

Статья. Библиографические данные:

Житников А.П.
Теория алгоритмов в практике применения и в учебном процессе.
// Применение инновационных технологий в преподавании математических дисциплин в школе и в вузе.
Сб. статей. всероссийской. науч.-практ. конференции.
– Борисоглебск: БГПИ, 2013. – С. 38-56.

ТЕОРИЯ АЛГОРИТМОВ В ПРАКТИКЕ ПРИМЕНЕНИЯ И В УЧЕБНОМ ПРОЦЕССЕ

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

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ИСХОДНЫЕ УСТАНОВОЧНЫЕ ПОЛОЖЕНИЯ	4
Алгоритмические системы: алгоритмы и системы их реализации	4
Состав комплексных алгоритмических систем	4
Обобщенные автоматизированные алгоритмические системы	5
2 ИСТОРИЧЕСКИЕ ОБЛАСТИ, НАПРАВЛЕНИЯ И СТАДИИ АЛГОРИТМИЗАЦИИ.....	6
Исторические области массовой алгоритмизации	6
Исторические направления и стадии массовой алгоритмизации	7
3 БАЗИСЫ И НАДСТРОЙКИ ОБЛАСТЕЙ АЛГОРИТМИЗАЦИИ.....	9
Проблемы соотношения алгоритмических базисов и надстроек	9
Базис и надстройка математической области алгоритмизации	9
Базис и надстройка технической области алгоритмизации	9
Базис и надстройка алгоритмизации области кибернетики	10
Следствия взаимного влияния базиса и надстройки.....	10
4 ПЕРЕКРЕСТНЫЕ ВИДЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ	12
Взаимное согласование перекрестной подготовки	12
Математическая подготовка техников	12
Техническая (инженерная) подготовка математиков	12
Кадровый состав преподавателей математики для техников	13
5 АЛГОРИТМИЧЕСКАЯ ПОДГОТОВКА И ЕЕ МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	14
Ключевые аспекты алгоритмической подготовки техников.....	14
Направления теории алгоритмов	14

	Математическая теория алгоритмов.....	14
	Техническая теория алгоритмов	16
	Теория алгоритмов кибернетики	16
	Общая теория алгоритмов	17
	Общая оценка состояния теории алгоритмов	17
	Выбор направлений теории алгоритмов в ее преподавании.....	17
6	НЕКОТОРЫЕ КРИТИЧЕСКИЕ (ПАРАДОКСАЛЬНЫЕ) АСПЕКТЫ.....	19
	Разочарование программистов в теории алгоритмов	19
	Сопутствующие замечания	19
	Уточнение "проблемы разочарования программистов"	20
	ЗАКЛЮЧЕНИЕ	21
	ЛИТЕРАТУРА.....	21

ВВЕДЕНИЕ

В статьях [1,2] выявляются *критические проблемы* формирования, развития, практического применения *теории алгоритмов* и ее внедрения в учебный процесс разных уровней и профилей обучения. Отражаются проблемы теории алгоритмов в целом и, в особенности, *прикладной (структурной) теории параллельных алгоритмов* и ее *логико-математического обеспечения* в перспективных приложениях в области массовой *информатики* и активно формируемой массовой *робототехники* (в рамках общей исходной и концепции конечной материализации информации).

Эта проблематика длительно накапливается и усугубляется в условиях высоких и нарастающих темпов стихийного научно-технического прогресса. Она в принципе поддается упорядочению и разрешению, но необходимы комплексные системные меры. Для обеспечения этого необходим системный анализ развития *алгоритмических (логико-алгоритмических) систем* информатики на всем интервале исторического процесса формирования и развития массовой алгоритмизации человеческой деятельности, включая выявление и анализ исторических причинных корней порождения и накопления указанной неразрешенной проблематики.

В данной статье представлена подготовка изложения такого анализа.

1 ИСХОДНЫЕ УСТАНОВОЧНЫЕ ПОЛОЖЕНИЯ

Алгоритмические системы: алгоритмы и системы их реализации

Принимается рабочее уточнение понятия **алгоритмических (логоико-алгоритмических) систем** информатики (Рис. 1.1):

это **комплексные системы отображения и реализации** алгоритмов разных структурных классов (включая их исходное структурное деление на последовательные и параллельные алгоритмы).



Рис. 1.1. Алгоритмические системы

Состав комплексных алгоритмических систем

Комплексные алгоритмические системы включают в себя два исходных взаимосвязанных вида составляющих их алгоритмических систем:

1) **Алгоритмы** (и логика в их составе) – предписания некоторым (единичным или групповым) исполнителям выполнить комплекс действий определенной последовательной или параллельной структуры во времени (с целью достижения определенных результатов):

- это **знаковые системы** определенной структуры, представляющие собой **тексты** директивного содержания (подлежащие выполнению), используемые в разных системах алгоритмических **языков** и их **грамматик**;
- при этом тексты (от латинского слова *textus* – связь, соединение, сплетение, ткань) интерпретируются в предельно широком смысле с позиций семиотики (теории знаков и знаковых систем):

тексты – это **системы знаков** любой природы, любого состава, объема и любой структуры, **связанные общим смыслом** [3,4];

- в частности, конкретно имеются в виду в первую очередь разные визуальные формы представления алгоритмов – **литерные** (буквенно-цифровые и т.п.) и **графические знаковые формы**:

структурные и функциональные **формулы** и **схемы** алгоритмов, **псевдокоды** алгоритмов, ориентированные на синтаксис и лексику исходных кодов программ в разных языках программирования и т.п.;

- в целом рассматриваются **полиморфные представления** текстов параллельных (и последовательных) алгоритмов:

представления алгоритмических (логоико-алгоритмических) текстов, языков и грамматик в разных структурно согласованных и взаимно обратимых знаковых формах (разного назначения).

2) **Системы реализации** (осуществления, воплощения) алгоритмов, то есть системы их **выполнения и обеспечения их выполнения** (так или иначе представленные в алгоритмах), включая:

а) Дискретные **процессы** определенной последовательной или параллельной структуры во времени (с определенными результатами по технологии их выполнения): это **процессные системы** реализации алгоритмов, которые непосредственно предопределяются и, следовательно, отображаются алгоритмами, а также протоколами их исполнения – литературными и табличными записями, масштабированными временными диаграммами разных типов, графиками и т.п.

б) **Объекты** определенной структуры, выполняющие процессы:

- это **объектные системы** реализации алгоритмов – исполнители алгоритмов (единичные или множественные), которые предопределяются или предполагаются в текстах алгоритмов и связанных с ними контекстах;
- они представляются структурными, принципиальными и компоновочными схемами, статическими и динамическими мнемосхемами и т.п.

Выделяются следующие основные формы реализации алгоритмов:

программная, аппаратная, организационная (персональная) и **комбинированная** реализация алгоритмов.

Обобщенные автоматизированные алгоритмические системы

В общем случае рассматриваются автоматизированные (человеко-машинные) системы реализации алгоритмов всех видов:

1) Во всем диапазоне по **степени автоматизации**:

- начиная от нуля автоматизации (автоматизация отсутствует) – системы ручного и умственного принципа действия;
- до автоматических систем, действующих автономно относительно человека в определенном комплексе функций.

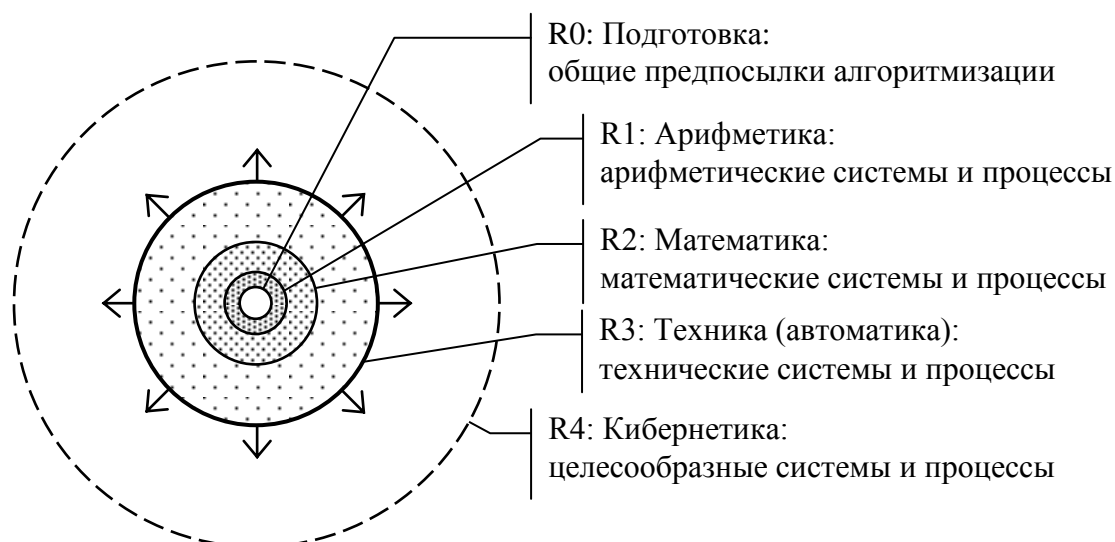
2) Во всем диапазоне по уровню **технического оснащения**:

- начиная от нуля – техническое оснащение отсутствует;
- до развитых комплексов средств технического оснащения, систем механизации и автоматизации деятельности любых видов и масштабов.

2 ИСТОРИЧЕСКИЕ ОБЛАСТИ, НАПРАВЛЕНИЯ И СТАДИИ АЛГОРИТМИЗАЦИИ

Исторические области массовой алгоритмизации

Анализ проблемы [2] выявляет **общий процесс развития** массовой алгоритмизации **в последовательном расширении** общей области массовых приложений алгоритмов – множества используемых алгоритмических систем информатики, их концептуального и теоретического обеспечения. Такое последовательное расширение происходит в порядке формирования следующих последовательно возникающих прикладных **областей** (regions) **массовой алгоритмизации** (Рис. 2.1):



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

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

Рис. 2.1. Исторические области массовой алгоритмизации

R0: **Подготовка**: Общая подготовка массовой алгоритмизации: она содержит **предпосылки** и алгоритмические **прототипы** всех областей и направлений формирования алгоритмики, поскольку массовая алгоритмизация определенного типа не начинается внезапно, а всегда имеет каких-то более ранних **исторических предшественников**, причем (не случайного) 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 оснащения и автоматизации:

- от систем последовательного устного и пальцевого счета;
- до параллельной компьютерной (микропроцессорной) арифметики.

R2: **Математика** (включая арифметику): Алгоритмизация математики. Это **исторически вторичная** качественно новая, очень широкая и разнообразная область массовой алгоритмизации математической деятельности человека. Она включает в себя арифметику как особый **базисный** частный случай математики, но **на порядки превосходит** ее по масштабам и разнообразию в **настройке** базиса. Она охватывает математические алгоритмические системы всех уровней оснащения и автоматизации:

- от ручных математических систем последовательного действия – с внешними носителями на глиняных табличках, папирусе или бумаге;
- до параллельной компьютерной математики разных локальных и глобальных сетевых масштабов – с машинными носителями информации.

В настоящее время это наиболее *развитая* и *господствующая* область приложений алгоритмов. При этом аспект ее "господства" имеет и *позитивные и негативные стороны* в современных (технических) условиях.

R3: *Техника* (и автоматика, включая технику математики и арифметики): Алгоритмизация техники. Техника любых (всех) видов деятельности интерпретируется в предельно широком смысле как профессиональное *искусство* высокоорганизованного выполнения деятельности. Она включает *методы и средства* (сумму технологий) ее выполнения и, в частности, *комплексы 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 в *надстройке* базиса. Она охватывает технические алгоритмические системы всех уровней автоматизации и технического оснащения и любых масштабов.

Это *активно формируемая область* алгоритмизации. Однако:

- она еще не представляет собой четко сформировавшееся направление с собственной, хорошо различимой теоретической базой;
 - в значительной мере она пока подчиненна господствующим математическим алгоритмическим представлениям, концепциям и подходам:
- такое подчиненное положение тормозит формирование ее собственной концептуальной и теоретической системы, но это не вина математики, а собственные "детские болезни" становления технической алгоритмики.

R4: *Кибернетика*: Алгоритмика кибернетики целесообразных систем (включая технические системы как особый вид целесообразных систем) – перспективная предельно широкая общая область алгоритмизации ключевого значения, причем:

- она находится в опытно-поисковой стадии первичного накопления алгоритмических приложений по многим разным научным фронтам;
- в целом идет ее интенсивная подготовка, но пока трудно говорить о самостоятельном таком виде массовой алгоритмизации.

Исторические направления и стадии массовой алгоритмизации

По ходу развития цивилизации формируется система областей (R_i), направлений (D_i) и стадий (S_i) общего исторического процесса массовой алгоритмизации (Рис. 2.2: временные границы стадий ориентировочные).

По мере интенсивного формирования областей (R_i) массовой алгоритмизации закладываются соответствующие им направления (D_i) массовой алгоритмизации, которые далее сохраняются и развиваются. При этом появление и развитие каждого нового направления массовой алгоритмизации происходит в условиях существования и развития ранее возникших областей и направлений. Последовательно накапливается система взаимосвязанных и взаимодействующих направлений алгоритмизации.

Появление новых направлений (D_i) представляет собой начало соответствующих им стадий (S_i), которые существуют в периоды времени:

- от начала интенсивного формирования соответствующих им областей массовой алгоритмизации (R_i);
- до начала интенсивного формирования очередных областей общего процесса массовой алгоритмизации ($R_{(i+1)}$).

При этом в конце интервала предшествующих стадий появляются этапы интенсивной подготовки в области предпосылок последующего (интенсивного) формирования очередных областей и стадий (на диаграмме Рис. 2.2 они отмечены пунктиром).

Общий процесс массовой алгоритмизации представляет собой последовательность таких стадий: $S1 \rightarrow S2 \rightarrow S3 \rightarrow S4$.

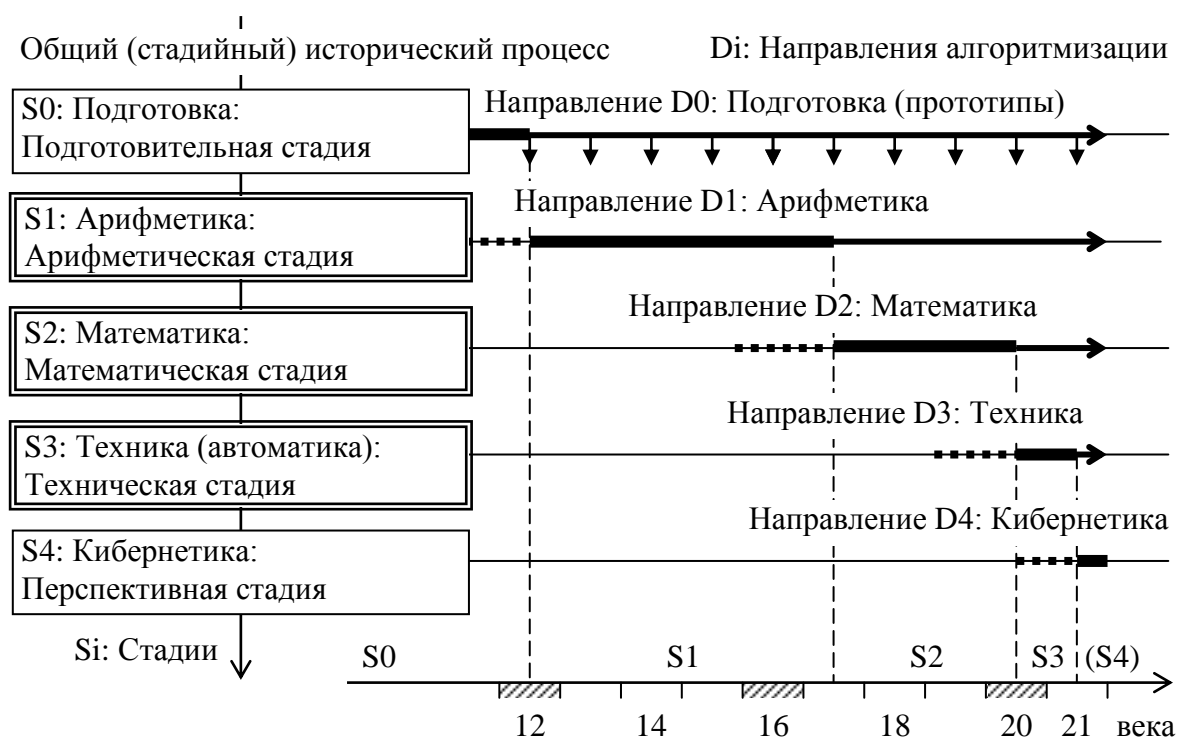


Рис. 2.2. Направления и стадии общего процесса массовой алгоритмизации

3 БАЗИСЫ И НАДСТРОЙКИ ОБЛАСТЕЙ АЛГОРИТМИЗАЦИИ

Проблемы соотношения алгоритмических базисов и надстроек

В статье [2] поставлена задача систематического анализа соотношения **базиса** и **надстройки** каждой области массовой алгоритмизации в разных отношениях, выявляются примеры некоторых проблемных аспектов и намечаются пути их возможного решения.

В данной статье выделяется следующий ключевой аспект: **взаимное влияние** специфики **базиса** и **надстройки** в каждой такой области массовой алгоритмизации – встречающиеся тенденции взаимного влияния.

Базис и надстройка математической области алгоритмизации

В этой области массовой алгоритмизации выявляются следующие встречающиеся тенденции (Рис. 3.1):

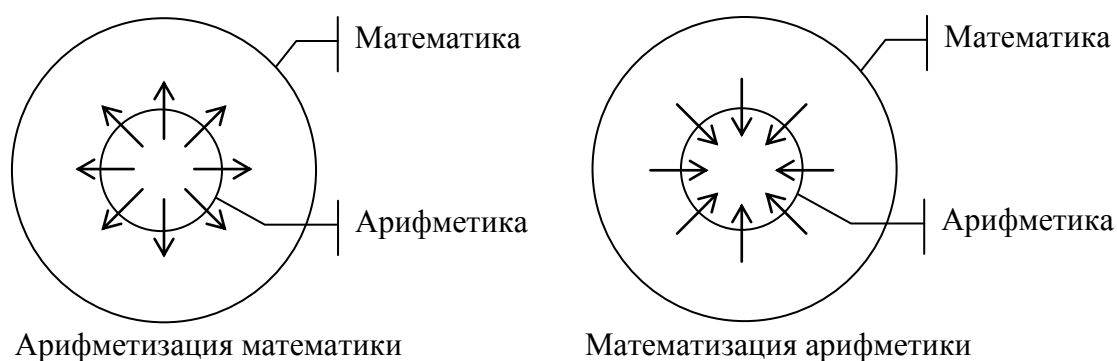


Рис. 3.1. Взаимодействие арифметики (базиса) и математики (надстройки)

1) Прямая тенденция – **арифметизация математики**:

- в основаниях математики все математические дисциплины переводятся на арифметическую аксиоматическую основу с целью их строгого логического обоснования на относительно простой аксиоматике разных числовых множеств и отношений;
- это оказалось недостаточно, и дополнительно потребовался их перевод на теоретико-множественную аксиоматическую базу числовых множеств.

2) Обратная тенденция – **математизация арифметики**:

- теория последовательных и параллельных арифметических операций и устройств в разных системах счисления формируется на вполне математической основе теории булевых функций (двоичных, троичных, многозначных и т.п.) – с выходом на теорию временных булевых функций;
- теория чисел – это вполне математическая дисциплина, использующая математические методы для изучения разных числовых проблем и т.п.

Базис и надстройка технической области алгоритмизации

Здесь также выявляются встречающиеся тенденции (Рис. 3.2):

1) Прямая тенденция – **математизация (и арифметизация) техники**. Это хорошо известная общая тенденция. Она включает автоматизацию технических систем на основе параллельной (и последовательной) арифметики центральных процессоров и параллельной (и последовательной) компьютерной (и в целом, микропроцессорной) математики:

- представляет интерес систематический классификационный анализ ее конкретного содержания в области технических приложений параллельных (и последовательных) алгоритмов;

- особый интерес представляет следующий факт – существование технических приложений алгоритмов **без (явного) использования математики**: это (внешне) противоречит представлениям относительно математического базиса технической алгоритмики и требует корректного анализа (что здесь имеет место на самом деле, и что из этого следует).

2) Обратная тенденция – **инженеризация (технизация) математики (и арифметики)**. В частности:

- параллельная компьютерная арифметика в цифровой схемотехнике: в настоящее время это уже практически вполне инженерная дисциплина и вполне техническая (высоко) математизированная теория;
- параллельная компьютерная математика – очевидны тенденции инженеризации современных машинных методов параллельных (и последовательных) вычислений, параллельного (и последовательного) программирования вычислений на проектной основе и т.д.:

представляет интерес специальный анализ данного аспекта (что это такое по своему существу, какой реальный текущий уровень показателей качества инженеризации математики, и что из этого следует).



Рис. 3.2. Взаимодействие математики (базиса) и техники (надстройки)

Базис и надстройка алгоритмизации области кибернетики

Здесь также выявляются встречные тенденции (схемы получаются аналогично Рис. 3.1, Рис. 3.2):

1) Прямая тенденция – **инженеризация (технизация) кибернетики**: применение технических методов и средств в исследованиях, описании, анализе кибернетических систем – в биологии, психологии, генетике, культуре, социологии и т.п.

2) Обратная тенденция – **кибернетизация техники (а также математики и арифметики)**. Также представляет интерес анализ этих вопросов (что это такое по своему существу, и что из этого следует).

Следствия взаимного влияния базиса и надстройки

Указанное типовое соотношение базиса и надстройки разных областей алгоритмизации требует систематического анализа. При этом относительно анализа технической области алгоритмизации:

1) Принимается общая исходная установка:

- в математике и в современной технике существуют свои **критерии** и **нормы**, своя **профессиональная культура** и свой **менталитет**: они не хуже и лучше друг друга – они разные по своей природе (по назначению и т.п.);
- в указанных встречных тенденциях их не следует подменять, навязывать или подавлять, но необходимо их **сочетать** и **согласовывать**.

2) Выделяются следующие первичные аспекты анализа:

- перекрестные виды профессиональной подготовки специалистов;
- алгоритмическая подготовка и ее математическое обеспечение.

4 ПЕРЕКРЕСТНЫЕ ВИДЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ

Взаимное согласование перекрестной подготовки

Далее (Рис. 4.1) в связи с общими встречными математическими и техническими тенденциями выделяется актуальный вопрос взаимного согласования **перекрестной подготовки** для соответствующих категорий специалистов с целью обеспечения их **взаимопонимания и взаимодействия**.

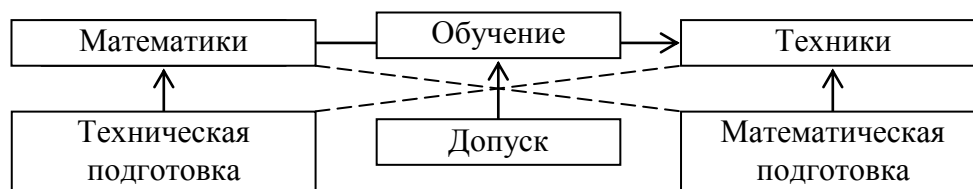


Рис. 4.1. Перекрестные виды подготовки техников и математиков

Математическая подготовка техников

В наше время практически все **технические специальности** всех уровней обучения имеют **математическую подготовку** (достаточную или недостаточную, хорошую или плохую). При этом существуют примеры более или менее удовлетворительного решения этой задачи (представляющие интерес для конструктивного анализа). Однако далее приводятся формулировки в специально обостренной форме (для убедительности):

- в целом, как правило, **техники** ("технари": инженерно-технические специальности) обучаются математике как "**плохие математики**";
- при этом отношение к ним со стороны обучающих математиков (в среднем) в лучшем случае сочувственное, снисходительное, покровительственное, неопределенное, а в худшем – с установками следующего типа:
не научить, а доказать, что он (технарь) "лодырь и бездарь", не выучить, а выгнать с экзамена и, как следствие, из учебного заведения – с массовым отсеком по причине "математической непригодности" (были в свое время такие факты в одном служебном разбирательстве с оргвыводами);
- низкая (в среднем) результативность математического обучения очень часто психологически компенсируется ссылками типа "не хотят учиться" (хотя, обычно, они "хотят учиться", но что-то не так в самой системе).

Техников ("технарей") необходимо готовить как хороших пользователей математики. Но тогда возникают проблемные вопросы:

- что такое "**техник – хороший пользователь математики**"?
- какие могут быть четкие **критерии** соответствия (необходимые и достаточные), и как их **обеспечить**?

Техническая (инженерная) подготовка математиков

Учитывая обратную тенденцию инженеризации математики, возникает задача обеспечения определенной (необходимой и достаточной) **инженерной (технической) подготовки математиков**. Например:

- даже "чистая математика" в современной форме существования – это не просто некая туманная сущность, в бумагах обитающая, а система индустриальных (машинных) информационных изделий, подготовленных на индустриальной (машинной) основе на машинных носителях;
- для разработки и решения современных математических задач и проблем (с применением и, возможно, с разработкой программного обеспечения) необходима **компью-**

терная подготовка математиков – это, по своей сути, некоторая **техническая подготовка** (достаточная или недостаточная, хорошая или плохая, инженерная или непонятно какая).

Компьютерная техническая подготовка математиков сейчас, так или иначе, выполняется, но необходим анализ этого вопроса в более общем техническом плане, в частности, в связи со следующим далее вопросом.

Кадровый состав преподавателей математики для техников

Для обучения техников в принципе недопустимо неконтролируемое массовое привлечение **"чистых" и "абстрактных" математиков**, для этого необходимы специализированные **прикладные математики**:

- с определенной общей и целевой **инженерной подготовкой**;
- с наличием (неформального) **допуска** к преподаванию математики конкретным техническим специальностям.

В настоящее время это задача, трудновыполнимая в массовом порядке, но объективно она существует и подлежит неформальному решению.

Формирование кадрового состава преподавателей математики для техников по многим причинам целесообразно на комбинированной основе:

- это могут быть специалисты с базовым математическим или техническим образованием, но с необходимой дополнительной подготовкой, с двумя базовыми образованиями, (реальные) инженеры-математики (а не просто назначенные на должность инженера-математика) и т.п.;
- представляет интерес совместная работа специалистов двух таких категорий в научных разработках, в подготовке учебных пособий и т.п.

5 АЛГОРИТМИЧЕСКАЯ ПОДГОТОВКА И ЕЕ МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Ключевые аспекты алгоритмической подготовки техников

Далее кратко излагается проблематика алгоритмической подготовки техников ("технарей") разных уровней и профилей обучения.

Что касается алгоритмической подготовки математиков, автор данной статьи не берется высказывать какие-то требования за исключением одного вопроса:

математики, привлекаемые к обучению техников (технарей) в связи с математическими проблемами теории алгоритмов, должны соответствовать общим требованиям указанного выше типа.

В алгоритмическую подготовку техников разных уровней и профилей обучения включатся, в частности, следующие вопросы:

- необходимые направления теории алгоритмов;
- необходимое математическое обеспечение теории алгоритмов.

Направления теории алгоритмов

Согласно прикладному составу областей и направлений массовой алгоритмизации (Рис. 2.1) логично выделить соответствующие направления теории алгоритмов (Рис. 5.1), хотя пока это достаточно проблематично:

- **арифметическая** теория (арифметических) алгоритмов;
- **математическая** теория (математических) алгоритмов;
- **техническая** теория (технических) алгоритмов;
- теория алгоритмов **кибернетики** (кибернетическая теория алгоритмов).

Арифметическая теория алгоритмов

Здесь существуют нерешенные проблемы [1,2], во-первых, ее формирования как самостоятельного направления теоретической алгоритмики и, во вторых, в полном диапазоне ее учебных приложений:

- от алгоритмического обучения арифметике в младшей школе;
- до алгоритмических основ параллельной компьютерной арифметики в средней и высшей профессиональной школе.

Тем более необходимы системный анализ этой проблемы и разработка адекватных системных мер.

Математическая теория алгоритмов

Это наиболее продвинутая, но проблемная область, в частности:

1) Существуют традиционные установочные **представления**:

- **алгоритм** – это математическое понятие, **теория алгоритмов** – это математическая теория:

все это вступает в противоречие с технической теорией алгоритмов;

- необходимы более точные формулировки:

математический алгоритм – это математическое понятие, **теория математических алгоритмов** – это математическая теория (все это становится тривиально справедливым почти на уровне тавтологии).

2) Анализ вопроса [2] выявляет следующий общий состав математической теории алгоритмов (Рис. 5.1) – с проблемами корректного понимания и объяснения наименований составляющих направлений теории [2]:

а) **Классическая** или **фундаментальная теория (математических) алгоритмов**, включая два основных составляющих направления:

- **дескриптивная** или **качественная** теория алгоритмов – возникла в приложениях к основаниям математики в области проблем алгоритмической разрешимости (то есть существования или отсутствия общих алгоритмов решения для заданных классов математических задач);
- **метрическая** или **количественная** теория алгоритмов или **теория сложности** выполнения алгоритмов (классификационные асимптотические оценки вычислений по длине их записи и числу операций).

б) **Неклассическая** или **прикладная теория алгоритмов** или **структурная теория алгоритмов** [2], изучающая строение (структуру) алгоритмов и систем их реализации, их структурные и другие характеристики и разрабатывающая методы структурного описания, анализа, синтеза алгоритмов, их структурных преобразований и т.п.:

- в настоящее время это малоизвестное под такими общими названиями множество ("россыпь") частных направлений, подходов и школ, связанных с разработкой алгоритмов;
- первоначально все это относилось к математическим алгоритмам, затем на этой основе появились приложения теории в технике и кибернетике.



Рис. 5.1. Общие направления теории алгоритмов

3) Сложилось некоторое "недопонимание" в отечественных, по крайней мере, массовых представлениях относительно теории алгоритмов:

- теория алгоритмов **по устаревшей традиции** отождествляется, как правило, с исторически первичной классической теорией алгоритмов;
- это легко проверить, например: поисковиками в интернете (по ключевым словам "теория алгоритмов") и, в частности, в русской версии Википедии; по первым отечественным образовательным стандартам (прошлого десятилетия) – по содержанию дисциплин типа "дискретная математика", "математическая логика и тео-

рия алгоритмов", "теория алгоритмов" и во множестве изданных соответственно им учебниках и учебных пособиях;

- для сравнения, в английской версии Википедии в теорию алгоритмов (кроме классической алгоритмической проблематики) включается **теория автоматов** – конечные, стековые и другие автоматы:

это выходит **за рамки классической теории** алгоритмов в область **прикладной теории** алгоритмов и отражается в некоторых отечественных стандартах и учебниках, но это не вся прикладная теория алгоритмов, и она вообще при всем этом **обычно не упоминается**;

- фактически пока очень неясно фигурирует безымянная "полуприкладная" или даже "на четверть прикладная" (и менее) теория алгоритмов.

4) Обучение в области неклассической прикладной теории (математических) алгоритмов, обычно:

- "распыляется" по разным другим учебным дисциплинам;
- представлено в целом в неполном "диффузном" составе и, главное, без целостных прикладных алгоритмических представлений.

Техническая теория алгоритмов

Это еще менее известное под таким наименованием множество частных направлений, подходов и школ, связанных с разработкой алгоритмов в технических алгоритмических системах информационных и материальных технологий (например, операционная система ЭВМ – это техническая, а не математическая программная алгоритмическая система управления):

- здесь ситуация еще более проблемная и неопределенная (чем в математической теории алгоритмов) во всех отношениях;
- более того, в конечном счете, в обучении техников (инженерно-технических специальностей) техническая теория алгоритмов пока **подменяется** математической теорией алгоритмов и даже хуже того – классической теорией алгоритмов (практически без понятий о прикладной теории);

- причем здесь обычно проявляется **"математический диктат"** в отношении того, **что и как преподавать** техникам в области теории алгоритмов и ее математического обеспечения – это определяет математик;
- в этом отношении необходима **смена предопределяющих позиций**: решающее слово должно быть за техником как (достаточно компетентным) заказчиком и хозяином, активным пользователем и участником разработок в области технических приложений алгоритмов;
- особое значение приобретает адекватная **ориентировка ответственного руководства** образованием (на всех уровнях – от кафедры до министерства) в существующей общей алгоритмической проблематике: это лица, принимающие ответственные (или безответственные) решения в области образовательных стандартов и их локальной адаптации.

Самое главное – в этой области необходимо **четкое разделение** массовых представлений относительно технической теории алгоритмов и ее математического инструментария:

не путать техническую алгоритмику (и вообще техническую информатику) с математикой – с ее инструментальным математическим аппаратом.

Теория алгоритмов кибернетики

Фактически такая теория вообще еще не существует, как специализированная теория, и пока представлена косвенно – множеством частных примеров алгоритмических задач и методов:

эти аспекты требуют самостоятельного анализа и здесь не отражаются.

Общая теория алгоритмов

Целесообразно формирование общей теории алгоритмов. При этом:

1) Трудно рассчитывать на возможность создания *единой универсальной общей теории алгоритмов*:

- попытки разработки таких теорий были в рамках математических приложений алгоритмов (часто в сопоставительных сравнениях с другими теориями);
- однако такие предельно общие, казалось бы, результаты в скором времени оказывались частными, и по разным причинам появлялись новые более общие теории;
- примерно аналогичная ситуация была с безуспешными попытками разработки единых универсальных языков программирования.

2) Однако, возможна, некоторая *метатеория* алгоритмов, обеспечивающая систематизацию и взаимную увязку частных теорий, их обобщения, выработку глубоких их общих основ и т.п. – это проблемный вопрос, требующий анализа и целенаправленных разработок.

Общая оценка состояния теории алгоритмов

Очевидно проявление известного закона диалектики – общее совпадение исторического и логического аспектов в процессах развития:

- история развития теоретического обеспечения массовой алгоритмизации в целом соответствует внутренней логике ее развития;
- однако, в историческом процессе развития возможны скачки и возвраты, отклонения, тупики и петли, параллели и пересечения и т.п. – в зависимости от разных исторических условий и обстоятельств.

При этом в области теории алгоритмов в настоящее время характерна большая *неопределенность*. Принимается интуитивная интерпретация этого термина [4]: "неполнота, неясность, нечеткость представлений, наличие сомнительных, запутанных и ошибочных положений и т.п."

Это объективная реальность, данная нам в наизидание (характерная и для многих других быстро развивающихся наук). Она подлежит анализу, проработке, оценке и т.п. с целью ее осознания и конструктивной адаптации к существующим реалиям – некоторое специальное приложение общей *индефинитики* [4] (теории неопределенностей) в науке и технике.

Выбор направлений теории алгоритмов в ее преподавании

Для разных специальностей обучения в принципе необходимо было бы изучение всех исходных общих направлений теории алгоритмов. Однако очевидны разные проблемы, в частности, большого дефицита часовых учебных ресурсов, повышения эффективности и, главное, результативности обучения, освоения новых учебных подходов, методов и средств и т.п.

Для техников, очевидно, *обязательной* должна быть *техническая теория* параллельных (и последовательных) алгоритмов – в том или ином объеме (но с обеспечением общей целостной ориентировки):

пока это проблематично, пока отдельные ее элементы присутствуют в составе разных дисциплин, и необходимы общие системные наработки.

Кроме того, техникам в той или иной мере, безусловно, необходимы:

- *математическая теория алгоритмов* – в функции математического базиса технической алгоритмизации;
- *арифметическая теория алгоритмов*, которая может быть включена в курсы типа цифровой схмотехники, вычислительной техники, основ информатики (в изучении систем счисления, арифметических операций в разных системах счисления и т.п.)

Реально пока под названием теории алгоритмов техникам, как правило, преподаётся только или в основном теория математических алгоритмов и хуже того – только или в основном классическая теория.

Эти вопросы требуют конкретного системного анализа и наработки соответствующих продуктивных системных мер.

6 НЕКОТОРЫЕ КРИТИЧЕСКИЕ (ПАРАДОКСАЛЬНЫЕ) АСПЕКТЫ

Разочарование программистов в теории алгоритмов

В одном пособии – конспекте лекций по специальной (дискретной) математике для студентов *инженерной специальности* "Автоматизированные системы управления (АСУ)" в самом начале краткого раздела по теории алгоритмов приводится такой ключевой по значению абзац:

"Главное *разочарование программистов* относительно теории алгоритмов состоит в том, что *классическая теория алгоритмов не занимается «правилами построения алгоритмов»*.

На *законный вопрос*, чем же она тогда занимается, можно *достойно ответить*: она занимается *более важной проблемой* – проблемой алгоритмической разрешимости.

То есть она занимается определением того, можно ли вообще построить алгоритм для решения задач данного типа".

Сопутствующие замечания

Сразу же появляется множество критических замечаний с позиций программиста (математика или техника), а также схемотехника и т.п.:

1) В данном разделе по имени "Теория алгоритмов" излагается именно "классическая теория алгоритмов" как раздел дискретной математики: это устаревшее традиционное узкое толкование теории алгоритмов.

2) При этом название "классическая теория алгоритмов" появляется именно в этом (указанном выше) начальном абзаце, а не в заголовке раздела, и не очень понятно в каком смысле использован этот термин:

известно ли автору что-то про "неклассическую теорию", и если известно, то почему об этом не говорится студентам этой специальности АСУ?

3) До этого раздела приводится раздел автоматов, но по ходу изложения не ясно, относится ли он к (неклассической) теории алгоритмов.

4) Проблемой алгоритмической разрешимости занимается *дескриптивная или качественная теория* алгоритмов, то есть, в указанном выше абзаце фактически приводится еще более узкое толкование теории алгоритмов – ее сведение к дескриптивной (качественной) теории:

это также распространенная традиция узкого толкования теории.

5) Однако, далее обсуждается также проблема сложности вычислений, то есть *метрическая или количественная теория алгоритмов*:

- классическая теория отражается условно в полном составе;
- то есть классическая теория алгоритмов занимается не только проблемой алгоритмической разрешимости, как указано в этом абзаце.

6) Спорной является оценка типа "более важная проблема":

- для математика, решающего *математическую задачу* определенного класса, алгоритмическая разрешимость этой математической задачи не выходит за рамки (класса) этой математической задачи;
- для техника, решающего *техническую задачу*, для которой он может привлекать *разные математические задачи* (модели), понятие алгоритмической разрешимости технической задачи приобретает несколько *другой смысл*;
- здесь работают принципиально *разные менталитеты* математика и техника (и их оценочные характеристики по "достойности" и "важности"):

для решения технической задачи техник может принять и решить общую математическую задачу, или частную задачу, или другую задачу, заменить точную модель приближенной и т.п. – "технар" всегда "вывернется" и решит свою проблему разрешимо-

сти, но классическая теория (математических) алгоритмов не дает никаких рекомендаций по этому поводу.

В принципе, такие замечания могли бы рассматриваться просто как бесконечные *придирки* к конспективному изложению краткого раздела теории алгоритмов и к его начальному абзацу. Однако, указанные выше нечеткости – это не вина автора, а совсем наоборот – большая заслуга:

здесь в миниатюре очень удачно и наглядно отражаются подобные *общепринятые* (и вполне привычные в массовых представлениях) *некорректности, противоречия и парадоксы* в области теории алгоритмов.

Уточнение "проблемы разочарования программистов"

Кроме того, большим достоинством указанного выше пособия является то, что здесь *прямо и честно говорится*, причем *с самого начала*:

- во-первых, что классическая теория алгоритмов не занимается правилами построения алгоритмов;
- во-вторых, что программисты разочарованы по этому поводу.

Фактически, поскольку в массовой подаче классическая теория алгоритмов по устаревшей традиции именуется как теория алгоритмов, такое разочарование программистов относится к теории алгоритмов вообще.

В других пособиях традиционно "наука об этом умалчивает". Правда иногда в некоторых книгах где-нибудь в конце, так или иначе, отмечается, что теория алгоритмов не занимается построением алгоритмов. Но о распространенном недовольстве программистов вообще нет речи. При этом сам факт такого массового умолчания (или утаивания) является очень странным, и представляет интерес анализ этой массовой ситуации.

В программистской литературе с алгоритмической проблематикой ситуация другая:

- часто в разных формах выражается недовольство, что теория алгоритмов (точнее, классическая теория) *"не умеет строить алгоритмы"*;
- более того, может быть жесткая критика подходов, установок и положений классической теории алгоритмов.

Например, в работе [5] критикуется некорректная сравнительная оценка мощности языков программирования путем их сведения к (гипотетической) машине Тьюринга в области классической теории алгоритмов. Поскольку все универсальные языки программирования к ней сводятся [5]:

"... споры об относительной «мощности» языков программирования – если под мощностью мы понимаем «способность решать задачи», – оказываются *бессмысленными*.

Позднее Алан Перлис ввел удачный термин для подобных аргументов, назвав их *«тьюрингова пропасть»*, поскольку из них так сложно выбраться, в то время как сами они столь *фундаментально бесполезны*".

В связи с подобными обстоятельствами появляется специфическая парадоксальная задача выяснения причин:

такой *"некачественной"* в данном отношении качественной (или дескриптивной) теории алгоритмов, которая "не умеет строить алгоритмы", и, вообще, столь *"фундаментально бесполезной"* фундаментальной (или классической) теории алгоритмов.

ЗАКЛЮЧЕНИЕ

Следует отметить, что на самом деле:

- качественная (дескриптивная) теория алгоритмов вполне качественная, но в ее корректном применении по своему прямому назначению;
- и, в целом, фундаментальная (классическая) теория алгоритмов весьма фундаментально полезная – также в своем прямом назначении.

Однако:

- есть очевидные множественные проблемы, суть которых очень неясная, противоречивая и запутанная:

это проявление фактора отмеченной выше типичной *индефинитики* [4] в приложении к алгоритмической проблематике;

- требуются конструктивный анализ данных проблем, и наработка соответствующих решительных мер для выхода из тупиков приведенного типа.

В данной статье подготовлены исходные данные для целевого анализа перечисленных в ней конкретных направлений теории алгоритмов. Дополнительно ставится проблемная задача формирования общей метатеории алгоритмов. Эти вопросы требуют самостоятельного изложения, что предполагается в других статьях автора. На сайте paralg.ucoz.com (paralg1000.narod.ru) доступны статьи [1,2], приводятся опорные рабочие данные по книге [4] и другие материалы по излагаемой тематике.

ЛИТЕРАТУРА

Статьи [1,2] автора доступны на сайте:

paralg.ucoz.com: Параллельные алгоритмы и логика

1. Житников А.П. [Параллельная \(и последовательная\) алгоритмика в массовой информатике и робототехнике](#). / В сб.: Современные технологии преподавания естественнонаучных дисциплин в системе общего и профессионального образования. – Борисоглебск: ФГБОУ ВПО "БГПИ", 2012. С. 52 – 65.
2. Житников А.П. [Исторические области и парадоксы параллельной \(и последовательной\) алгоритмики](#). / В сб.: Современные технологии преподавания естественнонаучных дисциплин в системе общего и профессионального образования. – Борисоглебск: ФГБОУ ВПО "БГПИ", 2012. С. 80 – 94.
3. Языкознание: Большой энциклопедический словарь. – М: Науч. изд-во "Большая Российская энциклопедия", 1998. – Статья "Текст". – С. 507.
4. Зверев Г.Н. [Теоретическая информатика и ее основания](#). В 2 т. Т. 1. – М.: Физматлит, 2007. – 592 с.
5. Бадд Т. Объектно-ориентированное программирование в действии. – СПб.: Питер, 1997. – 304 с.