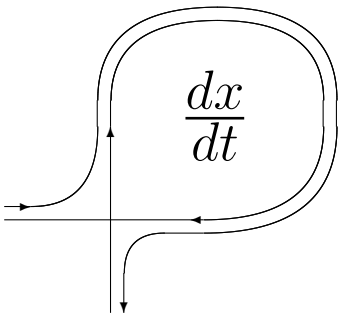


ДИФФЕРЕНЦИАЛЬНЫЕ УРАВНЕНИЯ
И
ПРОЦЕССЫ УПРАВЛЕНИЯ
N 1, 1998
Электронный журнал,
рег. N П23275 от 07.03.97

http://www.neva.ru/journal
e-mail: diff@osipenko.stu.neva.ru

Фильтрация и идентификация

Применение методов теории реактивных систем в задачах моделирования и качественного анализа непрерывно–дискретных систем.

Е.Ю. Парийская

Россия, 191187, Санкт-Петербург, Кутузовская наб., д.10
Институт теоретической астрономии РАН,
e-mail: epar@homesao.spb.su

Аннотация.

Статья представляет обзор и анализ современных методов качественного анализа и синтеза теории реактивных систем, которые могут быть использованы в задачах моделирования и анализа непрерывно–дискретных систем. Проведена классификация непрерывно–дискретных систем по признаку разрешимости существующих методов теории реактивных систем.

1. Введение

Непрерывно–дискретной системой называется параллельная и распределенная система, состоящая из взаимодействующих элементов различной природы, поведение которых описывается как непрерывными, так и дискретными процессами.

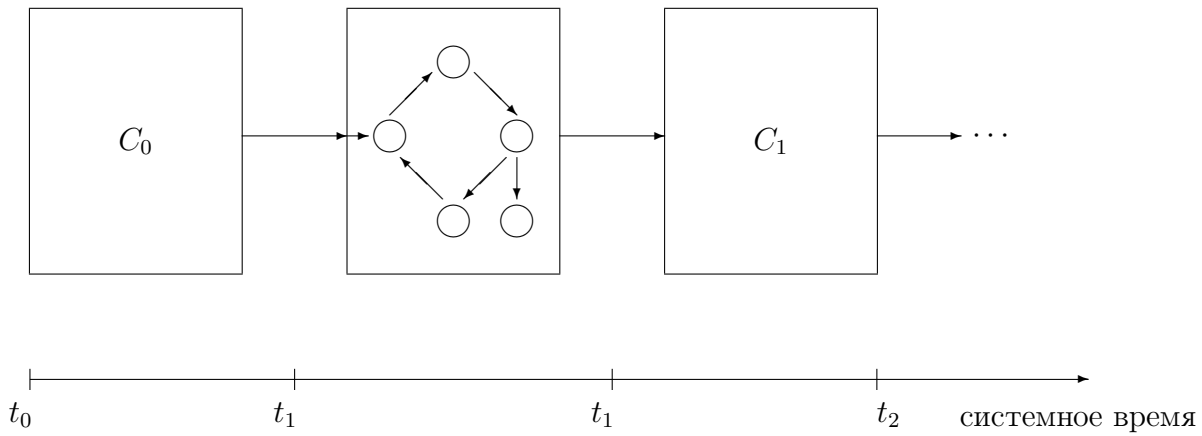


Рис. 1. Схема поведения непрерывно-дискретной системы, C_i – классические динамические системы.

Глобальное поведение непрерывно–дискретной системы описывается последовательностью локальных поведений, смена которых происходит под воздействием событий. Наступление того или иного события зависит от значений непрерывных параметров и, следовательно, от функций локальных поведений. С другой стороны, одно событие может порождать другие, и сам дискретный процесс, результатом которого является выбор нового локального поведения, описывается в общем случае нетривиальным дискретным алгоритмом.

Поведение непрерывно–дискретной системы можно, таким образом, представить бесконечной последовательностью сменяющих друг друга длительных непрерывных и мгновенных дискретных поведений (рис.1), а саму систему — графом смены поведений, в котором каждая вершина определяет поведение в текущий момент времени, а каждый переход — условия смены поведений [4],[6].

Примерами непрерывно-дискретных систем могут служить транспортные системы, системы слежения, надежные системы управления физическими процессами с аварийными ситуациями, системы, состоящие из параллельных взаимодействующих динамических подсистем.

Разработка программных комплексов, автоматизирующих процессы моделирования и анализа непрерывно–дискретных систем имеет уже более чем тридцатилетнюю историю[1] ([1],[2],[3],[9]). С появлением новых подходов к моделированию и анализу, в частности, технологии символьных вычислений ([28],[29]) и автоматических методов качественного анализа теории реактивных систем ([14],[30],[24]) в настоящее время открылись новые возможности для решения этой задачи.

Анализ методов моделирования, которые лежат в основе современных вычислительных комплексов, показывает, что сегодня применяются только два подхода к исследованию непрерывно–дискретных систем: численное моделирование отдельных траекторий и частичный качественный анализ свойств глобального поведения. Однако, непрерывная и дискретная компоненты входят в систему как равноправные, равнозначные части и пренебрежение или необоснованное упрощение поведения как одной, так и другой компоненты почти всегда приводит к неадекватности полученной вычислительной модели. Сейчас представляется очевидным, что задача автоматизации моделирования и анализа непрерывно–дискретных систем требует объединения в единую технологию технологии моделирования непрерывных процессов и технологии моделирования дискретных процессов. Такая технология была бы в данном случае наиболее адекватной и эффективной. Основная идея новой технологии — возможность единого описания непрерывной и дискретной компонент и предоставление пользователю возможности применения для каждой компоненты своих оптимальных методов моделирования и анализа [5].

Статья представляет обзор современных дискретных методов, разработанных в теории реактивных систем, которые могли бы быть включены в такую технологию.

Можно рассматривать две основные идеи использования дискретных методов в качестве инструмента исследования непрерывно–дискретных систем. Первая заключается в использовании классических алгоритмов автоматической верификации (”model checking”алгоритмов) дискретных систем [22],[14] при анализе дискретной компоненты и дискретных периодов развития системы. Этот анализ может заключаться, к примеру, в обнаружении циклов, решения проблем тупиков и в обнаружении недостижимых состояний в отдельно взятой дискретной компоненте, что значительно упростит

[1]термины ”смешанная”, ”агрегативная”, ”непрерывно–дискретная”, ”событийно–управляемая”, ”система переменной структуры”, ”гибридная”, как показано в [8], являются синонимами и вводились разными авторами для обозначения одного и того же класса систем

процедуру моделирования глобального поведения непрерывно–дискретных системы, для которой в целом может использоваться технология непрерывного моделирования.

Другая идея заключается в проведении оценок значений непрерывных переменных системы, участвующих в локальных поведениях, и в замене в математической модели непрерывно–дискретной системы соответствующих локальных поведений на их параматрические оценки с целью применения современных методов символьной верификации теории реактивных систем [23] для проверки качественных поведенческих свойств всей непрерывно–дискретной системы или некоторой ее части. Этот подход наиболее привлекателен для систем с дискретной частью произвольной сложности и с несложной непрерывной компонентой и может быть использован в новой технологии на начальном этапе проектирования и анализа систем.

Здесь мы ограничиваемся рассмотрением второй идеи, которая легла в основу "гибридного направления"(Hybrid Systems) — одного из современных направлений моделирования и анализа непрерывно-дискретных систем [9],[21].

Гибридное направление появилось в начале 1990-х годов на базе дискретного темпорального подхода, разработанного для систем реального времени в теории реактивных систем. С появлением метода символьной верификации для систем реального времени (1992 год, [22]), появилась надежда автоматизировать верификацию гибридных систем. В 1995-96 году создана система HyTech [27] автоматической верификации гибридных систем, основанная на символьной верификации.

Структура статьи следующая. Во второй главе представлена базовая математическая модель дискретной параллельной распределенной системы, основные средства описания поведенческих свойств и некоторые базовые понятия теории реактивных систем, на которых строится вся теория автоматической верификации. В третьей главе дано определение гибридной системы. Последняя часть посвящена обзору современных направлений верификации гибридных систем. Обсуждается проблема разрешимости различных методов для гибридных систем в целом и для отдельных подклассов, Представлен основной алгоритм символьной верификации гибридных систем и несколько простейших примеров, демонстрирующих возможность применения этого метода в задачах качественного анализа поведенческих свойств классических систем управления.

2. Основные понятия теории реактивных систем

Под реактивной системой понимается система, целью которой является поддержка взаимодействия с окружением. К классу реактивных систем относят операционные системы, мультипрограммные среды, параллельные и распределенные дискретные системы, дискретные системы реального времени, вычислительные сети.

Реактивную систему и ее окружение удобно представлять в виде параллельной системы, в которой обе эти компоненты функционируют параллельно, взаимодействуя друг с другом. Дискретный темпоральный подход, разработанный в теории реактивных систем для решения задач моделирования и анализа реактивных систем, основывается на предположении, что как сама система, так и ее окружение, моделируются как дискретные процессы. Это предположение удобно, так как позволяет выполнять симметричный анализ системы и ее окружения. Поэтому основным объектом исследования теории реактивных систем принято считать дискретную параллельную систему.

Элементами параллельной дискретной системы являются дискретные системы (называемые процессами), параллельно функционирующие во времени. Эти процессы могут быть независимыми друг от друга и конкурировать за общие ресурсы (асинхронные конкурентные системы), взаимодействовать между собой, выполняя общую задачу (кооперативные распределенные системы) и т.д. Глобальное поведение параллельной дискретной системы представляется множеством последовательностей смены ее глобальных состояний[2] во времени, которые можно привязать либо к событиям, вызывающим смену состояний, либо к тактам времени.

Можно сказать поэтому, что теория реактивных систем занимается изучением последовательностей состояний, порождаемых реактивной системой в процессе функционирования (анализом свойств этих последовательностей, синтезом "правильных"последовательностей, моделированием систем, которые порождают последовательности, обладающие заданными свойствами и т.п.). Понятие последовательности состояний, которое называется вычислением (сценарием, историей), является базовым понятием, на котором строится вся теория реактивных систем.

Основным свойством реактивной системы является отсутствие терми-

[2]глобальным состоянием параллельной дискретной системы называется кортеж, в который входят по одному состоянию для каждого параллельного процесса

нальности. В теории реактивных систем вводится понятие робастности, которое характеризует возможность безостановочной работы системы в любых ситуациях. Вторым важным свойством реактивной системы является то, что она является элементом параллельной системы и должна периодически обмениваться с окружением необходимой информацией. Поэтому важнейшими вопросами в теории реактивных систем являются вопросы параллелизма и синхронизации.

Теория реактивных систем занимается проблемами спецификации и верификации реактивных систем.

Под спецификацией понимают разработку математической модели реактивной системы. В качестве моделей реактивных систем обычно используются сети Петри [15], модель CSP Ч.Хоара [17], алгебра взаимодействующих процессов CCS Р.Милнера [18], системы переходов А.Пнуэли [19], временная алгебра процессов [16] и другие.

2.1. Системы переходов

Поведение произвольной дискретной системы может быть описано автоматом $M = \left\{Q, \Sigma, \delta, q_0, F\right\}$ с множеством состояний Q, выделенным начальным состоянием $q_0 \in Q$, множеством конечных состояний $F \subseteq Q$, входным алфавитом Σ (который называется также алфавитом событий) и заданной функцией переходов из состояния в состояние $\delta(q, a)$, определяющей следующее состояние по текущему состоянию q и полученному входному символу a и реализующей некоторое множество мгновенных действий.

В начале 1980-х годов для спецификациии параллельных и распределенных дискретных систем была предложена расширенная модель автомата, так называемая система переходов (А.Пнуэли). В отличие от автомата, основной функцией которого является распознавание конечных цепочек символов входного алфавита (то есть проверка возможности реализации автоматом конкретной цепочки), основной функцией системы переходов является порождение всевозможных (в том числе бесконечных) последовательностей состояний из заданного множества начальных состояний и при заданном множестве переходов. Эта модель легла в основу математической модели, разработанной для непрерывно–дискретных систем.

Определение 2.1. *Системой переходов называется следующая конструк-*

ция [7]:

$$S \;=\; \Big\{\; X\;,\;\; \Sigma\;,\;\; \mathcal{T}\;,\;\; \Theta\;\Big\}, \tag{1}$$

где

X – *множество переменных над областью определения* D

$\Sigma : \{\; X \to D\;\}$ – *множество состояний системы*

$\Theta \in \Sigma$ – *множество начальных состояний*

$\mathcal{T} \subseteq \{\Sigma \to \Sigma\}$ – *конечное множество переходов.*

Определение 2.2. *Вычислением системы переходов называется любая цепочка* $\sigma \in \Sigma^*, \sigma = s_0 s_1 \ldots s_n$ *(*n *произвольно), такая что*

a) $s_0 \in \Theta$

b) $(\forall\; i)(\exists\; !\tau \in \mathcal{T}) : \tau(s_i) = s_{i+1}$

Функционирование реактивной системы представляется множеством всех возможных вычислений ее системы переходов.

Системы переходов удобны при спецификации параллельных дискретных систем, так как параллельная композиция систем переходов будет также являться некоторой системой переходов, функционирование которой может быть представлено множеством ее вычислений. Часто совместное поведение параллельных процессов нагляднее и эффективнее представлять в виде дерева, в котором можно проследить пути развития каждого из них. Поэтому часто наряду с обычными вычислениями (которые называются в теории реактивных систем линейными вычислениями) используют так называемые деревья вычислений (computation tree, подробнее см. [12]).

Алгоритмы верификации, разработанные на основе модели системы переходов, позволяют исследовать свойства реактивной системы с точностью до порядка наступления событий[3] в системе, то есть проводить анализ качественных свойств ее поведения. Однако, во многих случаях (в системах реального времени) бывает необходимо учитывать конкретные моменты времени наступления событий и время выполнения отдельных операций.

[3]Понятие "события"здесь и далее не соответствует классическому "событию"как паре <состояние, время>. В теории реактивных систем оно заимствовано из теории автоматов и обозначает метку (букву алфавита) на некотором переходе. Порядок наступления событий в реактивной системе есть относительный порядок следования переходов (или состояний) в вычислении ее системы переходов (или порядок следования меток в допущенной цепочке автомата)

Имеется две альтернативные возможности ввести время в вычислительную модель реактивной системы. Одна заключается в учете длительностей нахождения в каждом дискретном состоянии и реализуется введением задержек срабатывания переходов в системе переходов. Другая заключается в отслеживании моментов наступления событий в реактивной системе и может быть реализована в модели таймированного автомата.

Определение 2.3. *Системой временных переходов называется тройка [7]:*

$$S^T \quad = \quad < \quad S, \quad L \ , \quad U \quad >, \tag{2}$$

где

S — система переходов (1), а L, U — множества нижних и верхних границ интервалов временных задержек срабатывания активизированных переходов $\tau \in \mathcal{T}$:

$L = \{l_\tau, \tau \in \mathcal{T}\}$

$U = \{u_\tau, \tau \in \mathcal{T}\}$

$0 \le l_\tau \le u_\tau, \quad l_\tau, u_\tau \in R$

Поведение системы временных переходов, фиксируемое одним последовательным наблюдателем представляется последовательностью событий двух типов: событие изменения времени при неизменном состоянии и последующего за ним одного или нескольких событий изменения состояний при фиксированном времени. Пример такого представления поведения реактивной системы, которое впервые было дано в [19], показан на рис.2.

Функционирование временной системы переходов характеризуется множеством всевозможных качественных вычислений (нетаймированных) и соответствующим множеством всевозможных "точечных"вычислений, у которых к каждому состоянию приписывается момент времени, в который система перешла в это состояние.

Определение 2.4. *Точечным вычислением временной системы переходов S^T называется цепочка $\rho_T = (s_0, t_0)(s_1, t_1) \dots (s_n, t_n) \in (\Sigma \times T)^*$, т.ч.*

а) $s_0 s_1 \dots s_n$ допустима в соответствующей системе переходов S, то есть $(\forall\ i)(\exists\ !\tau \in \mathcal{T}) : \tau(s_i) = s_{i+1}$

b) $\forall\ i < n : t_i \le t_{i+1}$ – все события происходят последовательно во времени

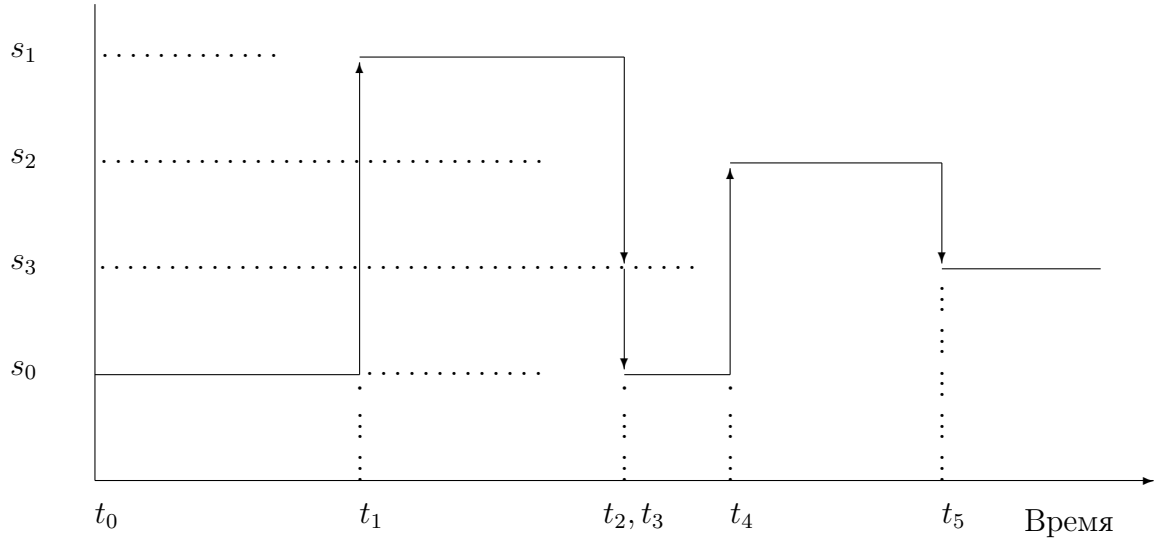


Рис. 2. Поведение системы переходов

c) $\forall \tau \in \mathcal{T}$ *и* $j \geq 0, \exists i \leq j$, *т.ч.* $t_i + l_\tau \leq t_j$ *и переход* τ *непрерывно возможен в состояниях* $s_i \ldots s_j$ *но еще не произошел*

d) $\forall \tau \in \mathcal{T}$ *и* $i \geq 0, \exists j, i \leq j$, *т.ч.* $t_i + u_\tau \geq t_j$ *и переход* τ *произошел в* j *или уже стал невозможен*

Каждому качественному вычислению системы временных переходов соответсвует множество (пучок) точечных вычислений.

Модель таймированного автомата представляет собой автомат, с котором каждому событию из алфавита событий автомата может быть поставлена в соответствие некоторая вещественная переменная x (часы, таймер), значение которой увеличивается в соответствии с увеличением реального времени (по линейному закону $dx/dt = 1$). На переходах автомата таймеры могут обнуляться. Таким образом, таймированный автомат может отслеживать абсолютное время наступления отдельных событий и относительное время между событиями одного или различных классов.

Определение 2.5. *Таймированным автоматом называется автомат вида [41]:*

$$A = < \Sigma, T, L, L_0, L_F, E > \qquad (3)$$

где

Σ — конечный входной алфавит (алфавит событий)

T — конечное множество таймеров

L — множество вершин автомата, $L_0, L_F \subseteq L$ — начальные и конечные вершины

E — множество дуг. Каждая дуга помечена буквой алфавита событий, условием перехода по дуге в виде линейных целочисленных неравенств над значениями таймеров и набором таймеров, которые должны быть обнулены на дуге.

Таймированный автомат распознает таймированные слова над алфавитом событий Σ — конечные цепочки $(a_0, t_0)(a_1, t_1) \dots (a_n, t_n), a_i \in \Sigma, t_i \in R^+$. Каждому слову входного алфавита, допущенному таймированным автоматом, соответствует множество (пучок) таймированных слов.

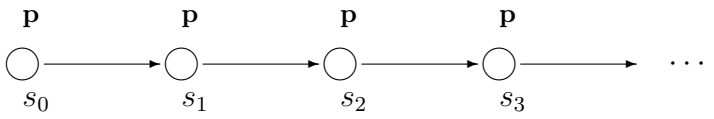
Можно показать, что модель системы временных переходов и модель таймированного автомата эквивалентны [41].

2.2. Темпоральная логика

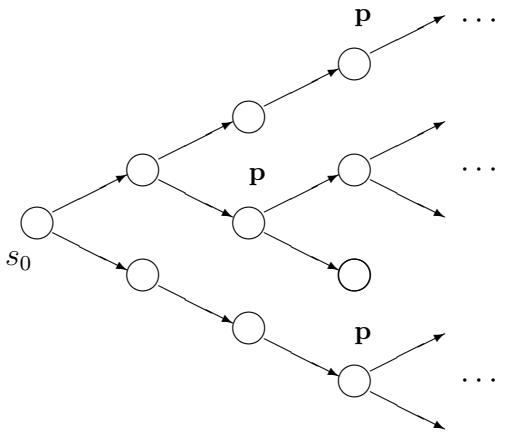
Теория реактивных систем занимается разработкой средств удобного и выразительного описания свойств, которыми должна обладать реактивная система, — спецификацией требований к поведению системы. К таким требованиям относятся принципиальная возможность или невозможность отдельных событий, порядок следования событий и причинно-следственные связи между событиями.

Типичными требованиями к параллельным системам являются, например, гарантия недопустимости определенных (аварийных) ситуаций в процессе функционирования системы, требование отсутствия тупиковых ситуаций при работе с общими ресурсами, требование взаимного исключения при вхождении в критическую секцию нескольких параллельных процессов, гарантия отсутствия бесконечного ожидания ресурса некоторым процессом, гарантия достижения некоторой цели (за определенное время) и т.д.

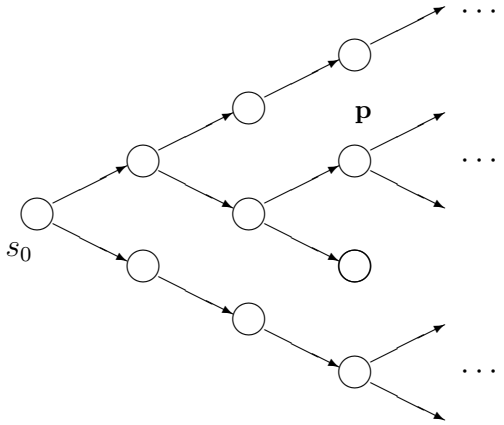
Требования к поведению делятся на два основных класса (классификация предложена Л.Лэмпортом [38],[39])— требования корректности (safety, гарантия того, что некоторое свойство сохраняется во всех состояниях всех вычислений системы) и требования жизнеспособности (liveness, гарантия того, что некоторое событие когда-то в будущем произойдет в системе,



а) требование корректности: $\Box p$



б) $\forall$-жизнеспособность: $\forall\Diamond p$



в) $\exists$-жизнеспособность: $\exists\Diamond p$

Рис. 3. Базовые требования к поведению реактивной системы.

или, другими словами, некоторое свойство будет истинно над некоторым достижимым состоянием системы). При представлении поведения системы деревьями вычислений требование жизнеспособности должно быть разделено на два подкласса — когда состояние, удовлетворяющее заданному свойству, достижимо на каждой ветви дерева вычислений системы ($\forall$-жизнеспособность) и когда оно достижимо хотя бы на одной ветви дерева вычислений ($\exists$-жизнеспособность) . Три вышеописанные базовые свойства поведения, которые исследует теория реактивных систем, представлены на рисунке 3.

Спецификацию требований к поведению системы удобно задавать формулами темпоральной логики[4].

Термами (атомами) темпоральной логики являются предикаты, которые на отдельных состояниях системы принимают значения "истина"или "ложь". Ими могут являться формулы счисления предикатов первого порядка, включающие стандартные операции отношения над целыми числа-

[4]Расширение математической логики, представляющее собой специальный математический аппарат для доказательства теорем верификации [11], [12].

ми. Темпоральная формула конструируется из термов с применением так называемых темпоральных операторов и описывает в виде функций и предикатов над множеством переменных системы свойства отдельного состояния, отдельного вычисления и множества (возможно бесконечного) вычислений в целом. Базовыми темпоральными операторами являются оператор $\bigcirc$("Next") и оператор $\mathcal{U}$("Until"). Одноместный оператор "Next"позволяет описать свойство непосредственно следующего состояния вычисления, двуместный оператор "Until"описывает длительность истинности некоторого свойства (первого операнда) относительно некоторого другого свойства (второго операнда). Для деревьев вычислений вводятся также два базовых оператора — $\exists$ ("для некоторого пути в дереве вычислений") и $\forall$ ("для всех путей в дереве вычислений").

Вводится отношение истинности "$\models$"формулы φ в состоянии i, вычисления σ длины $|\sigma|$ $(0 \le i < |\sigma|)$:

$$(\sigma, i) \models \varphi$$

и отношение истинности формулы над вычислением:

$$\sigma \models \varphi \;\Leftrightarrow\; (\sigma, 0) \models \varphi$$

Пусть q, p – термы, s_i – некоторое состояние траектории σ длины $|\sigma|$.

Семантика базовых темпоральных операторов следующая:

$\bigcirc$ (*Next*) $\quad (\sigma, i) \models \bigcirc q \;\Leftrightarrow\; i+1 < |\sigma| \wedge (\sigma, i+1) \models q$

$\mathcal{U}$ (*Until*) $\quad (\sigma, i) \models p\mathcal{U}q \;\Leftrightarrow\; \forall k(i \le k < |\sigma|) :$

$$\Big[(\sigma, k) \models p \vee \exists j(i \le j \le k) : (\sigma, j) \models q\Big]$$

Для деревьев вычислений семантика оператора $\mathcal{U}$ следующая:

$f_1 \exists\mathcal{U} f_2$ ($\exists$-*Until*) $\;\Leftrightarrow\; f_1\mathcal{U}f_2$ — по некоторому пути дерева вычислений

$f_1 \forall\mathcal{U} f_2]$ ($\forall$-*Until*) $\;\Leftrightarrow\; f_1\mathcal{U}f_2$ — по всем путям дерева вычислений

Три базовые свойства, изображенные на рисунке 3, описываются темпоральными формулами, использующими дополнительные темпоральные операторы:

$\Box$(*"Всегда"*)	$\Box p = p\mathcal{U}False$
$\forall\Diamond$(*"Когда-нибудь всюду"*)	$\forall\Diamond q = \neg\Box\neg q$— по всем путям
$\exists\Diamond$(*"Когда-нибудь где-то"*)	$\exists\Diamond q = \neg\Box\neg q$— по некоторому пути

Для спецификации требований к поведению систем реального времени используется темпоральная логика реального времени, в которой наступле-

ние любого события (и соответственно, базовые темпоральные операторы) имеет конкретные временные рамки, например:

$$\Diamond_{[l;u]}\ p$$

где $l, u \in R^{\geq}$ – нижняя и верхняя временные границы наступления события p.

2.3. Верификация реактивных систем

Среди задач моделирования и анализа, которыми занимается теория реактивных систем выделяются направление *верификации* (разработка аксиоматической базы и автоматических алгоритмов для проведения качественного анализа поведения реактивных систем), разработка *автоматических синтезирующих* алгоритмов (направление, которое соответствует теории абстрактной реализации в классической теории систем), разработка алгоритмов *построения замкнутого представления глобального поведения* для параллельных и распределенных систем и систем реального времени, а также разработка автоматических алгоритмов для *решения задач анализа и параметрического синтеза* в системах реального времени.

Верификацией называется доказательство истинности некоторого требования к поведению на некоторой вычислительной модели реактивной системы за ограниченное время. Верифицировать реактивные системы невозможно путем тестирования, так как количество возможных вариантов ее поведения может быть бесконечно велико.

• Теория реактивных систем предлагает несколько подходов к проблеме верификации:

— аксиоматический подход, который предполагает представление поведения системы в виде утверждений в некоторой логической системе и формулировну требований к поведению в виде некоторой теоремы, которая может быть доказана средствами темпоральной логики [11]

— разработка алгоритмов обхода системы переходов и проверки истинности темпоральной формулы в каждой ее вершине (классические "model checking"алгоритмы [14],[13]). Очевидно, размер верифицируемой реактивной системы при этом ограничен используемой оперативной памятью вычислительной машины.

— автоматические методы верификации, не требующие хранения в оперативной памяти вычислительной машины всей модели реактивной систе-

мы, а подразумевающие ее построение параллельно с процессом доказательства требования к поведению, сформулированого в виде темпоральной формулы ([30]).

• Направление разработки синтезирующих алгоритмов предполагает построение корректной реализации системы (структуры каждого процесса, участвующего в системе, и модели связей между процессами) по правильно сформулированной в виде формулы темпоральной логики спецификации требований к поведению. В классических работах по теории реактивных систем такие алгоритмы называются "decision procedures"(например, [13]).

• Алгоритмы параметрического синтеза для дискретных систем реального времени позволяют проводить оценку области временных параметров (задержек) в системе, гарантирующих заданное свойство ее поведения [7].

Специфицирующие формализмы и классические алгоритмы верификации реактивных систем можно найти, например, в работах [19],[20],[14].

3. Гибридные системы

Гибридное направление вслед за дискретными системами реального времени олицетворяет развитие идеи применения методов теории реактивных систем к непрерывным процессам реального мира.

Следует отметить, что поведение всякой динамической системы может быть аппроксимировано (с определенной степенью точности) кусочно-постоянной функцией и, следовательно, любая динамическая система может быть представлена системой переходов, которая является одной из базовых математичесмких моделей теории реактивных систем. Возможность построения системы переходов, отражающей качественное поведение динамической системы, говорит о том, что применение методов теории реактивных систем в задачах моделирования и анализа динамических систем, вообще говоря, имеет смысл.

Гибридной системой в теории реактивных систем называется система, которая в классических задачах моделирования и анализа называлась бы непрерывно–дискретной или событийно–управляемой, [4], [6].

К классу гибридных систем можно отнести сложные системы управления. В качестве примера можно привести систему космического слежения, состоящую из нескольких папаллельно работающих (и при необходимости

заменяющих друг друга) локаторов, каждый из которых является динамической подсистемой. В такой системе управления появление объекта в области видимости некоторого локатора является событием, определяющим его динамику на период слежения. Основным требованием к такой системе управления, очевидно, является надежность работы всей системы при возможных отказах отдельных подсистем и максимальное количество одновременно ведомых объектов. Произвольность траектории ведомого объекта и момента его появления, возможность отказов отдельных элементов делает систему событийно-управляемой и усложняет процесс ее моделирования и анализа. Аналогичным примером может служить система управления наблюдениями на радителескопах сложной структуры (например, на РАТАН-600), позволяющих наблюдать параллельно несколько небесных объектов и изменяющих программу наблюдений по наступлению разного рода событий (изменения погодных условий, наступления небесных событий, например затмения, отказа записывающих устройств).

Специфика подобных систем заключается с одной стороны, в сложности динамики каждого элемента системы (процесс слежения за объектом или наблюдательный эксперимент), а с другой — в возможности возникновения разного рода событий, которые по-разному в зависимости от момента их наступления мгновенно изменяют динамику элементов и структуру системы в целом.

Гибридными системами являются также транстпортные системы, например, система управления городскими перевозками или система управления сложной железно-дорожной развязкой. В таких системах могут возникать задачи разрешения тупиковых ситуаций (например, вследствие определенных правил городского движения), которые являются типичными дискретными задачами и которые разумнее анализировать дискретными методами, чем численным моделированием.

Более простым (с точки зрения законов динамики, но не с точки зрения моделирования) примером может служить так называемая задача о бильярде с несколькими шарами без трения (пример из [23]).

На рисунке 4 изображен упрощенный вариант игры с друмя шарами. Игра начинается с удара по шару "g", который находится в некоторой точке (x_0, y_0) бильярдного стола, и продолжается до столкновения со вторым неподвижным шаром "ω". В самом простом случае трение не учитывается и считается, что шар "g"движется по линейному закону. Событие удара шара о стенку стола изменяет траекторию его движения, не меняя скоро-

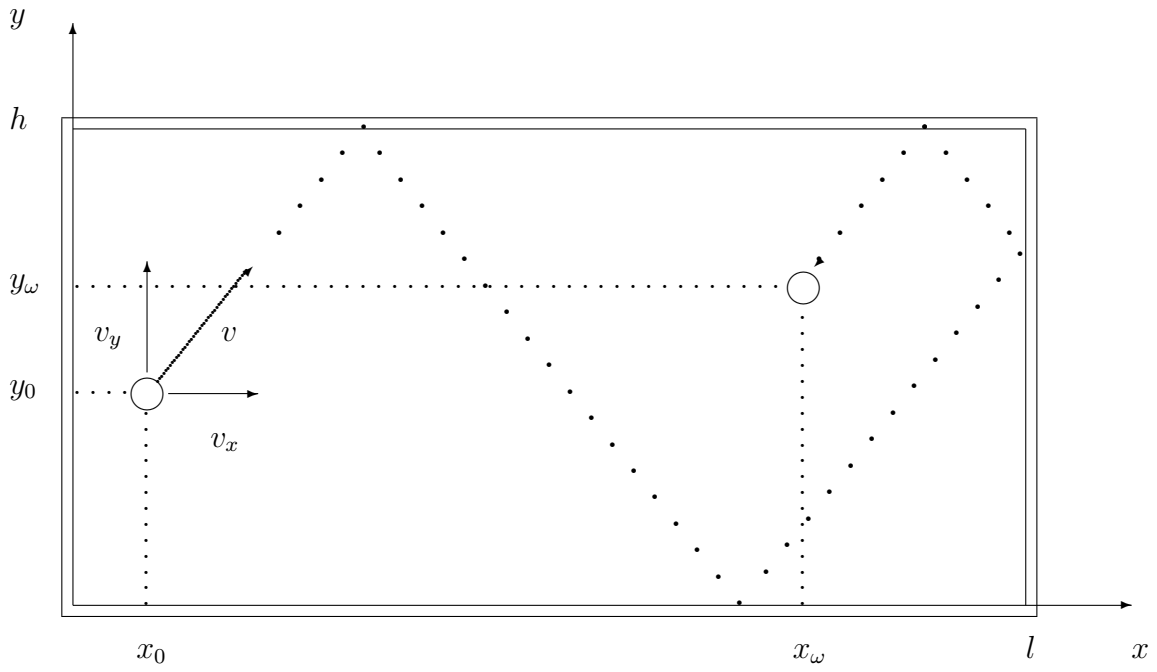


Рис. 4. Игра в бильярд с двумя шарами.

сти. Игрок задает начальную точку и скорость шара.

Гибридные системы обладают основными свойствами реактивных систем — отсутствием терминальности, наличием проблем параллелизма и синхронизации, поэтому кажется привлекательной попытка использования для этих систем методов теории реактивных систем.

В качестве математической модели гибридной системы используется *система фазовых переходов*, предложенная А.Пнуэли как расширение системы временных переходов. Системы фазовых переходов удобны как для спецификации гибридной системы, так и ее елементов. Для них, как и для систем переходов, верно утверждение, что переллельная их композиция будет также являться некоторой системой фазовых переходов.

Определение 3.1. *Системой фазовых переходов называется следующая система переходов [35]:*

$$S^P \quad = \quad < \quad X \ , \ S \ , \ \Theta \ , \ \mathcal{T} \quad >, \tag{4}$$

где

X — *конечное множество вещественных переменных*

S — *множество фаз, описанных набором локальных поведений*

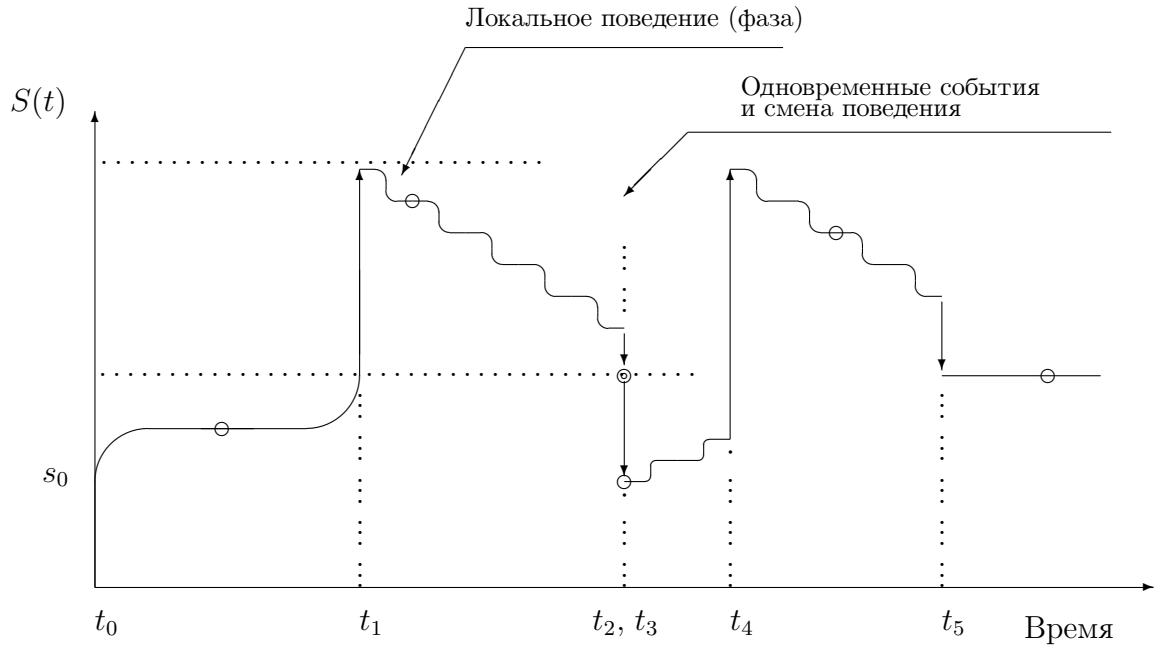


Рис. 5. Поведение гибридной системы

Θ — *множество начальных фаз, описанных некоторым предикатом над* X

$\mathcal{T}$ — *конечное множество переходов между фазами с определенными на них совокупностями мгновенных действий над переменными* X

Поведение системы фазовых переходов, фиксируемое одним последовательным наблюдателем, представляется последовательностью событий двух типов: событие увеличения времени внутри одной фазы с преобразованием переменных гибридной системы в соответствии с ее локальным поведением ("фазовый переход") и последующего за ним одного или нескольких событий дискретного изменения фазы при фиксированном времени с преобразованием переменных в соответствии с описанными мгновенными действиями. ("дискретный переход").

Пример такого поведения показан на рис.5.

Состоянием системы фазовых переходов называется пара $< x, t >$, $x \in X, t \in R$. Оно соответствует внутренней точке некоторой фазы.

Функционирование системы фазовых переходов, как и системы временных переходов, характеризуется множеством всевозможных качественных вычислений (нетаймированных) и соответствующим множеством всевозможных "точечных"вычислений. Каждому качественному вычислению со-

ответствует множество (пучок) точечных вычислений. В системе фазовых переходов допускаются бесконечные цепочки состояний.

Определение 3.2. *Точечным вычислением системы фазовых переходов* S^P *называется цепочка* $\rho_P = (s_0, t_0, x_0)(s_1, t_1, x_1) \ldots \in (\Sigma \times T \times D)^\omega$, *т.ч.*

a) $(\forall\ i)(\exists\ !\tau \in \mathcal{T}) : \tau(s_i) = s_{i+1}$

b) $\forall\ i < n : t_i \leq t_{i+1}$ – *все события происходят последовательно во времени*

c) каждый момент t_i *и значение* x_i *соответствуют конкретной точке фазы* i *фазовой траектории гибридной системы.*

На рисунке 5 точками изображено возможное точечное вычисление системы фазовых переходов.

Для спецификации требований к поведению гибридных систем может быть использована темпоральная логика реального времени или ее расширения (интервальная темпоральная логика [27], гибридная темпоральная логика [35]). В методах автоматической верификации часто используется темпоральная логика реального времени деревьев вычислений (Timed Computation Tree Logic), формулы которой имеют вид:

$$\delta\ ::=\ p\ \mid\ l \leq x \leq u\ \mid\ \neg\delta\ \mid\ \delta_1 \vee \delta_2\ \mid\ \delta_1 \exists \mathcal{U}_{\sharp n} \delta_2\ \mid\ \delta_1 \forall \mathcal{U}_{\sharp n} \delta_2$$

где p – терм, $x \in X,\ l, u \in R \cup \{\pm\infty\}, \sharp \in \{\ <\ ,\ \leq\ ,\ =\ ,\ >\ ,\ \geq\ \}$.

Примером типичного требования к поведению гибридной системы может служить спецификация требования в задаче о бильярде с двумя шарами: "запущенный в игру шар не может двигаться дольше периода T без столкновения с другим шаром", которое описывается темпоральной формулой:

$$\Diamond_{\leq T}\ (x = x_w \wedge y = y_w)$$

Такое требование порождает задачу верификации при конкретных значениях l, h, v_x, v_y, T и задачу параметрического синтеза T(в зависимости от начальных скоростей шаров и размеров бильярдного стола).

4. Верификация гибридных систем.

Для гибридных систем доказано, что порождаемый ими язык, не обладает регулярной (и даже ω–регулярной) структурой[5], и система не может быть

[5]Напомним, что язык называется регулярным, если он допускается конечным автоматом; язык называется ω-регулярным, если для его реализации может быть построен некоторый автомат, допускающий

представлена в общем случае конечным автоматом. Следовательно, методы верификации и процедуры синтеза программ теории реактивных систем, разработанные для дискретных систем, которые могут быть представлены в виде конечного графа состояний, не разрешимы в целом для гибридных систем и можно говорить только о применении этих методов к отдельным классам гибридных систем.

В этом заключается основная теоретическая проблема применения методов верификации к гибридным системам, которую мы условно будем называть проблемой разрешимости.

Исследования в области верификации гибридных систем последних лет можно условно разделить на три направления: выделение подклассов гибридных систем, у которых (бесконечное) множество состояний может быть разбито каким-либо образом на конечное число классов эквивалентности относительно верифицируемого требования к поведению (выделение подклассов требований) и применение к ним известных полностью разрешимых методов верификации реактивных систем (это направление условно назовем "реактивным направлением"); выделение подклассов гибридных систем и требований к поведению, для которых невозможно решить проблему достижимости, доказательство невозможности применения методов верификации для этих систем (условно "пессимистическое направление"); и, наконец, разработка полуразрешимых методов верификации для гибридных систем. К достижениям последнего направления (которое мы назвали бы "динамическим направлением") относятся появление гибридного автомата и гибридной темпоральной логики, пополнение дедуктивного аксиоматического подхода правилами вывода, отражающими динамические свойства гибридных систем[6], а также автоматический метод символьной верификации и метод интегрирующих графов. Два последних метода появились независимо в 1993-94 годах и представляют особый интерес, поскольку они отражают тенденции последних лет в области компьютерного моделирования гибридных систем.

Ниже представлен сначала краткий обзор первых двух направлений, а затем подробно рассмотрены автоматические методы динамического направления, которые выделены в отдельные подразделы.

бесконечные цепочки и обладающий свойством $Inf(\sigma) \cup F \neq 0$, где $Inf(\sigma) = \{s \in \Sigma, \text{т.ч.} \forall\, m \in N, \exists\, n > m : \sigma_n = s\}$ – множество вершин, посещаемых бесконечное число раз, F – терминальные вершины автомата (такой автомат называется Buchi-автоматом) [40].

[6]Расширение аксиоматической базы для гибридных автоматов и описание гибридной темпоральной логики (логики описания требований к поведению гибридных автоматов) можно найти в работах [21],[35].

В зависимости от методов верификации, которые используются в вышеописанных направлениях, предлагаются различные варианты представления математической модели системы фазовых переходов, в которых более конкретно определяются те или иные элементы гибридной системы (необходимые для применения соответствующих методов). Здесь нас интересует вариант модели системы фазовых переходов, используемый в автоматических методах верификации и называемый гибридным автоматом [9],[21].

Определение 4.1. *Гибридным автоматом называется следующая конструкция:*

$$H \;=\; \Big\{\; S\;,\;\; X\;,\;\; E\;,\;\; F\;,\;\; b\;\;,\;\; \psi\;\;,\;\; f\;\;\Big\}, \tag{5}$$

где

S *— конечное множество локаций (вершины автомата)*

X *— конечное множество вещественных переменных*

E *— конечное множество дуг. Любая дуга помечена условием перехода* b*, формулой преобразования переменных на дуге* f*, и буквой алфавита событий* $a \in \Sigma$

F *— оператор локального поведения внутри каждой локации (система дифференциальных включений, дифференциальных уравнений или набор вещественных непрерывных функций)*

ψ *— множество предикатов над* X*, описывающих области значений переменных* X *в каждой локации (предикаты–инварианты)*

В общем случае предикаты b, ψ имеют вид линейных неравенств:

$$\Sigma_{i=1}^{n} a_i x_i \sim c, \quad x_i \in X, a_i, c \in Q, \sim = \{<, >, =, \neq, \leq, \geq\} \tag{6}$$

Каждая локация автомата может представлять из себя гибридный автомат (вложенность), а параллельная композиция гибридных автоматов будет также являться гибридным автоматом.

Наиболее подходящими для применения к гибридным системам считаются две основные вычислительные модели систем реального времени — система временных переходов (опр. 2.3) и модель таймированного автомата (опр. 2.5). Для таймированного автомата существуют полностью

разрешимые методы атвоматической верификации свойств, описанных на языке темпоральной логики реального времени (алгоритмы символьной верификации [22]). Для системы временных переходов разработаны автоматические алгоритмы обхода для верификации свойств, описанных на языке линейной темпоральной логики [30], [14], процедуры автоматического синтеза системы переходов по спецификации системы, описанной на языке темпоральной логики деревьев траекторий [13], имеются алгоритмы построения замкнутого представления параллельной композиции временных систем переходов [7]. Система временных переходов используется также в аксиоматическом подходе [9].

Поэтому основной задачей реактивного направления является поиск возможных способов корректного приведения гибридного автомата к этим двум вычислительным моделям, для чего необходимо найти возможные способы перехода к конечному пространству состояний, при которых не нарушалась бы семантика верифицируемых свойств.

Заметим, что пространство состояний гибридного автомата может быть описано тройкой $\psi \times F \times R^N$, где ψ – множество инвариантов, F – множество его локальных поведений, а R^N – пространство векторов начальных значений переменных, описываемых оператором инициализации f. Поскольку ψ, F – конечные множества, то основной задачей является поиск конечного разбиения пространства векторов начальных значений переменных R^N.

Наиболее распостраненный способ перехода к конечному пространству состояний – это наложение сетки на ограниченном сверху и снизу по всем базисным компонентам подмножестве R^N.

Действительно, обозначив через L_i, U_i минимальное и максимальное целые значения компоненты x_i вектора переменных X, участвующие во всех предикатах гибридной системы, а через $\{<\}, \{>\}$ – множества всех значений компоненты, меньших L_i и больших U_i соответственно, можно наложить сетку на множество значений каждой компоненты и получить для каждой компоненты конечное множество состояний (шаг сетки Δ выбирается в соответствии с локальными поведениями таким образом, чтобы иметь целое число шагов для всех компонент вектора X во всех локациях автомата):

$$\Gamma_i = \{<\} \ \vee \ \{ \ L_i \ , \ L_i + \Delta \ , \ \ldots \ , \ U_i \ \} \ \vee \ \{>\}$$

Для гибридного автомата строится система переходов, в которой все состояния, характеризующиеся нахождением вектора X в одной клетке Δ-сетки, считаются эквивалентными и объединяются в одно состояние. Такая

система переходов будет иметь конечное число состояний

$$Q = \Psi \times \Gamma_1 \times \ldots \times \Gamma_n$$

Дискретизация фазового пространства этим способом возможна для гибридных автоматов, у которых:

a) локальные поведения описываются линейными дифференциальными уравнениями или включениями первого порядка с целочисленными коэффициентами

b) предикаты b и ψ имеют вид:

$$x_i \sim c \ \mid \ \neg\delta \ \mid \ \delta_1 \vee \delta_2, \quad \sim \in \{\leq, \geq\}, \quad c \in Z$$

(возможно сравнение значений компонент с целыми числами)

c) оператор преобразования переменных f описывается целочисленными интервалами:

$$f = \{ \ x := \lambda \in [\alpha; \beta], \ \alpha, \beta \in Z \ \}$$

Таким образом, для этого подкласса гибридных систем проблема верификации разрешима (подробнее см [31],[32]).

Другой способ конечного разбиения пространства состояний гибридного автомата иллюстрирует определение отношения эквивалентности на пространстве R^N, предложенное в работе [36] Предикаты гибридного автомата в этом случае могут иметь более сложный вид:

$$x_i \sim c \ \mid \ x_i - x_j \sim c \ \mid \ \neg\delta \ \mid \ \delta_1 \vee \delta_2, \quad \sim \in \{\leq, \geq\}, \quad c \in Z$$

(возможно сравнение значений двух компонент друг с другом)

Пусть $L_i, U_i \in Z$ — минимальное и максимальное значения непрерывой компоненты x_i, участвующие в предикатах системы.

Говорят, что два вектора $x, x' \in R^N$ эквивалентны, и пишут $x \approx x'$, если у них:

- присутствуют недостижимые снизу компоненты

  $\exists i, j, x_i < L_i, x'_j < L_j$

- или равные целые части и одинаковый относительный порядок дробных частей:

  $\forall i, x_i \geq L_i, \forall i, j$:

  либо $\quad x_i > U_i \wedge x'_i > U_i$

  либо $\quad x_i \leq U_i \wedge x'_i \leq U_i \wedge int(x_i) = int(x'_i) \wedge$

  $fract(x_i) \leq fract(x_j)$ если и только если $fract(x'_i) \leq fract(x'_j)$

Класс эквивалентности для вектора x обозначается как $[x]$ и называется "timer region", а получаемый таким образом новый автомат называется Timer Region Automaton. Введенное отношение эквивалентности, очевидно, порождает конечное разбиение R^N.

Построение Timer Region Automaton возможно для узкого класса гибридных систем, которые называются условно "зависающим автоматом"[7]. Для этого подкласса проблема верификации также разрешима.

Отдельно в реактивном направлении выделяется класс линейных гибридных автоматов, к которым возможно применение методов верификации таймированного автомата.

Определение 4.2. *Гибридный автомат, у которого предикаты b, ψ имеют вид линейных неравенств, оператор преобразования переменных на дугах f представляется целочисленным интервалом $\{\ x := \lambda \in [\alpha; \beta],\ \alpha, \beta \in Z \cup \{\pm\infty\}\ \}$ и переменные X меняются в каждой локации s по закону $dX/dt = \Lambda t$, где $\Lambda \in [A_s, B_s], A_s, B_s \in Z$ называется линейным гибридным автоматом.*

Определение 4.3. *Линейный гибридный автомат называется линейным гибридным атвтоматом с постоянными коэффициентами (Constant Slope Hybrid System), если $A_s = B_s$ для любой локации $s \in S$.*

Линейные гибридные автоматы с постоянными коэффициентами при наложении определенных ограничений[8], приводятся к модели таймированного автомата с помощью двойного преобразования ([32]):

Для $\forall s \in S$ определяется векторная функция k_s:

$$k_s(X) \ = \ \frac{K}{A_s}\ X \ + \ C_s, \text{ где}$$

[7]Зависающим автоматом называется гибридный автомат, у которого:

a) если R^s – вектор производных множества переменных X в вершине s, то любая его компонента $r_i^s = \{0, 1\}$

b) оператор преобразования переменных f имеет вид

$f = \{\ x_i := 0; x_i := x_i - c, c \in Z \cup \{\pm\infty\}\ \}$

c) разрешается использование предикатов вида:

$x_i \sim c \ \mid \ x_i - x_j \sim c \ \mid \ \neg\delta \ \mid \ \delta_1 \vee \delta_2$, где $\sim \in \{<, >, =, \leq, \geq\}, c \in Z$

d) для каждой компоненты $x_i \in X$ существуют абсолютные значения нижней и верхней границы ее изменения K_i^1, K_i^2, такие что как только $x_i \geq K_i^2$ $(x_i \leq K_i^1)$, так x_i уже больше не уменьшается (не увеличивается) и не входит ни в какие предикаты вида $x_i - x_j \sim c$

[8]а именно: 1) при детерминированности начальных значений X в каждой локации системы; 2) $\forall x \in X, \forall s \in S \ : \ x^{\cdot} \neq 0$, то есть A_s, B_s одного знака; 3)$\forall s \in S$ if $x^{\cdot} = A_s(x) > 0 \rightarrow l_x$ - конечно, if $x^{\cdot} = A_s(x) < 0 \rightarrow u_x$ - конечно .

$K = LCM\{abs(A_s(x)) \mid s \in S, x \in X\}$ – наименьший общий множитель (масштабирование по x)

и вектор C_s (сдвиг осей, чтобы в любой локации было бы $\alpha, \beta, l, u \geq 0$):

$$C_s(x) = max\{abs(l_x), abs(u_x), abs(\alpha), abs(\beta)\}\left\{\frac{K}{A_s(x)}\right\}$$

Затем производится замена переменных $X' = k_s(X)$

$x' = \frac{K}{A_s}x + C_s = \frac{K}{A_s}(x_0 + A_s t) + C_s = x'_0 + Kt$, где $x'_0 = \frac{K}{A_s}x_0 + C_s$

и всех значений α, β, l, u, фигурирующих в предикатах и начальных условиях системы $l \to l',\ u \to u',\ \alpha \to \alpha',\ \beta \to \beta'$,

$$l' = \frac{K}{A_s(x)}l + C_s(x) \ < \ x' \ < \ \frac{K}{A_s(x)}u + C_s(x) = u'$$

для $A_s > 0$ ($A_s < 0 - u' < x' < l'$)

$$x' := [\alpha', \beta'] \ = \ [\frac{K}{A_s(x)}\alpha + C_s(x); \frac{K}{A_s(x)}\beta + C_s(x)]$$

для $A_s > 0$ ($A_s < 0 - x' := [\beta'; \alpha']$)

Сконструированный таким образом гибридый автомат H_K обладает двумя важными новыми свойствами: 1) все переменные ≥ 0; 2) $\forall s \in S$: $dX/dt = K$ – одинаковый закон изменения. Такой автомат называется "K-таймированным автоматом.

Формулы темпоральной логики реального времени деревьев траекторий, описывающие требования к поведению, также необходимо преобразовать:

$$\delta \to |\delta|_K, \text{ где } x \sim c \quad \to \quad x \sim c'$$

Для перехода "Ктаймированного автомата к таймированному автомату ($H_K \to H_1$) достаточно изменить масштаб времени, то есть преобразовать законы изменения переменных:

$$X = X_0 + Kt \qquad \to \qquad X = X_0 + t$$

и все временные ограничения в темпоральных формулах $|\delta|_K$:

$$\sharp n \to \sharp Kn \quad (\text{соответственно } |\delta|_K \quad \to \quad ||\delta|_K|_{\frac{1}{K}})$$

Можно показать, что $\forall H \in CSHS_{\neq 0}$ и $\delta \in TCTL$ будет:

$$H \models \delta \quad \text{iff} \quad H_K \models |\delta|_K.$$

и для $\forall H \in CSHS_{\neq 0}$ и $\delta \in TCTL$:

$$H \models \delta \quad \text{iff} \quad H_1 \models ||\delta|_K|_{\frac{1}{K}}.$$

Следует отметить, что для линейного гибридного автомата общего вида невозможно построить эквивалентный H_K и, следовательно, методы автоматической верификации таймированного автомата к нему в целом не подходят. Однако, при наличии определенных условий, для него возможно построение модели, подобной H_K. При этом возможна верификация ме-

тодами таймированного автомата подмножества "неизбежных"($\forall$) формул логики реального времени деревьев траекторий, причем гарантируется доказательство только свойств системы, имеющих результат верификации "истина".

Работы, которые мы отнесли к пессимистическому направлению, посвящены доказательству неразрешимости задачи верификации для отдельных классов гибридных систем. Доказательство основывается на невозможности для определенных подклассов гибридных автоматов положительного решения проблемы достижимости:

Определение 4.4. *Проблемой P(oint)-достижимости в динамической системе называется следующая задача: "для двух точек* $\mathtt{x,x'} \in X$ *существует ли траектория* ξ, *т.ч.* $\xi(0) = \mathtt{x}$ *и* $\xi(t) = \mathtt{x}'$*?"*

Проблемой R(egion)-достижимости называется следующая задача: "для двух выпуклых многогранников $P, P' \subseteq X$ *существуют ли такие точки* $\mathtt{x} \in P, \mathtt{x}' \in P'$, *для которых проблема P-достижимости была бы разрешена положительно?".*

В работе [33] приводится доказательство эквивалентности 2-х-стекового автомата (и, следовательно, любой машины Тьюринга) и гибридного автомата с кусочно-постоянными производными[9] размерности 3. Это означает, что проблема достижимости эквивалентна проблеме остановки машины Тьюринга и не может быть решена для этого класса гибридных автоматов с более чем двумя переменными[10] .

Аналогично, эквивалентность N-счетной машины, для которой задача остановки неразрешима, и линейного гибридного автомата с постоянными коэффициентами, имеющего более 2 переменных, доказывает неразрешимость проблемы достижимости и для этого класса гибридных автоматов (причем даже для тех, в которых присутствуют самые простые предикаты $x = 0$ и $x \neq 0$). Доказательство эквивалентности можно найти в работе [25].

[9]Гибридный автомат с кусочно-постоянными производными есть модель динамической системы, у которой траектории являются ломаными линиями, то есть являются решением уравнения $\frac{d\mathtt{x}}{dt} = f(X)$, где $f : X \to X$ – (возможно частичная) фукнция, такая что $f = \{\mathtt{c}\}, \mathtt{c} \in X$ – множество векторов и $\forall \mathtt{c}$, $f^{-1}(\mathtt{c})$ есть конечное объединение выпуклых многогранников.

[10]автоматы с кусочно-постоянными производными размерности 2 представляют собой модели планарных систем, для которых проблема достижимости полностью решена, подробно см. в [33]

4.1. Символьная верификация для гибридных систем

Метод символьной верификации был предложен в 1992 году [20] в качестве автоматического алгоритма верификации для дискретных систем реального времени, специфицированных вычислителной моделью таймированного автомата. Разработанный на базе технологии символьных вычислений, он оказался очень привлекательным для гибридного направления. В работах по верификации систем реального времени он фигурирует под названием "алгоритм Кроноса"[22].

Основной идеей метода символьной верификации является отказ от времени как особой, первостепенной переменной, определяющей события смены состояний в реактивной системе. Время уходит из точечных вычислений и становится простой переменной, которая наравне с прочими может участвовать в предикатах и в требованиях к поведению системы.

Метод символьной верификации содержит алгоритм, с помощью которого можно получить (в виде некоторого предиката над переменными X) область пространства состояний системы переходов, достижимых из множества ее начальных состояний (описанных некоторым предикатом над X) в процессе функционирования или же определить (в виде предиката над X) множество начальных состояний, приводящих систему переходов в требуемые состояния (заданные предикатом над X).

Длительность фаз в системе фазовых переходов описывается новым типом предиката, введенного в определение гибридного автомата вместо инварианта локации ψ, — так называемого "предиката возможности увеличения времени в локации, time can progress predicate"(tcp-предикат). Он оказывается удобным для спецификации различных политик синхронизации в гибридной системе (подробнее см ниже или в [23]) и необходим для преобразования гибридных требований в требования реального времени с целью применения метода символьной верификации.

Гибридный автомат (см. опр. 4.1) при таком подходе называется tcp-автоматом, в нем инварианты ψ заменены на предикаты вида

$\mathtt{tcp}$: $S \times [D \times R^{\geq}] \to \{true; false\}$ — *предикат возможности локального увеличения времени*, обладающие свойствами:

1. $\mathtt{tcp}_s[X_0](0) = true$;

2. $\mathtt{tcp}_s[X_0](t) \Rightarrow \forall t' \leq t\ \mathtt{tcp}_s[X_0](t') = true$

($\mathtt{tcp}_s[X_0](t)$ означает, что если при входе в локацию s значение перемен-

ных было X_0, то в этой локации возможно находиться t единиц времени) а локальные поведения F описываются аналитическими функциями (*функциями эволюции*) $\varphi : S \times [D \times R^{\geq}] \to D$, которые обладают свойствами:

3. $\varphi_s[X_0](0) = X_0$;

4. $\varphi_s[X_0](t + t') = \varphi_s[\varphi_s[X_0](t)](t')$.

Важно заметить, что определение tcp-автомата и метод символьной верификации оперируют с абстрактным предикатом `tcp`, не уточняя его конкретный вид. Тем самым достигается универсальность метода для систем с различной политикой синхронизации параллельных процессов в системе, так как предикат `tcp` определяет эту политику. Действительно, пусть $P = \bigvee_i b_i (b_i$ – предикаты на дугах автомата, выходящих из локации $s)$ и пусть $\varphi_s(X_0)(t)$ – значение функции локального поведения в локации s в момент t при начальных условиях X_0.

Варианты предиката $\mathtt{tcp}_s(X)(t)$:

$$\mathtt{tcp}_s(X)(t) \equiv \exists t' \geq t \;\; P(\varphi_s(X_0)(t')) \vee (\forall t' \neg P(\varphi_s(X_0)(t'))) \tag{7}$$

— истинен максивально возможное время пребывания в локации s,"пока хотя бы один переход еще возможен"(асинхронная политика)

$$\mathtt{tcp}_s(X)(t) \equiv \forall t' \leq t \;\; \neg P(\varphi_s(X_0)(t')) \tag{8}$$

— истинен минимально возможное время пребывания в локации s, "до первого возможного перехода"(синхронизирующая политика)

$$\mathtt{tcp}_s(X)(t) \equiv \bigwedge_i \forall t' \leq t \;\; b_i(\varphi_s(X_0)(t')) \vee (\forall t'' \leq t' \neg b_i(\varphi_s(X_0)(t''))) \tag{9}$$

— истинен до первого отказа какого-нибудь перехода из локации s, "пока все еще возможны"(intermediate политика)

Поведение tcp-автомата, фиксируемого одним последовательным наблюдателем, представляется последовательностью чередующихся переходов двух типов:

мгновенное изменение значений переменных и дискретный переход по дуге `tcp`-автомата

$$\frac{s \ \rightarrow^{a,b,f} \ s',\ b(X_0)}{(s, X_0 > \ \rightarrow^{a} \ (s', f(X_0)} \tag{10}$$

изменение значения переменных внутри локации s с увеличением времени на t:

$$\frac{\mathtt{tcp}_s[X_0](t)}{(s, X_0 > \ \rightarrow^{t} \ (s, \varphi_s[X_0](t))} \tag{11}$$

Вычислением tcp-автомата является последовательность пар

$(s_0, X_0)(s_1, X_1) \ldots (s_n, X_n) \ldots$, т.ч.

a) $(s_i, X_i > \rightarrow^{t_i} (s_i, X_i' > \rightarrow^{a_i} (s_{i+1}, X_{i+1})$ или

b) $(s_i, X_i > \rightarrow^{a_i} (s_{i+1}, X_{i+1})$

Для верификации некоторого требования корректности, описанного с помощью формулы темпоральной логики, необходимо определить систему переходов для исследуемой гибридной системы, применить к ней алгоритм построения множества достижимых состояний и далее решить систему линейных неравенств, состоящую из неравенств, входящих в отрицание предиката требования корректности и в предикат достижимых состояний [прямой метод]. Алгоритм построения начальных состояний, приводящих систему переходов в требуемые состояния позволяет решать некоторые задачи параметрического синтеза гибридных систем [обратный метод].

Пусть P некоторый (не темпоральный) предикат над X, $s \in S$ – локация tcp-автомата.

Вводятся два новых предиката:

$$\overrightarrow{P}^s(X) \ \equiv \ \exists X_0, \exists t \ P(X_0) \ \wedge \ \mathtt{tcp}_s[X_0](t) \ \wedge \ \varphi_s[X_0](t)$$

$\overrightarrow{P}^s(X)$ характеризует множество состояний, достижимых из состояния, на котором истинен предикат P, на момент ухода из локации s.

$$\mathtt{post}_e[P](X) \ \equiv \ \exists X_0 \ P(X_0) \ \wedge \ b(X_0) \ \wedge \ X = f(X_0)$$

$\mathtt{post}_e[P](X)$ характеризует множество состояний, достижимых из состояния, на котором истинен предикат P, при дискретном переходе по дуге e tcp-автомата.

Для любого предиката P_0 можно составить *символьное* вычисление системы переходов:

$$(s_0, P_0)(s_1, P_1) \ldots (s_i, P_i) \ldots$$

где для $\forall i \geq 0$ существует дуга $e_i : s_i \to s_{i+1}$ и $P_{i+1} = \mathtt{post}_{e_i}[\overrightarrow{P_i}^{s_i}]$.

Это значит, что возможно проследить последовательное "развитие"предиката, определенного для некоторой локации на начинающемся в этой локации вычислении.

Пусть $I = (I_1, \ldots, I_m)$ — набор предикатов, характеризующих начальные состояния для каждой локации. (для локаций, которые являются начальными, I_i – начальный предикат над X, для всех других, $I_i \equiv false$).

Теорема 4.1.1. Итерационный алгоритм прямого метода. *Множество глобальных состояний гибридного автомата, достижимых из I , характеризуется набором предикатов $P = (P_1, \ldots, P_m)$, где каждый P_i соответствует локации s_i и является наименьшей устойчивой формулой следующего итерационного процесса:*

$$P_k = \overrightarrow{I_k \vee \bigvee_j \mathtt{post}_e[P_j]}^{s_k} \tag{12}$$

Этот метод называется прямым методом символьной верификации и может быть использован для решения проблем достижимости и для доказательства требований корректности в гибридных системах, специфицированных с помощью формул счисления предикатов первого порядка. Требование корректности P будет выполняться, если предикат $\neg P$ не достижим из множества начальных предикатов. То есть, если S – устойчивое решение описанных выше уравнений, то верификация требования корректности сводится к доказательству, что $S \wedge \neg P = false$.

Обратный метод предполагает введение двух предикатов, аналогичных введенным для прямого метода:

$$\overleftarrow{P}^s(X) = \exists t \mathtt{tcp}_s(X)(t) \wedge P(\varphi_s(X)(t))$$

$\overleftarrow{P}^s(X)$ характеризует множество состояний, из которых достижимо текущее состояние P внутри локации s.

$$\mathtt{pre}_e P(X) = b(X) \wedge P(f(X))$$

$\mathtt{pre}_e[P](X)$ характеризует множество состояний, из которых достижимо текущее состояние P при дискретном переходе по дуге e.

Любой (не темпоральный) предикат P, который предполагается верифицировать, можно представить набором предикатов $(P_1, \ldots, P_m)$, каждый из которых соответствует своей локации гибридного автомата.

Теорема 4.1.2. Итерационный алгоритм обратного метода. *Для множества глобальных состояний гибридного автомата, описанных некоторым предикатом* $R = \{R_1, \ldots, R_m\}$, *(*R_i *— целевой предикат*[11] *локации* s_i*), множество предшествующих состояний, из которых достижимы состояния* R*, характеризуется предикатом* $S = \bigvee_{k=1}^{m} P_k$*, где каждый* P_i *соответствует локации* s_i *и является наименьшей устойчивой формулой следующего итерационного процесса:*

$$P_k = \overleftarrow{R_k \vee \bigvee_j \mathtt{pre}_e[P_j]^{s_k}} \tag{13}$$

Требование корректности P будет выполняться на гибридном автомате, если предикат $R = \neg P$ достижим из множества состояний, описанных предикатом $S = \bigvee_{k=1}^{m} P^k$, и множество начальных состояний гибридного автомата I не пересекается с этим множеством. Таким образом, верификация требования корректности сводится к доказательству, что $S \wedge I = false$, то есть к решению некоторой системы линейных неравенств.

Метод символьной верификации может быть реализован как model-checking алгоритм [23], если ввести в темпоральную логику бинарный оператор "next"$\triangleright$, характеризующий темпоральные производные предикатов на множестве непосредственных наследников текущего состояния:

$$(P' \triangleright P)(X) \equiv \exists e, \exists t(\mathtt{pre}_e[P] \vee P)(\varphi_s[X](t)) \wedge$$
$$(\mathtt{tcp}_l[X](t) \wedge \forall t' \leq t(P' \vee P)(\varphi_l[X](t'))$$

где P, P' – темпоральные формулы, предикат $(P' \triangleright P)(X)$ описывает множество состояний, из которых можно попасть в состояние $P(X)$ за один шаг (путешествуя в некоторой локации и затем по дискретному переходу).

Ниже приведены итерационные правила вычисления достижимых состояний гибридного автомата для темпоральных предикатов, использующие характеристические множества термов, полученные прямым методом [здесь $\sim \in \{<, \leq, =, \geq, >\}$].

Характеристическое множество формулы $\phi_1 \exists \mathcal{U}_{\sim n} \phi_2$ есть $\bigvee_i P_i(0)$, где:

- $P_0(z) \equiv z \sim n \wedge \phi_2$
- $\forall i \geq 0,\ P_{i+1}(z) = P_i(z) \vee \phi_1 \triangleright P_i(z)$

Характеристическое множество формулы $\Box\phi$ есть $\bigwedge_i P_i$, где:

[11] например, если проверяется достижимость локации i, то $R_i \equiv l = i, R_j \equiv false$

- $P_0 \equiv \phi$
- $\forall i \geq 0,\ P_{i+1} = P_i \wedge \neg(\text{true} \triangleright \neg P_i)$

Характеристическое множество формулы $\forall\Diamond_{\sim n}\phi$ есть $\neg \bigvee_i P_i(0)$, где:

- $P_0(z) \equiv z \sim n$
- $\forall i \geq 0,\ P_{i+1}(z) = P_i(z) \vee \neg\phi \triangleright P_i(z)$

Автоматическая символьная верификация используется для гибридных автоматов с линейными предикатами общего вида

$$\bigvee_k \bigwedge_j \Sigma_i a_{ij} x_i \sim c, \quad \sim \in \{<, \leq, =, \geq, >\},\ a_i, c \in Q \tag{14}$$

однако, сходимость алгоритма гарантируется только для некоторых классов гибридных систем (см. сводную таблицу и *пессимистическое направление*).

Рассмотрим два примера символьной верификации требований к поведению систем автоматического управления.

Пример 1. Система газовой безопасности (прямой метод символьной верификации)

Система Gas Burner, используемая в качестве примера во многих работах гибридного направления, представляет из себя тривиальную "игрушечную"с точки зрения теории управления модель динамической системы, которая, однако, оказывается удобной для "ручной"демонстрации различных алгоритмов верификации (в частности прямого метода символьной верификации). Пример примечателен тем, что для него итерационный алгоритм символьной верификации не сходится, однако он дает итерационную формулу для всех достижимых состояний, с помощью которой можно доказать необходимое требование.

Система состоит из газового хранилища (танкера), кнопки запроса на пользование газом и системы контроля, которая может при необходимости прекратить или не разрешить очередной сеанс пользования газом (при помощи клапана).

Система может находиться в двух основных состояниях: leak (клапан открыт) и ¬ leak (клапан закрыт). При открытом клапане некоторое количество газа утекает, причем скорость утечки неизвестна, поэтому можно

контролировать только время пользования газом. Целью системы контроля является недопущение критической концентрации газа в окружающей среде. Для этого вводятся два ограничения на сеансы пользования газом:

1. каждый сеанс не должен продолжаться дольше n единицы времени

2. интервал времени между двумя последовательными сеансами должен быть не менее m единиц времени.

Эти ограничения определяют моменты наступления событий перехода из одного состояния системы в другое. Требование к поведению системы Gas Burner формулируется следующим образом (требование корректности):

"Суммарная длительность сеансов пользования газом для интервалов наблюдения больших некоторого времени (k единиц времени) не должна превышать 5 процентов общего времени, независимо от начала наблюдения."

Прямой метод позволяет верифицировать требование корректности при конкретных значениях параматров n, m, k. Пусть, например, $n = 1, m = 30, k = 60$. Метод состоит из следующих шагов:

- Построение гибридного автомата (рис.6). Локации 1 и 2 соответствуют основным состояниям системы, условия переходов по дугам соответствуют ограничениям 1) и 2) на поведения системы. Для верификации требования в систему вводятся три вещественные переменные, отслеживающие суммарную длительность сеансов пользования (переменная y), суммарное время между сеансами (переменная z) и локальное время нахождения в каждой локации (переменная x).

- Вычисление для каждой локации `tcp`-предиката по одной из формул (например, по формуле (7)):

$$\mathtt{tcp}_1(X)(t) \equiv \exists t' \geq t \quad x_0 + t \leq 1 \vee x_0 > 1 = x_0 + t \leq 1$$
$$\mathtt{tcp}_2(X)(t) \equiv \exists t' \geq t \quad x_0 + t \geq 30 = true$$

- Описание требования корректности (не темпоральным) предикатом:

$$y + z > 60 \Rightarrow 20y \leq y + z \tag{15}$$

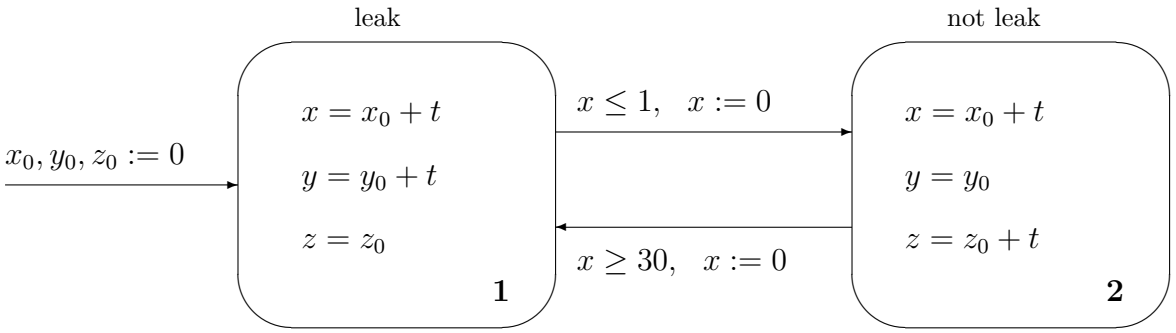


Рис. 6. Гибридный автомат системы контроля газовой безопасности.

• Построение достижимых состояний в виде предиката $P = (P^1, P^2)$ (P^1 описывает достижимые из локации 1 состояния, P^2 — из локации 2) для начального предиката $I = (I^1, I^2), I^1 \equiv x = y = z = 0, I^2 \equiv false$ по итерационной формуле (12):

$$P_0^1 = \overrightarrow{I^1}^1 \equiv \exists x_0, y_0, z_0, \exists t$$
$$(x_0 = y_0 = z_0 = 0 \wedge x_0 + t \le 1 \wedge x = x_0 + t \wedge y = y_0 + t) =$$
$$z = 0 \wedge x = t \wedge y = x \wedge t \le 1 = z = 0 \wedge x \le 1 \wedge y - x = 0 \wedge y \le 1$$

$$P_0^2 \equiv false$$

формула преобразования i→ i+1

$$P_i^1 = P_{i-1}^1 \vee \overrightarrow{\mathtt{post}_{21} P_{i-1}^2}^1$$
$$P_i^2 = P_{i-1}^2 \vee \overrightarrow{\mathtt{post}_{12} P_{i-1}^1}^2$$

$$P_1^1 = P_0^1 \vee \overrightarrow{\mathtt{post}_{21} false}^1 = P_0^1$$

$$P_1^2 = false \vee \overrightarrow{\mathtt{post}_{12} P_0^1}^2 \equiv$$
$$\overrightarrow{\exists x_0, y_0, z_0 (z_0 = 0 \wedge y_0 - x_0 = 0 \wedge x_0 \le 1 \wedge x = 0)}^2 =$$
$$\overrightarrow{z_0 = 0 \wedge y_0 \le 1 \wedge x = 0}^2 = \exists x_0, y_0, z_0, t$$
$$(z_0 = 0 \wedge y_0 \le 1 \wedge x_0 = 0 \wedge x = x_0 + t \wedge z = z_0 + t \wedge y = y_0) =$$
$$z - x = 0 \wedge y \le 1$$

$$P_2^2 = P_1^2 \vee \overrightarrow{\mathtt{post}_{12} P_1^1}^2 = P_1^2 \vee \overrightarrow{\mathtt{post}_{12} P_0^1}^2 = P_1^2$$

$$P_2^1 = P_1^1 \vee \overrightarrow{\mathtt{post}_{21} P_1^2}^1 \equiv (z = 0 \wedge x \le 1 \wedge y - x = 0 \wedge y \le 1) \vee$$
$$\overrightarrow{\exists x_0, y_0, z_0 (z_0 - x_0 = 0 \wedge y_0 \le 1 \wedge x_0 \ge 30 \wedge x = 0)}^1 =$$
$$(z = 0 \wedge x \le 1 \wedge y - x = 0 \wedge y \le 1) \vee \overrightarrow{z_0 \ge 30 \wedge y_0 \le 1 \wedge x = 0}^1 =$$
$$\exists x_0, y_0, z_0, t (z_0 \ge 30 \wedge y_0 \le 1 \wedge x_0 = 0 \wedge x_0 + t \le 1 \wedge x = x_0 + t \wedge z = z_0$$
$$\wedge y = y_0 + t) = \underline{[z = 0 \wedge x \le 1 \wedge y - x = 0 \wedge y \le 1] \vee [x \le 1 \wedge z \ge 30 \wedge y - x \le 2]}$$

$$P_3^1 = P_2^1 \vee \overrightarrow{\mathtt{post}_{21} P_2^2}^2 = P_2^1 \vee \overrightarrow{\mathtt{post}_{21} P_1^2}^2 = P_2^1$$

$$P_3^2 = P_2^2 \vee \overrightarrow{\mathtt{post}_{12} P_2^1}^2 \equiv (z - x = 0 \wedge y \le 1) \vee$$
$$\overrightarrow{\exists x_0, y_0, z_0 ([z_0 = 0 \wedge x_0 \le 1 \wedge y_0 - x_0 = 0 \wedge y_0 \le 1] \vee}$$
$$\overrightarrow{[x_0 \le 1 \wedge z_0 \ge 30 \wedge y_0 - x_0 \le 1] \wedge x_0 \le 1 \wedge x = 0}^2 = (z - x = 0 \wedge y \le 1) \vee$$
$$\overrightarrow{[z_0 = 0 \wedge x = 0 \wedge y_0 - x_0 = 0 \wedge x_0 \le 1 \wedge y_0 \le 1] \vee}$$
$$\overrightarrow{[x_0 \le 1 \wedge z_0 \ge 30 \wedge y_0 - x_0 \le 1 \wedge x = 0]}^1 = (z - x = 0 \wedge y \le 1) \vee$$
$$\exists x_0, y_0, z_0, t ([z_0 = 0 \wedge x_0 = 0 \wedge y_0 - x_0 = 0 \wedge x_0 \le 1 \wedge y_0 \le 1] \vee$$
$$[x_0 \le 1 \wedge z_0 \ge 30 \wedge y_0 - x_0 \le 1 \wedge x_0 = 0]) \wedge (x = x_0 + t \wedge z = z_0 + t \wedge y = y_0) =$$
$$(z - x = 0 \wedge y \le 1) \vee (z - x \ge 30 \wedge y \le 2)$$

Таким образом, получаем итерационную формулу для достижимых состояний:

$$P_n^1 = P_{n-1}^1 \quad \text{— } n \text{ – нечетное}$$
$$P_n^1 = P_{n-1}^1 \vee x \le 1 \wedge y - x \le n \wedge z \ge 30n \quad \text{— } n \text{ – четное}$$
$$P_n^2 = P_{n-1}^2 \quad \text{— } n \text{ – четное}$$
$$P_n^2 = P_{n-1}^2 \vee y \le n - 1 \wedge z - x \ge 30(n-2) \quad \text{— } n \text{ – нечетное}$$

Общий вид предиката, характеризующего достижимые состояния:

$$S \equiv [x \le 1 \wedge y = x \wedge z = 0 \vee \exists n \ge 1 (x \le 1 \wedge y - x \le n \wedge 30n \le z)]$$
$$\vee$$
$$[y \le 1 \wedge z = x \vee \exists n \ge 1 (y \le n + 1 \wedge 30n \le z - x)]$$

- Верификация требования — поскольку

$S \wedge \neg(y + z > 60 \Rightarrow 20y \le y + z) = false$

то система удовлетворяет требованию (15) корректности.

Пример 2. Система температурного контроля атомного реактора (обратный метод символьной верификации)

Система температурного контроля атомного реактора состоит из датчика температуры в котле реактора и двух независимых подвижных стержней, с помощью которых реактор может охлаждаться. Целью системы является поддержание температуры в реакторе в пределах между θ_m и θ_M. При достижении критического значения θ_M, котел должен быть охлажден одним из стержней. При этом использование каждого стержня возможно не чаще чем через T единиц времени после окончания его последнего использования. Если критическая температура θ_M в котле достигается раньше чем возможно использование обоих стержней, то необходима полная остановка реактора. Очевидное требование к поведению системы заключающется в том, что реактор не должен останавливаться. Это требование корректности можно выразить темпоральной формулой $\Box\neg stop$.

Для демонстрации обратного метода символьной верификации предположим, что температура в реакторе меняется по линейным законам[12]. Пусть v_r — скорость самопроизвольного увеличения температуры в реакторе, v_1 и v_2 — скорости охлаждения реактора с помощью первого и второго стержней соответственно. Система контроля в этом случае является линейной гибридной системой с постоянными коэффициентами.

Символьный метод состоит из следующих шагов:

- Построение `tcp`-автомата для гибридной системы (рис.7). Дополнительные вещественные переменные x_1 и x_2, отслеживат время между двумя последовательными использованиями каждого из стержней. Дополнительная управляющая переменная l принимает значение номера текущей локации.

- Описание для каждой локации гибридного автомата `tcp`-предикаты по одной из формул (например, по формуле (7)). В данном случае, очевидно, они будут выглядеть так:

$$\begin{aligned}
&\mathtt{tcp}_0(X) \equiv \theta + v_r t \leq \theta_M \\
&\mathtt{tcp}_1(X) \equiv \theta - v_1 t \geq \theta_m \\
&\mathtt{tcp}_2(X) \equiv \theta - v_2 t \geq \theta_m \\
&\mathtt{tcp}_3(X) \equiv true
\end{aligned}$$

- Спецификация требования формулой жизнеспособности темпоральной

[12] в работе [27] демонстрируется символьная верификация (с использованием системы HyTech) для решения задачи параметрического синтеза значения T системы температурного контроля атомного реактора в более общем случае, когда законы изменения температуры описываются дифференциальными включениями

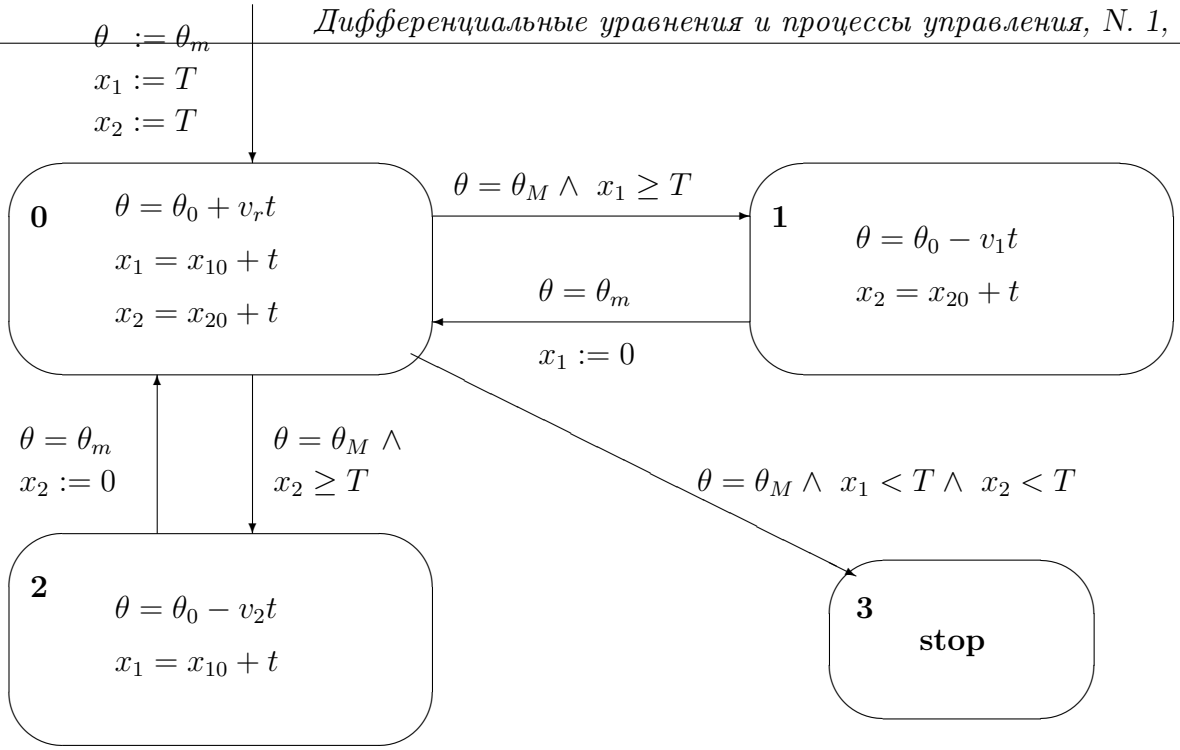


Рис. 7. Гибридный автомат системы температурного контроля атомного реактора.

логики реального времени. Требование корректности $\Box\neg(l = 3)$ преобразуется в требование жизнеспособности достаточно просто:

$$I \Rightarrow \Box(l = 3) \quad \equiv \quad \neg\exists\Diamond(l = 3)$$

где I — предикат, описывающий начальные значения всех переменных и значания параметров гибридного автомата. Пусть

$$I \equiv \theta_M = 15 \wedge \theta_m = 3 \wedge v_r = 6 \wedge v_1 = 4 \wedge v_2 = 3 \wedge T = 6$$

Таким образом, следует найти множество состояний, из которых достижимы состояния, описываемые предикатом $\exists\Diamond(l = 3)$ и доказать, что $I \wedge \neg\exists\Diamond(l = 3) \neq \emptyset$.

- Применение одного из итерационных правил обратного метода для темпоральной формулы $P = \exists\Diamond P_\varphi(P_\varphi \equiv l = 3)$. Очевидно, следует выбрать правило 1 для формулы $P_{\varphi_1}\exists\mathcal{U}P_{\varphi_2}$, так как $\exists\Diamond P_\varphi = true\exists\mathcal{U}P_\varphi$.

Характеристическое множество формулы P есть $\bigvee_i P_i$, где

$$P_0 = P_\varphi \equiv l = 3;$$

$$P_{i+1} = P_i \vee true \triangleright P_i;$$

Заметим, что

$(true \triangleright P)(X) \equiv$

$\exists e \exists t(\mathtt{pre}_e(P)(\varphi_s(X)(t)) \wedge (\mathtt{tcp}_s(X)(t)) \vee P(\varphi_s(X)(t)) \wedge \mathtt{tcp}_s(X)(t)$

кроме того, очевидно, что

$\mathtt{pre}_e P_1 \vee P_2 = \mathtt{pre}_{e_1} P_1 \vee \mathtt{pre}_{e_2} P_2$,

где $\mathtt{pre}_e P = b(X) \wedge P(f(X))$

Вычисление итерационных формул:

$P_1 \equiv (l = 3) \vee true \triangleright (l = 3) = (l = 3) \vee \exists e \exists t(l = 0) \wedge \theta + 6t = 15 \wedge x_1 + t < 6 \wedge x_2 + t < 6 \wedge \theta \leq 15) \vee (true \wedge l = 3) =$

$\{6t = 15 - \theta\} = (l = 0 \wedge \theta \leq 156x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta) \vee (l = 3);$

$P_2 = P_1 \vee true \triangleright P_1;$

$\mathtt{pre}_e[P_1 \vee P_0] = \mathtt{pre}_e[(l = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta) \vee (l = 3)] =$

$\mathtt{pre}_e(l = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta) \vee \mathtt{pre}_e(l = 3) = \mathtt{pre}_{e_{10}} P_1(X) \vee \mathtt{pre}_{e_{20}} P_1(X) \vee \mathtt{pre}_{e_{03}} P_0(X);$

$\mathtt{pre}_{e_{10}} P_1(X) \equiv \theta = 3 \wedge x_1 = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta;$

$\mathtt{pre}_{e_{20}} P_1(X) \equiv \theta = 3 \wedge x_2 = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta;$

$true \triangleright (P_1 \vee l = 3)(X) \equiv \{\theta^{(1)} = \theta - 4t; \theta^{(2)} = \theta - 3t; x_1 \to x_1 + t; x_2 \to x_2 + t\} =$

$l = 1 \wedge \theta - 4t = 3 \wedge x_1 = 0 \wedge \theta - 4t \leq 15 \wedge 6(x_2 + t) < 21 + (\theta - 4t) \wedge 6(x_1 + t) < 21 + (\theta - 4t) \wedge \theta \geq 3$

$\vee$

$l = 2 \wedge \theta - 3t = 3 \wedge x_2 = 0 \wedge \theta - 3t \leq 15 \wedge 6(x_2 + t) < 21 + (\theta - 3t) \wedge 6(x_1 + t) < 21 + (\theta - 3t) \wedge \theta \geq 3$

$\vee$

$l = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta$

$\vee$

$(l = 3) =$

$[l = 1 \wedge 3 \leq \theta \leq 15 \wedge 6x_2 + 10t < 21 + \theta \wedge \theta - 4t = 3] \vee$

$[l = 2 \wedge 3 \leq \theta \leq 15 \wedge 6x_1 + 9t < 21 + \theta \wedge \theta - 3t = 3] \vee$

$[l = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta] \vee [l = 3] =$

$\{4t = \theta - 3; \to 12x_2 + 3\theta < 57; 3t = \theta - 3 \to 6x_1 + 3\theta < 30\} =$

$[l = 1 \wedge 3 \leq \theta \leq 15 \wedge 4x_2 + \theta < 19] \vee$

$[l = 2 \wedge 3 \leq \theta \leq 15 \wedge 3x_1 + \theta < 15] \vee$

$[l = 0 \wedge \theta \leq 15 \wedge 6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta] \vee [l = 3];$

аналогичным образом получаем,

$P_3 \equiv [l = 0 \wedge \theta \leq 15 \wedge (6x_1 < 21 + \theta \wedge 6x_2 < 21 + \theta \vee 6x_2 + 3 < \theta)]$
$[l = 1 \wedge 3 \leq \theta \leq 15 \wedge 4x_2 + \theta < 19] \vee$
$[l = 2 \wedge 3 \leq \theta \leq 15 \wedge 3x_1 + \theta < 15] \vee [l = 3];$

$P_4 \equiv P_3$

Предикат $\neg \bigvee_{i=0}^{3} P_i(0)$, представляющий множество состояний, их которых достижимы состояния, описанные темпоральной формулой $\neg\exists\Diamond(l = 3)$, имеет вид:

$l = 0 \wedge \theta \leq 15 \wedge (\theta + 21 \leq 6x_1 \wedge \theta \leq 6x_2 + 3 \vee \theta + 21 \leq 6x_2) \vee$
$l = 1 \wedge 3 \leq \theta \leq 15 \wedge 19 \leq 4x_2 + \theta \vee$
$l = 2 \wedge 3 \leq \theta \leq 15 \wedge 15 \leq 3x_1 + \theta;$

• Предикат I, характеризующий множество начальных состояний, удовлетворяет предикату $\neg \bigvee_{i=0}^{3} P_i(0)$, следовательно требование корректности доказано.

Нетрудно доказать, что при значении $T = 8$ требование корректности уже не будет удовлетворяться. Действительно, результатом итерационного алгоритма для темпоральной формулы $\exists\Diamond l = 3$ после восьми итераций будет являться предикат (подробнее см. [23]):

$\neg \bigvee_{i=0}^{7} P_i(0) \equiv$
$l = 0 \wedge \theta > 15 \vee$
$l = 1 \wedge (\theta < 3 \vee \theta > 15) \vee$
$l = 2 \wedge (\theta < 3 \vee \theta > 15)$

Так как предикат $I \equiv l = 0 \wedge \theta \leq 15 \wedge x_1 \geq 6 \wedge x_1 \geq 6$, характеризующий множество начальных состояний системы, не удовлетворяет вышеописанному предикату, то локация 3 гибридного автомата достижима.

4.2. Интегрирующие графы

Альтернативным подходом (единственной серьезной альтернативой) к символьной верификации гибридных систем является использование вычислительной модели интегрирующего графа и сведение верификации линейных гибридных автоматов с постоянными коэффициентами и с предикатами вида

$$\Sigma_{i=1}^{n} a_i x_i \sim c, \quad x_i \in X, a_i, c \in Z, \sim = \{<, >, =, \neq, \leq, \geq\} \tag{16}$$

к задаче математического программирования [25].

Он иллюстрирует попытку объединить методы верификации таймированного автомата и теорию функций длительностей предикатов[13].

В общем случае, как было показано выше, для линейных гибридных автоматов проблема верификации неразрешима (см. *пессимистическое направление*), однако, оказывается, что если предикаты типа (16) присутствуют только на переходах, не участвующих в циклах, то задача верификации может быть успешна разрешена.

Определение 4.5. *Интегрирующим графом называется линейный гибридный автомат с постоянными коэффициентами с одной выделенной входной локацией $s_I \in S$, выделенным множеством терминальных локаций $S_F \subseteq S$, содержащий на циклических дугах только предикаты вида*

$$x_i \sim c,\ x_i - x_j \sim c, \quad c \in Z$$

где x_i, x_j — переменные-интеграторы

$$\Sigma\ x_i \sim c, \quad c \in Z$$

где x_i — переменные-таймеры

и содержащий на ациклических дугах предикаты вида (16).

Замечание 4.1. *Любой Интегрирующий граф может быть разбит на две части — "циклическую"часть L, которая имеет выходные дуги в нециклическую "терминальную"часть T, к которой относятся все терминальные вершины.*

Ограничение на класс линейных гибридных автоматов, специфицируемых данной моделью, заключается в требовании, чтобы предикаты вида (16) должны присутствовать только в части T. Переменные, участвующие в этих предикатах, называются терминальными.

Важно отметить, что модель исследует только конечные префиксы вычислений гибридного автомата.

На рисунке 8 изображен интегрирующий граф для системы Gas Burner (сравни с рис.6). В нем выделена начальная локация, описывающая множество состояний системы до начала произвольного наблюдения (локация s_0) и добавлена терминальная ациклическая часть, которая заключает в себе

[13]Теория функций длительности предикатов, Duration Calculus, появилась в теории реактивных систем в начале 1990-х годов как альтернатива темпоральной логики реального времени и метода символьной верификации ([26],[37]).

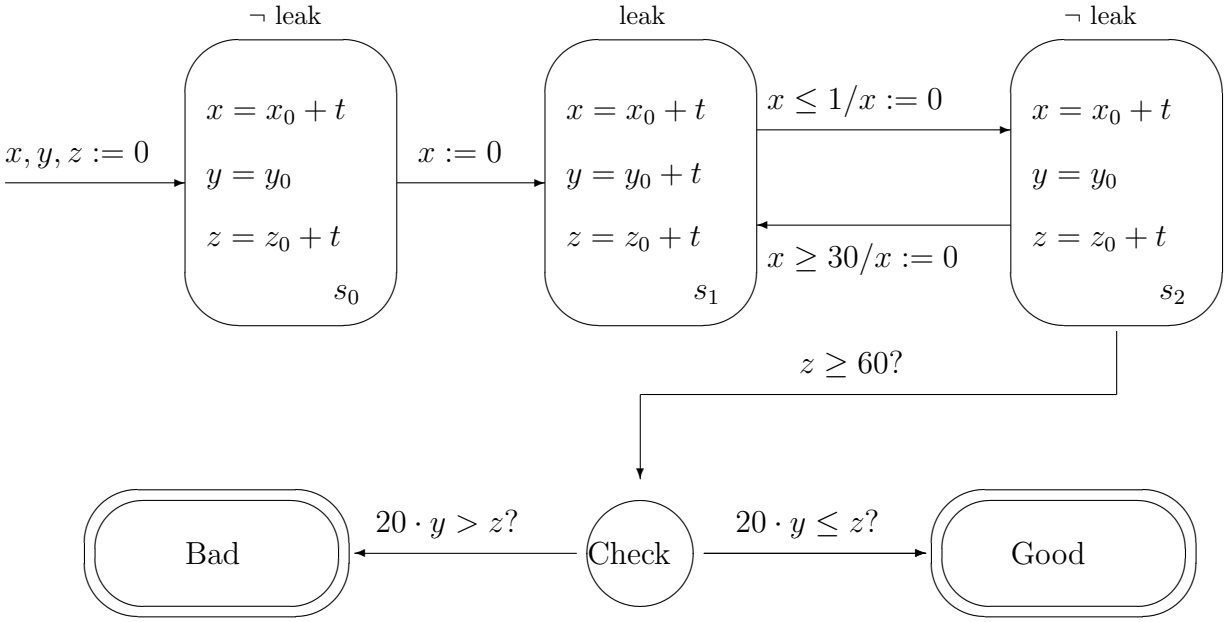


Рис. 8. Интегрирующий граф для гибридной системы Gas Burner.

проверку требования корректности (15) системы и имеет две терминальные локации "Bad"и "Good". Требование корректности (15) является сложным предикатом вида (16). Верификация этого требования методом интегрирующих графов заключается в доказательстве недостижимости терминальной локации "Bad".

Пусть $\mathcal{R} : S \times X \to Z$ — набор векторов целых констант, которые описывают значения производных интеграторов в каждой вершине интегрирующего графа, E — множество дуг графа.

Решение проблемы достижимости в интегрирующих графах сводится к верификации некоторого требования, описаного на языке логики счисления первого порядка, дополненной темпоральной функцией длительности и линейными неравенствами, над моделью конечного таймированного автомата, в который может быть преобразован интегрирующий граф.

Конечный таймированный автомат представляет собой упрощенную версию таймированного автомата, в которой рассматриваются только конечные траектории и опредено множество конечных локаций.

Для ясности дальнейшего изложения здесь необходимо ввести несколько определений теории функций длительностей предикатов.

Пусть φ — формула счисления предикатов первого порядка (терм), истинная в конкретных позициях пробега автомата $\tau = (s_0, t_0), \ldots, (s_n, t_n)$.

Определение 4.6. *Длительностью формулы φ ($\int \varphi$) в позиции j, $0 \leq j \leq n$, пробега τ называется суммарное время истинности формулы на пробеге до момента t_j:*

$$val(\tau, j, \int \varphi) = \Sigma_{0 \leq i < j, s_i \models \varphi}(t_{i+1} - t_i)$$

Определение 4.7. *Формулами длительности (Duration formula) называются логические комбинации предикатов с длительностью, которые определяются как неравенства вида*

$$\Sigma_{i=1}^{m} a_i \int \varphi_i \sim c,$$

где $\sim \in \{<, >, =, \neq, \leq, \geq\}, a_i, c \in Z$, φ_i – термы.

Примерами формул длительности могут служить формулы типа:

$$\bigvee \Sigma_{i=1}^{m} a_i \int \varphi_i \sim c$$
$$\bigwedge \Sigma_{i=1}^{m} a_i \int \varphi_i \sim c$$

Преобразование Интегрирующего графа к конечному таймированному автомату сводится к замене интеграторов (у которых $\mathcal{R} \neq 0$) и всех предикатов вида (16) на дизъюнктивную формулу длительности.

Пусть $\hat{s} \in s_F$ — выделенная терминальная вершина Интегрирующего графа, исследуемая на достижимость ($\hat{s} \in T$).

Конечный таймированный автомат конструируется из Интегрирующего графа по следующим принципам:

1. из Интегрирующего графа удаляются все дуги и вершины, не участвующие в путях $s_I \to \hat{s}$

2. предикаты вида (16) на дугах заменяются на тривиальный предикат "True"

3. из множества терминальных локаций конечного таймированного автомата удаляются все вершины кроме $\hat{s}$.

Затем конструируется формула длительности, которая изображает условие, при котором система может оказаться в вершине $\hat{s}$.

Значение некоторая терминальная переменная Интегрирующего графа $x \in X$ в текущий момент времени может быть описано формулой длительности[14]:

[14]Предикат at_s_i есть обозначение (уникальная метка) вершины s_i, он истинен только в этой вершине.

$$x(t) = x_0 + \mathcal{R}_1 \int at\_s_1 + \ldots + \mathcal{R}_i \int at\_s_i$$

где $x_0 \in X_0,\ s_1, \ldots, s_i \in S$ — вершины, в которых переменная x изменялась до момента t ($\mathcal{R}_i$ — соответствующие значения производной).

Очевидно, при $s_i = \hat{s}, x(t) = x_F$ — формула длительности, которая описывает значение переменной x в терминальной вершине.

Пусть теперь ψ_i — есть конъюнкция всех предикатов вида (16), которые встречались на дугах пути i (и были заменены на True), и в которых каждое вхождение переменной x заменено на соответствующую формулу длительности x_F. Поскольку T — ациклическая часть графа, то количество путей $s_I \to \hat{s}$ конечно (пусть n) и, следовательно, можно определить дизъюнктивную формулу длительности

$$\psi = \psi_1 \vee \ldots \vee \psi_n$$

Теорема 4.1. *Локация $\hat{s}$ интегрирующего графа достижима тогда и только тогда, когда существует пробег в соответствующем конечном таймированном автомате, на которой истинна формула ψ.*

На рисунках 9 и 10 изображены два варианта построения конечного таймированного автомата для интегрирующего графа системы Gas Burner. Первый вариант соответствует случаю, когда переменная z интерпретируется как таймер. Соответсвтующая формула длительности имеет вид:

$$\psi_1 : 20 \int at\_s_1 > \int at\_s_0 + \int at\_s_1 + \int at\_s_2$$

Во втором варианте переменная z интерпретируется как интегратор и формула длительности имеет вид:

$$\psi_1 : 20 \int at\_s_1 > \int at\_s_0 + \int at\_s_1 + \int at\_s_2 \wedge \int at\_s_0 + \int at\_s_1 + \int at\_s_2 \geq 60$$

В обоих случаях исследуется на достижимость вершина "Bad".

Верификация истинности некоторой формулы длительности на конечном таймированном автомате сводится к решению задачи линейного целочисленного программирования. Пусть

n_s – число визитов вершины s на пробеге

m_e – число переходов по дуге e на пробеге

r_e – последовательный номер дуги e в списке дуг траектории, упорядоченном в соответствии с последовательностью их первого визита (то есть если $r_e = k$, то e есть k-тая дуга пробега автомата)

В систему неравенств войдут следующие компоненты (здесь e^- – входная дуга, Δ – функция длительности пребывания в локации):

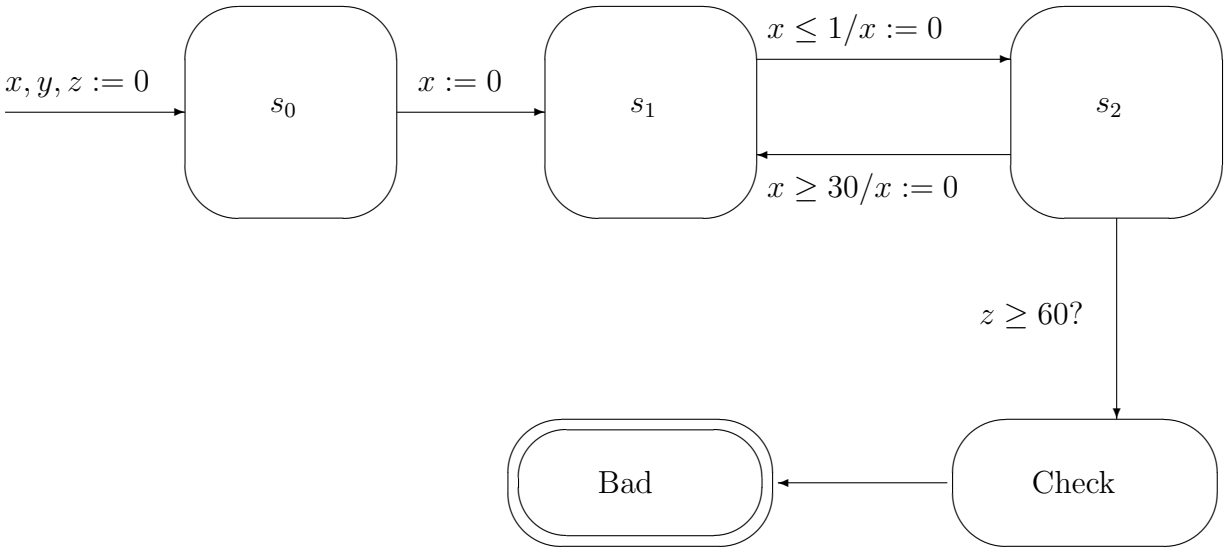


Рис. 9. Приведение интегрирующего графа системы Gas Burner к конечному таймированному автомату. z — таймер

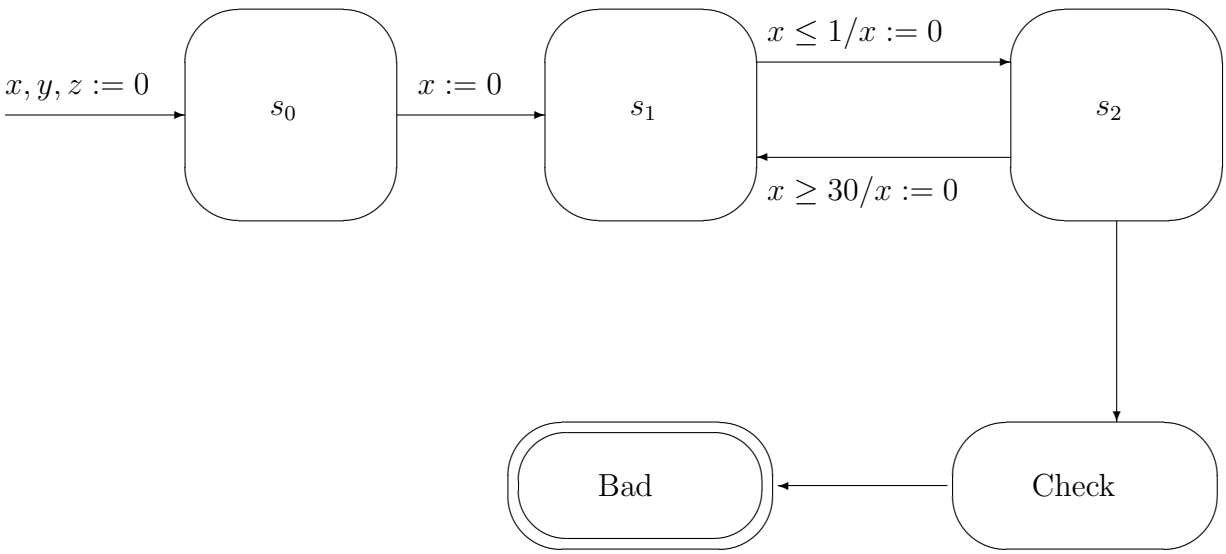


Рис. 10. Приведение интегрирующего графа системы Gas Burner к конечному таймированному автомату. z — интегратор

• **Flow:** $\forall s \in S - \{s_F\}$

$$n_s = \Sigma_{e_j \in pred(s)} m_j = \Sigma_{e_i \in succ(s)} m_i$$

(траектории не могут завершиться нигде, кроме как в s_F)

• **Initiation and Termination:**

$$n_{s_F} = \Sigma_{e_j \in pred(s_F)} m_j = m_{e^-} = 1$$

(войти в s_F и в s_0 можно только 1 раз)

• **Accessibility:**

$$m_{e^-} = r_{e^-} = 1 \quad \forall e \in E - \{e^-\}$$
$$(m_e = r_e = 0) \vee (\bigvee_{e' \in pred(e)} (0 < r_{e'} < r_e))$$

(упорядоченность переходов, $pred(e) = pred(s)$)

• **Visits Durations:** $\forall s \in S$

$$\Delta(s) = n_s v_s$$

• **The Duration Property:**

$$\bigvee \bigwedge \Sigma_{i=1}^{m} a_i \Sigma_{s \in S, s \models \varphi_i} \Delta(s) \sim c$$

Система должна быть решена относительно неизвестных

$$n_s \geq 0, \quad m_e \geq 0, \quad r_e \geq 0$$

Эффективный алгоритм решения этой классической задачи ЛЦП можно найти в работах по математическому программированию.

Разрешимые системы линейных неравенств могут быть получены для целочисленных пробегов конечного таймированного автомата, когда считается, что пребывание в любой его вершине может иметь только фиксированную длительность $v_s = \{0, 1\}$, а также для конечного таймированного автомата с одним вещественным таймером в траекториях (истинность общей формулы длительности) и для общего случая вещественных траекторий конечного таймированного автомата (истинность дизъюнктивной формулы длительности) ([25]).

5. Заключение

Методы, разработанные в гибридном направлении теории реактивных систем в последние несколько лет, преобретают все большую популярность в области компьютерного моделирования и анализа непрерывно-дискретных

систем. Об этом свидетельствует появление на западе программных комплексов по моделированию и анализу поведения непрерывно-дискретных и сложных встроенных систем управления, в основе которых лежит метод символьной верификации (система HyTech, 96 год [27]) и аксиоматический подход, разработанный на базе карт состояний (системы Statemate [10] и др.). Несомненно гибридное направление является привлекательным, так как позволяет проводить анализ качественных свойств поведения непрерывно-дискретных систем, не прибегая к численному моделированию отдельных траекторий, кроме того, как уже было замечено, непрерывно-дискретные системы имеют много общего с реактивными системами, изучаемыми теорией реактивных систем.

Однако, обзор достижений в этой области наглядно показывает, что методы автоматической верификации применимы только для некоторых (достаточно узких) классов непрерывно-дискретных систем и поэтому не могут претендовать на универсальность. В таблице 1 представлены известные на сегодня классы назрешимых и неразрешимых гибридных систем с точки зрения рассмотренных методов. Фактически таблица 1 демонстрирует максимальный уровень сложности непрерывных локальных поведений, при котором методы теории реактивных систем дают удовлетворительные результаты.

Кроме того, при использовании методов, основанных на дискретизации пространства состояний, возникают проблемы, которые связаны с динамической природой исследуемых систем. К ним относятся, например, проблема устойчивости и достоверности полученных решений в некоторой ε–окрестности выделенных областей, эквивалентность результатов анализа, получаемых различными методами (в том числе численным моделированием). Интересными вопросоми является оценка минимального уровня сложности дискретных процессов системы, при котором использование технологии моделирования непрерывных процессов, теории управления и оптимизационных методов затруднено или становится менее эффективным, чем использование методов гибридного направления, и с другой стороны оценка реальных объемов эффективно решаемых автоматическими методами верификации задач (максимальное количество параллельных динамических элементов, максимальный размер гибридных автоматов для каждого элемента). На сегодня примеры гибридных систем, демонстрирующих методы верификации, представляют из себя тривиальные системы автоматического управления. Гибридный автомат, описывающий поведение, к примеру,

системы управления наблюдениями на радиотелескопе РАТАН-600, о которой упоминалось ранее, даже в упрощенном случае имеет C_n^3 локаций , связанных дугами по принципу "каждый с каждым"(n – число наблюдательных циклов).

Пока что эти проблемы никак не освещаются в гибридном направлении, но они несомненно станут решающими на этапе внедрения методов верификации в программные комплексы моделирования непрерывно-дискретных систем.

Автор статьи выражает благодарность своим руководителям профессору Ю.Г.Карпову и Ю.Б.Сениченкову за постановку задачи, постоянный интерес, помощь в работе и за ценные замечания по статье.

Таблица 1. Верификация гибридных систем

класс гибридных систем	класс требований к поведению	метод верификации	ссылки
Линейные гибридные системы с постоянными целочисленными коэффициентами $F = \{dX/dt = K\}, K \in Z$ $\lambda = \{X := [\alpha; \beta]\}, \alpha, \beta \in Z$ $\phi : X \sim C,\ C \in Z$	формулы темпоральной логики деревьев вычислений	преобразование к модели таймированного графа, алгоритм Кроноса	стр.23, [32]
Линейные гибридные системы с постоянными целочисленными коэффициентами $F = \{dX/dt = K\}, K \in Z$ $\lambda = \{X := Y\}\ Y \in Z$ $\phi : \Sigma a_i x_i \sim c,\ a_i, c \in Z$ (запрещены на циклических дугах) $\phi : x \sim c; x_i - x_j \sim c,\ c \in Z$ (разрешены всюду)	формулы длительности $\vee \wedge \Sigma a_i \int \varphi_i \sim c$ $a_i, c \in Z$	преобразование к модели конечного таймированного графа и решение задачи линейного программирования[15]	стр.36, [25]
класс гибридных систем	класс требований к поведению	метод верификации	ссылки
Линейные гибридные системы $F = \{dX/dt = \Lambda, \Lambda \in [A, B]\}$, $A, B \in Z$ $\lambda = \{X := [\alpha; \beta]\}, \alpha, \beta \in Z$ $\phi : X \sim C,\ C \in Z$	множество неизбежных формул $(\forall)$ темпоральной логики деревьев вычислений	преобразование к модели таймированного графа, алгоритм Кроноса	стр.24, [32]
Гибридные системы с прямоугольными дифференциальными включениями $F = \{dX/dt \in [L; U]\},\ L, U \in Z$ $\lambda = \{X := [l; u]\},\ l, u \in Z$ $\phi : X \sim C,\ C \in Z$	формулы темпоральной логики деревьев вычислений	наложение Δ-сетки, алгоритм Кроноса	стр.21, [31]
Гибридные системы типа зависающего автомата $F = \{dX/dt = 0; 1\}$ $\lambda = \{X := Y\},\ Y \in Z$ $\phi : x_i \sim c,\ x_i - x_j \sim c, c \in Z$	формулы темпоральной логики деревьев вычислений	построение timer region automaton, алгоритм Кроноса	стр.22, [36]

[15] доказана неразрешимость проблемы достижимости для систем с $\lambda = \{X := Y\}\ Y \in Z$ и $\phi : \Sigma a_i x_i \sim c,\ a_i, c \in Z$ размерности > 2, [25]

Таблица 1. (Продолжение)

Гибридные системы с кусочно-постоянными производными $F = \{dX/dt = C\}, C \in R$ $\lambda =$ — (нет разрывов) $\phi : \Sigma a_i x_i \sim c,\ a_i, c \in R$	формулы гибридной темпоральной логики	символьная верификация[16] гибридных систем	[33]
Гибридные системы общего вида F – аналитическая функция $\lambda = \{X := Y\},\ Y \in Q$ $\phi : \Sigma a_i x_i \sim c,\ a_i, c \in Q$	формулы темпоральной логики деревьев вычислений	символьная верификация гибридных систем[17]	стр.25, [23]
Гибридные системы общего вида F – дифф. ур-ния 1-го порядка $\lambda = \{X := Y\},\ Y \in Q$ ϕ – терм	формулы гибридной темпоральной логики, требования корректности	аксиоматическая индуктивная система доказательств	[21], [35]

Список литературы

[1] Бусленко Н.П.: Моделирование сложных систем. М: "Наука", 1978.

[2] Программное обеспечение моделирования непрерывно-дискретных систем.(под ред. В.Глушкова), М: "Наука", 1975.

[3] Прицкер А.: Введение в имитационное моделирование и язык СЛАМ II. М: "Мир", 1987, 646с.

[4] Inichova M.A., Inichov D.B., Kolesov Y.B., Senichenkov Y.B.: Movel Vision for Windows. Graphical environment for hybrid systwr simulating. User's Guade. Moscow-St.Petersburg, 1995.

[5] Kolesov Y.B., Senichenkov Y.B.: Model Vision 3.0 for Windows 95/NT. The graphical environment for complex dynamic system design. ICI&C'97 PROCEEDINGS, v.2, p.704-711, St.Petersburg, 1997.

[6] Kolesov Y.B., Senichenkov Y.B.: Visual specification language intended for event-driven hierarchical dynamic system with variable structure. ICI&C'97 PROCEEDINGS, v.2, p.712-719, St.Petersburg, 1997.

[7] Борщев А., Карпов Ю., Колесов Ю.: Спецификация и верификация систем логического управления реального времени. В сб. "Системная иняформатика", вып.2, ИСИ СО РАН, Н-ск, 1993, 40с.

[8] Парийская Е.: Сравнительный анализ математических моделей и подходов к моделированию и анализу непрерывно–дискретных систем. Готовится к публикации.

[9] Maler O., Manna Z., Pnueli A.: From Timed to Hybrid systems. Real-Time: Theory in Practice, Lecture Notes in Comp.Sc 600, p.447-484. Springer-Verlag, 1992.

[16] доказана сходимость при размерности 2 и доказана неразрешимость проблемы достижимости при размерности > 2, [33]

[17] сходимость метода не гарантируется

[10] Harel D.: Statecharts: a Visual Formalism for Complex Systems. Sci. Comput. Prog. 8, p.231-274, 1987.

[11] Manna, Z., Pnueli A.: The Temporal Logic of Reactive and Concurrent Systems. Springer-Verlag, 1992.

[12] Ben-Ari M., Manna, Z., Pnueli A.: The Temporal Logic of Branching Time. Proc. 8th Annual Symposium on Principles of Programming Languages, 1981, ACM Press, Williamsburg, p. 164-176.

Springer-Verlag, 1992.

[13] Clarke E.M., Emerson E.A.: Design and synthesis of synchronisation skeletons using branching-time temporal logic. In Workshop on Logic of Program, Lecture Notes in Comp.Sc. 131. Springer-Verlag, 1981.

[14] Clarke E.M., Emerson E.A., Systla A.P.: Automatic verification of finite-state concurrent systems using temporal-logic specifications. ACM Transactions on Programming Languages and Systems, 8(2): p.244-263. 1986.

[15] Bertolomieu B., Diaz M.: Modeling and verification of time dependent systems using time Petri nets. IEEE Transactions on Software Engineering, SE-17,N 3, March 1991.

[16] Baeten J.C.M., Bergstra J.A.: Real time Process Algebra. Formal Aspects of Computing, 3, p.142-188, 1991.

[17] Reed G.M., Roscoe A.W.: A Timed Model for Communicating Sequential Processes. Lecture Notes in Comp.Sc. 571, Springer-Verlag, 1986.

[18] Moller F., Tofts C.: A Temporal Calculus of Communicating Systems. Proceedings of CONCUR'90. Lecture Notes in Comp.Sc. 458, Springer-Verlag, 1990.

[19] Henzinger T.A., Manna, Z., Pnueli A.: Temporal proof methodologies for real-time systems. Proc. 18th Annual Symposium on Principles of Programming Languages, 1991, ACM Press, p. 353-366.

[20] Henzinger T., Nicollin X., Sifalis J., Yovine S.: Symbolic Model-Checking for Real-Time Systems. In Proc. 7th LICS. IEEE Comp. Soc. Press, 1992.

[21] Alur R., Courcoubetis C., Henzinger T., Ho P-T.: Hybrid automata: an algorithmic approach to the specification and analysis of hybrid systems. In Workshop on Theory of Hybrid Systems,Lyndby, Denmark, June 1993. LNCS 736, Springer-Verlag.

[22] Olivero A., Yovine S.: Kronos: a tool for verifying real-time systems. User's Guide and Reference Manual. VERIMAG, Grenoble, France, 1992.

[23] Nicollin X., Olivero A., Sifalis Y., Yovine S.: An Approach to the Description and Analysis of Hybrid Systems. Hybrid Systems, Lecture Notes in Comp.Sci 736, p.149-178. Springer-Verlag, 1993.

[24] Alur R., Courcoubetis C., Dill D.L.: Model-Checking for real-time systems. 5th LICS (5th IEEE Simp. Logic in Comp. Sci.), p.414-425. IEEE Comp. Soc. Press, 1990.

[25] Kesten Y., Pnueli A., Sifalis Y., Yovine S.: Integration Graph: a class of decidable hybrid systems. Hybrid Systems, Lecture Notes in Comp.Sci 736, p.179-208. Springer-Verlag, 1993.

[26] Chaochen Z., Hoare C.A.R., Ravn A.P.: A calculus of durations. Informations Processing Letters, 40(5): p.269-276. 1991.

[27] Henzinger T., Ho P-T.: HyTech: The Cornell Hybrid Technology Tool. Hybrid Systems II, Lecture Notes in Comp.Sci 999, p.265-293. Springer-Verlag, 1995.

[28] Wolfram S.: Mathematica: A System for Doing Mathematics by Computer. Addisson–Wesley Publishing Company, 1988.

[29] Redfern D.: The Maple Handbook. 531 pp., Springer-Verlag, 1994.

[30] Lichtenshtein O., Pnueli A.: Checking that finite state concurrent programs satisfy their linear specification. 12^{th} Symp. on Principles of Program Languages. Austin, Texas. 97–107, 1984.

[31] Puri A., Varajya P.: Decidability of Hybrid Systems with Rectangular Differential Inclusions. 6^th International Conference CAV'94. Lecture Notes in Comp. Sci. 818, p.95–104, 1994.

[32] Olivero A., Sifakis J., Yovine S.: Using Abstractions for the Verification of Linear Hybrid Systems. 6^th International Conference CAV'94. Lecture Notes in Comp. Sci. 818, p.81–94, 1994.

[33] Asarin E., Maler O., Pnueli A.: Reachability Analysis of Dinamical Systems having Piecewise-Constant Derivatives. Theoretical Comp. Sci., v.138,N.1, 1995

[34] Asarin E., Maler O.: On some Relations between Dynamical Systems and Transition Systems. Proc. of ICALP'94, Lecture Notes in Comp. Sci 820, p.59–72, Springer-Verlag, 1994.

[35] Henzinger T., Manna Z., Pnueli A.: Towards Refining Temporal Specifications into Hybrid Systems. In Hybrid Systems, LNCS 736, p.60–76, Springer-Verlag, 1993.

[36] J.McManis, P.Varajya – Suspension Automata: A Decidable Class of Hybrid Automata. 6^{th} International Conference CAV'94. Lecture Notes in Comp. Sci. 818, p.105–117, 1994.

[37] Choachen Z., C.A.R. Hoare, A.P.Ravn: A calculus of durations. Inform. Process. Lett. 40, p.269–276, 1991.

[38] Lamport L.: Specifying concurrent program modules. ACM Trans. Prog. Lang. Syst. 5, p.190–222, 1983.

[39] Lamport L.: What good is temporal logic. Proc.IFIP Congress,North-Holland, p.657–668, 1983.

[40] Manna Z.,Pnueli A.: Specification and verification of concurrent program by $\forall$-automata. 14^{th} Symp. on on Principles of Program Languages. ACM, Minich, January 1987.

[41] Alur R, Fix L, Henzinger T.A.: A Determinizable Class of timed Automata. 6^{th} International Conference CAV'94. Lecture Notes in Comp. Sci. 818, p.1–13, 1994.