

Статья. Библиографические данные:

Житников А.П.
Исторические области и парадоксы
параллельной (и последовательной) алгоритмики.
// Современные технологии преподавания естественно-научных дисциплин
в системе общего и профессионального образования.
Сб. матер. Междунар. науч.-практ. форума.
– Борисоглебск: БГПИ, 2012. – С. 80-94.

ИСТОРИЧЕСКИЕ ОБЛАСТИ И ПАРАДОКСЫ ПАРАЛЛЕЛЬНОЙ (И ПОСЛЕДОВАТЕЛЬНОЙ) АЛГОРИТМИКИ

Представлен электронный вариант оригинала статьи.
Статья приводится с технической доработкой:
гиперссылки, оглавление, цветовые элементы и т.п.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	2
1 ОБЩИЕ ПРИКЛАДНЫЕ ОБЛАСТИ АЛГОРИТМИКИ.....	3
Исторические области массовой алгоритмизации.....	3
Система областей массовой алгоритмизации.....	3
Соотношение исторических областей алгоритмизации.....	5
Проблемы соотношения алгоритмического базиса и его надстройки.....	5
Следствия неопределенности человеческого фактора.....	7
2 АЛГОРИТМИЧЕСКИЕ ПАРАДОКСЫ В АРИФМЕТИЧЕСКОЙ ОБЛАСТИ.....	8
Исходные проблемные вопросы.....	8
Арифметические алгоритмы.....	8
Школьная проблематика.....	9
Алгоритмы параллельной арифметики.....	10
Пример алгоритмов параллельной арифметики.....	10
3 АЛГОРИТМИЧЕСКИЕ ПАРАДОКСЫ МАТЕМАТИЧЕСКОЙ ОБЛАСТИ.....	12
Проблема общего содержания теории алгоритмов.....	12
Фундаментальная (классическая) теория алгоритмов.....	13
Прикладная (неклассическая) теория алгоритмов.....	13
Неизвестная прикладная (структурная) теория алгоритмов.....	14
Антиалгоритмические настроения.....	14
4 АЛГОРИТМИЧЕСКИЕ ПАРАДОКСЫ В ТЕХНИЧЕСКОЙ ОБЛАСТИ.....	16
ЗАКЛЮЧЕНИЕ.....	16
ЛИТЕРАТУРА.....	17

ВВЕДЕНИЕ

В статье кратко анализируется **комплекс проблем** достаточно **парадоксального типа** в области **прикладной (структурной) теории параллельных (и последовательных) алгоритмов** (затронутый в статье [1]).

Исходной общей парадоксальной проблемой является:

отсутствие существенного по масштабам **внедрения** теории параллельных (и последовательных) алгоритмов **в массовую практику** разработок и пользовательского применения автоматизированных (и автоматических) систем разного уровня и сложности (начиная от учебных рабочих мест).

Парадоксальность этой проблемы заключается в том, что она существует при наличии (и избытии) многочисленных частных сходных и разных алгоритмических языков, теорий, методов, подходов и школ.

Далее приводится краткий анализ общего состояния **параллельных (и последовательных) алгоритмических систем**, включая:

1) Параллельные (и последовательные) **алгоритмы – предписания** некоторому (единичному или групповому) исполнителю:

это **знаковые системы**, представляющие собой литерные или графические тексты в системе разных алгоритмических языков и их грамматик.

2) **Системы реализации** (воплощения, осуществления) алгоритмов. Выделяются системы реализации двух взаимосвязанных видов:

а) **Процессы** определенной последовательной или параллельной структуры во времени, непосредственно предопределяемые алгоритмами.

б) Выполняющие их **объекты** определенной структуры, предопределяемые или предполагаемые в текстах и контекстах алгоритмов:

программная, аппаратная, организационная (персональная) реализация разных уровней автоматизации и технического оснащения (начиная от нуля и выше – от умственных и ручных алгоритмических систем и т.д.).

Этот анализ привязывается к системе последовательно расширяющихся исторических прикладных **областей массовой алгоритмизации**, которые кратко анализируются в данной статье.

1 ОБЩИЕ ПРИКЛАДНЫЕ ОБЛАСТИ АЛГОРИТМИКИ

Исторические области массовой алгоритмизации

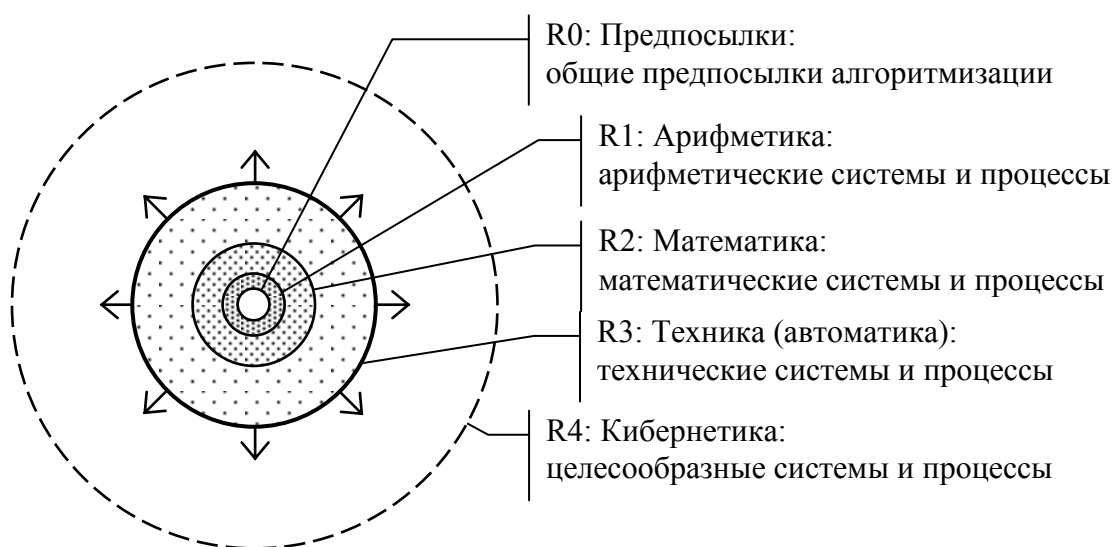
Алгоритмика (параллельная и последовательная алгоритмика) рассматривается далее в целом как **область назначения** – применения, существования и развития (параллельных и последовательных) **алгоритмических систем** и их **концептуального и теоретического обеспечения**.

Опыт анализа разных причинных обстоятельств и проблем параллельной (и последовательной) алгоритмики приводит к необходимости выделения основных частных **исторических областей массовой алгоритмизации**, порядка их появления, формирования и развития, их взаимосвязи и взаимодействия. Такие области представляют следующие аспекты:

- **господствующий тип алгоритмических систем** в некоторый характерный интервал времени;
 - соответствующие этому **господствующие алгоритмические представления** относительно систем такого массового типа:
- от интуитивных представлений до наличных теоретических наработок.

Система областей массовой алгоритмизации

Первичный анализ данного вопроса четко выявляет **общий процесс развития** массовой алгоритмизации в порядке формирования следующих последовательно расширяющихся **областей** (regions) прикладной **массовой алгоритмизации** (Рис. 1.1):



$R1 \rightarrow R2 \rightarrow R3 \rightarrow R4$: последовательное появление областей
 $R1 \subset R2 \subset R3 \subset R4$: последовательное включение областей

Рис. 1.1. Области массовой алгоритмизации

R0: Подготовка: Общая подготовка массовой алгоритмизации: она содержит предпосылки и алгоритмические прототипы всех последующих областей и направлений формирования алгоритмики, поскольку массовая алгоритмизация определенного типа не начинается внезапно, а всегда имеет какие-то более ранние исторические предпосылки и прототипы, причем корневого исторического причинного значения.

R1: Арифметика: Алгоритмизация арифметики.

Это **исторически первичная**, узкая по классу прикладных алгоритмических систем и высокоспециализированная область массовой алгоритмизации арифметической деятельности человека. Она включает в себя арифметические алгоритмические системы всех уровней автоматизации (от нуля автоматизации и выше):

- от древних **ручных (и умственных) систем** выполнения человеком арифметических операций **последовательного действия**: пальцевой счет, вычисления на абаке и т.п.;
- до современных электронных систем **параллельной компьютерной** (и, в более широком плане, **микропроцессорной**) **арифметики**: с параллельной обработкой числовых разрядов многоразрядных регистров с последовательными и параллельными межразрядными переносами.

R2: **Математика** (включая арифметику): Алгоритмизация математики.

Это **исторически вторичная** качественно новая, очень широкая и разнообразная область массовой алгоритмизации деятельности человека. Она включает в себя арифметику как некоторый особый (базисный) частный случай математики, но **на порядки превышает** ее по масштабам и разнообразию (как ее надстройка). Она включает в себя математические алгоритмические системы также всех уровней автоматизации (от нуля и выше):

- от древних и средневековых **ручных систем** решения математических задач **последовательного действия**: математика на глиняных табличках и папирусе и т.п., и затем, человеко-бумажные системы с ручным решением математических задач на бумажных носителях;
- до современных систем **параллельной компьютерной математики**: на машинных носителях, локальных и глобальных сетевых масштабах.

Это наиболее **развитая и господствующая** область приложений теории алгоритмов в настоящее время.

R3: **Техника** (и автоматика, включая технику математики и арифметики): Алгоритмизация техники.

Техника некоторой деятельности интерпретируется в предельно широком смысле как высокоорганизованное **профессиональное искусство** выполнения этой деятельности, включая **совокупность методов и средств** (сумму технологий) ее выполнения и, в частности, используемый **комплекс средств технического оснащения**.

Это **исторически третичная** еще более широкая и разнообразная область массовой алгоритмизации. Она включает в себя алгоритмизацию математики (и арифметики) как некоторый особый (базисный) частный случай техники и на порядки превышает ее по масштабам и разнообразию (как ее надстройка). Она охватывает технические алгоритмические системы всех уровней автоматизации (от нуля):

- от **ручных** и **механизированных** систем;
- до **автоматизированных систем** разных видов и масштабов: от древнегреческих ритуальных и театральных автоматов последовательного и параллельного действия, средневековых многофункциональных часовых механизмов и андроидов и т.д.; до современной систем технической и технологической автоматизации, промышленных и других робототехнических систем, гибких производственных модулей и систем, наземных, водных, воздушных и космических транспортных систем с их инфраструктурой и т.п.

Это **активно формируемая область приложений** теории алгоритмов. Однако:

- она еще не представляет собой самостоятельное направление техническое направление теории;

- в значительной мере она пока подчиненна господствующим математическим алгоритмическим представлениям, концепциям и подходам.

R4: **Кибернетика**: Алгоритмика кибернетики целесообразных систем (включая технические системы, как особый частный случай целесообразных систем) – перспективная предельно широкая общая кибернетическая область алгоритмизации.

Она находится в опытно-поисковой стадии накопления алгоритмических приложений по многим разным фронтам – в целом идет ее подготовка, но пока трудно говорить о самостоятельном таком направлении.

Соотношение исторических областей алгоритмизации

Исторические прикладные области массовой алгоритмизации **появляются последовательно** в перечисленном порядке:

$R1 \rightarrow R2 \rightarrow R3 \rightarrow R4$.

Каждая последующая область алгоритмизации **расширяет и включает** в себя исторически предшествующую область:

$R1 \subset R2 \subset R3 \subset R4$.

Каждая очередная область алгоритмизации $R_i = R(i)$ (начиная с математики) включает в себя (Рис. 1.2):

- исторически предшествующую область $R(i-1)$ как свой **базис**;
- дополнение базиса $R(i) \setminus R(i-1)$ как его **надстройку**.



Рис. 1.2. Базисы и надстройки исторических областей алгоритмизации

Проблемы соотношения алгоритмического базиса и его надстройки

Возникают неясные пока вопросы ключевой концептуальной и теоретической значимости следующего типа:

- базис является особым частным случаем более широкой охватывающей его области, но в чем заключается его принципиальное частное отличие?
- надстройка базиса в составе общей области алгоритмизации также является частным случаем более широкой охватывающей его области, но в чем заключается ее принципиальное частное отличие?
- чем принципиально различаются базис и его надстройка в каждой такой области?

Теория алгоритмов пока не дает ответы на такие вопросы. Это связано с тем обстоятельством, что сама теория алгоритмов находится в стадии накопления отдельных частных алгоритмических проблем и их решений.

Предварительно можно привести следующие ответы:

1) Различия арифметического базиса и математической надстройки:

а) Различия по уровню иерархии арифметических операций:

- **арифметические операции** в арифметическом базисе представлены последовательными и параллельными **арифметическими алгоритмами**: они представлены в некоторой конкретной системе счисления;
- но в математической надстройке они представлены как **элементарные арифметические функции**, и при этом (как обычно в теории): они представляются (формулами) безотносительно к системе счисления.

б) Различия в компьютерной реализации:

- **алгоритмы** арифметических операций в арифметическом базисе **реализуются аппаратно или микропрограммно** в управляющем устройстве центрального процессора ЭВМ (во взаимодействии с его арифметико-логическим устройством);
- а в математической надстройке они **представлены простыми арифметическими операторами** программ в оперативной памяти ЭВМ: программист может ими пользоваться, но не имеет доступа к их аппаратной или микропрограммной реализации (не может их изменять).

2) Различия математического базиса и его технической надстройки:

- в математическом базисе **математика** в алгоритмах (и их реализации) является **и целью и средством** (инструментом) достижения цели;
- в технической надстройке **математические результаты не являются целью** выполнения алгоритмов (и их реализации) – здесь математика является инструментом и теряет свои прежние целевые функции: это отличие интуитивно понятно, но пока неясна суть этого факта, и что из этого следует в концептуальном и теоретическом плане;
- более того, есть технические алгоритмы, которые **не используют явно математику** (например, обычное правило перехода улицы – в технической транспортной среде), но непосредственно не непонятно, почему (и куда) **пропадает математический базис**.

На сайте **paralg1000.narod.ru** (или **paralg.narod.ru**) – сейчас на сайте **paralg.ucoz.com** предполагается:

анализ по этому поводу модельной программы (Рис. 1.3) реализации такого алгоритма (в модельной среде NetLogo), а также других технических алгоритмов (в среде Scratch, на языке Python, Java, C++ и т.п.).

3) Различия технического базиса и его надстройки в кибернетике:

- в техническом базисе алгоритмики существуют **ключевой и ведущий человеческий фактор**, обеспечивающий, в частности, **целесообразную организацию и целенаправленное поведение** технических систем;
- однако практически непонятно, что такое данный человеческий фактор по существу вопроса, что такое цели (в их соотношении с причинами), и как они представлены в технических алгоритмических системах, а современная теория алгоритмов ограничивается математическим аппаратом и не дает ответов на такие вопросы;
- в кибернетической алгоритмической надстройке ведущий **человеческий фактор отсутствует**, но практически непонятно как обеспечивается целесообразная организация и целенаправленное поведение биологических, психологических, генетических алгоритмических систем и т.п.:

теория алгоритмов не дает и пока не может дать ответы на такие вопросы, поскольку наука только-только вплотную подходит к ответам на них.

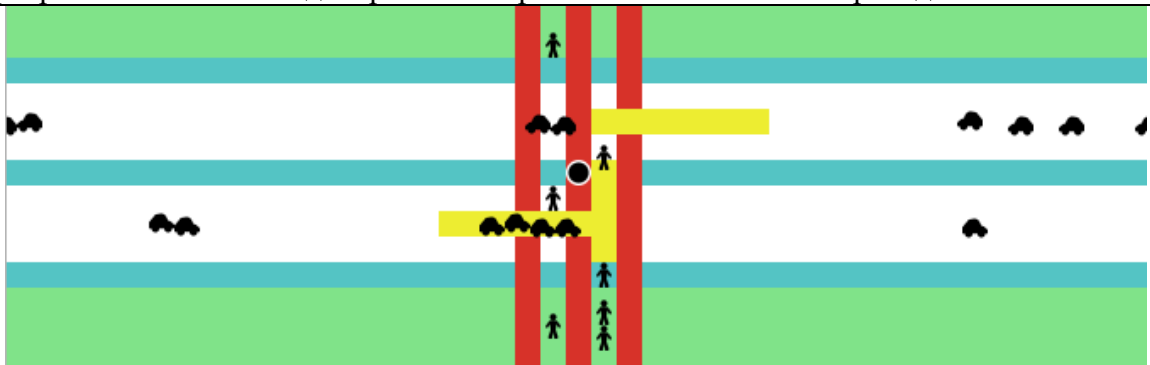
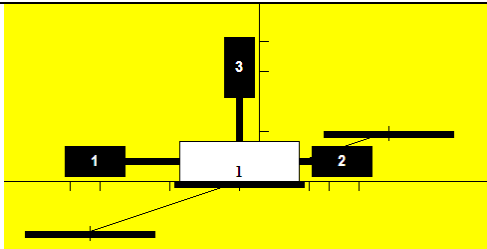
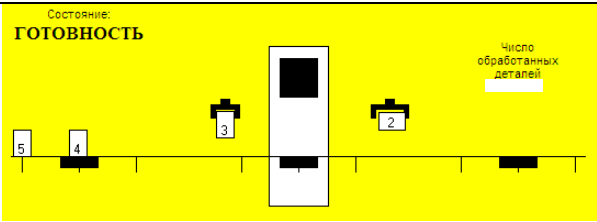
Программа Perehod – моделирование пересечения потоков на переходе	
	
Программа АТК: Агрегатный технологический комплекс	Программа РТК: Роботизированный технологический комплекс
	
Параллельная обработка деталей	Конвейерный режим продвижения деталей

Рис. 1.3. Модельные программы

Следствия неопределенности человеческого фактора

С этим обстоятельством связана такая парадоксальная ситуация:

- исторически более поздняя (формируемая) теория технических алгоритмов изначально более **предрасположена к конструктивному учету** технической (программной и аппаратной) **реализации алгоритмов**;
- более ранняя (реально существующая) теория математических алгоритмов, изначально ориентированная на человека (как исполнителя алгоритмов), по необходимости **абстрагировалась от** недоступной для того времени биологической (нейродинамической) **реализации алгоритмов**;

это обстоятельство объективно предрасполагает к историческому **алгоритмическому абстракционизму** (и алгоритмической мистике) первичной теории, который по традиции продолжает проявляться в разных формах.

2 АЛГОРИТМИЧЕСКИЕ ПАРАДОКСЫ В АРИФМЕТИЧЕСКОЙ ОБЛАСТИ

Исходные проблемные вопросы

Термин и понятие алгоритма, и первые сопутствующие алгоритмические представления возникли в арифметике еще в Средневековой Европе. Именно в арифметике сложилось **первичное понятие алгоритма**, как правила выполнения арифметической операции в десятичной системе счисления (алгоритмы сложения, вычитания, умножения и деления). А специальный термин **алгоритм (algorithm)** произошел от имени арабского ученого Аль-Хорезми (Из Хорезма, Мохаммеда Хорезмийского, algorizmi в латинской нотации). Он впервые в истории письменно изложил принципы индийского счета в позиционной десятичной системе счисления [2].

Однако до сих пор, фактически, не ясно, почему специальное **понятие алгоритма** и **специальный термин** для него (не важно, какой конкретно) **сложилось** именно:

- **в определенном отрезке средних веков** (XII – XVII вв.), а не ранее: десятичная система счисления уже существовала в Индии с V в. н.э. и затем к IX в. широко распространилась в арабском мире;
- **именно в Средневековой Европе**, а не в прежних областях: не в Индии, где десятичная система появилась, и не в Арабском мире, откуда десятичная система была заимствована в Европе в XII в., где с ней ознакомили по латинскому переводу учебника по индийской десятичной арифметике ученого Аль-Хорезми (написанного примерно в 825 году) и другим источникам (в частности «Книга абака» Фибоначчи 1202 г.).

Арифметическая сторона истории этих вопросов изучены достаточно хорошо [2-4], но пока нет специального их алгоритмического анализа, хотя там отражены определенные алгоритмические аспекты. И современная теория алгоритмов до сих пор не дает четких и конструктивных, общеизвестных и общепонятных ответов на эти принципиальные вопросы:

- это достаточно странная (парадоксальная) ситуация;
- представляет принципиальный интерес анализ причин ее наличия и, главное, самой возможности ее наличия в наши продвинутые алгоритмические времена.

Предположительно это было связано с появлением в Средневековой Европе компактных **письменных протоколов счета** – пошаговой записи результатов десятичных вычислений на бумаге (столбиком, уголком и т.п.).

Арифметические алгоритмы

Арифметический алгоритм, как арифметическое **правило** – это должно быть (в современном классическом понимании алгоритма) **строгое и точное предписание** выполнить поразрядные действия арифметической операции в определенном порядке.

Но таких строгих и точных правил арифметики не было и нет:

- ни в средние века – начиная с арифметического учебника Аль-Хорезми, его латинского перевода (и его современного русского перевода [2]);
- ни в наше время в массовом обучении арифметике и в массовой практике арифметических вычислений: обучение выполнению арифметических операций ведется длительно и чисто практически – на большой массе конкретных тренировочных примеров нарастающей разрядной сложности.

Другая связанная с этим парадоксальная ситуация:

- с одной стороны – **арифметических алгоритмов** при обучении **никто (в детстве) не видел**, и мало кто в общей массе людей может их расписать или предъявить готовые пользовательские десятичные алгоритмы: в этом отношении в массе народа **арифметических алгоритмов нет**;

- с другой стороны – все люди, обученные десятичной арифметике, выполняют арифметические операции правильно (не считая технических ошибок), а если забыли, как их выполнять, то быстро их восстанавливают:
следовательно, все-таки **алгоритмы есть** – где-то и в какой-то форме (а необученный человек не может выполнять арифметические операции в десятичной системе – у него действительно таких **алгоритмов нет**).

Возникает принципиальный вопрос – почему все это возможно?

На это современная теория алгоритмов также не дает четких, конструктивных, общеизвестных и общепонятных ответов.

Появляется задача выяснения этих проблем. Возможна следующая примерная схема анализа этой проблемной задачи:

- формирование (невидимой) **реализации алгоритма** в нейродинамических структурах исполнителя в ходе обучения и практики:
эти правил осваиваются почти до автоматизма в практике обучения и применения арифметических операций;
- наличие в десятичной арифметике видимого, удобного и компактного **протокола** результатов хода счета (запись вычислений столбиком, уголком и т.п.) – исполнение усвоенного **невидимого алгоритма**;
- **ориентация** исполнителя счета (вычислителя) **на протокол** в ходе исполнения невидимого алгоритма.

Такая (нейродинамическая) реализация алгоритмов пока недоступна теории алгоритмов. Однако уже начинаются исследования арифметических действий с применением компьютерной и магниторезонансной томографии, и более того: начинаются такие исследования так называемых быстрых счетчиков, у которых обнаруживается опора не на абстрактные численные манипуляции, а на образное мышление (они, в основном, видят эти результаты в той или иной образной форме представления, а не сознательно их вычисляют).

Школьная проблематика

Данные вопросы становятся крайне актуальными в связи с проблемами внедрения алгоритмов в обучение арифметике (которая, по крайней мере, в интернете уже практически обсуждается). Очевидно, что для младших школьных возрастов должна быть некоторая специальная алгоритмика, ориентированная: не столько на абстрактное понятийное мышление, сколько не конкретное наглядно-образное и наглядно-действенное мышление.

На указанном ранее сайте (paralg) предполагается привести две модельные программы для моделирования последовательного (и параллельного) счета в разных системах счисления с целью исследования разных аспектов арифметической алгоритмики (Рис. 2.1). В частности, выявляется (предположительно) целесообразность организации обучению счету на модели типа абак:

- эта модель с двумя многоразрядными десятичными регистрами (типа "сообщающихся" сдвоенных десятичных счет);
- при этом в принципе легко продемонстрировать параллельный поразрядный счет (с последовательными межразрядными переносами).

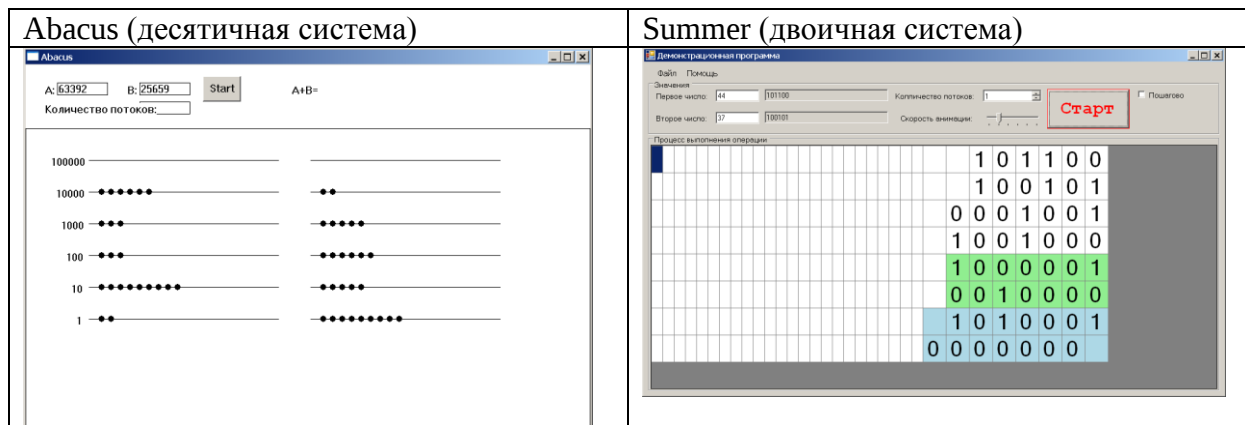


Рис. 2.1. Модельные арифметические программы

Алгоритмы параллельной арифметики

Выявляется **парадоксальная проблема** следующего типа.

С одной стороны, понятие алгоритма появилось в арифметике.

С другой стороны, в современной параллельной компьютерной арифметике в обучении цифровой схемотехнике и в практических схемотехнических разработках многочисленных видов параллельных сумматоров, умножителей и т.п. (в разных системах счисления):

- используются типовые (графические) аппаратные схемы основной вычислительной части арифметических устройств (в составе арифметико-логических устройств центральных процессоров), а также: дается общее словесное описание их работы; могут использоваться громоздкие и трудно понятные детальные временные диаграммы работы конкретной аппаратуры;
- однако **не используются** явно **параллельные арифметические алгоритмы** (которые должны были бы относиться к управляющим автоматам центральных процессоров).

Возможно, существуют подвижки в этом отношении в литературе или в проектной документации конкретных разработчиков, но они не являются общеизвестными и общепринятыми.

Появляется задача выявления причин такого положения. Теория алгоритмов также не дает ответов на такие вопросы. Предварительно можно указать следующие причинные обстоятельства:

- основная часть таких устройств (сумматоры, умножители и т.п.) реализуются как так называемые **комбинационные логические схемы** (или комбинационные автоматы);
- но в теории алгоритмов **не принято описывать алгоритмами** работу **комбинационных схем** (автоматов), и на это есть определенные причины;
- возникает задача осваивать **алгоритмическое описание комбинационных логических схем**:

с необходимым корректным концептуальным и теоретическим обеспечением (с учетом различий синхронных и асинхронных логических схем).

Пример алгоритмов параллельной арифметики

Далее (Табл. 2.1) в качестве примера приводятся структурные формулы и схемы алгоритмов параллельной арифметики в двоичной системе счисления – для параллельного сумматора с последовательными межразрядными переносами:

- алгоритм Ah параллельного полусумматора (half-adder – примерно половина полного одноразрядного сумматора) с параллельным выполнением вычислений: Zs суммы разрядов (по модулю 2) и Zc переноса в старший разряд (carry);

- алгоритм A4 четырехразрядного сумматора с параллельным выполнением алгоритмов Ah(0)..Ah(3) четырех (первых) полусумматоров в составе одноразрядных сумматоров с последующим последовательным выполнением процедуры Cr межразрядных переносов (с участием вторых полусумматоров полных одноразрядных сумматоров).

Это простая общая идея параллельного алгоритма, которая может уточняться в соответствии с конкретными схемными решениями параллельных сумматоров. Возможно, в той или иной форме такие алгоритмы используются где-то в литературе или в проектной документации. Целесообразно выносить их на анализ и обсуждение в широкой аудитории.

Указанные алгоритмы легко обобщаются на любые системы счисления, включая десятичную систему.

Табл. 2.1. Структурные формулы и схемы алгоритмов арифметики

Полусумматор	Четырехразрядный сумматор
ССА: Структурная формула алгоритма	
$Ah = (Zs \parallel Zc)$	$A4 = (Ah(3) \parallel Ah(2) \parallel Ah(1) \parallel Ah(0)) \rightarrow Cr$
ССА: Структурная схема алгоритма = БСА: Блок-схема алгоритма	
Diagram illustrating the structural formula of the half-summing unit (ССА). The unit takes two inputs and produces two outputs: Zs (sum) and Zc (carry). The sum is calculated as $s = a \oplus b$ and the carry as $c = a \& b$. The outputs are concatenated to form $Ah = (Zs \parallel Zc)$.	Diagram illustrating the structural formula of the four-bit summing unit (Четырехразрядный сумматор). The unit takes four inputs and produces four outputs: $Ah(3)$, $Ah(2)$, $Ah(1)$, and $Ah(0)$. The outputs are concatenated to form $A4$, which then feeds into a carry block Cr.

3 АЛГОРИТМИЧЕСКИЕ ПАРАДОКСЫ МАТЕМАТИЧЕСКОЙ ОБЛАСТИ

Перечисленные выше аспекты не исчерпывают парадоксальную проблематику в области арифметической алгоритмики.

В математической области последовательной и параллельной алгоритмики существует целый веер парадоксальных проблем, на порядки превышающий аналогичную арифметическую проблематику. Этот вопрос выходит за рамки возможностей изложения в данной статье. Далее затрагивает самое начало и конец этого проблемного комплекса.

Проблема общего содержания теории алгоритмов

Теория алгоритмов возникла в математике в 30-е годы 20-го века и далее активно и, более того, бурно развивалась по разным направлениям, что является источником множества разных парадоксальных проблем. В частности, существует следующий исходный общий парадокс.

Проблема связности и целостности теории алгоритмов:

1) В настоящее время теория алгоритмов не образует единую связную научную дисциплину и представлена множеством частных теорий и задач сходного и различного содержания и оформления:

практически отсутствует их общая системная интеграция.

2) Это обстоятельство является причиной распространенной массовой дезориентации по содержанию, целям и задачам теории алгоритмов не только неспециалистов в области теории, но многих представителей отдельных теоретических направлений.

Представляет существенный (во всех отношениях) интерес выявление и выяснение конкретных причин такого состояния вопроса.

В целом выделяются следующие два крупных направления современной условно существующей общей теории алгоритмов (Рис. 3.1):

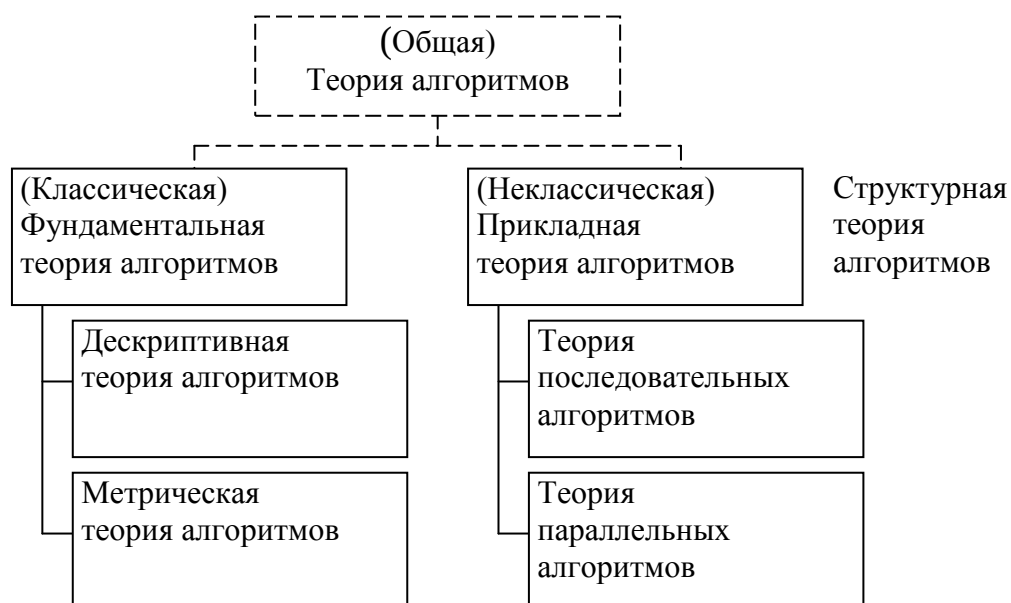


Рис. 3.1. Общий состав теории алгоритмов

1) **Классическая** или **фундаментальная теория алгоритмов**: исследования фундаментальных математических сущностей, связанных с алгоритмической проблематикой в области оснований математики, основной задачей которой является строгое логическое обоснование математики в целом и всех ее дисциплин.

2) **Неклассическая** или **прикладная теория алгоритмов**:

непосредственные исследования алгоритмов – первоначально в математических приложениях, а затем в технике и других областях.

Сопутствующую парадоксальную проблему составляет тот факт, что *такие наименования* этих направлений *ничего не говорят по их существу*, что является дополнительным источником неясностей и путаницы.

Фундаментальная (классическая) теория алгоритмов

Наиболее систематическое изложение общего содержания классической теории алгоритмов впервые было выполнено, по-видимому, в выдающейся обзорно-аналитической и итоговой в свое время работе [5] мирового уровня. С тех пор намеченная общая картина в целом сохраняется.

Согласно работе [5] фундаментальная теория алгоритмов включает в себя две составляющие научные дисциплины (Рис. 3.1):

1) *Дескриптивная* или *качественная теория алгоритмов*:

- это исторически первичная теория алгоритмов – возникла в 30-е годы 20-го века в области алгоритмических проблем оснований математики: основное ее содержание составляет *проблема (алгоритмической) разрешимости* – разработка общих методов и средств доказательства наличия или отсутствия общего алгоритма (метода, способа) решения заданного (бесконечного) класса математических задач (до попыток поиска, построения и предъявления такого алгоритма, которого может и не быть);
- первоначально она представляла собой всю наличную по тем временам теорию алгоритмов и именовалась общим термином "теория алгоритмов".

2) *Метрическая* или *количественная теория алгоритмов*:

- это дополнение первичной теории алгоритмов: *теория оценок* (скорости роста) *сложности* исполнения алгоритмов по длине (емкости) записи или длительности (числу шагов) их выполнения в зависимости от (скорости роста) объема исходных данных [6];
- она возникла после появления ЭВМ – с возникновением проблем недостатка памяти и большого (долгого) времени счета на ЭВМ;
- с ее появлением понятие теории алгоритмов расширилось, и по традиции до сих пор термином "теория алгоритмов" очень часто именуется совокупность этих двух теорий, что в настоящее время уже не соответствует реальности и вносит большую путаницу в общие представления.

Прикладная (неклассическая) теория алгоритмов

Прикладная теория алгоритмов представляется собой множество разных связанных и несвязанных между собой частных теорий, методов и задач. Они обобщенно группируются в два исторически последовательно связанных направления (Рис. 3.1):

1) Прикладная *теория последовательных алгоритмов*:

- она возникла в связи с появлением ЭВМ и программирования вычислений, а позднее (в более общем плане) обработки данных;
- основным объектом ее исследований является *строение или структура* (последовательных) алгоритмов, реализующих их (последовательных) дискретных процессов и выполняющих их объектов (объектных систем);
- основной ее целью является разработка, формализация и автоматизация задач *структурного описания, анализа и синтеза, структурных преобразований* алгоритмов и систем их реализации;

• по своему существу – *это структурная теория (последовательных) алгоритмов*: и это есть *второе название* теории, *отражающее ее суть*.

2) Прикладная *теория параллельных алгоритмов*:

она возникла на базе теории последовательных алгоритмов в связи с появлением параллельного программирования параллельных вычислений и (в более общем плане) параллельной обработки данных;

для нее характерны аналогичные сущностные структурные аспекты и по своему существу – это **структурная теория параллельных алгоритмов**.

В настоящее время существует тенденция интеграции этих двух направлений, причем более общая теория параллельных алгоритмов включает в себя теорию последовательных алгоритмов как вырожденный, но базисный частный случай параллелизма (параллелизм отсутствует).

Неизвестная прикладная (структурная) теория алгоритмов

Выявляется парадоксальная ситуация следующего типа:

- с одной стороны, существует обширная по составу частных направлений прикладная (структурная) теория алгоритмов;
- с другой стороны, она не представлена адекватно своим именем в научно-технической и учебно-методической литературе в массовом масштабе.

Этот факт легко проверить поисковиком в интернете:

- на термин "прикладная теория алгоритмов" выдается список всего в 3-4 десятка источников с повторами;
- на термин "структурная теория алгоритмов" вообще выдается список источников менее десятка с повторами.

Это имеет место, несмотря на то, что, например, термин "Прикладная теория алгоритмов" был известен еще в 50-е годы 20-го века, и он использовался в прямое противопоставление классической теории алгоритмов. В частности, он был представлен статьей "Прикладная теория алгоритмов" в "Словаре по кибернетике" [7] еще 1979 и затем 1989 года издания.

Представляет интерес выявление причин такого положения дел. Предположительно это связано с тем обстоятельством, что все источники рассматривают частные вопросы прикладной теории алгоритмов, и пока нет интегрированного изложения прикладной теории алгоритмов в целом и с таким явным ее названием (на обложке).

Это обстоятельство еще более запутывает суть вопроса. Дело доходит до того, что инженерным, экономическим и другим нематематическим специальностям, менеджерам и т.п. под названием "теория алгоритмов" вместо прикладной теории преподается классическая теория алгоритмов.

Предполагается организовать специальное обсуждение этого проблемного вопроса на указанном сайте (paralg). А в отношении **начальной массовой робототехнической грамотности** [1] принимается установка:

необходимо обеспечить жесткие **заградительные меры** против попыток навязать здесь классическую теорию алгоритмов – с проблемами алгоритмической разрешимости и сложности вычислений, включая представительные вычислительные модели – машины Поста и Тьюринга, рекурсивные функции, нормальные алгорифмы Маркова и т.п.

Антиалгоритмические настроения

В заключение этого раздела приводятся следующие данные.

С появлением и распространением декларативного (логического и функционального) программирования в его рамках возникло "разговорное" в основном течение, которое четко представляется следующей фразой:

"Растут и ширятся антиалгоритмические настроения".

Суть его заключается в далеко идущих выводах и заявлениях относительно отсутствия необходимости в использовании здесь алгоритмов и теории алгоритмов, которые катастрофически ограничены и исторически обречены (и вообще никчемные).

Правда, такие громогласные декларации были характерны, по-видимому, до 90-х годов. А в настоящее время это представлено на уровне "бытовых" антиалгоритмических настроений и простительно студентам в их рефератах, куда они радостно копируют подобную крамолу, не ведая, что творят.

В этой связи необходимо отметить следующий факт:

- теория алгоритмов уже не один десяток лет пребывает в роли побитой собаки и делает вид, что ничего предосудительного не происходит;
- перефразируя известную революционную фразу можно сказать: ничего не стоит такая теория, которая не может себя защитить.

Теории алгоритмов надо занять в этом отношении агрессивную позицию, уважительно, но твердо выявляя степень адекватности таких антиалгоритмических заблуждений. Суть дела в том, что отрицая классические алгоритмы, противники алгоритмов не слезают с алгоритмов других видов.

В настоящее время алгоритмическая практика и сама теория алгоритмов расширяют алгоритмические представления и атакуют классические алгоритмические представления, включая святая святых – алгоритмы как строгие и точные предписания, дискретность и определенность алгоритмов и т.п.

Например:

- существуют теоретические основы нечетких алгоритмов: в частности, известный алгоритм (правило) перехода улицы, который часто приводится в школьной информатике – это, строго говоря, разновидность нечетких алгоритмов (которой все успешно пользуются);
- актуальным объектом становятся непрерывные алгоритмы и т.п.

4 АЛГОРИТМИЧЕСКИЕ ПАРАДОКСЫ В ТЕХНИЧЕСКОЙ ОБЛАСТИ

В этой области еще больше неясности и критических проблем парадоксального типа. Этот вопрос выходит за рамки данной статьи. Но в целом необходимо выделить следующий ключевой проблемный факт:

1) Применительно к техническим алгоритмам выглядят архаизмом следующие категорические общеутвердительные высказывания:

- **понятие алгоритма** – это **математическое понятие**;
- **теория алгоритмов** – это **математическая теория**.

2) Этот справедливо для математических алгоритмов и теории математических алгоритмов. Но такие всеобщие представления связывают руки в области формирования полноценной собственной концепции и теории технических алгоритмов:

предполагаемая **теория технических алгоритмов** – это высоко **математизированная**, но не математическая, а **техническая теория** управления техническими дискретными процессами и выполняющими их техническими объектами (также как, например, теория механизмов и машин или теоретические основы электротехники – это высоко математизированные вполне технические теории).

Представляет интерес обзорный анализ теории алгоритмов в такой ее трактовке.

ЗАКЛЮЧЕНИЕ

В работе представлена общая (системная) ориентировка в иерархии исторических областей и связанных с ними общих направлений массовой алгоритмизации разнообразной деятельности человека.

Обнаруживается много, условно говоря, "позитивного негатива", позитивного в том плане, что его выявление и формулировка подсказывает пути и способы его преодоления. При этом данные аспекты увязываются с проблемами параллельной (и последовательной) алгоритмики в массовой информатике и робототехнике [1].

Появляется задача разделения: профессиональной алгоритмики (с одной стороны) и ее адаптации (с другой стороны) для разных уровней и подуровней ее пользователей в области массовой информатики и робототехники (без перегрузки теорией на начальных этапах обучения).

ЛИТЕРАТУРА

Статья [1] автора доступна на сайте:
paralg.ucoz.com: Параллельные алгоритмы и логика.

1. Житников А.П. Параллельная алгоритмика в массовой информатике и робототехнике. – В данном сборнике, 2012. 15 с.
2. Ал-Хорезми. Книга об индийском счете. / В сб.: Хорезми. Математические трактаты. – Ташкент: Фан, 1983. С. 5 – 19.
3. Юшкевич А. П. О труде по арифметике Мухаммеда ибн Муса ал-Хорезми. / В сб.: Хорезми. Математические трактаты. – Ташкент: Фан, 1983. – С. 142 – 149.
4. Матвиевская Г. П. Выдающийся математик Мухаммед ибн Муса ал Хорезми. / В сб.: Хорезми. Математические трактаты. – Ташкент: Фан, 1983. С. 266 – 399.
5. Успенский В. А., Семенов А. А. Теория алгоритмов: основные открытия и приложения. – М.: Наука, 1987. – 288 с.
6. Орехов Э.Ю., Орехов Ю.В. Введение в теорию сложности решения задач. – Уфа: УГАТУ, 2008. – 87 с.
7. Словарь по кибернетике. – К.: Гл. ред. УСЭ, 1979, 1989.