

С.М. Крылов

**Метаматематические
основы науки будущего**

Самара

Самарский государственный технический университет

2014



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Вычислительная техника»

С.М. КРЫЛОВ

Метаматематические основы науки будущего

Самара
Самарский государственный технический университет
2014

УДК 510.8+007.2+001.8+004.388

ББК 22.31+32.817+15.13

K85

Крылов С.М.

K85 Метаматематические основы науки будущего: монография. – Самара: Самар. гос. техн. ун-т, 2014. – 247с.

ISBN 978-5-7964-1686-0

Рассматривается использование метаматематических структур, основанных на базовой концепции алгебраических систем в математике и распространяющих эти концепции на различные алгоритмические системы операций над абстрактными и физическими объектами однородного и неоднородного (гетерогенного) характера.

Такие метаматематические структуры целесообразно использовать в различных направлениях науки: в теории самовоспроизводящихся и эволюционных систем; в точной философии, в научной версии метафизики; в теории познания; в теории универсальных технологических систем различного назначения, включая универсальные дистанционно управляемые макро-, микро-, нанолаборатории и ГАП (гибкие автоматические производства); в теории интеллектуальных систем; в теории гетерогенных автоматов, включая гетерогенные электронные системы, и т.д.

Предлагаемый подход объясняет также причины успешного внедрения в современные компьютерные технологии основной парадигмы объектно-ориентированного программирования (ООП) – концепции «объекта».

Книга рассчитана на широкий круг читателей, включая студентов и аспирантов, знакомых с основами информатики, философии, математики, биологии и физики, и может быть полезна всем, кому интересны глобальные общеполитические и общенаучные проблемы и тенденции.

Рецензент: д-р техн. наук, проф. П.К. Кузнецов

УДК 510.8+007.2+001.8+004.388

ББК 22.31+32.817+15.13

K85

ISBN 978-5-7964-1686-0

© С.М. Крылов, 2014

© Самарский государственный
технический университет, 2014

ПРЕДИСЛОВИЕ

Эта книга – рассказ о совершенно новом и в каком-то смысле неожиданном направлении в науке, возникшем на основе успешной разработки формальных методов описания самых различных алгоритмических процессов, которые можно в общем случае отнести к технологическим, – от алгоритмов технологии вычислений через самые различные природные (естественные) и придуманные нами искусственные (промышленные, социальные, творческие, познавательные и прочие) технологии до технологий, используемых живой природой для реализации различных эволюционных процессов. Естественно, что объектами операций таких формальных технологий и соответствующих технологических систем могут быть (и на самом деле являются) самые различные абстрактные и реальные, однородные по своей сути (т.е. гомогенные) или же неоднородные (гетерогенные) объекты.

Соответствующее научное направление по аналогии с ОТС («Общей теорией систем») получило вполне естественное для русского языка название – «Общая формальная технология» (сокращённо – ОФТ или просто ФТ).

Это направление позволило впервые реально объединить математику и физику, химию и биологию, философию и ОТС в рамках единой научной концепции, или, как говорят учёные, в рамках единой научной парадигмы. Благодаря ОФТ удалось найти достаточно простые решения очень многих задач, казавшихся трудноразрешимыми с позиций классической математики, а также открыть новые, весьма перспективные направления развития для самых различных технологий нашей цивилизации. Именно поэтому автор считает необходимым познакомить с основными концепциями, подходами, методами и результатами ОФТ максимально широкий круг читателей.

ВВЕДЕНИЕ

Большой Российский энциклопедический словарь определяет технологию как «совокупность приёмов и способов получения, обработки или переработки сырья, материалов, полуфабрикатов или изделий...»¹. Вполне закономерно ожидать от *общей формальной технологии* (ОФТ) *формализации* этих самых «приёмов и способов» (на научном языке – алгоритмов), чем, собственно говоря, ОФТ на самом деле и занимается. Более того, как научное направление ОФТ обязана также заниматься обобщением и классификацией как самих «приёмов и способов», так и объектов, к которым такие «приёмы и способы» могут применяться. В итоге получается, что ОФТ занимается изучением, формализацией, обобщением и классификацией самых различных алгоритмов синтеза, декомпозиции и анализа любых объектов материальной, информационной или иной природы. А поскольку таковыми являются все объекты и процессы окружающего нас мира, то претензии ОФТ на междисциплинарность и общесистемность выглядят вполне естественными и обоснованными.

Первые идеи формализации и обобщения алгоритмических процессов любой, в том числе нематематической, природы восходят к Августе Лавлейс – дочери известного английского поэта Лорда Байрона и первой в мире программистке, написавшей ряд программ для тогда ещё механической цифровой вычислительной машины, которую изобрел Чарльз Бэббедж [69]. К сожалению, идеи Августы Лавлейс не нашли должного отклика в научных кругах того времени.

Со стороны же другого общенаучного направления, а именно философии, первые шаги в нужном направлении были сделаны ещё Аристотелем в его «Метафизике» [2]. Однако ближе всего к нематематической концепции формальной технологии подошел со своей «Тектологией» известный российский революционер и философ Александр Богданов [3].

¹ Большой Российский энциклопедический словарь. – М.: Научное издательство «Большая Российская энциклопедия», 2005. – 1888 с. ISBN 5-86270-324-9.

Отличие ОФТ от математической версии теории алгоритмов состоит в том, что последняя обычно рассматривает в качестве объектов формальных операций какую-либо информацию, закодированную тем или иным образом, будь то математический символ, буква на ленте так называемой «машины Тьюринга» или двоичный код, тогда как в ОФТ объектами операций могут быть не только символы (коды), но и вообще любые объекты как материальной, так и нематериальной природы. Например, довольно часто под элементарными объектами в ОФТ могут пониматься некие аналоги атомов, молекул, механических деталей, конструкций, отрезки аналоговых сигналов определенной длины и т.д., и т.п. Отсюда более очевидна и наглядна связь ОФТ с реальностью, которая позволяет объединить многие философские концепции, как известные до работ Богданова, так и выдвинутые им, со строгими научными построениями, фактически доказывающими эти концепции математическими средствами.

Абстрагирование объектов операций в ОФТ, в отличие от математики, всё же оставляет за объектами некое физическое или информационное наполнение, тогда как основной абстрактный объект интересов математики — число — такого наполнения не имеет. Единственное смысловое наполнение понятия «число» — это количество, номер или код объекта в его «чистом» виде без какой-либо возможной внутренней структуры в традиционном, техническом (а не абстрактно-математическом) смысле этого слова. Исторически такая оторванность числа от реального физического наполнения вполне объяснима: количественные характеристики окружающего мира — наиболее фундаментальные и глобальные его свойства. Поэтому вполне естественно, что основой для одной из самых первых и фундаментальных абстракций человеческой цивилизации послужило именно «количество чего-либо», получившее название числа.

Однако такая предельная оторванность, абстрактность числа, давшая математике мощный импульс в плане возможностей для саморазвития, сыграла и свою отрицательную роль, чрезмерно «уведя» некоторые фундаментальные результаты от их простого и очевидного

смыслового наполнения. Отчаянные попытки пифагорейцев насильно вернуть числу реальное физическое содержание ни к чему, кроме мистицизма, не привели. В этой связи формальная технология как раз и может оказаться одним из тех промежуточных путей к знаниям, на которых постоянно поддерживается связь с реальностью и которые проходят где-то посередине между самой реальностью и её предельно абстрактным отражением, каковым является математика.

Этот путь, как уже говорилось, способен помочь в решении многих актуальных проблем. Таких, например, как проблемы познания, эволюции неживой природы и её превращение в «живую», понимание основ и сути мироздания, возможных путей и альтернатив его развития и т.д., и т.п. Простота, с которой в ОФТ формулируются и решаются некоторые из этих задач, поражает и захватывает.

Несколько слов о структуре книги. Она коренным образом отличается от структуры многих предшествующих изданий по формальной технологии [13 – 15]. Во-первых, в её основу положена обзорная статья, опубликованная в физико-математическом «Вестнике Самарского государственного технического университета» [17], которая в расширенном и несколько изменённом варианте приводится в начале первой главы.

Во-вторых, изложение соответствующих разделов ОФТ (кроме, естественно, математики) приводится в том порядке, который следует из общего (базового) варианта, рассмотренного в самом начале в изложении вышеупомянутой статьи.

Наконец, в-третьих: излагаемый материал предельно сжат и служит главным образом целям ориентации читателя в уже полученных результатах. Предполагается, что в случае необходимости он может обратиться к существующим публикациям.

Глава 1. МЕТАМАТЕМАТИКА И ТЕОРИЯ АЛГОРИТМОВ

1.1. МЕТАМАТЕМАТИЧЕСКИЕ ОСНОВЫ «ОБЩЕЙ ФОРМАЛЬНОЙ ТЕХНОЛОГИИ»

Приставка «мета-» в русском научном языке воспринимается как указание на нечто, превосходящее исходное понятие. В этом контексте смысл термина «метаматематика» обычно понимается как «над-математика» или даже «сверх-математика», изучающая саму себя с использованием математических методов. Тем не менее в последнее время всё чаще говорят о необходимости разработки некоей новой версии математики, получившей название «теории исчисления объектов» [21, 61], в которой объектами математикоподобных алгоритмических операций служат не абстрактные объекты-символы, к которым мы привыкли в классической математике, а более или менее адекватные реальным физические объекты, точнее – их более адекватные реальным (вещественным) объектам математические модели [13 – 17, 19 – 21]. При таком расширении базовых концепций математики под метаматематическими структурами понимаются в том числе такие метаматематические конструкции, которые применимы не только к классическим математическим объектам-символам, но и к объектам иной, так сказать, квазиматериальной природы, в нашем случае – к объектам – моделям реальных физических объектов. Сравнительному анализу возможностей и перспектив подобных метаматематических структур и посвящён настоящий раздел книги.

К упомянутым выше метаматематическим конструкциям, как показано в работах [13, 15, 17, 18], можно с полным основанием отнести и одну из фундаментальнейших концепций математики, определяющую почти все её основные конструкции, а именно концепцию алгебраической системы $\mathbf{U}$, которая в трактовке А.И. Мальцева [25] формально записывается так:

$$\mathbf{U} = \langle \mathbf{A}, \Omega_F, \Omega_P \rangle, \quad (1.1)$$

где $\mathbf{A}$ – основное множество системы; Ω_F – множество операций, определенных на $\mathbf{A}$; Ω_P – множество предикатов, заданных на множестве $\mathbf{A}$.

Все другие важные математические конструкции, например алгебры и модели, являются частными случаями (1.1) [25].

В то же время в новом междисциплинарном научном направлении общей теории систем, получившем название «Общая формальная технология» (ОФТ), аналогичные по сути алгоритмические метаматематические системы операций (называемые «технологиями») над физическими объектами (точнее, как уже отмечалось раньше, над их «квазиматериальными» моделями) записываются весьма похожим образом [13, 15, 17]:

$$\mathbf{T} = \langle \mathbf{B}, \mathbf{F}_T, \mathbf{F}_A \rangle, \quad (1.2)$$

где $\mathbf{B}$ – множество некоторых физических объектов материальной природы (атомов, молекул, деталей станка, компонентов какого-либо устройства, машины, судна, летательного аппарата, фрагментов здания, сооружения и т.д., и т.п.), являющихся исходными и конечными объектами – результатами операций из множеств $\mathbf{F}_T$ и $\mathbf{F}_A$. Такие объекты обычно называют «конструкциями», а множества $\mathbf{F}_T$ и $\mathbf{F}_A$ – это конечные множества конечноместных технологических (типа «синтеза» и «декомпозиции») и аналитических (анализирующих) операций над объектами («конструкциями») из $\mathbf{B}$, причём ни одна из операций в $\mathbf{F}_T$ и $\mathbf{F}_A$ не может быть выражена через другие. Аналитические операции $\mathbf{F}_A$, в отличие от предикатов множества Ω_P , могут «измерять» различные физические свойства объектов («конструкций») из $\mathbf{B}$, которые могут иметь отображение на любую конечную числовую или нечисловую шкалу [13 – 16, 19].

Заметим попутно, что на важность двух классов алгоритмических операций с реальными физическими объектами, а именно операций типа «синтеза» (т.е. соединения, сборки объектов из «меньших» частей) и «декомпозиции» (разделения, разборки объектов на более «мелкие» части, фрагменты), в фундаментальном устройстве нашего мира обращал внимание ещё Аристотель в восьмой главе первой книги своей «Метафизики» [2, 18].

Согласно [13, 15 – 18], любой объект O_i множества **В** может быть формально представлен следующим образом:

$$O_i = \langle \mathbf{Г}_i, \mathbf{М}_i \rangle = \langle \{\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in}\}, \{\gamma_{ij} = \varphi_{ij}(\gamma_{sk}, \gamma_{sm}, \dots, \gamma_{ir}); \dots\} \rangle; \quad (1.3)$$

где O_i – i -тый физический объект; $\mathbf{Г}_i$ – список интересующих нас его физических свойств (параметров) типа γ_{ir} , характеризующих объект O_i по важным для нас признакам – по его положению в пространстве, по его массе, скорости, цвету и т.д. и т.п.; $\mathbf{М}_i$ – список интересующих нас функциональностей φ_{ij} объекта O_i , описывающих его взаимодействие с другими физическими объектами типа O_s через их физические свойства типа $\gamma_{sk}, \gamma_{sm}, \dots$, и (или) собственные физические свойства самого объекта O_i типа γ_{ir} ; n – число параметров («свойств») типа γ_{ir} , используемых в данном (текущем) представлении объекта O_i , т.е. число вторых индексов свойств объекта O_i с номерами $r \in \{1, \dots, n\}$; k, m – номера (вторые индексы) тех «свойств» и функциональностей, которые актуальны для анализа функциональных взаимодействий φ_{ij} данного объекта O_i (в текущем его представлении) с соседними (или удалёнными – если речь идёт о взаимодействии, например, через поле) объектами типа O_s . Подчеркнём, что функциональных зависимостей φ_{ij} в списке функциональностей $\mathbf{М}_i$ объекта O_i может быть столько, сколько нужно для его адекватного представления в рамках интересующей нас модели, на что указывает многоточие после формулы φ_{ij} для расчёта параметров γ_{ij} , а сами функциональности могут быть записаны в любой удобной форме – в виде функций, физических законов, алгоритмических процедур и т.д. Однако всегда в конечном формальном представлении любого объекта типа O_i число таких функциональностей конечно [15, 17 – 18]. Заметим, что формула (1.3) очень близка к базовой парадигме чрезвычайно успешной прикладной ветви современной вычислительной математики, связанной с алгоритмизацией и решением самых различных прикладных задач – от математического (компьютерного) моделирования до управления сверхсложными техническими, технологическими и даже социаль-

ными системами. Эта парадигма называется *объектно-ориентированным программированием* – ООП. Именно с помощью ООП-подхода в современных компьютерных программах представляются самые различные моделируемые и имитируемые объекты-«сущности».

Более того, мысль о том, что любые физические объекты («сущности») как системы могут быть представлены с помощью набора своих физических свойств и связанных с этими свойствами особенностей своего поведения (в нашем понимании – функциональностей) в более узком, достаточно специфическом, контексте, была высказана ещё в работе [61].

В [13, 16 – 18] показано, что в аналогичной (1.3) нотации представление любых абстрактных математических объектов может быть записано следующим образом:

$$x = \langle \xi \rangle, \quad (1.4)$$

где символ ξ целесообразно трактовать как некоторый числовой параметр, *определяющий конкретное значение не физического свойства (представляющего количество) в конкретном абстрактном математическом объекте x .*

Вследствие нефизичности свойства ξ *какие бы то ни было функциональности у абстрактных математических объектов типа (1.4) отсутствуют.*

Для простоты и наглядности лучше всего считать $\xi=x$, что часто и имеет место на самом деле. Другим характерным примером неявного использования представления (1.4) в математике служит возможность кодирования любой конечной последовательности кодов (символов), описывающих (представляющих) объект x , с помощью функции Гёделя в виде одного, пусть и очень большого, числа [18]. То есть угловые скобки $\langle, \rangle$, символ ξ и знак равенства потребовались лишь для того, чтобы специально подчеркнуть то *очень важное* обстоятельство, что числа (и коды) представляют собой особую группу абстрактных математических объектов, имеющих одно *нефизическое*

(а потому – подчеркнём ещё раз – *нефункциональное*), свойство ξ . Заметим, что на эту характерную особенность чисел, а именно на отсутствие у них физических свойств, указывал ещё Аристотель в девятой главе первой книги своей «Метафизики» [2].

Таким образом, с формальной точки зрения обе фундаментальных концепции представления алгоритмических систем над различными объектами – как над абстрактными «числоподобными» объектами в математике типа (1.1), так и над «физическими» (точнее, подобными реальным «вещественным» физическим объектам) – в ОФТ в виде (1.2), оказываются очень схожими и по сути очень близкими с учётом особенностей представления объектов в форме (1.3), (1.4). Этот факт позволяет совершенно по-новому посмотреть как на саму математику, так и на ОФТ в целом, в которой оказывается вполне уместным и даже необходимым рассмотрение некоторых математических идей, не лучшим образом себя зарекомендовавших в рамках классического математического подхода (1.1), несмотря на изначальные надежды математиков.

Заодно отметим, что фундаментальную роль концепции алгоритмов в системе наших знаний о мире признают и сами математики [1]. Многие даже считают основной предпосылкой для выработки этой концепции как раз наблюдения над различными природными процессами [60]. Более того, как уже говорилось, часто речь заходит о необходимости создания некоей формальной теории «исчисления объектов» – наподобие той, с какой мы имеем дело в вычислительной математике. При этом особо подчёркивается, что природа – в отличие от нас – использует в своих алгоритмических «вычислениях» физические объекты напрямую, но никогда – через их числовое представление (описание) типа (1.4), как мы это привыкли делать в математике. Попытка построить именно такую, алгоритмическую теорию «исчисления физических объектов», как раз и предпринята в ОФТ, включая работы [15, 18, 21].

Впервые идеи, приближающиеся к концепции алгоритмической системы типа (1.2), высказала в середине 19-го века Августа Ада Лавлейс [13, 15, 18, 69].

Что же даёт или может дать подобный формальный подход?

Во-первых, сразу же становится очевидным, что в случаях интерпретации метаматематической структуры (1.2) в виде классической алгебраической системы (1.1), а формального представления объектов её операций типа (1.3) – в виде (1.4), мы имеем дело с традиционной математикой (с соответствующими наборами аксиом). Колоссальные успехи современной математики неоспоримы.

Во-вторых, если оставить обычную интерпретацию структуры (1.2), но объекты её операций типа O_i представлять следующим образом:

$$O_i = \langle \Gamma_i \rangle = \langle \{ \gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in} \} \rangle, \quad (1.5)$$

т.е. рассматривать *только физические* свойства объектов, а множество функциональностей, вызывающих то или иное поведение самих объектов (в соответствии с их физическими свойствами), игнорировать, то мы получим канонический (первоначальный) вариант задания формальных технологий как алгоритмических систем операций над физическими или абстрактными объектами [13 – 16], включая формальные модели таких объектов в усечённом по сравнению с (1.3) виде (1.5). Однако даже в этом случае легко доказывается существование иерархий в классе универсальных программируемых технологических систем, эквивалентных по своим вычислительным возможностям машинам Тьюринга (м.Т.), но не эквивалентных друг другу по своим потенциальным технологическим (точнее даже – алгоритмическим) возможностям [13, 15, 20].

В таких метаматематических структурах легко строятся модели универсальных (со строгой, математической точки зрения) программируемых технологических систем примерно на тех же формальных базовых принципах, что и м.Т. [13 – 16]. В свою очередь, эти модели позволяют использовать их в качестве базовых при построении практических универсальных программируемых технологических систем в самых различных областях – от нанотехнологий до химии и машиностроения и от различных компактных ГАП до гибких интернет-лабораторий, предназначенных для проведения самых различных

экспериментов с широкими возможностями их дистанционной реконфигурации и программирования [13, 15, 19, 20, 33, 66]. Одна из таких интернет-лабораторий с теоретически неограниченным потенциалом реконфигурирования и программирования дистанционного выполнения различных экспериментов в области электроники разработана и используется на практике [33]. Более подробно эти вопросы рассматриваются в параграфах 1.2, 1.3 и главе 2.

В-третьих, если метаматематическая структура (1.2) интерпретируется в полном соответствии со своим приведённым выше определением, а объекты её операций интерпретируются в полном соответствии с (1.3) и (1.4), то мы имеем дело с новым направлением ОТС, а именно – с ОФТ [13, 15, 17, 20], которое позволяет не только формулировать и доказывать самые разнообразные теоремы в рамках соответствующих формальных систем [13 - 20, 66], но и сравнивать потенциальные «технологические» возможности различных технологий, включая технологию вычислительной математики $T_{\text{ВМ}}$ (то есть, фактически, технологии компьютерной обработки информации); когнитивно-интеллектуальные технологии $T_{\text{КИТ}}$, используемые человеком в процессах познания, и различные другие варианты технологий, в частности биоподобные (или эволюционные) технологии $T_{\text{ЭТ}}$, используемые природой в эволюционных процессах [15, 17]. Следуя традициям оценки потенциальных вычислительных возможностей м.т. и соответственно компьютеров, аналогичные потенциальные «алгоритмическо-технологические» возможности других (различных) технологий можно было бы условно назвать их «мощностью». Договоримся обозначать это понятие прямыми скобками: $|\dots|$. Тогда, согласно последним результатам, полученным в рамках ОФТ [17], соотношение «мощностей» указанных выше технологий можно записать следующим образом:

$$|T_{\text{ВМ}}| < |T_{\text{КИТ}}| \approx |T_{\text{ЭТ}}|, \quad (1.6)$$

причём *строгое неравенство* между двумя первыми компонентами доказывается математически и связано *не только* с различными мощ-

ностями множеств, используемых при математическом описании свойств и функциональностей соответствующих объектов в форме (1.3) – как это сделано, например, в [15, 17, 20], *но и с возможностью эффективного использования соответствующими субъектами (человеком) операций анализа с неизвестными алгоритмами функционирования – наподобие оракулов в вычислительных алгоритмах*. Поскольку оракулы, введённые в теорию алгоритмов ещё Тьюрингом, не имеют математической интерпретации в классе частично-рекурсивных функций (ЧРФ), то и соответствующие алгоритмы с оракулами оказываются по этой причине неразрешимыми в системе ЧРФ [15, 39].

Что касается приближённого равенства между двумя последними компонентами в (1.6), то, строго говоря, на данный момент такая форма записи отношений между ними связана *только* с недостаточным фактическим материалом о действительном состоянии дел, накопленным к настоящему времени. С точки зрения базовых концепций, это приближённое равенство очень сильно тяготеет к строгому, поскольку технологии, используемые живой природой в ходе реализации эволюционных процессов, и технологии, используемые человеком в процессах познания им окружающего мира, одинаковы по своей сути [15].

Соотношение (1.6) существенно отличается от аналогичного соотношения, приведённого Г. Чайтиным (G. Chaitin) на стр. 5 в его последней книге, посвящённой так называемой «метабиологии» (фактически – теоретической математической биологии) [53]. С использованием приведённых выше обозначений указанное соотношение Чайтина можно записать так:

$$|T_{\text{ВМ}}| = |T_{\text{ЭТ}}|. \quad (1.7)$$

Заметим попутно, что исходя лишь из формального представления объектов в форме (1.4) в математике и в форме (1.3) – в ОФТ и не учитывая приведённые выше мотивации, можно утверждать, что равенство (1.7) недопустимо само по себе, поскольку неэквивалентны

между собой сами объекты операций – как по своей формальной структуре, так и по внутреннему смыслу используемых в этих структурах компонент. Напомним, что в записи (1.3) предполагаются в том числе и конкретные *различные (гетерогенные) физические свойства* объектов операций, а в записи (1.4) – *только однородные (гомогенные) числовые*, не имеющие никакого конкретного физического наполнения. Математические отношения между гетерогенными величинами никак нельзя сводить к равенству, поскольку «метр» никак не может быть равен «килограмму». Если же данное замечание всё же покажется читателю спорным, то можно обратиться и к более строгим основаниям, связанным с математическими доказательствами, упомянутыми выше, а также приведёнными в ряде работ [15, 20, 56]. Более подробно некоторые из перечисленных выше вопросов рассматриваются в главах 3-5 книги.

Возвращаясь к перечислению перспектив и возможностей соответствующих метаматематических структур, отметим, в-четвёртых, что если мы по-прежнему интерпретируем структуру (1.2) в соответствии с её определением, но под объектами в форме (1.3) понимаем исключительно физические объекты (или по крайней мере их более или менее адекватные «квазиматериальные» модели), то мы имеем дело со строгой математической («научной») версией метафизики с подходящим множеством аксиом, один из вариантов которых предложен в работе [18].

Помимо того, что данная интерпретация возрождает древнюю идею о метафизике как о строгой науке, логически объясняющей устройство, происхождение и взаимодействие «вещей» окружающего нас мира, она позволяет совершенно по-новому взглянуть и на некоторые хорошо известные математические идеи, не нашедшие до недавнего времени подтверждения в рамках классической математики, в частности идеи Колмогорова и Чайтина о важности концепций математической теории сложности объектов и алгоритмов в эволюционных процессах [52].

Действительно, если в рамках этой интерпретации рассмотреть структуру сравнительно простых линейных конструкций, содержащих n объектов типа (1.3), то при выполнении некоторых достаточно простых условий оказывается, что и энтропия $\mathbf{H}$, и потенциальная функциональность Φ , и закодированная в такой конструкции информация $\mathbf{I}$ могут описываться практически одной и той же формулой:

$$\Phi = \mathbf{H} = \mathbf{I} = k \log_2 s, \quad (1.8)$$

где k – число объектов типа (1.3) в конструкции, s – некоторый системный параметр (константа для заданного типа технологии $\mathbf{T}$), учитывающий не только число r различных типов объектов в конструкции, но и возможное число функциональных состояний каждого такого объекта [15], при наличии которых $s > r$.

Для случая отсутствия функциональных состояний у всех k объектов $\Phi = 0$ и выражение (1.8) применимо только для расчёта $\mathbf{H}$ и $\mathbf{I}$.

Нетрудно видеть, что величина Φ в (1.8) растёт с ростом числа функциональных объектов в конструкции и неизбежно достигает такого значения, что оказывается достаточной для реализации функциональной сложности процесса собственного воспроизведения [15]. К сожалению, Г. Чайтину в [52, 53] не удалось уловить этот тонкий момент обнуления величины Φ при отсутствии функциональностей у объектов, образующих линейную конструкцию, хотя для строк символов (напомним: *нефункциональных, но аналогичных по структуре линейных объектов*) в (классической) математической теории сложности объектов этот факт очевиден. Подробнее все эти вопросы рассматриваются в главе 4.

Кроме того, если мы интерпретируем объекты операций исключительно в виде (1.3), то, во-первых, для них легко вводятся формальные определения эмерджентных свойств и эмерджентных функциональностей [13, 15, 16], во-вторых, так же легко вводятся определения гомогенных и гетерогенных объектов [6, 13, 15, 16, 22, 23], на основе которых может быть разработана теория гетерогенных автоматов [15, 22, 23] и теория гетерогенных электронных схем [6, 15], уже нашедшая

практическое применение [6]. Наконец, в-третьих, при определённых дополнительных условиях или требованиях, выражение (1.2) для некоторых вариантов технологий оказывается избыточным. Последнее выполняется для так называемых «эволюционных технологий» (ЭТ). Согласно определениям, приведённым в [13, 15, 16] и в ряде других работ, технология называется эволюционной, если в ней возможно получение бесконечного числа новых конструкций (объектов), отличных от полученных ранее, причём и получение таких новых конструкций, и их отличие от полученных ранее реализуются исключительно средствами самой технологии [15], т.е. *за счёт собственных функциональностей самих элементов множества $\mathbf{B}$* в варианте интерпретации (1.3). Заметим, что обоснованию этого определения ЭТ в работе [15] уделено достаточно много места. Более того, ряд последующих результатов, касающихся возможных моделей ЭТ, подтверждают правомочность и эффективность такого определения.

Что же фактически означает данное определение? Ответ очевиден: согласно ему, и множество $\Omega_F(\mathbf{F}_T)$, и множество $\Omega_P(\mathbf{F}_A)$ должны быть пустыми, т.е. любая ЭТ (которую мы обозначили как $\mathbf{T}_{ЭТ}$) может быть записана в форме, отличной и от (1.1), и от (1.2):

$$\mathbf{T}_{ЭТ} = \langle \mathbf{B} \rangle. \quad (1.9)$$

Заметим при этом, что сами объекты множества $\mathbf{B}$ в (1.9) должны обязательно отвечать выражению (1.3), что, кстати, вполне соответствует формальному представлению свойств и функциональностей отдельных «клеток» в теории «клеточных самовоспроизводящихся автоматов» фон-Неймана [34, 67].

Как ни парадоксально, запись ЭТ в форме (1.9) соответствует и нашему внутреннему интуитивно-логическому представлению о возможных формальных записях исходных структур для известной нам эволюционной биотехнологии. Действительно, общим местом считается представление начала нашей земной биоэволюции в виде некоего «первичного бульона», в котором хаотически плавают фрагменты различных биомолекул, в ходе своего взаимодействия порождающие сначала

собственное самовоспроизведение, а затем – в результате дальнейшей эволюции – тот феномен, который мы и называем «жизнь».

В работе [34] фон Нейман определил два типа функциональностей для своих клеток, необходимых для обеспечения самовоспроизведения состоящих из них автоматов: логическую и конструктивную. В работах [13, 15, 16] аналогичный анализ необходимого набора свойств и функциональностей для самовоспроизводящихся физических объектов ЭТ (фактически – для «биологических», или по крайней мере – «биоподобных» организмов), выполненный с использованием формально-технологического (по сути, метаматематического) аппарата, позволил уточнить этот список. Оказалось, что помимо указанных выше двух типов функциональностей в состав функциональностей базовых элементарных конструкций-объектов эволюционной технологии (аналогов «клеток» в технологии фон-Неймана) должны входить ещё и так называемые «динамические» функциональности, обеспечивающие перемещения объектов друг к другу с целью образования более сложных конструкций, их изменения и анализа, а также «энергетические» – для обеспечения всех процессов энергией [15, 16]. Без этих двух типов функциональностей полноценность ЭТ оказывается весьма сомнительной. Точнее, возможность определения в ней средствами самой ЭТ новизны вновь синтезируемых конструкций практически отсутствует.

Кроме того, размерность пространства для реализации в нём ЭТ, как выяснилось, должна быть не меньше 2-х, а ещё лучше – 3-х измерений [15], что совпадает с аналогичным результатом Хокинга, полученного с использованием чисто физических концепций [37].

Возможные интерпретации формальных представлений метаматематических структур в виде (1.2) и (1.3) не ограничиваются только рассмотренными выше случаями. Достаточно очевидны различные варианты комплексных (математических) исчислений, заданных в виде (1.1), в которых формальная запись объектов типа (1.4) содержит не один числовой параметр, а несколько – два, три и т.д., т.е. любое конечное их число.

Весьма интересно было бы исследовать также и свойства метаматематических (точнее, с учетом вышесказанного – метафизических) структур вида (1.2), в которых число типов функциональностей объектов, отвечающих формуле (1.3), было бы ограничено определённым конечным их числом, т.е. в них участвовал бы конечный набор *конкретных* «физических» (или «квазифизических») свойств. Частично такой анализ начат в работах [13 – 16], а также в разделах 3.1 – 3.3 настоящей книги.

Возвращаясь к центральной теме данного параграфа, а именно к анализу перспектив метаматематических структур на основе концепции алгоритмов, трудно не согласиться с метафорическим высказыванием того же Г. Чайтина о важности факта изобретения компьютеров для нашей науки: «Компьютер был изобретен для того, чтобы прояснить вопросы относительно оснований математики» [52].

Действительно, наиболее успешное на сегодняшний день в практическом плане направление программирования – объектно-ориентированное (ООП) – декларирует представление любых объектов в компьютерных программах очень близко к тому, как это сделано в (1.3). Единственное (но существенное) отличие объектов ООП – это наличие у них так называемых свойств «наследования», которые связаны со спецификой современной технологии обработки информации, значительно облегчают сам процесс программирования, но не меняют сути представления объектов ООП в форме, близкой к (1.3).

Данный факт лишний раз подчёркивает актуальность и целесообразность использования метаматематических структур типа (1.2) и (1.3) при анализе самых различных вопросов, связанных с теорией систем, включая их моделирование.

Подводя итоги нашего обсуждения, можно констатировать, что ОФТ как новое направление междисциплинарных научных исследований базируется на принципах, весьма близких к фундаментальным метаматематическим конструкциям и структурам. Однако более значительный охват ею предполагаемых объектов исследований, в первую очередь – возможность конструировать и анализировать алго-

ритмы процессов над абстрактными и *реальными* объектами, делает её весьма подходящим кандидатом на роль базового формального метаматематического аппарата для исследования самых различных научных проблем, включая строгую математическую (научную) версию метафизики, процессы эволюции, познания и многие, многие другие актуальные задачи.

1.2. ФОРМАЛЬНЫЕ ТЕХНОЛОГИИ КАК АЛГОРИТМИЧЕСКИЕ СИСТЕМЫ

Один из вариантов первой версии аксиом ОФТ, тяготеющий к аксиомам метафизики (скорее всего, неполный и не окончательный), приведён в работах [15, 18]. В целях экономии времени мы сейчас не будем их здесь приводить, а просто вернёмся к базовым положениям ОФТ.

Поскольку в представлении (1.3) имеется два типа компонентов – физические свойства объектов и связанные с ними функциональности, то и в ОФТ оказываются полезными два уровня рассмотрения: первый – с учётом только возможных физических свойств объектов и второй – с учётом как физических свойств, так и связанных с ними функциональностей.

Первый уровень ОФТ, естественно, наиболее простой. Чтобы увидеть, что он даёт, мы должны всего лишь сделать допущение, что *объектами различных алгоритмических операций могут быть не только абстрактные (информационные) объекты типа (1.4), но и любые объекты с некоторым набором физических свойств, функциональности которых для простоты игнорируются*. То есть мы должны посмотреть, как и что качественно изменится в результатах различных формальных операций, если в них будут использоваться не только числа, представленные в форме (1.4), но и объекты типа (1.5):

$$y_i = \langle \{ \gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in} \} \rangle, \quad (1.10)$$

где γ_{ij} – некоторые физические свойства объекта y_i (неважно, реального или его упрощённой модели-макета).

Говоря иными словами, мы должны всегда иметь в виду, что у объектов формальных технологических операций могут быть некоторые физические свойства. При этом при выборе самих операций мы будем придерживаться идей, высказанных ещё Аристотелем в его «Метафизике», и в дальнейшем многократно использованных и интерпретированных многими учеными, включая Александра Богданова [3]. В первую очередь это касается операций «соединения» и «разделения» объектов, упомянутых Аристотелем и вошедших в аксиоматику ОФТ. Запишем операцию соединения двух объектов следующим образом:

$$F_j(x_1, x_2) \rightarrow \langle y \rangle; \quad (1.11)$$

где F_j , говоря формальным языком, – операция, получающая из двух отдельных объектов x_1, x_2 один общий объект y . Например, объектом y может быть какое-то устройство, собранное из двух деталей x_1, x_2 . Стрелка в (1.11) просто указывает направление процесса и не рассматривается как эквивалент знака равенства.

Естественно назвать операцию (1.11) операцией типа синтеза.

Операцию разделения одного объекта x на два y_1 и y_2 можно записать так:

$$F_d(x) \rightarrow \langle y_1, y_2 \rangle. \quad (1.12)$$

Операции типа (1.12) удобно назвать операциями типа декомпозиции. Ясно, что они представляют собой целый класс различных (в том числе и технологических) операций, выполняющих разборку какого-либо объекта на составляющие его компоненты (фрагменты).

Довольно часто в ходе анализа какой-либо ситуации мы должны получить об объектах определённую информацию, например, измерить длину бруска или определить цвет получившегося химического раствора. Все такого рода операции называются операциями анализа, даже если в их основе лежит процесс измерения какого-то конкретного параметра (представляющего соответствующее свойство) объекта. Записывать эти операции удобно в следующей форме:

$$F_i(x) \rightarrow \langle x, \beta \rangle, \quad (1.13)$$

где β – искомая характеристика (параметр) объекта x . Параметр β не обязательно должен быть числовым. Например, это может быть просто словесное описание цвета раствора x . Однако исходя из наших математических и технологических традиций параметр β , скорее всего, будет представлен каким-либо числом, например, двоичным кодом цвета того же раствора x , помещенного в ячейку, автоматически определяющую цвет.

Отметим, что операции анализа F_i , как правило, не изменяют сам исходный объект x , а лишь получают конкретные значения параметра β , характеризующего x . Поэтому для простоты операции типа (1.13) иногда могут записываться так:

$$F_i(x) \rightarrow \beta,$$

т. е. повторная запись неизменившегося исходного объекта x опускается.

Для элементов вещественной природы на *макроуровне* неизменяемость исходных элементов в большинстве процессов получения информации о них (т. е. в процессах анализа) выполняется автоматически исходя из законов сохранения вещества.

Одной из наиболее простых операций анализа является операция определения наличия или отсутствия объекта:

$$F_0(x) \rightarrow \beta, \tag{1.14}$$

где $\beta = 0$, если $x = \square$ (символ $\square$ означает «пусто», т. е. x – пустой, несуществующий объект), и $\beta = 1$, если $x \neq \square$ (т. е. x – не пустой объект).

Примером операции анализа в математике может служить операция определения равенства двух объектов x_1 и x_2 , которую в ОФТ-интерпретации можно записать следующим образом:

$$F_E(x_1, x_2) \rightarrow \beta, \tag{1.15}$$

где $\beta = 1$, если $x_1 = x_2$, и $\beta = 0$ в противном случае. В более традиционном виде эту же операцию можно записать так:

$$F_E(x_1, x_2) = \begin{cases} 1, & \text{если } x_1 = x_2, \\ 0, & \text{если } x_1 \neq x_2. \end{cases}$$

В математике подобные операции обычно называют предикатами и обозначают буквой P . Сразу отметим существенную разницу между операциями анализа в ОФТ и предикатами. В математике исходные объекты и значения предикатов, вычисляемые для них, – всегда абстрактные объекты (символы, числа или какие либо другие объекты-элементы какого-либо множества), причём значение предикатов принимает только две «величины»: «истина» (или «1») и «ложь» («0»). В ОФТ под исходными объектами операций анализа могут подразумеваться в том числе и реальные объекты, а результатами анализа почти всегда являются как сами исходные объекты (вследствие закона сохранения вещества), так и конкретные (числовые или нечисловые) значения параметров, представляющих интересующие нас свойства объектов. При этом с использованием математики легко доказать, что любая операция анализа, дающая конечное число из m возможных значений любого параметра, эквивалентна набору из m или $(m-1)$ подходящих предикатов, сравнивающих измеренное (т.е. конкретное) значение этого параметра с одним из m или $(m-1)$ возможных.

Чрезвычайно важной и полезной в ОФТ и во многих реальных технологиях оказывается операция анализа (предикат), позволяющая установить соответствие между некоторыми заданными (своего рода эталонными) формами и конкретной пространственной формой анализируемого объекта. Такая операция часто именно так и называется – предикат, определяющий свойство анализируемого объекта «иметь заданную форму». Обычно он обозначается как P_S :

$$P_S(x) \rightarrow \beta, \quad (1.16)$$

где $\beta = 1$, если форма объекта x соответствует заданной, и $\beta = 0$ в противном случае.

Числовые и нечисловые параметры могут использоваться не только в операциях анализа, но и в других технологических операциях. Например, если необходимо нагреть порцию жидкости до определённой температуры, то такую операцию в общем случае можно представить как

$$F_{ev}(x_i, \alpha) \rightarrow x_j, \quad (1.17)$$

где F_{ev} – операция с одним числовым параметром α , позволяющая из элемента x_i получить новый элемент $x_j \neq x_i$. Действительно, если, например, порцию воды x_i некоторое время подержать при температуре $\alpha = 100\text{ C}^\circ$ и нормальном атмосферном давлении, то вода превратится в пар (полностью или частично). Разумеется, при этом общее число молекул воды в каком-то ограниченном пространстве «вблизи» операции F_{ev} не изменится, однако первоначальной порции x_i с первоначальной температурой в том же месте, в том же агрегатном состоянии и в том же количестве, что и раньше, уже существовать не будет.

Технологические операции с одним числовым параметром типа (1.13), (1.17) или даже с несколькими числовыми параметрами довольно часто используются в различных программно-управляемых технологических системах, например, в операции умножения выборок аналоговых сигналов на фиксированные масштабные коэффициенты в системах сбора данных [13 – 16] или в операции, обеспечивающей установку какой-либо детали в заданное место (с заданными координатами) собираемого узла, операции фрезерования деталей по заданной траектории, на заданную глубину, и т.д. и т.п.

Некоторые другие виды операций синтеза, декомпозиции и анализа будут рассмотрены в следующих параграфах.

Обобщая все возможные типы операций формальной технологии, соответствующие выражениям (1.6) – (1.17), а также те, которые можно «вывести» из этих операций, вводя дополнительные объекты или параметры, нетрудно записать соответствующую «общую формулу» для всех таких операций:

$$F_i(x_1, x_2, \dots, x_n, \alpha_1, \alpha_2, \dots, \alpha_t) \rightarrow \langle y_1, y_2, \dots, y_m, \beta_1, \beta_2, \dots, \beta_s \rangle, \quad (1.18)$$

где $x_1, x_2, \dots, x_n$ – исходные объекты операции F_i , $y_1, y_2, \dots, y_m$ – новые (или прежние) объекты, являющиеся результатами операции F_i ; $\alpha_1, \alpha_2, \dots, \alpha_t, \beta_1, \beta_2, \dots, \beta_s$ – последовательности (цепочки) неких числовых или нечисловых значений (параметров) из конечных числовых или нечисловых множеств. Для сокращения записи цепочек параметров

будут использоваться обозначения типа: $\alpha = \alpha_1, \alpha_2, \dots, \alpha_i$. Аналогичные сокращения (типа x) могут использоваться и для записи последовательностей объектов (т.е. вместо $x_1, x_2, \dots, x_n$).

Таким образом, одним из центральных понятий в формальной технологии является понятие «объекта» технологических операций. Как и в реальных технологиях, а также в соответствии с декларированными выше концепциями, этими объектами могут быть любые объекты технологических операций: реальные или воображаемые фрагменты и узлы зданий и сооружений, конструкций и изделий, любые реальные или воображаемые порции вещества (в любом агрегатном состоянии и в любых объёмах, включая макро-, микро- и нано-порции), атомы и молекулы, их модели, и т.д., и т.п.

Чтобы искусственно не ограничивать области применения формально-технологических методов и результатов исследований, в состав возможных объектов формально-технологических операций включаются в том числе и любые информационные объекты, соответствующие интерпретациям (1.3), (1.4), (1.5), (1.10). При этом предполагается, что такие объекты могут быть физически представлены любыми типами и формами сигналов (цифровых и аналоговых, электрических и пневматических, гидравлических и световых). Кроме того, как уже не раз отмечалось, объектами операций в ОФТ могут быть разнородные (гетерогенные) «смеси» информационных и материальных объектов (как, например, в случае операций анализа или технологических операций с параметрами типа (1.18)) и т.д. Иными словами, под объектами в ОФТ на самом деле может пониматься всё, к чему могут быть применены технологические операции и операции анализа.

Во многих реальных технологиях все производные конструкции строятся (создаются, синтезируются) из некоторого исходного набора достаточно простых и стандартных объектов (веществ, деталей, электронных и прочих компонентов и т.д.). Например, все химические и биохимические технологии на самом деле используют конечный набор элементов – атомов различных типов, представленных в периодической таблице Менделеева. Подавляющее большинство совре-

менных микроэлектронных схем построены из резисторов, конденсаторов, транзисторов и диодов (того или иного типа). В теории двоичных функций в качестве исходных объектов-чисел рассматриваются только ноль и единица. Зарождавшиеся в глубокой древности строительные технологии использовали в первую очередь ограниченный набор подручных материалов: камни, глину, дерево, шкуры и кости больших животных и т.д. Естественно, имеет смысл как-то выделить такие первоначальные, исходные множества объектов. В ОФТ подобные множества объектов принято называть базовыми элементами или базой технологии. То есть среди всего многообразия объектов операций в любой формальной технологии обычно выделяется некоторое первоначально заданное множество объектов, называемое «**базой технологии**» или «**элементами базы**», которое часто обозначается латинской буквой **A**.

Теперь мы можем ввести каноническое *формальное* определение самой формальной технологии:

Каноническое определение формальной технологии. Под формальной технологией **T** понимается упорядоченная пара $\langle \mathbf{A}, \mathbf{F} \rangle$, состоящая из двух конечных множеств: множества **A** некоторых исходных элементов (объектов) материальной или нематериальной природы (к примеру, абстрактных объектов, в том числе объектов информационного характера – чисел и/или символов), называемых обычно **элементами базы** (или **базовыми элементами**): $\mathbf{A} = \{a_1, a_2, a_3, \dots, a_n\}$, и множества конечноместных операций $\mathbf{F} = \{F_1, F_2, \dots, F_m\}$ над элементами базы и (или) объектами (конструкциями), полученными в каких-либо предыдущих операциях технологии **T**, причём ни одна из операций в **F** не может быть выражена через другие.

Предполагается, что элементы базы – различные (или, по крайней мере, отличимые друг от друга) объекты, хотя первоначально этого может и не быть, а число одинаковых экземпляров объектов одного вида, например, a_i , «достаточно велико». То есть подразумевается, что для каждого объекта – элемента базы имеется некий «источник», откуда эти элементы можно черпать в нужном количестве. Таким образом, пере-

числение элементов базы $\{a_1, a_2, \dots, a_i, \dots, a_n\}$ фактически означает перечисление «источников» или «типов» элементов $a_1, a_2, \dots, a_i, \dots, a_n$.

Множество новых объектов, отличных от элементов базы **A** и полученных в результате каких-либо технологических операций в **T**, обычно называется **конструкциями** и обозначается символом **B** или каким-либо другим, специально оговариваемым символом. Таким образом, запись $A \cup B$ означает множество исходных элементов (элементов базы **A**) и множество объектов (конструкций), полученных путём применения операций из **F** к элементам базы и/или ранее полученным конструкциям. Использование для объектов из **B** термина «конструкция» подчёркивает конструктивный путь происхождения этих объектов из начального множества **A** путём выполнения какого-либо *алгоритма* с применением операций из **F**. Вообще же термины «элемент», «объект» и «конструкция» **считаются в ОФТ равноправными** в смысле их возможного использования в качестве имен для различных объектов операций. В противном случае чётко оговариваются соответствующие ограничения.

Поскольку в общем случае множество **B** может включать в себя и все элементы из **A**, наиболее общее определение формальной технологии записывается так: $T = \langle B, F \rangle$.

Напомним, что последнее определение хорошо совпадает с определением А.И. Мальцевым алгебраической системы в математике [25]. Согласно этому определению, алгебраической системой называется тройка $U = \langle A, \Omega_F, \Omega_P \rangle$, где **A** – основное множество системы **U**; Ω_F – множество операций, определённых на **A**, и Ω_P – множество предикатов, заданных на множестве **A**. Отличие, таким образом, заключается, во-первых, в том, что в общем определении ФТ под элементами множества **B** понимаются объекты, соответствующие представлению (1.5), тогда как в алгебраической системе под основным множеством **A** понимается обычно множество абстрактных объектов операций, отвечающих представлению (1.4). Во-вторых, в определении алгебраической системы под объектами операций в явном или

неявном виде подразумеваются только однородные (гомогенные) абстрактные математические объекты (числа, коды, символы, строки, другие математические структуры и конструкции, соответствующие (1.4)), а под операциями над этими объектами – только математические операции и предикаты. В случае ОФТ объектами операций могут быть любые объекты, в том числе и разнородного (гетерогенного) характера (т.е. у которых списки свойств значительно отличаются друг от друга по своему содержанию), включая гетерогенные физические компоненты различных природных и технических (искусственных) систем, а операциями над ними – самые различные физические, химические, технические, технологические и даже социальные операции, включая операции сборки (синтеза), разборки (декомпозиции), анализа (некоторых параметров или свойств) объектов и т.д.

Ничего удивительного в таком совпадении определений нет. Поскольку мы предполагаем отсутствие у объектов функциональностей, все операции («функциональные преобразования») над ними должны выполняться внешними по отношению к ним субъектами, например, машиной (автоматом) или человеком. Отсюда – необходимость во «внешних» по отношению к объектам операций технологических операциях и операциях анализа (предикатах). Для математики это естественно и очевидно вследствие отсутствия функциональностей у абстрактных математических объектов – чисел или кодов. Для ОФТ ситуация остаётся аналогичной до тех пор, пока мы игнорируем возможные собственные функциональности объектов согласно интерпретациям (1.5) или (1.10) – см. главу 3.

Базовым понятием любых алгоритмических процессов является так называемый «шаг алгоритма», то есть любая стандартная элементарная операция, входящая в процедуру выполнения данного алгоритма. Для того чтобы можно было использовать различные формальные алгоритмические построения в ОФТ, мы тоже должны ввести *формальное* определение шага технологии. Делается это на основе выражения (1.18):

Определение шага технологии. Однократное применение какой-либо операции $F_i \in F$ к элементам из $A \cup B$ называется шагом технологии $T = \langle A, F \rangle$ и записывается в общем виде следующим образом:

$$F_i(x_1, x_2, \dots, x_n, \alpha_1, \alpha_2, \dots, \alpha_t) \rightarrow \langle y_1, y_2, \dots, y_m, \beta_1, \beta_2, \dots, \beta_s \rangle.$$

Введение параметров в качестве возможных исходных объектов операций и их результатов, как уже отмечалось, необходимо по нескольким причинам. В первую очередь, многие реальные технологические операции на самом деле требуют определённой числовой «настройки» – указания температуры нагрева, скорости вращения сверла или шпинделя, интенсивности и длительности перемешивания ингредиентов смеси и т.д. и т.п. Во-вторых, некоторые технологические операции сопровождаются изменением определённых параметров, которые важны для последующего процесса, или же параметры обрабатываемых объектов измеряют специально, для контроля, при этом результаты таких измерений обычно отображаются на конечные числовые шкалы (рациональных или целых чисел) с какой-то фиксированной (конечной) точностью.

Подведём теперь некоторые предварительные итоги относительно введённых выше определений и типов технологических операций. Первое, что имеет смысл отметить, это то, что в отличие от классической концепции алгебраической (и, соответственно, алгоритмической) системы в ОФТ *разрешаются* в том числе *нематематические* (то есть технологические) *аналоги операций* (типа «синтеза», «разделения», «отъятия», «отображения на числовую ось», т.е. измерения параметров, и т.д.) *над неинформационными объектами любой природы*.

Во-вторых, необходимо постоянно иметь в виду, что формально-технологическая концепция множества элементов (объектов) технологических операций включает в себя в том числе и разнородные (гетерогенные) объекты, содержащие различные перечни свойств.

В-третьих, вместо узкой концепции предиката, используемой в математике, в ОФТ вводится концепция аналитических технологиче-

ских операций, которые фактически отображают интересующие нас свойства объектов на числовые (или нечисловые) множества.

Помимо чисел, в математике очень важную роль играют такие понятия, как функции. Обычно они задают правила, по которым можно вычислить конкретное значение функции при тех или иных значениях переменных, от которых она зависит (такие переменные называются аргументами). Всё множество возможных действительных значений числовой функции называется областью её определённости, а множества значений аргументов, при которых значения функции определены - областями задания функции. В самом общем виде функция от нескольких переменных $x_1, x_2, \dots, x_n$, может быть записана так:

$$y = f(x_1, x_2, \dots, x_n),$$

где y – значение функции, символ f обозначает правила (алгоритм) её вычисления, а число аргументов n (число мест для переменных в скобках) называется «местностью» функции.

Введём аналогичное формально-технологическое понятие – особую формальную конструкцию, которую условно можно назвать «формально-технологической функцией» и которая в ОФТ выполняла бы примерно такую же роль, как и различные функции в математике. Для удобства соответствующую конструкцию мы так и назовём: ОФТ-функция (в некоторых предыдущих версиях ОФТ она не совсем корректно была названа «схемой синтеза» [13]).

Определение ОФТ-функции. В заданной технологии T n -местной ОФТ-функцией называется процесс получения или анализа конструкций в T путём применения определённых последовательностей операций из F к набору элементов, принадлежащих множеству $A \cup B \cup \Omega$, где $A \cup B$ – область (множество) допустимых элементов и конструкций для данной ОФТ-функции, Ω – конечное множество допустимых параметров технологических операций и/или результатов операций анализа, участвующих в данной ОФТ-функции. В ОФТ-функции может также участвовать любое конечное число «фиксиро-

ванных» (неизменяемых) элементов, конструкций и (или) параметров из $A \cup B \cup \Omega$.

В этом определении набор из n элементов и параметров можно интерпретировать как соответствующие n аргументов ОФТ-функции, а используемые «фиксированные» элементы из объединённого множества $A \cup B \cup \Omega$ – как «константы» («фиксированные коэффициенты», «постоянные множители» и т.д.). Последовательности операций определяют порядок «вычисления» ОФТ-функции для возможно различных её ветвей, то есть реализацию её алгоритма. Упоминание о допустимости тех или иных значений тех или иных «аргументов» ОФТ-функции – не что иное, как ссылка на области задания формально-технологической функции.

Примерами простейших ОФТ-функций с соответствующими областями задания могут служить операция (1.17) нагрева порции жидкости с параметром α : $F_{ev}(x_i, \alpha)$ или операции (из следующего параграфа 1.3) типа примитивного синтеза F_s или просто синтеза F_t . Области задания для них, согласно определениям и сути самих операций, являются:

i) для операции с параметром $F_{ev}(x_i, \alpha)$: множество жидкостей из $A \cup B$ – для x_i , а для α – множество допустимых параметров Ω (например, для воды в нормальных условиях – диапазон от 0°C до 100°C);

ii) для операции примитивного синтеза $F_s(x_1, x_2)$: объединённое множество $\square \cup A \cup B$ для x_1 и множество $\square \cup A$ – для x_2 (напомним, что символом $\square$ в ОФТ обозначается пустой, отсутствующий элемент);

iii) для операции синтеза $F_t(x_1, x_2)$: множество $\square \cup A \cup B$ – как для x_1 , так и для x_2 .

Чтобы отличить «ОФТ-функцию» от традиционного («классического») понимания отдельного, конкретного синтеза, введём соответствующее определение:

Определение синтеза в ОФТ. Синтезом конструкции R называется процесс её получения в T путём последовательного применения операций из F к конечному числу конкретных элементов из объединённого множества $A \cup B \cup \Omega$.

Таким образом, ОФТ-функция и конкретный синтез конкретной конструкции могут иметь абсолютно одинаковую последовательность операций из $\mathbf{F}$, но отличаются в этом случае друг от друга тем, что ОФТ-функция позволяет использовать в качестве каждого из исходных n элементов некоторый достаточно разнообразный их набор из некоторого подмножества множества $\mathbf{A} \cup \mathbf{B} \cup \mathbf{\Omega}$ (совпадающего с областью «задания» ОФТ-функции), тогда как в аналогичном конкретном синтезе могут использоваться только конкретные элементы из того же множества $\mathbf{A} \cup \mathbf{B} \cup \mathbf{\Omega}$. Иначе говоря, конкретный синтез – это частный случай реализации соответствующей ОФТ-функции. Или (по аналогии с функцией в математике) конкретный синтез – это конкретный случай «вычисления» ОФТ-функции с конкретными значениями «аргументов». В дальнейшем любую j -тую n -местную ОФТ-функцию мы будем записывать как

$$\mathbb{F}_j(x_1, x_2, x_3, \dots, x_n) \rightarrow y,$$

где $x_1, x_2, x_3, \dots, x_n \in \mathbf{A} \cup \mathbf{B} \cup \mathbf{\Omega}$, и каждый из элементов (конструкций) x_i берётся из «области задания ОФТ-функции», а конкретный синтез конструкции R будем записывать так:

$$C_R(a_1, \dots, a_i, \dots, b_1, \dots, \alpha_t, \dots, a_k, \dots, b_j, \dots, \alpha_s) \rightarrow R, \quad (1.19)$$

где $a_1, \dots, a_i, \dots, a_k \in \mathbf{A}$, $b_1, \dots, b_j \in \mathbf{B}$, $\alpha_t, \dots, \alpha_s \in \mathbf{\Omega}$.

Если «начальное» множество исходных аргументов математических функций конечно (как, например, в примитивно-рекурсивных или частично-рекурсивных функциях, где исходным может быть множество $\{0, 1\}$, состоящее всего из двух чисел), то при их сопоставлении с ОФТ-функциями следует иметь в виду и конечное (согласно определению) множество типов исходных элементов в базе $\mathbf{A}$ технологии. Такое сопоставление иногда позволяет по-новому взглянуть на некоторые фундаментальные положения математики, а также легко переводить соответствующие математические теоремы в утверждения формальной технологии (и обратно) [13-16].

После всех этих определений и уточнений пришло время выяснить, что же могут дать введённые выше принципы, если в дело вступит та самая теория алгоритмов, с которой так много надежд (и, как мы убедимся, небезосновательно) связывали и связывают многие ученые.

1.3. РАЗЛИЧНЫЕ ТИПЫ ТЕХНОЛОГИЙ И ИХ СВОЙСТВА. ЭВОЛЮЦИОННЫЕ И ПОЛНЫЕ ТЕХНОЛОГИИ

Ряд свойств и особенностей различных технологий оказываются близкими или даже одинаковыми независимо от степени их сложности, то есть от числа и разнообразия компонентов в множествах **A** и **F**. Ясно, что анализировать и изучать эти особенности лучше на простых примерах. К таким простейшим технологиям имеет смысл отнести те, которые содержат минимальное число компонентов в множествах **A** и **F**, причём операции из **F** также должны иметь достаточно простой и очевидный смысл с точки зрения их действия на технологические объекты, то есть на элементы базы и конструкции из них.

Рассмотрим некоторые из разновидностей простых операций, введённых в параграфе 1.2, а также некоторые формальные свойства соответствующих технологий. Для облегчения восприятия некоторые второстепенные детали будут опущены, как и доказательства многих теорем. Желаящим познакомиться с ними можно порекомендовать работы [13-15]. Итак:

Определение операции «примитивного синтеза». Операцией примитивного синтеза называется операция $F_S(x, a) \rightarrow y$, присоединяющая (неважно каким образом) элемент $a \in \mathbf{A}$ к конструкции (элементу) $x \in \mathbf{A} \cup \mathbf{B}$, где **B** – множество конструкций, полученных на предыдущих шагах технологии, причём для случая $F_S(a, x)$, $x \in \mathbf{B}$ результат операции считается неопределённым (то есть порядок использования элементов в операции F_S тоже имеет значение).

Последнее обстоятельство (важность порядка использования элементов в операции F_S) сразу выводит соответствующий тип тех-

нологий за рамки алгебраических структур, называемых полугруппами. Такие структуры исследовались некоторыми философами на предмет их применимости для построения так называемой «точной философии». По отмеченным в [13-16, 81] причинам эти попытки не дали ожидаемых результатов (см. также М. Бунге, [49, 50]).

Примерами операций примитивного синтеза могут служить:

- 1) операция пришивания аминокислотного остатка к синтезируемой белковой цепочке в рибосомах;
- 2) операция укладки очередного кирпича в сооружаемое здание;
- 3) функция следования в системе примитивно-рекурсивных функций [26], добавляющая единицу к полученному ранее числу;
- 4) операция приписывания одной буквы к концу слова;
- 5) операция установки очередной детали в собираемый узел.

Нетрудно видеть, что операция примитивного синтеза $F_S(x, a)$ является разновидностью операций синтеза типа (1.11), но может присоединять к любой конструкции x только элементы базы. Между тем для расширения технологических возможностей было бы полезно иметь и другие операции типа (1.11), способные соединять между собой любые две конструкции. В реальных «естественных» (т.е. природных) и искусственных технологиях чаще всего используется именно такой вариант:

Определение операции синтеза. Операцией синтеза называется операция $F_t(x, y) \rightarrow x*y$, где $x, y \in A \cup B$, символ $\in$ означает принадлежность элементов (конструкций) x и y к объединённому множеству A и B , а символ $*$ означает соединение двух конструкций (элементов) x и y каким-либо образом, причём может иметь место:

$$F_t(x, y) \neq F_t(y, x), \text{ если } x \neq y;$$

т. е., как и для операции примитивного синтеза, порядок использования элементов (конструкций) в операции F_t также может иметь значение (но может, естественно, и не иметь).

Фактически в этом определении уточняется операция синтеза, введённая выражением (1.11).

Примеры операций синтеза:

- 1) операция сложения в математике;
- 2) добавление слова (нескольких слов) к концу предложения;
- 3) операция добавления порции раствора к другому раствору;
- 4) операция сложения двух уровней напряжения в технологии обработки аналоговых сигналов;
- 5) операции укладки очередной порции бетона в сооружаемое здание;
- 6) операции сборки устройства из узлов и деталей.

Заметим, что операции 5) и 6) в реальности, скорее всего, потребуют введения параметров, указывающих места укладки бетона или установки узлов и деталей, то есть тяготеют к операциям типа (1.17).

Разумеется, могут существовать и существуют другие разновидности операций синтеза, – например, когда соединяются (реагируют) между собой две различные молекулы, не всегда в состав нового синтезируемого вещества входят целиком атомы обеих молекул – часть атомов может образовывать молекулы другого вещества (или даже нескольких веществ). Пока мы оставим такие операции в стороне.

Очевидно, наибольший интерес (как в плане непрерывного развития возможностей технологии, так и с сугубо практической точки зрения) могут представлять технологии, в которых процесс получения (т.е. синтеза, создания) новых конструкций может продолжаться сколь угодно долго. Для работы с такими технологиями, ***возможности которых (что чрезвычайно важно!) потенциально не ограничены исходными множествами $\langle A, F \rangle$*** , вводятся следующие определения:

Определение креативной операции. Операция F называется креативной (т. е. создающей нечто новое), если в результате её выполнения получается новый элемент (конструкция) $y \notin A$ (символ $\notin$ означает «не принадлежит»).

Могут существовать и такие креативные операции, которые лишь однажды дают новый элемент (конструкцию). Естественно, с точки

зрения развития технологий они малоинтересны. Поэтому мы введём определение более мощной в этом смысле креативной операции:

Определение бесконечно-креативной операции. Операция F называется бесконечно-креативной, если при каждом её применении к элементам и конструкциям из $A \cup B$ можно получить новый элемент $y \notin A \cup B$, где B – множество элементов (конструкций), полученных на предыдущих шагах T .

Существование в технологиях бесконечно-креативных операций позволяет им в принципе развиваться неограниченно. Введём соответствующее *формальное* определение:

Определение бесконечно-креативной технологии. Технология, в которой можно получать неограниченное множество новых элементов $y \notin A \cup B$, отличных от полученных ранее, называется бесконечно-креативной.

Потенциальная способность бесконечно-креативной технологии к непрерывному развитию, к построению всё новых и новых оригинальных конструкций создаёт очень интересную возможность, а именно – возможность существования *саморазвивающихся* технологий, для которых нет пределов роста, самосовершенствования и *которые в потенциале способны породить нечто, похожее на разум*, на интеллект. Такие технологии естественно назвать *эволюционными*. Но как их различить? Какой *формальный* (и, желательно, простой) критерий нужно добавить к критерию бесконечной креативности, чтобы мы могли легко отличить эволюционные технологии от остальных?

Ответов может быть несколько. Один из лежащих на поверхности – это возможность появления среди конструкций эволюционной технологии таких, которые могут сами себя воспроизводить. На первый взгляд данный подход представляется вполне обоснованным, поскольку самовоспроизведение – очевидное свойство существующей на Земле эволюционной биотехнологии. Однако при этом вопрос о возможности появления среди её конструкций чего-то похожего на интеллект повисает в воздухе, поскольку прямой связи между самовоспроизведением и интеллектом нет.

Чтобы снять эту проблему, к условию самовоспроизведения следует, вероятно, добавить возможность построения в эволюционной технологии конструкций, способных, например, имитировать работу так называемой универсальной машины Тьюринга – своеобразного «формального аналога» интеллектуальных (точнее, вычислительных, согласно тезису Тьюринга-Чёрча) способностей человека. Для читателей, незнакомых с этими машинами, будет полезна следующая информация: машина Тьюринга (м.Т.) – это относительно простая модель автомата, точнее, программируемого вычислительного устройства, способного вычислять любые вычислимые (в принципе) математические функции. Соответственно, универсальная м. Т. – автомат, способный имитировать работу любых машин Тьюринга.

И всё же оба перечисленных выше требования не слишком корректны и не слишком удобны в рамках избранного нами формально-технологического подхода. Во-первых, возникает вопрос: на самом ли деле самовоспроизведение является обязательным условием эволюции? Ведь «эволюция» в переводе с латинского – «развёртывание», т.е. в первую очередь развитие, изменение. Во-вторых, существуют модели универсальных программируемых вычислительных устройств, эквивалентных по своим возможностям машинам Тьюринга, но имеющих совсем другую конструкцию. В-третьих, могут существовать и неизвестные нам модели интеллекта. Не обязательно ведь эволюция должна приводить к появлению чего-то похожего на м.Т. или компьютеры. То есть предполагаемая связь эволюционности технологии с самовоспроизведением и интеллектом (хотя бы на уровне машин Тьюринга) при ближайшем рассмотрении оказывается не очень оправданной. Как же эти требования скорректировать, оставив, в принципе, здравую идею появления в эволюционной технологии «чего-то похожего на развитие» и «чего-то похожего на интеллект»?

Определение эволюционной технологии. Технология **Т** называется эволюционной (сокращённо ЭТ), если в ней возможно получение бесконечного числа новых конструкций, отличных от полученных ранее, причём и получение новых конструкций, и их отличие от полученных ранее реализуются исключительно средствами самой технологии **Т**.

Заметим, что первый процесс – получение бесконечного числа новых конструкций – действительно способствует развёртыванию потенциальных возможностей технологии, так как если какая-то (даже очень сложная) из конструкций или их совокупность могут на самом деле обладать некоторыми интересными свойствами, например, интеллектуальными, то есть шанс, что рано или поздно такие конструкции будут получены. Причём не суть важно, с помощью каких конкретных механизмов будет реализовываться процесс оценки новизны вновь синтезируемых конструкций – путём ли запоминания всех особенностей (свойств!) полученных ранее конструкций и сравнения их со свойствами вновь получаемых или как-то иначе. Главное, что такой процесс в рамках эволюционной технологии имеет место и может быть реализован средствами самой ЭТ!

Важной особенностью введённого выше определения эволюционности является то, что, согласно ему, новизна конструкций должна определяться исключительно средствами самой технологии. То есть в технологии должны существовать конструкции (построенные из элементов базы **A**), реализующие необходимые операции, и алгоритмы, связанные с этими конструкциями, которые в совокупности позволяли бы выполнять такого рода анализ «на новизну». При этом никаких внешних субъектов или иных факторов, организующих эти процессы и конструкции извне и управляющих ими, быть не должно! Говоря иными словами, соответствующие управляющие алгоритмы должны быть реализованы в виде каких-либо других конструкций (фактически – «программ») в той же самой технологии. Но чтобы конструкции в технологии смогли выполнять какие бы то ни было функции (операции), образующие их элементы сами должны обладать функциональностями. Иначе функциональностям конструкций просто неоткуда взяться!

Любопытно, что именно такая функциональность базовых элементов (отдельных ячеек-клеток) и положена в основу существующих моделей самовоспроизводящихся клеточных автоматов. Причём уровень функциональности, обеспечивающий самовоспроизведение таких автоматов, оказывается не так уж и высок. Например, самовос-

производящиеся автоматы Кодда содержат ячейки всего с восемью различными состояниями, то есть с восемью возможными значениями результатов вычисления функций типа φ_j в выражении (1.3). К тому же, как будет показано дальше, введённое выше определение эволюционных технологий действительно соответствует необходимым и на самом деле оказывается близким к достаточным условиям для технологий самовоспроизводящихся автоматов, то есть для тех технологий, которые, как считают некоторые учёные, близки к технологиям, используемым живыми организмами.

Понятие бесконечно-креативной технологии более широкое, чем понятие ЭТ, поэтому следует ожидать, что не все бесконечно-креативные технологии могут быть эволюционными. Например, как показано в [13, 14], технологии, содержащие только операцию F_s или только операции типа F_t , являясь бесконечно-креативными, оказываются не эволюционными по достаточно простой и очевидной причине – они не содержат операций анализа. А поскольку мы заранее оговорили отсутствие у элементов базы собственных функциональностей, возможным операциям анализа неоткуда появиться, какие бы сложные конструкции в такой технологии не синтезировались. С другой стороны, очевидно, что все эволюционные технологии, согласно введённому определению, должны быть бесконечно-креативными. Таким образом, справедливо:

Утверждение о взаимоотношении эволюционных и бесконечно-креативных технологий. Любая эволюционная технология бесконечно-креативна. Обратное неверно.

В качестве иллюстрации свойств некоторых простейших технологий перечислим соответствующие результаты из [13, 14, 19]:

Утверждение 1.1. Технология с бесконечно-креативной операцией (или операциями) бесконечно-креативна.

Утверждение 1.2. Операция примитивного синтеза бесконечно-креативна.

Утверждение 1.3. Технология с одним-единственным элементом в базе A и единственной операцией примитивного синтеза (т.е. технология $T_s = \langle \{a\} \{F_s\} \rangle$) бесконечно-креативна.

Утверждение 1.4. Если в технологии T имеют место операции синтеза (отвечающие введённому выше определению), то T бесконечно креативна.

Теперь было бы интересно рассмотреть «устройство» возможных алгоритмов синтеза всё новых и новых конструкций в бесконечно креативных технологиях. Это позволило бы, во-первых, понять, насколько сложен сам процесс развития технологий в плане реализации всё новых и новых процессов, во-вторых, оценить необходимый для такого развития уровень сложности гипотетических автоматов, которые могли бы его (это развитие) реализовать, и в-третьих – сориентироваться в проблемах, связанных с автоматическим познанием самих технологий. Ответ на эти вопросы даёт следующее обобщающее утверждение ОФТ:

«Теорема об эффективности накопленных знаний». Для любой технологии с конечным числом типов элементов в базе A и конечным числом конечноместных операций в множестве F , включающем операции синтеза, существует алгоритм, генерирующий множество синтезов, имеющих взаимно-однозначное отображение на непрерывный ряд натуральных чисел $0, 1, \dots, n$, что позволяет по номеру (условному «названию») синтеза восстановить сам синтез в данной технологии.

Чуть позже мы рассмотрим причины, почему это утверждение получило название «теоремы об эффективности накопленных знаний» наряду с другими аналогичными теоремами, утверждающими тот же самый результат для конкретных технологий с разными типами операций. (Соответствующие технологии упоминаются в утверждениях 1.4.2-1.4.8 в [13].) Помимо технологий с операциями синтеза такие же по сути теоремы были доказаны (включая факт бесконечной креативности) для технологий со следующими операциями:

Операция расширенного примитивного синтеза. Операцией расширенного примитивного синтеза F_{ES} называется операция типа F_S с параметром:

$$F_{ES}(x, y, \alpha) \rightarrow x * y,$$

где $x \in \mathbf{A} \cup \mathbf{B}$, $y \in \mathbf{A}$; $\alpha \in [0, n]$ (т. е. α – любое целое число от 0 до n). Параметр α может означать всё что угодно: координату установки детали y на узле x , температуру проведения реакции между веществами x и y и т.д.

Операция «фиксированной дезинтеграции». Операцией «фиксированной дезинтеграции» называется операция типа $F_k(x) \rightarrow kx$, где $0 < k < 1$, $k = \text{const}$ (то есть значение k – постоянное и меньше единицы, но больше нуля), позволяющая получать из элементов $x \in \mathbf{A}$ новые элементы kx , являющиеся «фиксированной частью x ».

Фактически операция типа $F_k(x)$ выдает какую-то фиксированную меньшую часть исходного объекта x (например, некоторый фрагмент объекта x «по объёму», «по весу» или «по амплитуде»). Примерами операций типа F_k с некоторой степенью идеализации могут быть:

- 1) операция «масштабирования» $F_k(x) = -kx$ ($0 < k < 1$) при обработке уровней электрических сигналов, представленных величиной x ;
- 2) операция растворения фиксированной части металлического электрода при электролизе;
- 3) операция отражения фиксированной части светового потока полупрозрачным зеркалом;
- 4) операция генерирования («отражения») части входного тока «токовым зеркалом» в электронике (при коэффициенте отражения меньше единицы) и т.д.

Таким образом, для получения бесконечной последовательности операций синтеза, каждая из которых может быть восстановлена по ее номеру, а также для достижения бесконечной креативности технологий вовсе не обязательно иметь операции синтеза – в некоторых случаях аналогичную роль может сыграть операция «фиксированной дезинтеграции». Другое дело, что в реальных технологиях нашего мира получить абсолютно точное подобие такого рода операции практически невозможно – хотя бы по причине дискретной структуры вещества на атомно-молекулярном уровне.

Аналогичную (т.е. бесконечную и восстанавливаемую по номеру) последовательность синтезов можно построить и для разного рода «гибридных» технологий типа $T = \langle A, \{F_k, F_S\} \rangle$, содержащих как операции синтеза, так и операции «фиксированной дезинтеграции».

Название всех «теорем об эффективности накопленных знаний» имеет следующее объяснение. Согласно этим теоремам, все синтезы в технологиях могут быть последовательно пронумерованы так, что по номеру всегда может быть восстановлен соответствующий синтез. Очевидно, в ходе реализации таких пронумерованных синтезов можно пронумеровать и получаемые в них конструкции, фактически присвоить конструкциям некие условные «имена». Следовательно, можно составить (и запомнить) своего рода таблицу, в каждой строчке которой записаны «имя»-номер синтеза и «имя»-номер соответствующей ему конструкции. Используя такую таблицу, в дальнейшем легко по «имени»-номеру конструкции отыскать соответствующий синтез и получить ещё одну такую же конструкцию. То есть хранящаяся в таблице информация действительно позволяет эффективно использовать знание о выполненных ранее синтезах для получения нужных конструкций в нужных количествах. Более того, если в технологии есть средства (операции и механизмы), позволяющие определить одинаковость или неодинаковость различных конструкций, то такого рода таблицу легко отсортировать, выяснив, какие синтезы приводят к одним и тем же конструкциям, а какие – нет. Таким образом, становится возможным получение уже каких-то первичных знаний об особенностях той или иной технологии.

С другой стороны, все «теоремы об эффективности накопленных знаний» фактически говорят о том, что, хотя некоторые множества используемых в них синтезов могут быть достаточно запутанными (как, например, в технологии $T = \langle A, \{F_k, F_S\} \rangle$ – см. [13]), они всё же имеют в своей основе простую структуру, позволяющую отображать их на множество натуральных чисел и разрабатывать соответствующие алгоритмы для их восстановления по этим числам-«именам». Данный результат имеет простое объяснение: любая технология, бу-

лучи по определению конечной алгоритмической системой (т.е. системой с конечным числом типов начальных объектов операций и конечным числом типов самих операций над объектами), за конечное число шагов может породить лишь конечное множество объектов (синтезов или конструкций).

Полученные в «теоремах об эффективности накопленных знаний» результаты свидетельствуют также и о том, что процедуры генерирования последовательностей синтезов (с дальнейшим их восстановлением – при необходимости – по именам-номерам) могут быть реализованы *с помощью подходящего автоматически действующего устройства*. Конструкция такого устройства, получившего название универсального синтезатора-анализатора, подробно будет обсуждаться в главе 2.

Следующим очевидным шагом в исследовании самых общих свойств различных формально-технологических систем должны быть технологии с операциями, отличными от операций примитивного синтеза и фиксированной дезинтеграции. При этом – в продолжение одной из основных тем параграфа 1.2 – серьёзное внимание должно быть уделено изучению возможностей разных технологий при решении задач самого общего вида, и в первую очередь – в плане воспроизведения в таких технологиях различных конструкций по их образцам (возможно, единственным). Почему такие технологии представляют особый интерес? Ответ естественен: если та или иная конструкция может быть воспроизведена с использованием операций данной технологии, то, во-первых, сам этот факт говорит о том, что эта конструкция может быть отнесена к конструкциям данной технологии. Во-вторых, в такой технологии в принципе возможна расшифровка «устройства» любых её конструкций, даже если мы сталкиваемся с ними впервые. Для всех подобных «полноценных» технологий можно ввести формальное определение:

Определение полной технологии. Технология $T = \langle A, F \rangle$ называется полной, если она, во-первых, бесконечно-креативна, и, во-вторых, позволяет по любой конструкции $y \in A \cup B$ восстановить её структуру и синтез.

Вводимое определение полноты технологии не связывается напрямую с существующим в математике понятием функциональной полноты, в частности с полнотой, например, систем булевских или частично-рекурсивных функций, но тем не менее оказывается достаточно близко к ней [15, 33].

Какими же особенностями должна обладать технология, чтобы быть «полноценной», т.е. полной? Ответ на самом деле очень и очень непрост. Оказалось, что в отличие от математики, базирующейся в основном на полной технологии, называемой системой частично-рекурсивных функций, *в рамках ОФТ существует несколько различных типов полных технологий.*

Исходя из определения полноты технологий можно сделать предположение, что в них, вероятно, должны быть, как минимум, хоть какие-то операции анализа. Например, позволяющие установить эквивалентность (одинаковость) двух конструкций – вновь полученной и исходной. Действительно, одной из таких (полных) технологий оказалась достаточно простая бесконечно-креативная технология (т.е. отвечающая первому условию полноты) всего с двумя операциями: операцией (предикатом) определения равенства двух объектов P_E (операция анализа) и операцией примитивного синтеза F_S , т.е. технология $T = \langle A, \{F_S, P_E\} \rangle$, где $A = \{a_1, a_2, \dots, a_n\}$. Объяснение полноты этой технологии достаточно простое: согласно «теореме об эффективности накопленного знания» в ней можно генерировать пространство синтезов и получать соответствующие конструкции, которые с помощью предиката равенства сравниваются с «образцом». Если сравнение достигнуто, искомый синтез образцовой конструкции найден (т.е. и второе условие полноты технологии, введенное соответствующим определением, также выполнено).

Технологии типа $T = \langle A, \{F_S, P_E\} \rangle$ весьма близки некоторыми своими особенностями к основной технологии вычислений в математике – к уже упоминавшейся технологии «частично-рекурсивных» функций. Эта близость определяется, в первую очередь, наличием и в том, и в другом случае предикатов равенства. Математики предпола-

гают, что к композиции «частично-рекурсивных» функций сводятся все функции, которые придумал и реально может вычислить человек.

Что же касается роли предиката равенства в ОФТ, то следующая теорема демонстрирует всю его мощь:

Утверждение 1.5. Все технологии с конечным числом различных элементов в базе, с конечным числом конечноместных операций синтеза (типа F_S и/или F_I) и с операцией анализа типа предиката равенства P_E являются полными технологиями.

Естественен вопрос: а могут ли в ОФТ существовать полные технологии, существенно отличающиеся от технологий с предикатом равенства? Например – вообще не иметь этого предиката? Оказалось – могут. Но для этого в таких технологиях необходимо расширить наборы технологических и (или) аналитических операций. В частности, за счёт введения операции декомпозиции (т.е. «разборки» объекта), «обратной» (или «противоположной») по отношению к операции примитивного синтеза:

Определение операции примитивной декомпозиции. Операцией примитивной декомпозиции называется операция $F_d(x) \rightarrow \langle y, a \rangle$, позволяющая «отсоединить» элемент $a \in A$ от конструкции x , если x – результат хотя бы однократного применения операции примитивного синтеза F_S к элементам из множества $A \cup B$, то есть:

$$F_d(x) \rightarrow \langle y, a \rangle, \text{ если } x \text{ получен в результате } F_S(y, a) \rightarrow x.$$

Примеры операций примитивной декомпозиции:

- 1) операция изъятия ровно одной конкретной детали из собранного узла;
- 2) операция уменьшения на единицу исходного целого числа;
- 3) операция отделения последней буквы в слове;
- 4) операция вырезания фрагмента металла на металлорежущем станке за один шаг подачи режущего инструмента.

Последняя (четвёртая) операция, конечно, не в полной мере отвечает определению примитивной декомпозиции, поскольку предполагает наличие несуществующего в традиционной технологии металло-

обработки обратного процесса «примитивного синтеза», то есть «добавления» фрагмента металла (или какого-либо другого материала) к синтезируемой детали за один шаг подачи инструмента на «синтезирующем» станке. Однако и теоретически, и практически подобные операции и устройства синтеза возможны и даже необходимы. Более того, в некоторых типах таких синтезирующих устройств – в так называемых «фабберах» – подобные или же очень близкие операции применяются для синтеза самых различных объектов.

Использование операции примитивной декомпозиции наряду с операцией примитивного синтеза позволяет достичь в некоторых случаях полноты технологии с добавлением всего одной и очень простой операции анализа – операции P_0 , определяющей наличие или отсутствие объекта. Примером может быть технология $T = \langle \{a\}, \{F_S, F_D, P_0\} \rangle$, в которой база A содержит элементы только одного типа: $A = \{a\}$. Эта же технология интересна и в другом плане, а именно – при некоторых дополнительных требованиях к операции F_S она может быть «линейной технологией».

Линейные технологии играют важную роль во многих биохимических процессах, поскольку именно в таких технологиях в первую очередь может возникнуть спонтанный эволюционный процесс роста структурной и функциональной сложности конструкций, при определённых условиях приводящий к появлению феноменов самоорганизации и саморазмножения.

Имеется несколько вариантов определения линейных технологий. Приведём один из них, взятый из работы [65]:

Определение линейной технологии. Если любая новая конструкция u получается из предшествующей конструкции x путём присоединения к *последнему* элементу $a_j \in A$ конструкции x некоторого элемента базы $a_i \in A$ с помощью операции синтеза типа: $F(x, a_i) \rightarrow u$, то такая конструкция называется линейной, а элемент a_i становится *последним* элементом конструкции вместо элемента a_j , причём в одноэлементных конструкциях элемент, её составляющий, одновременно

считается и *последним*. Если все конструкции в технологии – линейные, то технология называется линейной.

Таким образом, если в $T = \langle \{a\}, \{F_s, F_d, P_0\} \rangle$ операция синтеза F_s отвечает критерию линейности, то технология T , будучи полной, может рассматриваться как одна из простейших для разработки модели эволюционной технологии [14].

Более сложные варианты операций декомпозиции тоже, очевидно, способны привести к полным технологиям. Однако введение, например, операции декомпозиции, обратной операции синтеза типа F_t (параграф 1.2), затруднено тем обстоятельством, что в синтезированной с помощью F_t конструкции z сложно сохранить информацию о том, какие конкретно её части были соединены между собой в последней операции синтеза F_t .

Для разрешения неопределённостей такого рода существует несколько путей. Например, можно ввести «параметризованную» операцию декомпозиции, в которой конкретную разрываемую связь указывает числовой параметр. А можно ввести и «случайную» декомпозицию, в которой разрываемые связи определяются случайным образом. Ввиду относительной сложности реализации «параметризованной» (управляемой) декомпозиции рассмотрим в первую очередь «случайную»:

Определение «случайной» декомпозиции. Операцией «случайной» декомпозиции F_b называется операция, разделяющая исходную конструкцию y случайным образом на фрагменты, число которых – также случайная (но обязательно конечная) величина:

$$\begin{aligned} F_b(y) &\rightarrow \langle x_1, x_2, \dots, x_n \rangle, \text{ если } y \in \mathbf{B}, \text{ и} \\ F_b(y) &\rightarrow \langle y \rangle, \text{ если } y \in \mathbf{A}, \end{aligned}$$

где n – число фрагментов исходной конструкции y , которое не может быть больше, чем число элементов базы в конструкции y .

Как следует из определения, операция «случайной» декомпозиции «разбивает» исходную конструкцию на более мелкие фрагменты, но не меньшие, чем элементы базы.

Примеры операции «случайной» декомпозиции:

- 1) «разборка» здания путём его разрушения;
- 2) «разборка» хрупких изделий путём их «разбивания молотком»;
- 3) «разборка» молекул путём «бомбардировки» их высокоэнергетическими частицами.

Могут ли технологии с операцией случайной декомпозиции быть полными? Оказывается, могут, и такой технологией является, например, технология $T = \langle A, \{F_b, F_e, P_e\} \rangle$ ($A = \{a_1, a_2, \dots, a_n\}$). Но, как нетрудно заметить, для обеспечения полноты в ней снова пришлось воспользоваться предикатом равенства P_e . А достижима ли полнота технологий с операциями типа F_b без предиката равенства? Оказывается, достижима! Правда, для этого кроме простой операции определения наличия или отсутствия объекта типа P_0 потребуется ещё как минимум одна операция анализа, обладающая весьма интересными особенностями. Называется эта операция «случайным стационарным отображением». Для того чтобы ввести её формальное определение, сначала нужно формально же определить и «стационарность» операций:

Определение «стационарной» операции. Операция анализа P_A называется «стационарной» (или «устойчивой») в данной технологии, если её результат для любой конструкции остаётся постоянным при анализе этой конструкции любого числа раз.

Строго говоря, особой нужды в этом определении нет, поскольку с самого начала данное требование предполагалось выполнимым (по умолчанию) для любых операций из F , а не только для операций анализа. Тем не менее оно поможет разобраться в сути следующего определения, в котором используется понятие «случайного» отображения некоего свойства конструкций, так или иначе связанного с их структурой и составом, на числовую (или иную) ось:

Определение операции «случайного стационарного отображения». Операция анализа P_A называется операцией «случайного стационарного отображения» в данной технологии, если, во-первых, она стационарна, а во-вторых, её результат всегда находится в определённом (фиксированном) конечном диапазоне рациональных чи-

словых значений (или иных, в том числе нечисловых параметров), однако *при первом применении операции к любой новой конструкции* в данной технологии *он заранее неизвестен* (т.е. как бы «случаен» для внешнего наблюдателя).

В этом определении термины «случайное отображение» фактически означают, что нам на самом деле неизвестен внутренний механизм преобразования некоего свойства конструкций в тот параметр, который «измеряет» (отображает) операция анализа P_A . В свою очередь, термин «стационарное» говорит о том, что такой скрытый механизм на самом деле существует и что он исправно срабатывает всегда, когда для анализа предъявляются одинаковые конструкции. Именно в этом смысле результат анализа и является «стационарным», т.е. значение параметра постоянно для одних и тех же конструкций. «Случайной» же является величина измеренного параметра для любых новых, впервые анализируемых конструкций, так как – повторимся ещё раз – мы не знаем устройства самого механизма отображения данного свойства конструкции на числовую ось, и поэтому заранее предсказать результат такого измерения для новой, не анализировавшейся ранее конструкции мы не можем. Яркими примерами таких «случайных», но «стационарных» отображений на ранних этапах развития химии служили цвет вещества, его вкус и запах. Механизмы формирования «значений» этих параметров ни в самих веществах, ни в организме человека тогда ещё не были известны, однако «стационарность» процессов (при отсутствии дальтонизма, гриппа или других отклонений от нормы) была: соль оставалась солёной, пока была солью. Пурпур окрашивал ткани в пурпурный цвет, пока не превращался в другое вещество, и т.д.

Простейший пример анализа структуры конструкций с использованием операции случайного стационарного отображения – когда все элементы базы имеют абсолютно различные значения по измеряемому с помощью P_A параметру. В этом случае, используя операцию примитивной декомпозиции F_d , можно разобрать любую конструкцию на элементы и с помощью P_A определить тип каждого из них, ус-

тановив тем самым порядок сборки конкретных элементов базы в данную неизвестную раньше конструкцию. Таким образом, оказывается справедливым следующее утверждение:

Утверждение 1.6. Если в технологии $T = \langle A, \{F_S, F_d, P_0, P_A\} \rangle$ ($A = \{a_1, a_2, \dots, a_n\}$) все различные элементы базы имеют различные значения параметра β операции «случайного стационарного отображения» $P_A(a_i) \rightarrow \beta$, то технология T – полная. (Доказательство следует из приводимого ниже утверждения 1.7.)

К сожалению, вследствие декларированной выше «случайности» стационарного отображения не исключено, что различные типы элементов базы будут иметь одинаковые значения по параметру P_A , особенно если число типов элементов в базе достаточно большое, а используемое в операции P_A числовое множество для представления измеряемого параметра невелико (например, содержит всего два значения: 0 и 1). Тем не менее, если скрытый от нашего понимания механизм, делающий отображение соответствующего параметра различных новых конструкций на числовую ось как бы «случайным», продолжает работать, т.е. условия «стационарности» и «случайности» выполняются для всех конструкций, то полнота технологии по-прежнему достижима. Однако при этом одновременно появляется и вероятность того, что текущая (т.е. полученная на данный момент) модель предположительно полной технологии всё же на самом деле не совсем корректна. Впрочем, по мере увеличения числа дополнительных экспериментов в ней (по синтезу и анализу новых объектов) она становится всё более правильной. Эти особенности формально декларируются следующей теоремой ОФТ:

Утверждение 1.7. Если в технологии $T = \langle A, \{F_S, F_d, P_0, P_A\} \rangle$ ($A = \{a_1, a_2, \dots, a_n\}$) операция $P_A(x) \rightarrow \beta$ «случайного стационарного отображения» выполняется для всех конструкций, то по мере реализации всё большего числа новых синтезов и анализов новых конструкций в T её полнота достигается с вероятностью, сколь угодно близкой к единице.

В утверждении 1.7 полнота технологии достигается не только благодаря двум операциям анализа (P_0 и P_A), отличным от предиката

равенства, но и благодаря весьма мощной и точной операции примитивной декомпозиции F_d , позволяющей аккуратно, элемент за элементом, разбирать любые конструкции. Тем не менее ситуация не изменится, если эту операцию заменить на операцию «случайной декомпозиции». Но – при некоторых дополнительных условиях.

Одно из них связано с возможностью получения любого числа так называемых «неизвестных» конструкций, необходимых для экспериментов по поиску их синтезов в заданной технологии. Чтобы сохранить достигнутый на текущий момент уровень формализма, введём определение «неизвестной», а заодно и «известной» конструкций:

Определение «известной» и «неизвестной» конструкций. Конструкция R в технологии T называется известной, если синтез R известен: $C_R \rightarrow R$. Наоборот, если синтез R неизвестен, то конструкция R называется неизвестной.

Теперь можно сформулировать следующее утверждение:

Утверждение 1.8. Если в технологии $T = \langle A, \{F_t, F_b, P_0, P_A\} \rangle$ имеются источники неизвестных конструкций y_j , а случайная стационарная операция анализа $P_A(x) \rightarrow \beta$ выполняется для всех конструкций и имеет примерно одинаковые частоты появления различных значений параметра β на различных конструкциях, то полнота T достижима с вероятностью, сколь угодно близкой к единице.

Утверждение 1.8, таким образом, показывает, что для достижения полноты технологии действительно не обязательно иметь не только мощный аналитический аппарат в виде операций сравнения двух объектов, но и мощный и точный механизм разборки конструкций в виде операции «примитивной декомпозиции» (фактически – «поэлементного отщепления») F_d . Вместо него вполне достаточно операции «случайной» (т. е. «грубой» и «непредсказуемой») декомпозиции F_b . Тем не менее механизмы синтеза новых конструкций (в виде операции F_t) остаются весьма точными – вплоть до реализации операции присоединения одного-единственного элемента базы a_i к любой конструкции x с помощью $F_t(x, a_i)$.

Нетрудно также заметить, что операции случайного стационарного отображения P_A фактически соответствует любая аналитическая операция измерения некоторого параметра (свойства) объекта. Причём операция остаётся с формально-технологической точки зрения эффективной даже тогда, когда мы не знаем или не понимаем физических особенностей формирования измеряемого параметра. Поэтому удачный выбор свойства P_A , которое должно давать различные, но преимущественно равновероятные значения измеряемого параметра на различных конструкциях, – весьма сложная проблема. Есть, однако, очень простой критерий для выбора такого свойства, даже если мы ничего не знаем о его механизме, а именно: его главной (и чрезвычайно важной для нас) характеристикой как раз и является его *непостоянство*, его «вариабельность», его частая изменчивость при переходе от одной конструкции к другой. То есть наиболее интересно и наиболее информативно то, *что чаще всего и сильнее всего меняется* при переходе от одного объекта к другому.

Новая формулировка утверждения 1.8, изменённая с учётом указанного обстоятельства, выглядит так:

Утверждение 1.8(а). Если в технологии $T = \langle A, \{F_s, F_d, P_0, P_A\} \rangle$ случайная стационарная операция анализа $P_A(x) \rightarrow \beta$ выполняется для всех конструкций и имеет примерно одинаковые частоты появления различных значений параметра β на различных конструкциях, то полнота технологии достигается с вероятностью, сколь угодно близкой к единице.

Утверждения 1.8, 1.8(а) дополнительно подтверждают тот факт, что для достижения полноты некоторых технологий совсем не обязательно иметь мощный и сложный аналитический аппарат, каким является операция анализа равенства двух объектов. Достаточно иметь надёжное («стационарное») средство отображения некоторого (пусть даже и непонятного!) свойства всего (счётно-бесконечного) множества конструкций на некоторое подмножество натуральных чисел (параметров), чтобы можно было определять структуру любой конструкции. Природа блестяще демонстрирует справедливость этих тео-

рем на примере эволюции вида *homo sapiens*, особи которого, обладая лишь зрением, слухом, обонянием и осязанием, смогли познать структуру атомного ядра, которое они не видели, не слышали, не осязали и не обоняли. Аналогичная ситуация наблюдалась и в ходе развития химии: атомно-молекулярное учение появилось не тогда, когда удалось «увидеть» и «сравнить» между собой отдельные атомы различных веществ с помощью электронного, туннельного или «силового» микроскопа, а гораздо раньше, когда о такой возможности никто даже не помышлял. Разумеется, определённую роль в получении необходимой информации о структуре материи играли и играют различные приборы, но, в конце концов, и они выдают нам информацию в том виде, в котором её способны воспринимать всё те же наши органы чувств. Фактически можно считать, что любые приборы на самом деле просто преобразуют определённые свойства исследуемых объектов с помощью соответствующих операций «стационарного отображения» в привычные нам ощущения, а уже последние – с помощью наших органов чувств и соответствующих операций «случайного стационарного отображения» – в нервные импульсы в нашем мозге.

В каком-то смысле аналогом операции случайного стационарного отображения может служить другой тип аналитических операций – так называемые «классифицирующие» операции:

Определение классифицирующей операции. Операция анализа $P_C(x) \rightarrow \beta$ для группы элементов (конструкций) называется «классифицирующей», если она даёт одинаковые значения своего выходного параметра β для всех конструкций данной группы и другие значения β – для всех других конструкций.

Если у нас имеется возможность выбирать различные классифицирующие аналитические операции, способные различать отдельные группы элементов базы, то оказывается полезным следующее определение:

Определение рационального разбиения классифицирующих операций. *Рациональным* разбиением классифицирующих операций

называется такое их разбиение, которое позволяет для любого из n элементов базы определить его не более чем за $\lceil \log_2 n \rceil$ шагов с помощью соответствующего дихотомического алгоритма (здесь запись $\lceil \dots \rceil$ означает ближайшее большее целое к содержимому, обозначенному многоточием «...» внутри развернутых квадратных скобок] [).

Напомним, что дихотомический алгоритм на первом шаге определяет принадлежность элемента к одной из двух половинок конечного множества из n элементов, на втором – к одной из двух его четвертинок (или к одной из половинок уже выбранной половинки), на третьем – к одной из двух его $1/8$ частей, и т.д., пока на последнем шаге не определяется принадлежность элемента к одному из двух возможных элементов множества – то есть к одной из двух его $(1/n)$ -ных частей.

Таким образом, при рациональном разбиении каждая из классифицирующих аналитических операций должна разделять всё множество элементов базы на две (желательно) одинаковые группы по $n/2$ элементов базы в каждой, причём для различных классифицирующих операций эти группы не должны полностью совпадать. В то же время принадлежность различных элементов базы к той или иной из двух классифицируемых с помощью данной операции групп может быть как бы случайной, необъяснимой (пока!) для нас, как и значение параметра в операции случайного стационарного отображения. Число самих классифицирующих операций при этом также оказывается равным $\lceil \log_2 n \rceil$ и на каждом очередном шаге алгоритма используется только одна классифицирующая операция. (На самом деле возможно огромное количество других вариантов «не самого рационального» разбиения классифицирующих операций, и, соответственно, такое же число не дихотомических, но конечных алгоритмов определения точного типа элементов базы при таких разбиениях.)

Ввиду рассмотренных выше свойств дихотомического алгоритма и выбранного нами критерия рационального разбиения классифицирующих операций нетрудно доказать следующее:

Утверждение 1.9. Технологии с рациональным разбиением аналитических классифицирующих операций и содержащие, кроме них, только операции F_S и F_d , являются полными технологиями.

Рассмотрим теперь технологии с операциями, которые можно называть операциями «поэлементного» анализа:

Определение операций поэлементного анализа. Для технологии с конечной базой $A = \{a_1, a_2, \dots, a_n\}$ операциями поэлементного анализа называются операции вида $P_i(a_j) \rightarrow \beta$, причём $\beta = 1$, если $i=j$, и $\beta = 0$ – если $i \neq j$ или конструкция a_j отсутствует (т.е. $a_j = \square$). Для случая $P_i(x)$, $x \notin A$, результат операции не определён (можно считать его равным 0).

Утверждение 1.10. Технологии $T = \langle A, \{F_s, F_d, P_1, P_2, \dots, P_n\} \rangle$, в которых для каждого элемента базы a_i есть соответствующая операция поэлементного анализа P_i , являются полными.

(Доказательство см. в [13] для аналогичного утв. 1.5.2.)

Из сравнения утверждения 1.5 с утверждениями 1.6 – 1.10 следует, что предикат равенства P_E , т.е. операция анализа, определяющая идентичность любых двух объектов (элементов и/или конструкций) технологии, позволяет избавиться не только от множества операций поэлементного анализа, но и от операций декомпозиции. Другой вопрос – как обеспечить практическую реализацию операции P_E на всём множестве конструкций, в том числе на конструкциях, насчитывающих, например, миллионы и миллиарды элементов. Представить какие-либо реальные механизмы здесь сложно. И если математика совершенно свободно использует эту операцию, то в формальной технологии, имея в виду прежде всего её практическую направленность, обращаться с ней следует чрезвычайно осторожно. Необходимо обратить внимание и на то, что в реальных «вычислительных» технологиях (т.е. технологиях обработки информации) определение равенства двух закодированных объектов аналогичных «размеров» (миллионы и миллиарды бит) даже при простейших способах кодирования – тоже не самая простая операция. Тем не менее, как мы уже видели, существуют аналоги операции P_E , имеющие более простые и более удобные механизмы своей реализации, пригодные для работы в том числе и с большими конструкциями. Рассмотрим другие интересные варианты.

Утверждение 1.11. Технологии $T = \langle A, \{F_s, F_d, P_0, P_u\} \rangle$, где $A = \{a_1, a_2, \dots, a_n\}$, $P_u(x) = P_E(x, u)$, а u – известная конструкция, содержащая все элементы базы A (кроме, возможно, одного), являются полными технологиями.

Конструкция u обычно называется представительной [13].

Согласно утверждению 1.11 получается, что если вновь ввести в состав операций технологии операцию примитивной декомпозиции F_d , то можно ограничить требуемое для достижения её полноты множество операций анализа всего двумя операциями: очень простой операцией анализа на отсутствие конструкций (то есть на «пустую» и «не пусто») типа P_0 и несколько более сложной операцией «сравнения представительной конструкции» u с любой конструкцией y в данной технологии. Последняя операция, кстати говоря, может быть легко сведена к операции сравнения форм представительной конструкции u и анализируемой конструкции y . Для этого необходимо лишь выполнить ряд условий, гарантирующих «разность» форм конструкций при различии их состава и структуры. Не углубляясь в детали этой процедуры, отметим, что предпосылками для таких гарантий являются, во-первых, по крайней мере, двухмерность пространства, в котором действует технология [14, параграфы 1.2, 3.1], и, во-вторых, различие размеров (а может быть, и форм) всех разнотипных элементов базы A . Формулировка соответствующей теоремы (утверждения) может выглядеть следующим образом:

Утверждение 1.11(b). Технологии $T = \langle A, \{F_s, F_d, P_0, P_u\} \rangle$, где $A = \{a_1, a_2, \dots, a_n\}$, $P_u(x)$ – операция сравнения формы произвольной конструкции x с уникальной формой известной конструкции u , содержащей все элементы базы A (кроме, возможно, одного), являются полными технологиями.

Отметим также, что конструкция, представляющая все (за исключением, может быть, одного) элементы базы, в предельном (идеальном) случае может быть не очень большой и содержать не более чем $(n-1)$ элементов базы для $A = \{a_1, a_2, \dots, a_n\}$. Более того, совершенно не обязательно иметь только одну представительную конструкцию

и. Их может быть несколько, причём в этом случае «размеры» конструкций могут быть ещё меньше, поскольку каждая из них теперь не обязана представлять все элементы базы – достаточно, чтобы это выполнялось лишь для всей совокупности представительных конструкций в целом:

Утверждение 1.12. Технологии $T = \langle A, \{F_s, F_d, P_0, P_{R1}, P_{R2}, \dots, P_{Rt}\} \rangle$, где $A = \{a_1, a_2, \dots, a_n\}$, а $P_{Ri} = P_{Ri}(x) = P_E(x, Ri)$, причём $R1, R2, \dots, Rt$ – известные конструкции в T , содержащие в совокупности все элементы базы A (за исключением, быть может, одного), являются полными технологиями.

Таким образом, если операции «поэлементного» анализа по каким-то причинам реализовать затруднительно, то можно свести их к набору операций типа P_{Ri} , причём выбрать те из них, которые имеют наиболее простую техническую реализацию. Отметим, что по этой же причине операции поэлементного анализа можно рассматривать как частный случай утверждения 1.12, когда роль очередной представительной конструкции выполняет в точности один элемент базы.

В комментариях к утверждению 1.8 уже говорилось о том, что, пожалуй, самое важное требование к операциям типа «случайного стационарного отображения» – это «вариабельность» их результатов, т.е. частая «сменяемость» значений (параметров) результатов анализа на различных объектах. Какое же наиболее общее (наиболее массовое) свойство физических объектов ближайших к нам (с точки зрения нашего «чувственного» восприятия) микро- и макромиров отвечает этим требованиям? Ответ достаточно очевиден: это – форма объектов. Действительно, форма бактерий, растений, животных, ландшафта, камней, зданий, созданных нами различных машин, устройств, механизмов и их частей, формы макро- и микромолекул несут на себе как отпечаток истории их создания и развития (их эволюции), так и особенности их внутренней структуры, их «устройства». Причём это свойство характерно в первую очередь именно для двух-, трёх- и n -мерных ($n > 3$) объектов. В одномерных мирах аналог «формы» конструкции не имеет такого большого значения (см., например, [14], параграф 1.2).

По этим причинам для n -мерных ($n > 1$) миров очень полезными аналогами рассмотренных выше полных технологий могут быть такие, в которых основным анализируемым «параметром» была бы как раз форма объектов. Оказывается, что такие аналоги существуют – если и не для всех, то для достаточно большого числа технологий [13].

В некоторых технологиях ключевую роль для достижения их полноты могут играть не только формы объектов, но и особенности связей между составляющими их элементами базы, приводящие к различиям в форме конструкций [13-16].

Для сравнительного анализа возможностей полной технологии T_{BM} вычислительной математики, представленной системой частично-рекурсивных функций (ЧРФ), и других различных полных технологий ОФТ типа $T = \langle A, F \rangle$, полезно записать технологию T_{BM} в каноническом виде как технологию $T_{ЧРФ}$ с использованием традиционных обозначений системы частично-рекурсивных функций. Один из вариантов такой записи приведён в [15]:

$$T_{BM} = T_{ЧРФ} = \langle \{0, 1\}, \{s(x), \mu_y(f(x_1, \dots, x_{n-1}, y) = x_n)\} \rangle, \quad (1.20)$$

где $A = \{0, 1\}$, $s(x)$ – функция следования, $\mu_y(f(x_1, \dots, x_{n-1}, y) = x_n)$ – функция минимизации [15, 26].

Запись системы ЧРФ в классической математической форме содержит также ряд дополнительных функций и операций: $o(x)$, I_j^n , S^i , R , которые в записи формальных технологических систем типа $T = \langle A, F \rangle$ опущены (для удобства), хотя и предполагаются. Коротко суть всех этих функций и операций заключается в следующем: функция константы нуля $o(x) = 0$ всегда выдаёт нулевое значение (пустой объект) при любом значении аргумента x . Функция выбора I_j^n выдаёт объект с номером j из набора из n объектов. Операция суперпозиции S^i вычисляет какую-либо функцию от i аргументов ($i \geq 1$), являющихся результатами вычисления других функций, а оператор примитивной рекурсии R вычисляет очередное значение какой-либо функции (от очередного значения аргумента) на основе её предыдущего значения и предыдущего же значения аргумента. В ОФТ предполагается,

что все такого рода функции и операции могут быть выполнены либо неким «внешним» по отношению к технологии субъектом (например, человеком), либо автоматически действующим устройством типа «универсального синтезатора-анализатора объектов», который рассматривается в параграфе 2.1.

Согласно записи (1.20), множество элементов базы **A** в технологии **T_{чрф}** содержит всего два *нефункциональных* элемента: пустой (0) и непустой (1). Смысл оставшихся «технологических» операций следующий:

Функция следования $y = s(x) = x + 1 = s(x, 1)$ добавляет, согласно своему определению, один непустой элемент базы (единицу) к аргументу-объекту x (т.е. к «конструкции» x). В этом плане функцию следования можно считать «математическим» аналогом операции примитивного синтеза $y = F_S(x, a) = x * a$ в технологии **T** с двумя элементами в базе – пустым и непустым: $A = \{\square, a\}$. Или наоборот – операцию примитивного синтеза $F_S(x, a)$ в соответствующей **T** можно считать формально-технологическим аналогом функции следования в «технологии» **T_{чрф}**.

Операция минимизации $\mu_y(f(x_1, \dots, x_{n-1}, y) = x_n)$, в свою очередь, ищет такое минимальное значение y , при котором $f(x_1, \dots, x_{n-1}, y) = x_n$. Фактически для этого генерируются последовательные значения y , равные 0, 1, 2, ... n (n – целое число), и каждый раз вычисляется функция $f(x_1, \dots, x_{n-1}, y)$, пока её результат не будет равен x_n . Прямых аналогов операции минимизации в перечисленных в табл. 1.1 в четырнадцати (!) вариантах полных технологий нет [13, 15], однако соответствующие процессы (т.е. аналоги функции μ_y) широко используются во всех полных технологиях для доказательства самых различных теорем. Более того, эти аналоги на самом деле строятся из набора не задаваемых в явном виде (но предполагаемых) ОФТ-функций типа $o(x)$, I_j^n , S^i , R , а также из основных технологических операций технологии (типа F_S , F_t , F_b и т.д.), из предиката равенства P_E или его эквивалентов. То есть ключевым компонентом в операции минимизации с

точки зрения принятых в ОФТ условностей является как раз предикат равенства P_E . Поэтому самая короткая запись «технологии» $T_{\text{чрф}}$ в формально-технологическом варианте выглядит так: $T_{\text{чрф}} = \langle \{0, 1\}, \{s(x), P_E\} \rangle$. Из неё чётко видно, что $T_{\text{чрф}}$ действительно близка к технологии $T = \langle \{\square, a\}, \{F_S, P_E\} \rangle$. (Заметим попутно, что на основе $T_{\text{чрф}}$ функционируют все современные цифровые компьютеры.)

А теперь – очевидный и закономерный вопрос: как соотносятся между собой полные и эволюционные технологии? Наиболее вероятный ответ на него даёт следующее утверждение, пока не имеющее чёткого доказательства в силу своей специфической формулировки, а потому приведённое в виде гипотезы:

Гипотеза о соотношении между эволюционными и полными технологиями. Все эволюционные технологии – полные технологии. Обратное неверно.

Многие соображения говорят в пользу справедливости этой гипотезы. Во-первых, то, что и те, и другие технологии – бесконечно креативные (по определению). Во-вторых, возможные алгоритмы проверки конструкций на новизну очень близки по своей структуре и принципам работы к алгоритмам, восстанавливающим синтез неизвестных конструкций в полных технологиях. В-третьих, элементарно доказывается, что не все полные технологии могут быть эволюционными (в частности, технология частично-рекурсивных функций вычислительной математики $T_{\text{чрф}}$ полна, но не эволюционна – вследствие отсутствия каких-либо функциональностей у чисел). По этим причинам мы примем сформулированную выше гипотезу как истинную. С учётом данного обстоятельства на рис. 1.1 изображена диаграмма, иллюстрирующая взаимные соотношения между различными классами (типами) технологий.

Согласно этой диаграмме, эволюционные технологии оказываются, естественно, самыми сложными (и, соответственно, самыми «мощными») из всех. Именно поэтому они находятся как бы внутри всех остальных типов технологий. Далее по нисходящей идут полные технологии, затем – бесконечно-креативные и, наконец, просто креативные технологии (дающие новые объекты конечное число раз).

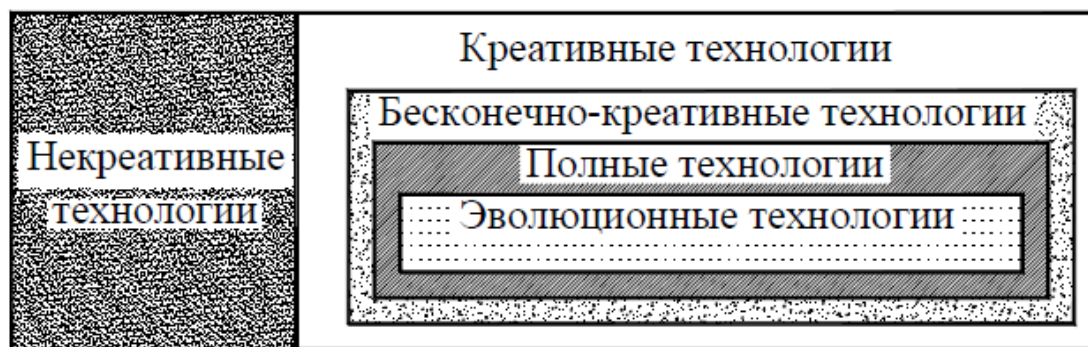


Рис. 1.1. Взаимное положение различных типов технологий

Самые же неинтересные – некреативные технологии, в которых, напомним, нельзя создать ничего нового. Они образуют самостоятельную группу, не пересекающуюся ни с каким другим классом технологий из рассмотренных выше.

Подведём теперь краткие итоги:

Базовые концепции и определения ОФТ, введенные в параграфах 1.2 – 1.3, действительно позволяют проанализировать некоторые важные свойства и потенциальные возможности различных классов достаточно простых технологий, в первую очередь такие, как их частичная или бесконечная креативность, полнота и эволюционность. Благодаря этим концепциям и подходящим формальным определениям удалось установить наличие нескольких групп технологических и аналитических операций, гарантирующих достижение полноты технологии даже при отсутствии в ней столь важного для математики (точнее, для обслуживающей её вычислительной технологии $T_{ЧРФ}$) предиката (аналитической операции) равенства. Более того, некоторые из найденных групп операций, эквивалентных предикату равенства и гарантирующих полноту соответствующих технологий, допускают простую «техническую» (или технологическую) реализацию – в отличие от математики и предиката равенства. С точки зрения технологий окружающего нас реального мира, данное обстоятельство представляется достаточно ценным, поскольку гарантирует движение соответствующих технологических систем по эволюционной лестнице вверх на основе очевидного и универсального принципа: «от простого – к сложному».

Глава 2. УНИВЕРСАЛЬНЫЕ ПРОГРАММНО-УПРАВЛЯЕМЫЕ ТЕХНОЛОГИЧЕСКИЕ СИСТЕМЫ

Прообразами таких систем можно считать полностью автоматизированные гибкие производства (ГАП) и, как ни странно, компьютеры. Действительно, если иметь в виду различные технологии обработки информации, то компьютеры (и, соответственно, их математические модели, получившие название машин Тьюринга – м.Т.) – идеальное воплощение предельно-многофункциональных (универсальных) программно-управляемых автоматических устройств для решения любых задач в рамках этих технологий. Более того, математики доказали, что более универсальных устройств обработки информации, чем м.Т. (то есть компьютеров) – по крайней мере на сегодняшний день – не существует. Есть, правда, некоторые надежды на так называемые «квантовые» компьютеры. Но они – дело будущего.

Что касается ГАП, то здесь ничего похожего на компьютеры по своей универсальности, гибкости и компактности пока, к сожалению, нет. Может ли формальная технология помочь в анализе причин столь существенных диспропорций в структурной организации и возможностях систем, предназначенных для решения задач практически одного уровня технологической сложности и диапазона? Можно ли с её помощью разработать столь же эффективные, как и используемые в компьютерах, методы структурного построения универсальных программируемых автоматических систем для других технологических областей? Например, для технологии обработки аналоговых и цифровых сигналов в так называемых многофункциональных программируемых аналого-цифровых «системах на кристалле» (сокращённо – МПАЦ СНК)? Или для различных «химических» технологий? Оказывается – можно, и результаты разработок весьма интересны.

2.1. МОДЕЛЬ УНИВЕРСАЛЬНОГО СИНТЕЗАТОРА-АНАЛИЗАТОРА ОБЪЕКТОВ ДЛЯ РАЗЛИЧНЫХ ТЕХНОЛОГИЙ

Наиболее подходящим прототипом для создания такой модели является, естественно, структура машины Тьюринга, рассматриваемая с позиций ОФТ. Соответствующая схема, основанная на этой концеп-

ции и адаптированная к универсальным программно-управляемым системам для различных технологий, изображена на рис. 2.1.



Рис. 2.1. Структура модели универсального синтезатора-анализатора объектов на основе концепций машины Тьюринга

Схема содержит одномерный массив $a_0 - a_n$ ячеек хранения (ЯХ) для объектов множества **В** заданной технологии **Т**, вместе с регистром сдвига выполняющий функции аналога одномерной ленты с ячейками для записи/чтения символов и механизма сдвига этой ленты в м.Т. (рис. 2.2).

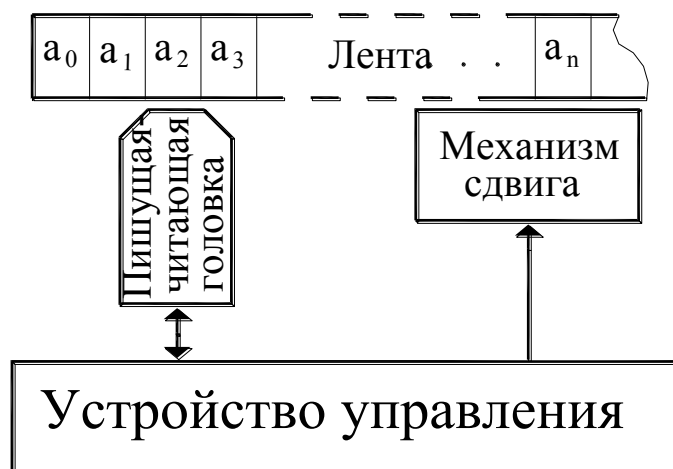


Рис. 2.2. Структура машин Тьюринга

Регистр сдвига вместе с устройством управления обеспечивают, во-первых, наращивание числа ячеек хранения массива однотипными блоками без изменения числа управляющих сигналов – наподобие подклейки, в случае необходимости, к ленте м.Т. нового её куска без изменения самого механизма сдвига; во-вторых, операции «извлечения из» и «размещения в» ячейках хранения a_i объектов операций из $\mathbf{B}$ в соответствии с положением «указателя» – единички в одном из разрядов регистра сдвига, во всех остальных разрядах которого записаны нули. Такая конструкция необходима для сохранения аналогии с м.Т. в плане возможности подвода к пишущей-читающей головке м.Т. любой ячейки ленты путем её сдвига влево или вправо на одну или несколько позиций. Наконец, вместо пишущей-читающей головки м.Т., выполняющей три типа «технологических» операций: печать нужного символа (из конечного множества), чтение символа (как правило, из того же множества) и стирание текущего символа (что можно свести к уже упоминавшейся операции печати – на этот раз пустого символа), в модели рис. 2.1 используется блок из соответствующего числа так называемых технологических ячеек (ТЯ) $F_1 - F_m$, выполняющих операции множества $\mathbf{F}$ технологии $\mathbf{T}$. Демultipлексор и мультиплексор на рис. 2.1 – устройства, обеспечивающие попадание нужного объекта из ЯХ в соответствии с положением «указателя» в нужную ТЯ и наоборот – попадание объекта-результата операции множества $\mathbf{F}$ в ту ЯХ, на которую в данный момент указывает «указатель». Механизмы извлечения и размещения объектов в ЯХ могут быть самыми различными, как и реализация самих ТЯ.

Нетрудно показать, что структура рис. 2.1 может выполнять в том числе и функции машины Тьюринга (рис. 2.2), если объекты операций интерпретировать, например, как карточки, хранящиеся в ЯХ $a_0 - a_n$, на которых могут печататься и стираться буквы. Действительно, функционирование любой м.Т. описывается таблицей, состоящей из конечного числа строк вида $q_j c_k v_{jk} q_{jk}$, имеющих следующий смысл: из состояния q_j , если над пишущей-читающей головкой в соответствующей ячейке ленты (в нашем примере – на карточке в соответствующей ЯХ) находится

символ $c_k \in C$, перейти в состояние q_{jk} и выполнить действие, предписываемое символом $v_{jk} \in C \cup \{c_0, R, L, s\}$, где C – рабочий алфавит машины Тьюринга; c_0 – пустая буква (пустой символ); R, L – сдвиг пишущей-читающей головки вдоль ленты (с помощью механизма сдвига) на одну ячейку вправо или влево соответственно; s – символ остановки м.Т. Если $v_{jk} \in C \cup \{c_0\}$, то соответствующий символ печатается в текущей ячейке (на карточке) ленты [26, 27].

Для выполнения операций чтения и печати символов в соответствующих ячейках аналога «ленты», в качестве которого используется одномерный массив ЯХ в модели на рис. 2.1, «указатель» регистра сдвига подводится к нужной ЯХ (выполняющей функции соответствующей ячейки ленты м.Т.), карточка извлекается и через демультиплексор помещается в ТЯ, которая выполняет «технологические операции», предписываемые символами c_k и $v_{jk} \in C \cup \{c_0\}$ из множества $\mathbf{F}$ технологии $\mathbf{T}_{\text{м.Т.}}$ ($|\mathbf{T}_{\text{м.Т.}}| = |\mathbf{T}_{\text{ВМ}}| = |\mathbf{T}_{\text{ЧРФ}}|$) конкретной (моделируемой) машины Тьюринга, т.е. выполняет операции чтения, печати или стирания соответствующего символа на карточке. После этого карточка возвращается на своё место в ячейке массива ЯХ, а устройство управления, в котором хранится алгоритм работы м.Т. (в виде таблицы строк типа $q_j c_k v_{jk} q_{jk}$), выполняет переход в состояние q_{jk} и действие из множества $\{R, L, s\}$, если соответствующее действие необходимо согласно смыслу v_{jk} .

Несмотря на простоту, этих операций оказывается вполне достаточно для решения любых задач обработки информации, процедура решения которых представима в алгоритмической форме. А поскольку, как считают многие математики, все реально решаемые человеком задачи на самом деле имеют соответствующие алгоритмы решения, то получается, что м.Т. и наша модель могут решать любые задачи, для которых решение существует хотя бы в принципе.

Кроме м.Т., имеется много других моделей универсальных вычислительных (алгоритмических) систем, таких, например, как системы Поста (с.П.) [26], машины Минского (м.М.) [28], машины Колмогорова (м.К.) и ряд других. Для всех этих моделей доказана как их

эквивалентность между собой, так и в отношении м.Т., то есть их «вычислительные мощности» одинаковы: $|T_{м.т.}| = |T_{вм}| = |T_{с.п.}| = |T_{м.м.}| = |T_{м.к.}|$. Поэтому при анализе «источков универсальности» программно-управляемых систем в области технологий обработки информации достаточно рассматривать только одну из них, в данном случае – машину Тьюринга.

Попробуем сформулировать в рамках формально-технологического подхода общие требования к структуре любых универсальных технологических систем, выполнение которых *гарантирует их универсальность*.

Прежде чем перейти к решению этой задачи, попытаемся сначала разобраться в самом смысле слов «универсальная технологическая система».

Естественно понимать универсальность системы в любой технологии $T = \langle B, F \rangle$ как возможность реализации в ней *любой* последовательности *любых* операций типа (1.18) из F над *любыми* разрешёнными для данных операций элементами и конструкциями из B , в том числе *с учётом всех возможных результатов операций анализа* из F , которые могут изменять дальнейшие последовательности действий. Такая точка зрения хорошо совпадает с попытками интерпретировать универсальность цифровых и аналоговых вычислительных машин в некоторых работах основоположников исследований в этой области – изобретателя первого программно-управляемого механического цифрового компьютера Чарльза Бэббеджа [45] и Клода Шеннона, доказавшего универсальность аналоговых компьютеров (аналоговых вычислительных машин) с определённым составом операций. Аппарат формальной технологии позволяет ввести соответствующее формально-технологическое определение:

Определение универсальной технологической системы. Система S_U в технологии $T = \langle A, F \rangle$ называется универсальной, если в S_U выполняема *любая* последовательность операций из F над *любыми* элементами из B , для которых соответствующие операции разрешены, причём последовательность операций может быть любым обра-

зом изменена по любым результатам любых имеющихся в $\mathbf{F}$ операций анализа.

Определение универсальности системы в таком виде представляется вполне обоснованным, поскольку позволяет реализовать все те возможности, которые предоставляются данной конкретной технологией, и не требует от неё ничего сверхъестественного. Если же на самих операции в $\mathbf{F}$ наложены какие-то ограничения, то в тех случаях, когда эти операции используются в реализуемых $\mathbf{S}_U$ алгоритмах, соответствующие ограничения, естественно, действуют как обычно.

Теперь введём следующее важное определение:

Определение рекурсивной схемы. Будем говорить, что в системе $\mathbf{S}$ для операции $F_i \in \mathbf{F}$ задана рекурсивная схема, если в $\mathbf{S}$ установлены такие связи, которые позволяют использовать элементы и конструкции, полученные при выполнении операции F_i на любом (конечном) предыдущем шаге $\mathbf{T}$, в последующих шагах $\mathbf{T}$.

С использованием этих двух определений легко доказать первую теорему о необходимых условиях универсальности формальных технологических систем:

Утверждение 2.1. Система $\mathbf{S}_U$ универсальна в $\mathbf{T} = \langle \mathbf{B}, \mathbf{F} \rangle$, если и только если:

- а)* технологические ячейки $\mathbf{S}_U$ реализуют любую операцию типа (1.18) из множества $\mathbf{F}$;
- б)* для каждой операции из $\mathbf{F}$ в $\mathbf{S}_U$ задана рекурсивная схема;
- в)* $\mathbf{S}_U$ содержит неограниченное число ячеек хранения.

Доказательство. Необходимость условий *а)* и *б)* очевидна. Если не выполнено условие *а)*, то это сразу ведёт к выходу из определения универсальной системы. Если же не выполнено условие *б)*, то такой выход возможен на любом шаге $\mathbf{T}$, на котором потребуются результаты соответствующей (предыдущей) операции, не имеющей рекурсивной схемы, и которые по этой причине использовать нельзя. Необходимость в условии *в)* связана с тем обстоятельством, что не накладывается никаких ограничений на количество исходных объектов и промежуточных результатов реализуемого в $\mathbf{S}_U$ процесса, а также на

число объектов (конструкций), считающихся итогом его реализации. Конец доказательства.

Легко проверяется, что модель системы на рис. 2.1 отвечает всем требованиям утв. 2.1. Однако сформулированные в утверждении 2.1 условия являются необходимыми, но отнюдь не достаточными для обеспечения её универсальности в любой технологии $\mathbf{T}$ в соответствии с определением. Для обеспечения полноты технологий в них, кроме алгоритмов синтеза, естественно, должны быть реализуемы и алгоритмы анализа, которые предполагают возможность *изменения порядка последующих действий в зависимости от результатов выполнения той или иной операции анализа*. Именно для этого устройство управления S_U должно иметь возможность «анализировать» результаты операций анализа. Это означает не что иное, как возможность выделить из всего допустимого для данной операции анализа множества значений параметра (или параметров) те (любые!) значения, которые нас интересуют в данном конкретном алгоритме, и в зависимости от того, попадают или не попадают реальные значения этих параметров в заданный диапазон, – изменить порядок последующих действий, т. е. выбрать один из нескольких возможных путей продолжения алгоритма. Конечно, можно преднамеренно или случайно определить операции анализа так, что многие реально получаемые значения окажутся «некритичными» (т. е. попросту ненужными) ни при каких обстоятельствах, однако следует всегда стремиться к тому, чтобы всё доступное для данной операции множество значений параметров так или иначе представляло какие-либо важные свойства анализируемых конструкций. Конкретные рекомендации в этом плане могут дать материалы главы 1. Тем не менее условия для достижения универсальности систем типа S_U лучше сформулировать так, чтобы подобного рода «случайности» не играли никакой роли.

Итак, вторая теорема об универсальности:

Утверждение 2.2. Для того чтобы система S_U была универсальной в полной технологии $\mathbf{T} = \langle \mathbf{A}, \mathbf{F} \rangle$, достаточно, чтобы:

а) структура системы отвечала условиям утверждения 2.1;

б) устройство управления S_U могло выполнять любую технологическую операцию типа F_i , а также операции перемещения и хранения элементов базы и конструкций T в соответствующих ячейках хранения;

в) устройство управления S_U могло различать *любые* возможные значения *любых* параметров β_i , получаемых в операциях анализа данной технологии, и могло изменять в соответствии с этими значениями дальнейшие последовательности технологических операций.

Необходимость и достаточность п.п. а) и б) (в смысле структурной организации универсальной системы и управления ею) очевидна и следует из утверждения 2.1. Достаточность условий п. в) связана с тем только что рассмотренными обстоятельствами, а именно с необходимостью *произвольно* изменять последовательность технологических операций в зависимости от *любого* полученного в операциях анализа результата. Конец доказательства.

Заметим, что условия утв. 2.2 в устройстве управления модели S_U на рис. 2.1 будут выполнены автоматически, если, во-первых, в качестве такового использовать микроконтроллер, и, во-вторых, все операции множества F технологии T запрограммировать в виде, например, стандартных подпрограмм микроконтроллера, с тем чтобы он мог выполнять их в любой нужной последовательности.

Теперь нетрудно интерпретировать модель на рис. 2.1 как одну из следующих универсальных систем:

1) $A = \{0, A\}$ – элементы множества действительных чисел, заданного отрезком действительной числовой оси $[-A, A]$ ($A > 0$); $F = \{-F_k, -F_{сл}, P\}$, где $-F_k(x) \rightarrow -kx$; $0 < k < 1$; $-F_{сл} = -F_{сл}(x_1, x_2) \rightarrow -(x_1 + x_2)$; $P = P(y) \rightarrow \beta$, причём $\beta = 1$, если $y > 0$ и $\beta = 0$, если $y < 0$. Система может использоваться как базовая (исходная) модель для универсальной (по Тьюрингу) системы преобразования и обработки аналоговых данных [20];

2) $A = \{\text{набор аминокислот}\}$, F – операции «сборки» аминокислот в белковые цепи с помощью рибосом с использованием в качестве программы сборки соответствующей матричной РНК, а в качестве механизмов определения и доставки к месту сборки нужных аминокислот;

кислот – соответствующих транспортных РНК. То есть схема рис. 2.1 интерпретируется как универсальная программно перестраиваемая система для синтеза белков в клетках. В этой интерпретации интересно то, что механизмы анализа текущей конструируемой белковой цепи в данной системе на сегодняшний день неизвестны (т. е. будут синтезироваться и «ошибочные», и неверные, и даже вредные белки, если в программу синтеза по каким-либо причинам вкрались или преднамеренно внедрены ошибки). Таким образом, данная система – пример реализации «синтезатора белков» в «чистом» виде, без использования операций анализа для контроля результатов синтеза;

3) **A** – некоторое подмножество целых n -разрядных двоичных чисел (например, 0 и 1), **F** – множество арифметических и логических операций над этими числами, выполняемых конкретным процессором конкретного компьютера. То есть схема рис. 2.1 интерпретируется как структура компьютера с определённой системой команд;

4) **A** – некоторый набор химических веществ, **F** – стандартные операции химической технологии над ними (смешивание, нагревание, дозирование, фильтрование, сушка, измельчение и т.д.). Схема интерпретируется как универсальная система для синтеза различных веществ, собранная, например, из малогабаритных технологических модулей типа Системы Приборов Лабораторной АВтоматики «СПЛАВ», разработанных С.И. Бородиным в Институте биологической физики Российской академии наук [15, 16];

5) **A** – набор каких-либо стандартных конструктивных элементов (например, элементов конструктора «Лего»), **F** – множество стандартных операций сборки (соединения) этих элементов между собой. Схема интерпретируется как универсальная система синтеза различных механических конструкций в заданной элементной базе;

6) **A** – исходные детали и сырьё; **F** – операции обработки, сборки и анализа промежуточных и «конечных» конструкций. Схема рис. 2.1 интерпретируется как некое гибкое (универсальное) автоматическое программно-управляемое производство (ГАП) в конкретной технологии;

7) **A** – множество дискретных (фиксированных) значений сигнала в линии связи, **F** – операции приёма, передачи и преобразования дис-

кретных сигналов (из множества A). Система рис. 2.1 интерпретируется как многофункциональное приёмо-передающее устройство, способное реализовать *любые* (в том числе ещё не изобретённые!) протоколы последовательного приёма-передачи сигналов по линиям связи [15, 16].

В случае п. 3) для специалистов по вычислительной технике очевидно несоответствие между реальной системой адресации для обращения к ячейкам хранения, используемой в компьютерах, и системой, изображённой на рис. 2.1. Для использования аналогичной системы адресации в модели рис. 2.1 каждой ЯХ необходимо присвоить свой уникальный номер-код («адрес»), который и будет указываться в командах «извлечения» (чтения) данных из ячеек или в командах «размещения» (записи) результатов операций в ячейках хранения (в компьютерах их принято называть ячейками памяти). При таком способе адресации объектов и результатов операций для указания нужной ЯХ необходимо выполнить всего лишь одну операцию (одну команду) «извлечения» или «размещения» нужного объекта по уникальному коду-адресу соответствующей ЯХ. Такой способ гораздо удобнее в практическом плане, поскольку для перемещения указателя к нужной ячейке (в случае использования схемы с регистром сдвига рис. 2.1 или механизма сдвига ленты в м.Т.) может потребоваться не один десяток операций сдвига из одной позиции в другую (а при больших массивах ячеек хранения – и гораздо больше). То есть с точки зрения технической реализации схему адресации ЯХ, эквивалентную компьютерной, имеет смысл применять тогда, когда число ячеек ограничено и не предполагается их увеличение до бесконечности.

Возможны и другие механизмы адресации, отличные от рассмотренных выше. Например, в микробиотехнологиях, используемых живой природой, довольно часто применяется механизм адресации, основанный на совпадении комплементарных (дополняющих друг друга) поверхностей форм конструкций, т.е. на схеме типа «ключ-замок».

Для схемы рис. 2.1 доказано несколько теорем, сравнивающих возможности м.Т. и различных, достаточно простых, технологий [13, 15, 16]. Например:

Утверждение 2.3. Универсальная система S_U (рис. 2.1) для технологии $T = \langle \{a\}, \{F_T, P_0\} \rangle$, где $F_T(x) \rightarrow x$ и в базе A имеются «источники» элементов a и «пустых конструкций» $\square$, может имитировать работу м.Т.

Заметим, что утверждение 2.3 дополнительно подтверждает истинность утверждения 2.2.

Утверждение 2.4. Универсальная система S_U (рис. 2.1) для технологии $T = \langle \{a\}, \{F_S, P_0\} \rangle$, в которой имеются «источники» элементов a , может имитировать работу м.Т.

Утверждение 2.5. Универсальная система S_U (рис. 2.1) для технологии $T = \langle \{a\}, \{F_T, P_a\} \rangle$, где $P_a(x) \rightarrow \beta$ и $\beta = 1$, если $x = a$, или $\beta = 0$, если $x = \square$, а в базе A имеются «источники» элементов a , может имитировать работу м.Т.

Все три утверждения (2.3 – 2.5) говорят о том, что для достижения «алгоритмической полноты», т. е. для создания достаточно мощного информационного базиса в различных технологиях, *эквивалентного по своим потенциальным возможностям вычислительной математике*, нужен не такой уж сложный механизм: необходимо, по крайней мере, иметь «что-то», что можно отличить от «ничего», и уметь это «что-то» располагать в нужных местах пространства (в нужных ячейках хранения) в нужной последовательности с тем, чтобы потом эту последовательность можно было вновь обработать.

С точки зрения гарантии высокого уровня потенциальных возможностей («мощности») технологических систем типа S_U имеет смысл ввести ещё одну универсальную технологическую структуру, которая в явном виде содержит универсальное вычислительное устройство – машину Тьюринга или её аналог – микроконтроллер, гарантирующую, что «алгоритмические способности» («мощность») такой системы никоим образом не будут ниже «алгоритмических способностей» м.Т. или любого из ныне существующих компьютеров.

Структура одной из моделей соответствующей системы, которую можно назвать универсальным синтезатором-анализатором объектов или для краткости – просто синтезатором и которая учитывает «компьютерный» вариант адресации ЯХ (с использованием регистра адреса РА и стандартного двоичного дешифратора), приведена на рис. 2.3. Использование нескольких мультиплексоров и демультиплексоров на рис. 2.3 демонстрирует опять же возможность (с практической точки зрения) реализации различных механизмов размещения объектов и результатов операций в технологических ячейках и ячейках хранения.

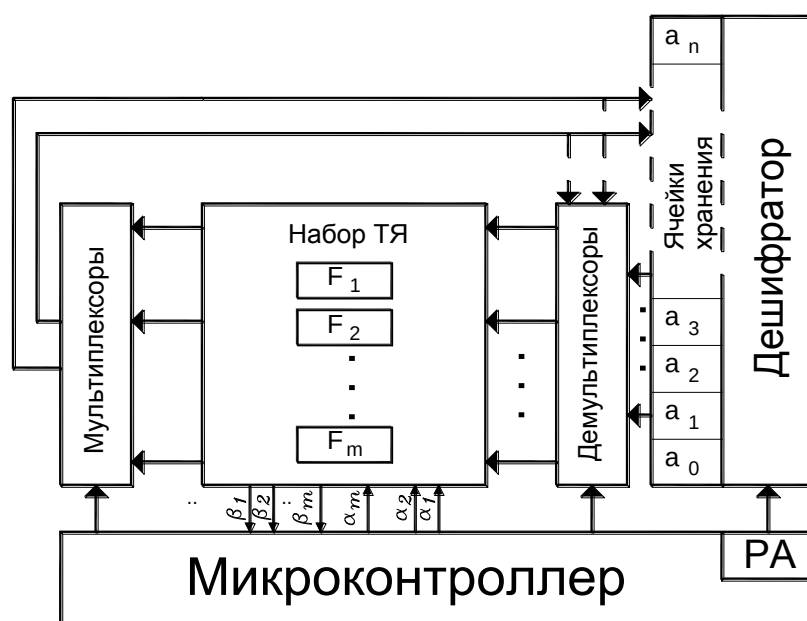


Рис. 2.3. Модель универсального синтезатора-анализатора объектов

Универсальность систем S_U (рис. 2.1) и U_S (рис. 2.3) гарантирует реализуемость в них любых рассмотренных ранее (в том числе в главе 1) процессов (алгоритмов) синтеза и анализа объектов – как, впрочем, и тех, которые будут рассмотрены дальше. Чтобы не перечислять соответствующие утверждения из этой главы ещё раз, мы выберем более простой путь, используя один из вариантов интерпретации «тезиса Тьюринга-Чёрча», утверждающий, что все неформальные алгоритмы, т.е. алгоритмы в «интуитивном» смысле или, как в данном случае, алгоритмы, сформулированные на обычном языке, могут быть формализованы и записаны в строгой математической форме. Используя его, мы сформулируем одно простое, но зато очень общее

утверждение, в котором нет строгой интерпретации конкретных алгоритмов синтеза и (или) анализа объектов:

Утверждение 2.6. Синтезатор-анализатор U_S (рис. 2.3), реализующий конкретную технологию $T = \langle B, F \rangle$, может реализовать в ней любые алгоритмы синтеза и анализа объектов, доступных для данной технологии.

Таким образом, системы типа U_S действительно оказываются потенциально способными к реализации самых различных процессов в конкретной технологии T : синтеза бесконечного разнообразия конструкций в ней, анализа любых конструкций в T и восстановления их синтезов, включая синтезы неизвестных конструкций, и т.д. и т.п. Некоторые из возможных практических реализаций подобных систем будут рассмотрены в следующих параграфах.

2.2. ПРАКТИЧЕСКИЕ СТРУКТУРЫ УНИВЕРСАЛЬНЫХ ПРОГРАММНО-УПРАВЛЯЕМЫХ ТЕХНОЛОГИЧЕСКИХ СИСТЕМ

Высокоуровневая модель универсальной программно управляемой технологической системы, представленная на рис. 2.1 и рис. 2.3, ввиду её близости к структурам м.Т. достаточно удобна для теоретического анализа. Однако для практических целей в неё необходимо внести ряд существенных изменений и уточнений. Эти изменения никоим образом не снижают универсальность модели, но облегчают её практическое использование в конкретных технологических областях и повышают некоторые важные системотехнические параметры. Важнейшим из таких параметров является, естественно, производительность. Для её оценки оказывается полезным следующее формально-технологическое утверждение:

Утверждение 2.7. Если перемещение элементов (конструкций) и установка параметров в операциях $F_i \in F$ универсальной системы S_U (U_S) происходят достаточно быстро и за время, равное некоторой величине τ , то при $\tau \rightarrow 0$ для реализации любого процесса в данной технологии требуется время, приближающееся ко времени реализации аналогично-

го процесса в соответствующей специализированной системе, *имеющей тот же состав технологических ячеек*, что и система S_U .

При доказательстве этого утверждения используется аксиома математической индукции в, так сказать, формально-технологической интерпретации [15, 18]. Само доказательство приведено в [13].

Утверждение 2.7 показывает, что никаких «фундаментальных дефектов» в базовых универсальных системах типа приведённой на рис. 2.3 по сравнению со специализированными нет и быть не может. Особенно если учитывать тот факт, что в реальных специализированных системах для перемещения объектов между технологическими ячейками тоже требуется некоторое время. Проявляющиеся же в созданных нами образцах универсальных систем кажущиеся недостатки (в плане их якобы принципиально меньшего быстродействия в сравнении со специализированными) – лишь следствие не очень правильного подхода к их конструированию. Следуя утверждению 2.7 и структуре рис. 2.3, при проектировании универсальных технологических систем основное внимание нужно обращать на следующие моменты: а) использование высокоэффективных технологических ячеек, выполняющих как можно большее число операций из F (а потому не требующих дополнительных перемещений объектов); б) на сокращение времени транспортировки объектов между различными ячейками (как технологическими, так и ячейками ввода-вывода и хранения); в) на повышение скорости такой транспортировки. В этом контексте имеет смысл обратить внимание на пунктирную связь, «замыкающую» выходы мультиплексоров сразу на входы демультиплексоров на рис. 2.1, рис. 2.3, т.е. на возможность замыкания реализуемой последовательности технологических операций непосредственно между технологическими ячейками. Фактически это эквивалентно временному созданию в массиве технологических ячеек S_U (U_S) специализированной подсистемы, если число имеющихся для этого технологических ячеек достаточно.

Практика подтверждает, что эволюция наиболее продвинутых в эволюционном плане универсальных систем, созданных человеком, – компьютеров – идёт именно по этим направлениям, а именно:

- 1) создание параллельно работающих технологических блоков (процессоров);
- 2) расположение обрабатывающих (технологических) устройств по цепочке, как бы вдоль «конвейера», т.е. создание фрагментов, типичных для специализированных систем;
- 3) введение сверхоперативной локальной кэш-памяти различного уровня, т.е. обеспечение возможно более «короткого» (более быстрого) «замыкания» часто используемых данных с выхода процессора на его вход;
- 4) введение специальных схем предсказания возможных вариантов продолжения программ обработки данных (в частности, создание схем предсказания результатов выполнения так называемых «команд условных переходов»);
- 5) совершенствование операционных (технологических) ячеек для обработки данных, полное согласование производительности подустройств, отвечающих за конкретные операции обработки (RISC-архитектура, аппаратная реализация умножения и деления);
- 6) повышение скорости и пропускной способности магистралей компьютеров для перемещения обрабатываемой и управляющей информации;
- 7) разделение и специализация магистралей на основные и ввода-вывода с различными скоростями передачи данных.

В соответствии с этими рекомендациями более эффективной организацией блока технологических ячеек может быть структура, изображённая на рис. 2.4.

Структура содержит блок технологических ячеек с программируемыми межсоединениями и, в отличие от схемы рис. 2.1, может в максимальной степени реализовать все рекомендации, приведенные выше в п.п. 1) – 7). В частности, в этой структуре возможно (при соответствующей организации самих программируемых межсоединений) непосредственное подключение выхода любой ТЯ к любым (технологическим) входам любых других ТЯ, как это делается в так называемых «универсальных аналоговых вычислительных машинах» (АВМ), которые широко применялись в середине 20-го века для ре-

шения различных систем дифференциальных уравнений. Соответствующая схема межсоединений называется полной матрицей коммутации – ПМК. Помимо максимального сокращения времени транспортировки объектов между ячейками ПМК позволяет удовлетворить перечисленные выше требования п.п. 1), 2). Она также легко вписывается в требования п.п. 3), 5), 6) (за счёт введения в массив ТЯ соответствующих специализированных технологических ячеек, внутренних высокопроизводительных коммутируемых транспортных магистралей, а также возможности подключения входов и выходов ТЯ к нужным устройствам ввода-вывода объектов операций напрямую).



Рис. 2.4. Базовая структура $U_s (S_U)$ для практических реализаций

Требование пункта 4) приведённого выше списка путей повышения производительности компьютеров для многих реальных технологий мало актуально, поскольку многие из них выполняют сами технологические операции достаточно медленно.

Основным недостатком ПМК являются большие аппаратные затраты как на сами транспортные магистрали, так и на коммутационные элементы. Так, для блока ТЯ всего из 4-х ячеек число транспортных магистралей ПМК уже превышает число самих ТЯ. Тем не менее следует признать, что для универсальных систем с достаточно большим набором технологических ячеек ПМК является единственным вариантом, гарантирующим создание любых допустимых для данного набора конфигураций ТЯ с максимально возможной производительностью.

Блок ТЯ на рис. 2.4 отличается от структуры аналогичного блока на рис. 2.3 двумерной матричной организацией. Такие структуры часто используются в аналого-цифровых «системах на кристалле» (СНК) ввиду того, что сами блоки обработки данных (т.е. ТЯ) изготавливаются в виде преимущественно двумерных (планарных) электронных схем на поверхности кремния или другого полупроводникового материала. Других причин, по которым следует отдавать предпочтение подобным двумерным (планарным) матричным структурам в аналого-цифровых СНК, не существует. Более того, с точки зрения сокращения длины транспортных магистралей между ТЯ более эффективными оказываются структуры с компактной трёхмерной и даже n -мерной ($n > 3$) организацией матрицы связей между операционными блоками.

К сожалению, и технологии электронных микросхем, и технологии микроэлектромеханических и микропотоковых систем ориентированы прежде всего на планарные топологии. Поэтому трёхмерные и тем более n -мерные ($n > 3$) технологические блоки СНК – дело будущего.

Сформулируем теперь основные этапы разработки конкретных многоцелевых программно управляемых технологических систем с использованием базовых решений, приведённых на рис. 2.3 – рис. 2.4:

1. На основе анализа (например, статистического) различных процессов обработки (преобразования) объектов *в заданной технологической области выделяются наиболее часто используемые типы технологических и аналитических ячеек и выполняемые ими опера-*

ции. Такой выбор может быть сделан на основе различных умозрительных (экспертных) заключений и оценок, если данная область хорошо знакома разработчикам. Главное, чтобы получающиеся при этом наборы технологических и аналитических операций могли образовывать совокупности (группы) операций, отвечающих критерию полноты, то есть *чтобы можно было выделить* в данной технологической области *полные технологии*. Определёнными ориентирами для такого выбора могут служить материалы главы 1.

2. С использованием базовой рекурсивной модели универсальной технологической системы S_U , приведённой на рис. 2.4, а также рассмотренных в параграфах 1.2 – 1.3 формально-технологических подходов и методов анализируются потенциальные возможности различных вариантов предположительно полных технологий как на предмет их реальной полноты, так и в плане синтеза и анализа максимально широкого класса объектов в заданной технологической области. *Выбирается тот вариант полной технологии, который позволяет синтезировать и анализировать максимально широкий класс объектов и достаточно просто реализуется технически.*

3. В соответствии с результатами предыдущего пункта и материалами данного параграфа (в первую очередь базовых структур универсальных технологических систем, рис. 2.4) *выбирается конкретный вариант структуры для технической реализации, который корректируется с учётом реального состава технологических операций (технологических ячеек).*

Рассмотрим несколько примеров.

2.3. МНОГОФУНКЦИОНАЛЬНЫЕ АНАЛОГО-ЦИФРОВЫЕ ПРОГРАММИРУЕМЫЕ СИСТЕМЫ ОБРАБОТКИ СИГНАЛОВ

Дискретно-аналоговый процессор (ДАП) – одно из первых многоцелевых аналого-цифровых устройств, разработанных с использованием подходов и методов ОФТ. В общих словах ДАП можно определить как универсальное устройство для организации эффективной

взаимосвязи между различными аналоговыми и цифровыми системами и подсистемами, т.е. как устройство, предназначенное для решения задач, связанных с вводом, выводом и преобразованием аналоговой и цифровой информации, представленной самыми различными типами аналоговых и цифровых (дискретных) сигналов [13 – 16].

Число таких (различных) типов дискретных и аналоговых сигналов, представляющих самые различные параметры интересующих нас процессов, достаточно велико, да и сами эти сигналы могут значительно отличаться друг от друга. Разрабатывать каждый раз специальные устройства для решения конкретных задач ввода, преобразования и вывода конкретных типов сигналов для компьютеров или других микропроцессорных систем оказывается накладно. Особенно заметным это становится на фоне постоянно дешевеющих цифровых устройств обработки информации – тех же микропроцессоров, микроконтроллеров, микрокомпьютеров и т.д. Было бы намного разумнее и экономически выгоднее не тратить каждый раз время и деньги на разработку, сборку и отладку таких специализированных устройств ввода-вывода, а попытаться сделать максимально многофункциональное (т.е. достаточно универсальное) устройство, которое, во-первых, можно было бы выпускать большими сериями (по причине его универсальности), а во-вторых, легко было бы запрограммировать, подобно микропроцессорам, на решение любой конкретной задачи ввода-вывода и преобразования аналоговых сигналов.

Впервые такого рода задача была сформулирована и решена (в одном из вариантов) в статье, опубликованной в отечественном отраслевом научно-техническом журнале «Электронная промышленность» в 1981 г.² Через шесть лет актуальность и важность этой задачи была подтверждена ведущими американскими специалистами в области микроэлектроники³. Однако решения, предложенные ими, оказались гораздо менее эффективными, чем решения, заявленные в

2 Крылов С.М. Программируемый аналоговый интерфейс для микроЭВМ // Электронная промышленность. – 1981. – № 7-8. – С. 126-130.

3 Коул Б.К. Развитие техники и технологии однокристалльных аналоговых БИС- и СБИС-систем // Электроника. – 1987. – № 11. – С. 33-38.

упоминавшейся статье. Поэтому через некоторое время фирмы, разработки которых основывались на концепциях американских инженеров, вынуждены были просто уйти с рынка, отказавшись от производства соответствующих микросхем.

Чем же так радикально отличались решения, сформулированные на базе ОФТ-подхода? В первую очередь – буквально дословным выполнением всех рекомендаций п.п. 1-3 основных этапов алгоритма разработки многофункциональных систем, приведённом в конце параграфа 2.2. Подробно этот анализ и соответствующие выводы рассмотрены в [13 – 15], здесь же приведём наиболее важные результаты.

С учётом специфики обрабатываемых данных, исходная структурная схема рис. 2.3, рис. 2.4 модифицирована и уточнена в первую очередь с учётом важности тех или иных операций, используемых в различных технологиях T_i обработки сигналов (см. рис. 2.5).

Структура рис. 2.5 содержит технологические ячейки, выполняющие следующие функции: ячейка $F_{\text{СЛИ}}$ – операции аналогового сложения, интегрирования, масштабирования, умножения на программно задаваемый коэффициент (вместе с ячейкой F_k) и – при необходимости – дифференцирования.

Ячейка F_k предназначена для умножения на различные, устанавливаемые программно, коэффициенты. Ячейка $F_{\text{ИЭКС}}$ может выполнять операции интегрирования и поиска экстремумов сигнала на интервале. Ячейка $F_{\text{ПНТ}}$ предназначена для непрерывного усиления сигналов напряжения и их преобразования в ток. P_2 – сравнивающее устройство (аналитическая технологическая ячейка), вычисляющее предикат P_2 , причём $P_2(x) = 1$, если $x \geq 0$, и $P_2(x) = 0$, если $x < 0$. Наконец, $F_{\text{СЧ}}$ – специальная технологическая ячейка, выполняющая операции счёта времени и событий (то есть просто-напросто так называемый «счётчик-таймер», часто используемый в микропроцессорных системах для привязки к шкале реального времени, подсчёта числа событий, представленных электрическими импульсами; для деления частоты входного сигнала и для некоторых других целей).

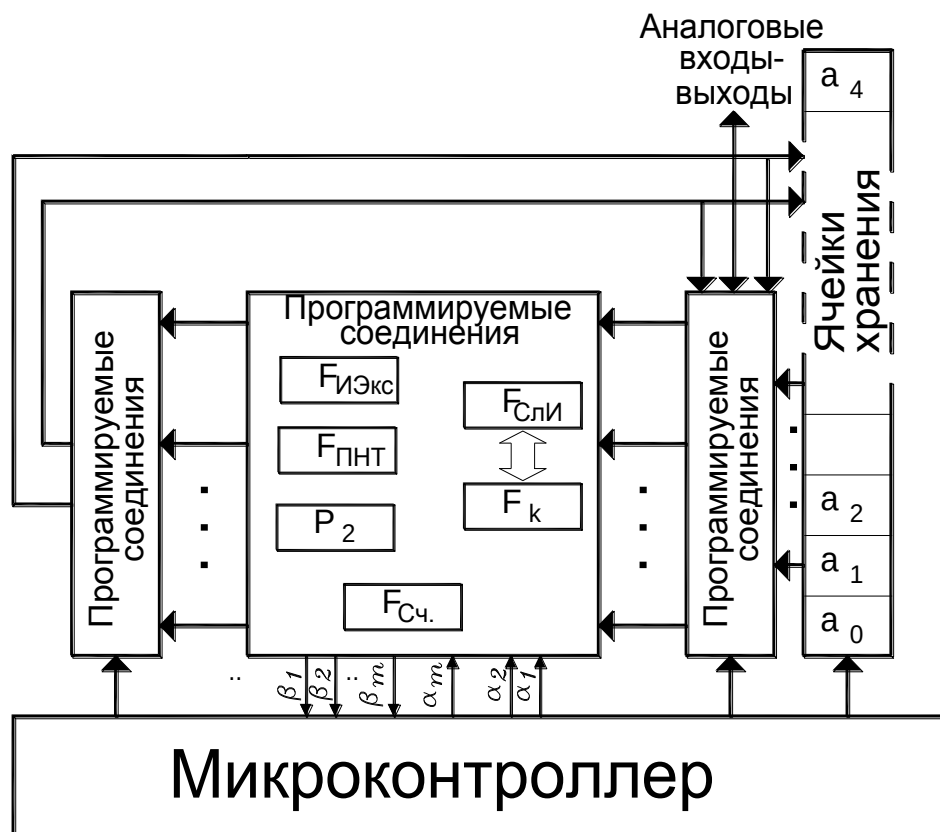


Рис. 2.5. Базовая структура ДАП, полученная в ходе теоретического и практического анализа различных наборов технологических операций для решения задач ввода-вывода и преобразования аналоговых сигналов

Двунаправленная двойная стрелка на рис. 2.5 между ТЯ $F_{\text{СЛИ}}$ и ТЯ F_k означает, что некоторые операции обработки сигналов эти две ТЯ выполняют вместе [13 – 15].

Для подробного теоретического анализа потенциальных возможностей систем типа ДАП и их сравнения с вычислительными возможностями («мощностью») м.т. на основе структур рис. 2.1 и рис. 2.5 был разработан целый ряд моделей, названных дискретно-аналоговыми машинами – ДАМ⁴. Для этих моделей проанализированы нескольких вариантов полных технологий с тем, чтобы выбрать из них наиболее эффективную для решения следующих задач: распознавание (анализ) различных аналоговых сигналов, генерирование (т.е. синтез) различных аналоговых сигналов, преобразование различных аналоговых сигналов (друг в друга, а также в цифровой код и обратно).

⁴ Крылов С.М. Модели универсальных дискретно-аналоговых машин на основе машины Тьюринга // Электронное моделирование. – 1982. – № 3. – С. 6-10.

В частности, были рассмотрены ДАМ со следующими технологиями:

1. $M_0 = T_0 = \langle [A_1, A_2], F_T, P_1 \rangle$, где $F_T = F_T(x) \equiv x$ – операция повторения входного сигнала; предикат $P_1(x)=1$, если $x \geq \alpha_1$, и $P_1(x)=0$, если $x < \alpha_1$, $\alpha_1 \in (A_1, A_2)$. A_1, A_2 – некоторые действительные числа (константы, причём $0 \leq A_1 < A_2$);

2. $M_1 = T_1 = \langle [A_1, A_2], F_T, F_k, P_1 \rangle$, где F_k – «креативная» (см. параграф 1.3) операция, формирующая из $x \in [A_1, A_2]$ элемент y , отличный от x : $y = F_k(x) = kx$; $0 < k < 1$; $x, y \in [A_1, A_2]$; $F_k^i(x) = k^i x$.

3. $M_2 = T_2 = \langle [A_1, A_2], F_T, F_k, F_{\text{сл}}, P_1 \rangle$, где $F_{\text{сл}}(x_1, x_2) = x_1 + x_2 = y$ – операция сложения; $x_1, x_2 \in [A_1, A_2]$.

4. $M_3 = T_3 = \langle [-A, A], F_k, -F_{\text{сл}}, P_2 \rangle$, где $-F_{\text{сл}} = -(x_1 + x_2)$ – сложение с инверсией; $P_2(x) = 1$, если $x \geq 0$, и $P_2(x) = 0$, если $x < 0$.

5. $M_4 = T_4 = \langle [-A, A], -F_k, -F_{\text{сл}}, -F_{\text{и}}, P_2 \rangle$, где $-F_{\text{и}}$ – операция аналогового интегрирования (по времени) с инверсией знака результата операции.

6. $M_6 = T_6 = \langle [-A, A], -F_k, -F_{\text{сл}}, -F_{\text{и}}, -F_{\text{д}}, P_2 \rangle$, где $-F_{\text{д}}(x)$ – операция дифференцирования с инверсией знака результата операции: $-F_{\text{д}}(x) = -dx(t)/dt = y$; $x, y \in [-A, A]$.

7. Ряд других ДАМ с соответствующими технологиями.

Поскольку все ДАМ имеют одинаковую структуру, в пунктах 1 – 7 машины M_i как бы приравниваются к конкретным технологиям T_i , так как единственное их отличие друг от друга – это состав операций, реализуемых набором ТЯ.

Кроме этого, исследовалась также модель типа приведённой на рис. 2.5, но ориентированная на технологию обработки и преобразования последовательности дискретных сигналов в многофункциональных (универсальных) системах приёма-передачи данных⁵, которую мы обозначим как $M_{\text{УППП}} = T_{\text{УППП}}$ (модель и соответствующая технология универсального последовательного приёмника-

⁵ Крылов С.М. Универсальность устройств преобразования, генерирования и приёма дискретных сигналов // Управляющие системы и машины. – 1983. – № 2. – С. 7-10.

передатчика), причём $T_{уппп} = \langle [-A, +A], F_{\Gamma}, P_0 \rangle$, где F_{Γ} – операция генерирования выходного сигнала в линию связи: $F_{\Gamma}(x) = +A$, если $x = 1$, и $F_{\Gamma}(x) = -A$, если $x = 0$; а предикат $P_0(x)$ определяет уровень сигнала в линии связи: $P_0(x) = 1$, если $x \geq 0$, и $P_0(x) = 0$, если $x < 0$, A – некоторое действительное число (константа).

Выяснилось, что все перечисленные выше машины могут имитировать работу м.Т. (то есть все они, так сказать, универсальны «по Тьюрингу»), но задачи преобразования уровней аналоговых сигналов в цифровой код и обратно могут решать лишь машины начиная с M_3 . Более того, для машин $M_0 \dots M_3$ легко указать очень простые задачи на множестве действительных чисел, которые решаются одним типом машин и не решаются другим практически независимо от точности выполняемых вычислительных операций. Выяснилось также, что при решении задач синтеза и анализа аналоговых сигналов увеличение числа интеграторов и дифференциаторов ведёт к повышению способностей соответствующих ДАМ генерировать и распознавать всё большее число таких сигналов. Иными словами, среди ДАМ, универсальных по Тьюрингу, существует сложная иерархия, в отличие от чисто дискретных универсальных вычислительных машин Минского, Поста, Колмогорова, Маркова, $M_{уппп}$ и других (которые по своим «вычислительным способностям» эквивалентны друг другу, в том числе и машинам Тьюринга). То есть получается, что действительно универсальную дискретно-аналоговую машину в «дискретно-аналоговой технологии обработки и преобразования аналоговых сигналов» построить нельзя! Используя введённые выше договоренности об обозначении потенциальных возможностей соответствующих технологий обработки сигналов, полученный результат можно записать так:

$$|T_{м.т.}| = \dots = |T_{уппп}| = |T_0| < |T_1| < |T_2| < |T_3| < |T_4| < |T_6| < \dots \quad (2.1)$$

К счастью, при более конкретном учёте точности решения задач по анализу и синтезу сигналов выяснилось, что на самом деле ситуация не так безнадежна, как это следует из рассмотрения идеализиро-

ванного варианта. Оказалось, что погрешность генерирования различных сигналов в машине с n интеграторами в сравнении с машинами с $(n-1)$ интеграторами (т.е. на 1 интегратор меньше) с увеличением n падает, приближаясь к нулю при $n \rightarrow \infty$. При этом для практических целей во многих случаях достаточно располагать всего двумя-тремя интеграторами.

Аналогичная ситуация складывается и при анализе сигналов – во многих случаях достаточно всего двух-трёх дифференциаторов, чтобы получить более или менее полную информацию о характере анализируемых сигналов. Договоримся дискретно-аналоговые машины с n интеграторами и n дифференциаторами обозначать как T_6^n , где $n > 1$, считая, что остальные вычислительные операции в T_6^n совпадают с технологией T_6 . Тогда с практической точки зрения соотношение (2.1) можно переписать так:

$$|T_{м.т.}| = |T_0| < |T_1| < |T_2| < |T_3| < |T_4| < |T_6| < |T_6^2| < |T_6^3| \approx |T_6^4| \approx \dots \approx |T_6^n|. \quad (2.2)$$

С учётом (2.2), а также *дополнительного анализа других практических аспектов решения различных типовых задач ввода-вывода и преобразования аналоговой информации*, и получается базовая структура ДАП для практической реализации, изображённая на рис. 2.5.

На основе схемы рис. 2.5 разработаны также версии дискретно-аналогового процессора сопряжения цифровых и аналоговых систем, пригодные для изготовления в виде больших интегральных схем – БИС⁶.

Следует отметить, что в соответствии с требованиями утв. 2.1 и утв. 2.2 получающаяся на основе рис. 2.5 рекурсивная структура ДАП помимо экономии оборудования (за счёт многократного использования одних и тех же ТЯ в процессах обработки сигналов) позволяет легко реализовать различные алгоритмы коррекции разного рода систематических погрешностей путём их запоминания и последующего автоматического учёта в вычисляемом результате. На основе данного

⁶ Дмитриев Д.П., Крылов С.М. БИС дискретно-аналогового процессора // Электронная техника. Сер. 3. Микроэлектроника. – Вып.1 (153). – 1999. – С. 32-36.

обстоятельства создан целый ряд изобретений, реализующих рекурсивные алгоритмы коррекции такого рода погрешностей.

Используя полученные выше результаты, в том числе некоторые элементы методики анализа и выбора состава операций для блока ТЯ, попытаемся теперь выработать требования к структурной организации универсальных программируемых аналого-цифровых систем максимально широкого назначения, т.е. таких, входными и выходными сигналами в которых могут быть как цифровые, так и аналоговые сигналы с самыми различными характеристиками и параметрами. Ясно, что именно такие структуры и должны лежать в основе разрабатываемых многофункциональных программируемых аналого-цифровых «систем на кристалле» (МПАЦ СНК), поскольку именно они гарантируют максимально широкую сферу их применения и, соответственно, максимальный объём потенциального рынка сбыта таких изделий.

Основным отличием цифровых, аналоговых и аналого-цифровых систем на кристалле от схемы ДАП на рис. 2.5 является наличие в них большого числа так называемых «конфигурируемых блоков». Такие блоки могут состоять из наборов функционально полных компонентов, которые путём программирования соединений как внутри блоков, так и между ними могут формировать нужные потребителю подсистемы обработки информации.

Основой полноценных аналого-цифровых СНК является универсальное цифровое вычислительное устройство – микроконтроллер (см. рис. 2.4 или рис. 2.5) с полным набором необходимых (основных) функциональных блоков, а именно: с полноценным арифметико-логическим устройством (АЛУ), выполняющим все основные («технологические») операции технологии обработки цифровой информации (в том числе операции умножения и деления); с достаточно большим набором РОН – регистров общего назначения; с блоками оперативной (ОЗУ) и постоянной (ПЗУ) памяти достаточной ёмкости; со стандартным набором таймеров-счетчиков, последовательными и параллельными портами; с гибкой системой прерываний и т.д. Набор

таких блоков гарантирует не только полноту технологии цифровой обработки информации в формально-технологическом и математическом смыслах, но и эффективные методы этой обработки, а также использование стандартных протоколов приёма и передачи цифровых данных для подключения к стандартным источникам и приёмникам.

Что же касается функциональной полноты базовых блоков и элементов, используемых в цифровых технологиях обработки информации, то ситуация здесь выглядит следующим образом.

Во-первых, основой практически всех современных цифровых схем обработки информации являются так называемые двоичные логические элементы, реализующие функционально полные наборы булевских функций (т.е. логические элементы типа «И-НЕ», «ИЛИ-НЕ» и др.). Как мы знаем, с помощью таких наборов можно реализовать любую вычислительную функцию, поэтому, собственно, такие наборы и называются функционально полными⁷.

Во-вторых, в технологии обработки цифровой информации совершенно чётко прослеживается второй – более высокий – уровень отдельных функциональных узлов и блоков: счётчиков, регистров, сумматоров, мультиплексоров и демультиплексоров, связанный с необходимостью параллельной обработкой и передачей двоичной информации – для повышения производительности. По этим причинам при реализации реконфигурируемой цифровой части аналого-цифровых СНК также возможны несколько вариантов достижения необходимой функциональной полноты, а именно:

1. Использование только набора коммутируемых элементарных функционально полных логических элементов (т.е. массивов программируемых логических вентилей, называемых обычно FPGA) – как, например, в аналого-цифровых СНК FIPSOC⁸. При этом функциональная гибкость системы оказывается максимальной, поскольку из таких эле-

⁷ см., например, *Адельсон-Вельский Г.М., Кузнецов О.П.* Дискретная математика для инженера. – 2-е изд. – М.: Энергоатомиздат, 1988. – 480 с.

⁸ микросхемы разработаны испанской фирмой SIDSA во второй половине 90-х годов 20-го века. В настоящее время не выпускаются.

ментарных блоков могут быть реализованы как стандартные, так и любые нестандартные устройства обработки цифровой информации. Расплатой за это оказываются высокие аппаратные затраты, требуемые для реализации FPGA сколько-нибудь существенных размеров.

2. Использование набора программно настраиваемых стандартных цифровых блоков – тех же счётчиков, регистров, сумматоров, коммутаторов и т.д. Реконфигурационная гибкость в этом случае оказывается минимальной, так как вне рамок структур блоков никакие функциональные устройства создать невозможно. Однако аппаратные затраты при этом становятся относительно небольшими за счёт частичной структурной предорганизации и предкоммутации блоков.

Преимущественно именно по этому пути пошла фирма Cypress Semiconductor со своими первыми успешными аналого-цифровыми СНК, получившими название PSoC1⁹.

3. Сочетание FPGA и стандартных программно настраиваемых цифровых блоков. Этот вариант наиболее эффективен как с точки зрения аппаратных затрат, так и с точки зрения функциональной гибкости. Действительно, если состав укрупненных цифровых блоков выбран оптимально, то он в основном закрывает требуемые дополнительные функциональные потребности СНК, а небольшой добавочный массив FPGA обеспечивает их эффективную подстройку и модификацию в тех случаях, когда функциональных возможностей стандартных цифровых блоков не хватает. Такой вариант фирма Cypress выбрала для последующих моделей своих МПАЦ СНК – PSoC3, PSoC4 и PSoC5.

Функциональная полнота аналоговых вычислительных устройств достигается, согласно К.Шеннону и другим авторам, уже в функциональном базисе операций аналогового сложения и интегрирования (с масштабированием)¹⁰. Для упрощения многих устройств обработки

⁹ Объем продаж PSoC1 в 2006 г. превысил 100 млн долл. США. См., например, <http://www.cypress.com>.

¹⁰ Шеннон К. Математическая теория дифференциального анализатора. В кн.: Работы по теории информации и кибернетике: Пер.с англ. – М.: Иностранная литература, 1963. – С. 709-728.

аналоговой информации в них часто вводят также операции умножения аналоговых величин. То есть с теоретической и практической точек зрения основой конфигурируемых аналоговых блоков (КАБ) должны быть технологические ячейки, соответствующие технологиям T_6^2 , T_6^3 из соотношения (2.2), т.е. похожие на технологические ячейки ДАП (см. рис. 2.5). Весьма близкий состав конфигурируемых аналоговых блоков (фактически, программируемых ТЯ) выбрала и фирма Cypress для своих микросхем PSoC3 – PSoC5.

В то же время разработчики программируемых аналоговых матриц типа TRAC020LH и AN10E40 определили число КАБ в максимальных конфигурациях равным 20, хотя впоследствии фирма Anadigm Inc. (разработчик AN10E40) выпустили версии таких матриц с 2-мя и 4-мя КАБ-ами.

Очевидно, исходя из возможности параллельной обработки аналоговой информации, а также из соображений, приведенных в параграфе 2.2, конфигурируемые аналоговые блоки МПАЦ СНК должны иметь близкую к полной матрицу коммутации и достаточное число аналоговых входов и выходов (по крайней мере, соизмеримое с числом КАБ). Реализация ПМК для числа КАБ больше 9 представляется малоцелесообразной ввиду больших аппаратных затрат на сами межсоединения и их коммутацию. Так, для 9-ти КАБ число межсоединений в ПМК равно 36, а для 10 – уже 45!

Наличие в МПАЦ СНК цифровых и аналоговых конфигурируемых блоков (КБ) требует также введения смешанных, т.е. аналого-цифровых конфигурируемых блоков, призванных решать задачи, связанные с сопряжением аналоговых и цифровых подсистем СНК, а также задачи преобразования аналоговых сигналов в цифровые и обратно. Функциональная полнота различных наборов операций для эффективного решения этих задач анализировалась выше для ДАП.

Один из вариантов структуры МПАЦ СНК, соответствующей сформулированным выше требованиям и содержащий все необходимые для достижения максимальной гибкости (т.е. фактической универсальности) компоненты, показан на рис. 2.6.

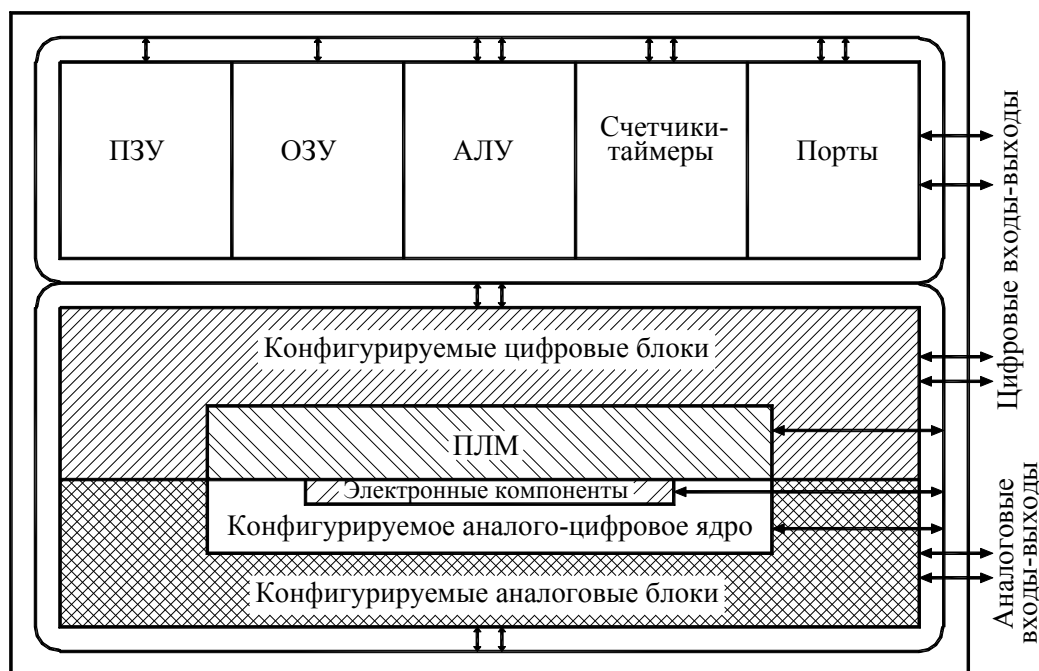


Рис. 2.6. Перспективная структура многофункциональной программируемой аналого-цифровой «системы на кристалле»

Следует подчеркнуть, что *топологическая (пространственная) близость отдельных конфигурируемых блоков на рис. 2.6 означает на самом деле ещё и их реальную тесную связь в плане создания возможных электрических соединений*. То есть блок ПЛМ (программируемых логических матриц – более простой аналог FPGA) должен, например, иметь возможность эффективного соединения как с входами-выходами набора цифровых КБ, так и с входами-выходами конфигурируемых блоков аналого-цифрового ядра и набором конфигурируемых электронных компонентов.

Аналогично, конфигурируемые аналоговые блоки (например, на основе схем на переключаемых конденсаторах) должны иметь возможность образовывать достаточно тесные связи как с аналого-цифровым ядром, так и – в меньшей степени – с набором цифровых КБ, и т.д. Невыполнение этих требований в PSoC1 снизило гибкость и эффективность применения данной микросхемы.

В качестве основных цифровых КБ в такой МПАЦ СНК могут использоваться 4-х или 8-разрядные счётчики, регистры сдвига, по-

следовательные (поразрядные) сумматоры, конфигурируемые по режимам работы, источникам входных сигналов, приёмникам выходных сигналов и т.д. Возможен и более универсальный вариант реализации таких КБ¹¹. Общее число конфигурируемых цифровых блоков – порядка 10-20 штук двух-трёх типов (счётчики – в обязательном порядке, регистры сдвига, сумматоры по модулю и т.д.).

Блок ПЛМ должен содержать как минимум 30-40 двухвходовых логических элементов с конфигурируемыми межсоединениями, с выходами и входами, ориентированными на подключение как к входам и выходам цифровых КБ, так и к входам и выходам аналого-цифрового ядра, которое фактически *представляет собой несконфигурированную структуру аналого-цифровой части ДАП* (см. рис. 2.5). Такое ядро должно включать небольшой набор отдельных («единичных») электронных компонентов (резисторы, полевые транзисторы и проч.), выводы которых подключены либо к выводам СНК, либо к программируемой матрице коммутации.

Собственно аналоговая конфигурируемая часть может содержать до 10-15 аналоговых КБ на переключаемых конденсаторах, способных выполнять операции аналогового масштабирования, инвертирования, суммирования, интегрирования, фильтрования, цифро-аналогового и аналого-цифрового преобразования и др.

Структура МПАЦ СНК (см. рис. 2.6) позволяет создавать оригинальные аналого-цифровые аппаратно-программные конфигурации, уже опробованные в различных универсальных аналого-цифровых вычислительных системах типа ДАП. Конкретные варианты таких конфигураций защищены рядом авторских свидетельств и патентов. Эти конфигурации дают на широком классе задач производительность и быстродействие, в 2-10 раз превышающее параметры PSoC1, выполненного на той же элементной базе.

Наличие многоуровневых конфигурируемых компонентов в МПАЦ СНК оставляет возможность для создания других оригиналь-

11 см., например, Крылов С.М., Хлопотов И.А. Многофункциональный цифровой блок для систем на кристалле // Вестник Самар. гос. техн. ун-та. – 2005. – Вып. 33. – С. 179-184.

ных и эффективных аналого-цифровых структур (так называемых IP-модулей, т.е. оригинальных аппаратных конфигураций с программно-информационной поддержкой), неизвестных на момент разработки самих МПАЦ СНК и обладающих, в том числе, патентной новизной.

Ещё больше возможностей открывается перед МПАЦ СНК, если при их разработке учитывать возможности гетерогенных (разнородных) электрических сигналов и соответствующих электронных систем для их обработки [6, 15].

В заключение отметим, что структура МПАЦ СНК, приведённая на рис. 2.6, хотя и совпадает в основных чертах со структурой PSoC, но отличается от них большей гибкостью, позволяющей с минимальным увеличением аппаратных затрат достичь максимальной эффективности и быстродействия при решении самых различных задач. Более того, с помощью подобных МПАЦ СНК легко реализовать дистанционные (управляемые, например, через Интернет) универсальные (*предельно гибкие*) лаборатории в различных областях. Структура одной из них приведена на рис. 2.7.

Левая часть МПАЦ СНК на рис. 2.7 (со своим микроконтроллером) обеспечивает реализацию набора так называемых «виртуальных приборов», необходимых для задания нужных тестовых сигналов на правую, экспериментальную часть МПАЦ СНК, реакция которой снова измеряется подходящими «виртуальными приборами» левой части.

Информация о результатах эксперимента собирается левыми микроконтроллерами и через интерфейс, сервер и адаптер сети в нужном формате передаётся экспериментатору, который оценивает результаты проведённого эксперимента и опять же через сеть может нужным образом реконфигурировать экспериментальную схему, дистанционно «собираемую» в правой части. Важное преимущество такой лаборатории – то, что экспериментатор практически полностью свободен в выборе исследуемых конфигураций схем, в отличие, например, от дистанционной лаборатории по микроэлектронике Массачусетского технологического института iLab¹².

¹² см. <https://wikis.mit.edu/confluence/display/ILAB2/iLabs>

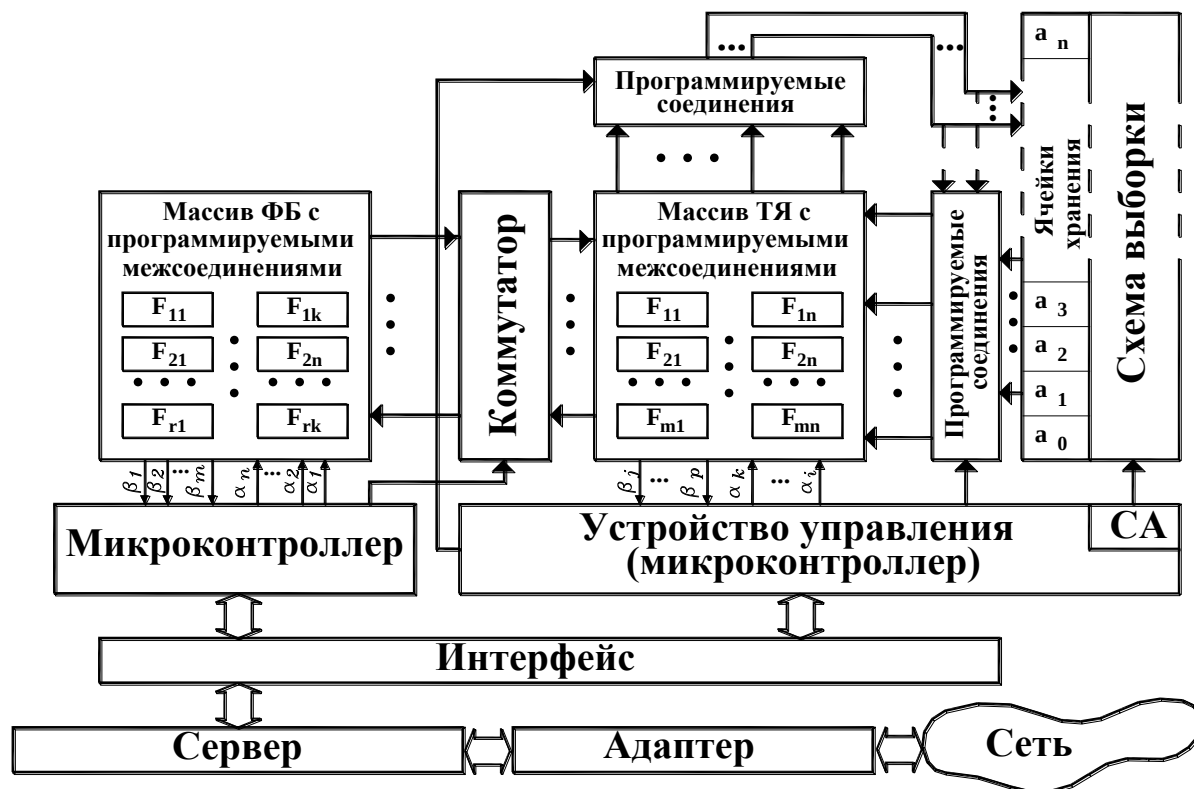


Рис. 2.7. Структура дистанционной лаборатории по электронике для выполнения реальных экспериментов

На тех же принципах, с соответствующими изменениями в правой части рис. 2.7, могут быть построены дистанционные лаборатории для проведения реальных экспериментов в других областях, включая химию и машиностроение (см. параграфы 2.3 – 2.6).

Важной особенностью таких лабораторий (см. рис. 2.7) является использование МПАЦ СНК в левой части системы для создания необходимых для тестирования и измерения реакции правой части «виртуальных приборов», которые можно оперативно и дистанционно менять в зависимости от меняющихся требований и условий экспериментов. Например, вместо сравнительно низкоскоростного цифрового осциллографа можно – при исследовании, например, быстрых стационарных переходных процессов – использовать более скоростные цифровые осциллографы на основе стробоскопических эффектов¹³.

¹³ Крылов С.М., Лавров Д.В., Сараев М.В. Синтез стробоскопических алгоритмов для многофункциональных гетерогенных информационно-измерительных систем // Вестник Самар. гос. техн. ун-та. Сер. Технические науки. – 2009. – № 3 (25). – С. 82-87.

Подобные дистанционные лаборатории могут работать 24 часа в сутки и предоставлять свои ресурсы всем желающим – как для учебных целей (например, для дистанционного выполнения самых различных *реальных* лабораторных работ при изучении различных дисциплин, включая электронику), так и для экспериментальных, исследовательских. Одна из таких дистанционных лабораторий, как уже отмечалось, создана и работает на кафедре «Вычислительная техника» Самарского гостехуниверситета [33].

2.4. УНИВЕРСАЛЬНЫЙ ПРОГРАММИРУЕМЫЙ ХИМИЧЕСКИЙ СИНТЕЗАТОР-АНАЛИЗАТОР

В предыдущем параграфе рассмотрены структуры многоцелевых программируемых систем в основном для одного класса задач: для смешанных (аналогово-цифровых) систем обработки информации (т.е. сигналов). Рассмотрим теперь пример формально-технологического синтеза структуры многоцелевой программируемой системы для совершенно другой технологической области, а именно для области химических технологий.

Согласно формально-технологическому определению, полнота технологии означает, во-первых, возможность бесконечного синтеза в ней всё новых и новых объектов (называемых конструкциями). Во-вторых, – возможность восстановления по любой конструкции процесса её синтеза в этой технологии, т.е. возможность полного анализа состава и структуры конструкции для точного восстановления процесса её получения в соответствующем универсальном синтезаторе-анализаторе объектов по соответствующей программе.

Предварительный анализ показывает, что наиболее подходящей для изготовления такого многофункционального синтезатора-анализатора может быть простейшая химическая технология, оперирующая с растворами веществ, т.е. «генерирующая» в качестве «конструкций-объектов» растворы заданной группы веществ различной концентрации и, кроме того, способная определять параметры этих

растворов (т.е. их состав и текущую концентрацию). Для определённости будем считать, что такой синтезатор-анализатор должен работать с некоторым набором *водных* растворов некоторого конечного (исходного) набора растворимых в ней веществ $\{a_1, a_2, \dots, a_n\}$, которые будем считать элементами базы **A** данной технологии **T_Р**. Если мы хотим получать в **T_Р** все возможные растворы любой возможной концентрации – от максимальной (предельно насыщенные растворы) до минимальной (чистый растворитель), исходные растворы $\{a_1, a_2, \dots, a_n\}$ должны быть насыщенными. Кроме того, для разбавления исходных растворов и получения «растворов с нулевой концентрацией» мы должны ввести в состав элементов базы **A** сам растворитель (в данном случае – воду), который обозначим символом a_0 . Таким образом, база технологии **T_Р** полностью определена: $\mathbf{A} = \{a_0, a_1, a_2, \dots, a_n\}$.

С точки зрения ОФТ бесконечная креативность для конечного набора исходных объектов технологии $\{a_0, a_1, a_2, \dots, a_n\}$, каждый из которых представляет собой либо раствор (включая смеси разных растворов), либо растворитель, означает возможность получения любых новых растворов в различных сочетаниях и с любыми концентрациями, не превышающими, естественно, исходных (предельных) концентраций растворов-элементов базы $\{a_1, a_2, \dots, a_n\}$. Говоря формально-технологическим языком, мы должны обеспечить максимально плотное заполнение «пространства возможных конструкций» в технологии **T_Р**. Нетрудно показать, что для технологии **T_Р** это можно сделать с помощью двух технологических операций: операции взятия фиксированной части (объема) d исходного объекта (раствора или растворителя) x , если $x \leq d$: $F_d(x) = \langle d, (x-d) \rangle$, и операции синтеза нового объекта (раствора) y из двух объектов-растворов x_1 и x_2 : $F_S(x_1, x_2) \rightarrow x_1 + x_2 \rightarrow y$. Для технологии **T_Р** операции типа $F_d(x)$ интерпретируются как операции дозирования (т.е. «отъятия» заданной части x), а операции типа $F_S(x_1, x_2)$ – как операции смешивания двух исходных компонентов (растворов или раствора и растворителя) x_1 и x_2 для получения (синтеза) нового объекта (раствора) y .

Для обеспечения полноты технологии T_p можно использовать различные системы операций анализа (см. параграф 1.3). В химии для качественного и количественного анализа часто используют операции, связанные с применением специальных веществ-индикаторов (т.е. своего рода меток, «маркеров» или «маркираторов»), образующих соединения только с определёнными типами веществ (или даже с определёнными фрагментами макромолекул, особенно при микробиохимических анализах), которые обладают специфическими оптическими свойствами. В качестве примеров можно привести различные вещества-индикаторы уровня кислотности (или щелочных свойств) растворов, специфические реагенты на элементы конкретных подгрупп периодической системы Менделеева и т.д. Для определения не только качественного, но и количественного присутствия того или иного из базовых элементов $\{a_1, a_2, \dots, a_n\}$ в неизвестном растворе-«конструкции» z , необходимы также специфические реагенты-«нейтрализаторы», которые блокируют соединение базовых элементов $\{a_1, a_2, \dots, a_n\}$ со своими индикаторами, как бы замещая последние в такого рода соединениях. Желательно при этом, чтобы «нейтрализаторы», в отличие от индикаторов, не меняли оптические свойства среды (т.е. исходных растворов). Таким образом, для выполнения полного количественного и качественного анализа любого раствора, содержащего любые возможные для данного исходного набора $A = \{a_0, a_1, a_2, \dots, a_n\}$ комбинации (смеси-растворы) элементов из множества $\{a_0, a_1, a_2, \dots, a_n\}$, может потребоваться, в худшем случае, для каждого компонента a_i ($i > 0$, т.е. кроме a_0) свой индикатор b_i и свой «нейтрализатор» c_i , не взаимодействующие с другими компонентами типа a_j ($i \neq j, i, j > 0$). На самом деле такое жёсткое условие не является обязательным. Нетрудно подобрать другие, к примеру, «классифицирующие» системы индикаторов-нейтрализаторов с *рациональным* разбиением (см. параграф 1.3). Напомним, что такие аналитические операции работают по принципу «дихотомического спуска (или алгоритма)». Сначала с их помощью исходные анализируемые вещества разбиваются на две группы (по какому-либо «химиче-

скому признаку»), затем каждая из групп – снова на две, и т.д., пока не будет найдена принадлежность анализируемого компонента к конкретному элементу из множества $\{a_1, a_2, \dots, a_n\}$. Рассмотренная выше система с полными наборами специфических индикаторов $\{b_1, b_2, \dots, b_n\}$ и нейтрализаторов $\{c_1, c_2, \dots, c_n\}$ – одна из наиболее эффективных. По сути дела, она соответствует набору операций поэлементного анализа в различных технологиях, обеспечивая их полноту. Естественно, что для использования в рассматриваемом химическом синтезаторе-анализаторе как индикаторы, так и нейтрализаторы должны быть доступны опять же в виде растворов. Лучше всего – в том же растворителе a_0 (т.е. – в воде).

Для реализации операций определения конкретных оптических характеристик растворов, получающихся в ходе реакций нейтрализации и «маркировки», в состав синтезатора-анализатора необходимо ввести соответствующие датчики (тоже своего рода технологические ячейки). Большинство веществ-индикаторов отмечает присутствие в растворе «незаблокированных» нейтрализатором анализируемых компонентов путём придания ему либо специфичной окраски, либо свойства светиться (флуоресцировать) при облучении, например, ультрафиолетовым светом. В первом случае в состав синтезатора-анализатора следует ввести, например, стабильный источник белого света и три фотодатчика, определяющие интенсивность окраски анализируемого раствора в трёх основных цветах – красном, зеленом и синем, во втором – стабильный источник ультрафиолетового света и простой фотодатчик, измеряющий интенсивность свечения раствора. Обозначим соответствующие операции колориметрического или флуоресцентного анализа символами $P_C(x)$ и $P_F(x)$ соответственно.

Выполнение перечисленных выше условий и наличие в синтезаторе-анализаторе операций $F_d(x)$, $P_C(x)$ (или $P_F(x)$) и $F_S(x_1, x_2)$ гарантирует выполнение в нём, в соответствии с утв. 2.1 и утв. 2.2, всех процессов, связанных с получением (синтезом) различных растворов индикаторов и нейтрализаторов заданных концентраций, а также стан-

дартных аналитических программ количественного и качественного анализа, подобных процессам титрования¹⁴, для любых растворов, содержащих любые вещества из $\mathbf{A}=\{a_0, a_1, a_2, \dots, a_n\}$ в любых допустимых (для данной базы $\mathbf{A}$) концентрациях.

Более того, есть все основания отнести операцию цветоопределения $P_C(x)$ к классу случайных стационарных операций анализа (см. параграф 1.3 или [13, 15, 16]). Это даёт надежду на то, что в перспективе в таком синтезаторе-анализаторе можно будет анализировать не только компоненты, принадлежащие к множеству $\{a_1, a_2, \dots, a_n\}$, но и другие вещества, не относящиеся к элементам выбранной нами базы $\mathbf{A}$. То есть синтезатор-анализатор, снабжённый соответствующей аналитической ячейкой, имеет потенциал к бесконечному расширению числа растворов веществ, для качественного и количественного анализа которых он окажется пригоден. Причём для этого не нужно будет даже менять его структуру – достаточно лишь подобрать подходящие растворы индикаторов и нейтрализаторов и написать соответствующие программы анализа. Единственный ресурс, которого нам может не хватить для этого (как, кстати, и в компьютерах, и в м.т.), – количество доступных свободных ячеек хранения.

Таким образом, возможны два варианта состава технологических операций, обеспечивающих полноту технологии $\mathbf{T}_P$. Их можно записать в следующем виде: $\mathbf{F}_P'=\{F_d(x), F_S(x_1, x_2), P_C(x)\}$ или $\mathbf{F}_P''=\{F_d(x), F_S(x_1, x_2), P_F(x)\}$.

После уточнения состава операций $\mathbf{F}_P$ и компонентов $\mathbf{A}=\{a_0, a_1, a_2, \dots, a_n\}$, а также вспомогательных компонентов $\{b_1, b_2, \dots, b_n\}$ и $\{c_1, c_2, \dots, c_n\}$, выполняющих фактически технологические и аналитические функции, разрабатывается структурная схема и конкретная техническая реализация (т.е. конкретная конструкция) синтезатора-анализатора (в полном соответствии с параграфами 2.1 и 2.2). Пример структурной схемы синтезатора-анализатора, отвечающей утв. 2.1, утв. 2.2 и рис. 2.3 (рис. 2.4), приведён на рис. 2.8.

¹⁴ см., например, Аналитическая химия. Химические методы анализа / Под ред. О.М. Петрухина. – М.: Химия, 1992. – 400 с.

Синтезатор-анализатор (рис. 2.8) содержит однотипные ячейки хранения (ЯХ), т.е. ёмкости четырёх типов: 2, 3, 4 и 5, предназначенные для различных растворов. Например, ЯХ типа 2 могут предназначаться для хранения исходных растворов $\{a_1, a_2, \dots, a_n\}$, аналогичные ячейки типа 3 – резервные, а ячейки типов 4 и 5 предназначены для хранения растворов «маркираторов» $\{b_1, b_2, \dots, b_n\}$ и «нейтрализаторов» $\{c_1, c_2, \dots, c_n\}$ соответственно. Ячейка увеличенного объёма 1 предназначена для хранения растворителя a_0 . Специальные ячейки типа 6 являются ячейками ввода-вывода, т.е. предназначены для пополнения (из внешних источников) запасов различных растворов $\{a_1, a_2, \dots, a_n\}$, $\{b_1, b_2, \dots, b_n\}$, $\{c_1, c_2, \dots, c_n\}$ в синтезаторе-анализаторе или для вывода из него ненужных или загрязнённых растворов.

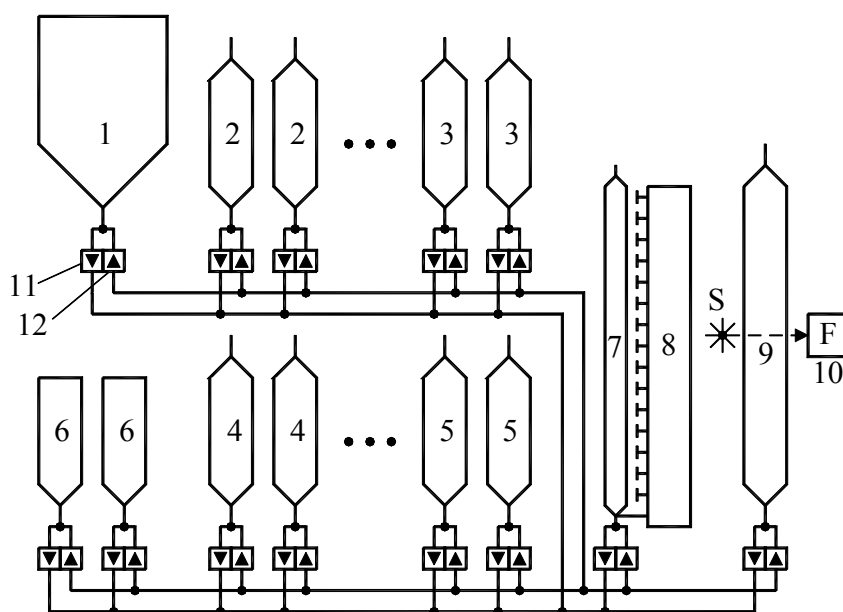


Рис. 2.8. Структура синтезатора-анализатора для растворов

Технологическая ячейка (ТЯ) 7 – дозирующая, выполняющая операцию $F'_d(x, \alpha) = \langle \alpha d, (x - \alpha d) \rangle$, т.е. операцию дозирования с параметром α , позволяющую получать (с помощью системы электрических ёмкостных или иных датчиков 8) различные «дозы» (порции) αd исходного раствора x в зависимости от параметра α , измеряемого набором датчиков 8.

ТЯ 9 – технологическая ячейка, выполняющая операции смешивания $F_S(x_1, x_2)$ и определения оптических характеристик полученных растворов с помощью источника света S и фотодатчиков F (блок 10), т.е. операции анализа типа $P_C(x)$ или $P_F(x)$. Все ячейки (как ЯХ, так и ТЯ) отделены от транспортных магистралей (соединительных трубок) входными (12) и выходными (11) электромеханическими (электромагнитными) вентилями (переключателями потоков жидкостей).

Поскольку вспомогательные «технологические» наборы веществ $\{b_1, b_2, \dots, b_n\}$ и $\{c_1, c_2, \dots, c_n\}$ также представляют собой растворы, то ЯХ для них тоже включены в общую рекурсивную схему технологии T_p , как и ЯХ для элементов базы A . Это необходимо для обеспечения уже упоминавшейся выше возможности получения растворов индикаторов и нейтрализаторов определённых концентраций, а также для выполнения стандартных аналитических подпрограмм количественного и качественного анализа неизвестных растворов с помощью процессов типа титрования. Кроме того, включение соответствующих ЯХ и транспортных магистралей в общий рекурсивный технологический цикл обеспечивает стандартизацию программ и процедур очистки транспортных магистралей и ячеек и позволяет более гибко использовать потенциальные возможности синтезатора-анализатора. Например, временно использовать часть ячеек хранения элементов базы для хранения дополнительных индикаторов и/или нейтрализаторов, и наоборот. Наконец, наличие достаточно большого числа свободных (резервных) ЯХ позволяет в перспективе, как уже отмечалось, *наращивать «синтетические» и «аналитические» технологические возможности синтезатора-анализатора без его конструктивного изменения* подобно тому, как наличие достаточно большого числа свободных ячеек памяти в компьютере позволяет в принципе решать более сложные задачи, чем те, которые имелись в виду при его создании.

Устройство управления для такого синтезатора-анализатора, отвечающее критериям утверждения 2.2, может быть легко выполнено на базе микроконтроллера, как это следует из рис. 2.3. Более того, подпрограммы, реализующие собственную систему команд синтезатора-анализатора, легко разрабатываются по хорошо известным в технологии проектирования компьютеров формальным методам.

Ясно, что если добавить в состав ТЯ такого синтезатора-анализатора технологические ячейки для выполнения других операций, не входящих в множества F_p' или F_p'' , то его технологические возможности только возрастут. Более того, можно, например, существенно расширить область применения подобного синтезатора-анализатора, добавив в его структуру необходимый набор ячеек (включая ТЯ и ЯХ) и соответствующие транспортные магистрали для газообразных веществ. Несколько более сложными получаются структура и техническая реализация для химического синтезатора-анализатора, способного работать ещё и с твёрдыми (сыпучими) веществами. Тем не менее такие структуры и соответствующие конструкции вполне реализуемы при современном уровне развития промышленности¹⁵. Потенциальные сферы их применения огромны. Например, это могут быть гибкие малотоннажные химические производства, выпускающие по мере появления потребностей нужные химические вещества в небольших объёмах.

При соответствующем подборе технологических ячеек (т.е. для подходящей полной Т) такие синтезаторы-анализаторы могут использоваться и в кулинарии (с не слишком изощрённым на первых порах меню), и в сфере автоматизации различных научных исследований, касающихся химии, электрохимии, биохимии, микробиохимии и других смежных областей¹⁶. Для реализации подобных устройств существует даже подходящая «элементная база» – наборы малогабаритных технологических модулей типа модулей системы «СПЛАВ», разработанных в Институте биологической физики РАН¹⁷, или модулей типа Africa фирмы Syrris¹⁸. Важнейшее отличие таких многоцелевых синтезаторов-

¹⁵ «Устройство для синтеза и анализа различных веществ». Заявка на изобретение № 1838293/18-24 от 16 октября 1972г., автор – С.М. Крылов, заявитель – Куйбышевский политехнический институт (сейчас – Самарский государственный технический университет).

¹⁶ Крылов С.М. Синтезатор – химическая скатерть-самобранка // Техника-молодёжи. – 1990. – № 4. – С. 10-12.

¹⁷ Крылов С.М., Шмигельская Н.М. Настольный лабораторный комплекс на основе локальной микросети и малогабаритных модулей системы «СПЛАВ» // Компьютерные технологии в науке, практике и образовании: труды десятой Всерос. межвуз. науч.-практ. конференции. – Самара: СамГТУ, 2011. – С. 51-53.

¹⁸ [Http://www.syrris.com](http://www.syrris.com)

анализаторов от автоматизированных систем фирмы Syrris на основе тех же модулей Africa заключается в том, что их не нужно будет переконфигурировать, перестраивать, поскольку вся необходимая «перестройка» будет заключаться в том же, в чём она заключается для компьютеров, т.е. в написании и отладке новой программы, решающей нужную задачу. Кстати, число команд, необходимых для управления таким синтезатором-анализатором, невелико - по крайней мере, если его строить на основе модулей типа «СПЛАВ»¹⁹.

Существование различных алгоритмов поискового (типа используемого в «теореме об эффективности накопленных знаний» – см. параграф 1.3) или исследовательского (типа алгоритмов «познавателя» – см. [13]) характера, а также возможность реализации в рамках ОФТ процессов автоматического синтеза новых функциональных систем [15] гарантируют эффективное выполнение в таких системах различных научных исследований, в том числе в *полностью автоматическом режиме*.

Ничто не запрещает ввести ещё большую специализацию подобных систем и в других близких к химии областях человеческой деятельности, например, в ботанике, биологии, микробиологии, медицине и т.д.. Сочетание возможности полностью автоматических научных исследований поискового характера с возможностью быстрой практической проверки полученных результатов в самых различных условиях, включая какие-то варианты «клинических тестов», несомненно, будут способствовать ускорению нашего прогресса в этих отраслях науки.

Очевидны и перспективы использования подобных универсальных (или многоцелевых) синтезаторов-анализаторов в дистанционных экспериментальных или учебных лабораториях, доступных через Интернет.

Аналогичные программно-управляемые синтезаторы-анализаторы могут быть выполнены и в микроминиатюрном варианте

¹⁹ Крылов С.М., Шмигельская Н.М. Структура и система команд настольного лабораторного комплекса «СПЛАВ» на основе локальной микросети // Компьютерные технологии в науке, практике и образовании: труды одиннадцатой Междунар. межвуз. науч.-практ. конференции. – Самара: СамГТУ, 2012. – С. 200-202.

с использованием технологий, близких к технологиям изготовления «программируемых лабораторий на кристалле» (ПЛнК или – в англоязычной версии – PLoC) типа AquaCore²⁰. Такие микросинтезаторы-анализаторы в перспективе могут встраиваться в самые различные многофункциональные системы – начиная от дистанционных космических лабораторий и вплоть до вживляемых в организм многопараметрических биомониторов, следящих за его состоянием.

Заметим попутно, что авторы AquaCore так и не смогли решить задачу доказательства универсальности своей системы, которая очень близка по своей структуре и идеологии к структуре синтезатора-анализатора, изображённого на рис. 2.8. Причиной неудачи послужило то, что они попытались найти теоретическое обоснование универсальности AquaCore в классической модели универсальной машины Тьюринга, которая, как мы знаем, предназначена для совершенно иной технологии, а именно для технологии обработки данных. Доказательство универсальности систем типа приведённой на рис. 2.1 для различных технологий, а также формулировка основных требований к ним, были выполнены в параграфе 2.1. Для синтезатора-анализатора с технологией T_p (рис. 2.8) такое доказательство было приведено в начале данного параграфа.

2.5. МИНИ-, МИКРО- И НАНО-ГАП

ГАП – гибкие автоматические производства – задумывались как полностью автоматические заводы, способные быстро перестраиваться с производства одного вида изделий на другой. Поэтому первые версии ГАП мыслились именно как заводы, на которых вместо людей работают роботы. Ясно, что сложность и стоимость таких ГАП даже при современном уровне развития микроэлектроники и роботостроения очень и очень высока. Этим, собственно, и объясняется тот факт, что подобные ГАП до сих пор можно пересчитать по пальцам.

²⁰ Aquacore: a programmable architecture for microfluidics / Amin A.M., Thottethodi M., Vijaykumar T.N., Wereley S., Stephen C., Jacobson S.J. // ACM SIGARCH Computer Architecture News archive. – Vol. 35. – № 2 (May 2007). – P. 254-265.

В базовой модели универсальной системы для синтеза и анализа различных объектов (т.е. различных *изделий*) для различных технологий роботов нет! Это, конечно, не означает, что они в подобных системах не нужны вообще. Но то, что с точки зрения теории они совершенно не обязательны – факт, который может иметь весьма солидное экономическое подкрепление. Действительно, для различных версий химических синтезаторов-анализаторов, рассмотренных в предыдущем параграфе, необходимость применения роботов в соответствующих областях в явном виде не возникала. Можно лишь предполагать, что, возможно, для синтезаторов, так сказать, «кулинарной» ориентации они могут потребоваться для того, чтобы приблизить особенности технологий приготовления некоторых блюд к технологиям, разработанным человечеством за тысячелетия существования кулинарии как своего рода искусства, в котором не только содержание, но и форма имеют значение.

Как же могут выглядеть структуры подобных «безrobotных» ГАП? Естественно, они должны в первую очередь отвечать структуре базовой модели рис. 2.1 и, конечно, структуре, изображённой на рис. 2.3. Эти структуры существенно отличаются от традиционных структур ГАП, построенных, как уже говорилось, на базе концепций «автоматического цеха» или «автоматического завода» с широким использованием роботов и манипуляторов. Единственное, что их объединяет, – это применение для транспортировки обрабатываемых или собираемых деталей и узлов (объектов технологии) так называемых «носителей» (в англоязычной версии «pallets» или «carriers»)²¹. Использование таких носителей связано с тем, что в соответствующих технологических областях (обычно механообработка, машиностроение и т.д.) объекты технологических операций могут иметь самую разнообразную форму и конструкцию. Носители, на которых располагаются обрабатываемые объекты, имеют стандартные размеры и форму и таким образом нивелируют для транспортных систем ГАП индивидуальные особенности таких объектов.

²¹ Flexible Manufacturing Systems. [Http://www.ignou.ac.in/upload/UNIT6-55.pdf](http://www.ignou.ac.in/upload/UNIT6-55.pdf)

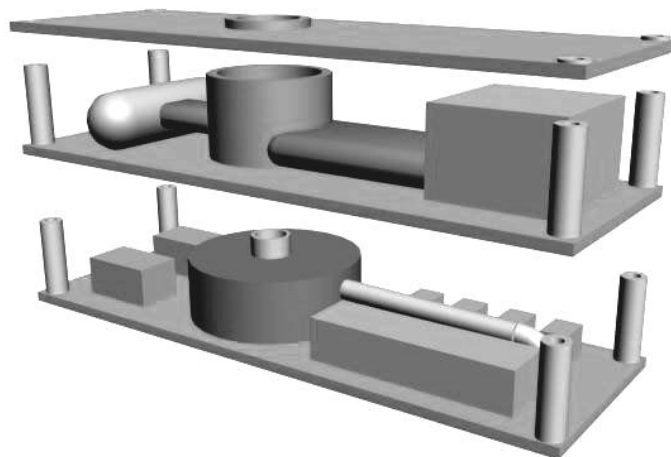


Рис. 2.9. Различные технологические ячейки (ТЯ), устанавливаемые на стандартных платах-носителях

Возможны самые различные практические конфигурации ГАП, соответствующие структурам рис. 2.1 и 2.3. Один из вариантов предполагает, что и ТЯ, и ЯХ для различных объектов технологии также размещаются на стандартных носителях типа поддонов или плат (рис. 2.9). С помощью достаточно простого манипулятора такие носители (а вместе с ними ТЯ и объекты технологий) могут перемещаться между пустыми ячейками каркаса ГАП, выполненного в виде многосекционной этажерки с направляющими для поддонов (рис.2.10).

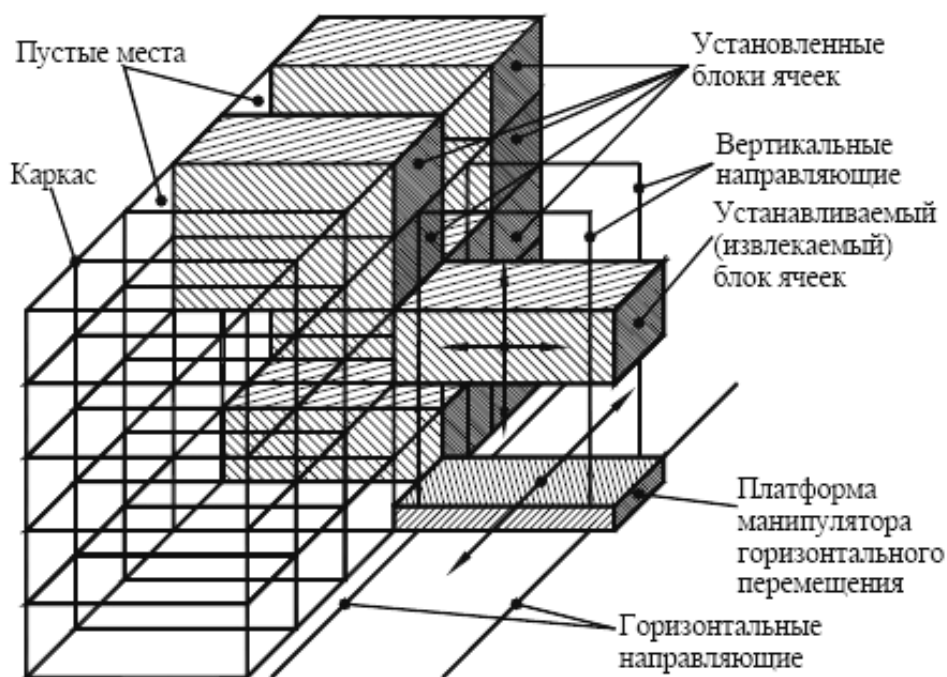


Рис. 2.10. Каркас-основа для компактного ГАП с рекурсивной перестраиваемой структурой из ЯХ и ТЯ, отвечающей структуре, изображённой на рис. 2.3.

Более того, если реализуемая в таком ГАП технология достаточно разнообразна и позволяет создавать компоненты для собственных ТЯ и ЯХ, то подобные ГАП оказываются способны к саморемонту. Ну а при реализации в таком ГАП алгоритмов «изобретения новых гетерогенных функциональных систем» типа рассмотренных в параграфе 7.4 работы [15] они становятся способными и к саморазвитию, самосовершенствованию – *без вмешательства человека!*

Кроме машиностроительных технологий, в рассмотренные выше варианты практической реализации структуру компактных ГАП могут эффективно вписываться и другие технологии – например, как ни странно, технология автоматического выращивания различных растений на тех же носителях-поддонах. По мере развития растений поддоны с ними перемещаются к различным ТЯ для полива, контроля, освещения светильниками со специальным спектром, для других видов обработки, для сбора урожая, наконец. Разумеется, ГАП такого типа будут стоять недёшево, но, может быть, возможность их автоматической работы и частичная способность к саморемонту окажутся ценнее? Особенно в некоторых критических ситуациях – на других планетах, например. Аналогичное замечание относится и к вариантам ГАП с максимально широким спектром обрабатываемых объектов и выполняемых технологических операций. Совсем не лишним будет иметь «под рукой» на Луне или на Марсе такого рода многопрофильные ГАП, кстати, вместе с такими же «широкопрофильными» химическими синтезаторами-анализаторами.

Другая актуальная задача современной науки и промышленности – переход к нанотехнологии, т.е. к производству наноизделий, собираемых из деталей, размеры которых соизмеримы с размерами атомов. Подобные производства должны быть полностью автоматическими, поскольку оперативное вмешательство человека в технологические процессы, протекающие на атомно-молекулярном уровне, вряд ли целесообразно и, кроме того, трудно осуществимо.

С точки зрения соизмеримости масштабов производственных систем и производимых на них наноизделий удобнее всего к нанотехно-

логии подойти не со стороны привычных нам макротехнологических систем, а со стороны микросистем. Наиболее подходящим (и пока единственным) научно обоснованным подходом к созданию подобных многоцелевых автоматических программно-управляемых нанофабрик является подход, основанный на методах ОФТ, рассмотренных выше в параграфах 2.1, 2.2 и в работе [66].

Так как соответствующие СНК должны быть многоцелевыми, предназначенными для производства достаточно разнообразных наноизделий, для транспортировки таких изделий между различными ячейками (технологическими и хранения) необходимы соответствующие унифицированные микроносители, имеющие стандартные размеры и форму для перемещения их подходящими транспортными системами. В последнее время в различных микросистемах для транспортировки разного рода объектов всё чаще используются электрокинетические технологии, основанные на воздействии на заряженные участки или фрагменты таких объектов электрическими полями.

На рис. 2.11 показана возможная конструкция микроносителя для «многоцелевой программируемой нанофабрики на кристалле» и соответствующей транспортной магистрали (микро-канала) для его перемещения.

Магистраль представляет собой изготовленный с помощью технологии МЭМС (микроэлектромеханических систем) в массиве кристалла кремния микроканал прямоугольной формы с открытым верхним окном и выполненными по всей внутренней поверхности канала секциями из электродов, изолированными (например, двуокисью кремния) как от поверхности канала, так и от всей массы материала кристалла. При направленной сверху вниз силе тяжести и при подаче на эти электроды и сам микроноситель электрического потенциала одного знака микроноситель вследствие действия сил электростатического отталкивания приподнимется над поверхностью микроканала и при наличии разности потенциалов на соседних секциях электродов

канала начнёт перемещаться в сторону секции с меньшим потенциалом (см. разделы 6.7 в [13, 15]).

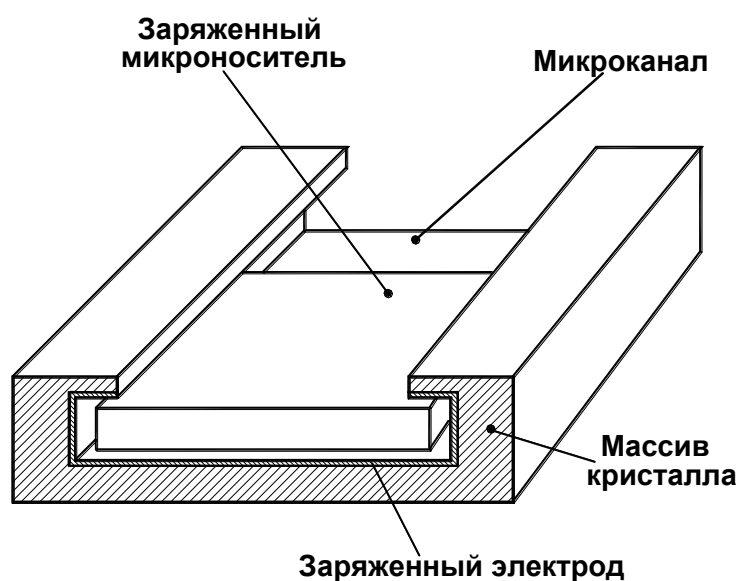


Рис. 2.11. Микроканал с микроносителем

Анализ возможного состава технологических и аналитических операций, обеспечивающих полноту нанотехнологии для такой нанофабрики, показывает, что в состав таких операций могли бы войти: 1) операции туннельной микроскопии, позволяющие определять форму нанообъектов (т.е. операции анализа типа $P_u(x)$ или P_{Ri} для случайного стационарного отображения или аналогичная операция P_A определения формы объектов из утверждений параграфа 1.3); 2) операции нанолитографии (т.е. операции типа F_S , F_d – см. параграфы 1.2, 1.3); 3) разнообразные микроэлектрохимические технологические операции (типа $F_t(x, a_i)$, $F_k(x)$ – см. параграф 1.3).

Теоретически все эти операции могут выполняться с использованием электростатически управляемого микроманипулятора, названного «многофункциональной технологической ячейкой» (МТЯ), и так называемого «капельного процессора» [13, 15].

На рис. 2.12 приведена возможная топология технологической части такой многоцелевой программируемой нанофабрики. Её особенностью является кольцевая транспортная магистраль, обеспечивающая рекурсивную организацию процессов, повышающая надёжность и

увеличивающая скорость транспортировки микроносителей между отдельными ячейками [13, 15].

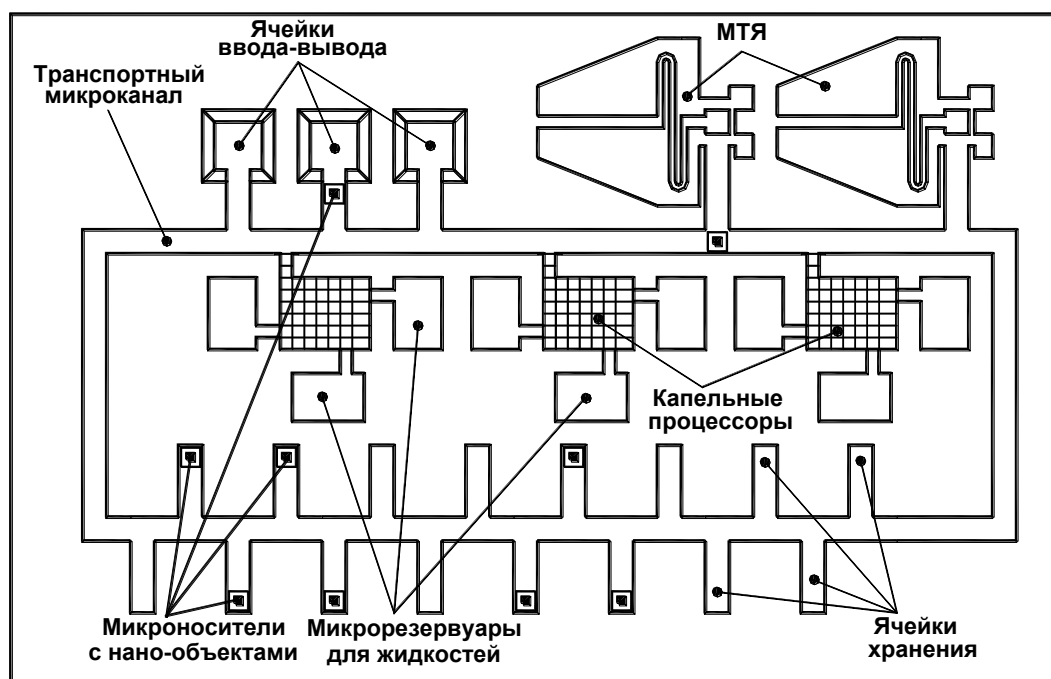


Рис. 2.12. Топология технологической части гипотетической нанофабрики на кристалле, имеющей рекурсивную структуру и содержащей 3 устройства ввода-вывода микрообъектов, 2 многофункциональных технологических ячейки на основе микроманипуляторов, 3 капельных процессора и 16 ячеек хранения микроносителей с микро- или нанообъектами

Возможны и другие варианты реализации подобного рода микро- и нанофабрик. Одно из направлений предполагает применение так называемых 3D-принтеров для создания микро- и нанообъектов, т.е. устройств, в которых в качестве основы алгоритма синтеза любой 3-мерной структуры используется её послойная развёртка [13, 15, 16]. При этом сам объект синтезируется из микрочастиц (часто – микро-капель) различных материалов, которые после нанесения на очередную слой тем или иным способом спекаются, сплавляются, склеиваются или полимеризуются.

При использовании этого способа возникает ряд серьёзных технических проблем, в первую очередь, связанных с конечным и достаточно большим размером исходных микрочастиц, которые существенно превышают размеры атомов и молекул, что для нанотехноло-

гии неприемлемо. Даже если предположить, что технологии будущего смогут справиться с этой проблемой, т.е. смогут использовать микрочастицы с размерами, близкими к размерам молекул, возникает другая сопряжённая проблема – скорость нанесения нужного слоя микрочастиц в таких системах становится низкой. В итоге время изготовления объекта, содержащего наноструктуры, но имеющего более или менее осязаемые размеры (порядка нескольких миллиметров или сантиметров), может достигать часов и даже суток [15]. Тем не менее существуют пути решения этой проблемы за счет совмещения в одном устройстве различных способов синтеза исходных объектов с различным пространственным разрешением, обеспечивающие в итоге вполне приемлемую скорость синтеза объекта в целом²². Эти же подходы могут использоваться и в компактных ГАП со структурой, соответствующей рис. 2.1 и рис. 2.3.

²² Крылов С.М. Руководство по конструированию вселенных. Алгоритмы и технологии. – М.: Изд-во ЛКИ (URSS), 2014. – 208 с.

Глава 3. ФУНКЦИОНАЛЬНОСТИ ОБЪЕКТОВ

3.1. СВЯЗЬ МЕЖДУ СВОЙСТВАМИ И ФУНКЦИОНАЛЬНОСТЯМИ

Что же даёт учёт физических свойств объектов технологий? Рассмотрим возможные подходы к формальному определению самих свойств объектов и связанных с этими свойствами различных особенностей поведения объектов, которые мы в дальнейшем будем называть функциональностями.

Сразу подчеркнем, что трактовка смысла термина «свойство объекта» в каждом конкретном случае применительно к каждому варианту представления этих объектов, их компонентов и отношений между компонентами – один из самых сложных моментов в общей теории систем (ОТС). Довольно часто соответствующая трактовка связана с контекстом излагаемого материала и может интерпретироваться различными авторами по-разному. Например, говоря о свойствах отдельных объектов и систем объектов, обычно не подчеркивают принципиальную разницу в свойствах, связанных с их поведением – динамикой, и в свойствах, как бы статически связанных с самими объектами, т.е. таких, которые позволяют нам отличать одни объекты от других независимо от их динамики. Именно такого рода свойства и имеются в виду в формальном представлении объектов технологий в вариантах их интерпретации в виде (1.3) и (1.5), приведённых в первой главе.

Сложность формализации и интерпретации понятия «свойства» подтверждает также обилие философских подходов и дискуссий на эту тему²³. Между тем очевидно, что в самых различных возможных смыслах этого термина многие свойства различных технологий (и соответствующих алгоритмических технологических систем) во многом

²³ Swoyer C. Properties. In: Stanford Encyclopedia of Philosophy. – Stanford University, 1999. [Http://www.plato.stanford.edu/](http://www.plato.stanford.edu/)

определяются свойствами именно объектов, с которыми они работают, т.е. свойствами элементов базы и конструкций из них (см. параграфы 1.1, 1.2, 1.3). В работах [13, 15] рассмотрены некоторые существующие формальные подходы к этой проблеме, предложенные рядом авторов [9, 49, 50, 81, 84].

Наиболее близким к вариантам типа (1.3), (1.5) оказался подход, предложенный Е.М. Карповым. В работе [9] он пишет: «...представляет интерес разработка специфической алгебраической системы, элементами которой являются двухкомпонентные числопонятийные объекты, в которых каждая компонента по своей сущности отличается от другой. В качестве таких объектов... выступают свойства материальных предметов, алгебраическая сущность которых совпадает с общепризнанным представлением, что любой *«материальный объект – это есть совокупность свойств, которыми этот объект и выделяется, отличается от других объектов»* [9, с. 22]. Далее Е.М. Карпов вводит формальное представление любого объекта A в следующем виде:

$$A = \{a_A, b_A, ..., q_A, ..., x_A, ... \}, \quad (3.1)$$

где $a_A, b_A, ..., q_A, ..., x_A$ – *различные свойства* объекта A .

В виде (3.1) представление объекта в концепции Е.М. Карпова практически совпадает с представлением свойств объекта в ООП и с представлением свойств в форме (1.5) в ОФТ.

Таким образом, имеет смысл ввести следующее:

Определение свойств объекта. Свойство объекта (элемента или конструкции) – это некоторая числовая, геометрическая, физическая или любая другая, в том числе воображаемая, его характеристика, стабильная при определенных стабильных условиях, по которой (по которым) они (объекты, элементы базы, конструкции) *могут отличаться друг от друга* или от своего предшественного во времени и/или пространстве состояния.

Подчеркнем, что важным отличием данного подхода от других, часто рассматриваемых в научной литературе, является отсутствие

деления свойств объектов на атрибуты (качества) и собственно свойства (характеристики конкретных объектов). Это связано с тем, что введенное нашим определением понятие свойства – сугубо техническое, связанное, во-первых, с необходимостью более точно определить смысл некоторых аналитических операций (предикатов) формально-технологических систем, и, во-вторых, с необходимостью ввести общее понятие функциональности объектов. По этим причинам мы также не станем обсуждать возможные философские последствия данного подхода. Наша цель – показать, что с помощью данного определения общая концепция функциональности объектов не только естественным образом вводится и определяется, но и даёт весьма интересные результаты.

Нетрудно видеть, что под это определение подпадают практически любые достаточно стабильные свойства или любые статические характеристики объектов, в том числе и такие, которые мы не привыкли ассоциировать с понятием «свойство» в обычном, традиционном понимании этого термина. Например, свойство «быть инерционным объектом», то есть сохранять в (пустом) пространстве определенное направление движения, или свойства «иметь определенную форму», «иметь определенную скорость вращения», или «иметь определенное положение в пространстве» и т.д. На это обстоятельство следует обращать внимание во всех последующих случаях применения указанного термина. Таким образом, здесь, как и при определении эволюционности технологий (см. [13-16] или параграф 1.3), указанный термин используется в рамках, введенных соответствующим определением, которые могут значительно отличаться от традиционной интерпретации.

Важным моментом в данном определении являются также слова *«могут отличаться...»*. Это сразу приводит к возможности существования как одинаковых свойств у разных объектов, так и *в действительности* разных объектов с одинаковыми наборами свойств. Последнее в нашей интерпретации будет просто означать, что список свойств таких объектов не закончен. То есть некоторые свойства объ-

ектов, благодаря которым они согласно определению на самом деле отличаются друг от друга (например, своим положением в пространстве), могут быть обнаружены (или не обнаружены!) в последующих экспериментах (или просто при расширении рамок рассматриваемого списка свойств). Естественно, что если список свойств двух (или нескольких) объектов исчерпывающий и все их свойства одинаковы, то мы имеем дело с одинаковыми (идентичными) объектами.

Требование некоторой стабильности свойств (во временном или любом другом смысле) прямо вытекает из их предназначения – служить мерой отличия одного объекта от другого. Говорить о свойствах, которые появляются и исчезают «случайно», можно лишь с позиций, предполагающих наше незнание условий сохранения их стабильности. Это важное замечание потребуется нам в дальнейшем при сравнительном анализе свойств, введенных данным определением, и «динамических свойств», связанных с «поведением» объектов и конструкций из них.

Наличие свойств объектов не обязательно предполагает наличие среди операций соответствующей технологии операций анализа, напрямую определяющих «присутствие» или «отсутствие» тех или иных свойств и даже «измеряющих» их величину, хотя такие возможности тоже оказываются не лишними (см. параграфы 1.2, 1.3). Свойства могут присутствовать, так сказать, по умолчанию. При этом среди аналитических операций технологии может не быть ни одной, которая способна напрямую определять или измерять эти свойства. Например, объекты могут обладать или не обладать свойством «быть инерционным объектом», что может оказаться чрезвычайно важным для саморазвития (эволюции) технологии²⁴ (см. также [13-16]). Вместе с тем в самой технологии могут отсутствовать операции анализа, напрямую определяющие наличие или отсутствие этого свойства у объектов.

²⁴ Крылов С.М. Руководство по конструированию вселенных. Алгоритмы и технологии. – М.: Изд-во ЛКИ (URSS), 2014. – 208 с.

Следует также обратить внимание на первое слово в перечислении возможных «типов» свойств, а именно на возможность «числового», т.е. чисто количественного, негеометрического и нефизического, свойства, причем именно свойства, а не числового параметра, представляющего какое-либо геометрическое или физическое свойство. Этому вопросу, т.е. сущности и особенностям таких свойств объектов соответствующих технологий, будет в дальнейшем уделено особое место, поскольку математические объекты характеризуются именно такими нефизическими свойствами. Мы заранее обращаем внимание на то, что в рамках ОФТ свойство элемента (конструкции) представлять некоторое количество в его, так сказать, «чистом» виде, безотносительно к другим, геометрическим, физическим и прочим свойствам, также рассматривается как некое самостоятельное специфическое (нефизическое) свойство. Кроме того, для удобства работы и с другими, не только числовыми свойствами, мы (как это отмечено в определении свойств) будем допускать их отображение на числовую ось в виде значения параметра, определяющего конкретную величину «свойства γ » объекта x . Такой параметр может принимать не только одно из двух значений типа: «присутствует» – «отсутствует» (или, как в логике, «истина – ложь»), но и некоторую величину из заданного диапазона натуральных чисел: $1, 2, \dots, n$, или из заданного интервала действительных чисел $[-R, +R]$ или $[R_1, R_2]$. То есть вполне правомерной может быть запись двух различных свойств γ_1 и γ_2 в следующем виде: $\gamma_1 = 7$, или $\gamma_2 = -2,35$. Это очень важное отличие данной концепции от свойств, связываемых во многих математических представлениях с так называемой представительной функцией предиката, соответствующего тому или иному свойству. Такая представительная функция принимает только два значения: «истина» – на тех объектах, которые обладают данным свойством, и «ложь» – на всех остальных. В нашей же концепции помимо качественного формата представления свойств (присутствует – отсутствует) может иметь место еще и количественный формат. Тем не менее во всех случаях следует иметь

в виду, что в данном представлении числа играют лишь роль своеобразной шкалы, на которую *отображаются* реальные (или предполагаемые абстрактные) свойства. Такое отображение может быть верным или неверным в зависимости от степени нашего понимания или непонимания соответствующей проблемы, но почти всегда остается лишь отображением, а не самостоятельным фактором, за тем редким исключением, когда речь идет о свойстве представлять некоторое количество. Здесь и только здесь мы будем иметь дело с тем уникальным случаем, когда само свойство представляется взаимно однозначным (один к одному) отображением на себя!

Еще раз подчеркнем, что проблема формального описания свойств объектов с философской точки зрения, да и сама по себе – безотносительно к философии – достаточно сложна. Однако рассмотренного выше весьма простого, возможно, грубого и почти очевидного подхода оказывается вполне достаточно для того, чтобы в рамках ОФТ исследовать многие близкие к реальным модели технологических систем, компоненты которых обладают определенными свойствами, и делать на основе этих исследований далеко не тривиальные и неочевидные выводы.

Таким образом, суть нашего подхода можно вкратце охарактеризовать так: любые параметры объектов, которыми они могут отличаться друг от друга, которые нам интересны или которые каким-то образом влияют на их поведение, мы относим к свойствам. Очевидно, чем полнее список этих свойств (или представляющих их параметров), чем лучше он согласуется с реальным и предсказываемым поведением объектов – тем лучше, тем правильнее сделан наш выбор.

С другой стороны, в такой нотации становится вполне очевидной возможность ограничения реального (т.е. достаточно полного) списка свойств объектов тем его сокращенным вариантом, который представляет полноценную картину их поведения в конкретном контексте, т.е. в конкретной области той научной или практической деятельности, где мы проводим формально-технологический анализ соответствующих технологий (социальных, биохимических, физических и т.д.).

В этой связи уже на данном этапе представляется полезным воспользоваться некоторыми важными результатами в плане систематизации и формализации описания свойств, которые выработаны к настоящему времени одной из самых быстро развивающихся технологий нашего времени – технологией программирования. За примерно полувековой период своего стремительного эволюционного развития она сумела выработать универсальные принципы описания любых типов объектов, будь то различные графические объекты, демонстрируемые на экране дисплея, большие массивы данных или даже сами программы. Это – принципы ООП.

Подчеркнем, что породившие программирование теория алгоритмов, а в более глубоком историческом аспекте (т.е. начиная с алгоритмов сложения и умножения) – и сама математика за более чем тысячелетнюю эволюцию не смогли сделать ничего подобного. Тем не менее использование принципов ООП для формального представления свойств объектов могло бы помочь прояснить ситуацию в том числе и в многочисленных дискуссиях о роли и месте самой математики в общей системе наук, о её границах и возможностях и т.д. (см., например, [32, 50, 52, 62, 72]). Эти универсальные принципы, как уже отмечалось, в развёрнутом варианте называются объектно-ориентированным программированием или объектно-ориентированной парадигмой.

Есть разные трактовки сущности и особенностей ООП, которые связаны чаще всего с конкретными традициями и языками программирования. Для нас же в этой концепции наиболее важно её самое главное, центральное звено, а именно – любой объект должен представляться единой совокупностью как параметров, описывающих его особенности («свойства»!), так и методов (способов, или функций), описывающих использование (взаимодействие) параметров («свойств»!) как внутри объекта, так и вне его, т.е. с другими объектами. Действительно, любой реальный объект реального мира – это не просто некоторый набор его статических характеристик. Это еще и *процесс его существования в окружающем мире*, его взаимодействие с другими объектами, его *динамика*, его *эволюция*. Лишать объект как всего многообразия его

свойств, так и его динамики – как это делает, например, арифметика, – хотя и полезно в плане предельного упрощения основных соотношений между конструкциями (объектами) соответствующей технологии (т.е. в арифметике – между числами), но одновременно чревато и потерей всей его реалистической сущности.

Таким образом, в дальнейшем мы будем считать, что любой объект O_i в любой технологии может быть представлен *математической моделью* в виде совокупности двух множеств:

$$O_i = \langle \Gamma_i, \mathbf{M}_i \rangle = \langle \{\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in}\}, \{\gamma_{ij} = \varphi_{ij}(\gamma_{sk}, \gamma_{sm}, \dots, \gamma_{ir}); \dots\} \rangle; \quad (3.2)$$

где, напомним, O_i – i -тый физический объект; Γ_i – множество (список) интересующих нас его физических свойств (параметров) типа γ_{ir} , характеризующих объект O_i по важным для нас признакам – по его положению в пространстве, по его массе, скорости и т.д. и т.п.; $\mathbf{M}_i$ – множество (перечень) интересующих нас функциональностей φ_{ij} объекта O_i , описывающих его взаимодействие с пространством или с другими физическими объектами типа O_s через их физические свойства $\gamma_{sk}, \gamma_{sm}, \dots$, и (или) собственные физические свойства самого объекта O_i типа γ_{ir} ; n – число параметров («свойств») типа γ_{ir} , используемых в данном представлении объекта O_i , т.е. число вторых индексов свойств объекта O_i с номерами $r \in \{1, \dots, n\}$; k, m – номера (вторые индексы) тех «свойств» и функциональностей, которые актуальны для анализа функциональных взаимодействий φ_{ij} данного объекта O_i (в текущем его представлении) с соседними или удалёнными (при взаимодействии, например, через поле) объектами типа O_s . Функциональности (часто называемые «методами» в ООП) будут более подробно рассмотрены далее. Пока же отметим, что такое унифицированное математическое представление объектов легко позволяет переносить их практически в любую поддерживающую ООП компьютерную среду для проведения различных компьютерных экспериментов, испытаний и аналитических расчетов. Более того, из последующих примеров будет ясно, что такой формальный анализ свойств и функциональностей объектов в стиле ООП оказывает

ся полезным не только при компьютерном моделировании их поведения в тех или иных технологиях, но может рассматриваться и как самостоятельный, весьма эффективный метод самой формальной технологии.

В зависимости от типа и характера объектов списки (т.е. множества) параметров, представляющих свойства, могут иметь разную длину и различный состав. При этом можно считать, что разные свойства как бы *ортогональны друг другу*, т.е. являются разнотипными, что с формальной точки зрения часто и имеет место на самом деле.

3.2. ФУНКЦИОНАЛЬНОСТИ КАК ИЗМЕНЕНИЯ СОСТОЯНИЙ

Функциональность – термин, часто встречающийся в работах, связанных с общей теорией систем, но редко определяемый в них формально или хотя бы объясняемый достаточно детально. В то же время в математике существует целое направление, посвященное так называемой «функциональной полноте» систем различных функций. В параграфе 1.3 мы уже исследовали это понятие на предмет его близости к понятию «полноты технологий» в ОФТ. Здесь же нас будет интересовать несколько другое, а именно возможное функциональное поведение различных объектов различных технологий, связанное со свойствами этих объектов, как мы их определили в предыдущем параграфе. То есть нас будет интересовать поведение объектов, определяемое теми их свойствами, которые зависят от каких-то других свойств данного объекта или других объектов – в полном соответствии с формальными представлениями объектов (1.3) и (3.2). Правильный ответ на возникающие при этом вопросы может прояснить многие не совсем ясные места как в функционировании различных технологических систем, так и в эволюции живой природы, то есть в функционировании больших наборов сложных конструкций из биологически активных молекул.

Как уже отмечалось, одна из первых попыток ввести некоторое единообразие в описание функциональности (точнее взаимодействия) различных физических объектов была предпринята Е.М. Карповым в [9]. Суть его подхода заключается во введении специальной операции

реального физического взаимодействия ψ активной сущности одного объекта с пассивной сущностью другого, в результате которой происходит некоторое изменение в структуре свойств, описывающих объекты [9, с. 23].

К похожим концепциям можно подойти и на основе анализа нерешённых задач в теории свойств М. Бунге [49]. Так, в работе [81] подчёркивается, что одним из существенных недостатков этой теории явилось отсутствие математического аппарата, описывающего изменение состояния объектов, т.е. описывающего их *функциональность*. Покажем, как такого рода функциональность может быть естественным образом сформулирована во введённой выше формально-технологической концепции свойств. Для этого нам необходимо формальное определение состояния объекта.

Определение состояния объекта. Состоянием объекта (т.е. элемента базы или конструкции) называется совокупное конкретное значение всех его свойств.

Фактически (имея в виду договоренность представлять свойства в виде списка параметров) из этого определения следует, что состояние — это конкретное положение вектора свойств данного объекта по всем его «координатам» (по всем «взаимно ортогональным» типам присущих ему свойств, т.е. по всем компонентам существующего списка параметров — как числовым, так и нечисловым).

Такой список может быть представлен в различных, наиболее удобных для конкретного случая формах. Например, если список содержит m компонентов, каждый из которых может принимать независимо от других только одно из двух значений 0 или 1, то общее число состояний соответствующего объекта будет равно 2^m . Все их можно интерпретировать как соответствующее значение *одного* параметра, способного принимать любое из этих 2^m значений. То есть можно — для удобства! — совокупность из m независимых (и ортогональных!) свойств представить некоторым одним сложным (комплексным) свойством с соответствующим (сложным) параметром. И наоборот, при наличии параметра, способного принимать большое число различных значений, попытаться «разложить» его на более простые (элементарные) составляющие. Этот

подход, кстати, положен в основу одного очень эффективного метода ОФТ, который позволяет конструировать и минимизировать нужные свойства и функциональности объектов для обеспечения необходимого их взаимного поведения (динамики) [15].

Заметим сразу, что при математическом абстрагировании подобным образом поступают довольно часто, т.е. проецируют всю гамму различных реальных (физических) ортогональных свойств объектов на однородную (одномерную!) числовую шкалу. В ряде случаев это позволяет значительно снизить размерность задачи и тем самым облегчить поиск её числового (т.е. математического) решения. ***Однако при таком отображении разнородных физических свойств объектов на однородную числовую шкалу может легко нарушиться очень важное условие, обеспечивающее адекватность математической модели и её физического прототипа, а именно – может пропасть изоморфность (взаимная однозначность) между значимыми (ортогональными!) свойствами исходного физического объекта и его упрощённой математической модели.***

С другой стороны, важной особенностью любого элемента или конструкции является их возможность реагировать на изменения, происходящие во внешней (относительно объекта-конструкции) среде, включая соседние объекты, собственно, и образующие эту самую внешнюю среду. Естественно, что подобного рода изменения имеет смысл связать с функциональностью объекта-конструкции, т.е. с тем, что определяет его самые различные динамики – от «тривиального» перемещения в пространстве до обработки информации в компьютерах и развития социальных систем. При подобных изменениях, естественно, должно меняться нечто и в системе описания самого объекта-конструкции, точнее в том, посредством чего мы отличаем данную конкретную конструкцию от других. Поскольку мы только что определили всё то, чем одна конструкция (объект) может отличаться от другой, как «свойства», то, естественно, эти изменения должны менять положение вектора свойств в пространстве свойств, присущих данной

конструкции, иначе мы просто не сможем обнаружить эти изменения. В свою очередь, изменения вектора свойств одной конструкции может через соответствующие функциональные механизмы повлечь изменение вектора свойств другой конструкции, «физически связанной» с первой (и с её свойствами). Таким образом, можно сформулировать простое и вместе с тем всеохватывающее определение функциональностей объектов через их свойства.

Определение функциональных свойств. Свойство объекта (элемента или конструкции) называется функциональным, если оно позволяет изменять состояние данного объекта и (или) другого объекта, с которым взаимодействует это свойство, при их физическом сближении или соединении.

Иными словами, некоторое свойство γ_{ij} объекта $O_i = \langle \Gamma_i, \mathbf{M}_i \rangle$, $\gamma_{ij} \in \Gamma_i$, называется функциональным, если среди функциональностей $\mathbf{M}_i$ самого объекта O_i и (или) среди функциональностей $\mathbf{M}_k$ других объектов O_k найдутся такие, в записи которых свойство γ_{ij} входит в качестве аргумента, например: $\gamma_{ir} = \varphi_r(\gamma_{it}, \dots, \gamma_{ij}, \dots, \gamma_{ml})$ или $\gamma_{ks} = \varphi_s(\gamma_{kt}, \dots, \gamma_{ij}, \dots, \gamma_{kh})$.

Требование физического сближения или соединения функционально взаимодействующих объектов связано с тем обстоятельством, что в реальных системах (т.е. в различных конструкциях, состоящих из элементов-объектов) взаимодействие компонентов системы происходит в основном как раз тогда, когда эти элементы находятся либо достаточно близко друг к другу (и взаимодействие происходит через поля), либо в прямом физическом контакте – например, оптическом, электрическом, механическом и т.д. (более подробно этот вопрос рассмотрен чуть ниже).

Необходимо сразу же оговорить возможность взаимодействия объектов без «заметного» изменения каких бы то ни было их свойств, т.е. без явного изменения положения их векторов свойств. Такие свойства, согласно определению функциональных свойств, вроде бы не являются функциональными, но тем не менее могут порождать не-

которое реальное и видимое изменение – например, медленное перемещение небольших и достаточно удаленных друг от друга материальных объектов в пустом пространстве под действием взаимного притяжения (гравитация). Эти свойства можно назвать слабофункциональными, чтобы не выходить за рамки интуитивного смысла термина «функциональность», означающего всё же наличие некоторой изменчивости, динамики в состоянии или поведении объектов. Строго говоря, гравитация при длительном своём действии может существенно изменить некоторые параметры даже небольших взаимодействующих объектов, например, скорость их взаимного сближения. Поэтому при определённых условиях (при больших временах взаимодействия или больших взаимодействующих массах) даже слабое гравитационное взаимодействие нормальным образом переходит в класс «традиционных» функциональностей, подпадающих под определение функциональных свойств.

Теперь мы покажем, что двоичная логика электронных логических схем вполне вписывается в концепцию функциональности, введенную соответствующим определением.

Действительно, все логические элементы в таких схемах находятся в определённых физических (конкретно – электрических) соединениях друг с другом. С другой стороны, конкретное значение выходного сигнала (конкретное «значение свойства» любого двоичного логического элемента иметь на своем выходе один из логических уровней «0» или «1»), т.е. состояние (выхода) логического элемента определяется значениями всех его логических входов. А они, в свою очередь, определяются конкретными значениями состояний (конкретными «значениями параметров, представляющих свойства») логических выходов других – связанных с данным – логических элементов. Изменение значений этих входов изменяет и состояние самого логического элемента (с учетом задержки распространения сигналов по

его внутренним схемам), что и требуется для любой функциональности согласно её определению.

Рассмотрим для наглядности формализованное описание вектора свойств и функциональности логического элемента типа «2И-НЕ» (рис. 3.1), который обозначим как элемент E_1 .

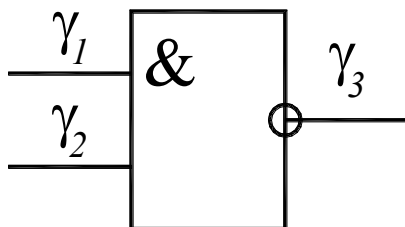


Рис. 3.1. «Свойства» абстрактного элемента «2И-НЕ»

Сразу отметим, что в данном примере нас будут интересовать только свойства и функции самого элемента как одного из представителей двухвходовых логических элементов вообще, а не его конкретные свойства и функции в какой-либо схеме, в которой важную роль играет ещё и топология (то есть «геометрическое» положение элемента и его связь с другими элементами схемы).

Итак, элемент E_1 в интересующем нас аспекте (а именно как один из возможных представителей класса двухвходовых двоичных логических элементов, а не как его конкретная техническая реализация на электронных компонентах определённого типа) может быть представлен следующим образом:

$$E_1 = \langle \Gamma_1, \mathbf{M}_1 \rangle = \langle \{\gamma_1, \gamma_2, \gamma_3\}, \{\gamma_3 = \varphi_3(\gamma_1, \gamma_2)\} \rangle, \quad (3.3)$$

где в списке (векторе) свойств Γ_1 указано только три интересующих нас в данном случае свойства (представленные соответствующими параметрами): $\Gamma_1 = \{\gamma_1, \gamma_2, \gamma_3\}$, причём все компоненты вектора Γ_1 отображаются только на числа «0» и «1»: $\gamma_i \in \{0, 1\}$, а в списке методов $\mathbf{M}_1$ имеется только один метод: $\mathbf{M}_1 = \{\gamma_3 = \varphi_3(\gamma_1, \gamma_2)\} = \{\gamma_3 = \neg(\gamma_1 \& \gamma_2)\}$, где $\neg$ – символ инверсии, $\&$ – операция «И» (логическое умножение).

В этом примере «свойства» γ_1 , γ_2 выступают именно как функциональные свойства (условно их можно назвать «входными»), значения которых определяют значение выходного «свойства» γ_3 . Последнее также может быть функциональным, но только при соединении (контакте) выхода γ_3 с каким-либо логическим входом какого-либо другого логического элемента *или с чем-либо ещё*. При отсутствии такого соединения γ_3 не может считаться функциональным свойством – в полном соответствии с нашим определением. То есть при наличии *физического* соединения выхода элемента E_1 с входами других элементов свойство γ_3 автоматически становится функциональным, поскольку оказывается «входным» для соответствующих элементов.

Теоретически для двоичного логического элемента с двумя логическими входами и одним выходом возможны 8 различных состояний, однако ввиду ограничений, накладываемых функциональностью («методом») φ_3 на взаимосвязь γ_3 с γ_1 и γ_2 при правильном функционировании E_1 реализуются только 4 из них. Остальные 4 являются запрещёнными для E_1 и могут трактоваться как аварийные или неисправные состояния.

В приведённом примере несколько необычным для традиционных рамок рассмотрения логических элементов является трактовка двух входных сигналов γ_1 и γ_2 , описывающих состояния соответствующих входов элемента E_1 в качестве компонентов списка свойств E_1 . Однако ни с точки зрения ООП, ни с позиций концепции Е.М. Карпова, ни, тем более, с позиций ОФТ-определений свойств, их состояний и их функциональностей ничего противоречивого в такой трактовке нет. И вот почему: любой подобный элемент всегда интересуется нас именно как элемент, способный воспринимать двоичные сигналы со своих «входов» и преобразовывать их в одно из двух возможных состояний своего «выхода» в соответствии с реализуемой логической функцией («методом»). То есть *наиболее важные* для нас

свойства такого элемента — это свойства *иметь один из двух возможных логических уровней на каждом из двух своих входов и соответствующее этим входам и реализуемой элементом логической функции состояние на выходе*. Таким образом, при детальном анализе рассмотренная трактовка свойств и функциональностей элемента E_1 действительно не противоречит введённым нами определениям.

Приведённый пример интересен ещё и потому, что он дополнительно подтверждает обоснованность включения в определение свойств возможность существования разных элементов с одинаковыми свойствами. Действительно, согласно формуле (3.3), все абстрактные двухвходовые двоичные логические элементы с одним выходом описываются однотипным набором свойств, однако *могут отличаться друг от друга своими функциональностями («методами»)*.

Теперь попытаемся построить аналогичную «двухвходовую» функциональность в самом общем — формальном — виде, т.е. в соответствии с введённым нами определением функциональности, не привязываясь к какой-либо конкретной функции, логике или системе счисления. Соответствующее формальное функциональное преобразование (*функциональность*), связанное с двумя «входными» (γ_1, γ_2) и одним «выходным» (γ_3) свойствами, входящими в список свойств одного и того же объекта, в самом общем виде можно записать так:

$$\gamma_1 * \gamma_2 \rightarrow \gamma_3, \quad (3.4)$$

причём обратим внимание на то, что сейчас индексы означают не конкретные свойства из списка свойств какого-то объекта O_j , а просто их порядок в модели типа (3.3).

Выражение (3.4) описывает простейшую функциональность, связанную с взаимодействием двух (и только двух) свойств одного и того же объекта, на что указывает символ «*» (понимаемый именно как символ какого-то функционального взаимодействия). Рассмотрим теперь возможные конкретные реализации этого взаимодействия с учё-

том принадлежности взаимодействующих свойств – для начала к одному и тому же объекту. Для этого введём верхние индексы в виде символа j , которые будут означать принадлежность соответствующего свойства к данному (j -тому) объекту O_j , а дополнительными нижними индексами (i, k, s) теперь уже будем обозначать индекс данного свойства в общем списке всех свойств объекта O_j , соответствующих форме (3.2). Формально возможны следующие варианты взаимодействий, соответствующие введённому выше определению функциональности и общему выражению (3.4), описывающему функциональное взаимодействие свойств (γ_1, γ_2) , изменяющих третье (γ_3) :

$$(\gamma_1)^1_i * (\gamma_2)^1_i \rightarrow (\gamma_3)^1_i; \quad (3.5)$$

$$(\gamma_1)^1_i * (\gamma_2)^1_i \rightarrow (\gamma_3)^1_k; \quad (3.6)$$

$$(\gamma_1)^1_i * (\gamma_2)^1_k \rightarrow (\gamma_3)^1_i; \quad (3.7)$$

$$(\gamma_1)^1_i * (\gamma_2)^1_k \rightarrow (\gamma_3)^1_k, \quad (3.8)$$

$$(\gamma_1)^1_i * (\gamma_2)^1_k \rightarrow (\gamma_3)^1_s, \quad (3.9)$$

где индексы i, k, s могут – теоретически – пробегать все возможные типы свойств, а в случае «симметричности» входных аргументов γ_1 и γ_2 выражения (3.7) и (3.8) описывают к тому же один и тот же тип функциональности.

Из (3.5) – (3.9) видно, что функциональность обычной двоичной логики может быть реализована только в рамках (3.5), поскольку только для этой записи все входные и выходные параметры однотипны (т.е. однородны, или *гомогенны*). Все остальные типы функциональностей выходят за её рамки, так как предполагают использование *разнотипных* (т.е. разнородных, *гетерогенных*) свойств (и соответствующих параметров) для своей реализации. Покажем тем не менее, что любые из функциональностей (3.5) – (3.9) обеспечивают – по крайней мере с принципиальной, теоретической точки зрения – выполнение всех необходимых функций обработки и преобразования информации.

Для начала рассмотрим представленную на рис. 3.2 схему так называемого *триггера* – устройства, выполняющего функцию запоминания и хранения информации в компьютерах и других цифровых системах обработки данных.

В схеме имеется четыре «носителя» свойств: объекты a , b , c , d . С каждым из них связываются некоторые свойства и их возможные числовые значения. С объектом a – свойство γ_1 . С объектом b – свойство γ_4 . С объектом c – свойства γ_1 , γ_2 и γ_3 . Причём свойства γ_1 и γ_2 для объекта c являются как бы «входными», т.е. теми, которые приводят к изменению «значения» собственного («выходного») свойства γ_3 . С объектом d связаны свойства γ_2 , γ_3 и γ_4 («входные» свойства – γ_3 и γ_4 , собственное, или «выходное» – γ_2).

Легко проверяется, что схема рис. 3.2 вписывается во многие типы функциональностей из (3.5) – (3.9), причём в каждом из таких типов может быть бесконечно много числовых решений.

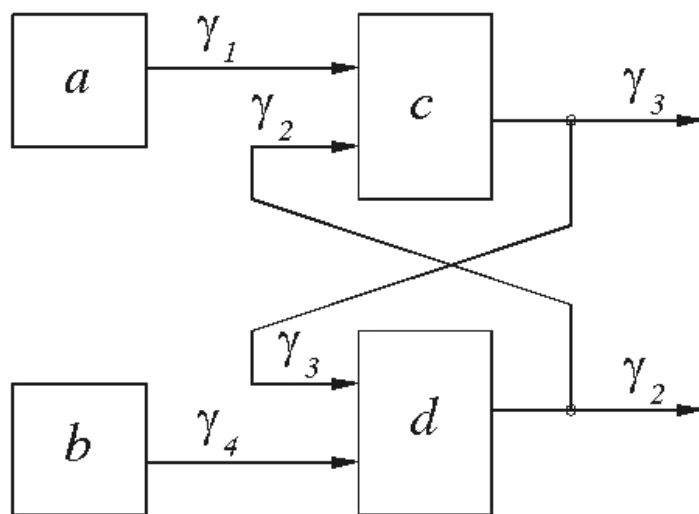


Рис. 3.2. Схема «триггера» на основе функциональных объектов самого общего вида

Действительно, для выражения (3.5) с учётом особенностей схемы рис. 3.2 возможны следующие числовые наборы значений соответствующих свойств (которые будем записывать через знак равенства рядом с переменными, обозначающими свойства):

$$\begin{aligned}
(\gamma_1=1)_i^a * (\gamma_2=1)_i^d &\rightarrow (\gamma_3=2)_i^c, (\gamma_2)_i^d \neq (\gamma_3)_i^c; \\
(\gamma_1=1)_i^a * (\gamma_2=2)_i^d &\rightarrow (\gamma_3=1)_i^c, (\gamma_2)_i^d \neq (\gamma_3)_i^c; \\
(\gamma_1=2)_i^a * (\gamma_2=1)_i^d &\rightarrow (\gamma_3=1)_i^c, (\gamma_2)_i^d = (\gamma_3)_i^c; \\
(\gamma_1=2)_i^a * (\gamma_2=2)_i^d &\rightarrow (\gamma_3=1)_i^c, (\gamma_2)_i^d \neq (\gamma_3)_i^c.
\end{aligned} \tag{3.10}$$

Аналогичную таблицу можно составить и для функциональностей, связанных с объектом d . Кроме того, легко заметить, что характер функциональностей (3.10) не изменится, если все значения γ_i увеличить (уменьшить) одновременно на 1, 2, ..., n . Наконец, нетрудно обнаружить в системе выражений (3.10) слегка завуалированное описание логической функции типа «2ИЛИ-НЕ». Действительно, если из всех значений вычесть единицу, мы получим:

$$\begin{aligned}
(\gamma_1=0)^a * (\gamma_2=0)^d &\rightarrow (\gamma_3=1)^c; \\
(\gamma_1=0)^a * (\gamma_2=1)^d &\rightarrow (\gamma_3=0)^c; \\
(\gamma_1=1)^a * (\gamma_2=0)^d &\rightarrow (\gamma_3=0)^c; \\
(\gamma_1=1)^a * (\gamma_2=1)^d &\rightarrow (\gamma_3=0)^c,
\end{aligned}$$

что соответствует таблице истинности указанной логической функции от двух логических переменных γ_1 и γ_2 . Более того, известно, что эта логическая функция относится к так называемому функционально-полному базису, т.е. с её помощью (и с помощью одной из логических констант «0» или «1») можно построить все остальные логические функции от любого числа переменных.

Нетрудно составить аналогичные таблицы с использованием других типов функциональностей (3.5) – (3.9). Причём в функциональных преобразованиях не обязательно должны участвовать одни и те же свойства объектов a, b, c, d или конкретные значения соответствующих параметров, представляющих такие «параметризованные» свойства. Возможен, к примеру, такой вариант, где некоторые кон-

кретные значения свойств могут лежать в достаточно большом числовом диапазоне:

– для объекта c :

$$\begin{aligned}(\gamma_1 > 0)_i^a * (\gamma_2 > 0)_k^d &\rightarrow (\gamma_3 = 0)_s^c; \\(\gamma_1 = 0)_i^a * (\gamma_2 > 0)_k^d &\rightarrow (\gamma_3 > 0)_s^c; \\(\gamma_1 > 0)_i^a * (\gamma_2 = 0)_k^d &\rightarrow (\gamma_3 > 0)_s^c; \\(\gamma_1 = 0)_i^a * (\gamma_2 = 0)_k^d &\rightarrow (\gamma_3 > 0)_s^c;\end{aligned}$$

– для объекта d :

$$\begin{aligned}(\gamma_4 > 0)_f^b * (\gamma_3 > 0)_s^c &\rightarrow (\gamma_2 = 0)_k^d; \\(\gamma_4 = 0)_f^b * (\gamma_3 > 0)_s^c &\rightarrow (\gamma_2 > 0)_k^d; \\(\gamma_4 > 0)_f^b * (\gamma_3 = 0)_s^c &\rightarrow (\gamma_2 > 0)_k^d; \\(\gamma_4 = 0)_f^b * (\gamma_3 = 0)_s^c &\rightarrow (\gamma_2 > 0)_k^d.\end{aligned}\tag{3.11}$$

Более того, система функциональностей (3.4) допускает реализацию запоминающих устройств типа «триггер» на одном объекте. Возможная схема такого запоминающего объекта приведена на рис. 3.3.

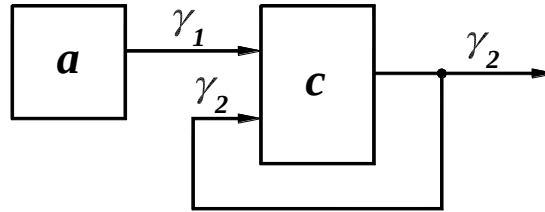


Рис. 3.3. Схема триггера на одном пороговом элементе (объекте C)

Его переключение из одного состояния ($\gamma_2 = 0$) в другое ($\gamma_2 > 0$) и обратно может быть выполнено, например, с использованием следующей числовой интерпретации параметров, представляющих значения свойств:

$$\begin{aligned}(\gamma_1 = 0)_i^a * (\gamma_2 = 0)_k^c &\rightarrow (\gamma_2 = 0)_k^c; \\(\gamma_1 > 0)_i^a * (\gamma_2 = 0)_k^c &\rightarrow (\gamma_2 > 0)_k^c; \\(\gamma_1 = 0)_i^a * (\gamma_2 > 0)_k^c &\rightarrow (\gamma_2 > 0)_k^c; \\(\gamma_1 < 0)_i^a * (\gamma_2 > 0)_k^c &\rightarrow (\gamma_2 = 0)_k^c; \\(\gamma_1 > 0)_i^a * (\gamma_2 > 0)_k^c &\rightarrow (\gamma_2 > 0)_k^c.\end{aligned}\tag{3.12}$$

Система функциональностей типа (3.12) в какой-то мере напоминает функционирование так называемого «порогового элемента» в нейроподобных системах. Особенно наглядно это проступает в четвертой строке (3.12): для того чтобы «погасить» выходной сигнал ($\gamma_2 > 0$) объекта с, необходимо на его вход подать «тормозящее» воздействие в виде сигнала «противоположного» относительно γ_2 знака ($\gamma_1 < 0$).

Рассмотренный пример показывает, что при подходящих числовых соотношениях и типах функциональностей выражения (3.5) – (3.9) могут соответствовать другой, пороговой, модели элементарного устройства обработки информации – нейрона, который широко используется биологическими системами.

В свою очередь, интерпретация числовых значений того или иного типа функциональностей также может быть весьма различной, в том числе и нечисловой. Рассмотрим, например, «цветовую» логику (табл. 3.1), в основе которой может лежать обычная двоичная логика, например, всё та же логическая функция от двух переменных типа «2ИЛИ-НЕ».

При наличии «функциональных констант», т.е. объектов-источников с неменяющимися свойствами (цветами «красным» и/или «зелёным»), такая функциональность может обеспечить функциональную полноценность (функциональную полноту) соответствующей системы «логических элементов» для обработки информации на уровне любых изобретённых человечеством двоичных цифровых вычислительных устройств, включая компьютеры.

Таблица 3.1

Вход γ_1	Вход γ_2	Выход γ_3
<i>красный</i> (0)	<i>красный</i> (0)	<i>зеленый</i> (1)
<i>красный</i> (0)	<i>зеленый</i> (1)	<i>красный</i> (0)
<i>зеленый</i> (1)	<i>красный</i> (0)	<i>красный</i> (0)
<i>зеленый</i> (1)	<i>зеленый</i> (1)	<i>красный</i> (0)

Функциональности могут быть не только «многоходовыми». Рассмотрим пример одной из простейших «одновходовых» функциональностей (рис. 3.4), которая описывает передачу электрического потенциала через электрический проводник Z_i .

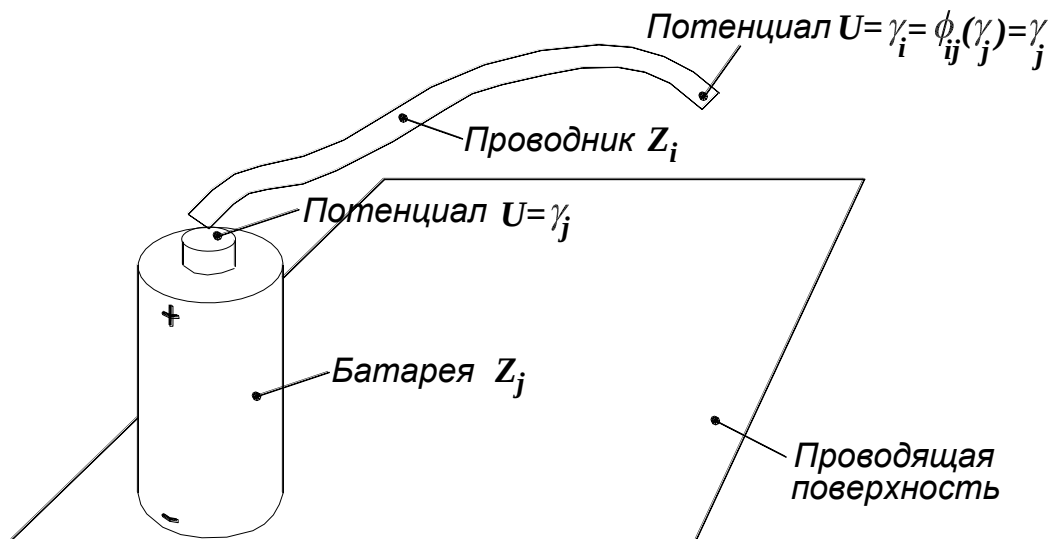


Рис. 3.4. Свойства и функциональность проводника Z_i , подключённого к клемме батареи Z_j

Интересующее нас свойство γ_j батареи как некоего j -того объекта $O_j = Z_j$ – генерирование (с помощью некоего метода-процесса f_j) на своей центральной клемме электрического потенциала $\gamma_j = U$. При этом можно записать: $Z_j = \langle \Gamma_j, \mathbf{M}_j \rangle = \langle \{\gamma_j, \dots\}, \{\gamma_j = f_j(\dots) = U; \dots\} \rangle$.

Интересующее нас свойство γ_j проводника Z_i – передача этого электрического потенциала с одного его конца, находящегося в физическом контакте с центральной клеммой батареи, на другой его конец: $Z_i = \langle \Gamma_i, \mathbf{M}_i \rangle = \langle \{\gamma_i\}, \{\gamma_i = \gamma_j = U\} \rangle$. То есть в данном случае объект Z_i реализует очень простой и единственный метод, входящий в $\mathbf{M}_i$, – передачу потенциала U без потерь по проводнику Z_i . Однако не исключено, что со временем мы обнаружим, что если второй (правый) конец проводника подключить к какому-нибудь электрическому устройству, например к лампочке, то в проводнике появится ток и потенциал на его правом выводе уменьшится за счёт падения напряжения на самом проводнике из-за его небольшого, но всё же реально-

го внутреннего сопротивления. В этом случае мы можем *уточнить* метод $\mathbf{M}_i$ и перечень свойств Γ_i проводника $\mathbf{Z}_i$, введя параметры (свойства), характеризующие внутреннее сопротивление проводника и падение напряжения на нём при прохождении тока. То есть мы можем уточнять используемую нами модель проводника $\mathbf{Z}_i$ по мере накопления знаний о нём!

Проанализируем теперь иную, неинформационную функциональность. Назовём её «механической» (рис. 3.5).

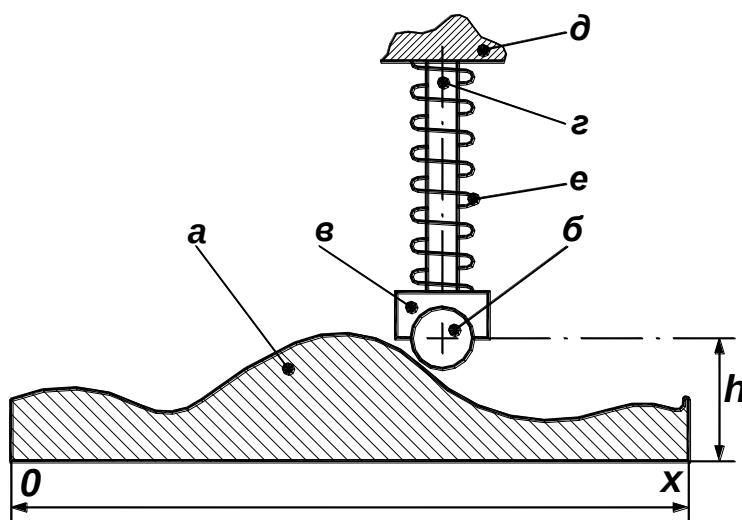


Рис. 3.5. Пример механической функциональности:

a – подвижный профиль, $б$ – ролик, ось которого закреплена на держателе $в$,двигающемся вертикально вдоль стержня $з$, который жёстко закреплён в основании $д$ верхним торцом. Ролик $б$ прижимается пружиной $е$ к верхней грани профиля a

При горизонтальном смещении профиля a вдоль координаты x (в обоих направлениях) вертикальное смещение h ролика будет зависеть от формы верхней грани профиля a и – незначительно – от диаметра самого ролика. Обозначим конкретное текущее положение профиля a относительно какой-то «базовой» точки символом x , а соответствующее текущее положения самого ролика $б$ и держателя $в$ (опять же относительно какой-либо «базовой» точки) – символом h . Ясно, что, меняя форму верхней грани профиля a , можно менять в достаточно широких пределах *закон* (функцию φ) смещения ролика $б$ по вертикали: $h = \varphi(x)$, т.е. *менять характер перемещения ролика вверх-вниз в*

зависимости от формы и пространственного положения самого профиля, которые мы договорились относить к его свойствам (как, кстати, и пространственное положение ролика). То есть мы можем легко менять механическую функциональность (функцию φ) объектов $\bar{b}$, $\bar{v}$ в достаточно широких пределах, задавая тот закон перемещения ролика в пространстве, который нас интересует.

Если обозначить соответствующие объекты как O_a (профиль) и $O_{\bar{b}}$ (ролик), то, игнорируя (временно) остальные свойства объектов O_a и $O_{\bar{b}}$, мы можем записать: $O_a = \langle \Gamma_a, \mathbf{M}_a \rangle = \langle \{ \dots, x, \dots \}, \{ \dots \} \rangle$; $O_{\bar{b}} = \langle \Gamma_{\bar{b}}, \mathbf{M}_{\bar{b}} \rangle = \langle \{ \dots, h, \dots \}, \{ \dots, h = \varphi(x), \dots \} \rangle$ для конкретного закона φ .

Несколько иной пример механической функциональности приведён в [15] на рис. 2.4 и на рис. 2.2.4 в [13].

Следующий пример: имеется две сцепленных друг с другом шестерни. Поворачивая одну из них, т.е. меняя параметр, определяющий конкретное её положение в пространстве относительно некоторого начального («исходного») положения, мы заставляем поворачиваться и другую, изменяя и её состояние.

Ещё один пример. Имеется магнитное поле Земли и стрелка компаса. В пределах Земли стрелка показывает направление (т.е. принимает соответствующее состояние) на геомагнитный северный полюс Земли.

Ясно, что подобного рода примеров можно привести множество, поскольку (обратим на это внимание ещё раз) все окружающие нас предметы-объекты в этом мире имеют, во-первых, вполне конкретные свойства, включая их местоположение, которые позволяют нам отличать их один от другого, даже если некоторая часть их свойств одинакова. Во-вторых, они *меняют* некоторые свои свойства (например, местоположение) при взаимодействии (контактном или бесконтактном) с другими объектами-предметами или иными физическими объектами-сущностями (типа микрочастиц, квантов и т.д.).

Из рассмотренных выше примеров ясно также, что при увеличении числа «входных» и (или) «выходных» свойств объектов в базовом выражении типа (3.4), в том числе за счёт включения в него

свойств объектов, взаимодействующих на расстоянии (например, через поле), число возможных комбинаций взаимодействий и их типов существенно увеличится.

Рассмотрим в этой связи в качестве примера ситуацию с химическими реакциями (т.е. с взаимодействиями «химического типа», в которых могут участвовать самые разные известные и даже неизвестные нам вещества и физические законы). Сначала проанализируем некую абстрактную химическую реакцию между двумя веществами a и b , результатом которой является новое вещество c ²⁵:

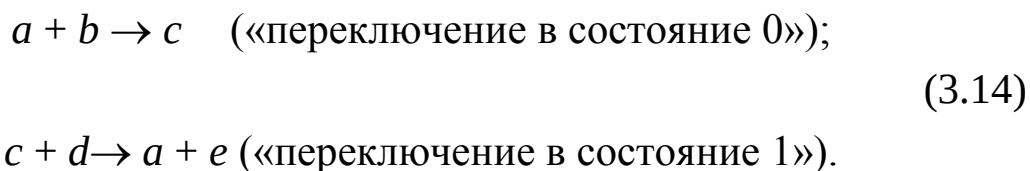


Вещества a и b , так же, кстати, как и вещество c , могут иметь достаточно разнообразные наборы собственных физико-химических свойств, которые могут иметь (или не иметь) отображения на различные числовые шкалы. Эти свойства вполне отвечают введённым выше определениям свойств, а их комбинации (т.е. конкретные наборы физико-химических свойств) определяют состояния соответствующих объектов. Получившееся в ходе реакции (3.13) вещество c , естественно, отличается по своим физико-химическим свойствам от исходных свойств компонентов a и b хотя бы в силу того, что в него, согласно записи реакции, входят как атомы вещества a , так и атомы вещества b , обладающие своими специфическими свойствами. При этом общее число атомов в молекуле вещества c – опять же согласно записи реакции (3.13) – равно сумме атомов молекул веществ a и b . То есть получившаяся молекула вещества c обладает иным набором свойств (по крайней мере, в количественном плане), нежели исходные объекты a и b , что отвечает условиям функциональности, заданному определением функциональных свойств объектов. То, что при этом исходные объекты a и b перестали существовать, не имеет принципиального значения, поскольку в условиях определения функциональных свойств нет никаких указаний на необходимость сохранения первоначальных носителей исходных свойств, т.е. исходных

²⁵ пример взят из книги: Общая химия / Под ред. Е.М. Соколовской, Г.Д. Вовченко, Л.С. Гузея. – М.: МГУ, 1980. – С. 192.

объектов a и b . Более того, для формального соответствия записи реакции (3.13) условиям определения функциональных свойств можно считать, что объект, подвергающийся «функциональному» воздействию со стороны некоторого функционального свойства (набора свойств), – это, например, объект b , а носитель набора функциональных свойств – объект a . В результате химического взаимодействия (реакции), при которой на объект b с его набором свойств воздействуют свойства объекта a , объект b меняет свои свойства, превращаясь в объект c с иным, нежели у b , набором свойств. Или же наоборот – можно считать исходным объект a , который под действием свойств объекта b меняет свое состояние на состояние, соответствующее свойствам объекта c .

Чтобы окончательно убедиться в правомерности такой трактовки химических реакций, приведем ещё один пример, содержащий две реакции, в совокупности описывающие работу устройства, которое рассматривалось выше на рис. 3.2 и которое поэтому можно было бы назвать «химическим триггером»:



Исходя из закона сохранения вещества, вещество e должно содержать в себе компоненты веществ b и d ($e=b*d$). С другой стороны, вещество e можно рассматривать как «выходной» сигнал «химического триггера», свидетельствующий о его переключении из состояния «0» (т. е. из « c ») в состояние «1» (в « a »). Соответственно, вещество b – это «входной» сигнал, переключающий триггер из состояния «1» (« a ») в «0» (« c »), а вещество d – «входной» сигнал, переключающий триггер из «0» (« c ») в «1» (« a »). Триггер такого типа принято называть RS-триггером.

Отметим попутно существенное различие в реализации функций «химического триггера» в примере (3.14) и в трактовке Г. Хакена [36, с. 333]. Если в «схеме» аналогичного бистабильного запоминающего

объекта (триггера) в трактовке Хакена роль двух устойчивых состояний играют две различных концентрации реагентов, весьма сильно зависящие от множества трудно контролируемых факторов, то в примере (3.14) эту роль выполняют всего две различных молекулы *a* и *c*. То есть предлагаемая «формально-технологическая» интерпретация химических функциональностей в «булевском» контексте оказывается существенно проще и соответствует процессам, гораздо чаще встречающимся в природе, чем интерпретация, основанная на двух уровнях концентраций взаимодействующих веществ (соответствующих уровням логических сигналов в булевой логике) [6, 13, 15, 23]. Аналогичные примеры можно привести и для других типов химических реакций.

Таким образом, многие (а скорее всего – все²⁶) химические взаимодействия в чистом, так сказать, виде полностью вписываются во введённое выше определение функциональности объектов. При этом нетрудно заметить, что уже простейшие примеры абстрактных химических реакций типа синтеза (3.13) или «RS-триггера» (3.14) могут порождаться простейшими функциональностями любого из рассмотренных выше типов – от (3.5) до (3.9) включительно. Понятно, что более сложным реакциям, в которых синтезируется и/или разлагаются на более простые компоненты более сложные вещества, соответствует огромное многообразие других типов функциональностей, не попавших в классификацию (3.5) – (3.9). Например, в научной литературе имеются публикации, в которых такого рода функциональности трактуются как функциональности, аналогичные процессам обработки данных в экспертных системах, в криптографии, для маркировки органических продуктов, для организации вычислительных процессов и т.д., и т.п. [13, 15]. Хотя эти публикации не связаны напрямую с обсуждаемым общим способом представления функциональностей объектов, они дополни-

²⁶ Гребеничиков Е.Н., Крылов С.М. Системный анализ гетерогенных слабоструктурированных автоматов. II. Переход от преобразований «химического типа» к логическим схемам и обратно // Вестник Самар. гос. техн. ун-та. Сер. Технические науки. – 2012. – № 2 (34). – С. 65-71.

тельно подтверждают тот факт, что химические реакции – это мощный класс операций преобразования объектов, в который входят не только простейшие их типы, соответствующие функциональностям двоичных логических элементов, повсеместно используемых в компьютерах [15], но и огромное число других.

На основе подобных преобразований можно существенно расширить классическую теорию автоматов, добавив к ней, во-первых, новое направление с предварительным названием «теория гетерогенных автоматов», в котором входные и выходные сигналы могут носить разнородный (т.е. гетерогенный) характер. Такого рода автоматы могут иметь определённые преимущества перед обычными, гомогенными, и в первую очередь – в микроэлектронике, поскольку позволяют существенно расширить потенциальную элементную базу используемых электронных компонентов [6]. Во-вторых, на основе преобразований «химического типа» можно построить и другую ветвь этой теории – так называемую «теорию слабоструктурированных самосинхронных гетерогенных автоматов с индивидуальными именами объектов операций» (ТССГА с ИИОО) [23].

На рис. 3.6 слева приведена схема, поясняющая функционирование одного из подобных автоматов, реализующего преобразование типа (3.14), и традиционный для классической теории автоматов граф его переходов (справа).

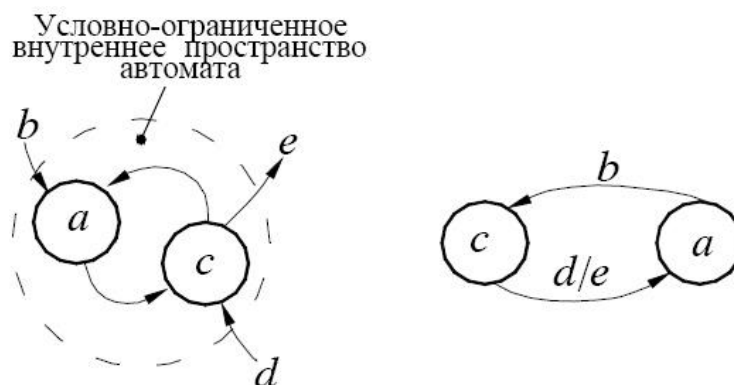


Рис. 3.6. Условное изображение (слева) и граф переходов (справа) гетерогенного слабоструктурированного самосинхронного автомата с индивидуальными именами объектов операций, реализующего функцию «химического» RS-триггера

В рамках ТССГА с ИИОО удалось показать, что любой обычный (гомогенный) конечный автомат (состоящий, например, из традиционных логических элементов) можно преобразовать в подходящий «гетерогенный слабоструктурированный самосинхронный автомат с индивидуальными именами объектов операций», и наоборот, каждый такой автомат – в обычный автомат на двоичных логических элементах [15, 23]. То есть оказывается, что природные химические процессы на самом деле «гораздо умнее», чем мы до сих пор о них думали, и что среди них вполне закономерно появление процессов типа самовоспроизведения клеточных автоматов фон Неймана и Лангтона, функции которых легко записываются с помощью логических схем [34, 67].

С некоторыми фундаментальными свойствами и функциональностями атомов и молекул связано и такое распространённое физическое понятие, как агрегатное состояние вещества. В проведённых компьютерных экспериментах «структурообразующая» функциональность шарообразных упругих объектов единичной массы (названных «квазиатомами») обеспечивалась силами притяжения (разноименных) и отталкивания (одноименных) «зарядов», а энергетическая – свойством квазиатомов «быть инерционными объектами».

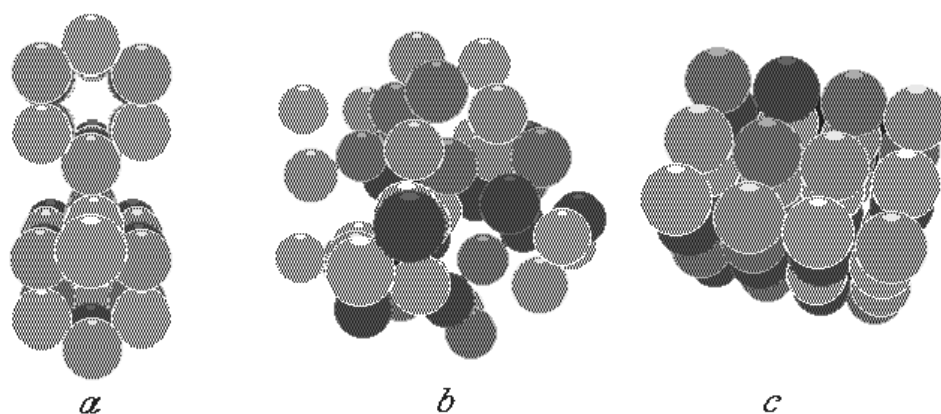


Рис. 3.7. Исходный фрагмент квазивещества в виде двух шестигранных трубок (а) сначала «разогревается» до температуры «плавления», превращаясь в бесформенный комок (б), который при «охлаждении» самопроизвольно кристаллизуется в прямоугольный брусок-«кристалл» (с). Разными оттенками серого отмечены квазиатомы, имеющие разный заряд и относящиеся к разным слоям исходного фрагмента квазивещества

Соответствующим образом «заряженные» квазиатомы помещались в трёхмерную среду с регулируемой вязкостью. Этих свойств и

функциональностей оказалось вполне достаточно, чтобы наблюдать эффект плавления сформированных из подходящих «квазиатомов» твёрдых шестигранных трубок и их переход в жидкое, а затем – в газообразное состояние, либо при охлаждении из жидкого состояния – их кристаллизацию в твёрдый прямоугольный брусок-кристалл из тех же «квазиатомов» (рис. 3.7).

Заканчивая этот небольшой обзор базовых принципов и истоков функциональности различных функциональных систем, образуемых различными, в том числе разнородными (гетерогенными) объектами, ещё раз подчеркнем, что реальное многообразие таких систем значительно богаче тех простейших примеров, которые соответствуют выражениям (3.5) – (3.9). Тем не менее даже в этом ограниченном наборе тот тип функциональностей, который гарантирует полноценную обработку информации (т.е. фактически обеспечивает универсальность используемых нами вычислительных систем, включая такие хорошо известные математикам модели, как машины Тьюринга), занимает крайне незначительное место. Более того, нетрудно догадаться, что на самом деле конкретные значения свойств, влияющих на функциональные преобразования, могут определяться гораздо более сложными формулами, чем использованные в (3.10) или (3.11) знаки равенства или сравнения на больше/меньше.

Подведём кратко некоторые итоги. Итак, конкретные математические выражения, описывающие характер взаимодействия соответствующих свойств в нашей объектно-ориентированной модели, должны перечисляться, согласно определению (1.3) или (3.2), во множестве методов (функциональностей) $\mathbf{M}_i$ с обязательным указанием тех свойства, к которым данные методы (способы взаимодействия объектов) применимы. В частности, если множество методов для объекта O_i пусто, то предполагается, что ОФТ-модель этого объекта не имеет в данной интерпретации собственной функциональности и подчиняется только операциям множеств $\mathbf{F}_T$ и $\mathbf{F}_A$ соответствующей технологии. Как уже отмечалось в параграфе 1.1, все числа (основные объекты операций арифметики и вычислительной математики) имеют

пустое множество методов, а список их параметров содержит, в случае обычных чисел, только одну позицию для одного параметра, значение которого в точности равно представляемому числу (или, к примеру, два параметра – для комплексных чисел).

3.3. ЭМЕРДЖЕНТНЫЕ СВОЙСТВА И ФУНКЦИОНАЛЬНОСТИ

Эмерджентные свойства объектов и систем – ещё одна тема, введенная в теорию систем Россом Эшби²⁷ и не имеющая пока чёткого формального определения. ОФТ позволяет это сделать:

Определение эмерджентного свойства. Эмерджентным называется такое свойство объекта O_i (с непустым значением параметра, представляющего свойство), которое отсутствовало на предыдущих шагах технологии в тех объектах, из которых был получен объект O_i .

Действительно, согласно Эшби, эмерджентными называются свойства, которые появляются у объекта или системы объектов неожиданно, непредсказуемо, то есть оказываются совершенно новыми, необычными. С формальной точки зрения принятого нами подхода это означает, что вследствие каких-то операций над объектами совокупный список (множество) их свойств изменился так, что в нём появилось некое свойство, отсутствовавшее ранее. Например, вместо множества свойств $\Gamma_i = \{\gamma_1, \gamma_2, \dots, \gamma_n\}$ мы вследствие какого-то преобразования объекта (системы) O_i имеем $\Gamma_i = \{\gamma_1, \gamma_2, \dots, \gamma_n, \gamma_{n+1}\}$.

Достаточно очевидным примером эмерджентного свойства, которое может проявляться сколь угодно долго по мере развития («эволюции») технологии, является свойство «иметь заданную форму». Подчеркнем, что этот тезис справедлив только для n -мерных технологий при $n > 1$. Обоснование эмерджентности этого свойства приведено на стр. 61-64 в [14]. Суть его заключается в том, что как только мы получаем новую конструкцию, в которую входит максимальное на

²⁷ Ashby W.R. An Introduction to Cybernetics, Chapman & Hall, London, 1956.

данный момент (на данный шаг T) число элементов и которая имеет, естественно, какую-то пространственную форму, соответствующую этому числу элементов и их взаиморасположению, являющуюся новой для данной технологии (по той причине, что никакая другая конструкция еще не имела *такого* числа элементов базы и *такого* их взаиморасположения), мы, используя подходящую операцию синтеза, можем присоединить к этой конструкции еще один элемент базы и снова получить новую конструкцию (с новым числом и составом элементов) и, соответственно, с новым их пространственным взаиморасположением, не встречавшимся на предыдущих шагах T . Таким образом, имеет смысл сформулировать следующее утверждение:

Утверждение 3.1. Свойство «иметь заданную форму» является эмерджентным (в технологиях с размерностью объектов не менее 2).

Другим примером появления эмерджентного свойства могут служить свойства объектов, изображённых на рис. 3.5, иллюстрирующем механическую функциональность ролика b с держателем a при горизонтальном перемещении профиля a . При абсолютно горизонтальной форме верхней поверхности профиля a перемещение ролика b будет нулевым при любом смещении профиля (в диапазоне координат $0-x$), т.е. как свойство параметр h , представляющий высоту положения ролика, вводить в перечень свойств объектов b и a не имеет смысла – она постоянна. Однако при неровной верхней поверхности движущегося профиля a ролик «неожиданно» начнёт перемещаться вверх-вниз, и соответствующее свойство, представленное величиной h , нам нужно будет ввести в перечень свойств объектов b , a (рис. 3.5).

Очевидным примером неэмерджентного свойства является уже упоминавшееся свойство «представлять некоторое количество или число», например, свойство кучи разнородных предметов, собранных в одном месте, представлять то самое количество, которое равно числу предметов в этой куче. Действительно, согласно определению эмерджентности, при всех своих изменениях, начиная с первых шагов соответствующей технологии, это свойство «кучи предметов» не вы-

ходит за свои собственные «рамки», т.е. за пределы свойства «представлять некоторое количество или число» предметов в этой куче. Таким образом, справедливо следующее утверждение:

Утверждение 3.2. Свойство чисел «представлять некоторое количество» неэмерджентно.

Собственно, этот факт является своего рода следствием ОФТ-представления чисел в виде (1.4). В дальнейшем для краткости мы будем называть подобные свойства «числовыми» или «количественными». При этом не следует путать числовое неэмерджентное свойство, равное какому-либо конкретному числу, и некое эмерджентное свойство, представляемое этим же числом (т.е. количественным значением параметра, представляющим данное свойство)! К сожалению, легкость, с какой сами числа начинают ассоциироваться с параметрами, представляющими различные свойства различных предметов реального мира, постоянно приводит к неправильной оценке роли и сущности математики в общей системе наук. Никким образом не умаляя её действительно выдающихся достижений, напомним, что заблуждения одной из первых математических школ – пифагорейцев – имели место как раз на этой почве – подмены реальных предметов и их свойств некими числовыми «сущностями». В этой связи совсем не случайно в последнее время математика и её основания подвергаются самому серьезному анализу и критике именно с позиций неадекватности её центрального объекта – числа – предметам и явлениям окружающего мира [5, 46, 58, 62, 72, 78].

Как и обычные, эмерджентные свойства могут быть функциональными и нефункциональными. Например, свойство «иметь заданную форму» само по себе (т.е. без рассмотрения других свойств) в n -мерных ($n > 1$) технологиях, как уже говорилось, является эмерджентным (точнее, бесконечно эмерджентным). Однако при этом, ещё раз подчеркнем, – *при отсутствии других свойств* оно оказывается нефункциональным, поскольку возможное изменение формы (а вместе с ним и состояния) конструкции, с которой сближается (или физически контактирует) данная конструкция, и даже изменение в аналогичной ситуации собственной формы (т.е. собственного состояния)

может происходить только при обязательном наличии у конструкций некоторых других – *физических* – свойств. Например, свойства отталкивания, или свойства жёсткости, или, наоборот, – «гибкости связей» между элементами в конструкции и т.д. Таким образом, можно сформулировать следующее:

Утверждение 3.3. Свойство иметь заданную форму (в пространствах с размерностью не менее двух) само по себе эмерджентно, но нефункционально.

Кстати, другим моментом, подтверждающим нефункциональность этого эмерджентного свойства, является то обстоятельство, что многие пространственные формы могут быть достаточно полно описаны математически с использованием топологических подходов и методов.

Подчеркнем также, что аналог этого свойства для одномерного пространства, а именно «длина конструкции», согласно определению эмерджентности не является даже эмерджентным, поскольку параметр свойства «длина» в одномерном пространстве выражается одним *числовым* значением в соответствующих единицах.

Любопытно, что *не*функциональная эмерджентность свойства «иметь заданную форму» может сыграть положительную роль в формировании технологий с очень интересными и многообещающими свойствами, а именно – с функционально-эмерджентными свойствами! Действительно, если воспользоваться эмерджентностью этого свойства и добавить к нему (т.е. как бы «посадить», «наклеить» на него) некоторое другое, возможно даже неэмерджентное, но зависящее от формы функциональное свойство, то мы можем получить систему свойств, с одной стороны, (бесконечно) эмерджентных, с другой – функциональных, функциональность которых связана с формой (согласно нашей концепции). То есть получить систему (бесконечной) *функциональной эмерджентности* свойств!

Смысл и главным образом значение подобной системы настолько велики (в том числе и с философской точки зрения), что стоит остановиться на них подробнее. Фактически они означают, что если такая система имеет место в определенном физическом мире, то эволюция

(совершенствование систем) этого мира может протекать не только в материальном плане или в плане усложнения «поведения» (динамики) систем. Она может протекать еще и в направлении конструирования новых, в том числе (а может быть, и в первую очередь) физических свойств с принципиально новыми функциональностями, не существовавшими раньше!

Данные положения хорошо согласуются с общепринятыми представлениями о появлении эмерджентных свойств у сложных систем. Действительно, появление сложной системы сопровождается, очевидно, и появлением новой, не встречавшейся раньше формы (имея в виду не только физическую, но и другие аспекты топологической структуры сложной системы). При этом вследствие *физической* реализации всех её компонентов возможно появление в том числе и функционально-эмерджентных свойств, а не только эмерджентных (но нефункциональных) свойств.

В рассматриваемой концепции получения бесконечной функциональной эмерджентности есть один тонкий момент, который часто скрывает очень существенное различие между просто «новой, неожиданной динамикой» некоторой конструкции как сложной системы, состоящей из простых функциональностей типа (3.5) – (3.9), и действительно эмерджентным свойством, которое тоже может обладать «неожиданной функциональностью», отличной от функциональностей исходной системы. «Новая, неожиданная динамика» есть по сути дела лишь необычное (не наблюдавшееся ранее) «поведение» конструкции, запрограммированное конкретными известными свойствами и конкретными известными функциональностями конкретного соединения её компонентов. То есть это, согласно нашему определению свойств, – не свойство! Это – именно динамика, которую для конечного числа элементов и известных их начальных состояний (взятых из соответствующих списков или множеств свойств) и функциональных таблиц (взятых из соответствующих списков методов или функциональностей) всегда можно рассчитать заранее!

В качестве иллюстрации рассмотрим неэмерджентное функциональное свойство, задаваемое некоторой «таблицей переходов» в двумерной клеточной среде, где каждая клетка (рассматриваемая как элемент базы A) имеет квадратную (или прямоугольную) форму и окружена четырьмя ближайшими соседями (рис. 3.7). Такие структуры были использованы фон Нейманом для конструирования самовоспроизводящихся автоматов [34] и с тех пор широко применяются для исследования самых различных проблем, связанных с самовоспроизведением (см., например [67]).

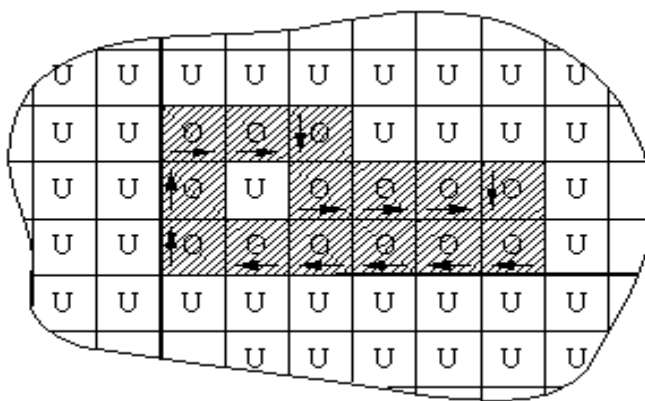


Рис. 3.8. Пример простого автомата фон Неймана (заштрихованная область), реализованного в клеточной среде

Клеточный автомат фон Неймана, точнее, каждая ячейка плоской клеточной среды, имеет 29 внутренних состояний, которые удобно кодировать символами или числами. Одно из этих состояний – исходное, «покоящееся» – является как бы выделенным и обозначается через U (рис. 3.8). Другие (неисходные) состояния одной клетки-автомата могут влиять на состояния ближайших соседних с ней клеток-автоматов: «левую», «правую», «верхнюю» и «нижнюю». Для удобства такого рода технологию, состоящую из множества клеток $u_{ij} \in A = \{u\}$, т.е. из однотипных клеток u с координатами i, j , расположенных в виде непрерывной плоской матрицы и имеющих n внутренних состояний, переходы между которыми (т.е. их переключения) заданы функциональной таблицей, реализующей «переключающие» операции из набора соответствующих операций f_{FN} , мы будем назы-

вать технологией фон Неймана (сокращенно ТФН, или $\mathbf{T}_{\text{FN}}$). Каждая клетка ТФН легко описывается в нашей объектно-ориентированной системе описания свойств и функциональностей объектов. Для этого в списке её свойств достаточно выделить одну позицию для указания текущего (одного из n возможных) состояния γ_{ij} клетки u_{ij} , которое соответствует вполне конкретному и определенному её свойству (закодированному, например, числом), а в списке методов указать характер переключения состояний (свойств) клетки в зависимости от её собственного текущего состояния и состояния её ближайших 4-х соседей в соответствии с f_{FN} (обычно такое переключение задается, как уже говорилось, таблицей переходов). То есть мы можем записать:

$$\mathbf{T}_{\text{FN}} = \langle \mathbf{B} \rangle = \langle \{ u_{ij} \} \rangle, \quad (3.15)$$

где $u_{ij} = \langle \mathbf{T}_{\text{FN}}, \mathbf{M}_{\text{FN}} \rangle = \langle \{ \gamma_{ij} \}, \{ \gamma_{ij}(t+1) = f_{\text{FN}}(\gamma_{ij}(t), \gamma_{(i-1)j}(t), \gamma_{(i+1)j}(t), \gamma_{i(j-1)}(t), \gamma_{i(j+1)}(t)) \} \rangle$, t – текущее дискретное время: $t \in \{0, 1, 2, \dots, m\}$.

В записи (3.15) отсутствуют множества $\mathbf{F}_T$ и $\mathbf{F}_A$ операций технологии $\mathbf{T}_{\text{FN}}$, поскольку в ней никакие «внешние» по отношению к объектам-клеткам из множества $\mathbf{B}$ операции не применяются: вся функциональность клеток заключена в них самих и задаёт – при определённой исходной конфигурации клеток – процесс самовоспроизведения всего автомата, как совокупности клеток [34]. Для удобства такую совокупность мы будем в дальнейшем называть «надклеточным» автоматом.

Казалось бы, совпадение формы записи (3.15) $\mathbf{T}_{\text{FN}}$ с формой записи эволюционных технологий (1.9) $\mathbf{T}_{\text{ЭТ}}$ свидетельствует в пользу признания $\mathbf{T}_{\text{FN}}$ эволюционной технологией, однако более тщательный анализ показал, что это не совсем так, точнее – почти так, но с определёнными оговорками. На самом деле оказалось, что $\mathbf{T}_{\text{FN}}$ находится вблизи границы между эволюционными технологиями и неэволюционными, вероятность эволюционного развития в которых хотя и близка к нулю, но всё же чуть-чуть, на очень незначительную величину, больше [13, 15, 16]. Не зря, видимо, многие учёные связывают самовоспроизводящиеся автоматы фон-Неймана с решением фундаментальных вопросов развития живой природы.

Фактически же фон Нейман доказал следующее.

Утверждение 3.4. В клеточной двухмерной среде, организованной в виде прямоугольной матрицы из однотипных элементов базы, имеющих 29 внутренних (функциональных) состояний, возможно самовоспроизведение клеточных структур (надклеточных автоматов) любой конечной сложности, включая машины Тьюринга (доказательство приведено в [34]).

Покажем теперь, что в такой среде (то есть в ТФН) можно ввести понятие формы и, в соответствии с этим понятием, определить бесконечно эмерджентное свойство «иметь заданную форму». Действительно, если ассоциировать состояние U с неким (условно) пустым элементом (как бы с «пустой» клеткой), а все другие – с «заполненными» (некоторой «активированной» функциональностью!), то естественным образом возникает понятие двумерной формы, занимаемой конкретной конструкцией (надклеточным автоматом) в данной среде и совпадающей с формой фигуры, где все «непустые» клетки, принадлежащие данной конструкции, как бы затемнены на фоне «светлых» клеток, находящихся в состоянии U (рис. 3.8). То есть справедливо следующее.

Утверждение 3.5. В ТФН возможно эмерджентное свойство «иметь заданную форму».

Все свойства (включая функциональные) клеток (элементов базы) в этой технологии описаны как свойства некоторых (условно говоря, электрических) схем, способных пропускать, преобразовывать и воспринимать импульсы двух типов – «обычные» и «специальные» – от своих ближайших соседей. То есть преобразование любой клетки из состояния U в любое другое состояние и обратно, а также передача и «обработка» импульсов мыслятся как работа некоторого встроенного в клетку механизма (внутриклеточного автомата), переключающего состояния клетки по «сигналам», поступающим на его «контактные площадки», которыми являются все четыре стороны (границы) клетки. Поскольку в ТФН вся среда – это сплошные контактирующие между собой клетки, без каких-либо пустот и пробелов, то любой сигнал

всегда имеет и свой источник, и свои конкретные адресаты – грани ближайших клеток. Процесс изменения состояний клеток, таким образом, всегда как бы «замкнут» внутри таких клеток-автоматов и строго детерминирован (подчиняется определенной закономерности). По этой причине он и может быть представлен так называемой «таблицей переходов», которая фактически связывает текущее состояние самой клетки, текущие состояния её ближайших четырех соседей и будущее (следующее) своё состояние. Выйти за рамки этой таблицы – за рамки указанных выше 29 состояний – *никакая клетка не может*. То есть свойства каждой такой клетки с очевидностью неизмердженны, поскольку определены раз и навсегда. При этом форма, которую занимает какая-либо функциональная конструкция из нескольких клеток (надклеточный автомат), никакого влияния на характер этой таблицы не имеет. Действительно, функциональные свойства клеток среды, исходя из таблицы переходов, никак не связаны с формой соответствующего надклеточного автомата. Последняя (и единственная) возможность для такой связи – это *возможность перехода сигналов состояний клеток через их границы*. Однако такой переход, как уже отмечалось ранее, выполняется и учитывается всегда (в таблице переходов), поскольку любая клетка всегда окружена другими (пустоты отсутствуют), и соответствующая соседняя клетка реагирует на такой сигнал в точном соответствии со всё той же таблицей переходов. Таким образом, новому свойству (и новому состоянию, связанному с этим новым свойством) просто неоткуда появиться. Тем не менее рассмотренных 29-ти состояний клеток (которые мы связали с функциональными свойствами) оказывается вполне достаточно для реализации в подобной клеточной среде самых различных процессов (динамик), включая такой далеко не тривиальный процесс, как самовоспроизведение надклеточного автомата подходящей сложности [34, с. 179-321]. Подчеркнем, что других свойств, кроме рассмотренного возможного эмерджентного свойства «иметь заданную форму» и чётко описанных 29 функциональных и с очевидностью неизмердженных состояний каждой клетки, в данной технологии нет.

Чтобы в дальнейшем не путать очень близкие на первый взгляд (но существенно различные по внутреннему смыслу) понятия функциональной эмерджентности и новой функции (точнее, нового процесса или новой динамики), мы впредь будем пользоваться только оговорёнными выше терминами и только в оговорённом выше смысле. То есть оказывается справедливо:

Утверждение 3.6. ТФН нефункционально эмерджентна, но позволяет создавать новые функции (новые процессы или динамики), не существовавшие ранее.

Кстати, последнюю возможность (новые динамики) тоже нетрудно связать с эмерджентным нефункциональным свойством «иметь заданную форму», поскольку новая динамика как раз и определяется новым соединением (новой «формой») двумерных элементов базы (клеток среды) и начальным распределением состояний этих клеток.

На настоящий момент (в рамках ОФТ) удалось выделить несколько возможных механизмов склейки эмерджентных нефункциональных и неэмерджентных функциональных свойств, способных породить функциональную эмерджентность. Во-первых, это квантово-механические эффекты, возникающие на атомарном уровне при формировании новых молекул («размазанность» и специфичность форм электронных «орбит-облаков», ковалентная связь, молекулярные орбитали, поляризация молекул, множественное дробление энергетических уровней при формировании механизмов металлической проводимости и т.д.). Во-вторых, это механическая эмерджентная функциональность типа приведённой на рис. 3.5, основанная опять же на эмерджентности формы объектов, а также на жёсткости, упругости и «скользкости» некоторых их материалов. Именно благодаря успешному освоению этого типа эмерджентной функциональности выдающиеся специалисты-механики 19 века смогли сконструировать удивительные механические куклы, поражавшие своими движениями современников. Наконец, в-третьих, это использование концепции «поля». Последний вариант достаточно подробно рассмотрен в [13, 15].

Глава 4. МЕХАНИЗМЫ ЭВОЛЮЦИИ

4.1. ПРЕДПОСЫЛКИ

Сложность используемых в биологической эволюции объектов-конструкций, несомненно, является почти очевидным «внешним» фактором, мешающим до конца понять её внутренние механизмы. Даже попытки использовать математическую теорию сложности объектов Колмогорова, на которую в своё время возлагали большие надежды многие выдающиеся учёные, ожидаемых «прорывных» результатов в понимании базовых основ эволюционных процессов не принесли [52, 53]. Теперь можно понять, почему. По «неписаной», так сказать, математической традиции объектами математической теории сложности оказались не реальные физические объекты, а строки символов, т.е. объекты типа (1.4), не имеющие, согласно параграфу 1.1, никаких физических свойств и связанных с ними функциональностей. Что же изменится в результатах применения теории сложности Колмогорова, если вместо абстрактных объектов типа (1.4) рассматривать более адекватные реальным, но всё же абстрактные объекты-модели типа (1.3) или (3.2)?

В теории сложности Колмогорова сложность любой линейной конструкции – строки символов (букв или цифр) – ассоциируется с возможной сложностью алгоритма, порождающего эту последовательность [10]. Чем проще последовательность, чем больше повторяющихся участков она содержит, тем ниже её сложность, тем проще может быть алгоритм её синтеза. Поскольку мы с самого начала допускаем применимость понятия алгоритма (и, как мы убедились, небезосновательно) в том числе и к последовательностям технологических операций, т.е. к алгоритмам процессов над физическими (материальными) объектами, естественно, сложность синтеза материальных конструкций имеет смысл рассматривать в аналогичном аспекте. Но алгоритмы процессов над физическими объектами должны выполнять какие-то реальные действия-операции над ними. Единственный источник таких операций-

действий, согласно интерпретации эволюционных технологий в форме (1.9), – это функциональности самих объектов, т.е. их возможность совершать какие-то конфигурационные или иные изменения, приводящие к нужным результатам, как это только что выяснилось на примере T_{FN} . Следовательно, необходимо научиться как-то формально описывать потенциальную (возможную) функциональность конструкций исходя из потенциальной функциональности её компонентов, с тем чтобы потом можно было оценить уровень требуемой функциональности, необходимой, например, для реализации тех же процессов (алгоритмов) самовоспроизведения.

Для простоты рассмотрим сначала ситуацию с линейными конструкциями, тем более что природа пошла по аналогичному пути, избрав для своих основных функциональных компонентов – белков – именно первоначальную (при синтезе) линейную структуру. Аналогичные линейные конструкции используются в живых организмах и при передаче генетической информации, и во многих других сопутствующих процессах.

Как известно, структура любой линейной информационной конструкции, например, той же строки символов в сообщении, полностью описывается заключённой в ней информацией:

$$I = -k \sum_{i=1}^r p_i \log_2 p_i, \quad (4.1)$$

где k – число символов в передаваемом сообщении (т.е. его длина), p_i – вероятность появления i -того символа (из некоторого конечного алфавита в r символов) в передаваемом сообщении (см., например, [38]).

Единицу информации принято называть «битом».

Из формулы (4.1) следует, что величина передаваемой в сообщении информации зависит не только от числа символов в нём, но и от вероятностей появления этих символов в сообщении.

Между тем при проектировании наиболее массовых систем связи или обработки информации (типа Интернета, компьютеров и т. п.) почти никогда не используется такой вероятностный подход! То есть

наиболее массовые современные системы кодирования символов для различных алфавитов конструируются так, как будто вероятности появления различных символов в сообщениях одинаковы! Например, широко используемые системы кодирования типа ASCII или КОИ-8 преобразуют любые символы в коды абсолютно одинаковой длины независимо от реальной частоты появления этих символов. Исключение – почти вышедшая из употребления азбука Морзе. То есть современные массовые практические системы кодирования, ориентированные на компьютерные технологии, пренебрегают формулой (4.1)! Делается это в угоду простоте реализации соответствующих кодирующих и декодирующих алгоритмов и автоматов. При этом формула, определяющая необходимый минимальный объем информации I_0 для представления одного символа из исходного алфавита A , содержащего r символов (с априори разными частотами появления символов в словах), в другом – двоичном («рабочем») алфавите (используемом в большинстве современных систем обработки и передачи информации), записывается следующим образом:

$$I_0 = \log_2 r . \quad (4.2)$$

Соответственно, для передачи сообщения из k символов исходного алфавита A необходимо I_n бит информации:

$$I_n = k \log_2 r . \quad (4.3)$$

Нет ничего удивительно в том, что природа тоже пошла по схожему пути – система кодирования генетической информации, описывающей структуру нужных белков, слабо связана с частотами появления тех или иных аминокислот в этих белках. Например, на самом нижнем уровне кодирования любой аминокислоте независимо от частоты её появления в белках соответствует ровно три нуклеотида. То есть аналогично тому, как любому символу кода ASCII или КОИ-8 отводится ровно 8 двоичных разрядов (8 бит информации независимо от частоты использования символа), в генетическом коде для каждой аминокислоты резервируется ровно три позиции нуклеотидов, называемых кодонами или триплетами. Правда, в отличие от наших сис-

тем кодирования, некоторые аминокислоты в генетическом коде кодируются не одним триплетом, а несколькими. Этому есть ряд объяснений. Во-первых, в генетическом коде не должно быть «бессмысленных» (пустых) кодонов, т.е. таких, которые механизм трансляции генетической последовательности в структуру конкретного белка не могут интерпретировать. Иначе сам этот механизм просто остановится или допустит грубую ошибку, вследствие чего жизнедеятельность клетки прервется. Во-вторых, избыточность кодовых комбинаций (всего возможны 64 различных кодона для 20 основных аминокислот и двух управляющих комбинаций, начинающих и останавливающих процесс трансляции) может использоваться для повышения надежности передачи генетической информации при случайных сбоях в механизме наследования, т.е. при случайных заменах (мутациях) одного нуклеотида на другой. Наконец, в-третьих, – эти особенности кодирования генетической информации могут быть связаны с ещё одной возможностью, а именно – с предпочтительными направлениями мутаций, приводящих к замене одной аминокислоты на другую, т.е. с наиболее предпочтительным выбором возможного кратчайшего эволюционного пути развития (ЭПР) биологической системы к новой функции белка (и, соответственно, к его новой структуре – см. главу 4 и работы [13-15]).

Кроме рассмотренных выше фундаментальных общесистемных концепций информации, к столь же фундаментальным параметрам имеет смысл отнести функциональные возможности систем [47, 51, 59, 62, 83]. Например, Г. Кампис (G. Kampis) вслед за В. Ксани (V. Csanyi), анализируя концепцию информации в биологических системах, выделяет два основных её типа: структурную и функциональную информацию. Структурная информация характеризуется им в терминах взаиморасположения и свойств частей, составляющих макромолекулярную конструкцию. Функциональная же концептуализируется как «информационное влияние структуры на её системные функции» [62, с. 423].

Попытаемся ввести соответствующий общесистемный параметр, используя предложенный в главе 3 формальный подход к представ-

лению функциональных свойств объекта. В соответствии с определением функциональности назовём этот параметр *потенциальной функциональностью* или *функциональной сложностью* системы или объекта, для краткости – просто его функциональностью.

Согласно определениям, (функциональное) состояние объекта O_i можно представить в виде n -мерного вектора, где n – число интересующих нас функциональных свойств объекта O_i (т.е. важных для нас с точки зрения его функциональности). При этом каждая компонента (независимая от других «координата») вектора может иметь столько значений (в том числе нечисловых!), сколько нам нужно для требуемой адекватности нашей модели. Подчеркнём ещё раз, что функциональные состояния в нашем случае могут ассоциироваться не только со статическими компонентами или параметрами объекта. Это вполне могут быть и какие-либо периодически повторяющиеся (стационарные) динамические (но устойчивые, согласно тому же определению свойств и их функциональностей) процессы в самом объекте, его компонентах или параметрах. Например, это могут быть резонансные колебательные процессы, связанные с объектом («резонансное вращение» электрона вокруг атомного ядра по определенной – «разрежённой» – орбите, или же колебания пружины с достаточно малым затуханием – с точки зрения используемого временного масштаба, и т.д.). Все такого рода явления с внешней стороны могут рассматриваться как функциональные свойства объекта и использоваться, согласно определению, как соответствующие компоненты n -мерного вектора интересующих нас функциональных свойств объекта O_i .

Обозначим возможное число различных «значений» j -того функционального свойства через m_j . Тогда максимальное теоретически возможное число всех функциональных состояний Φ_i объекта O_i будет равно произведению различных значений всех компонентов вектора его функциональных свойств:

$$\Phi_i = \prod_{j=1}^n m_j. \quad (4.4)$$

Предположим, что объекты O_i – это элементы базы, общее число которых равно r . При этом будем считать, что все элементы базы – абсолютно разные объекты, каждый из которых имеет *своё* число *своих собственных* (не повторяющихся в других элементах базы) функциональных состояний Φ_i ($i = \{1, 2, \dots, r\}$), и на любом месте в линейной конструкции O_k , состоящей из k элементов, может быть любой из r элементов базы. Тогда максимально возможное число Φ_{max} различных функциональных состояний, приходящихся на одну позицию в такой линейной конструкции, составит величину

$$\Phi_{max} = \sum_{i=1}^r \prod_{j=1}^{n_i} m_{ji} \quad (4.5)$$

4

где m_{ji} – число функциональных состояний j -того компонента вектора функциональных свойств i -того элемента базы, n_i – число этих компонентов для i -того элемента базы.

Для полного описания линейной конструкции O_k , содержащей k элементов, т.е. для полного описания состояния каждой из k её позиций, включая конкретные используемые в данной позиции функциональные элементы базы, потребуется уже I_k бит информации, поскольку мы договорились, что компоненты векторов функциональных состояний разных элементов базы в общем случае могут не совпадать:

$$I_k = k \log_2 \Phi_{max}. \quad (4.6)$$

Логично считать, что такого рода информация полностью устраняет неопределённость, связанную с описанием максимальной потенциальной функциональности линейной конструкции O_k . Иными словами, можно по аналогии с энтропией считать формулу (4.6) определяющей эту самую потенциальную функциональность Φ_k конструкции O_k [13, 15]:

$$\Phi_k = I_k = k \log_2 \Phi_{max}. \quad (4.7)$$

Ясно также, что величина Φ_{max} , вычисленная по формуле (4.5), при наличии в базе только функциональных элементов (даже если число

функциональных состояний для каждого из элементов равно всего двум), превышает величину r . В итоге, сравнивая (4.2) и (4.7), получаем:

$$\Phi_k = I_k = k \log_2 \Phi_{max} > I_n = k \log_2 r. \quad (4.8)$$

Если предположить, что *не все* элементы базы в формулах (4.4), (4.5) имеют функциональность, т.е. значение Φ_i для некоторых таких элементов

$$\Phi_i = \prod_{j=1}^n m_j = 1, \quad (4.9)$$

то всё равно соотношение (4.8) будет выполняться, поскольку остальные Φ_i в (4.4) и (4.5) будут больше единицы [17].

Заметим попутно, что если даже предположить наличие у отдельных символов каких-то свойств (но, естественно, не функциональных, поскольку, если отбросить мистику и возможности анимации, никто пока ещё не видел каким-то образом самостоятельно меняющихся,двигающихся или взаимодействующих букв или цифр), то для любого такого символа с очевидностью выполняется соотношение (4.9) и, соответственно, неравенство (4.8) превращается в равенство. При этом, однако, потенциальная функциональность Φ_k самих объектов, в данном случае – строк символов, отсутствует!

В ряде других работ аналогичный анализ (с аналогичными результатами) проведён с учётом ещё одного важного общесистемного параметра – энтропии [13, 15].

4.2. МОЛЕКУЛЫ КАК КОНЕЧНЫЕ АВТОМАТЫ

Фон Нейман, рассматривая задачу самовоспроизведения клеточных автоматов с математической точки зрения, выделил два типа функциональностей, необходимых, по его мнению, для этого: логическую (булевскую) и конструктивную. При этом под конструктивной функциональностью он понимал возможность создавать схемы, реализующие функции внутриклеточных автоматов, т.е. запоминать своё собственное состояние, определять состояния соседних клеток и реа-

гировать на них переключением своего состояния в соответствии с таблицей переходов [34]. На самом деле эта функция легко реализуется в любом универсальном булевском логическом базисе путём соответствующего соединения электронных логических элементов — как, кстати, и все вычислительные функции современных компьютеров. В дальнейшем под конструктивной полнотой в вышеуказанном смысле также будет пониматься логическая (т.е. булевская, или «вычислительная») полнота. Одновременно вводится очень похожий по звучанию, но принципиально отличный по смыслу термин — *конструкционность*, или *конструкционная полнота*, под которой будем понимать *принципиальную возможность конструирования* в изначально пустой n -мерной среде n -мерной же конструкции.

Определение конструкционной полноты. Под конструкционной полнотой понимается принципиальная возможность реализации в n -мерной среде *любой* конструкции соответствующей (не меньшей n) размерности.

Проанализируем этот вопрос с самых общих позиций с использованием уже упоминавшегося выше «метода назначенных функциональностей».

Будем считать, что мы располагаем в некоей n -мерной среде соответствующими n -мерными (т.е. «круглыми», если $n = 2$, или «сферообразными», если $n > 2$) элементами. В соответствии с ОФТ-концепцией свойств введём символ γ_c для обозначения некоего абстрактного и неизвестного нам свойства, представляющего необходимую для построения n -мерных конструкций неизвестную *конструкционную функциональность*. Легко проверяется, что в этом случае для образования *линейных* конструкций из неких элементов a_i , базы **A**, способных соединяться «слева» (условно) и «справа» от себя с другими (или такими же) элементами, достаточно иметь всего три состояния свойства γ_c .

Обозначим эти состояния цифрами 0, 1, 2 и назначим им следующие функциональности: состояние $\gamma_c = 0$ будет означать, что элемент a_i

готов присоединиться к уже готовой линейной конструкции к её крайнему «условно левому» элементу. Если $\gamma_c=1$, то элемент a_i уже присоединился «слева» к готовой линейной конструкции, т.е. сам стал её «условно левым» крайним элементом и сам теперь готов к присоединению «условно слева» от себя другого (очередного) элемента. Если же $\gamma_c=2$, то элемент a_i уже имеет присоединённые «слева» и «справа» от себя элементы (т.е. полностью встроен в линейную конструкцию) и больше не способен к присоединению других элементов.

Соответствующая таблица переключений состояний γ_c для i -того элемента, описывающая изменение его собственного текущего состояния (момент времени t) и текущего состояния двух его соседних элементов: $(i-1)$ -го «условно левого» элемента, уже входящего в состав конструкции, и пока свободного $(i+1)$ -го элемента, который будет присоединяться к конструкции после соединения с ней i -того элемента, приведена ниже (табл. 4.1). Момент времени сразу после переключения состояний обозначен как $(t+1)$, а прочерк «—» означает отсутствие элемента «слева» или «справа» от i -того элемента.

Таблица 4.1

$(\gamma_c)^{i+1}(t)$	$(\gamma_c)^{i-1}(t)$	$(\gamma_c)^i(t)$	$(\gamma_c)^i(t+1)$
0	2	1	2
—	2	0	1
0	—	1	2

Предполагается, что остальные возможные комбинации состояний соседних элементов не меняют состояние i -того элемента.

Несмотря на кажущуюся простоту, табл. 4.1 переключения состояний на самом деле является исчерпывающей. На рис. 4.1 показаны все этапы наращивания линейной конструкции sba , первоначально состоящей всего из 3-х элементов базы: a, b, c , до конструкции из 4-х элементов $dcba$. Текущие состояния показаны цифрами внутри элементов.

После присоединения элемента d к конструкции sba (первая строчка на рис. 4.1) состояние элемента c переключается в соответст-

вии с первой строчкой табл. 4.1, т.е. из 1 в 2. После этого состояние элемента d в соответствии со второй строчкой табл. 4.1 переключается из 0 в 1. В итоге конструкция становится на один элемент длиннее и снова готова к присоединению очередного элемента e . Заметим, что в данном примере к уже готовой линейной конструкции присоединяется *любой* находящийся вблизи неё элемент, имеющий состояние 0.

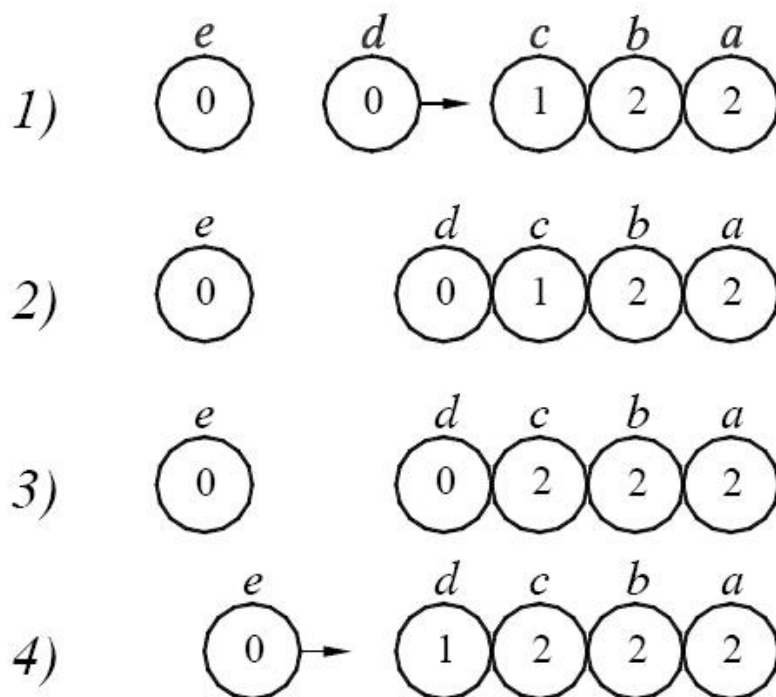


Рис. 4.1. Последовательность 1)...4) изменения состояний элементов в процессе синтеза линейной конструкции

Подчеркнем также особенность переключения элемента a , начинающего конструкцию. Его первичное состояние 0 должно быть каким-то образом переведено в состояние 1, после чего он в принципе может стать способным к присоединению одного элемента с нулевым состоянием, как это следует из рис. 4.1. Однако после такого присоединения состояние элемента a должно смениться на 2, для чего и служит третья строчка табл. 4.1. Ясно, что элемент a должен обладать несколько иными свойствами в сравнении с остальными элементами $b \div d$ линейной конструкции $dcba$, а именно в состоянии 2 он связан только с одним элементом базы и не должен присоединять никакой другой. По этой причине данное уникальное состояние элемента a

имеет смысл обозначить другой цифрой, например, 3. Аналогичным свойством должен обладать и крайний «левый» элемент в конструкции. Такого рода элементы имеет смысл назвать терминальными.

Для того чтобы линейная конструкция могла иметь двух- и трёхмерные конфигурации, необходимы, помимо рассмотренных выше, некоторые дополнительные состояния элементов, входящих в эту конструкцию, которые способны изменять её линейную (при синтезе) структуру. Возможны, по крайней мере, два пути.

Первый – предположить появление соответствующих эмерджентных функциональных свойств, изменяющих линейную структуру конструкции и зависящих от некоторых последовательностей включенных в конструкцию элементов.

Второй – назначить некоторые дополнительных свойства некоторым (или всем) элементам, также меняющим (после синтеза) её линейную конфигурацию при наличии определённой (или определённых) последовательностей элементов. Следуя принципу бритвы Оккама, можно было бы считать достаточным для формирования трёхмерной структуры всего одного дополнительного функционального состояния, которое обозначим цифрой 4. При этом желательно, чтобы «излом» конструкции в соответствующем месте был достаточно резким, т.е. под углом, значительно меньшим 180° , но всё же большим 90° . При этом плоскость поворота угла также должна поворачиваться в пространстве относительно предыдущей (и последующей) плоскостей поворота, иначе конструкция останется плоской.

Таким образом, минимальное число функциональных состояний, обеспечивающих в трёхмерном пространстве трёхмерную конструкционную полноту линейных при синтезе конструкций, оказывается равным пяти: $\gamma_c = \{0, 1, 2, 3, 4\}$.

Для рассмотренного на рис. 4.1 процесса в соответствии с табл. 4.1 можно построить блок-схему алгоритма наподобие алгоритмов вычислительных систем или алгоритмов функционирования так называемых «конечных автоматов». Пример такого алгоритма приведён

на рис. 4.2.2 в [13] и на рис. 4.2 в [15]. Фактически этот алгоритм формально описывает работу некоего абстрактного «молекулярного автомата», который способен самостоятельно сдвигаться вдоль синтезируемой линейной цепочки объектов типа линейной конструкции $dcba$ и «пришивать» к ней очередные элементы базы.

Для такого автомата (который условно обозначим как автомат **A**) по формальным правилам подсчитано число состояний, оказавшееся равным 12, что не так уж много для столь непростого алгоритма (процесса). Для кодирования такого числа состояний достаточно всего четырёх запоминающих элементов типа «RS-триггер».

На рис. 4.2 показаны основные этапы функционирования автомата **A** при синтезе линейной (в момент синтеза) конструкции со случайной последовательностью в ней элементов. Процесс начинается при воздействии на автомат **A** некоего иницирующего фактора γ_0 , благодаря которому автомат присоединяет к себе и удерживает в зоне синтеза элемент a_0 , «переводя» его состояние в 1 (следуя табл. 4.1), после чего процесс идёт в полном соответствии со схемой рис. 4.1, пока какой-либо «сигнал» (внешний «останавливающий» фактор γ_s) не останавливает текущий синтез текущей случайной линейной (при синтезе) последовательности.

Случайность элементов базы в такой последовательности связана с тем, что вблизи места очередной «сшивки» какого-либо элемента базы и наращиваемой линейной конструкции могут оказаться в принципе любые элементы базы **A** – как вследствие броуновского движения самих молекул, так и по причине многих других хаотических процессов в окружающей автомат **A** среде.

Вероятно, с учётом ряда других, не рассмотренных здесь факторов, необходимых для реализации процесса синтеза линейной конструкций типа изображённой на рис. 4.2 из реальных молекул, в действительности автомат **A** может оказаться более сложным устройством, чем приведённые выше оценки.

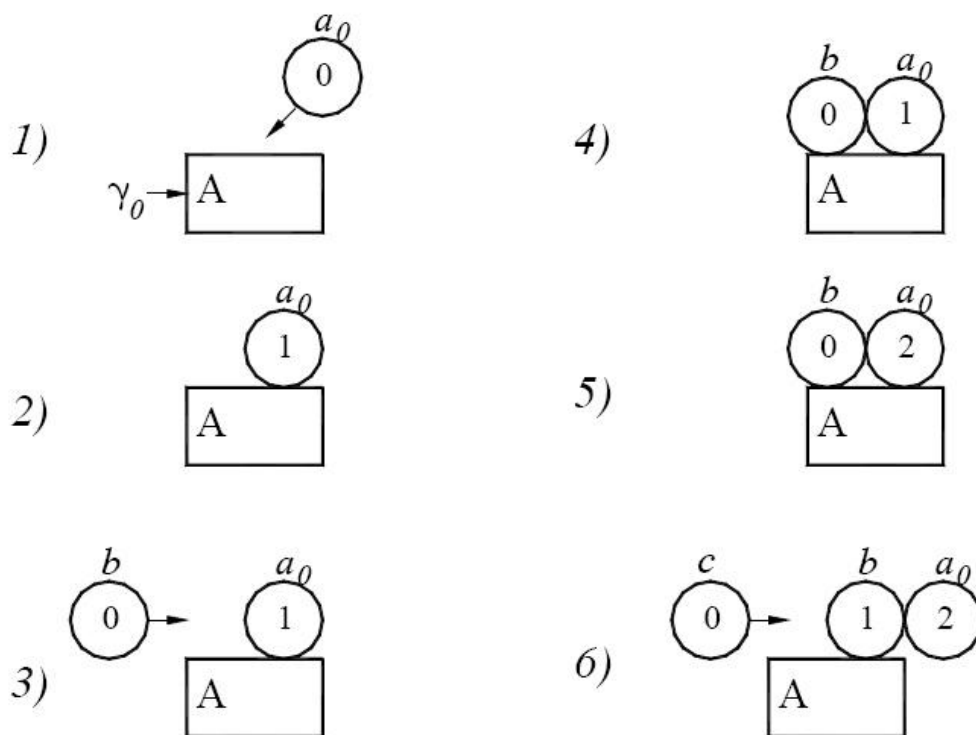


Рис. 4.2. Этапы 1) – 6) синтеза линейной конструкции автоматом $\mathbf{A}$ со случайной последовательностью компонентов

Тем не менее функциональность $\Phi_{\mathbf{A}}$ автомата $\mathbf{A}$, т.е. *описание его структуры, состава и состояния элементов*, даже в этом случае должна быть некоторой конечной и постоянной величиной – константой (для заданного базой $\mathbf{A}$ конкретного набора элементов с подходящими функциональностями и конкретного алгоритма функционирования автомата). **Обозначим эту константу (т.е. функциональность $\Phi_{\mathbf{A}}$) через $S_{\mathbf{A}}$.** Заметим также, что автомат $\mathbf{A}$, работая по схеме соответствующего алгоритма, выполняет не что иное, как операцию примитивного синтеза типа F_S для линейной технологии со случайной последовательностью элементов в синтезируемой конструкции. Конкретный характер этой последовательности определяется, как уже отмечалось, близостью тех или иных элементов базы (находящихся в нулевых состояниях) к месту работы автомата. Назовем его по этой причине *автоматом случайного синтеза*.

Аналогичный подход (представление поведения некоего объекта в виде последовательности действий) широко применяется в последнее время при интерпретации функционирования сложных биомоле-

кул [76]. При этом часто в качестве базовой шкалы, на которую отображаются основные этапы функционирования биомолекул, используется энергетическая шкала. Попробуем проанализировать, чем хорош и чем плох такой метод.

Анализ алгоритма синтеза, приведенного в [13, 15] для схемы, изображённой на рис. 4.2, показывает, что в нём используются преимущественно следующие типы операций: а) «поиск» и «вытягивание» из окружающего пространства определённых типов объектов (в нулевом состоянии), б) целенаправленное изменение их состояний, в) перемещение автомата в пространстве относительно синтезируемой линейной конструкции. Нетрудно видеть, что с точки зрения современной физики при интерпретации элементов конструкции и автомата **A** как неких молекулярных образований все эти типы операций могут быть связаны с энергетическими состояниями компонентов автомата **A** и/или синтезируемой конструкции. С другой стороны, критерием, по которому может выбираться из окружающей среды тот или иной молекулярный объект, является, прежде всего, его пространственная конфигурация и (атомный) состав. С первым из этих двух факторов, т.е. с пространственной конфигурацией, может быть связано также и изменение состояний, и перемещение объектов относительно друг друга. Таким образом, изменение энергетических компонент взаимодействующих объектов на самом деле может присутствовать на всех этапах синтеза, а потому его отображение на энергетическую шкалу носит определённо информативный характер. Тем не менее не следует забывать, что энергетическая шкала – одномерная, и проектирование сложных (трёх- и более мерных) процессов на неё – сознательное и очень сильное упрощение общей картины. Попытки ввести двух- или даже трёхмерные энергетические шкалы призваны в какой-то мере снять этот недостаток [76], однако опосредованный характер роли энергии в конкретных механизмах функционирования молекулярных автоматов типа **A** при этом всё равно остается очевидным – главным является сам алгоритм синтеза и способы его реализации. По крайней мере, в рассмотренном на рис. 4.2 варианте необ-

ходимость учёта энергетического фактора никак себя не проявила. По тем же причинам, кстати, фон Нейман также исключил энергетический фактор из рассмотрения принципов функционирования своих клеточных автоматов [34]. Однако мы знаем (приведенный выше анализ и пример с молекулярной машиной в [76] это подтверждают), что в реальных биомолекулярных системах энергетические факторы играют существенную роль. Возникает вопрос: если эти факторы – не ключевые (с точки зрения логики функционирования), то какие? Почему они вообще необходимы? И почему так часто возникает желание связать, а иногда и отождествить феномен под названием «жизнь» с разного рода «энергетическими потоками»?

Простейший и очевидный ответ на него – потому что любые молекулярные автоматы, как и все реальные автоматы вообще, требуют для своего функционирования энергии. То есть энергетические факторы необходимо и полезно учитывать постольку, поскольку они тесно связаны с работой реальных биомашин в реальном, а не идеальном мире. Но сводить всю сложность и многогранность их функционирования к перераспределению энергии, как мы видели, – неверно. Это и огрубляет, и затеняет реальную картину.

Мы, однако, пойдём ещё дальше и попытаемся прояснить роль энергетических факторов более детально. Согласно определению, технология T называется эволюционной ($ЭТ$ или $T_{ЭТ}$), если в ней: а) возможно получение бесконечного числа новых конструкций, отличных от полученных ранее, и б) это отличие для всех таких новых конструкций может быть определено исключительно средствами самой технологии T .

Условие а) означает бесконечную креативность $T_{ЭТ}$ и достигается достаточно простыми средствами. Условие б) требует гарантии получения в $ЭТ$ конструкций, обладающих определенными интеллектуальными способностями, которые в рамках классических подходов принято сравнивать с возможностями машин Тьюринга или их аналогами.

Ещё раз – теперь уже с позиций формального представления любых технологий согласно их определению – рассмотрим потенциальную эволюционную технологию $T_{ЭТ}$ на начальном этапе своего раз-

вития, т.е. тогда, когда какие-либо конструкции ещё отсутствуют. Поскольку эволюционный процесс предполагается саморазвивающимся, то какие-либо внешние операции над элементами базы в $T_{ЭТ}$ должны быть исключены и множества F_T и F_A – пусты: $F_T = \{\emptyset\} = F_A$. Следовательно, на начальном этапе потенциальная ЭТ может быть записана как $T_{ЭТ} = \langle V \rangle$. При этом для объектов множества V обязательно должны выполняться условия $O_i = \langle G_i, M_i \rangle = \langle \{\gamma_{i1}, \dots, \gamma_{in}\}, \{\gamma_{ij} = \varphi_j(\gamma_{is}, \dots, \gamma_{jk}); \dots; \gamma_{ir} = \varphi_r(\gamma_{it}, \dots, \gamma_{ml})\} \rangle$, иначе объекты, согласно приведенным в главе 3 определениям функциональностей, просто не будут взаимодействовать друг с другом и никакого эволюционного развития технологии $T_{ЭТ}$ не будет. Отметим ещё раз, что заданные в такой форме эволюционные технологии **не могут быть отнесены ни к алгебраическим системам, ни к алгебрам, ни к моделям**. Таким образом, в исходном виде запись ЭТ не соответствует никакой из известных стандартных структур, изучаемых в классической математике!

Если предположить, что исходные элементы базы находятся первоначально в беспорядке в некотором ограниченном пустом пространстве (для определённости – в 3-х-мерном), то для обеспечения выполнения условий а) и б) они должны обладать, по крайней мере, следующими типами функциональностей:

i) динамической (или кинематической) – для обеспечения перемещения друг к другу с целью образования конструкций и их изменения;

ii) конструкционной – обеспечивающей формирование соответствующих многокомпонентных конструкций;

iii) логической – для реализации «систем управления» функционированием конструкций (молекулярных автоматов), выполняющих технологические операции согласно пункту ii), а также для реализации «интеллектуальных функций», требуемых условием б) определения эволюционных технологий;

iv) энергетической функциональностью – для обеспечения энергией функциональностей, перечисленных выше в пп. i)-iii).

Определение конструкционной функциональности и её полноты приведено выше. Логическая функциональность и её полнота обычно (традиционно) связываются с функционально полными наборами булевских функций, которые могут быть реализованы с помощью функциональностей «химического типа» (см. главу 5 и работу [23]).

Что же касается динамической функциональности и её полноты, то один из простейших путей её достижения - дать возможность элементам базы перемещаться в пространстве. Для этого им необходимо придать свойство, определенное в [13-16] как свойство «быть инерционным объектом». Вместе с этим свойством естественно возникает понятие «энергии» [13, 15]. Таким образом, две из рассмотренных выше 4-х функциональностей (динамическая и энергетическая) оказываются жёстко связанными друг с другом. Но и это ещё не всё. В определённом смысле энергетическая функциональность (т.е. возможность запасать, хранить и отдавать энергию) может оказаться полезной именно в качестве базисной «универсальной» функциональности, в которую удобно «превращать» и из которой удобно «извлекать» все другие типы функциональностей. Более того, эту базисную «универсальную» функциональность удобно «наклеить» на эмерджентное, но не функциональное свойство «иметь заданную форму», после чего она естественным образом становится эмерджентной функциональностью, легко трансформируемой во все три остальные функциональности и обратно [15]!

Таким образом, энергетическая функциональность может действительно играть главенствующую роль во всех процессах, связанных с жизнью и её эволюцией, но не на уровне конструктивных, динамических и логических функциональностей, определяющих формы и направления эволюционного развития; и не только на уровне, обеспечивающем энергетические потребности соответствующих молекулярных автоматов; а на гораздо более важном и тонком уровне – на уровне *базисной универсальной функциональности*, обеспечивающей все другие типы функциональностей, а также управление ими и необходимые для этого их взаимные преобразования друг в друга!

Следовательно, энергетическую функциональность можно рассматривать как своего рода аналог универсальной (базисной) булевой функциональности (которая позволяет строить любые компьютеры и выполнять любые вычисления). Эта универсальная функциональность, в отличие от булевой, оказывается пригодной для реализации любых объектов и процессов эволюционирующего мира, включая саму жизнь. Таким образом, сущность знаменитой формулы Эйнштейна $E=mc^2$, связывающей энергию и массу, может оказаться неизмеримо глубже, чем простая констатация фундаментального физического факта для нашей Вселенной.

Некоторые дополнительные обоснования этого тезиса приведены в [13, 15].

На этом же уровне рассмотрения очень большое значение приобретает возможность (и необходимость) квантования энергии. При наличии квантуемости энергии все связанные с ней (или замкнутые на неё) функциональности тоже, естественно, квантуются, и в первую очередь – форма.

Квантуемость энергетической функциональности на достаточно низком («базовом») уровне гарантирует, во-первых, дискретность пространств всех других – производных – типов функциональностей. Во-вторых, – достаточно плотное заполнение соответствующих функциональных пространств функциональными конструкциями и структурами (при достаточно малом шаге квантования). В-третьих, – хорошую повторяемость разных экземпляров функциональностей, реализующих одно и то же функциональное преобразование (например, в разных экземплярах однотипных объектов). Наконец, в-четвёртых, – гарантирует достаточно высокую вероятность наличия комплементарных (сопряженных) функциональностей типа «ключ – замок», в том числе и для конструктивно-энергетических типов функциональностей, когда комплементарность (или «сопряженность») «энергетических поверхностей» двух конструкций приводит к их объединению в одну (временную или стабильную, т.е. устойчивую).

Таким образом, функциональности, необходимые для синтеза линейных (и даже более пространственно разнообразных) конструкций, оказываются не столь уж сложно устроенными. Кроме того, все четыре основных типа функциональностей, необходимые для этого, могут иметь в своей основе один, базовый (универсальный) тип, а именно – квантованную энергетическую функциональность.

4.3. ВОЗНИКНОВЕНИЕ ПРОЦЕССА САМОВОСПРОИЗВЕДЕНИЯ МОЛЕКУЛ

Выше уже отмечалось, что в математических основах теории сложности объектов вводится связь между сложностью какого-либо информационного объекта (например, слова, записанного в некотором алфавите, или числа в какой-либо системе счисления) и описанием процедуры (алгоритма) его получения (т.е. синтеза) [10].

В параграфе 4.2 на основе формально-технологического анализа было показано, что сложность алгоритмов (и, соответственно, функциональная сложность соответствующих автоматов) для случайного синтеза линейных биоподобных конструкций (например, *случайной* последовательности нуклеиновых кислот, т.е. белков с первичной линейной *случайной* структурой) конечна и сравнительно невысока. Величина этой (функциональной) сложности самого автомата A была условно определена как C_A .

В то же время параграфе 4.1 было показано, что величина Φ_k в формуле (4.7), определяющая потенциальную функциональность любых (линейных при синтезе) объектов, растёт с ростом числа k элементов в синтезируемой линейной конструкции O_k . Таким образом, оказывается справедливой так называемая «ОФТ-теорема о точке самовоспроизведения», одна из версий формулировок которой звучит так:

Утверждение 4.1. Существуют технологии, в которых потенциальная функциональная сложность линейных (при синтезе) конструкций постоянно растёт с ростом числа элементов k в синтезируемой конструкции, тогда как функциональная сложность процесса (авто-

мата) случайного синтеза самих таких конструкций остаётся величиной постоянной. Поэтому обязательно найдутся такие (линейные при синтезе) конструкции, потенциальная функциональная сложность которых окажется равной или даже большей потенциальной функциональной сложности автомата их случайного синтеза, т.е. обязательно появятся конструкции, которые могут случайно воспроизводить самих себя. ■

Подчеркнём, что в утверждении 4.1 речь идёт *не о систематическом* самовоспроизведении автоматов типа **A**, а о принципиальной возможности такого – пока *случайного* – самовоспроизведения. То есть такого типа самовоспроизведения, когда автомат **A** случайно синтезирует конструкцию (из случайной последовательности мономеров – элементов базы), алгоритм функционирования которой совпадает с алгоритмом функционирования самого автомата **A**. При этом нет нужды точно воспроизводить саму первичную линейную структуру конструкции автомата **A**. Достаточно лишь, чтобы синтезированная конструкция воспроизводила алгоритм его работы.

Заметим также, что уже на данном этапе возможно использование механизма естественного отбора для выбора оптимальных структур автоматов типа **A**, поскольку чем проще он будет устроен (т.е. чем меньше мономеров – элементов базы будет в его первичной линейной структуре), тем выше будет вероятность его повторения при случайном синтезе и тем чаще в среде буду синтезироваться подобные автоматы.

Теперь зададимся вопросом: а как должна измениться функциональная сложность (т.е. потенциальная функциональность) автомата случайного синтеза S_A , чтобы он смог *воспроизводить* самого себя не случайным образом, а следуя какому-либо варианту своего структурного описания?

Во-первых, для этого, очевидно, нужно изменить сам алгоритм функционирования автомата случайного синтеза, заставив его соединять мономеры в цепочку не случайным образом, а в соответствии с некоторым описанием-программой, содержащей информацию о точной

структуре синтезируемой конструкции. Обозначим такой автомат (т.е. транслирующий или копирующий описание структуры в саму структуру) символом **В**, а необходимую для этого функциональность – C_B .

Во-вторых, для упрощения процесса синтеза автомата **В** желательно, чтобы он также был построен на основе первичной линейной конструкции. Утверждение 4.1 говорит о принципиальной возможности подобной реализации (сложность C_B , очевидно, как и сложность C_A , конечна).

В-третьих, необходимо снабдить автомат **В** с функциональностью C_B программой-описанием собственной структуры. Самый простой путь для этого – сделать эту программу опять же в виде линейной конструкции, изоморфно (то есть один к одному) повторяющей последовательность элементов базы в первичной линейной структуре самого автомата **В**. Ещё лучше – сделать конструкцию-программу так, чтобы до предела упростить поиск в окружающем пространстве элементов базы, нужных для синтеза первичной линейной структуры **В**. То есть попытаться частично вовлечь функциональные особенности конструкции-описания программы синтеза структуры **В** в алгоритм функционирования самого автомата **В**.

В биологических системах подобные функциональности на уровне биохимических автоматов используются очень широко и известны под названием принципа «ключ-замок». Согласно этому принципу, подбор подходящего компонента для взаимодействия двух макромолекул выполняется на основе комплементарности (трёхмерного взаимно-обратного соответствия друг другу) определённой части их «внешней молекулярной поверхности». То есть там, где в одной молекуле есть выступ, в другой имеется как бы углубление тех же размеров и формы, и наоборот – углублению в первой молекуле соответствует выступ аналогичной формы и размеров во второй.

Если вновь обратиться к табл. 4.1, рис. 4.1 и рис. 4.2, то становится ясно, что с точки зрения самой процедуры (алгоритма) синтеза линейной цепочки-конструкции, в автомате **В** вряд ли что-либо радикально изменится. Он, как и автомат **А**, должен «отбирать» из окру-

жающей среды теперь уже нужные (а не случайные) элементы базы и выполнять их сшивку (рис. 4.3). Однако для придания синтезируемой цепочке дополнительных функциональностей, связанных с «подбором комплементарных элементов базы», число необходимых состояний элементов, образующих такую цепочку, должно быть, естественно, увеличено (рис. 4.3).

Заметим, что согласно выражению (4.7) функциональность линейной конструкции растёт тем быстрее, чем больше различных *типов* элементов базы и соответствующих им возможных (различных) функциональных состояний имеет технология. С другой стороны, требование комплементарности элементов базы автоматически удваивает их число, поскольку для каждого из них обязательно должен существовать свой комплементарный «двойник». То есть в простейшем случае технология с «комплементарной функциональностью» должна содержать минимум два типа элементов базы (естественно, комплементарных друг другу). При этом функциональность соответствующих линейных конструкций будет расти пропорционально их длине, но самым медленным из всех возможных вариантов образом – см. формулу (4.5).

Следующий по сложности вариант – это наличие двух пар комплементарных элементов базы при общем их числе, равном четырём. В такой технологии функциональность конструкций в зависимости от их длины будет расти примерно в два раза быстрее, чем в предыдущей (если число функциональных состояний Φ_i дополнительных комплементарных элементов базы будет примерно совпадать с числом функциональных состояний элементов базы в предыдущей версии) – см. формулу (4.5).

Минимальное необходимое для реализации двух пар «комплементарных» функциональностей число состояний элементов базы равно четырём (по два «комплементарных» функциональных состояния на пару). Однако если эти «комплементарные» функциональности «наклеиваются» на элементы базы, которые реализуют упомянутые в табл. 4.1 и комментариях к ней те же 3-5 функциональных состояний, нужных для синтеза самих линейных конструкций (фактически – полимерных цепочек), то общее число возможных состояний

одного элемента может возрасти, согласно формуле (4.4), до 12-20. Это значительно усложнит и сам алгоритм синтеза линейных конструкций, и молекулярный автомат, необходимый для его выполнения. Поэтому было бы проще и лучше, если бы 4 дополнительных комплементарных функциональности были пространственно независимыми от первых пяти, т.е. реализовывались бы другими, геометрически (пространственно) удалёнными от первых, частями элементов, и, естественно, не участвовали бы непосредственно в образовании состояний, необходимых для сшивки цепочки.

На рис. 4.3 приведён пример именно такого «устройства» элементов базы с четырьмя дополнительными попарно-комплементарными функциональными состояниями (отмеченными цифрами 5-6 и 7-8 соответственно), которые конструктивно (и пространственно) отделены от областей, реализующих состояния, нужные для «полимеризации».

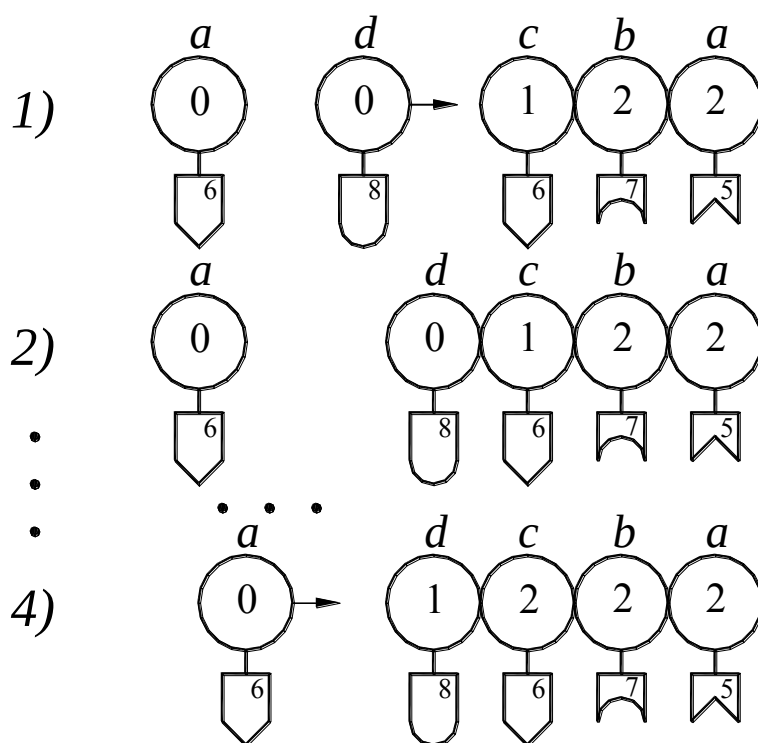


Рис. 4.3. Синтез (например, автоматом случайного синтеза типа A_1) линейной конструкции из элементов, дополнительно снабжённых попарно комплементарными состояниями 5-6 и 7-8 соответственно, которые пространственно отделены от состояний, необходимых для синтеза самой линейной конструкции

Из рисунка видно, что при таком устройстве элементов процесс синтеза самой полимерной цепочки может практически не измениться. Следовательно, автомат случайного синтеза (который обозначим теперь как A_1) для такой цепочки также может иметь близкую или даже равную A функциональную сложность, т.е. близкий к алгоритму работы автомата A алгоритм своего функционирования.

После синтеза автоматом (случайного синтеза) типа A_1 верхней цепочки может сразу начаться синтез комплементарной нижней цепочки другим автоматом, который имеет смысл назвать автоматом комплементарного синтеза или комплементарного копирования. Это – тот самый автомат, который был обозначен как B (рис. 4.4). Особенность его работы в том, что свободные комплементарные состояния 5...8 верхней цепочки (возможно, вкупе с автоматом B) должны как бы «вытягивать» из окружающего пространства свои комплементарные аналоги вместе с прикрепленными к ним фрагментами, образующими под воздействием автомата B нижнюю полимерную цепочку, изображённую на рис. 4.4.

Напомним, что соответствующий процесс удвоения (дублирования) линейной конструкции типа молекул ДНК или РНК в клетках существующих биологических организмов называется репликацией.

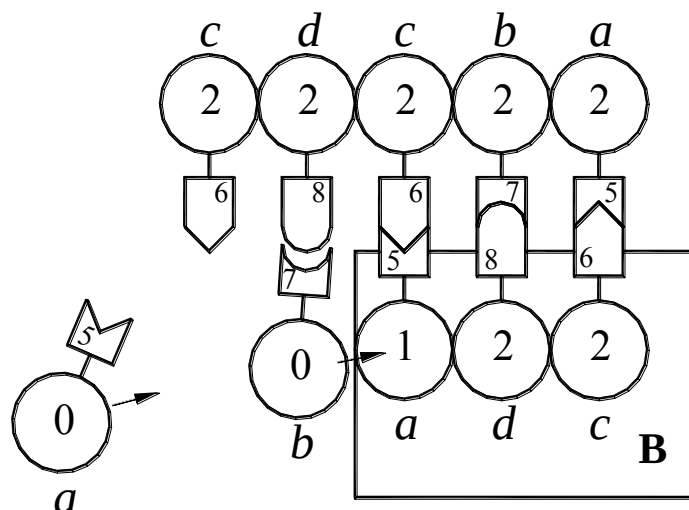


Рис. 4.4. Синтез комплементарной линейной конструкции с помощью автомата B и элементов с «комплементарными хвостами»

Нетрудно проверить, что все состояния табл. 4.1 для всех элементов в нижнем ряду справа налево (без учёта возможных терминальных состояний) рис. 4.4 по-прежнему имеют место и могут быть реализованы. Для начала процесса синтеза достаточно лишь каким-либо образом перевести крайний правый нижний элемент из состояния 0 в состояние 2, а следующий, т.е. ближайший слева элемент, – в состояние 1. Таким образом, алгоритм синтеза линейной конструкции по схеме рис. 4.1 вполне пригоден и для полимеризации нижней цепочки элементов автоматом **В**, изображённым на рис. 4.4. С учётом внешних факторов запуска γ_0 и остановки γ_s процесса полимеризации, а также отсутствия необходимости в специальных терминальных элементах типа a_0 , алгоритм работы автомата **В** синтеза нижней полимерной цепочки будет очень похож на алгоритм, реализуемый автоматом **А** на рис. 4.2. Один из вариантов реализации этого алгоритма приведён в работе [15] на рис. 4.8. Анализ этого алгоритма показывает, что автомат **В** комплементарного синтеза может быть даже немного проще, чем автомат случайного синтеза **А**, поскольку операции поиска очередных элементов линейной конструкции в нём могут реализовываться за счёт комплементарности «хвостов» элементов базы. То есть с определённым «запасом прочности» можно считать, что функциональные сложности C_A и C_B этих автоматов примерно равны:

$$C_A \approx C_B . \quad (4.10)$$

Объём информации, необходимый для точного описания структуры комплементарной линейной конструкции типа изображённых на рис. 4.3, рис. 4.4 при принятых выше допущениях, определяется формулой (4.3). Для рассматриваемого варианта он составляет величину

$$I = k \log_2 4 = 2 k, \quad (4.11)$$

где k – длина конструкции в числе позиций мономеров (элементов базы), а коэффициент $2 = \log_2 4$, т.е. получен из формулы (4.3) с учётом того, что в нашем конкретном случае число различных мономеров r (с четырьмя типами комплементарных «хвостов»), образующих линейные конструкции, равно четырём: $r = 4$.

Очевидно, суммарная сложность C воспроизведения с помощью автомата комплементарного синтеза B линейной комплементарной конструкции, структура которой описывается информацией I , будет равна сумме информации I , необходимой для синтеза соответствующей линейной конструкции, и функциональной сложности C_B самого автомата B :

$$C = k \log_2 r + C_B . \quad (4.12)$$

Сравним формулу (4.12) с формулой из работы Звонкина и Левина²⁸ для колмогоровского критерия сложности K получения (или, говоря языком формальной технологии, «синтеза») линейного информационного объекта (строки символов):

$$K = l \log_2 m + C_0, \quad (4.13)$$

где m – число символов в алфавите, в котором конструируется линейный информационный объект (строка символов), l – длина этой строки в числе символов, C_0 – константа, определяющая сложность механизма (алгоритма, т.е. функции или процедуры, или – как в нашем случае – автомата, реализующего эту процедуру), необходимого для «вычисления» («синтеза») такой конструкции в соответствии с информацией $I = l \log_2 m$, описывающей структуру синтезируемой (линейной) конструкции.

Совершенно очевидна их полная смысловая и структурная идентичность, что вполне естественно, поскольку при построении выражения (4.12) мы руководствовались теми же принципами, которые положены в основу формулы (4.13) для колмогоровского критерия сложности получения линейных объектов типа строк символов.

Некоторые другие аспекты начала самовоспроизведения линейных биоподобных молекул рассмотрены в работах [13-16].

²⁸ Звонкин Ф.Л., Левин Л.А. Сложность конечных объектов и обоснование понятия информации и случайности с помощью теории алгоритмов // Успехи математических наук. – 1970. – Т.25. – № 6. – С. 85-127.

4.4. КРАТЧАЙШИЙ ЭВОЛЮЦИОННЫЙ ПУТЬ РАЗВИТИЯ И СКОРОСТЬ ЭВОЛЮЦИИ НА НАЧАЛЬНЫХ ЭТАПАХ

После появления простейших типов самовоспроизведения биоподобных систем, основанных на генетической памяти, включаются очень мощные механизмы их совершенствования и модернизации, получившие в работах [13-16] название эволюционного пути развития (ЭПР) и кратчайшего ЭПР (КЭПР). Суть последнего заключается в том, что при достаточном числе самовоспроизводящихся или даже частично самовоспроизводящихся биоподобных систем (ЧСБС) в среде их дальнейшее совершенствование идёт весьма быстрыми темпами и по самому короткому из всех возможных (во временном отношении) пути. При этом **эволюционным путём развития (ЭПР)** называется любая последовательность мутаций (изменений) отдельных позиций элементов базы в линейных конструкциях генетической памяти конечного множества копируемых (реплицируемых) линейных конструкций, приводящая к изменению соответствующих функций системы и сохраняющая её жизнеспособность.

Кратчайшим эволюционным путём развития называется соответственно самая короткая цепочка из таких последовательных замен позиций элементов базы в линейных конструкциях генетической памяти, приводящая к нужной функции с сохранением жизнеспособности всей самовоспроизводящейся системы [13-16].

Подчеркнём ещё раз, что и ЭПР, и КЭПР при каждой своей мутации (т.е. при каждом изменении одной позиции элемента базы в текущей линейной конструкции генетической памяти) должны обеспечивать (сохранять) жизнеспособность всей биоподобной самовоспроизводящейся системы, иначе её эволюционное развитие просто прервётся.

Обозначим длительность цикла самовоспроизведения одной ЧСБС через t . При этом в случае нормального хода событий через время t из одной ЧСБС получится две, а через время T из одной ЧСБС получится n её копий:

$$n = 2^{\lceil T/t \rceil}, \quad (4.14)$$

где]...[теперь означает ближайшее *меньшее* целое к содержимому «...», записанному внутри развёрнутых в обратную сторону квадратных скобок.

Все ЧСБС, выполняющие очередной цикл самовоспроизведения, будем называть поколением, не уточняя возраст каждой ЧСБС в данном поколении.

Предположим теперь, что длина линейной конструкции (или их набора), выполняющей функцию генетической памяти ЧСБС, в сумме равна l единицам базы информационной (в самом начале – информационно-функциональной [13-16]) ветви технологии, а число типов самих элементов базы информационной ветви равно r . При этом **информационной** (или **информационно-функциональной**) **ветвью** биоподобной технологии называются её элементы базы, используемые для построения линейных (при синтезе) конструкций, передающих генетическую (или генетическую и функциональную одновременно) информацию от поколения к поколению ЧСБС [13].

Будем считать также, что все изменения в структуре конструкции носят случайный характер и связаны, прежде всего, с изменением наследуемой информации, т.е. с мутациями в последовательности элементов, образующих линейную конструкцию генетической памяти системы. Обозначим через f частоту мутаций (изменения) одной позиции в таких конструкциях. Определим её как величину, обратную вероятности замены одного элемента базы на любой другой, включая пустой элемент (т.е. исчезновение соответствующего элемента базы), и ситуацию с добавлением нового элемента условно «справа» к текущей позиции конструкции. Тогда, согласно введённым выше предположениям, вероятность p любого подобного изменения для каждой позиции элемента базы в реплицируемых конструкциях запишется как

$$p = 1/f. \quad (4.15)$$

При этом возможны следующие исходы:

а) замена элемента одного типа на элемент любого другого из оставшихся $(r-1)$ типов элементов базы;

б) изъятие данного элемента из конструкции;

с) добавление нового (любого из r) элемента «справа» от текущей позиции (понятие «справа» условно – оно взято для определённости исходя из предполагаемой линейности реплицируемых конструкций, чтобы не рассчитывать по два раза вероятности появления новых элементов «справа» и «слева» от текущей анализируемой позиции).

Нетрудно подсчитать, что общее число возможных исходов каждого акта мутации в данном случае равно $2r = (r - 1) + 1 + r$. Возможна также и «пустая» мутация, когда текущий элемент заменяется на самого себя. Мы, впрочем, не будем рассматривать этот вариант. Таким образом, общее число δ вариантов различных исходов, меняющих структуру реплицируемой линейной конструкции, для одной позиции равно:

$$\delta = 2r. \quad (4.16)$$

Естественно, в реальном физическом мире вероятности каждого из перечисленных выше исходов, скорее всего, различны. Тем не менее для простоты оценки будем считать их равновероятными и равными p . Тогда вероятность того, что текущая позиция в конструкции не изменится, будет равна, естественно, разности между единицей и p : $(1 - p)$, а вероятность P_0 того, что ни одна из l позиций в конструкции не изменится, равна, естественно,

$$P_0 = (1 - p)^l. \quad (4.17)$$

Таким образом, для конструкции длиной l вероятность любой мутации p_l (т.е. вероятность того, что конструкция не останется неизменной) будет определяться выражением

$$p_l = 1 - (1 - p)^l. \quad (4.18)$$

С учётом того, что величина p должна мало отличаться от нуля, а значение l для первых ЧСБС ещё не слишком велико, выражение (4.18) можно упростить:

$$p_l \approx 1 - (1 - lp) = lp \quad (4.19)$$

при очевидном условии $1 \gg lp$ (символ $\gg$ означает «значительно больше»).

Рассмотрим случай, когда вероятность $p_l \approx 1/l$, т.е. вероятность мутации некоторого набора линейных конструкций генетической памяти суммарной длиной в l позиций достаточно мала и соответствует наиболее вероятной мутации *ровно одной позиции* в данном наборе конструкций генетической памяти. Тогда для того, чтобы иметь весьма реальный (близкий к единице) шанс получить аналогичное изменение (в среднем в одной из l позиций) в данном наборе линейных конструкций генетической памяти (общей длиной в l позиций), желательно иметь по крайней мере l таких наборов. Действительно, если считать все мутации независимыми, то величина общего числа участвующих в эксперименте конструкций, равная l , гарантирует, что при вероятности обратного исхода, т.е. отсутствия любой мутации в любой позиции в одном наборе суммарной длиной в l позиций, определяемая как $(1 - p_l) = (1 - 1/l)$, даст вероятность P_0 отсутствия такой мутации уже во всех l наборах, равную достаточно малой величине. Действительно, в соответствии с формулой (4.17), получим:

$$P_0 = (1 - 1/l)^l = 1 - l(1/l) + \{l(l-1)/2!\}(1/l)^2 + \dots + (-1)^m \{l(l-1) \times \dots \times (l-m+1) \times 1/m!\} \times (1/l)^m + \dots + (-1)^l (1/l)^l = 1 - l/l + \{l \times (l-1)/2!\}(1/l)^2 + \dots + (-1)^m \{l(l-1) \times \dots \times (l-m+1)/m!\} (1/l)^m + \dots + (-1)^l (1/l)^l = 1 - 1 + \{l(l-1)/2!\} \times (1/l)^2 + \dots + (-1)^m \{l(l-1) \times \dots \times (l-m+1)/m!\} (1/l)^m + \dots + (-1)^l (1/l)^l \approx 0,3 - 0,4 \quad (4.20)$$

ввиду знакопеременного характера членов ряда и их малого значения. То есть такое событие станет достаточно маловероятным (для $l \in [10^2 \div 10^6]$, $P_0 \approx 0,37$) и наоборот – вероятность мутации в среднем одной позиции в каждом из l наборов конструкций генетической памяти будет близка если и не к единице, то к достаточно близкой к ней величине, лежащей в диапазоне значений около 0,63 (для того же $l \in [10^2 \div 10^6]$).

Если же мы захотим, чтобы изменение в каждой позиции могло реализовать все возможные исходы в соответствии с формулой (4.16), то нам потребуется уже минимум $(\delta l + r)$ таких конструкций при примерно тех же уровнях вероятностей. Значение r добавлено, чтобы учесть возможные появления дополнительных элементов базы «сле-

ва» от самого «левого» элемента конструкции, в отличие от уже учтённой возможности появления дополнительных новых элементов «справа» от очередной позиции в случае «с» – см. варианты исходов мутаций после формулы (4.15).

С другой стороны, если вначале имелась всего одна ЧСБС, то в соответствии с формулой (4.16) для получения $(\delta l + r)$ идентичных ЧСБС (при низкой вероятности мутаций) потребуется время T_1 , примерно составляющее величину

$$T_1 \approx t \log_2(\delta l + r) \approx t \log_2(\delta l). \quad (4.21)$$

Для любой ЧСБС (как и вообще для любого биоподобного организма) мутации могут быть *полезными* (т.е. давать какие-либо дополнительные преимущества по сравнению с другими ЧСБС), «*нейтральными*» (т.е. не давать никаких явных «видимых» преимуществ за исключением, может быть, некоторых возможных в ходе дальнейших мутаций потенциально полезных изменений) и «*вредными*» (т.е. приводить в итоге к невозможности реализации самовоспроизведения данной ЧСБС). Последний вариант и назван «мутациями, несовместимыми с жизнью».

Для одних и тех же исходных конструкций и одного и того же набора «конечных» (но *изменённых*) конструкций может быть очень много обычных ЭПР. Более того, с учётом возможных мутаций колебательного характера, когда идёт многократное повторение одних и тех же «прямых» и «обратных» (инверсных) замен, в результате чего конструкция то меняется, то возвращается в исходное состояние, вообще возможно бесконечно большое число таких путей. Из всех таких путей естественным образом отсеиваются ЭПР, дающие конструкции, не совместимые с жизнью. В итоге общее число задействованных на данный момент ЭПР всегда меньше теоретически возможного. Это, кстати, является мощным фактором экономии ресурсов, используемых в развитии ЧСБС, поскольку сразу отсекает использование ресурсов невоспроизводящимися ЧСБС.

Чтобы заставить данный фактор работать ещё эффективнее, необходимо, очевидно, ограничить продолжительность жизни всех ЧСБС с тем, чтобы уже существующие, но не самовоспроизводящиеся особи ЧСБС возвращали накопленные в них ресурсы (элементы базы) обратно в среду. Вообще-то в таких не самовоспроизводящихся особях могут протекать некоторые процессы синтеза новых фрагментов ЧСБС, поэтому спектр последствий деятельности подобных ЧСБС достаточно широк. Он может быть даже полезным в некоторых частных случаях. Однако общая стратегия на прекращение жизнедеятельности ЧСБС через определённое время всё же даёт в итоге более ощутимые преимущества, поскольку позволяет в заведомо ограниченном ресурсном пространстве реализовать весьма быстрый эволюционный прогресс всей популяции.

Число необходимых для этого замен, т.е. исходов типа а), б) и с), естественно назвать «длиной» кратчайшего эволюционного пути. Обозначим эту величину через h . Отметим, что в кратчайшем ЭПР «пустые» мутации отсутствуют, поскольку они ничего не дают, а лишь удлиняют эволюционный путь.

Очевидно, при самом удачном стечении обстоятельств, когда на одно поколение приходится ровно одна (совместимая с жизнью) мутация, точно совпадающая с соответствующей заменой в кратчайшем эволюционном пути, для реализации этого пути потребуется время $T_{min} = ht$ (где t — напомним — длительность цикла самовоспроизведения ЧСБС). При этом вероятность P_{min} такого события при равновероятных возможностях изменения (мутации) каждой позиции в множестве конструкций генетической памяти суммарной длиной в l позиций будет равна возведённому в степень h произведению вероятностей попадания нужной (из $2r$ возможных) мутации в нужную (из l возможных) позицию и вероятности p мутации этой одной конкретной позиции:

$$P_{min} \approx ((1/l) \cdot (1/2r) \cdot p)^h = \left(\frac{p}{2lr} \right)^h. \quad (4.22)$$

Величина P_{min} в выражении (4.22) чрезвычайно мала при более или менее разумных значениях p , l , r и h . Чтобы сделать её ощутимее, необходимо, естественно, резко увеличить число особей, участвующих в «эксперименте». Обозначим это число через m .

Сначала определим вероятность P_{nm} события, когда нужная мутация одной-единственной особи, связанная с реализацией очередного шага ЭПР, не произойдёт. Очевидно, она составит величину

$$P_{nm} = \left(1 - \frac{p}{2lr}\right). \quad (4.23)$$

Естественно, что при наличии m особей вероятность аналогичного события, т.е. что *ни одна* из m особей не промутирует нужным образом, будет равна m -ной степени выражения (4.23). Нетрудно заметить, что с увеличением m это значение, т.е. m -ная степень выражения (4.23), будет уменьшаться, а вероятность противоположного исхода – увеличиваться.

Подчеркнём ещё раз, что нас как раз и интересует вероятность противоположного события, т.е. когда *хотя бы одна* из m особей промутирует должным образом за один шаг ЭПР. Обозначим эту вероятность через P_1 . В соответствии с теорией вероятность P_1 будет равна разности между единицей и вероятностью того, что ни одна из m особей не промутирует каким-либо образом, т.е. разности между единицей и m -ной степенью выражения (4.23). Таким образом, величина P_1 может быть вычислена по следующей формуле:

$$\begin{aligned} P_1 = 1 - \left(1 - \frac{p}{2lr}\right)^m &= \frac{mp}{2lr} - \frac{m(m-1)p^2}{2!(2lr)^2} + \frac{m(m-1)(m-2)p^3}{3!(2lr)^3} - \dots \\ &\dots + (-1)^n \frac{m(m-1)\dots(m-n+1)p^n}{n!(2lr)^n} + \dots + (-1)^m \frac{p^m}{(2lr)^m}. \end{aligned} \quad (4.24)$$

Очевидно, для кратчайшего ЭПР длиной в h шагов соответствующая вероятность P_h составит величину

$$P_h = \left(1 - \left(1 - \frac{p}{2lr}\right)^m\right)^h. \quad (4.25)$$

Из (4.25) следует, что чем меньше величина p , тем больший запас ЧСБС необходимо иметь для того, чтобы получить вероятность реализации в первом же поколении одношагового кратчайшего ЭПР, равную P_1 . При этом не все ЧСБС будут мутировать (подвергаться изменению). Часть из них останутся неизменными (согласно формуле (4.20), примерно 37 % особей при принятых нами допущениях) и через время t эта часть самовоспроизведётся, т.е. удвоится. Одновременно удвоится и та часть изменённых ЧСБС, мутации в которых оказались «совместимыми с жизнью». Среди них, естественно, будет и ЧСБС (с вероятностью P_1), в которой уже реализован интересующий нас одношаговый кратчайший ЭПР. Для того чтобы реализовать теперь следующий шаг многошагового кратчайшего ЭПР с прежней вероятностью P_1 (или, что то же самое, – очередной одношаговый кратчайший ЭПР), в популяции ЧСБС вновь необходимо накопить m особей с изменениями, реализованными на первом шаге. При этом пополнение группы правильно промутировавших ЧСБС будет идти двумя путями: одна часть сформируется за счёт саморазмножения (без мутаций) тех ЧСБС, которые уже прошли первый шаг кратчайшего ЭПР, другая – за счёт «правильных» мутаций саморазмножающихся «исходных» ЧСБС (т.е. не мутировавших ранее). Одновременно будут продолжаться и «эксперименты» (по дальнейшему изменению линейных конструкций генетической памяти) с формирующейся группой ЧСБС, прошедших первый шаг кратчайшего ЭПР.

В самом общем виде, считая начальное число особей, участвующих в реализации кратчайшего ЭПР, равным единице, а максимальное число участвующих в нём особей по-прежнему равным m , время T_h реализации кратчайшего ЭПР длиной в h шагов и с вероятностью, определяемой выражением (4.25), можно вычислить следующим образом:

$$T_h \approx t(h+1)\log_2(m). \quad (4.26)$$

Ограниченность ресурсов среды (сверху) определяется конечным числом активных и пассивных участников реализации кратчайшего эволюционного пути, вытекающим из конечного значения T и соот-

ветствующих формул. Напомним, что введение ограничений на продолжительность жизни ЧСБС сокращает эти ресурсы, хотя и увеличивает, естественно, время, необходимое для восстановления нужного числа конструкций m , требуемых для надежного продолжения реализации кратчайшего ЭПР.

Мы не будем сейчас углубляться в дальнейшие математические выкладки – они уведут нас в сторону от сути рассматриваемых процессов и явлений. Отметим лишь, что формулы (4.21) – (4.26) дают вполне достаточно информации для более глубокого анализа этого процесса и проведения необходимых численных экспериментов.

Рассмотрим соответствующие оценки.

На рис. 4.5 приведены графики зависимости вероятности P_h реализации ЭПР для различных значений m и h , вычисленные по формулам (4.25), (4.26) при следующих значениях: $l = 10^4$, $r = 4$, $p = 10^{-6}$, $t = 10$ секунд.

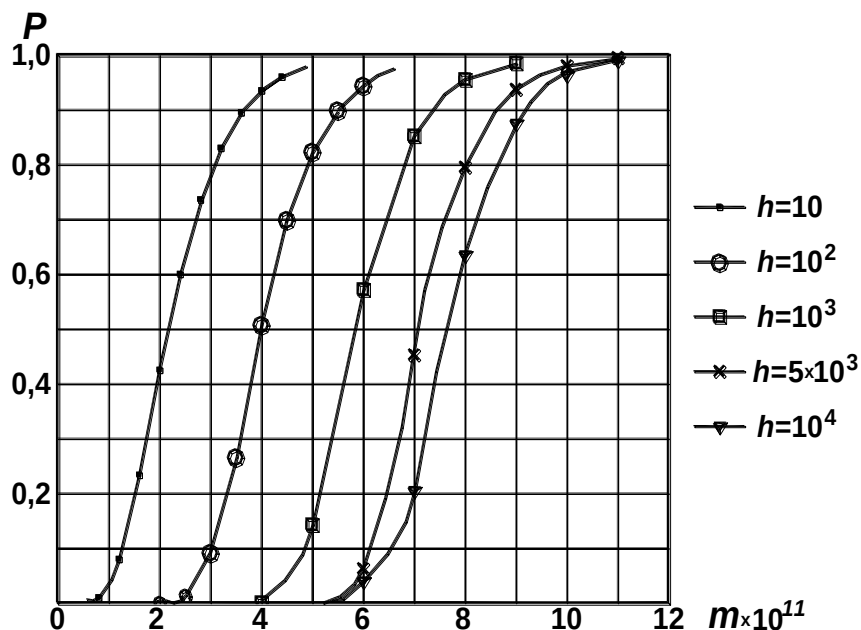


Рис. 4.5. Изменение вероятности P_h при различных m , h

Из графиков видно, что вероятности реализации кратчайших ЭПР не очень сильно зависят от числа особей, участвующих в эволюционном эксперименте, но существенно зависят от времени. Например, для того, чтобы с вероятностью выше 0,5 оптимизировать всего лишь 10 позиций в генетической памяти ЧСБС, насчитывающей 10^4 пози-

ций, необходимо чуть больше 200 миллиардов особей и время для реализации ЭПР чуть больше часа, а для того, чтобы изменить с такой же вероятностью все 10^4 позиций, нужно всего лишь примерно 760 миллиардов особей. Правда, времени для этого потребуется почти в 1000 раз больше.

Оценим теперь возможность получения оптимизированной конструкции из 10^4 элементов базы (при том же значении $r = 4$) методом «полного перебора вариантов». Для этого нам придётся провести $4^{10000} \approx 10^{6000}$ проб. При длительности одной проверки в те же 10 секунд и при том же числе в m конструкций, необходимом для начала реализации процесса оптимизации «методом кратчайшего ЭПР», длительность процесса оптимизации «методом полного перебора» окажется равной 10^{5981} лет!

Столь колоссальное различие в скорости процессов оптимизации реплицируемых конструкций ЧСБС (при всей несхожести конкретных принципов замен позиций элементов базы в конструкциях по методам «кратчайшего ЭПР» и «полного перебора») объясняется в первую очередь следующими факторами:

1) в методе «кратчайшего ЭПР» колоссальное количество конструкций, несовместимых с жизнью, либо вообще не участвует в игре, либо регулярно выбывает из игры на каждом шаге оптимизации и в дальнейшем никак не влияет на продолжение процесса;

2) в методе «случайного перебора» все такие нежизнеспособные конструкции продолжают оцениваться с точки зрения получения на их основе каких-либо потенциально жизнеспособных образований.

По этим двум причинам метод «кратчайшего ЭПР» может «обходить» некоторые конструктивные решения в принципе жизнеспособных конструкций, поскольку все эволюционные пути к ним (т.е. к этим решениям) либо идут через конструкции, несовместимые с жизнью, либо эти конструкции оказываются гораздо дальше (в смысле «длины» эволюционного пути), чем менее подходящие, но в функциональном плане достаточно полноценные конструкции-заменители. (Хотя в дальнейшем, имея в виду непрерывность эволю-

ционного процесса, эти более «отдалённые» оптимальные конструктивные решения могут быть всё же найдены.)

Закономерен вопрос: как сделать так, чтобы число несовместимых с жизнью вариантов (эволюционных путей) было как можно меньше, с тем чтобы механизм «кратчайшего ЭПР» отыскивал максимально большее число возможных оптимальных конструктивных решений? Один из возможных ответов на него – необходимо, чтобы среди элементов базы функциональной ветви биоподобной технологии, которые кодируются соответствующими элементами (или последовательностями элементов) базы, было достаточно мало элементов, резко отличающихся своими свойствами и функциональностями от других. То есть чтобы замена одного функционального элемента на другой не приводила слишком часто к фатальным (несовместимым с жизнью) результатам. Следует, однако, помнить, что это одновременно приведёт и к меньшей функциональной насыщенности набора элементов базы функциональной ветви, т.е. потребует увеличения их числа.

В качестве другого варианта возможно введение специального «среднего» по своим функциональным свойствам элемента базы функциональной ветви, замена на который любого из остальных элементов базы «наименее болезненна», т.е. введение специального промежуточного («буферного») элемента, который даёт наименьший шанс перехода системы в состояние, несовместимое с жизнью.

Ввиду всеобщности законов формальной технологии (являющихся следствием всеобщности основных идей и положений теории алгоритмов) интересно сравнить механизмы реализации метода «кратчайшего ЭПР» с методами совершенствования технологической базы машиностроительных технологий. Как это ни парадоксально, но при тщательном рассмотрении оба этих метода оказываются чрезвычайно похожими. Так, и в первом, и во втором основой дальнейшего прогресса является уже накопленный опыт, в первом случае – в виде «записей» на множестве реплицируемых конструкций, во втором – в ви-

де чертежей, расчётов и описаний действующих механизмов и систем. И там, и тут совершенствование этих механизмов и систем идёт путём постепенной замены отдельных узлов и блоков на новые или/и путём добавления сравнительно небольшого числа новых блоков с новыми связями. Однако такая замена всегда происходит в направлении, сохраняющем работоспособность системы в целом (т.е. «мутации» идут в направлении, сохраняющем «совместимость с жизнью» проектируемой системы). И там, и тут преимущества получают (с определёнными оговорками) наиболее эффективные системы, в первую очередь – более качественно и быстро выполняющие свои функции с использованием меньших ресурсов и т.д. Отметим, что и там, и тут трудно «ввести в оборот» конструктивные решения принципиально нового типа, т.е. такие, которые получены каким-либо образом из областей, либо стоящих далеко впереди на эволюционном пути развития, либо «отгороженных» от традиционных технологий «несовместимыми с жизнью» эволюционными путями и потому «недостижимыми» для большинства действующих субъектов. Интересно также, что в данном контексте спонтанный акт изменения одного элемента реплицируемой линейной структуры по своей «изобретательской» значимости может оказаться эквивалентным «гениальному озарению» инженера, разрабатывающего новейшую технологическую установку. Вся разница лишь в том, что в первом случае неверные решения отсеиваются в ходе самого процесса реализации кратчайшего эволюционного пути, а во втором они уже прошли некоторый предварительный (но не окончательный!) отсев в мозге инженера с помощью некоторой внутренней модели того, что должно происходить на самом деле.

Анализ других возможных аспектов развития ЧСБС, выполненный в аналогичном формально-технологическом ключе, приведён в работах [13-16].

Глава 5. ГЕТЕРОГЕННЫЕ СИСТЕМЫ

Гетерогенные системы включают в себя очень разнообразные типы и классы систем. Практически все создаваемые природой и человеком системы и устройства при ближайшем рассмотрении оказываются в той или иной степени гетерогенными, поскольку охватывают самые разнообразные и разнородные подсистемы, будь то, например, простейшая клетка, многоклеточный организм, растение, животное или наше жилище, одежда, инструменты, станки, приборы, самолёт, ракета, автомобиль или всё тот же компьютер. Несмотря на то, что почти все его основные «вычисляющие» механизмы реализованы на гомогенных логических элементах, обрабатывающих однородные по своему характеру электрические сигналы двух уровней напряжения – «0» и «1», все остальные его компоненты – клавиатура, дисплей, блоки питания, вентиляторы – работают на иных принципах, с другими типами сигналов, сделаны из других веществ и по другим технологиям, т.е. гетерогенны по отношению и друг к другу, и к элементам, выполняющим основную – вычислительную – функцию.

В отличие от нас живая природа максимально широко использует именно гетерогенные системы в виде определённых цепочек химических реакций с различными по своим физико-химическим свойствам молекулами. С помощью таких (гетерогенных по своей сути) реакций-преобразований природа реализует многие сложные функции, включая управление процессами, связанными с самовоспроизведением, самостабилизацией, регенерацией, ростом, с поведением различных биологических подсистем и существ и т.д.

Как ни странно, выясняется, что к гетерогенным системам имеет прямое отношение и такая, казалось бы, далёкая от материальной подоплёки проблема, как искусственный интеллект. Действительно, в предыдущих главах книги рассматривались примеры алгоритмов и технологических систем, способных автоматически анализировать

любые физические (т.е. гетерогенные – см. параграф 1.1) объекты соответствующей (полной) технологии и воссоздавать их точные копии. Возникает естественный вопрос: а нельзя ли аналогичными (интеллектуальными!) возможностями наделить некие искусственные существа (например, роботов), которые могли бы изучать окружающий их мир (похожий на наш, или, наоборот, совершенно иной, не имеющий с нашим ничего общего), получая о нём вполне адекватные знания?

Ответ на этот вопрос положительный. Соответствующие результаты приведены в работах [13 – 16]. Здесь же мы коснёмся только нескольких граней этой задачи, с которой традиционно связывают интеллектуальные возможности и компьютеров, и человека, а именно – попытаемся проанализировать «интеллектуальный потенциал» машины Тьюринга. Как уже было сказано, оказалось, что к интеллекту имеют прямое отношение именно гетерогенные системы, т.е. такие, где реальные физические объекты представлены в форме (1.3), а не в форме (1.4). В итоге с помощью всё той же общей формальной технологии мы попытаемся разобраться в ещё одной очень важной для математики проблеме. Именно с ней многие учёные и связывают не только познавательные (когнитивные) способности автоматов вообще и компьютеров в частности, но и границы познавательной эффективности самой математики. Называется эта проблема «Тезис Чёрча».

5.1. ТЕЗИС ЧЁРЧА И «УНИВЕРСАЛЬНЫЙ ПОЗНАВАТЕЛЬ»

Тезис Чёрча или, как его часто называют в последнее время, тезис Тьюринга-Чёрча²⁹, – одна из самых интригующих гипотез математики 20-го века. В формальной версии он звучит так:

Тезис Чёрча. Класс алгоритмически (или машинно-) вычислимых частичных числовых функций совпадает с классом всех частично-рекурсивных функций [26].

²⁹ Copeland, B.J. The Church-Turing Thesis. In: E. Zalta, ed., The Stanford Encyclopedia of Philosophy, 1998. <http://plato.stanford.edu/>

Поскольку все реально вычислимые человеком функции, как доказано в математике, есть композиции частично-рекурсивных функций (ЧРФ) [26], и более того – все функции, вычислимые человеком в интуитивном смысле, также сводятся к композициям ЧРФ, получается, что в неформальной трактовке тезис формулирует признаваемую многими учёными в качестве истинной гипотезу о том, что все функции, вычислимые человеком (в том числе и в интуитивном смысле), есть, по сути, частично-рекурсивные функции, вычисляемые машинами Тьюринга (м.Т.). Таким образом, в неформальной трактовке тезис утверждает, что класс всех вычисляемых человеком функций совпадает с классом частично-рекурсивных функций, вычисляемых в м.Т. Это, в свою очередь, порождает идею о том, что умственные способности человека и интеллектуальные возможности м.Т., а следовательно – компьютеров, эквивалентны.

К сожалению, понятия «интуитивно вычисляемые функции», как и «функции, вычисляемые человеком», в строгом математическом смысле неформализуемы. Поэтому среди философов и математиков до сих пор идут дискуссии о том, соответствует ли «интеллект» компьютеров (потенциальные вычислительные возможности которых, как многократно доказано, соответствуют потенциальным вычислительным возможностям м.Т.) человеческому интеллекту [56, 72].

Вместе с тем в ряде работ [13, 20, 56, 72] предприняты попытки показать, что возможности некоторых типов вычислительных систем, включая нейрокомпьютеры и системы, использующие *вычислительные операции с действительными числами* либо различные *физические эффекты*, включая так называемые «квантовые компьютеры» [20, 56, 72], превышают вычислительные возможности (мощность) м.Т. Такие системы названы «гипервычислительными» [56].

Для корректности изложения, точности ссылок и лучшего понимания обсуждаемого вопроса приведём полный текст соответствующего утверждения ОФТ и суть его доказательства:

Утверждение 5.1. Частично-рекурсивные функции над натуральными (и рациональными) числами со всеми их свойствами предста-

вимы в общей формальной технологии. Обратное неверно (т.е. не все формальные технологии со всеми своими свойствами представимы в классе частично-рекурсивных функций).

Суть его доказательства заключается в следующем:

Истинность первой части утв. 5.1 легко доказать, продемонстрировав, что с помощью сравнительно простой технологии можно моделировать работу некоторой универсальной машины Тьюринга и, следовательно, процесс вычисления любой частично-рекурсивной функции. В качестве подходящей технологии для моделирования работы универсальной двух-символьной м.Т. выбирается технология $T = \langle A, F \rangle = \langle \{\square, a\}, \{F_T(x), P_0(x)\} \rangle$, где $\square$ – пустой элемент базы A , a – непустой: $a \neq \square$; $F_T(x) = x$ – операция, повторяющая объект x , а $P_0(x)$ – операция анализа, принимающая значение «1», если $x=a$, и «0», если $x = \square$. Каждому из символов в моделируемой двухсимвольной м.Т. ставится в соответствие элемент $\square$ или a . Тогда их «чтение», «печать» новых символов из $\{\square, a\}$ и «различение» можно осуществить с помощью операций F_T и P_2 соответственно (подробно доказательство приведено в [13]). Поскольку двухсимвольные м.Т. эквиваленты по своим вычислительным возможностям (т.е. по своей «мощности») в том числе и универсальной м.Т., способной моделировать работу любой м.Т. [26], первая часть утв. 5.1 оказывается доказанной.

Чтобы показать истинность обратного утверждения, вводится технологическая операция анализа с неспецифицированным (т.е. неизвестным) алгоритмом (или законом) функционирования – наподобие оракулов, введённых в теорию алгоритмов ещё Аланом Тьюрингом³⁰. Такие операции широко использовались человеком (и используются до сих пор) в самых различных процессах (алгоритмах) познания и освоения окружающего мира, в частности операции определения цвета и формы наблюдаемых объектов, вкуса и запаха вещества и т.д. и т.п. Действительно, до недавнего времени конкретные механизмы преобразования и обработки такого рода информации в раз-

³⁰ Turing A.M. Systems of Logic Based on Ordinals // Proceedings of the London Mathematical Society, series 2. – 1939. – Vol. 45. – P. 161-228.

личных рецепторах или в нейронах головного мозга человека были неизвестны. Сейчас многие из подобных операций могут выполняться в том числе и техническими устройствами, однако не секрет, что в период становления основ современной науки таких устройств ещё не существовало. Тем не менее этот факт не помешал успешному развитию науки. То есть введение упомянутых выше формально-технологических абстракций операций анализа с научной точки зрения вполне допустимо и оправдано. Что же это в итоге даёт?

Вполне очевидный результат – поскольку такого рода операции есть, по сути, формально-технологическое воплощение оракулов в алгоритмах Тьюринга, которые не имеют реализации в системе ЧРФ, формально-технологические алгоритмы с подобного рода операциями анализа в состоянии решать задачи, не решаемые в классе ЧРФ и – согласно тезису Тьюринга-Чёрча – в классе функций, вычислимых м.Т.

Таким образом, утв. 5.1 можно считать доказанным в полном объёме. Отметим при этом, что в работе [13] приводится конкретный пример из физики, подтверждающий и поясняющий данный факт.

Иные по содержанию утверждения, отличающиеся от утв. 5.1, но также доказывающие *неэквивалентность* некоторых формально технологических систем и системы частично-рекурсивных функций, приводятся в работах [20, 56, 72]. В частности, в работе [20] доказывалось существование иерархии среди универсальных по Тьюрингу дискретно-аналоговых машин, связанной с неразрешимостью некоторыми такими машинами ряда сравнительно простых задач на множестве действительных чисел по различению и генерированию с некоторой разумной конечной точностью (подчеркнём – не абсолютной!) различных аналоговых уровней и сигналов. Этот результат совпадает с рядом результатов, приведённых в работе [56] и связанных с использованием действительных чисел либо в качестве операндов в вычислениях, либо в качестве некоторых параметров таких вычислений. Поскольку мощности множеств действительных и натуральных или рациональных чисел (с последними из которых только и может работать м.Т.) разные, то и получение принципиально разных результатов

в вычислительных системах, работающих в этих множествах, не должно вызывать удивления. Как следует из работы [20], аппарат ОФТ предлагает для рассмотрения соответствующих вопросов вполне подходящие средства, не используемые в системе ЧРФ или машинами Тьюринга. То есть аппарат ОФТ не имеет тех формальных *технологических ограничений*, какие присутствуют в неявном виде в модели м.Т. или в системе ЧРФ.

Обратим внимание теперь на то, что и сама концепция оракулов была введена А. Тьюрингом именно для того, чтобы обойти ситуации, когда математика в силу своих *технологических ограничений* не позволяет получить ответ на поставленный вопрос. В формальной технологии такого рода ограничения снимаются, как мы уже видели, введением соответствующих технологических операций (в том числе аналогового сложения, вычитания и интегрирования) и некоторых операций анализа.

Наконец, в параграфе 1.1 данной книги и в работе [18] приводятся обоснования существенного отличия формального представления физических и математических объектов в различных алгоритмических процедурах, хотя сами системы, формально описывающие эти процедуры, имеют одинаковую структуру, эквивалентную алгебраическим системам в математике. Напомним ещё раз, что, согласно этому представлению, любой физический объект O_i в таких процедурах может быть представлен так:

$$O_i = \langle \Gamma_i, \mathbf{M}_i \rangle = \langle \{\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in}\}, \{\gamma_{ij} = \varphi_{ij}(\gamma_{sk}, \gamma_{il}, \dots, \gamma_{sm}); \dots\} \rangle, \quad (5.1)$$

где $\Gamma_i = \{\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in}\}$ – множество числовых и нечисловых параметров, представляющих различные физические свойства объекта O_i , благодаря которым мы отличаем этот объект O_i от другого физического объекта, например, O_k ; $\mathbf{M}_i = \{\gamma_{ij} = \varphi_{ij}(\gamma_{kt}, \gamma_{il}, \dots, \gamma_{sm}); \dots, \gamma_{in} = \varphi_{in}(\gamma_{ip}, \gamma_{if}, \dots, \gamma_{zq})\}$ – набор функций (функциональных зависимостей, алгоритмов, физических законов), описывающих процессы взаимодействия различных физических объектов через их различные физические свойства между собой. Как уже не раз подчёркивалось, такая форма

представления объектов очень сильно коррелирует с представлением объектов в наиболее успешной области современной прикладной вычислительной математики, а именно в объектно-ориентированном программировании (ООП), что лишний раз доказывает её эффективность и целесообразность.

В аналогичной (5.1) или (1.3) нотации формальная запись для любого математического объекта x выглядит так:

$$x = \langle \xi \rangle, \quad (5.2)$$

где ξ тоже трактуется как некий *числовой* параметр, *определяющий конкретное числовое значение нефизического, а потому нефункционального свойства опять же нефизического абстрактного объекта x гомогенного характера* [18].

Существенное различие в сущности физических свойств реальных объектов и нефизических количественных свойств абстрактных математических объектов носят гораздо более фундаментальный характер, чем даже различия в мощности множеств действительных и рациональных (натуральных) чисел, приводящих к разным алгоритмическим возможностям соответствующих вычислительных систем [20, 56], поскольку, как уже подчёркивалось в параграфе 1.1, метр никак нельзя сравнивать с килограммом и тем более – с любым числом, представляющим какое-либо количество.

Суммируя все приведённые выше факты, можно сделать следующий вывод:

Вычислительная математика как «технология» целых (натуральных) и рациональных объектов-чисел (гомогенного характера) имеет вполне конкретные ограничения, связанные со специфическим контекстом формулировки тезиса Тьюринга-Чёрча. Эти ограничения касаются основных математических объектов – чисел и кодов, которые не обладают никакими физическими свойствами и функциональностями. В то же время собственно математические (точнее, алгоритмические) принципы и методы могут с успехом применяться (в том числе в формально-технологической трактовке, а также самим человеком) к гораздо более широкому классу объектов, включая объекты

из множества действительных чисел и физические объекты гетерогенного характера, обладающие различными свойствами и функциональностями, в том числе и такими, внутренние механизмы которых нам до сих пор не ясны или непонятны. Таким образом, наше восприятие окружающего нас мира существенно отличается от того, что можно сделать в рамках системы ЧРФ, сколь бы сложными и изощрёнными соответствующие вычисления (т.е. алгоритмы) в системе ЧРФ ни были. Именно на это указывают утв. 5.1 и результаты работ [13, 18, 56, 72].

Обратим ещё раз внимание на то, что мозг человека выполняет многие операции, которые могут быть отнесены к операциям анализа в утв. 5.1, именно так, как это делают оракулы – т.е. не понимая до конца их внутренних механизмов!

Следовательно, можно сформулировать следующее утверждение:

Утверждение 5.2. Интеллектуально-когнитивные способности человека превышают возможности машины Тьюринга.

Иными словами, интеллект человека вкупе с его органами, включая мозг, различные органы восприятия, двигательные органы, превышает «интеллект» (мощность) м.Т. и, соответственно, всех существующих компьютеров.

Доказательство этого утверждения напрямую связано с результатами утв. 5.1 и некоторыми дополнительными фактами. Действительно, согласно утв. 5.1, не все формальные технологические системы могут моделироваться в классе частично-рекурсивных функций и, соответственно, вычисляться в м.Т. В первую очередь это невозможно для таких полных технологий, в которых существуют операции анализа некоторых свойств физических объектов с неясным или не совсем детализированным алгоритмом функционирования, например, операции типа «случайного стационарного отображения», операции определения формы объекта, классифицирующие операции, операции поэлементного анализа и др. – см. табл. 1.1 в работе [13] и комментарии к ней.

Согласно определению, приведенному в [13], а также в параграфе 1.3, «случайность» упомянутого стационарного отображения есть, по

сути, признание того факта, что нам на самом деле неизвестен детальный механизм (функция) преобразования некоторого физического свойства объекта в тот параметр, который «измеряет» операция анализа. А его «стационарность» означает, что такой скрытый механизм на самом деле существует и что он исправно срабатывает всегда, когда для анализа предъявляются одинаковые (или одни и те же) конструкции, дающие, естественно, одинаковые результаты. Аналогичные соображения приводятся и в отношении других типов упомянутых выше операций анализа [13].

Нетрудно, как уже упоминалось, прийти к заключению, что наши органы чувств реализуют именно такие операции, т.е. операции со скрытым («неизвестным») механизмом функционирования. Действительно, до сих пор до конца не ясны на клеточном и молекулярном уровнях многие тонкие механизмы работы головного мозга, обеспечивающие запоминание и узнавание различных объектов и ситуаций. Следовательно, мы не можем их в точности воспроизвести в компьютерах или в имитационных моделях. Тем не менее мы с успехом пользуемся этими своими способностями в различных жизненных ситуациях, включая интеллектуальное освоение окружающей нас среды. Более того, в самом начале промышленной и научной революции наши знания о механизмах функционирования наших органов чувств были еще меньше. Однако это не помешало нам сформулировать и проверить оказавшиеся корректными предположения относительно как основных законов функционирования наших органов чувств, так и основных законов физики.

Кроме неизвестных или частично неизвестных (по механизмам своего функционирования) операций анализа, в состав операций, гарантирующих полноту (фактически, в переводе с ОФТ – познаваемость) технологий, перечисленных в табл. 1.1 работы [13], входят операции, позволяющие выполнять физические действия с конструкциями технологии. Это операции типа декомпозиции и синтеза. С их помощью мы можем создавать различные *физические* приборы и установки, в том числе и на основе пока неизвестных или непонятных

нам *физических эффектов и явлений*, которые в ходе их изучения часто описываются с использованием уравнений *с действительными числами*. То есть и наши интеллектуальные способности, и наши математические гипотезы с лёгкостью оперируют абстракциями *действительного числа*, которые *непредставимы* в известных нам моделях м.Т. или в системе ЧРФ.

В итоге получается, что все перечисленные выше операции, необходимые как для более полноценного в сравнении с системой ЧРФ восприятия сущности физических объектов (т.е. для определения их «образа» – их формы, цвета, запаха, вкуса, сопровождающих звуков, температуры, характерных движений и т.д. и т.п.), так и для достижения полноты (т.е. для познания) любой технологии за счёт операций типа синтеза и декомпозиции (реализуемых нашими двигательными органами, в первую очередь, руками), выполняются соответствующими органами человека вкуче, разумеется, с основным органом обработки информации – его мозгом. Причём все эти операции могут выполняться человеком как без помощи каких-либо внешних инструментов и приспособлений, так и с помощью таких орудий и инструментов – от самых простых до весьма сложных. Ещё раз напомним, что многие тонкие внутренние механизмы таких операций, выполняемых нашим организмом на субклеточном и молекулярном уровне, нам до сих пор неизвестны, и, следовательно, не могут быть в деталях промоделированы (вычислены) компьютерами (или м.Т.). Тем не менее в случае отсутствия каких-либо нарушений в психическом и физическом здоровье человека они позволяют получать достоверную информацию об окружающем нас мире, способствуя его постепенному познанию и пониманию.

Указанные факторы, то есть: а) эффективное использование человеком физических аналитических операций со «стационарными» результатами и неизвестными до конца алгоритмами функционирования, б) эффективное использование операций физического воздействия на исследуемые объекты или их физического взаимодействия на основе абстракции действительных чисел (или других «невычис-

лимых» в м.Т. абстракций), и в) наличие физических свойств и связанных с ними функциональностей у окружающих нас физических объектов выводят, в итоге, наши когнитивные и интеллектуальные способности за рамки ограничений, присущих м.Т., и, соответственно, за рамки ограничений, присущих компьютерам и системе частично-рекурсивных функций. Конец доказательства.

Этот факт неплохо согласуется с известным утверждением Ю. Манина: «В действительности класс задач, доступных классическим средствам, в некотором трудно уточняемом смысле строго шире класса задач, решаемых алгоритмически»³¹. В приведённых выше доказательствах нам удалось сузить эту самую область «трудно уточняемого смысла» до вполне конкретных и вполне формализуемых понятий.

В заключение следует отметить, что, возможно, кое у кого из читателей возникнет определённое неприятие попытки соединить в вышеприведённом доказательстве, казалось бы, трудно формализуемые когнитивные и интеллектуальные особенности человека и достаточно формальные моменты, связанные с использованием логико-математического аппарата ЧРФ и ОФТ. Однако трудно не заметить, что данная попытка предпринята на уровне логических концепций, гораздо более близких к формальным, чем это делалось когда-либо ранее. Автору, по крайней мере, аналогичные попытки неизвестны. Да и спорить, на его взгляд, с приведёнными выше мотивами будет гораздо сложнее, чем на существующем уровне в основном гипотетических философско-гуманитарных рассуждений о потенциальных возможностях человеческого интеллекта.

Таким образом, человек может эффективно использовать в своих алгоритмах эффекты и процессы, напоминающие функционирование оракулов в алгоритмах Тьюринга. Кроме того, он с успехом применяет во многих своих построениях и действиях абстракции, связанные как с множеством действительных чисел, так и с другими «невычислимыми» в м.Т. понятиями и концепциями. Ничего этого пока не в состоянии де-

³¹ см., например, *Манин Ю.И.* Математика как метафора. – М.: МЦНМО, 2008. – С. 61.

лать ни система ЧРФ, ни м.Т. Следовательно, для приближения уровня интеллекта моделей типа м.Т. к человеческому необходимо приложить усилия в указанных выше направлениях. Как было показано выше, а также в ряде упомянутых в этом параграфе работ, соответствующие алгоритмы процессов над физическими объектами с операциями, выходящим за рамки объектов и операций системы ЧРФ и м.Т., дают весьма достоверные результаты и позволяют нам эффективно строить более адекватную модель окружающего мира, т.е. осуществлять его вполне полноценное *познание* [13, 15, 18, 65].

В частности, уже на базе ОФТ-моделей типа синтезаторов-анализаторов из параграфа 2.1 (см. рис. 2.1, рис. 2.3), в том числе с использованием м.Т., может быть построено устройство, способное автоматически познавать окружающий его мир, устроенный как по законам физики нашей вселенной, так и по каким-то другим, не известным нам законам, отвечающим аксиомам метафизики, приведённым, например, в работе [18]. Схема такого устройства для одномерной познаваемой среды (линейного массива ячеек хранения) показана на рис. 5.1.

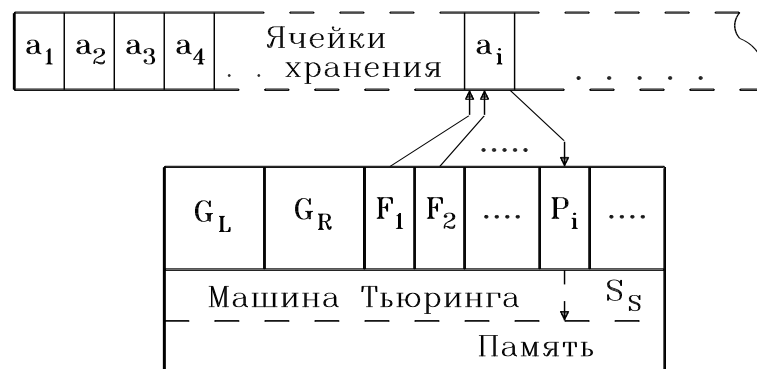


Рис. 5.1. Структура «универсального познавателя» – синтезатора-анализатора U_S' с «динамическими» операциями его перемещения G_L и G_R вдоль одномерного массива ячеек хранения

В отличие от универсальных систем типа S_U (рис. 2.1) и U_S (рис. 2.3), в которых перемещение вдоль массива ячеек хранения выполнялось с помощью «указателя», теперь сам синтезатор-анализатор U_S' может перемещаться вдоль неподвижного одномерного массива таких ячеек с помощью специальных «динамических» операций G_L и

G_R . Кроме того, U_S' может выполнять некоторые аналитические операции (прежде всего типа P_A и P_S) над конструкциями, хранящимися в обозреваемой в текущий момент времени ячейке хранения, без извлечения этих конструкций из ячеек. Ясно, что никакого преимущества в плане аналитических способностей U_S' это не даёт (как, впрочем, и S_U или U_S по сравнению с U_S'). Во всём остальном U_S' полностью аналогичен U_S , т.е. реализует то же множество технологических операций F , определённых для тех же элементов базы A . Операции сдвига «указателя» на одну позицию вправо или влево соответствует (в U_S') операция перемещения всего синтезатора-анализатора на одну позицию вправо (G_R) или влево (G_L) соответственно. Для полноты набора динамических операций будем предполагать также возможность «пустой» динамической операции G_0 , которая означает: «ничего не делать». Как и U_S , U_S' может запоминать на ленте приданной м.т. значения параметров («образы») различных операций анализа типа P_i .

Для приведённых выше уточнений справедливо следующее:

Утверждение 5.3. При идентичности содержимого ячеек хранения и реализуемых в S_U , U_S и U_S' технологических операций все такие синтезаторы-анализаторы могут моделировать (имитировать) работу друг друга, т.е. их «алгоритмические возможности» («мощность») эквивалентны. ■

Операции сдвига синтезатора-анализатора U_S' на одну позицию (ячейку) можно рассматривать как своеобразные технологические операции, образующие свою, отдельную группу «динамических» операций (наряду, например, с группами операций анализа, синтеза или декомпозиции), причём использовать эти операции U_S' может независимо от остальных. При этом оказывается справедливым следующее утверждение:

Утверждение 5.4. Если в синтезаторе-анализаторе U_S' с частично неизвестной технологией $T = \langle B, \{F_1, \dots, P_A, \dots, P_j, G_L, G_R\} \rangle$ (или $T = \langle B, \{F_1, \dots, P_S, \dots, P_j, G_L, G_R\} \rangle$), где P_A (или P_S) известна, к группе неизвестных динамических операций имеется доступ, независимый от остальных технологических операций, а линейный массив ячеек хра-

нения содержит непустые ячейки с конструкциями $b_i \in \mathbf{B}$, отличающимися по $P_A (P_S)$, и хотя бы один конечный непрерывный массив пустых ячеек, причём значение по $P_A (P_S)$ для пустых ячеек отличается от непустых, то существует реализуемый U_S' алгоритм, обеспечивающий нахождение операций G_L, G_R и формирующий в памяти U_S' адекватную (изоморфную) модель начального фрагмента массива ячеек хранения (т.е. начального фрагмента модели одномерной познаваемой среды).

Доказательство приведено в работе [13]. Предложенный в этом доказательстве алгоритм характеризуется использованием вначале как бы случайных («хаотических») перемещений U_S' между соседними ячейками хранения (ЯХ) и «акцентированием внимания» на различиях, неоднородностях в их содержимом (т.е. во «внешней по отношению к U_S' среде», в данном случае – в содержимом одномерного линейного массива ячеек хранения с конструкциями из $\mathbf{B} \cup \{\square\}$). Как известно, очень похожее, как бы «хаотическое» поведение демонстрируют на самых ранних этапах освоения окружающего мира новорождённые многих высокоорганизованных животных, включая человека.

Теперь можно сформулировать более сильное утверждение о познавательных способностях U_S' .

Утверждение 5.5. Если в синтезаторе-анализаторе U_S' с частично-неизвестной технологией $\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, P_A, \dots, P_j, G_L, G_R\} \rangle$ (или $\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, P_S, \dots, P_j, G_L, G_R\} \rangle$), где случайная стационарная операция P_A (или P_S) известна; к группе неизвестных динамических операций имеется независимый от остальных технологических операций доступ; линейный массив ячеек хранения содержит непустые ячейки с источниками конструкций $b_i \in \mathbf{B}$, отличающимися по $P_A (P_S)$, и хотя бы один конечный непрерывный массив пустых ячеек, причём значение по $P_A (P_S)$ для пустых ячеек отличается от непустых; среди технологических операций $\mathbf{T}$ отсутствуют операции типа F_k ; имеется полное по декомпозиции множество неизвестных стационарных операций и неизвестные операции синтеза типа F_t или F_S , то существует реализуемый U_S' алгоритм, обеспечивающий нахождение операций

G_L , G_R и переводящий T в полную технологию с вероятностью, сколько угодно близкой к единице.

Доказательство также приведено в работе [13].

Из теории алгоритмов известна функциональная эквивалентность машин Тьюринга с одномерной и многомерными лентами. Приведём без доказательства (оно базируется на тех же принципах, что и доказательства утверждений 5.4, 5.5) аналогичное утверждение ОФТ.

Утверждение 5.6. Если в синтезаторе-анализаторе типа U_S' с частично неизвестными технологиями типа $T = \langle \mathbf{B}, \{F_1, \dots, P_A, (P_S), \dots, P_j, G_L^x, G_R^x, G_L^y, G_R^y, (G_L^z, G_R^z)\} \rangle$ случайная стационарная операция P_A (P_S) известна; к группе неизвестных динамических операций перемещения по каждой из координат $X, Y, (Z)$ имеется независимый от остальных технологических операций доступ; массив ячеек хранения имеет размерность, соответствующую набору динамических операций (т. е. является двух- или трёхмерным) и содержит непустые ячейки с конструкциями $b_i \in \mathbf{B}$, отличающимися по P_A (P_S), и хотя бы один конечный непрерывный массив пустых ячеек, причем значение по P_A (P_S) для пустых ячеек отличается от непустых; среди технологических операций T отсутствуют операции типа F_k ; имеется полное по декомпозиции множество неизвестных операций и неизвестные операции синтеза типа F_t или F_s , то существует реализуемый U_S' алгоритм, обеспечивающий нахождение операций $G_L^x, G_R^x, G_L^y, G_R^y, (G_L^z, G_R^z)$ и переводящий T в полную технологию с вероятностью, сколько угодно близкой к единице. ■

Таким образом, в ОФТ возможно получение (конструктивным путём) ответа на вопрос о существовании алгоритмов «познавательной деятельности» и соответствующих автоматически действующих устройств в одно-, двух- и трёхмерных «технологических средах». Эти алгоритмы (автоматы) и связанные с ними особенности и эффекты коррелируют с известными из биологии и истории фактами развития живых организмов и человеческого общества, что может рассматриваться как дополнительное косвенное подтверждение досто-

верности полученных конструктивным путём результатов, а также как ещё один аргумент в пользу неоднократно выдвигавшегося в различных работах тезиса о глубокой связи теории алгоритмов не только с основаниями математики и теорией вычислений, но и с самыми различными природными, биологическими и социальными процессами, т.е. об отражении в основах теории алгоритмов (и, соответственно, в базирующейся на ней ОФТ) фундаментальных законов развития природы и общества [1, 21, 29, 60, 69].

Более того, дальнейший тщательный анализ результатов, сформулированных в утверждениях 5.1-5.6, вместе с последними данными об особенностях обработки информации в мозге человека позволяет наметить выход из очевидного тупика на пути создания искусственного интеллекта, действительно эквивалентного человеческому, а может быть, даже и превосходящего его.

5.2. ТЕОРИЯ ГЕТЕРОГЕННЫХ АВТОМАТОВ

В параграфах 1.1, 3.1, 5.1 достаточно много говорилось о том, что функциональность реальных объектов окружающего нас физического мира *не* сводится только к взаимодействию на уровне их гомогенных (т.е. однотипных, однородных) физических свойств. Хотя такая однородность и позволяет нам сравнительно легко разрабатывать соответствующие математические теории и способы расчёта сложных систем, состоящих из огромного числа гомогенных компонентов (электрических, электронных, механических, гидравлических, пневматических и т.п.), всё же очень многие природные (естественные) и искусственные (созданные нами) системы носят ярко выраженный гетерогенный характер. Особенно наглядно данный факт иллюстрируют разнообразие химических взаимодействий между молекулами веществ с различным атомным составом, структурой и – самое главное – с различными (гетерогенными) физико-химическими свойствами. Видимо, не зря природа предпочла для своих самых сложных конструкций – биологических систем – использовать именно гетерогенные

функциональности, реализуемые чрезвычайно миниатюрными объектами – атомами и молекулами. Так почему бы и нам не воспользоваться этой же возможностью и не научиться делать наши автоматы на подобных принципах? Ведь пока все цифровые электронные системы управления строятся на основе гомогенных микроэлектронных элементов типа изображённого на рис. 3.1. Куда меньше места заняли бы аналогичные молекулярные конструкции – даже в сравнении с уже существующими и перспективными микроэлектронными компонентами с размерами, близкими к десяткам и даже единицам нанометров. Плюс ещё дополнительные преимущества, связанные с самосинхронизацией и возможностью эффективного распараллеливания работы таких гетерогенных схем и устройств, столь широко используемых природой в различных биологических процессах.

Исследованию некоторых перспектив и возможностей в этой области и посвящён данный раздел. Основная его цель – расширить класс автоматов, которые естественно назвать гетерогенными, включив в него различные гетерогенные функциональности, в том числе и функциональности «химического типа», что позволило бы рассматривать различные химические реакции в более простом логическом контексте, чем это предполагает, например, λ -исчисление Чёрча [61] или теория логической интерпретации уровней динамической концентраций реагентов в некоторых системах непрерывных химических реакций Г. Хакена³².

Напомним, что в последнем случае в качестве модели простейшего двухстабильного элемента памяти (называемого в вычислительной технике «триггером»), который мог бы сыграть ключевую роль в формировании различных алгоритмов синтеза предбиологических и биологических компонентов, предложена химическая система, в которой роль двух устойчивых состояний играют две различные концентрации реагентов. Помимо громоздкости самой модели, которая

³² *Хакен Г.* На пути к динамической теории информации. В кн.: Термодинамика и регуляция биологических процессов. Теория информации, управление в живых системах, проблема самоорганизации, эволюция и онтогенез. – М.: Наука, 1984. – С. 57-63.

должна ещё каким-то образом выделиться («компартиментироваться») в достаточно однородном пространстве первичного биохимического бульона, её поведение ещё и сильно зависит от множества различных сложноуправляемых факторов, влияющих на концентрацию (и, соответственно, состояние) реагентов. Вероятно, именно по этим причинам данная модель не нашла в среде биохимиков достаточного числа приверженцев.

Согласно другому подходу, рассмотренному в [61], основная операция λ -исчисления Черча – редукция – записывается так:

$$(\lambda x.A)B \rightarrow A[x:=B], \quad (5.3)$$

где правая часть выражения означает преобразование строки A путём замены вхождения подстроки x на строку B и трактуется как взаимодействие молекулы A , имеющей активный участок x , с молекулой B .

При детальном рассмотрении можно сделать следующие серьезные замечания относительно представления химических реакций с помощью редукций типа (5.3):

1) λ -исчисление Черча сохраняет линейную (одномерную) структуру объектов, тогда как структура реальных химических молекул трёхмерна и достаточно редко бывает близка к линейной;

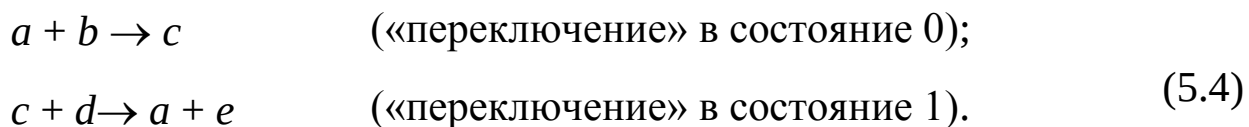
2) λ -исчисление требует так называемой «операции абстрагирования», т.е. выделения некоторой подстроки из линейной конструкции (строки). Аналогию такой операции в реальных системах химических взаимодействий подобрать сложно;

3) в операции редукции происходит обязательная замена исходной «абстрагированной» (выделенной) подстроки в линейной конструкции на другую линейную конструкцию, к которой применяется данная. При этом «абстрагированная» (выделенная) подстрока как бы исчезает, чего в реальных молекулах не происходит никогда;

4) формула (5.3) никак не учитывает возможные физические свойства взаимодействующих объектов A и B и их изменения. Такого представления реальных объектов с точки зрения ОФТ недостаточно, в том числе и для полноценного моделирования их поведения [13 – 16].

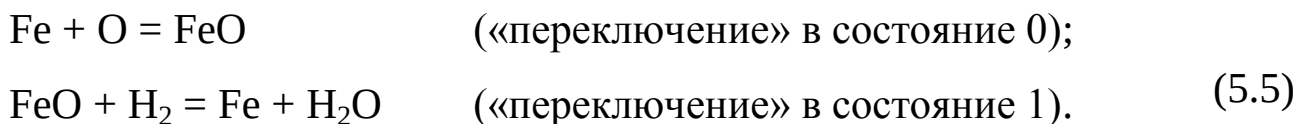
Таким образом, λ -исчисление Черча как модель химических взаимодействий (функциональностей) представляется достаточно неадекватной в сравнении с подходом, рассматриваемым ниже.

Напомним ещё раз две неравновесные реакции «химического триггера» из параграфа 3.2 (уравнение 3.14):



По закону сохранения вещества молекула «e» должна содержать компоненты «b» и «d» («e» = «b*d», где «*» — символ соединения компонентов «b» и «d»). Вещество «e» можно рассматривать, как «выходной» сигнал «химического триггера», свидетельствующий о его переключении из состояния 0 («c») в состояние 1 («a»), соответственно, молекула «b» — это «входной» сигнал, переключающий «триггер» из состояния 1 («a») в 0 («c»), а «d» — «входной» сигнал, переключающий триггер из 0 («c») в 1 («a»). Напомним, что в электронике такой триггер принято называть RS-триггером.

Как говорят сами химики, реакций типа (5.4) в химии «очень много». К ним могут быть отнесены, например, реакции окисления двухвалентного железа кислородом и его восстановления водородом:



Согласно определению конечного автомата³³, им называется система $S = \langle A, Q, V, \delta, \lambda \rangle$, где A, Q, V — конечные множества (алфавиты) соответственно входных сигналов, сигналов состояния автомата и выходных сигналов, а $\delta: Q \times A \rightarrow Q$ и $\lambda: Q \times A \rightarrow V$ — функции, определённые на этих множествах. Потребуем, чтобы в качестве элементов множеств A, Q, V использовались гетерогенные объекты, наделённые гетерогенными свойствами и некоторыми специфическими функциональностями, связанными с преобразованием параметров этих

³³ Адельсон-Вельский Г.М., Кузнецов О.П. Дискретная математика для инженера. 2-е изд-е. — М.: Энергоатомиздат, 1988. — 480 с.

свойств согласно определению функциональностей гетерогенных объектов [13]. Введём следующее определение:

Определение гетерогенного автомата (ГА). Гетерогенным (неоднородным) автоматом называется система $S = \langle A, Q, V, \delta, \lambda \rangle$, где A, Q, V – конечные множества соответственно входных гетерогенных объектов; гетерогенных объектов, конкретные совокупности которых представляют (кодируют) различные состояния автомата, и выходных гетерогенных объектов соответственно, а $\delta: Q \times A \rightarrow Q$ и $\lambda: Q \times A \rightarrow V$ – совокупности функциональных преобразований (параметров) свойств гетерогенных объектов для переключения состояний автомата и состояний его выходных объектов, определённых на указанных множествах (в которых любой гетерогенный объект представлен конечной совокупностью своих конечных свойств и функциональностей).

Как и любой классический автомат, гетерогенный автомат можно задать с помощью набора таблиц функциональных преобразований (параметров) свойств гетерогенных объектов, соответствующих таблицам переходов и выходным сигналам, функции которых выполняют соответствующие объекты, или с помощью ориентированного мультиграфа. Следует, однако, обратить внимание на то, что состояние гетерогенного автомата определяется как *совокупность всех объектов*, относящихся к множеству его состояний, *со всеми присущими этим объектам конкретными значениями конкретных (гетерогенных) свойств*. То есть число возможных состояний любого конечного гетерогенного автомата всегда конечно, поскольку не может выйти за рамки декартова произведения конечного числа состояний, приписываемых конечному числу объектов, даже если среди свойств объектов есть так называемые эмерджентные свойства и функциональности [13].

Определение гетерогенного слабоструктурированного автомата. Гетерогенный автомат называется слабоструктурированным, если объекты, совокупность которых определяет его состояние, могут находиться в любом месте некоторого (условно) ограниченного пространства, через (условную) границу которого поступают входные объекты и удаляются выходные.

Примером слабоструктурированного автомата может быть некоторая область физического пространства внутри живой клетки, в которой происходит конечная совокупность химических реакций, связанных с её метаболизмом. При этом в данную область должны поступать исходные вещества и удаляться продукты реакций. Условная граница такого физического пространства может быть связана как с частично проницаемыми оболочками самой клетки, так и с другими структурами внутри неё.

Иным примером может быть специально выделенный объект, используемый в ООП, – так называемая «рамка» (frame), «окно» или «панель» обычно фиксированных размеров (связанных или с размерами экрана дисплея, или с размерами памяти компьютера), в которую можно в любом «месте» вставлять (помещать) другие объекты. При этом информационные связи между вставляемыми объектами могут создаваться различными способами, например, последовательно от объекта к объекту по мере вставки их в рамку. Такую достаточно нерегулярную связь между объектами удобно интерпретировать как некоторый «поток», «омывающий» поочередно все объекты и «доносящий» любой входной объект до любого объекта, вставленного в рамку.

В примере с областью клетки «поток», омывающим все объекты, относящиеся к автомату, может быть броуновское движение, переносящее «входные» объекты-молекулы к различным молекулам-переработчикам, образующим автомат.

Чтобы входные объекты точно попадали к местам своего назначения внутри автомата даже тогда, когда их конкретный маршрут непредсказуем, удобно снабдить их уникальными именами, а компоненты автомата (объекты-переработчики) научить распознавать эти имена. Естественно, разные объекты должны получать разные имена, а одинаковые – одинаковые. Присваивание разным объектам различных уникальных имён назовём «индивидуализацией».

Введение уникальных имён и соответствующих механизмов их распознавания позволяет снять проблему синхронизации. Действительно, пока объект-переработчик («оператор») не получит свой

входной объект-сигнал («операнд») с конкретным именем, никаких изменений ни в совокупности объектов, представляющих состояние автомата, ни в выходных объектах-сигналах произойти не может. Автоматы с такими свойствами называются самосинхронными³⁴.

Таким образом, можно ввести следующее определение:

Определение гетерогенных слабоструктурированных самосинхронных автоматов с индивидуальными именами объектов операций (ГССА с ИИОО). Гетерогенный слабоструктурированный автомат называется самосинхронным с индивидуальными именами объектов операций, если объекты, совокупность которых определяет его состояние, и объекты-сигналы (входные и выходные) имеют какие-либо метки-имена, позволяющие отличить их друг от друга и распознать друг друга при сближении или соединении.

Заметим, что в ООП присвоить индивидуальные имена входным объектам-сигналам («операндам»), объектам-переработчикам («операторам») и выходным объектам-сигналам («результатам»), а также научить «операторов» их распознаванию – не проблема. Собственно, в этом и заключается одна из функций программиста, работающего в среде ООП.

С химическими реакциями ситуация сложнее. Мы пока не очень хорошо понимаем, какие физико-химические свойства «заставляют» молекулы «различать» друг друга на расстоянии и взаимодействовать именно так, а не иначе. Разумеется, некие общие представления о такого рода механизмах есть, но как одна молекула может на приличном расстоянии «почувствовать» другую, нужную ей для предпочтительной реакции молекулу, в деталях объяснить пока не можем.

Другое важное достоинство всех самосинхронных автоматов – возможность параллельной работы их внутренних компонентов. Действительно, как только какой-либо из объектов-переработчиков освободился от своего объекта-«операнда», он может тут же приступить к следующей операции. Тем не менее для сохранения аналогии с клас-

³⁴ см., например, *Josephs M.B., Udding J.T.* An algebra for delay-insensitive circuits. Technical Report WUCS89-54. – St Louis, USA: Washington University, 1989.

сическими автоматами будем предполагать, что пока гетерогенный самосинхронный автомат не закончил работу с очередным входным объектом-сигналом, следующий входной объект на «вход» автомата не подаётся.

Вернёмся теперь к рис. 3.6, на котором приведены условная схема и граф переходов ГССА с ИИОО типа «RS-триггер». Работа этого устройства задана уравнениями (5.3). Из рис. 3.6 следует, что множество A входных объектов автомата есть $A = \{ \langle b, \Gamma_b \rangle, \langle d, \Gamma_d \rangle \}$, где Γ_b и Γ_d – списки свойств (представленных соответствующими значениями параметров) соответственно объектов « b » и « d ». Множество состояний автомата содержит тоже два объекта: $Q = \{ \langle a, \Gamma_a \rangle, \langle c, \Gamma_c \rangle \}$, а множество выходных объектов – один: $V = \{ \langle e, \Gamma_e \rangle \}$. Для удобства на графе переходов состояния представлены только объектами, которые в данном состоянии играют активную роль – взаимодействуют с входными и формируют выходные объекты.

Для задания гетерогенного автомата (ГА) графом переходов формулируются аналогичные классической теории автоматов условия автоматности или корректности³⁵: 1) для любого входного объекта a_j имеется ребро, выходящее из состояния q_i , на котором написано a_j (условие полноты); 2) любой входной объект a_j встречается только на одном ребре, выходящем из состояния q_i (условие непротиворечивости). Представленный на рис. 3.6 автомат задан неполностью и не отвечает условию 1 (для состояния c , например, отсутствует выходящее ребро с символом входного объекта b). Способы доопределения таких автоматов рассмотрены в [15].

Можно определить аналог автоматного отображения, зафиксировав в автомате начальное состояние q_i и поставив в соответствие каждой входной последовательности объектов $\alpha = a_{j1}a_{j2}...a_{jk}$ (т.е. «входному слову») выходную последовательность ω :

$$\omega = \lambda(q_i, a_{j1}) \lambda(q_i, a_{j1}, a_{j2}) ... \lambda(q_i, a_{j1}, ... a_{jk}).$$

³⁵ см., например, *Адельсон-Вельский Г.М., Кузнецов О.П. Дискретная математика для инженера. – М.: Энергоатомиздат, 1980. – С. 248.*

Достижимые состояния и сильно связный гетерогенный автомат определяются аналогично классическому варианту.

Определение автономного гетерогенного автомата. Гетерогенный автомат называется автономным, если множество его входных объектов состоит из одного объекта: $A=\{a\}$.

Дальше начинаются расхождения с классической теорией автоматов, некоторые из них рассмотрены в [15].

Ещё сложнее ситуация с изоморфизмом гетерогенных автоматов. Если мы примем, например, классическое определение изоморфизма автоматов применительно к гетерогенным, то изоморфизма на уровне свойств и функциональностей объектов мы можем и не получить. Покажем это на примере с автоматом «химический RS-триггер» (рис. 3.6). Действительно, все объекты, участвующие в его функционировании, могут иметь и разные списки свойств, и разные значения параметров, представляющих свойства в этих списках. Кроме того, в ходе взаимодействия («химической реакции») объектов между собой многие из этих параметров могут (и даже должны) изменяться.

Говорить об изоморфизме соответствующих процессов изменения различных параметров, представляющих различные свойства объектов, можно будет только тогда, когда мы тщательно специфицируем всё многообразие возможных функциональностей, связанных с этими свойствами, и параметров, их представляющих. Рассмотрим один из вариантов, который ещё раз проиллюстрирует эффективность одного из методов ОФТ, названного «Методом назначенных функциональностей». Метод позволяет построить некоторую модель изменения свойств, когда реальную картину взаимодействия объектов мы не знаем. Этот метод уже использовался нами в параграфе 4.2 при составлении табл. 4.1, когда нужно было найти минимальный набор свойств и функциональностей гипотетических молекул для синтеза линейных молекулярных конструкций (полимерных цепочек). Воспользуемся им ещё раз.

Пусть в схеме рис. 3.6 объект «a» обладает свойством γ_1 , и пусть значение этого свойства, равное 1, обеспечивает «достаточно удалён-

ное» взаимодействие его со свойством $\gamma_2=0$ объекта « b » таким образом, что объекты « a » и « b » как бы притягиваются друг к другу, причём в момент соединения (касания) объектов свойство γ_2 принимает значение 1 и объекты « a » и « b » фиксируются в таком состоянии (т.е. комбинация свойств $\gamma_1=1$ и $\gamma_2=1$ вызывает сцепление объектов). Будем полагать, что новое состояние синтезированного из « a » и « b » объекта « c » (« c »=« $a*b$ ») состоит из двух свойств: $\Gamma_c=<\gamma_1, \gamma_2>$, причём согласно нашему построению $\gamma_1=1, \gamma_2=1$.

Предположим дальше, что такая комбинация свойств вызывает взаимодействие со свойством $\gamma_3=0$ объекта « d », заключающееся в том, что объекты « c » и « d » также начинают как бы притягиваться друг к другу. При контакте объектов « c » и « d » свойство $\gamma_3=0$ объекта « d » получает значение («переключается в») 1, что способствует сцеплению объектов « b » и « d », которое, в свою очередь, приводит уже к отталкиванию объекта « a » от объекта « e »=« $b*d$ » (т.е. комбинация свойств $\gamma_1=1, \gamma_2=1$ и $\gamma_3=1$ вызывает отталкивание объектов « a » и « e »). В результате образуется новый объект со свойством $\Gamma_e=<\gamma_2, \gamma_3>$, а объект « a » возвращается в исходное состояние, готовое к встрече с новым объектом « b », как только тот поступит через условную границу автомата.

Запись соответствующих функциональных взаимодействий в форме, принятой в параграфе 4.2, даёт:

$$\begin{aligned} &(\gamma_1=1)^a+(\gamma_2=0)^b\rightarrow(\gamma_2=1)^c, \text{ (притяжение и сцепление объектов «}a\text{» и «}b\text{»)}, \\ &((\gamma_1=1)\&(\gamma_2=1))^c+(\gamma_3=0)^d\rightarrow(\gamma_3=1)^c, \text{ (притяжение объектов «}c\text{» и «}d\text{»,} \\ &\text{сцепление «}d\text{» с «}b\text{», отталкивание «}a\text{» и «}e\text{»)}, \end{aligned} \quad (5.6)$$

где верхние индексы в (5.6) означают принадлежность данного параметра к свойствам соответствующих объектов, а символы γ_i обозначают i -тые свойства самих объектов.

На рис. 5.2 приведён алгоритм (точнее, блок-схема алгоритма) переключения состояний объектов « a », « b », « c », « d » и их взаимодействия в форме, принятой в теории алгоритмов.

Можно также видоизменить в сравнении с (5.6) характер переключения свойств объектов, например:

$$\begin{aligned}
 (\gamma_1=0)^a + (\gamma_2=0)^b &\rightarrow (\gamma_1=1)^c, \text{ (притяжение и сцепление объектов «a» и «b»)}, \\
 (\gamma_1=1)^c + (\gamma_3=0)^d &\rightarrow (\gamma_3=1)^e, \text{ (притяжение объектов «c» и «d»)}, \\
 &\text{сцепление «d» и «b», отталкивание «a» и «e»}.
 \end{aligned}
 \tag{5.7}$$

Для восстановления исходного состояния объекта «a» $\gamma_1=0$, системе (5.7) необходимо дополнить строчкой $(\gamma_1=1)^a + (\gamma_3=1)^b \rightarrow (\gamma_1=0)^a$. Заметим также, что запись $\gamma_i=0$ не означает отсутствия свойства γ_i . Это может быть просто некоторое условное значение параметра, отображающего величину γ_i на числовую ось.

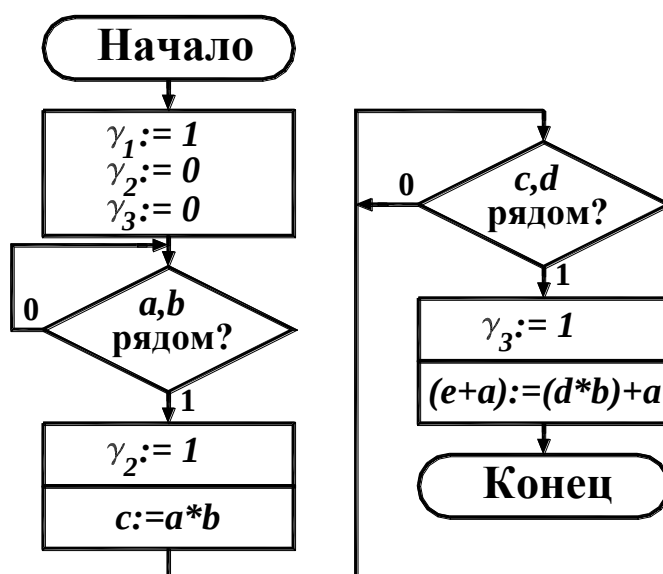


Рис. 5.2. Переключение свойств объектов «a», «b», «c» и «d» в процессе реализации алгоритма системы уравнений (5.6). Выход из ромбовидных блоков проверки условий, помеченный 0, означает невыполнение условия, помеченный 1 – выполнение

Из рис. 5.2 следует, что важную функцию в рассматриваемых вариантах алгоритмов переключения состояний свойств гипотетических объектов, обладающих поведением «химического типа», играет проверка их близости. Именно она синхронизирует последовательность операций «переключения» свойств, обеспечивая самосинхронность автоматов. Контроль сближения объектов можно выполнить с помощью проверки наличия какого-либо определённого значения

конкретного свойства одного из двух сближающихся объектов. Например, для записи (5.6) при сближении « a » и « b » достаточно дожидаться появления в ближайшем пространстве различного значения свойства ($\gamma_2=0$) или ($\gamma_1=1$). Проверку при этом можно сокращённо записать как ($\gamma_2=0?$) или ($\gamma_1=1?$). Естественно, что проверку ($\gamma_1=1?$) лучше выполнять в рамках объекта « b », поскольку положительный результат проверки позволит сразу перейти к переключению свойства γ_2 этого же объекта: $\gamma_2:=1$. (Два следующих друг за другом символа «двоеточие» и «знак равенства» будем в дальнейшем понимать как операцию присвоения соответствующего значения, стоящего справа от знака равенства, свойству, стоящему слева, т.е. перед двоеточием). Чтобы указать, что именно объект « b » должен производить проверку, модифицируем запись проверки следующим образом: $b(\gamma_i=x?)$, где x — ожидаемое значение свойства γ_i при сближении объектов « b » и « a ».

В работе [15] приведены другие способы сокращения формальной записи алгоритмов типа рис. 5.2, позволяющие свести представление такого алгоритма к одной короткой строке.

Нетрудно предложить и другие варианты переключения функциональных свойств, в которых участвуют параметры типа γ_1 , γ_2 , и γ_3 , обеспечивающие последовательность операций, соответствующих выражениями (5.6), (5.7).

Таким образом, если рассматривать процесс функционирования гетерогенного автомата с точки зрения «назначенных» функциональностей, обеспечивающих его физическую реализацию, то очевидной становится множественность решений, дающая внешне изоморфное, но, по сути, различное его поведение. То есть справедливо следующее:

Утверждение 5.7. Изоморфизм на уровне графов гетерогенных автоматов допускает их множественную *не*изоморфную реализацию на уровне функциональностей гетерогенных объектов, участвующих в функционировании автомата.

Это утверждение фактически поясняет и частично даже доказывает известный тезис Людвига фон Берталанфи, считающегося осно-

вателем общей теории систем, о том, что поведение систем может быть вполне изоморфным в самых различных своих проявлениях (т.е. в различных физических реализациях).

Кроме того, утв. 5.7 вкупе с введением индивидуальных имён объектов и, соответственно, свойства их индивидуальной различимости, позволяет вводить в условные границы гетерогенного слабо-структурированного самосинхронного автомата с индивидуальными именами объектов операций столько функций (в том числе, если нужно, – логических), сколько может вместить в себя его физическое внутреннее пространство. Для этого нам необходимо доказать, что функциональности типа (3.13), (3.14) из параграфа 3.2 и аналогичное (3.14) выражение (5.4) образуют, с точки зрения булевых функций, функционально полную систему. Сделать это можно по-разному. Но сначала сформулируем само утверждение:

Утверждение 5.8. Система функциональностей «химического типа», заданных уравнениями (3.13), (3.14) или (5.4), образует функционально полный базис для реализации любых булевых функций от конечного числа переменных и, соответственно, для реализации алгоритмов функционирования любых конечных самосинхронных слабо-структурированных гетерогенных автоматов с индивидуальными именами объектов операций.

Справедливость этого утверждения базируется на ряде лемм, формулировка и доказательства которых приведены в работе [15] и статье «Системный анализ гетерогенных слабо-структурированных автоматов. II»³⁶.

Основными моментами в этих леммах и их доказательствах являются, в частности, следующие: 1) понятие так называемого «разветвителя» (сигналов), введённого при описании функционально полных наборов гетерогенных функциональных блоков (ФБ)³⁷. Этот

³⁶ Крылов С.М., Гребенчиков Е.Н. Системный анализ гетерогенных слабо-структурированных автоматов. II. Переход от преобразований «химического типа» к логическим схемам и обратно // Вестник Самар. гос. техн. ун-та. Сер. технические науки. – 2012. – № 2 (34). – С. 65-71.

³⁷ Крылов С.М., Сараев М.В. Синтез конфигурируемых блоков для аналого-цифровых

ФБ (рис. 5.3 слева) реализует функции разветвления любых типов сигналов (в том числе сигналов-объектов) на любое конечное число направлений, если это невозможно сделать более привычным для нас образом, например с помощью проводов (как это обычно делается для сигналов напряжения) или трубопроводов (для гидравлических и пневматических «сигналов»). Справа на рис. 5.3 как раз и показан вариант разветвления сигнала x на 4 направления. Нетрудно построить похожие схемы и для разветвления сигналов на 3, 5, 6, ..., n направлений (где n – любое целое число) [6, 15].

Нетрудно доказать, что с помощью традиционных функционально полных наборов двух- и одноходовых логических элементов, к которым добавлена схема «разветвителя», может быть реализована любая логическая функция для любого числа входных гомогенных (однородных) логических переменных [6, 15], т.е. соответствующие наборы образуют функционально полную систему (гомогенных) булевских функций для любых типов сигналов.

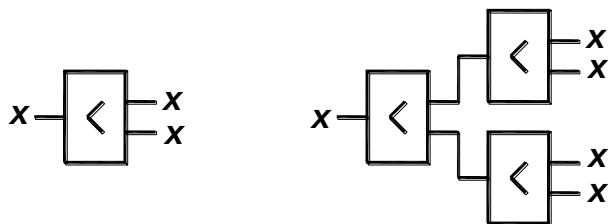


Рис. 5.3. Реализация с помощью «разветвителя» (слева) функции разветвления сигнала x на 4 направления (справа)

На основе этого результата в тех же работах [6, 15] показано, что аналогичная система логических ФБ функционально полна и для любых наборов *гетерогенных* сигналов, включая сигналы, представленные, например, уровнями тока, частотой, фазой, длительностью импульсов, другими электрическими параметрами (сопротивлением, ёмкостью, индуктивностью) и т.д.

Вторым важным моментом в доказательстве утв. 5.8 является доказательство возможности преобразования любой комбинационной логи-

ческой схемы (фактически – любой логической функции от конечного числа переменных) в эквивалентные цепочки преобразования «химического типа» в варианте (3.13). В этих цепочках функции различных сигналов в различных логических цепях выполняют различные объекты типа молекул. Причём в отличие от «разветвителей» с однотипными и одинаковыми входными и выходными сигналами, изображёнными на рис. 5.3, в таких цепочках функции «разветвителей» уже могут выполнять различные взаимодействия «химического типа», имеющие *различные* молекулоподобные объекты на «входе» и «выходе».

На рис. 5.4 приведена одна из возможных таблиц (схем) «перевода» элементарных логических функций (слева), образующих функционально полную систему (включая «разветвитель»), в подходящие схемы взаимодействий «химического типа» (справа).

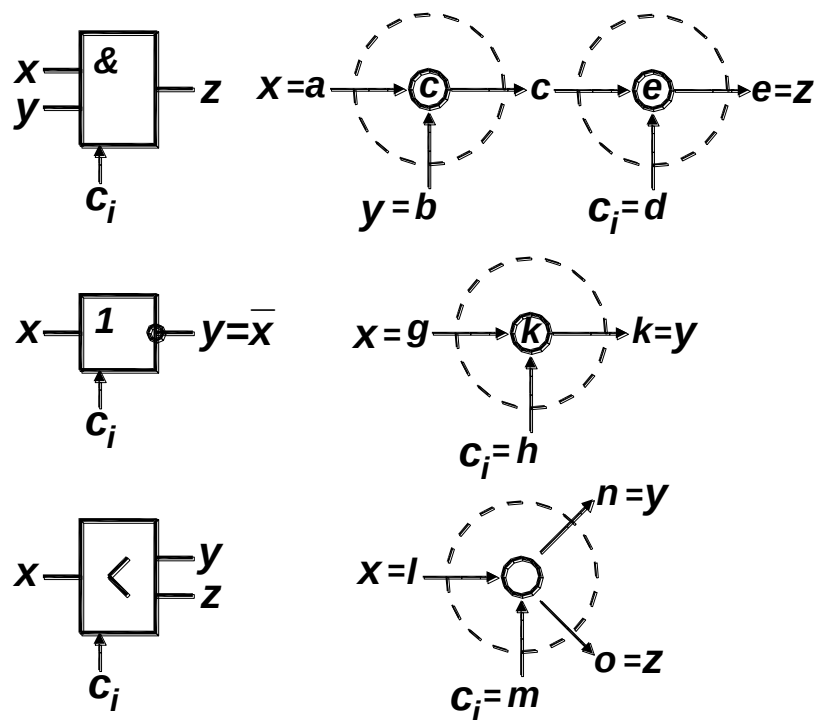


Рис. 5.4. Один из вариантов «перевода» стандартного функционально полного набора логических вентилях с синхронизацией (сигналы C_i) в эквивалентные преобразования «химического типа»

Сигналы синхронизации типа C_i у логических вентилях введены в [6, 15] для упрощения доказательства, а сами схемы преобразований

«химического типа» соответствуют либо химическим реакциям типа (3.13), либо – нижняя схема «разветвителя» – реакции типа (5.8):



где l, m, n, o – различные «объекты-молекулы».

Как и на рис. 3.6, пунктирной окружностью на рис. 5.4 выделена локальная область пространства, в которой происходит соответствующее преобразование.

После подтверждения справедливости утв. 5.8 легко доказывается следующее:

Утверждение 5.9. Функциональности «химического типа», эквивалентные схемам (3.13), (3.14) и (5.8), образуют функционально полный базис для реализации любых конечных булевых функций и элементов памяти конечного объёма.

Чтобы доказать утверждение 5.9, нам нужно лишь показать реализуемость в системе упомянутых выше функциональностей элементов памяти. Но это уже было сделано раньше – см. интерпретацию схемы функциональности (3.14) в третьей главе или функциональности (5.1) в этом параграфе, а также на рис. 3.6. Таким образом, утв. 5.9 доказано в полном объеме. ■

Утверждение 5.10. С помощью функциональностей утв. 5.9 может быть реализован любой конечный автомат Мили (Мура) в виде системы (сети) гетерогенных самосинхронных автоматов с индивидуальными именами объектов операций и внешней синхронизацией.

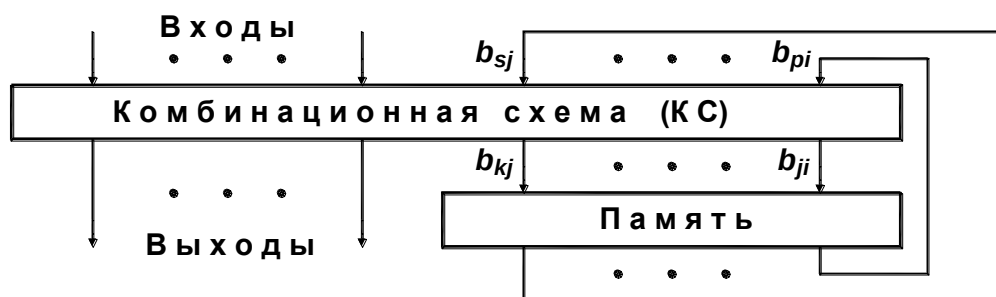


Рис. 5.5. Стандартная реализация автомата Мили. Символами типа b_{ij} обозначены молекулуподобные объекты с индивидуальными именами

Структурная схема соответствующего автомата Мили приведена на рис. 5.5.

Как известно, любой классический конечный автомат (в том числе Мили или Мура) может быть реализован в виде конечной комбинационной схемы (КС) и запоминающего устройства конечной ёмкости, в котором запоминается текущее состояние автомата. Реализуемость этих компонентов в виде слабоструктурированных самосинхронных гетерогенных автоматов уже была доказана в утв. 5.9. ■

Утверждение 5.11 (обратное к 5.10). Любая конечная схема функциональностей «химического типа», состоящая из выражений типа (3.13), (5.8) и/или (3.14), может быть преобразована в эквивалентную схему классического конечного автомата (в том числе в любом функционально полном базисе булевых функций).

Простейший вариант доказательства истинности этого утверждения – приписать каждому объекту-сигналу уникальное двоичное кодовое имя и заменить все элементарные химические преобразования на схемы сравнения соответствующих сигналов-имён с заданными и генерирования новых сигналов-имён в соответствии с эмулируемыми элементарными химическими функциональностями. ■

Таким образом, мы показали, что при определённых условиях сети (схемы) преобразований «химического типа» (как представители слабоструктурированных самосинхронных гетерогенных автоматов с индивидуальными именами объектов операций) могут быть заменены подходящей схемой обычных автоматов Мили или Мура, и наоборот, для конечных схем автоматов Мили или Мура можно подобрать соответствующие сети преобразований «химического типа», выдающих определённые типы сигналов-объектов в ответ на определённые внешние сигналы-объекты и текущее состояние автомата. Но вот существуют ли реальные наборы реальных химических реакций для нужных нам самых различных схем преобразований, пока вопрос открытый. Ясно лишь, что на основе рассмотренных здесь систем преобразований «химического типа» могут быть построены самые различные аналоги существующих вычислительных систем и их моделей, включая модели самовоспроизводящихся и саморазвивающихся автоматов.

5.3. ГЕТЕРОГЕННЫЕ СИСТЕМЫ В ЭЛЕКТРОНИКЕ

Гетерогенные автоматы, а в более широком аспекте – вообще любые гетерогенные устройства обработки и преобразования информации, могут быть реализованы не только на основе молекулярных конструкций. Как уже отмечалось, широкое поле для них открыто и в области электроники, поскольку до сих пор разработка и производство электронных функциональных блоков (ФБ) для компьютеров и различных систем управления шли по традиционному, не самому эффективному, но зато самому простому для понимания и разработки формальных методов их расчёта пути, а именно по пути использования в первую очередь гомогенных сигналов и компонентов, например. логических вентилях, работающих только с двумя уровнями напряжений (соответствующих «0» и «1») в цифровой электронике, или аналоговых дифференциальных операционных усилителей, опять же обрабатывающих преимущественно непрерывные уровни напряжений – в аналоговой. Лишь в последнее время достаточный интерес стал проявляться к токовым схемам, которые в современных технологиях производства интегральных схем реализуются весьма эффективно и просто.

Между тем носителями информации могут быть не только уровни напряжения и тока, но и другие типы сигналов, например частота, фаза сигнала, их длительность и т.д. и т.п. Более того, не только электрические сигналы могут передавать дискретную и (или) «непрерывную» («аналоговую») информацию. Это могут быть и какие-то электрические параметры элементов – сопротивление, ёмкость, индуктивность. Кстати, многие датчики основаны именно на использовании таких параметров в качестве источников нужной при измерении информации. Это могут быть также оптические, акустические, магнитные и прочие типы сигналов.

На рис. 5.6 приведены возможные сочетания различных типов электрических сигналов и параметров для гомогенных и гетерогенных электронных ФБ, распределённые по парам «входной сигнал» (Вх.) – «выходной сигнал» (Вых.).

Вх. Вых.	U _ц	I _ц	U _а	I _а	U _д	I _д	U _ф	I _ф	U _в	I _в	U _ч	I _ч	R	C	L
U _ц															
I _ц															
U _а															
I _а															
U _д															
I _д															
U _ф															
I _ф															
U _в															
I _в															
U _ч															
I _ч															
R															
C															
L															

Рис. 5.6. Примерная распространённость гомогенных и гетерогенных ФБ для различных типов сигналов и параметров по сочетаниям входной сигнал (Вх.) – выходной сигнал (Вых.)

Латинскими заглавными буквами на рис. 5.6 обозначены общепринятые сигналы или параметры: U – напряжение, I – ток, R – сопротивление, C – ёмкость, L – индуктивность, индексами (в качестве которых используются русские буквы) – тип сигналов: ц – цифровые, а – аналоговые, д – длительность импульса, ф – фаза, в – величина импульса, ч – частота сигнала. Интенсивность штриховки примерно соответствует интенсивности использования данных сочетаний входных и выходных сигналов как в гомогенных вариантах ФБ (лежащих на главной диагонали слева-направо сверху-вниз), так и в гетерогенных (все остальные квадратики, лежащие вне главной диагонали).

Легко заметить, что наиболее интенсивно используются именно гомогенные ФБ – как цифровые, так и аналоговые. Все они лежат на главной диагонали рис. 5.6. Большинство остальных (гетерогенных) со-

четаний пока не используется, поэтому соответствующие квадратики не заштрихованы вообще. Но никто не запрещает их использовать!

Более того, не исключено, что какие-то варианты таких гетерогенных ФБ могут оказаться весьма полезными, особенно в случаях, когда электронная система должна работать именно с различными (гетерогенными) наборами сигналов, поскольку управляемый (или контролируемый) объект зачастую требует именно их! Особенно часто подобные требования могут предъявляться к многоцелевым аналого-цифровым системам, от которых в первую очередь и ожидается максимальная приспособляемость к различным типам и наборам входных и выходных сигналов. В параграфе 2.3 такие системы уже рассматривались. Они называются «Многофункциональные программируемые аналого-цифровые системы на кристалле» (МПАЦ СНК). Правда, там они анализировались с точки зрения многофункциональности в «гомогенном», так сказать, варианте (только для сигналов напряжения), а не с точки зрения «гетерогенности» набора самих типов сигналов, требуемых для того или иного объекта управления или контроля. Ясно, что подобные МПАЦ СНК полезно рассматривать в том числе и с точки зрения «гетерогенности» обрабатываемых и генерируемых ими типов сигналов. Причём чем шире будет соответствующий диапазон, тем, естественно, большую потенциальную область применения найдёт соответствующая многофункциональная электронная система.

Напомним, что в параграфах 1.1 и 3.2, а также в работе [6] показано, что практически любые гомогенные и гетерогенные физические объекты, включая различные электронные функциональные блоки (ФБ), могут быть формально представлены в виде записей следующего вида:

$$O_i = \langle \Gamma_i, \mathbf{M}_i \rangle = \langle \{\gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in}\}, \{\gamma_{ij} = \varphi_{ij}(\gamma_{sk}, \gamma_{il}, \dots, \gamma_{sm}); \dots\} \rangle; \quad (5.9)$$

где O_i – i -тый объект (включая электронные ФБ); γ_i – список интересующих нас его входных и выходных параметров γ_{ik} (в частности электрических); $\mathbf{M}_i$ – список интересующих нас функциональностей

φ_{ij} объекта (блока) O_i , которые могут быть записаны в любой удобной форме – в виде функций, физических законов, алгоритмических процедур и т.д.; n – число параметров, используемых в данном представлении блока (объекта) O_i ; $j, l \in \{1, \dots, n\}$, k, m – номера (вторые индексы) тех «свойств» и функциональностей, которые актуальны для анализа функциональных взаимодействий φ_{ij} данного блока (объекта) O_i (в текущем представлении) с другими блоками (объектами) типа O_s , которым блок (объект) O_i «передаёт» свои сигналы или параметры. Подчеркнём, что функциональных зависимостей φ_{ij} в списке функциональностей $\mathbf{M}_i$ объекта O_i , как и его соответствующих «сигнальных» выходов, в общем случае может быть столько, сколько нужно. В традиционном представлении электронного ФБ выход предполагается один, поэтому здесь, если не оговорено иное, в множестве $\mathbf{M}_i$ предполагается одна функциональная зависимость, определяющая выходной сигнал (или параметр) в виде некоторой функции от входных сигналов или параметров.

Различие в описании физических объектов типа электронных ФБ (5.9) и абстрактных математических объектов типа (1.4) тем не менее (как следует из параграфа 1.1) позволяет использовать как однотипные математические конструкции для формального представления различных формальных технологий проектирования вычислительных систем в ОФТ, так и саму технологию вычислений (т.е. саму технологию обработки информации). С учётом этого в ОФТ тоже возможны два уровня соответствующих формальных технологий для электронных гетерогенных систем:

1. Уровень *технологии разработки (проектирования)* $\mathbf{T}_{\text{РГ}}$ самих гетерогенных систем: $\mathbf{T}_{\text{РГ}} = \langle \mathbf{B}_{\text{РГ}}, \mathbf{F}_T, \mathbf{F}_A \rangle$; где $\mathbf{B}_{\text{РГ}}$ – набор необходимых гетерогенных ФБ, отвечающий (или не отвечающий – в зависимости от задач разработки) критерию функциональной полноты; $\mathbf{F}_T$ – набор операций синтеза гетерогенных систем из гетерогенных блоков, $\mathbf{F}_A$ – набор операций анализа фрагментов гетерогенных систем.

2. Уровень самой технологии *обработки* гетерогенных сигналов и параметров $T_{ГСП}$: $T_{ГСП} = \langle V_{ГСП}, F_T, F_A \rangle$; где $V_{ГСП}$ – набор необходимых сигналов и параметров гетерогенного характера; F_T – набор операций обработки этих сигналов и параметров, F_A – набор операций определения параметров этих сигналов, т.е. набор операций анализа (желательно, чтобы последние два набора представляли функционально полный базис для реализации всех функций технологии типа $T_{ГСП}$, иначе технология не будет обладать всеми потенциальными возможностями).

Наиболее существенные на сегодняшний день результаты с использованием концепций ОФТ достигнуты именно в области синтеза (проектирования) нового класса электронных схем, названных «гетерогенными» [6].

Согласно этому подходу и записям типа (5.9), если множество M_i описывает некоторую единственную интересующую нас функциональность (функцию) с одной выходной переменной и несколькими аргументами (т.е. «входными» переменными), представляющими один и тот же тип сигналов (параметров), то соответствующие электронные ФБ называются гомогенными, в противном случае – гетерогенными. Пример схемы гомогенного ФБ – логического элемента (вентиля) типа «2И-НЕ» – приводился раньше на рис. 3.1. Пример вентиля «2ИЛИ-НЕ» с входными и выходными сигналами напряжения ТТЛ-уровней, обозначенного E_2 , приведён ниже на рис. 5.7.

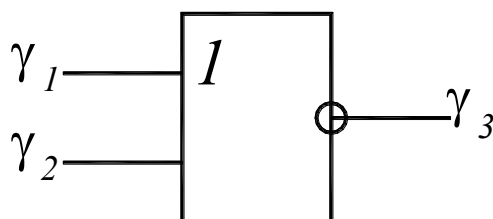


Рис. 5.7. Логический элемент «2ИЛИ-НЕ»

Совокупность входных и выходных сигналов и логической функции, т.е., в соответствии с выражением (5.9), функциональности этого элемента, можно записать следующим образом:

$$E_2 = \langle \Gamma_2, \mathbf{M}_2 \rangle = \langle \{\gamma_1, \gamma_2, \gamma_3\}, \{\gamma_3 = \neg(\gamma_1 \vee \gamma_2)\}, \quad (5.10)$$

где символ « $\neg$ » – символ инверсии, а « $\vee$ » – символ операции логического сложения (логическое «или»).

Точно так же функция логического элемента «2ИЛИ-НЕ» записывается и в случае, если сигналы $\gamma_1, \gamma_2, \gamma_3$ относятся к сигналам *различного* типа или даже будут какими-либо *различными* электрическими параметрами. То есть сама функция ФБ при замене гомогенных сигналов (параметров) на гетерогенные не изменится!

В [6] приведены данные, показывающие, что для n типов электрических сигналов или параметров число их возможных гетерогенных парных сочетаний «вход-выход» составляет $(n^2 - n) = n(n - 1)$, а число гомогенных – всего n , т.е. в $(n - 1)$ раз меньше! Ясно, что пространство возможных схмотехнических решений при реализации электронных систем на гетерогенных элементах с различными сочетаниями типов сигналов (параметров) по парам «вход-выход» также будет значительно больше, чем для гомогенных схем. Эта разница ещё более возрастет, если парные сочетания гетерогенных сигналов (параметров) по входу-выходу ФБ заменить на полностью гетерогенные сигналы (параметры) по всем входам и выходам. В случае двухвходовых элементов соответствующую картинку иллюстрирует рис. 5.8.

Из рис. 5.8 видно, что для двухвходовых ФБ с одним выходом общее число гетерогенных сочетаний сигналов (параметров) по каждому из двух входов и выходу составит уже $(n^3 - n)$, если оба входа двухвходового ФБ могут иметь в том числе и одинаковые типы входных сигналов (параметров). Если же типы обоих входных сигналов (параметров) должны быть обязательно разными, то общее число различных гетерогенных сочетаний по обоим входам и выходу уже запишется как $(n^3 - n^2) = n^2(n - 1)$.

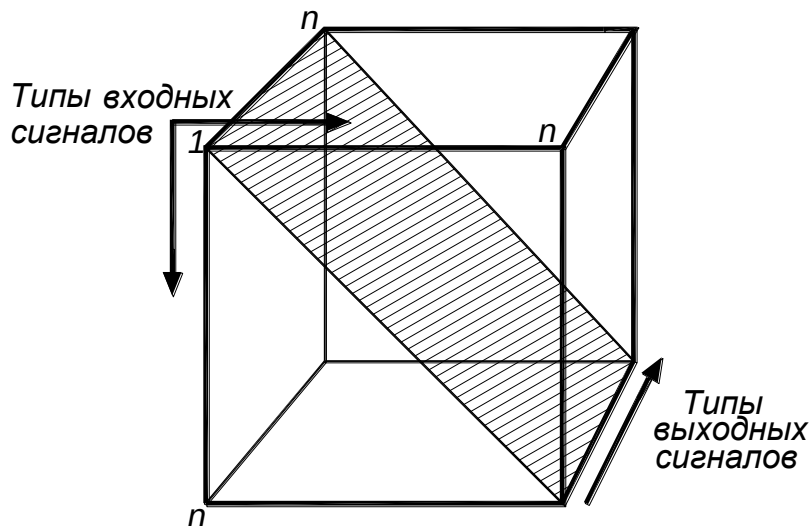


Рис. 5.8. Куб возможных гетерогенных сочетаний для n типов сигналов и параметров в ФБ с двумя входами и одним выходом

Общее число ФБ, соответствующих заштрихованной плоскости, равно, естественно, n^2 , а общее число ФБ, попадающих в куб на рис. 5.8, – n^3 . Поскольку на диагонали куба в этой же заштрихованной плоскости (идущей из точки с координатами $\langle 1, 1, 1 \rangle$ к точке с координатами $\langle n, n, n \rangle$) лежат и все гомогенные сочетания сигналов и параметров на обоих входах и выходе соответствующих ФБ, то разница между общим числом ФБ, принадлежащих кубу на рис. 5.8, и числом ФБ, соответствующих заштрихованной плоскости на том же рисунке, даст величину полностью гетерогенных сочетаний по всем двум входам и одному выходу для всех возможных гетерогенных ФБ, равную $(n^3 - n^2)$.

При числе различных типов сигналов и параметров, равном 15 (как и на рис. 5.6), число вариантов таких полностью гетерогенных ФБ будет равно 3150. Это существенно больше, чем число возможных гетерогенных ФБ, отличающихся только гетерогенными сочетаниями сигналов (параметров) по парам «вход-выход», равное 210 [6]. Таким образом, поле для потенциальных (и, возможно, более эффективных) схмотехнических реализаций соответствующих гетерогенных ФБ действительно будет существенно больше аналогичных возможностей для систем, отличающихся только гетерогенными сочетаниями сигналов в паре «вход-выход», и ещё больше, чем для гомогенных систем.

В [6] показано, что для реализации многих функционально полных систем как булевых, так и алгебраических (т.е. аналоговых) функционально полных по Шеннону функций³⁸ достаточно от трёх до пяти различных типов ФБ с одним-двумя входами и таким же количеством выходов. На рис. 5.9 приведён конкретный набор соответствующих гомогенных ФБ, гарантирующий практическую реализацию таких функционально полных систем как для булевых, так и для алгебраических (аналоговых) функционально полных по Шеннону функций.

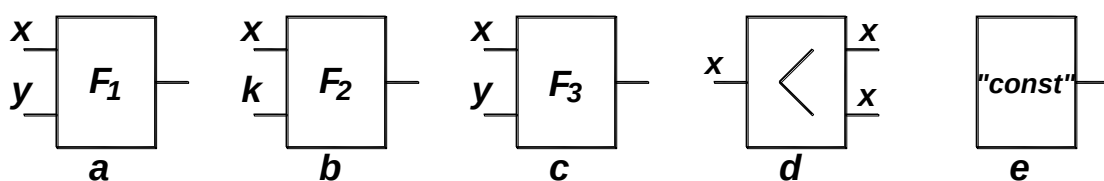


Рис. 5.9. Типовой набор гомогенных ФБ, достаточный для реализации функционально полных булевых или аналоговых (по Шеннону, т.е. алгебраических) систем функций

Двухвходовые ФБ типа $F_1, \dots, F_3$ на рис. 5.9 – это три однотипных блока, реализующих до трёх (разных) функций соответствующей функционально полной системы. Например, для булевой алгебры один из функционально полных наборов включает всего одну двухвходовую логическую функцию, например, $F_1 = \neg(x \& y)$ (функция «2И-НЕ»). То есть для реализации функционально полного набора в данном базисе нужен всего один (из трёх) двухвходовых ФБ.

Для функционально полного (по Шеннону) набора аналоговых (алгебраических) функций такой набор содержит уже все три типа двухвходовых ФБ: F_1 – двухвходовой аналоговый интегратор, выполняющий интегрирование входного сигнала x по времени, равному времени решения задачи с начальными условиями, равными y ; $F_2 = kx$ (двухвходовой аналоговый ФБ масштабирования с коэффициентом масштабирования $k \in (0, m/n]$, $m, n \in \mathbb{N}$; и $F_3 = -(x+y)$ – двухвходовой аналоговый сумматор-инвертор [6].

³⁸ см., например, Шеннон К. Математическая теория дифференциального анализатора. В кн.: Шеннон К. Работы по теории информации и кибернетике: пер. с англ. – М.: Иностранная литература, 1963. – С. 709-728.

ФБ, обозначенный на рис. 5.9 как «const», – источник некоторого фиксированного сигнала или параметра. Для функционально полных систем булевых функций это может быть просто источник уровней «логический 0» или «логическая 1», для аналоговых – какой-либо опорный сигнал, отличный от нуля.

Для приведённых выше примеров число различных ФБ, необходимых для реализации функционально полного набора булевых функций, равно 3, для аналоговых (по Шеннону) – 5, что представляется достаточно небольшой величиной. А поскольку, как уже отмечалось раньше, характер вычисляемой функции не зависит от характера (типа) носителя информационного сигнала, то таким же будет и набор *типов* функционально полных ФБ для различных гетерогенных вариантов. Изменится лишь их общее число, поскольку для разных сочетаний типов сигналов нужны (возможны) и различные ФБ, выполняющие, правда, однотипные с точки зрения математики операции. Исключение составляют лишь ФБ, выполняющие функции преобразования сигналов одного типа в другой. Поскольку в общем случае эти функции шире функций аналого-цифровых и цифро-аналоговых преобразователей, то в [6] соответствующие ФБ названы «конверторами».

На основе рассмотренных выше подходов оказалось возможным разработать формальные правила как для проектирования различных *гетерогенных* (в том числе аналого-цифровых) *многоцелевых программируемых* систем типа «система на кристалле» (ГМП СНК), так и для модернизации фрагментов существующих (преимущественно гомогенных) электронных схем. В частности, последнее направление было опробовано при разработке новых гетерогенных вариантов схем коррекции напряжения смещения нуля непрерывных дифференциальных операционных усилителей и компараторов. Разработанные по этим правилам схемы отличаются от классических вариантов коррекции значительно меньшей площадью, занимаемой ими на кристалле [6], что для современной микроэлектроники, стремящейся к увеличе-

нию числа различных функций, приходящихся на единицу площади, очень важно.

На рис. 5.10 показана структурная схема гетерогенной системы коррекции напряжения смещения нуля непрерывного повторителя, полученная на основе анализа и модернизации классической схемы коррекции смещения нуля повторителя, реализованной на обычном цифро-аналоговом преобразователе «код – напряжение».

Модернизация классической схемы коррекции выполнялась по формальным правилам анализа и синтеза гетерогенных систем, разработанным в соответствии с концепциями ОФТ в рамках технологии Т_{РГ}.

Новая схема коррекции смещения нуля непрерывного повторителя занимает на кристалле место, в несколько раз меньшее классического варианта.

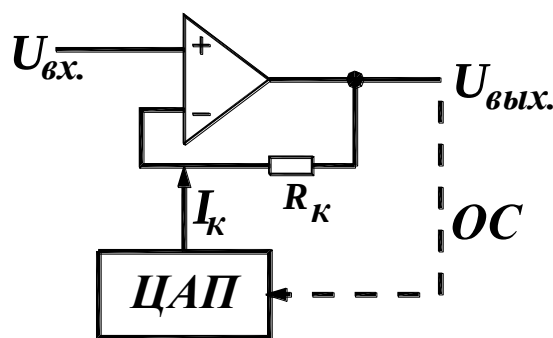


Рис. 5.10. Гетерогенная схема коррекции смещения нуля повторителя.

Для упрощения сложные цепи обратной связи (ОС), используемые в процессе коррекции, показаны пунктиром

Небезынтересно отметить, что схема коррекции на рис. 5.10, полученная в ходе формального анализа и синтеза альтернативных гетерогенных решений, использует практически тот же принцип формирования сигнала коррекции, что и схемы ряда отечественных и зарубежных патентов и изобретений для аналогичных целей, но не совпадает с ними по конкретным техническим решениям! То есть и сам аппарат ОФТ, и методы формального описания и представления в нём различных гетерогенных ФБ позволяют в принципе выйти на автоматическое генерирование неизвестных ранее схемотехнических решений, т.е. на автоматическое изобретение новых гетерогенных

схем, реализующих полезные и нужные функции более экономично и более эффективно, чем это делают существующие. Подробнее о таких «изобретательских» перспективах ОФТ можно прочитать в других работах³⁹.

ЗАКЛЮЧЕНИЕ

Метаматематические структуры, основанные на расширенной концепции алгебраической системы как системы алгоритмов (процессов) над различными формальными моделями реальных и абстрактных объектов, заключают в себе очень мощный потенциал, выводящий многие классические математические идеи и конструкции на принципиально новый уровень. Этот уровень позволяет в расширенном контексте взглянуть и на основания самой математики, и на логико-философские предпосылки возникновения её базовых идей, среди которых идея основных гомогенных абстрактных математических объектов – чисел, цепочек символов или кодов – приобретает новую, более осмысленную и более трезвую оценку. Разумеется, это несколько не снижает потенциала математики как действительно универсальной и весьма удобной науки в смысле средств, требуемых для её использования и развития, а также надёжности и точности получаемых результатов, но позволяет более чётко определить и реальное место самой математики в системе наших знаний об окружающем мире.

Этот новый метаматематический взгляд дает возможность существенно расширить границы применения самого математического аппарата, раздвинув их до новых, неожиданных горизонтов, включающих точную философию, теорию познания, точную метафизику, теорию социальных систем, теорию универсальных технических и тех-

³⁹ см., например, Крылов С.М. Автоматизация изобретения новых технических функциональных систем с использованием формально-технологического подхода // Компьютерные технологии в науке, практике и образовании: Труды седьмой Всерос. межвуз. науч.-практ. конференции. – Самара: СамГТУ, 2008. – С. 162-165. Или параграф 7.4 в работе [15].

нологических систем, теорию изобретений и многие другие нетрадиционные для классической математики области. Он же позволяет основательно продвинуться и в тех областях логики и соответствующего ей уровню понимания устройства нашего мира, которые более двух тысяч лет пребывали в своеобразном летаргическом сне, несмотря на колоссальные усилия многих выдающихся учёных. Только появление компьютеров дало, наконец, этим попыткам вполне подходящий рабочий инструмент. Так что не зря один из выдающихся математиков современности Грегори Чайтин написал на этот счёт уже цитировавшиеся выше примечательные слова: «Компьютер был изобретён для того, чтобы прояснить вопросы относительно оснований математики» [52]. Правда, в итоге оказалось, что он был изобретён в том числе ещё и для того, чтобы стать одним из важнейших инструментов этой новой математики – той, которую принято называть с приставкой «мета-».

Подчеркнём, что главная цель данной книги состоит не в том, чтобы научить читателя основаниям этой новой метаматематики или её базовым определениям – за исключением, может быть, важных концепций, принципов и понятий ОФТ, которые к сегодняшнему дню проработаны достаточно полно. Подобного рода комплексные многогранные задачи, связанные с разработкой фундаментальных основ новой метанауки, – несомненно, предмет её будущего становления и развития. Основная же цель книги, как уже не раз говорилось, – показать эффективность применения концепции алгоритмов не только к числам и кодам, но и к объектам совершенно иного характера, которые могут рассматриваться как модели объектов нашего реального физического мира со всеми их свойствами и вытекающим из этих свойств последствиями. То, что в итоге эти концепции оказались вполне работоспособными, выводящими нас на новые горизонты понимания логики устройства этого самого мира – вещь вполне ожидаемая и даже предсказуемая, ведь идеям научной метафизики – более двух тысяч лет.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ, ПОЛУЧЕННЫЕ В РАМКАХ ВЫПОЛНЕННЫХ ИССЛЕДОВАНИЙ ПО ОФТ

1. Общая формальная технология (ОФТ) максимально расширила перечень объектов и операций, над которыми могут выполняться алгоритмы. Таковыми объектами в ОФТ могут быть любые абстрактные (воображаемые) и (или) реальные объекты и (или) их упрощённые модели. А операциями над ними, помимо прочих, могут быть формальные технологические и аналитические операции типа сборки-разборки объектов, их синтеза или разрушения, измерения их различных параметров: массы, веса, длины, ширины, высоты, диаметра, цвета, температуры и т.д. Таким образом, фактически впервые полностью доказана общенаучная гипотеза о всеобщей применимости теории алгоритмов к изучению фундаментальных свойств процессов синтеза и анализа различных объектов и разработан соответствующий формальный аппарат.
2. Обнаружено и доказано существование различных классов технологий: некреативных, креативных, бесконечно креативных, полных, эволюционных. Вскрыта их взаимосвязь.
3. Доказана эквивалентность некоторых типов формальных и реальных технологий, включая «технологию» вычислительной математики.
4. Обнаружена эквивалентность определённых наборов формальных технологических и аналитических операций, т.е. доказана эквивалентность фундаментальных свойств некоторых типов технологий, подтверждающая со строгих теоретических позиций тезис Л. Берталанфи об общности фундаментальных свойств различных (технологических) систем.
5. Доказано, что кроме полных технологий с аналитической операцией (предикатом) определения равенства двух объектов (типа системы частично-рекурсивных функций в математике) существует много других эквивалентных полных технологий с другими аналитическими операциями.
6. Доказана тесная связь между тектологией А. Богданова и формальной технологией, что позволяет считать последнюю математической версией первой, а в более широком контексте – рассматривать ОФТ как логически строго обоснованный вариант метаматематики, материалистической метафизики или точной философии.
7. Предложен формальный метод унифицированного описания абстрактных и реальных объектов, включая модели объектов разного уровня детализации. Метод позволяет работать с различными свойствами объектов – как с функциональными, так и с нефункциональными. Кроме того, он хорошо согласуется с базовыми концепциями ООП, что позволяет легко использовать его в компьютерных программах для решения различных задач.

8. На основе метода п.7 разработана ОФТ-версия общей теории свойств и функциональностей объектов, более корректная, чем онтология (или точная философия) М. Бунге.
9. В рамках ОФТ показано, что простейшие «двухвходовые» функциональности одиночных объектов могут образовывать по крайней мере 4-5 различных типов. При этом функциональность универсального булевского логического базиса (функционально полных наборов двоичных функций) относится только к одному из них, тогда как функциональный базис объектов химических и биохимических реакций (т.е. атомов и молекул) может использовать все пять типов. Таким образом, природа в своих системах использует все возможные типы функциональностей, а мы в своих самых мощных вычислительных инструментах – компьютерах, построенных из тех самых функционально-полных наборов двоичных функциональных элементов, – только одну.
10. Итогом разработки методов ОФТ явилось создание нового направления в теории автоматов – «теории гетерогенных автоматов», в том числе «теории гетерогенных слабоструктурированных самосинхронных автоматов с индивидуальными именами объектов операций». Важным результатом исследований в этих направлениях является доказательство возможности НЕизоморфной реализации автоматов, имеющих изоморфные графы. Этот результат с теоретической точки зрения также подтверждает предположение Л. Берталанфи о возможной общности законов функционирования различных систем в их различных физических воплощениях.
11. Разработан «метод назначенных функциональностей», позволяющий исследовать различные аспекты поведения систем, детали функционирования которых (т.е. их функциональные особенности) неизвестны или непрозрачны. Метод позволяет исследовать (в том числе) свойства биоподобных систем на начальных этапах эволюции, в частности, построить модель самоорганизации линейных структур и проанализировать сложность их самовоспроизведения.
12. Показана потенциальная эффективность использования гетерогенных объектов при конструировании технических систем (в частности, при разработке электронных схем).
13. Показана логическая эквивалентность возможностей функционально-полных наборов булевских функций и некоторых типов химических реакций. Получающиеся при этом решения оказываются значительно проще, компактнее и точнее, чем использование для доказательства аналогичных эквивалентностей λ -исчисления Чёрча или «градиентов концентраций химических реагентов» Г. Хакена.

14. Построены модели предельно простых (элементарных) объектов («квазиатомов»), проявляющие при нагревании свойства «расплавления», а при охлаждении – «кристаллизации». То есть определены необходимые и достаточные условия для самоорганизации элементарных объектов по типу кристаллических структур.
15. Введено строгое формальное определение эмерджентного свойства и эмерджентной функциональности, что не удавалось сделать с середины 20-го века – с момента введения Россом Эшби неформального определения.
16. Показано, что самовоспроизведение клеточных автоматов не требует и не порождает эмерджентных свойств и функциональностей (в рамках ОФТ).
17. Построены модели элементарных объектов («квазиатомов»), способных формировать устойчивые системы объектов типа «квазимолекул» с использованием функциональности, названной «спариванием электронов». То есть получены условия для самоорганизации объектов по типу сравнительно сложных молекул.
18. Разработана ОФТ-модель «потенциально эволюционной вселенной», которая оказалась весьма близкой к окружающей нас действительности.
19. Показана и детально исследована связь между информацией, описывающей структуру систем, её энтропией и функциональностью.
20. Доказана теорема о фиксированной точке самовоспроизведения, утверждающая, что при определенных условиях появление самовоспроизводящихся систем неизбежно.
21. Показана эффективность применения критерия сложности Колмогорова при ОФТ-анализе процессов самовоспроизведения, что противоречит господствующему в настоящий момент мнению.
22. Разработан сценарий эволюции простейших линейных биоподобных молекул вплоть до уровня появления аппарата трансляции генетического описания частично самовоспроизводящихся биоподобных систем (ЧСБС) в функциональные (белковые) компоненты ЧСБС.
23. Получена формула для расчёта скорости эволюции ЧСБС, которая оказалась неожиданно высокой.
24. Введено и строго обосновано определение универсальной программно управляемой технологической системы, значительно расширяющее класс универсальных технологических систем относительно универсальной машины Тьюринга.
25. Разработана и математически строго обоснована структура универсального программно управляемого синтезатора-анализатора различных объектов для любой конкретной формальной технологии.

26. Структура «синтезатора-анализатора» адаптирована для практических применений. На её основе разработаны очень компактные универсальные технологические системы (аналоги гибких автоматических производств – ГАП), а также универсальные программно управляемые устройства для следующих технологических областей:
- а) для связи компьютеров с внешним аналоговым миром (ДАП – дискретно-аналоговый процессор);
 - б) для многоцелевых (универсальных) программируемых аналого-цифровых «систем на кристалле» («system-on-a-chip»);
 - с) для многоцелевого химического анализатора-синтезатора;
 - д) для простейшей микроминиатюрной универсальной программируемой «лаборатории на кристалле» («lab-on-a-chip»);
 - е) для мини-, микро- и нано-ГАП.
27. Разработаны принципы построения практических структур универсальных синтезирующих автоматов (САВТов) минимальной сложности, способных к самоодернизации (самоусовершенствованию), саморемонту и самовоспроизведению. В структуре САВТ могут быть объединены самые различные «синтезаторы» для самых различных технологий, а конструкция САВТ оказывается на несколько порядков проще конструкции «Самовоспроизводящейся лунной фабрики НАСА».
28. Доказана ограниченность действия гипотезы Тьюринга-Чёрча применительно к объектам реального физического мира. Фактически показано, что тезис Тьюринга-Чёрча неверен для широкого класса реальных физических систем.
29. Разработана ОФТ-версия теории познания, в рамках которой доказано существование порога сложности для автоматически действующих систем, способных к познанию окружающей среды.
30. Построена модель «универсального познавателя» и доказана его способность познавать окружающий его одномерный, двумерный или трёхмерный мир. Поведение модели на начальных этапах хорошо коррелирует с поведением новорождённых высших животных, включая человека.
31. Продемонстрирована эффективность использования подходов и методов ОФТ при анализе социальных систем.
32. Доказана неэффективность революционных процессов в развитии социальных систем и эффективность эволюционных, в том числе быстрых, «революциеподобных», но всё же эволюционных по своей структуре процессов.
33. Показана бесперспективность создания «идеальных обществ» в ходе эффективного эволюционного развития.

34. Продемонстрирована глубокая схожесть (эквивалентность) базовых процессов генерирования новых решений («изобретений») в ходе эволюций биосферы, социума, науки, техники и технологий, что позволяет в перспективе разработать универсальные изобретающие алгоритмы (и системы).
35. Получено ОФТ-отношение, объясняющее рост функциональной и структурной сложности технологической цивилизации в ходе эволюции.
36. Показано место самой математики как технологии вычислений в системе других возможных технологий. Продемонстрированы возможности таких технологий в рамках единой метаматематической конструкции, основанной на расширенной концепции алгебраической системы как системы алгоритмов процессов над моделями абстрактных и реальных (физических) объектов.
37. Предложена концепция строгой (научной) версии метафизики, включая предварительную формулировку её аксиом.
38. Показано, что «мощность» (фактически «интеллектуальные способности») машины Тьюринга строго меньше «мощности» человеческого интеллекта.

УКАЗАТЕЛЬ ОПРЕДЕЛЕНИЙ

Определение:	Стр.
Автономный гетерогенный автомат	212
База технологии A	26
Бесконечно креативная операция	36
Бесконечно креативная технология	36
Гетерогенный автомат (ГА)	208
Гетерогенный слабоструктурированный автомат	208
Гетерогенные слабоструктурированные самосинхронные автоматы с индивидуальными именами объектов операций (ГССА с ИИОО)	210
«Известная» и «неизвестная» конструкция	51
Информационная ветвь (биоподобной технологии)	178
Информационно-функциональная ветвь	178
Каноническое определение формальной технологии	26
Классифицирующая операция P_C	53
Конструкция	8, 27
Конструкционная полнота	158
Кратчайший эволюционный путь развития (КЭПР)	177
Креативная операция	35
Линейная технология	46
Метаматематическая версия алгебраической системы в ОФТ	7
«Общая формула» операций ОФТ	24
Объекты алгоритмических операций в ОФТ	25
Операция анализа	21
Операция F_0 определения наличия или отсутствия объекта	22
Операция P_E определения равенства двух объектов	22
Операция P_i поэлементного анализа	55
Операция F_S примитивного синтеза	33
Операция F_d примитивной декомпозиции	45

Определение:	Стр.
Операция F_{ES} расширенного примитивного синтеза	40
Операция F_t синтеза	34
Операция P_A «случайного стационарного отображения»	48
Операция F_b «случайной» декомпозиции	47
Операция типа декомпозиции	21
Операция типа синтеза	21
Операция F_k «фиксированной дезинтеграции»	41
Операция (предикат) P_s , определяющая свойство объекта «иметь заданную форму»	23
ОФТ-функция	30
Полная технология	43
Представительная конструкция	56
Рациональное разбиение классифицирующих операций	53-54
Рекурсивная схема	67
Свойства объектов	112
Синтез в ОФТ	31
Состояние объекта	120
«Стационарная» операция	48
Тезис Чёрча	190
Универсальная технологическая система	63, 66
Функциональные свойства объекта	122
Функциональная сложность C_A автомата случайного синтеза	163
Шаг технологии	29
Эволюционный путь развития (ЭПР)	177
Эволюционная технология	17, 37
Элемент базы	26
Эмерджентное свойство	141
Эмерджентные функциональные и нефункциональные свойства	143-144

ЛИТЕРАТУРА

1. Алгоритмы в современной математике и её приложениях: Материалы международного симпозиума, Ургенч, Узбекистан, 16-22 сент. 1979 г. – Новосибирск: СО АН СССР, 1982. – 364 с.
2. *Аристотель*. Метафизика. Переводы. Комментарии. Толкования / Сост. и подготовка текстов С.И. Еремеев. – СПб.: Алетейя, 2002. – 832 с.
3. *Богданов А.А.* Тектология: Всеобщая организационная наука. В 2-х кн. – М.: Экономика, 1989. Кн.1 – 304 с.; Кн. 2. – 351 с.
4. *Владимиров Ю.С.* Метафизика. – М.: БИНОМ, Лаборатория знаний, 2002. – 550 с.
5. *Витгенштейн Л.* Философские работы. Замечания по основаниям математики: пер. с нем. – М.: Генезис, 1994. Ч.2. Кн. 1. – 214 с.
6. *Гребенчиков Е.Н., Крылов С.М.* Онтология проектирования гетерогенных электронных систем // Онтология проектирования. – 2012. – № 1. – С. 65-71.
7. *Грин Б.* Элегантная вселенная. Суперструны, скрытые размерности и поиски окончательной теории: пер. с англ. / Под ред. В.О. Малышенко. Изд. 2-е. – М.: Едиториал УРСС, 2005. – 288 с.
8. *Кант И.* Пролегомены ко всякой будущей метафизике, могущей появиться как наука. Соч. В 6-ти томах. – М.: Мысль, 1965. Т.4. Часть 1. – С. 67-210.
9. *Карпов Е.М.* Об алгебре физических взаимодействий // Математическое обеспечение САПР: Межвуз. сб. научн. трудов. – Куйбышев: КуАИ, 1989. – С. 20-29.
10. *Колмогоров А.Н.* Комбинаторные основания теории информации и исчисление вероятностей. В кн.: Алгоритм, информация, сложность – М.: Знание, 1991. – С. 34-40. (Новое в жизни, науке, технике. Сер. «Математика, кибернетика», № 1).
11. *Колмогоров А.Н., Успенский В.А.* К определению алгоритма. В кн.: Алгоритм, информация, сложность. – М.: Знание, 1991. – С. 5-23. (Новое в жизни, науке, технике. Сер. «Математика, кибернетика», № 1).
12. *Корогодин В.И.* Информация и феномен жизни. – Пушкино, 1991. – 204 с.
13. *Крылов С.М.* Формальная технология и эволюция. Изд. второе, перераб. и доп. – Самара: СамГТУ, 2012. – 324 с. (ISBN 978-5-7964-1547-4).
14. *Крылов С.М.* Формальная технология в философии, технике, биоэволюции и социологии. – Самара: СамГТУ, 1997. – 180 с. (ISBN 5-7964-0008-8).
15. *Крылов С.М.* Онтология метанауки. Аксиомы, технологии, алгоритмы,

- эволюция. – LAP Lambert Academic Publishing, Saarbrücken, Deutschland, 2012. – 404 p. (ISBN 978-3-659-27964-5).
16. *Крылов С.М.* Неокибернетика. Алгоритмы, математика эволюции и технологии будущего. – М.: Изд-во ЛКИ, 2008. – 288 с.
 17. *Крылов С.М.* Перспективы метаматематических структур в науке / Вестник Самар. гос. техн. ун-та. Сер. Физ.-мат. науки, 2013. – № 2 (31). – С. 101-110.
(<http://www.mathnet.ru/links/6a367b2c723a8eb0ee748e644606e9e0/vsgtu1203.pdf>)
 18. *Крылов С.М.* Математические основы метафизики / Вестник Самар. гос. техн. ун-та. Сер. Физ.-мат. науки. – 2012. – № 1 (26), – С. 233-242.
(<http://www.mathnet.ru/links/188a909f5aa6c40085d5b4d6696c2e6c/vsgtu1043.pdf>)
 19. *Крылов С.М.* Формальная технология и универсальные системы I, II. / Кибернетика. – 1986. – Часть I. – № 4. – С. 85-89. Часть II. – № 5. – С. 28-31.
 20. *Крылов С.М.* Модели универсальных дискретно-аналоговых машин на основе машины Тьюринга // Электронное моделирование. – 1982. – № 3. – С. 6-10.
 21. *Крылов С.М.* Общая формальная технология и алгоритмическая «Теория исчисления объектов» / Компьютерные технологии в науке, практике и образовании: Труды одиннадцатой Междунар. межвуз. науч.-практ. конференции. – Самара: СамГТУ, 2012. – С. 27-30.
 22. *Крылов С.М.* Формально-технологические модели в общей теории систем // Известия Самарского научного центра РАН. – 2003. – Т. 5. – № 1. – С. 83-90.
 23. *Крылов С.М.* Системный анализ гетерогенных слабоструктурированных автоматов. I Основные определения и свойства // Вестник Самар. гос. техн. ун-та. Сер. Технические науки. – 2012. – № 1 (33). – С. 17-23.
 24. *Курдюмов С.П., Малинецкий Г.Г., Потапов А.Б.* Синергетика – новые направления. – М.: Знание, 1989. – 48 с. (сер. «Новое в жизни, науке, технике»).
 25. *Мальцев А.И.* Алгебраические системы. – М.: Наука, 1970. – 392 с.
 26. *Мальцев А.И.* Алгоритмы и рекурсивные функции. – М.: Наука, 1965. – 392 с.
 27. *Машины Тьюринга и рекурсивные функции / Г.Д. Эббинхауз, К. Якобс, Ф.К. Манн, Г. Хермес.* – М.: Мир, 1972. – 264 с.
 28. *Минский М.* Вычисления и автоматы. – М.: Мир, 1971. – 364 с.
 29. *Моисеев Н.Н.* Алгоритмы развития. – М.: Наука, 1987. – 304 с.
 30. *Николис Г, Пригожин И.* Познание сложного. – М.: Мир, 1990. – 344 с.
 31. *Пригожин И., Стенгерс И.* Порядок из хаоса: новый диалог человека с природой: пер. с англ. – М.: Прогресс, 1986. – 432 с.

32. *Рассел Б.* Словарь разума, материи и морали. – Европейский университет: Port-Royal, 1996. – 368 с.
33. *Толчев В.Н., Крылов С.М.* Многофункциональные дистанционные лаборатории для проведения реальных лабораторных работ и экспериментов // Вестник Самар. гос. техн. ун-та. Сер. Технические науки. – 2011. – № 1 (29). – С. 85-91.
34. *Фон Нейман Дж.* Теория самовоспроизводящихся автоматов: пер. с англ. – М.: Мир, 1971. – 284 с.
35. *Хакен Г.* Информация и самоорганизация: Макроскопический подход к сложным системам. – М.: Мир, 1991. – 240 с.
36. *Хакен Г.* Синергетика. – М.: Мир, 1980. – 408 с.
37. *Хокинг С.* Теория всего / С. Хокинг. Краткая история Вселенной. – СПб.: ТИД Амфора, 2010. – С. 482-495. (ISBN 978-5-367-01335-1).
38. *Шеннон К.* Математическая теория связи. В кн.: *Шеннон К.* Работы по теории информации и кибернетике: пер. с англ. – М.: Иностранная литература, 1963. – С. 243-332.
39. *Шенфилд Дж.* Степени неразрешимости: пер. с англ. – М.: Наука, 1977. – 192 с.
40. *Штернберг М.И.* Синергетика и биология // Вопросы философии. – 1999. – № 2. – С. 95-108.
41. *Шустер Г.* Детерминированный хаос. – М.: Мир, 1988.
42. *Щербак В.И.* Информационный артефакт генетического кода (кооперативная симметрия) и интерпретация его контекстуального и ассоциативного содержания. Рукоп. депонир. в КазНИИТИ, № 1694, 17 июня 1987 г. См. также «Journal of Theoretical Biology», v. 132, pp. 121-124, v. 162, pp. 399-401, v. 166, pp. 475-477. «BioSystems», v. 70, 2003, pp. 87-209).
43. Advanced Automation for Space Missions: Part 5. Replicating Systems Concepts: Self-Replicating Lunar Factory and Demonstration / Edited by Robert A. Freitas, Jr. and William P. Gilbreath. In: Proceedings of the 1980 NASA/ASEE Summer Study. Held at the University of Santa Clara, June 23 – August 29, 1980. NASA Conference Publication 2255.
44. *Ashby W.R.* An Introduction to Cybernetics, Chapman & Hall, London, 1956.
45. *Babbage Ch.* Passages from the Life of a Philosopher, corrected by A.W. Van Sinderen // Annals of the History of Computing, 1980, N 2, pp. 169-85. Chapter 8.
46. *Benacerraf P.* What Numbers Could Not Be // Philosophical Review, 1965, 74; pp. 47-73.
47. *Berlinski D.* The Language of Life. In: Complexity, Language, and Life.

- Mathematical Approaches. Ed. By J.L.Casti, A.Karlqvist. – Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1986, pp. 231-267.
48. *Blomberg C.* On the Appearance of Function and Organisation in the Origin of Life // *J. theor. Biol.* (1997), V. 187, pp. 541-554.
 49. *Bunge M., Sangalli A.A.L.* A theory of properties and kinds // *Int. J. of General Systems*, 1977, v. 3, pp. 183-190.
 50. *Bunge M.* The Furniture of the World, vol. 3, of the Treatise on Basic Philosophy. – D. Reidel Publ. Company: Dordrecht and Boston, 1977.
 51. *Casti J.L.* On System Complexity: Identification, Measurement, and Management. In: *Complexity, Language, and Life. Mathematical Approaches.* Ed. By J.L. Casti, A. Karlqvist. – Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1986, pp. 146-173.
 52. *Chaitin G.J.* A Century of Controversy over the Foundations of Mathematics. In: *C. Calude and G. Paun, Finite versus Infinite.* – Springer-Verlag, London, 2000, pp. 75-100
 53. *Chaitin G.J.* Proving Darwin. Making Biology Mathematical – New-York: Pantheon Books, 2012. – 124 p. (ISBN 978-0-375-42314-7).
 54. *Copeland J.B.* Super Turing-Machines // *Complexity*, 1998, 4, pp.30-32.
 55. *Copeland J.B., Proudfoot D.* Alan Turing's Forgotten Ideas in Computer Science // *Scientific American*, 280 (April), 1999, pp. 99-103.
 56. *Copeland J.B.* Hypercomputation // *Minds and Machines*, 12: 2002, pp. 461-502.
 57. *Gilbert W.* The RNA world // *Nature*, V. 319, 1986, p. 618 (20 February, 1986).
 58. *Harbour H.* What 'What Numbers Could Not Be', by Paul Benacerraf, Is. <http://hilton.org.uk/index.phtml>
 59. *Heylighen F.* The Growth of Structural and Functional Complexity during Evolution, in: *F. Heylighen & D. Aerts (eds.) «The Evolution of Complexity»* – Kluwer, Dordrecht, 1997.
 60. *Jones B.* General System Theory and Algorithm Theory // *Int. J. General Systems*, 1983, v. 9, pp. 157-160.
 61. *Fontana W., Buss L.W.* The barrier of objects: From dynamical systems to bounded organizations / In: *Boundaries and Barriers*, J. Casti and A. Karlqvist (eds.), Addison-Wesley, 1996, pp. 56-116.
 62. *Kampis G.* Self-Modifying Systems in Biology and Cognitive Science. – Oxford, Pergamon, 1991. – 544 p.
 63. *Kauffman S.A.* Emergent properties in random complex automata // *Physica D*,

- 1984, V. 10, pp. 145-156.
64. *Klyce B.* The RNA World. [Http://www.panspermia.org/rnaworld.htm](http://www.panspermia.org/rnaworld.htm), 2002.
 65. *Krylov S.M.* Formal Technology and Cognitive Processes // Int. Journal of General Systems, 1996, v. 24, pp. 233-243.
 66. *Krylov S.M.* Universal Programmable Completely Automated Factories-on-a-Chip. Proceedings of the COMS2004. Aug. 29 – Sept. 2, 2004. Edmonton, Canada. – MANCEF, 2004, pp. 269-273.
 67. *Langton C. G.* Self-reproduction in cellular automata // Physica D, 1984, vol. 10, pp. 135-144.
 68. *Laszlo E.* Evolution. The Grand Synthesis. – Boston: New Science Library, 1987.
 69. *Lovelace A.A.* Notes by the Translator. In: Faster than Thought. A Symposium on Digital Computing Machines. Ed. By B.V. Bowden. – London, 1957, pp. 362-408 (364-366).
 70. *Mao Ch. Labean, T.H., Reif, J.H., & Seeman, N.C.* Logical computation using algorithmic self-assembly of DNA triple-crossover molecules // Nature, 2000, v. 407, N 6803, pp. 493-496.
 71. *Pattee H.H.* Universal Principles of Measurement and Language Functions in Evolving Systems. In: Complexity, Language, and Life. Mathematical Approaches. Ed. By J.L. Casti, A. Karlqvist. – Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1986, pp. 268-281.
 72. *Penrose R.* Shadows of the Mind: A Search for the Missing Science of Consciousness. – Oxford, New York, Melbourne: Oxford University Press, 1994, 457 p.
 73. *Perrier J.-Y., Sipper M. and Zahud J.* Toward a viable, self-reproducing universal computer // Physica D, 1996, vol. 97, pp. 335-352.
 74. *Schmidhuber, J.A.* Algorithmic Theories of Everything. Technical Report IDSIA-20-00, Version 2.0; 20 Dec 2000. <http://www.idsia.ch/~juergen/>
 75. *Schmidhuber, J. A.* Computer Scientist's View of Life, the Universe, and Everything. In C. Freksa, ed., Foundations of Computer Science: Potential – Theory – Cognition // Lecture Notes in Computer Science, Springer, 1997, pp. 201-208.
 76. *Schneider, T. D.* Theory of molecular machines. I. Channel capacity of molecular machines // J. Theor. Biol. 1991, v. 148, pp. 83-123.
 77. *Siegelmann, H.T.* Computation Beyond the Turing Limit // Science, 268, 1995, pp. 545-548.
 78. *Sjolander, S.* On the Evolution of Reality - Some Biological Prerequisites and Evolutionary Stages // J. theor. Biol. 1997, v. 187, pp. 595-600.
 79. *Stewart, I.* Deciding the Undecidable // Nature, 1991, v. 352, pp. 664-665.

80. *Turchin, V.* The Phenomenon of Science. A cybernetic approach to human evolution - Columbia University Press, New York, 1977.
81. *Van Rootselaar, B.* On Bunge's theory of things // Int. J. of General Systems, 1977, v. 3, pp. 175-180.
82. *Whitehead, A.N.* Process and Reality. – New York: Free Press, 1929.
83. *Yockey, H.P.* Origin of life on earth and Shannon theory of communication // Computers and Chemistry, 2000, N 24, p. 105-123.
84. *Zalta, E.N.* The Theory of Abstract Objects. – Center for the Study of Language and Information, Stanford University. - Stanford (Metaphysics Research Lab), 1999.

ОГЛАВЛЕНИЕ

Предисловие.....	3
Введение.....	4
Глава 1. Метаматематика и теория алгоритмов.....	7
1.1. Метаматематические основы «Общей формальной технологии».....	7
1.2. Формальные технологии как алгоритмические системы.....	20
1.3. Различные типы технологий и их свойства. Эволюционные и полные технологии.....	33
Глава 2. Универсальные программно-управляемые технологические системы	62
2.1. Модель универсального синтезатора-анализатора объектов для различных технологий.....	62
2.2. Практические структуры универсальных программно-управляемых технологических систем.....	74
2.3. Многофункциональные аналого-цифровые программируемые системы обработки сигналов.....	79
2.4. Универсальный программируемый химический синтезатор-анализатор	94
2.5. Мини-, микро- и нано-ГАП.....	103
Глава 3. Функциональности объектов.....	111
3.1. Связь между свойствами и функциональностями.....	111
3.2. Функциональности как изменения состояний.....	119
3.3. Эмерджентные свойства и функциональности.....	141
Глава 4. Механизмы эволюции.....	151
4.1. Предпосылки.....	151
4.2. Молекулы как конечные автоматы.....	157
4.3. Возникновение процесса самовоспроизведения молекул.....	169
4.4. Кратчайший эволюционный путь развития и скорость эволюции на начальных этапах.....	177
Глава 5. Гетерогенные системы.....	189
5.1. Тезис Чёрча и «универсальный познаватель».....	190
5.2. Теория гетерогенных автоматов.....	204
5.3. Гетерогенные системы в электронике.....	221
Заключение.....	231
Основные результаты, полученные в рамках выполненных исследований по ОФТ.....	233
Указатель определений.....	238
Литература.....	240

Научное издание

КРЫЛОВ Сергей Михайлович

Метаматематические основы науки будущего

Редактор *Е.С. Захарова*
Компьютерная верстка *И.О. Миняева*
Выпускающий редактор *Н.В. Беганова*

Подписано в печать 30.12.13
Формат 60×84 1/16. Бумага офсетная
Усл. п. л. 14,34. Уч.-изд. л. 14,31.
Тираж 50 экз. Рег. № 274/13

Федеральное государственное бюджетное
образовательное учреждение
высшего профессионального образования
«Самарский государственный технический университет»
443100, г. Самара, ул. Молодогвардейская, 244. Главный корпус

Отпечатано в типографии
Самарского государственного технического университета
443100, г. Самара, ул. Молодогвардейская, 244. Корпус № 8