ГИБРИДНАЯ МОДЕЛЬ ПРОЦЕССА ВЫЧИСЛЕНИЙ: ОБОБЩЕНИЕ КОНЦЕПЦИИ МАШИНЫ ТЬЮРИНГА

В статье предлагается новая концепция вычислительного устройства, обобщающая схему классической машины Тьюринга. В рамках новой модели переосмысливаются ставшие традиционными понятия “такта”, “памяти”, “программы” и “состояния”. В частности, в новой модели “память” представляет собой модель динамической системы, “программа” — “скачкообразную” эволюцию памяти по достижении терминальных множеств, а “тактом” считается достижение терминальных множеств. Кроме того рассматривается динамика взаимодействия параллельных динамических систем с общей памятью в применении к такой модели. Представленная в данной статье концепция упрощает и обобщает описанную ранее.

1. Введение

Хорошо известно, что компьютеры становятся все миниатюрнее и миниатюрнее. Когда-то предполагали, что более мощные машины будут требовать больше места под периферию, память и т.д. Это предположение оказалось неверным. В 1965 г. Г. Мур [1] сформулировал действующее и сейчас правило (впоследствии названное законом Мура), согласно которому производительность вычислительных систем удваивается каждые восемнадцать месяцев. Г. Мур вывел свой эмпирический закон, построив зависимость числа транзисторов в интегральной микросхеме от времени. Как следствие, из этого закона можно вывести темпы миниатюризации отдельного транзистора. Развитие цифровых электронных технологий приводит к тому, что размер элементарного вычислительного устройства приближается к размеру молекулы или даже атома. На таком уровне законы классической физики перестают работать и начинают действовать квантовые законы, которые для многих важных динамических задач еще не описаны теоретически.

Увеличение быстродействия вычислительных устройств и уменьшение их размеров с неизбежностью приводит к необходимости операций с “переходными” процессами. Вместо примитивных операций с классическими битами в будущем было бы естественно перейти операциям, задаваемыми теми или иными динамическими моделями микромира. Под примитивной “моделью” вычислений можно понимать выполнение классических операций с битами. Конечно же введение более широкого класса моделей было бы более обоснованным, если бы удалось, например, для функции, значения которой записаны в кластере квантовых битов, определить операцию, эффективно выполняющую преобразование Фурье. Причем может оказаться, что время на ее выполнение будет вполне соизмеримым со временем выполнения одной классической операции, так как операции типа “свертки” функций вполне могут обнаружиться “в природе”. Последние исследования похожих моделей показывают, что их выполнение за счет присущей природе способности к самоорганизации не обязательно “раскладывается” на более простые кирпичики, т.е. не всегда может быть записано в виде классического алгоритма.

Канонической моделью для описания классических алгоритмов является общеизвестная машина Тьюринга (далее, МТ) — абстрактное вычислительное устройство, предложенное в 1937 г. английским математиком А. Тьюрингом [2]. Реализуемость алгоритма для такой машины оказывается эквивалентна конечному числу тактов (моментов передвижения головки автомата). В 1982 г. Р. Фейнман [3] обосновал невозможность представления результатов квантовой механики на таком вычислительном устройстве даже с использованием вероятностных алгоритмов. Это связано с тем, что число элементарных операций будет расти экспоненциально с ростом числа степеней свободы системы. В 1985 г. Д. Дойч [4] предложил модифицировать тезис Черча-Тьюринга (1932) [5], утверждающий, что “любая конечно реализуемая физическая система может быть достаточно точно смоделирована с помощью универсальной модели компьютера, оперирующего конечными средствами" следующим образом: "каждая конечно реализуемая физическая система может быть полностью промоделирована универсальной моделирующей вычислительной машиной, действующей конечными средствами". Д. Дойч построил модель такого универсального компьютера, основанную на идее квантовых вычислений, и показал, как квантовый компьютер может моделировать физические системы, которые находятся за пределами области, доступной универсальной МТ.

В информатике в рамках теории сложности алгоритмов получен целый ряд результатов, обосновывающих теоретическую невозможность эффективного решения многих значимых с практической точки зрения задач, в частности, связанных с моделированием естественных процессов. Это означает, что при попытке описания интересующего нас процесса с заданной точностью при помощи машины Тьюринга ее алфавиты состояний и памяти и/или время функционирования оказываются недопустимо велики. Одними из первых с этими теоретическими трудностями столкнулись разработчики систем автоматического управления, которые ведут исследования на стыке информатики и той или иной конкретной научной дисциплины (физики, биологии, химии и т.п.). В научной литературе многие авторы высказывают сомнения в корректности тезиса Черча-Тьюринга (например, Коплэнд (2004) [6], да Коста и Дориа (1991) [7], Дойл (1982) [8], Хогарф (1994) [9]).

В то же время реализации целого ряда современных методов теории автоматического управления не укладываются в рамки дискретной модели. Один из возможных способов ее обобщения — это использование так называемых "гибридных систем", свойства которых активно изучаются в последнее время (например, Гао, Лихерос, Кэнкампуа (2007) [10], Гёбель, Санфеличе и Теел (2009) [11]). В практике автоматического управления оказалась плодотворной идея релейного регулирования, заключающаяся в переключении системы управления с одного "скользящего" режима на другой. В последнее десятилетие в системах управления активно внедряется нейросетевой подход. Часто говорят об использовании в управляющих контурах нейрокомпьютеров.

Но работа перечисленных устройств не описывается полностью классической схемой машины Тьюринга. Одним из вариантов расширения тезиса Черча-Тьюринга является введение нового понятия вычислительного устройства, в котором вычисления строятся не на традиционных алгоритмах, а на моделях. Следующий естественный шаг — обобщение на модели динамических систем. Первым, основным и наиболее спорным, с современной точки зрения, понятием дискретной схемы машины Тьюринга является понятие "такт". В рамках классической теории множеств его противоречивый смысл выражается в рамках неразрешимости проблемы континуума в рамках аксиоматики Френкеля-Цермело.

Другим серьёзным ограничением классической модели вычислений является разбиение памяти на изолированные биты, потому как, во-первых, сокращение длины такта и расстояний между битами с определенного уровня делает невозможным рассматривать их изолированно в силу законов квантовой механики. Во-вторых, отказ от "скалярных битов" позволит, может быть, говорить о выполнении многомерных (векторных) операций за один "такт". Например, П. Шор (1997) [12] предложил алгоритм квантового преобразования Фурье, которое может выполняться за время пропорциональное $(\log N)^2$, а не за $N \log N$ как классическое быстрое преобразование Фурье. В работе Тиена (2003)[13] обсуждается опирающийся на квантовую адиабатическую теорему гипотетически возможный «физический»' способ решения за конечное время 10-ой проблемы Гильберта, в работе Сысоева С.С. и др. (2006)[14] предложен эффективный квантовый алгоритм вычисления “за такт” оценки вектора-градиента многомерной функции.

В статье Граничина и Жувикиной (2006) [15] была предложена и обоснована новая абстракция вычислительного устройства, включающая в себя все перечисленные выше подходы. Новая концепция принципиально отличается от классической включением “непрерывности"” эволюции и переосмыслением понятий “лента” и “ячейка памяти”. Если в классическом подходе ячейки памяти используются для хранения дискретной информации и их изменение возможно только в тех случаях, когда указатель ленты показывает на нее, то в новой концепции “ячейка памяти” представляет собой постоянно функционирующую модель какой-то динамической системы (возможно, достаточно сложной), а “лента” в память, имеющую гетерогенную природу, к которой может быть применена общая для всех ячеек программа эволюции. Очевидно, что классическая ячейка памяти для хранения “бита” является частным случаем такого обобщения.

В любых современных вычислительных устройствах хранение информации основано на тех или иных физических принципах, только эволюция состояния ячейки во время хранения более простая — сохраняется постоянной какая-то физическая характеристика. Другими словами, традиционную информатику можно сравнить с арифметикой, она оперирует цифрами — значениями ячеек памяти (или, более точно — арифметикой конечных двоичных дробей). Предлагаемая модель вычислений позволит перейти в информатике от “арифметики” к “функциональному анализу”, исследующему процессы эволюции информации внутри новых “ячеек”.

Статья организована следующим образом. Во втором разделе описывается математическая модель новых процессов вычислений как совокупности работающих параллельно динамических систем с пересекающимися частями фазовых пространств (общей памятью) и “скачкообразными” переключениями между ними. В третьем разделе дается формализованное описание обобщенной машины Тьюринга, включающее в себя уточнение более ранних попыток описания подобной системы (Граничин, Жувикина (2006)) [15]. Далее демонстрируется универсальность новой модели через описание классической машины на обобщённой.

2. Взаимодействие параллельных динамических систем с общей памятью

Определим процесс вычисления как совокупность работающих параллельно динамических систем с пересекающимися частями фазовых пространств (общей памятью). Для этого рассмотрим вычислительную систему состоящую из счётного набора аналоговых компонент $\{M_i\}_{i=1}^{\infty}$, такую что каждая компонента M_i определяет динамику некоторых компонент x_i общей памяти x (имеющей некоторую гетерогенную природу) по правилу $\dot{x}_i(t) = H_i(x_i, t)$, где $x_i(t) \in \mathrm{X}_i, \mathrm{X}_i \subset \mathrm{X}$.

Пусть в каждый момент времени t лишь небольшая часть моделей (компонент вычислительного устройства) оказывает влияние на изменение состояния памяти всей системы. Другими словами, в каждый момент времени t можно определить бесконечную строку битов $S(t) = \{s_i(t)\}_{i=1}^{\infty}$, состоящую из конечного числа единиц: $s_i(t) \in \{0,1\}$, для $i \le i_{\max}(t)$ и $s_i(t) = 0$, для $i > i_{\max}(t)$. Тогда общая динамика системы описывается как $\dot{x}(t) = \sum_{i=1}^{\infty} s_i(t) H_i(x_i, t)$, где $x(t) \in \mathrm{X}$ и $x_i(t) \in \mathrm{X}_i$.

Строку $S(t)$ в некотором смысле можно считать аналогом состояния в классической МТ. Множество всевозможных состояний обозначим Σ.

Не умаляя общности, можно считать, что $S(t)$ — кусочно-постоянная функция времени, т.е. на определенных интервалах те или иные модели (компоненты ВУ) «включены». Т.о. получаем гибридную динамическую систему в традиционном понимании этого термина. Общая динамика системы на k-м интервале постоянства $S(t)$ от t_k до t_{k+1} (со значением S_k) описывается как:

$$\dot{x}_{S_k}(t) = \sum_{i=1}^{i_{\max}(t_k)} s_i(t_k) H_i(x_i, t) = \sum_{i \in \sup p(S_k)} H_i(x_i, t), t \in [t_k, t_{k+1}),$$

где $x_{S_k}(t) \in \overline{\mathrm{X}}_{S_k} = \bigcup_{i \in \sup p(S_k)} \mathrm{X}_i \subseteq \mathrm{X}$.

Для полного описания динамики всей системы необходимо определить правило изменения $S(t)$. Для описания процессов решения (или моделирования) большого количества практических задач достаточно ограничиться заданием направленного графа переходов, связывающего возможные переходы $S_k \xrightarrow{J_{k,j}} S_j; S_k, S_j \subset \Sigma$ с указанием условий $J_{k,j} \in J$, при выполнении которых происходит переключение изменения динамики всей системы с набора моделей S_k на S_j.

Таким образом, можно определить, что в новом смысле программа P — это правила изменения $S(t)$. Если правила хранить в памяти ВУ и дать возможность их корректировки со временем, то с точки зрения программирования это в некотором смысле будет соответствовать архитектуре фон Неймана, когда и данные и программа хранятся в одной памяти и программа меняется в процессе работы системы.

Условием останова системы является $S(t) = 0$, которое эквивалентно тому, что ни один из компонентов памяти не является активным.

Пример 1: Рассмотрим описание классической МТ в рамках представленной модели.

Переходя к классической машине, каждую ячейку ленты можно рассматривать как некоторую модель, принимающую значения на конечном алфавите, а всю ленту — как вычислительную систему, состоящую из работающих параллельно динамических систем. В этом случае программу работы машины можно представить в виде направленного графа P, в узлах которого содержатся $S_{j,k}$, а в качестве рёбер — условия переключения $J_{j,k}$.

Если проиндексировать ячейки используя $k \in \mathrm{N}$, то на каждом такте работу машины можно описать простой дискретной системой

$$\begin{cases} x_{k\pm1}^{t+1} = f_{k\pm1}^{t}(x_k) \\ x_k^{t+1} = f_k^{t}(x_k) \end{cases},$$

где x — это вектор, хранящий и состояние машины, и содержимое памяти.

На каждом такте классической машины Тьюринга активной будет ровно одна ячейка ленты, то есть каждое S_k будет содержать только одну единицу. Соответственно, только одна из трёх функций $f_{k,k\pm1} \neq 0$.

Работу такой системы иллюстрирует *рис. 1*.

$$S_I \dashrightarrow S_\bullet \xrightarrow{J_{\bullet,k-1}} S_{k-1} \xrightarrow{J_{k-1,k}} S_k \xrightarrow{J_{k,k+1}} S_{k+1} \xrightarrow{J_{k+1,\bullet}} S_\bullet \dashrightarrow S_F$$

Рис. 1

Здесь S_I и S_F — инициальное и финальное состояния. $S_{k-1} = (0,0,0,..1,0,0,...0,0,0)$, $S_k = (0,0,0,..0,1,0,...0,0,0)$, $S_{k+1} = (0,0,0,..0,0,1,...0,0,0)$, такие что единицы соответствуют ячейкам k, $k-1$ и $k+1$ ленты.

Отметим также, что в реальной реализации машины должен присутствовать тактовый генератор $x_t(t) \in \{0,1\}$.

Пример 2: Рассмотрим систему, состоящую из трёх моделей $x_{1,2,3}$ (под моделью подразумевается комплексная система — память, состояние), которая совершает качественные прыжки из инициального состояния $S_I = (1,1,1)$, в состояния $S_1 = (0,1,1)$, или $S_2 = (1,0,0)$, и из них в финальное состояние $S_F = (0,0,0)$, при попадании в условия $J_{I,1}$, $J_{I,2}$, $J_{I,F}$, $J_{1,F}$, $J_{2,1}$ $\in J$. Пусть также $x_{1,2,3}$ делят общую, 2-хкомпонентную память $\mathrm{X} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}$ таким образом, что $x_1 = (\tilde{x}_1)$, $x_2 = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}$ и $x_3 = (\tilde{x}_2)$.

Такая система может быть представлена графом P (*рис. 2*).

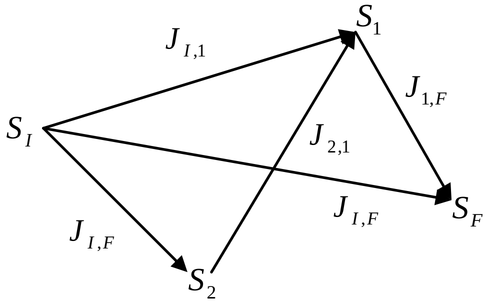

Рис. 2.

В состоянии S_I работа моделей описывается системой

$$\begin{cases} \dot{x}_1 = f_{I,1}(\tilde{x}_1, t) \\ \dot{x}_2 = f_{I,2}(x_2 = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}, t) \\ \dot{x}_3 = f_{I,3}(\tilde{x}_2, t) \end{cases}.$$

Аналогично, для состояния S_1: $\begin{cases} \dot{x}_1 = 0 \\ \dot{x}_2 = f_{1,2}(x_2 = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}, t) \\ \dot{x}_3 = f_{1,3}(\tilde{x}_2, t) \end{cases}$ и S_2: $\begin{cases} \dot{x}_1 = f_{2,1}(\tilde{x}_1, t) \\ \dot{x}_2 = 0 \\ \dot{x}_3 = 0 \end{cases}$.

В финальном состоянии S_F динамика системы отсутствует, т. е. $\dot{x} = 0$.

Такт работы системы отсчитывается в момент попадания в терминальное множество J. Представленные выше уравнения описывают межтактовую динамику.

Более общая схема получается при задании стохастических правил переходов. При этом детерминированный случай хорошо обобщается, задавая достаточно высокие вероятности переходов для определенных в нем переходов и оставляя равновероятный, почти нулевой, уровень вероятности для всех остальных. Но даже такая схема может оказаться более эффективной (по аналогии с известным методом отжига для поиска глобального минимума функции). Случайно установленные “новые связи” могут оказаться полезными для системы и можно допустить корректировку вероятностей переходов в определенных ситуациях с целью повышения их значимости. Такой механизм будет приводить к элементам самоорганизации моделей в процессе работы.

Отметим принципиальную разницу между понятиями состояния и памяти, отсутствующую в большинстве реальных вычислительных устройств. Память предполагается разбитой на отдельные ячейки, содержимое которых изменяется, вообще говоря, независимо друг от друга. Состояние же — это интегральная характеристика машины, возможно, влияющая на изменение содержимого памяти. Результатом вычисления в машине Тьюринга считается именно содержимое памяти по окончании работы.

3. Обобщенная модель машины Тьюринга

Модель обобщенной машины Тьюринга (далее, ОМТ) соответствует представленному выше описанию динамики. Машина предполагает наличие гетерогенной памяти и множества состояний, изменяющихся в процессе работы. Кроме того, заданы операторы эволюции состояний и памяти.

Итак, определим ОМТ как структуру $\langle S, s, s_0, \mathrm{X}, x, x_0, E, J, P, F \rangle$, в которой:

S — множество состояний;
s — текущее состояние машины;
s_0 — инициальное состояние;
X — множество состояний, которые может принимать память;

x — текущее содержимое памяти, где $x \in \mathrm{X}$;
x_0 — начальное содержимое памяти;
$E: Q \to Q$ — оператор эволюции состояний и памяти, где $Q = (S, \mathrm{X})$ — множество всех возможных наборов (s, x) состояний и памяти;
$P: J \to Q$ — программа машины, где $J \subset Q$ — множество задания программы, причём $J \cap F = \varnothing$;
$F \subset Q$ — финальное множество состояний и памяти.

В классической машине Тьюринга память (лента) — это упорядоченный набор ячеек в каждой из которых содержатся значения из некоторого алфавита Γ. В ОМТ понятие ленты преобразуется в понятие памяти x, которая имеет гетерогенную природу. Алфавитом для символов, которые располагаются в каждой ячейке МТ, является произвольное пространство состояний X.

Память x эволюционирует под действием операторов E и P из начального состояния x_0, пока текущее состояние и память машины (s, x) не оказываются в финальном множестве F. В этом случае машина прекращает работу и текущее состояние и память считается результатом работы. Если память и состояние попадает в терминальное множество J, то машина совершает скачок согласно программе P, а затем продолжает эволюционировать под действием оператора E до повторного попадания в J. При этом оператору эволюции E соответствуют системы дифференциальных уравнений, описанные в предыдущем разделе.

Тактом работы системы, в отличие от классической МТ, где такты отсчитываются сдвигами головки, считается переменный промежуток времени между попаданиями памяти и состояния машины в терминальное множество J.

Для описания параллельных процессов выгодно рассматривать множество машин M, где отдельная машина (далее, “модель”, или “эволюционный примитив”) представляется как

$$M^j = \langle S^j, s^j, s_0^j, \mathrm{X}^j, x^j, x_0^j, E^j \rangle,$$

где отдельное для каждой машины состояние $s^j \in S^j$ и память $x^j \in \mathrm{X}^j$, эволюционируют под действием оператора E^j.

В этом случае общая память машины это счётное множество $\{x^j\}_{j \in \mathrm{Z}}$. При этом для всех моделей задаётся общая программа P, которая работает для всей памяти сразу и учитывает состояние S_j каждой модели.

Отметим также, что ранее определённая модель (Граничин, Жувикина (2006)) [15] представляла собой структуру вида

$$\langle S, s, s_0, Y, X, x, x_0, F, J, P, E \rangle,$$

где Y — лента (память), X — множество состояний памяти, x — текущее содержимое памяти: $x(y) \subset X, \forall y \in Y$; x_0 — начальное содержимое памяти; терминальное множество $F \subset Q$, где Q — множество всевозможных наборов $q = (s, x)$ состояний и памяти; $J \subset Q$ — множество задания программы, причем $F \cap J = \varnothing$; $P: J \to Q$ — программа; $E: Q \to Q$ — оператор эволюции состояний и памяти.

Как видно, в данной статье удалось отказаться от указателя x на компоненты памяти и представить память как объект гетерогенный природы (или, в случае памяти поделённой на ячейки — в виде декартова произведения компонент). При этом для предложенной модели остаются верны все утверждения старой модели (поскольку всегда можно использовать координатные функции для получения соответствующих компонент) и, в том числе, оказываются верны все описанные ранее примеры эффективно вычислимых функций.

4. Описание классической МТ с использованием ОМТ

Согласно формальному определению, данному Хопкрофтом и Улманом (1979) [15], машина Тьюринга (одноленточная) — это структура вида

$$M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle \text{, где}$$

Q — конечно множество состояний;
Γ — конечный алфавит ячеек;
$b \in \Gamma$ — пустой символ;
$\Sigma \subseteq \Gamma \setminus b$ — множество входных символов;
$\partial : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ — частичная функция, называемая функцией переходов, где L означает сдвиг ленты влево, R — вправо;
$q_0 \in Q$ — начальное состояние;
$F \in Q$ — множество финальных состояний.

Алгоритм работы состоит в том, что на каждом такте в соответствии с правилом ∂ происходит изменение содержимого рабочей ячейки и состояния, а также сдвиг ленты. Если новое состояние q машины попадает в финальное множество F, то машина останавливается и результатом её работы считается текущее содержимое памяти. Кроме того, эквивалентным является описание машины, когда вместо сдвига ленты происходит сдвиг указателя $x = x \pm 1$ на текущую рабочую ячейку.

Покажем, как можно описать такую систему в обобщенном виде.

Ленте соответствует память $x = \Gamma^{\infty}$, где Γ — конечный алфавит состояний классической МТ. Ячейки ленты пронумерованы индексами $j \in Z$ и x_j — значение ячейки j, такое что $x_j = \varphi_j(x)$, где φ_j — соответствующая координатная функция. Текущее состояние $\tilde{q} = \langle q, k, t \rangle$ обобщённой машины хранит индекс $k \in Z$ текущей рабочей ячейки (положение головки чтения/записи), долю времени $t \in [0,1]$ внутри такта длиной δ, и q — состояние классической машины. Соответственно, пространство состояний $\tilde{Q}$ описывается, как $Q \times Z \times [0,1]$ и начальное состояние $\tilde{q}_0 = \langle q_0, 0, 0 \rangle$, то есть в инициальном состоянии головка показывает на нулевую рабочую ячейку.

Кроме того, финальное множество $\tilde{F} = \langle F, X \rangle$ и аналогично множество J, задания программы P определяется как подмножество $\langle \tilde{Q} \setminus \tilde{F}, \mathrm{X} \rangle$.

При переходе с такта i к $i+1$ сдвиг ленты влево или вправо эмулируется сдвигом указателя на рабочую ячейку $k_{i+1} = k_i \pm 1$. При этом программа P ОМТ — это кусочнозаданная функция, где её компоненты $P_j(x_i,\langle k,q\rangle)=(\tilde{x}_i,\langle k+\begin{cases}\pm1\\0\end{cases},q\rangle)$, где $P_j : J \to \tilde{Q}$.

Эволюционный оператор E равномерно изменяет время $t : \dot{t} = 1/\delta$, и является тождественным для остальных компонент состояния и содержимого памяти.

Заметим, что можно было бы момент останова машины определять только значением состояния, но введенное более общее правило, очевидно, включает в себя и этот случай. В новой модели исключено понятие рабочей ячейки, т.е. изменение состояния может зависеть от содержимого всей памяти в данный момент, а содержимое памяти может меняться одновременно на всем пространстве X.

Разделение описания процесса изменения состояния и памяти на две составляющие: программу P и эволюцию E на первый взгляд, может показаться искусственным. На самом деле такое разделение позволяет размежевать "принудительные" изменения, обычно вносимые в систему извне, задаваемые кем-то "осознанно", и те "естественные" процессы, которые происходят в определенной физической (или биологической) системе в силу законов природы.

Другим обобщением машины Тьюринга может быть вероятностное задание отображений P и E, что позволит реализовывать с помощью новой модели динамические системы, неописываемые детерминированными законами, а также стохастические гибридные системы, вероятностные автоматы, системы со стохастическим управлением и т. п. Кроме того, в статье Граничина и Хантулевой (2004) [17] показан пример эффективности использования рандомизированных программных воздействий на систему в условиях динамически изменяющейся структуры пространства состояний. Рандомизация позволяет частично устранить влияние на работу системы систематических погрешностей (Граничин (2004)) [18,19], которые практически неизбежны при изменяющейся со временем модели динамической системы.

5. Дальнейшее развитие

Предложенная новая модель вычислений позволяет описывать если не все, то подавляющее большинство процессов реального мира, а также работу всевозможных существующих и будущих вычислительных устройств, включая аналоговые и биокомпьютеры, нейрокомпьютеры, квантовые компьютеры и т.д. Особенностью предлагаемого подхода является отказ от редукции сложности в процессе вычисления. Сложность вычислимого объекта должна быть эквивалентна сложности вычисляемого.

Таким образом, понятие вычислительной сложности правильнее рассматривать относительно выбранной системы базисных эволюционных примитивов, а не относительно традиционно рассматриваемых битовых преобразований $\{0,1\}$. Квантовые и нейрокомпьютеры обещают сильно изменить представления о вычислительной мощности современных вычислительных устройств. Увеличение вычислительной мощности, возможное за счет использования новых моделей вычислений, основывающихся на физических явлениях, позволяет предположить, что в будущем новые компьютеры смогут решать задачи, невыполнимые для обычных компьютеров.

6. Список литературы

[1] Moore, H., (1965). *Electronics, 38.*

[2] Turing, A.M., (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 2,* 230-265.

[3] Feynman, R., (1982). Modeling of physics on computers. *International Journal of Theoretical Physics,* 21.

[4] Deutsch, D., (1985). Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society*: *series A,* 97-117.

[5] Church, A., (1936). A note on the Entscheidung problem. *Journal of Symbolic Logic, 1,* 56-68.

[6] Copeland, J.B., (2004). The Church-Turing thesis. *NeuroQuantology,* 2, 101-115.

[7] da Costa, N.C.A., & Doria, F.A., (1991). Classical Physics and Penrose's Thesis. *Foundations of Physics Letters,* 4, 363-374.

[8] Doyle, J., (1982). What is Church's Thesis? An Outline. Laboratory for Computer Science, MIT.

[9] Hogarth, M.L., (1994). Non-Turing computers and Non-Turing computability. *PSA 1994, 1,* 126-138.

[10] Gao, Y. G., Lygeros, J., & Quincampoix, M., (2007). On the reachability problem for uncertain hybrid systems, *IEEE Transactions on Automatic Control, 52(9),* 1572–1586.

[11] Goebel, R., Sanfelice, R., G., &Teel, A. R., (2009). Hybrid dynamical systems, *IEEE Control Systems Magazine, 29(2),* 28–93.

[12] Shor, P., (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput. 26,* 1484-1509.

[13] Tien, D.K., (2003) Computing the non-computable, *Contemporary Physics, 44,* 51-71.

[13] Tien, D.K., (2003) Computing the non-computable, *Contemporary Physics, 44,* 51-71.

[14] Вахитов А.Т., Граничин О.Н., Сысоев С.С. Точность оценивания рандомизированного алгоритма стохастической оптимизации // Автоматика и телемеханика, 2006, № 4 , с. 86-96.

[15] Граничин О.Н., Жувикова И.А., (2006) Новая модель процесса вычислений, основанная на эволюционных примитивах: обобщение концепции машины Тьюринга. Нейрокомпьютеры: разработка, применение.

[16] Hopcroft, J.E., & Ullman, J.D., (1979). Introduction to Automata Theory, Languages, and Computation.

[17] Granichin, O.N., & Khantuleva, T.A., (2004). Hybrid systems and randomized measuring in nonequilibrium processes, *Differential Equation and Control Processes,* 3.

[18] Granichin, O.N., (2004). Linear regression and filtering under nonstandard assumptions (Arbitrary noise), *IEEE Transactions on Automatic Control, 49 (10)*, 1830-1835.

[19] Граничин О.Н., Поляк Б.Т. “Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах”. М.:Наука, 2003. 291с.

Hybrid model of the computing process: generalization of the Turing machine concept

Abstract

Paper proposes generalization of classical Turing machine scheme. New concept generalize traditional concepts of "tape" and "tape cell". In particular, "tape cell" represents permanently running model of some dynamical system. "Natural" evolutions of cells proceed with jumpings. This model, for example, can describe systems with variable state spaces. Using this model one can avoid problem of simulation continuous dynamic in the discrete time environment. Classical Turing machine is an extreme case of the proposed model. In addition paper describes possibilities of application to simulate biochemical processes.

Сведения об авторах:

Граничин Олег Николаевич
звание:
доктор физ.-мат. наук, профессор;
должность:
профессор кафедры системного программирования математико-механического факультета, СПбГУ;
заведующий лабораторией стохастических вычислительных систем института информационных технологий;
область научных интересов:
теоретическая кибернетика, адаптивное и оптимизационное управление, квантовые компьютеры, рандомизированные алгоритмы.
контакты:
дом. 323-01-99 моб. +7-921-740-03-37, раб. 428-49-10 e-mail: Oleg_granichin@mail.ru

Васильев Валентин Игоревич
аспирант кафедры СПбГУ;
область научных интересов:
теоретическая кибернетика, пиринговые сети, мультиагентные системы, рандомизированные алгоритмы.
контакты:
дом. 378-20-85 моб. +7-952-242-16-68 e-mail: gnome@bk.ru