

УДК 004

О НЕКОТОРЫХ РЕЗЕРВАХ И НАПРАВЛЕНИЯХ РАЗВИТИЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

Бельтюков Анатолий Петрович, доктор физико-математических наук, профессор, заведующий кафедрой теоретических основ информатики ФГБОУ ВО «Удмуртский государственный университет»

Маслов Сергей Геннадьевич, кандидат технических наук, доцент, доцент кафедры теоретических основ информатики ФГБОУ ВО «Удмуртский государственный университет»

Аннотация

В работе рассматриваются нетрадиционные точки зрения на высокопроизводительные вычисления. Формируется система понятий, которая расширяет классы задач для высокопроизводительных вычислений и подходы к их решению. При этом предлагается рассматривать физико-антропо-технические системы и сети как среду организации высокопроизводительного компьютеринга. На верхнем уровне организации систем постановок задач предлагается рассматривать семантику декартова замкнутых категорий. Представленная точка зрения может служить основой для всестороннего критического обсуждения.

КЛЮЧЕВЫЕ СЛОВА: компьютеринг, высокопроизводительные вычисления, эргатические и физико-антропо-технические системы и сети, система знаний, онтология, дедуктивный синтез алгоритмов, теория категорий, декартова замкнутость, конструктивное доказательство, автоматический логический вывод.

ON SOME RESERVES AND DIRECTIONS OF DEVELOPMENT OF HIGH-PERFORMANCE COMPUTING

Beltiukov Anatoly Petrovich, Doctor of Physical and Mathematical Sciences, professor, head of Department of theoretical informatics basis of the Udmurt State University Department

Maslov Sergey Gennadyevich, PhD in Technical Sciences, associate professor of Department of theoretical informatics basis of the Udmurt State University

Abstract

The paper considers non-traditional points of view on high-performance computing. A system of concepts is being formed that expands the classes of problems for high-performance computing and approaches to their solution. At the same time, it is proposed to consider physical-anthropological-technical systems and networks as an environment for organizing high-performance computing. At the upper level of the organization of task setting systems, it is proposed to consider the semantics of Cartesian closed categories. The presented point of view can serve as the basis for a comprehensive critical discussion.

KEYWORDS: computing, high-performance computing, ergatic and physical-anthropological-technical systems and networks, knowledge system, ontology, deductive synthesis of algorithms, category theory, Cartesian closure, constructive proof, automatic logical inference.

Введение

Один из основных барьеров в высокопроизводительных вычислениях можно выразить в форме следующего высказывания: «Вычисляется не то, что нужно, а то, что можно». Текущий этап развития высокопроизводительных вычислений характеризуется тем, что требуется системное переосмысление направлений их развития, поиска резервов и новых подходов. Фактически мы находимся в тупике. Один из индикаторов этого тупика — исчерпание физических возможностей осуществления «закона Мура». Это связано с

димензиальной (размерностной) недостаточностью, то есть недостаточностью существующих направлений. В основном рассматривают традиционную физическую основу компьютеринга. Требуется выйти за пределы этих ограничений. В работе анализируются различные точки зрения и составляющие проблемы, которые призваны выявить перспективные резервы для формирования новых подходов к ее решению.

Текущее положение дел в исследованиях рассматриваемой области

Современные вычисления реализуются в разных физических средах (см., например [1]). Кроме того, ведутся эксперименты по использованию для вычислений биологических сред (вычисления на грибнице и на других живых организмах [2, 3]), есть идеи использования биохимических процессов (ДНК-компьютеринг и др. [4, 5]). Решается и обратная задача: моделирование биологического компьютеринга, т. е. моделирование функционирования нервной системы. Кроме того, принципы организации компьютеринга меняются, например, используются такие физические приборы как различные виды квантовых компьютеров. Природа аппликативного компьютеринга исследовалась в работах [6-13], в том числе, исследовался и компьютеринг в естественных системах. Конструктивно-логическая природа компьютеринга затрагивалась в работах [14-17]. Вопросы онтологии компьютеринга для технических систем рассматривались в работе [18]. Вопросы автоматизации синтеза программ в компьютеринге рассматривались в работах [19-25]. Все эти работы являются предпосылками настоящего исследования.

О вычислениях и компьютеринге

Прежде всего, необходимо расширенное понимание вычислений, т. е. требуется не просто работа с количественными характеристиками, но и с качественными, когда включаются все основные процессы: поиск, навигация, логический вывод, построение, самоорганизация, преобразование и управление. В этом случае, от термина «вычисление» целесообразно перейти к термину «компьютеринг» (как к обобщению понятия вычисления).

О среде компьютеринга

Среду компьютеринга необходимо рассматривать не только с физической точки зрения, но и с химической, биологической, кибернетической и либернетической (с точки зрения степеней свободы). Следует рассматривать среду компьютеринга как аналог «живой» системы и с учетом масштабируемости, локальности и глобальности, цифровых и аналоговых преобразований (например, с использованием фотоники и нанофотоники [1]). В этом в контексте выявляются границы системы, а для решения жизненно важных проблем, по необходимости, активизируется расширяющаяся среда конструктивного построения и

процессы самоорганизации. Кроме того, сам процесс решения задачи конкретизируется в виде развивающейся структуры постановок задач, которые решаются не просто эффективно, надежно и безопасно, а в первую очередь — своевременно. Разнообразие среды компьютеринга приводит к его расслоению и распределению, которые строятся на аттрактивных (притягивающих) и детрактивных (отталкивающих) связях, опережающем синтезе и самоорганизации.

На компьютеринг, как и на любой другой объект, можно смотреть с разных точек зрения. Это делается, чтобы наиболее полно выявить его свойства для выявления и решения различных проблем. Например, можно рассматривать компьютеринг, с одной стороны, как преобразование дискретных объектов, с другой стороны, как преобразование протяжённых объектов. Информационно-математическая сущность протяжённости — в наличии инвариантов, отношений близости и вложенности. Требование непрерывности протяжённого компьютеринга заключается в сохранении этих инвариантов, а нарушения их — это разрывы в компьютеринге. При учёте непрерывности в компьютеринге требуется адекватная детализация, учёт точности при принятии решений в условиях альтернативных возможностей. Зачастую использование некоторых специфических сред для компьютеринга приводит к автоматической поддержке такой непрерывности, соблюдение которой требуется для решения некоторых задач. К тому же, игнорирование требований непрерывности часто приводит к чрезмерной сложности компьютеринга. По сути требование непрерывности при решении многих задач диктуется необходимостью поддержать «природоподобную» структуру объектов компьютеринга.

Отдельное внимание следует уделить единству количественных и качественных аспектов компьютеринга, которые устраняют проблему бессмысленных вычислений и разумного использования ресурсов.

Об идеальном и материальном в компьютеринге

Человеческая мысль до сих пор эффективно формулируется только непосредственно человеком, личностью, которая значительную часть своего времени затратила на усвоение опыта человечества и обладает достаточным любопытством и творческими способностями. Однако, значительная часть идей и мыслей остаются нереализованными. Частично это можно объяснить тем, что отсутствуют хорошо проработанные теории и технологии непрерывного перехода от идеального к материальному, отсутствуют адекватные оценки для действительно перспективных решений, большинство из которых формируется не в рамках мейнстримовых (общепринятых) направлений развития. В этом случае, востребованными становятся

компьютерные модели, которые способны учитывать скорости, точности и ресурсоемкость решений (или цену и ценность) и их влияние на изменение среды и доступных ресурсов в будущем. А это позволяет более точно определять моменты появления новых проблем и потребностей, действительно улучшающих качество жизни. Знания не устаревают, а становятся непригодными для применения в новой сложившейся ситуации. Никто не гарантирует, что ситуация не может повториться, тогда «старые» знания вновь будут востребованы.

О формах и системе знаний

Реальное решение жизненно важных проблем базируется на знаниях и творчестве, которые на современном уровне неизбежно приводят к синтезу новых знаний, возникающих в конвергенции искусства, науки и инженерии. В этом случае важную роль играют формы представления знаний, процессы формализации и деформализации, а также — различные виды симбиоза, полисенсорности и синестезии, которые позволяют строить конструктивный процесс на основе компактных и адекватных средств для решения возникающих проблем. Особая роль отводится взаимосвязи форм знаний, которая строится на прямых и обратных, обратимых и необратимых преобразованиях, эффективно отражающих различные субъективные и объективные стороны системы знаний, результатов и процессов их получения. Это позволяет проявлять как индивидуальные, так и коллективные возможности субъектов.

Переход к эргатическим и физико-антропо-техническим системам и сетям

Ключевую роль в решении жизненно важных проблем играют процессы понимания, разумного вовлечения и использования ресурсов и новых средств обработки информации.

Существующее жесткое разделение процессов обработки информации между человеком и компьютером, часто сопровождается порождением чрезмерно больших данных и потерей понимания и ответственности за полученные результаты.

Поскольку главным поставщиком целей и использования результатов их достижения является человек, то необходимы не просто частные или локализованные результаты вычислений, а процессы, соответствующие решению жизненно важных проблем при непрерывном переходе от идеи до её воплощения, через абстрагирование и материализацию. В этой ситуации, человеку необходим не просто заменитель его функций, а реальная поддержка в достижении целей, способная повышать его интеллектуальный и физический потенциал, а также — гармонизировать коллективные усилия в решении проблем.

Это приводит к развитию высокопроизводительных вычислений в рамках эргатических (человеко-машинных) и физико-антропо-технических (природно-человеко-искусственных) систем и сетей. Здесь решаются вопросы более глубокого понимания, формирования расслоений и распределений, необходимых уровней абстракций, а также эмоциональных механизмов концентрации внимания и мотивации поведения и жизнедеятельности человека и общества при гармонизации их отношений с природой и искусственной средой обитания. Под расслоениями понимается построение моделей одного и того же объекта с различных позиций, обладающих собственными семантиками компьютеринга, а под распределением – разделение модели на взаимодействующие части. Когнитивные и конструктивные процессы, в этом случае, могут активно использовать положительные и отрицательные эффекты иллюзий.

В физико-антропо-технических системах имеется собственное содержание интерпретации коллаборативных, антагонистических и конкурентных процессов: победитель конкурентных процессов (сотрудничества) не только получает решение, но и может использовать альтернативные (побеждённые) решения для повышения надёжности и безопасности. Для антагонистических процессов должен строиться аналог иммунной системы, чтобы препятствовать их взаимоуничтожению и достигать стабильной жизнеспособности системы. Коллаборативные процессы существенно снижают сложность решения задачи.

Пример спецификации внешних аппликативных объектов: элементы управления системой связи

Этот пример иллюстрирует соединение разнородных объектов, в которых могут участвовать технические системы, люди, природные объекты с требуемой эффективностью, надёжностью организации потоков информации. При этом высоконагружаемыми могут оказаться самые разные элементы, гарантирующие необходимые специфицируемые результаты вычислений.

Далее для иллюстрации описываем модельную ситуацию конструирования системы управления связью (передачей сигналов, данных, сообщений). Считаем, что у нас имеется некоторая программируемая система компьютеринга, оборудование, интерфейсы для управления им и для анализа отклика этого оборудования.

Пусть ситуация может быть формализована следующими предложениями (здесь мы используем некоторый конструктивно-логический язык спецификаций задач, неформально поясняя на ходу смысл употребляемых выражений):

- *Node :Type* — тип значения: «имя узла связи»,

- *Term: Pred(Node), Term(n)* — тип значения: «спецификация терминального оборудования, которое находится в узле связи n », где $n : Node$;
- *Base:Pred(Node), Base(n)* — тип значения: «спецификация базовой станции, которая может находиться в узле связи n », где $n : Node$;
- *Connect: Pred (Node, Node), Connect (m, n)* — тип значения: «команда к системе связи, которая позволяет установить канал связи от узла m к узлу n », где $m, n : Node$;
- *Error :Pred (Node), Error (n)* — тип значения: «сообщение об ошибке в узле связи n », где $n : Node$;
- *base_search:(n:Node, Term(n)=>Error(n) | m:Node, Base(m), Connect(n,m))* — «*base_search* — имя объекта, выдающего по терминальному узлу связи сообщение об ошибке, или находящего базовую станцию, с которой может связаться этот узел», при обычной интерпретации такой объект работает с внешним миром: пытается найти базовую станцию, доступную из данного узла, а иначе формирует сообщение об обнаруженной ошибке, обычно это что-нибудь типа: «терминал находится вне зоны доступа».

Объект *base_search* в данной ситуации обычно — технико-природный объект, состоящий из оборудования, соединённого с управляющей системой, и среды передачи сигналов (она может полностью природной, без технических элементов).

Для того, чтобы система управления была работоспособна (в смысле связывания доступных терминальных узлов), требуется ещё три объекта:

- *base-connect: (m, n :Node, Base (m), Base (n) => Connect (m,n))* — объект соединения базовых станций между собой (для простоты примера ошибки работы сети не рассматриваются),
- *trunc:(m,n,k: Node, Base(n), Connect(m,n), Connect(n,k)=>Connect(m,k))* — объект «проключающий» соединение через промежуточный узел, т.е. осуществляющий соединение каналов,
- *revers:(m, n : Node, C(m,n) => C(n,m))* — объект, осуществляющий реверс соединения, превращающий симплексное соединение в дуплексное (понятно, что не во всех реальных системах связи такой объект доступен, но в нашем примере он есть).

Система управления, которой доступны эти объекты, может решать задачу:

term_connect :(m,n :Node, Term (m), Term (n) =>Connect (m,n) | Error (m) | Error (n)),

то есть своим процессом работы будет эмулировать объект *term_connect* с данной спецификацией. Надо отметить, что программа для *term_connect* легко автоматически

порождается методом дедуктивного синтеза алгоритмов. Пример такой программы на языке дедуктивного программирования:

```
term_connect =
{m,n, tm, tn, -- входы
 ocmn, -- нормальное завершение
 oem, oen; -- завершения с ошибками
 base_search (m,tm){em; oem (em)} /* недоступность первого терминала */
 i,bi,cmi; -- поиск первой базы
 base_search (n, tn){en, oen (en)} /* недоступность второго терминала */
 j, bj, cnj; -- поиск второй базы
 base_connect (i, j, bi, bj ) = cij ; -- соединение баз
 trunc (m,i,j, bi, cmi, cij ) = cmj ; -- первое проключение
 revers (n, j, cnj ) = cjn ;
 trunc (m, j, n, cmj, cjn) = cmn ; -- окончательное соединение
 ocmn (cmn) -- нормальное завершение
}
```

Это простой начальный пример. Далее он может эволюционировать для разрешения задач множественных соединений, оптимизации соединений и т.д.

Физико-антропо-технические системы и сети с точки зрения декартово замкнутых категорий

Широко используемая практика терминологических систем и онтологий нуждается в существенном пересмотре, поскольку она плохо отражает процессы расслоения и распределения знаний, а также уязвима с точки зрения безопасности и страдает отсутствием компактности.

Предлагается подход к описанию предметных областей решаемых конструктивных задач в виде конструктивной онтологии на уровне абстракции декартово замкнутых категорий. Конструктивность онтологии даёт возможность корректного автоматического или автоматизированного (человеко-машинного) построения решений задач. По идее онтология строится так, что она сама решает поставленные задачи.

Предлагаемая парадигма строится на основе парадигмы дедуктивного синтеза алгоритмов. разница состоит в том, что синтезируемые алгоритмы получаются как морфизмы декартово замкнутой категории. В сущности, онтология призвана описывать предметную область как декартово замкнутую категорию. При этом первоначально объектами категории становится то, что традиционно мыслится как классы объектов предметной области. Элементы большинства самих этих классов на первоначальном этапе часто не рассматриваются.

Для конструктивности онтология снабжается логической машиной построения вывода. Эта машина мыслится как неотъемлемая часть самой конструктивной онтологии. С практической точки зрения для надёжности важно иметь независимые логическое исчисление и машину построения вывода. Эти механизмы контролируют друг друга. Результат работы логической машины проверяется исчислением.

В настоящей работе мы рассматриваем подход к решению задач информатики в рамках декартово замкнутых категорий. Идеи использования декартово замкнутых категорий для программирования рассматриваются уже сравнительно давно. Эти идеи тесно основаны на изоморфизме Карри-Ховарда и тесно связаны с лямбда-исчислением с типами.

В теоретико-категориальном подходе в качестве доменов предметной области рассматриваются абстрактные объекты, структура которых первоначально не уточняется. С точки зрения программирования такой объект категории рассматривается не как множество возможных значений при теоретико-множественном подходе, а как интерфейс между программами, позволяющий осуществлять их композицию. Связь с теоретико-множественной моделью состоит в том, что упомянутые интерфейсы используются для передачи данных, кодирующих элементы множеств, соответствующих объектам.

Имеющиеся и создаваемые программы на этом языке являются морфизмами, соединяющими некоторые объекты. Действие морфизма не ограничивается отображением значения из исходного объекта в конечный, как в функциональном программировании.

Это может быть сложное взаимодействие с передачей сигналов, данных и программ во встречных направлениях. Например, программа может произвести несколько попыток выдачи результата, получая сигналы об ошибках применения результатов предыдущих попыток.

Пример формализации физико-антропо-технических систем и сетей с точки зрения декартово замкнутых категорий

Основные понятия

Считаем, что у нас имеются некоторые начальные объекты: $A, B, \dots$

Кроме того, между этими начальными объектами имеются связи, называемыми морфизмами:

$$f : A \rightarrow B, f : \text{Hom}(A, B), \dots$$

Здесь f — «морфизм из A в B ».

Вводится понятие «экспоненциального объекта» (результата возведения одного объекта в степень другого): $I = (A \Rightarrow B)$, здесь объект B возводится в степень A , « $\Rightarrow$ » — операция возведения в степень. Этот экспоненциальный объект состоит из морфизмов

$$f:(A \Rightarrow B) \Leftrightarrow f:A \rightarrow B$$

Также используются операции прямого (декартова) произведения:

$$(A \times B) = (A \& B)$$

и прямой суммы (объединения):

$$(A + B) = (A \vee B) = (A | B).$$

С точки зрения конструктивной (реализационной семантики) логики операция прямого произведения соответствует операции конъюнкции ($\&$, «И»), а операция прямой суммы — операции дизъюнкции ($\vee$, «ИЛИ»). В этом случае конструктивные логические значения понимаются как «множества реализаций», с которыми выполняются соответствующие операции.

Используются также понятия «семейств объектов»: $P, Q, R, \dots$

Объекты, получаемые из этих семейств:

$P(i), Q(i, j), \dots$ где $i, j, \dots$ — параметры семейств, как правило, это морфизмы.

Например, для $P(i)$ можно считать, что $i:I, I = (N \Rightarrow V)$, т.е. $i:N \rightarrow V$, i — морфизм из объекта N в объект V . К примеру, объект N — «характеристики», а объект V — «значения» параметра. Тогда сам морфизм i — некоторая характеристика параметра семейства.

Тогда имеет смысл рассматривать более сложные операции:

прямое суммирование:

$(x:I \& B(x))$, где $I = (N \Rightarrow V)$, оно соответствует в логике конструктивному квантору существования;

прямое мультиплицирование:

$(x:I \Rightarrow B(x))$, соответствует в логике конструктивному квантору общности.

С точки зрения компьютеринга объекты можно понимать как интерфейсы. У каждого объекта есть вход и выход. Это обозначается так:

$$in \rightarrow A \rightarrow out$$

Для экспоненциального объекта имеем:

$$in \rightarrow (A \Rightarrow B) \rightarrow out,$$

где in и out состоят из пар компонент.

Здесь $in = (-\rightarrow A, B \rightarrow)$ — активизация объекта (получает A и даёт B),

$out = (A \rightarrow, -\rightarrow B)$ — использование объекта (даёт A и получает B).

Конструктивная онтология состоит из следующих частей:

- 1) начальные объекты,
- 2) начальные семейства,

- 3) схемы аксиом: описания существующих морфизмов,
- 4) исчисление,
- 5) машина вывода.

Заметим, что здесь нами используется минимальная позитивная конструктивная логика, в которой отрицания и обработки ошибок интерпретируются следующим образом:

- интерпретация отрицания A :

$$(A \Rightarrow \text{Error})$$

- объект Error используется для обработки ошибок, при этом формула

$$(\text{Error} \Rightarrow A)$$

означает, что ошибки обрабатываются за пределами основной постановки задачи.

Пример конструктивной онтологии

Рассмотрим следующие начальные основные объекты:

$N : \text{Type}$ — индекс некоторой характеристики,

$V : \text{Type}$ — значение этой характеристики.

Пусть начальный экспоненциальный объект —

$I = (N \Rightarrow V)$ — “идентификаторы узлов связи”.

Рассмотрим следующие семейства:

$i : I \Rightarrow P(i) : \text{Type}$ — соединение узла i с основным сервисом

$i, j : I \Rightarrow Q(i, j) : \text{Type}$ — передача сервиса к узлу i от узла j

Рассмотрим следующие аксиомы – исходные морфизмы:

нахождение или передача соединения узла с некоторым сервисом:

$$fc : (i : I \Rightarrow P(i) + j : I \& Q(i, j)),$$

Выполнение передачи соединения от одного узла к другому:

$$ct : (i : I \Rightarrow j : I \Rightarrow Q(i, j) \Rightarrow P(j) \Rightarrow P(i))$$

Соединение на высшем уровне (достаточно длинная цепочка приводит к узлу, который может осуществить соединение):

$$hl : (i : I \Rightarrow j : I \Rightarrow k : I \Rightarrow Q(i, j) \Rightarrow Q(j, k) \Rightarrow P(k))$$

(здесь рассмотрены три уровня узлов, можно задать любое число уровней).

Формальное логическое исчисление строится на основе следующих положений.

- Используется минимальная позитивная конструктивная логика первого порядка (без отрицаний и аналогов закона исключённого третьего).
- Доказательства могут быть легко проверены известными компьютерными системами.

- Конструкции выведенных морфизмов также могут быть легко извлечены известными компьютерными системами.
- Используется машина логического вывода
- Проводится прямой поиск вывода от цели к аксиомам на ограниченную глубину. Он может быть легко выполнен системами логического программирования.

Пример задачи, которая может быть решена такой онтологией:

$goal : (i : I \Rightarrow P(i))$ — «Обслужить любой узел».

Решение (терм доказательства – программа) порождённое машиной логического вывода выглядит так:

$i : I;$
 $fc(i)\{P(i)\} \text{ or}$
 $j : I, Q(i,j);$
 $fc(j)\{P(j); ct(i,j, Q(i,j), P(j)) = P(i)\} \text{ or}$
 $k : I, Q(j,k);$
 $hl(i, j, k, Q(i,j), Q(j,k)) = P(k);$
 $ct(j, k, Q(j,k), P(k)) = P(j);$
 $ct(i, j, Q(i,j), P(j)) = P(i).$

Здесь *or* — разветвление в программе после применения морфизма, результирующий объект которого — прямая сумма.

Заключение

Рассмотренный в работе компьютеринг и условия его формирования выявляют резервы повышения его производительности, а также эффективности решения жизненно важных проблем. Это происходит за счёт согласования разнородных, расслоённых и распределённых форм информации, а также за счёт нетрадиционных форм распределения действий и используемых ресурсов между человеком, искусственными и природными объектами. В дальнейшем следует особое внимание уделить ключевой роли человека в формировании физико-антропо-технического компьютеринга. Ведущая роль человека выражается в принятии решений и инициализации разных процессов и объектов, в интерпретации результатов и ошибок компьютеринга. Процессы компьютеринга с участием человека становятся более осмысленными на стратегическом и тактическом уровнях. Человек строит и реализует неформализуемые процессы, управляет ранжированием и обработкой ошибок.

Решения, полученные в категориальной семантике, имеют стратегический и концептуальный характер. За счёт этого происходит рост производительности решения полезных задач.

Таким образом мы инициируем переход от состояния, когда человек — «*раб процесса*» и «*раб цифры*», к состоянию осмысленной деятельности в компьютеринге и развитию жизни.

Литература

1. Сойфер В.А. Фотоника и ее применение в сенсорике, обработке информации и управлении. XII мультikonференция по проблемам управления (МКПУ-2019): материалы XII мультikonференции (Дивноморское, Геленджик, 23–28 сентября 2019 г.): в 4 т. Южный федеральный университет [ред.: И.А. Каляев, В.Г. Пешехонов и др.]. – Ростовна-Дону; Таганрог: Издательство Южного федерального университета, Т.2, 2019. С. 24–26.
2. Adamatzky A. Towards fungal computer. Interface Focus. 2018 Dec 6;8(6):20180029. doi: 10.1098/rsfs.2018.0029. Epub 2018 Oct 19. PMID: 30443330; PMCID: PMC6227805.
3. LINKs – Special Issue 1 Unconventional Computing, Andrew Adamatzky (ed.), 2021, 111 p.
http://links-series.com/wp-content/revues/LINKs_series_Special_Issue_1-Unconventional_Computing-2021.pdf
4. Сахабутдинова А.Р., Михайленко К.И., Гарафутдинов Р.Р., Кирьянова О.Ю., Сагитова М.А., Сагитов А.М., Чемерис А.В. Небиологическое применение молекул ДНК // Биомика. 2019 Т.11(3). С. 344-377. DOI: 10.31301/2221-6197.bmcs.2019-28.
5. Паун Г., Розенберг Г., Саломеа А. ДНК-компьютер. Новая парадигма вычислений, М.: Мир, 2004. - 528 с.
6. Вольфенгаген В.Э., Исмаилова Л.Ю., Косиков С.В. Структура компьютеринга и конструирование вычислений // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. 2010. № 8. 9 с. <https://cyberleninka.ru/article/n/struktura-kompyutinga-i-konstruirovaniye-vychisleniya/viewer>
7. Curry, H. B., Feys, R. Combinatory Logic Vol. I. —Amsterdam: North-Holland, 1958.
8. Howard, W. A. The formulae-as-types notion of construction // To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism. — Boston: Academic Press, 1980. – pp. 479–490.
9. Curien P.-L. Categorical combinatory logic. – LNCS, 194, 1985, pp.139-51.
10. Roy L. Crole, Categories for Types, CambridgeUniversity Press, 1994.
11. Friedman H. Equality between functionals. LogicColl. '73, pp. 22-37, LNM 453, 1975.
12. Barendregt H. Lambda Calculi with Types, Handbook of Logic in Computer Science, Volume II, Oxford University Press.

13. Cousineau G., Curien PL., Mauny M. (1985) The categorical abstract machine. In: Jouannaud JP. (eds) Functional Programming Languages and Computer Architecture. FPCA 1985. Lecture Notes in Computer Science, vol 201. Springer, Berlin, Heidelberg, pp. 50-64.
14. S.C. Kleene Introduction to Metamathematics North-Holland Pub. Co., 1950. 550 p.
15. Alonzo Church, Introduction to mathematical logic, Vol. 1 Princeton mathematical series, Annals of Mathematics Studies, (Vol. 17) Princeton University Press, 1956. 378 p.
16. Helena Rasiowa, Roman Sikorski, The Mathematics of Metamathematics. Panstwowe Wydawnictwo Naukowe, Warszawa, January 1, 1963, 520 p.
17. Bartosz Milewski, Category Theory for Programmers. Scala Edition, Oct 21, 2018. 396 Pages.
18. G.G. Kulikov, V. V. Antonov, A.R. Fakhru'llina, L.E. Rodionova. Method of structuring the self-organized intellectual system on the basis of requirements of the ISO/IEC 15288 standard in the form of the Cartesian closed category. (On the example of design of information and analytical system) // The 20th International Workshop on Computer Science and Information Technologies - 20th CSIT '2018. September 24-27, 2018, Bulgaria, Varna, pp. 135-139.
19. Alur, Rajeev; al., et. "Syntax-guided Synthesis". Proceedings of Formal Methods in Computer-Aided Design. 2013, IEEE. p.8.
20. Basin D., Deville Y., Flener P. Hamfelt, A., Nilsson J. F.: Synthesis of Programs in Computational Logic. In: Bruynooghe M., Lau K.-K. (Eds.) Program Development in CL, 2014, LNCS 3049: pp. 30–65.
21. Alpuente M., Cuenca-Ortega A., Escobar S., Meseguer J.: Partial Evaluation of Order-Sorted Equational Programs Modulo Axioms. In: Hermenegildo M., Lopez-Garcia P. (eds) LogicBased Program Synthesis and Transformation. LOPSTR 2016. Lecture Notes in Computer Science, vol 10184 (2017). Springer, Cham, pp. 3-20.
22. Albert E., Bezirgiannis N., de Boer F., Martin-Martin E.: A Formal, Resource Consumption-Preserving Translation of Actors to Haskell. In: Hermenegildo M., Lopez-Garcia P. (eds) Logic-Based Program Synthesis and Transformation. LOPSTR 2016. Lecture Notes in Computer Science, vol 10184 (2017). Springer, Cham, pp. 21-37.
23. De Angelis E., Fioravanti F., Meo M.C., Pettorossi A., Proietti M.: Verification of Time-Aware Business Processes Using Constrained Horn Clauses. In: Hermenegildo M., Lopez-Garcia P. (eds) Logic-Based Program Synthesis and Transformation. LOPSTR 2016. Lecture Notes in Computer Science, vol 10184 (2017). Springer, Cham, pp. 38-55.
24. Alpuente M., Pardo D., Villanueva A.: Symbolic Abstract Contract Synthesis in a Rewriting Framework. In: Hermenegildo M., Lopez-Garcia P. (eds) Logic-Based Program Synthesis and

- Transformation. LOPSTR 2016. Lecture Notes in Computer Science, vol 10184 (2017). Springer, Cham, pp. 187-202.
25. Tarau P.: A Hiking Trip Through the Orders of Magnitude: Deriving Efficient Generators for Closed Simply-Typed Lambda Terms and Normal Forms. In: Hermenegildo M., LopezGarcia P. (eds) Logic-Based Program Synthesis and Transformation. LOPSTR 2016. Lecture Notes in Computer Science, vol 10184 (2017). Springer, Cham, pp. 240-255.