergstra J A. and Klop J.W.* Process algebra for synchronous communications // Information and Control. — 1984. — 60 (1/3). — P. 109–137.
15. *Petri C.A.* Kommunikationmit automaten. — Bonn : Institut fur InstrumentelleMathematik, Schriften des IIM. — 1962. — N 2.
16. *Carl Hewitt, Peter Bishop, and Richard Steiger.* A universal modular actor formalism for artificial Intelligence, IJCA, 1973.
17. *Letichevsky A.A., Gilbert D.R.* : A universal interpreter for nondeterministic concurrent programming languages, in M. Gabbrielli, ed. // Fifth Compulog network area meeting on language design and semantic analysis methods. — 1996.
18. *Letichevsky A. and Gilbert D.* A general theory of action languages // Cybernetics and Systems Analysis. — 1998. — N 1.
19. *Letichevsky A. and Gilbert D.* A model for interaction of agents and anvironments, in D. Bert, C. Choppy, P. Moses, ed. Recent trends in Algebraic Development Techniques // Lecture Notesin Computer Science . — 1999. — **1827**.
20. *Глушков В.М.* Теория автоматов и вопросы проектирования структур цифровых машин // Кибернетика. — 1965. — № 1 — С. 3–12.
21. *Glushkov V.M. and Letichevsky A.A.,* Theory of algorithms and discrete processors // Advances in Information Systems Science. — N. Y. : Plenum Press. — 1969. — **1.** — P. 1–58.
22. *Капитонова Ю.В., Летичевский А.А.*, Математическая теория проектирования вычислительных систем. — М. : Наука, 1988. — 295 с.
23. *Baranov S., Jervis C., Kotlyarov V., Letichevsky A., and Weigert T.* Leveraging UML to deliver correct telecom applications in UML for Real: Design of Embedded Real-Time Systems by L.Lavagno, G. Martin, and B. Selic, ed. — Kluwer : Academic Publishers, 2003. — P. 323–342
24. *Kapitonova J., Letichevsky A., Volkov V. and Weigert T.* Validation of embedded systems, in R., Zurawski, ed. // The embedded systems handbook. — Miami : CRCPress, 2005.
25. *Летичевский А. Ад., Капитонова Ю.В., Волков В.А., Летичевский А.А., Баранов С.Н., Котляров В.П., Вейгерт Т.* Спецификация систем с помощью базовых протоколов // Кибернетика и системный анализ. — 2005. — № **4.** — С. 3–21.
26. *Letichevsky A., Kapitonova J., Letichevsky A.Jr., Volkov V., Baranov S., Kotlyarov V., Weigert. T.* Basic Protocols, message sequence charts, and the verification of requirements specifications // Computer Networks. — 2005. — **47**. — P. 662–675.
27. *Letichevsky, A.A. Letychevskyi O.A., Peschanenko V.S., Weigert T.* Insertion modeling and symbolic verification of large systems // Lecture Notes in Computer Science. — 2015 — **9369.** — P. 3–18.
28. *Letichevsky A.A., Letychevskyi O.A. and Peschanenko V.S.* Insertion Modeling System LNCS. — 2015. — **7162**. — P. 262–272.
29. *Letichevsky A.* Algebra of behavior transformations and its applications in V.B. Kudryavtsev and I.G. Rosenberg, ed. // Structural theory of automata, emigroups, and universal algebra, NATO Science. Series II. Mathematics, Physics and Chemistry. — 2005. — **207**. — P. 241–272.
30. *Hoare C. A. R.* An axiomatic basis for computer programming // Communications of the ACM. — 1969. — **12(10)** — P. 576–580, 583.
31. *Floyd R.* Assigning meanings to programs. In Mathematical Aspects of Computer Science // Proceedings of Symposia in Applied Mathematics. — American Mathematical Society, Providence, Rhode Island. — 1967. — **19.** — P. 19–32.
32. https://www.microsoft.com/en-us/research/pro-ject/boogie-an-intermediate-verification-language, 2008.
33. *Patrick Cousot.* Abstract interpretation based formal methods and future challenges // Informatics, 10 years back — 10 years ahead, R. Wilhelm, ed. // Lecture Notes in Computer Science. — 2001. — **2000**. — P. 138–156.
34. *Edmund M. Clarke and Qinsi Wang.* 32 years of model checking. International Andrei Ershov. Memorial Conference on Perspectives of System Informatics, PSI 2014: Perspectives of System Informatics. — 2014. — P. 26–40.

*Получено 19.06.2017*