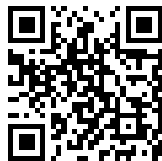


УДК 519.68: 681.51

**МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ ПОЗНАНИЯ
НА ОСНОВЕ ЭКСПЕРИМЕНТОВ***С. М. Крылов*Самарский государственный технический университет,
Россия, 443100, Самара, ул. Молодогвардейская, 244.**Аннотация**

В статье рассматриваются математические предпосылки для разработки теории познания, использующей информацию о реальных экспериментах, проводимых над реальными объектами с помощью формально-технологических аналогов машин Тьюринга. Такие аналоги, получившие название «универсальных синтезаторов-анализаторов объектов», теоретически позволяют выполнять синтез и анализ самых разнообразных конструкций (в том числе объектов, полученных путем соединения конечного числа исходных более мелких объектов, так называемых элементов базы) по различным алгоритмам в рамках некоторых ограничений, накладываемых алгоритмическими системами, названными формальными технологиями. Эти алгоритмические системы оказываются весьма близкими по своей формальной структуре и сути к структурам алгебраических систем Мальцева. Формальная близость таких алгоритмических систем позволяет, во-первых, выдвинуть гипотезу об алгоритмической основе практически всех окружающих нас понятных или непонятных (пока) нам физических процессов, что в какой-то степени объясняет широкую применимость математики к объяснению самых различных особенностей окружающего нас мира; во-вторых, она же позволяет сформулировать и доказать ряд теорем (называемых утверждениями), касающихся особенностей и ключевых характеристик алгоритмов познания в одномерных, двухмерных и трехмерных средах для различных формальных технологических систем, включая так называемую «теорему об эффективности накопленных знаний». Теорема (утверждение) оказывается применимой к очень широкому классу технологий, использующих в качестве операции анализа предикат равенства двух объектов. В статье приводятся и доказываются утверждения о существовании алгоритмов познания с различными наборами технологических операций типа синтеза и декомпозиции, а также с различными наборами операций анализа, включая предикат равенства, операцию «случайного стационарного отображения» (механизм действия которой неизвестен, а потому близок к концепции оракулов в машинах Тьюринга), операции определения формы объектов и др. Приводится структура соответствующего автоматически действующего устройства, названного «познавателем».

© 2015 Самарский государственный технический университет.

Образец для цитированияКрылов С. М. Математические основы теории познания на основе экспериментов // Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки, 2015. Т. 19, № 3. С. 534–558. doi: [10.14498/vsgtu1427](http://dx.doi.org/10.14498/vsgtu1427).**Сведения об авторе***Сергей Михайлович Крылов* (д.т.н., проф.; s_m_krylov@mail.ru), профессор, каф. вычислительной техники.

Ключевые слова: алгоритмы познания, физические процессы над физическими объектами, общая формальная технология, общая теория систем, свойства объектов, функциональности объектов, универсальный синтезатор-анализатор объектов, универсальный познаватель.

doi: <http://dx.doi.org/10.14498/vsgtu1427>

Существующие системы получения знаний основываются, как правило, на использовании различных баз знаний в заданной предметной области [1, 2]. Обычно в них объекты реального мира описываются в виде символьных (или текстовых) структур. Однако чёткого и строгого математического обоснования принципов выбора, разработки и функционирования таких систем нет. В частности, в [1] отмечается, что до сих пор создание такого рода систем — скорее искусство, чем наука. Отсутствие в них корректного формального представления моделей объектов реального мира приводит в итоге ко многим сложностям [2], которые существенно затрудняют разработку такого рода интеллектуальных систем. В настоящей статье предпринята попытка показать, что предложенная в рамках теории систем новая универсальная модель представления объектов реального мира снимает ряд таких сложностей. Более того, она позволяет построить строгую математическую версию теории познания. Направление, в котором эта модель появилась и неоднократно опробовалась, получило название «Общей формальной технологии» (ОФТ).

Сутью предложенного в [3–6] подхода является формальное представление любого реального (физического) объекта O_i в виде двух множеств-списков:

$$O_i = \langle \gamma_i, \mathbf{M}_i \rangle = \{ \{ \gamma_{i1}, \gamma_{i2}, \dots, \gamma_{in} \}, \{ \gamma_{ij} = \varphi_{ij}(\gamma_{sk}, \gamma_{il}, \dots, \gamma_{sm}); \dots \} \}, \quad (1)$$

где O_i — i -тый физический объект; γ_i — множество (список) интересующих нас его свойств (обозначенных символами типа γ_{ik}), включая, например, геометрические, описывающие форму объекта, его положение в пространстве и даже, при необходимости, направление, скорость движения и, к примеру, массу и (или) цвет; $\mathbf{M}_i$ — множество (список) интересующих нас функциональностей φ_{ij} объекта O_i (эти функциональности могут быть записаны в любой удобной форме — в виде функций, физических законов, алгоритмических процедур и т.д.); n — число параметров («свойств»), используемых в данном представлении 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 объекта O_i (в текущем представлении) с соседними (или удалёнными — если речь идёт о взаимодействии, например, через поле) объектами типа O_s , причём в общем случае значение некоторого параметра («свойства») γ_{ij} объекта O_i может зависеть как от параметров самого этого объекта (т.е. от «свойств» типа γ_{il} самого объекта O_i), так и от параметров («свойств») γ_{sk} других объектов типа O_s , контактирующих с данным объектом O_i или удалённых от него (точнее — удалённо-взаимодействующих с O_i). Подчёркнём, что функциональных зависимостей φ_{ij} в списке функциональностей $\mathbf{M}_i$ объекта O_i может быть столько, сколько нужно для его адекватного представления в рамках данной модели, на что указывает многоточие после формулы φ_{ij} для расчёта параметров γ_{ij} .

Однако всегда в конечном представлении любого объекта типа O_i это число конечно.

Форма записи (1) моделей объектов очень хорошо согласуется с версией аксиом «математически обоснованной метафизики» [4], а именно — полностью соответствует аксиомам 3, 6 и 7, приведённым в этой работе; а сама алгоритмическая (метаматематическая) система $\mathbf{T}$ над физическими объектами соответствует всем восьми аксиомам и записываемая аналогично алгебраическим системам (структурам) А. И. Мальцева [5, 6]:

$$\mathbf{T} = \langle \mathbf{B}, \mathbf{F}_T, \mathbf{F}_A \rangle = \langle \mathbf{B}, \mathbf{F} \rangle, \quad (2)$$

где $\mathbf{B}$ — конечное множество физических объектов материальной природы (атомов, молекул, деталей станка, компонентов какого-либо устройства, фрагментов здания, и т.д., и т.п.) или нематериальной природы (например, моделей указанных выше физических объектов, или объектов информационного характера — чисел, кодов и/или символов, конечных по времени отрезков аналоговых сигналов, и т.д. и т.п.). Все такие объекты в любой конкретной $\mathbf{T}$ называются конструкциями, причём часто в $\mathbf{B}$ выделяются конструкции, называемые элементами базы или базовыми элементами: $\mathbf{A} = \{a_1, a_2, a_3, \dots, a_n\}$, и если $\mathbf{B} = \mathbf{A}$, то мы имеем дело с каноническим вариантом задания некоторой формальной технологии (алгоритмической системы) $\mathbf{T}$. При каноническом задании $\mathbf{T}$ компонента $\mathbf{M}_i$ в представлении (1) обычно отсутствует (опускается) [6]. Её учёт (и реальное наличие) приводит к радикальному пересмотру формального представления (2), поскольку в этом случае возможна ситуация, когда набор функциональностей объектов в $\mathbf{B}$ (или в $\mathbf{A}$) оказывается достаточным для реализации эволюционного развития самой технологии $\mathbf{T}$ [6]. В этом случае любая эволюционная технология $\mathbf{T}_\Theta$ записывается так:

$$\mathbf{T}_\Theta = \langle \mathbf{B} \rangle. \quad (3)$$

Заметим, что формальная запись системы самовоспроизводящихся клеточных автоматов фон Неймана вполне соответствует (3) [5, 6]. В [3] показано, что эта технология «почти» эволюционна. Более того, наше интуитивное представление о начальном этапе земной биоэволюции (как о некоем «первичном бульоне» с хаотически движущимися в нём наборами различных молекул с различными свойствами и функциями) также совпадает с формой (3) [5].

Далее, множества $\mathbf{F}_T$ и $\mathbf{F}_A$ в алгоритмической системе («технологии») $\mathbf{T}$ определяются соответственно как конечные множества конечноместных технологических и аналитических операций над объектами-конструкциями из $\mathbf{B}$: $\mathbf{F}_T = \{F_1, F_2, \dots, F_m\}$ и $\mathbf{F}_A = \{P_1, P_2, \dots, P_n\}$, причём ни одна из операций в $\mathbf{F}_T$ и $\mathbf{F}_A$ не может быть выражена через другие [3, 6]. Кроме того, множества $\mathbf{F}_T$ и $\mathbf{F}_A$ могут быть пустыми и обозначаться одним символом $\mathbf{F} = \mathbf{F}_T \cup \mathbf{F}_A$. Легко доказать, что любая операция анализа из множества $\mathbf{F}_A$, имеющая конечное число возможных числовых и (или) нечисловых значений, может быть представлена в виде подходящего конечного набора предикатов, анализирующих только совпадение или несовпадение измеряемого параметра с заданным [7]. Именно по этой причине для обозначения различных аналитических операций множества $\mathbf{F}_A$ часто выбираются символы предикатов P_i , хотя в других работах по ОФТ для аналогичных аналитических операций используются и символы типа F_i [3–6].

Подобные совпадения глубинной сущности различных математических и технологических операций делает алгоритмические системы типа (2) ещё более близкими к алгоритмическим системам, задаваемым в форме алгебраических структур (алгебраических систем) в математике [3–8]. Более того, эти совпадения в какой-то степени объясняют и «непостижимую эффективность математики» при исследовании самой природы и тех явлений, которые мы наблюдаем или используем [9]. Почти очевидно, что все окружающие нас процессы — как природные, так и созданные нами — по своей сути процессы алгоритмические, будь они понятны нам, или пока еще непонятны или же не совсем понятны.

Таким образом, идея, положенная в основу концепции ОФТ, очень проста — все открытые нами (и пока не открытые) законы физики — не что иное, как предельно краткая (математическая) запись алгоритмов реализации этих законов. А раз это так, то наилучшим инструментом для их исследования, анализа и использования как раз и является сама математика, поскольку в её основе — те же алгоритмы, их фундаментальные особенности и свойства. Более того, исследование алгоритмических процессов во всём их многообразии — одна из важнейших задач современной математики [10].

Гипотеза об алгоритмической основе всех природных процессов в какой-то степени объясняет и, казалось бы, совершенно противоположную мысль о том, что в основе структуры окружающего нас мира лежит именно математика [11], а также очень близкую к ней по своей философской сути теологическую идею, заключающуюся во фразе: «математика — это наука Бога». Хотя в принципе обе базовые идеи — и та, что в основе устройства мира лежат процессы, описываемые алгоритмически, и та, что в основе устройства мира лежат математические структуры, как мы видели, при ближайшем рассмотрении оказываются весьма похожими. Собственно, предложенный в статье формальный анализ некоторых алгоритмических систем, способных работать как с физическими, так и с математическими объектами, подтверждает это со всей очевидностью.

Шагом формально-технологического алгоритма (или шагом технологии **T**) имеет смысл назвать однократное применение каких-либо технологических или аналитических операций из множеств **F_T** и **F_A** к объектам из **B**, включая используемые параметры операций и интересующие нас параметры получающихся в результате операций объектов. В общем случае такой шаг может быть записан в следующем виде:

$$F_i(x_1, x_2, \dots, x_k, \alpha_1, \alpha_2, \dots, \alpha_t) \rightarrow \langle y_1, y_2, \dots, y_l, \beta_1, \beta_2, \dots, \beta_s \rangle, \quad (4)$$

где $F_i \in \mathbf{F}_T \cup \mathbf{F}_A$; $x_1, x_2, \dots, x_k, y_1, y_2, \dots, y_l \in \mathbf{B}$; $\alpha_1, \alpha_2, \dots, \alpha_t$ — список входных параметров операции (если они есть); $\beta_1, \beta_2, \dots, \beta_s$ — список выходных параметров операции (если они есть), стрелка ($\rightarrow$) может рассматриваться как направление преобразований объектов (и параметров) в данной операции, но не как эквивалент операции равенства.

Кроме того, в формально-технологических алгоритмах, как и в классических вычислительных алгоритмах, могут использоваться операции безусловных и условных переходов по результатам операций анализа [3, 6].

Важную роль в различных формально-технологических алгоритмах играет так называемая «полнота» технологий. В ОФТ это понятие по своему

смыслу существенно отличается от понятий полноты системы частично рекурсивных функций в математике, полноты формальных теорий (в смысле выводимости формул или их отрицаний) или от полноты системы булевых функций в логике. В ОФТ технология $\mathbf{T} = \langle \mathbf{A}, \mathbf{F} \rangle$ называется полной, если она, во-первых, бесконечно креативна (т.е. позволяет для заданной базы $\mathbf{A}$ и заданного состава операций $\mathbf{F}$ генерировать счётно-бесконечное множество новых объектов), и, во-вторых, позволяет по любой конструкции данной технологии $y \in \mathbf{B}$ восстановить её состав и синтез, даже если раньше эта конструкция не синтезировалась в рамках данной конкретной технологии $\mathbf{T}$ (то есть, например, была получена «где-то в другом месте») [6].

Нетрудно видеть, что набор символов, описывающих различные алгоритмы синтезов различных объектов в любой конечной технологии $\mathbf{T}$, конечен, поэтому все эти синтезы (процессы, алгоритмы) могут быть лексико-графически упорядочены и взаимно однозначно отображены на натуральную числовую ось множества натуральных чисел $\mathbf{N}$ [12]. То есть мы для любого процесса (синтеза) в данной технологии $\mathbf{T}$ можем указать соответствующий ему номер из множества $\mathbf{N}$ и наоборот — по заданному номеру из $\mathbf{N}$ можем указать соответствующий ему синтез из $\mathbf{T}$.

Для формулировки ряда следующих теорем введём формально-технологическое определение предиката равенства двух объектов и некоторых технологических операций.

ОПРЕДЕЛЕНИЕ 1. Предикатом (аналитической операцией) равенства двух объектов x_1 и x_2 в любой технологии $\mathbf{T}$ называется операция анализа $P_E(x_1, x_2) \rightarrow 1$, если $x_1 = x_2$ и $P_E(x_1, x_2) \rightarrow 0$, если $x_1 \neq x_2$ [3, 6].

ОПРЕДЕЛЕНИЕ 2. Операциями типа синтеза F_S называются операции, соединяющие (неважно каким образом) минимум 2 объекта x_1 и x_2 между собой и получающие, таким образом, третий объект y в некоторой технологии $\mathbf{T}$: $F_S(x_1, x_2) \rightarrow \langle x_1 * x_2 \rangle = \langle y \rangle = y$, где «*» — символ операции соединения (без уточнения и конкретизации). Как и в самом общем случае (4), операции типа синтеза могут иметь или не иметь параметры [3, 6].

Теперь мы можем сформулировать ряд теорем относительно существования алгоритмов познания и их эффективности для различных типов технологий.

УТВЕРЖДЕНИЕ 1 («ТЕОРЕМА ОБ ЭФФЕКТИВНОСТИ НАКОПЛЕННЫХ ЗНАНИЙ»). Для любой технологии $\mathbf{T}$ с конечным множеством $\mathbf{A}$, с предикатом равенства двух объектов P_E , с конечным числом конечно-местных аналитических и технологических операций (типа синтеза) с конечным числом параметров, принимающих конечное число значений, существует алгоритм, перечисляющий в этой технологии синтезы новых получаемых в ней объектов (конструкций).

Доказательство. Согласно выполненному выше анализу, в таких технологиях процессы синтеза образуют перечислимое пространство синтезов, имеющее взаимно однозначное отображение на натуральный ряд [12]. Назовём соответствующие числа основными номерами лексикографически-упорядоченного пространства синтезов. Введем еще одни — дополнительные номера для тех синтезов, которые дают новые для данной технологии объекты и положим для исходного дополнительного номера i такого синтеза $i = 0$,

а для последнего дополнительного номера синтеза, также дающего новый объект, — символ k . Естественно, что начальное значение k тоже равно нулю: $k = 0$.

Далее выполняем синтезы из их лексикографического порядка, начиная с первого, каждый раз проверяя результат x очередного синтеза с помощью предиката P_E , пока не получим $P_E(x, \square) = 1$ (здесь символ $\square$ означает пустую конструкцию, т.е. в данном случае значение предиката $P_E(x, \square) = 1$ говорит о том, что x — пустая, несуществующая конструкция или, другими словами, — текущий синтез никакой конструкции не даёт). В этом случае просто переходим к следующему (очередному) синтезу в их лексикографическом порядке, не меняя текущее значение дополнительного номера i . Если же $P_E(x, \square) = 0$, то здесь возможны следующие варианты. Если раньше уже был хоть один синтез, для которого имело место $P_E(x, \square) = 0$ (то есть уже была получена какая-то конструкция и текущий дополнительный номер i равен или больше 1), то сравниваем полученный объект x с полученными ранее, восстанавливая эти объекты по дополнительным номерам их синтезов и сравнивая эти объекты с помощью $P_E(x_m, x)$ (здесь m — дополнительные номера синтезов, дающих новые объекты, начиная с самого первого такого синтеза, для которого $i = 1$, и кончая последним — с дополнительным номером k), пока не получим $P_E(x_m, x) = 1$, т.е. полученный в текущем синтезе объект x уже имеет более простой синтез точно такого же объекта x_m . В этом случае переходим к проверке следующего в лексико-графическом порядке синтеза из их последовательности. Если же ситуация $P_E(x_m, x) = 1$ не наступит вплоть до получения и сравнения последнего нового объекта x_k (из полученных ранее), то увеличиваем номер k на единицу и ставим его в соответствие текущему лексико-графическому номеру синтеза из их лексико-графического списка (т.е. полученный объект — действительно новый, получаемый в текущем синтезе с соответствующим лексико-графическим номером). Естественно, если во втором списке дополнительных номеров (для синтезов, дающих новые объекты) их ещё нет, т.е. текущее значение $i = 0$, то присваиваем i значение 1, тем самым начиная перечисление новых объектов, получаемых в анализируемой технологии, и продолжаем дальше испытывать очередные синтезы из их лексико-графического списка по рассмотренному выше алгоритму (т.е. на предмет получения новых объектов в исследуемой технологии).

Таким образом, мы в ходе реализации этого алгоритма получаем список из двух колонок номеров, первая из которых содержит основные номера лексико-графического списка синтезов в порядке их возрастания, а вторая — дополнительные, в которой, в свою очередь, пронумерованы в порядке возрастания уже те синтезы, которые дают действительно новые объекты начиная с номера $i = 1$. Утверждение доказано. $\square$

В результате, во-первых, оказывается, что практически в любой технологии, в которой мы можем получать новые объекты и определять равенство (одинаковость) любых двух объектов, мы в состоянии все такие новые объекты уникальным образом обозначать (перенумеровывать, т.е. присваивать им разные «дополнительные» последовательные числа — «имена»), во-вторых, мы можем построить (и запомнить) таблицу, в которой в каждой строчке первой колонки, например, перечисляются номера синтезов из их лексико-графического списка, а во второй — дополнительные номера («имена») новых

объектов, которые дают эти синтезы. В дальнейшем, используя полученную таблицу, в случае необходимости получить копию того или иного объекта мы, зная его дополнительный номер («имя»), можем отыскать его синтез по его лексико-графическому номеру из первой колонки и, выполнив этот синтез, получить нужный нам объект.

По этой причине данное утверждение и получило название «теоремы об эффективности накопленных знаний». Оно гарантирует, что при некоторых достаточно простых условиях мы можем эффективно использовать информацию об уже исследованных возможностях соответствующей технологии.

В работах [3, 6] показано, что технология вычислительной математики (фактически — согласно тезису Тьюринга—Чёрча — технология частично рекурсивных функций) является полной (но не эволюционной) технологией. Полнота в ней достигается за счёт предиката равенства, используемого в функции минимизации [3, 6, 13]. Существуют также варианты полных технологий, в которых для достижения полноты не используется предикат равенства двух объектов в форме определения 1. Более того, для достижения полноты различных технологий могут использоваться операции анализа (предикаты), внутренний механизм (алгоритм) функционирования которых нам может быть неизвестен [3, 6]. Фактически это означает, что соответствующие алгоритмы используют так называемые «оракулы» — то есть некие «решатели проблем» с неизвестным способом получения правильного результата. Такие алгоритмы образуют неразрешимые в рамках классической теории вычислительных алгоритмов системы, поскольку не входят в систему частично рекурсивных функций (ЧРФ), на основе которых построена вся современная классическая теория вычислений [14]. Тем не менее человек, как, кстати, и живая природа, в своей творческой и познавательной деятельности широко и с успехом используют подобные операции [3, 5, 6]. Этот факт, собственно, и даёт основание считать, что потенциальные возможности (био)эволюционных технологий T_{Σ} , как и познавательных (когнитивных) технологий человека $T_{КЧ}$, превышают потенциальные возможности технологии ЧРФ ($T_{ЧРФ}$) [5, 6]. Обозначая такого рода потенциальные возможности двумя прямыми скобками «| »|», можно записать [5, 6]:

$$|T_{ЧРФ}| < |T_{\Sigma}|; \quad |T_{ЧРФ}| < |T_{КЧ}|. \quad (5)$$

В вычислительной математике аналогичным образом принято обозначать так называемую «вычислительную мощность» соответствующих вычислительных систем (машин Тьюринга, машин Минского, Колмогорова, Поста и т.д.) [15]. По аналогии соответствующие «потенциальные алгоритмические возможности» рассматриваемых алгоритмических систем можно назвать «алгоритмической» или «когнитивной» («интеллектуальной») мощностью.

В ОФТ важную роль играет модель устройства, сконструированного на основе модели машины Тьюринга (м.Т.) и обладающего аналогичными (в своей области применения, то есть для своей технологии T) свойствами, в первую очередь — свойством реализации любого алгоритмического процесса для заданной технологии T . Эта модель получила соответствующее название — «универсальный синтезатор-анализатор объектов» [3, 6, 7, 16]. Один из вариантов его структуры показан на рис. 1.

Одномерный массив ячеек хранения (в которых могут «храниться» как

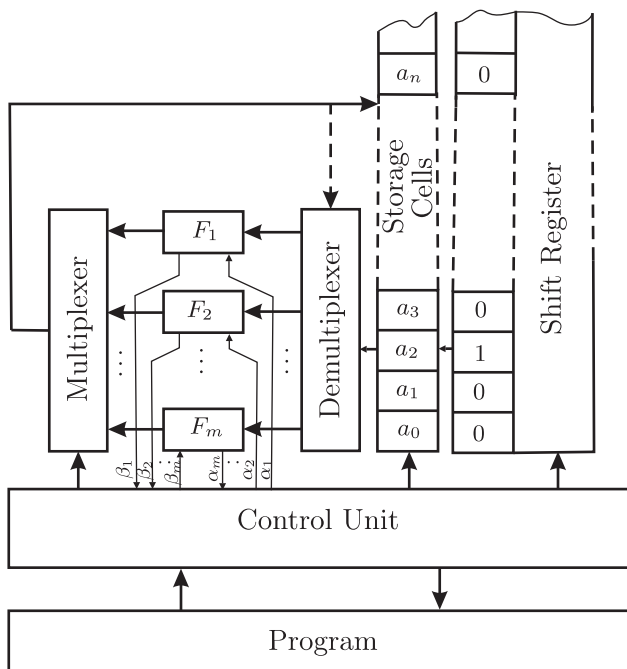


Рис. 1. Структура универсального синтезатора-анализатора объектов
[Figure 1. The structure of the universal object synthesizer-analyzer]

объекты из $\mathbf{A} \cup \mathbf{B}$, так и, естественно, «пустые» конструкции) вместе с блоком сдвигателя (рис. 1) образуют аналог одномерной, наращиваемой по мере необходимости (в одну сторону — вверх), ленты м.Т. Сдвигатель — это фактически регистр сдвига, хранящий только в одном разряде единицу (называемую «указателем»), а во всех остальных разрядах — нули. Для управления положением указателя при любой длине регистра требуется всего два управляющего сигнала — «сдвиг влево (вниз)» и «сдвиг вправо (вверх)» на один разряд, поэтому при его наращивании число управляющих сигналов не меняется.

Для извлечения нужного объекта из нужной ячейки хранения (ЯХ) указатель устанавливается в положение, указывающее на эту ячейку, и выполняется команда извлечения объекта, который далее через «распределитель» помещается в нужную технологическую ячейку (ТЯ) массива $F_1, \dots, F_m$, выполняющего в совокупности все технологические и аналитические операции множества $\mathbf{F} = \mathbf{F}_T \cup \mathbf{F}_A$ заданной технологии $\mathbf{T}$ в соответствии с формулой (4). Аналогичным образом выполняется размещение объекта-результата (результатов) выполненной операции из множества $\mathbf{F}$: объект-результат извлекается из соответствующей технологической ячейки $F_1, \dots, F_m$ и через «коммутатор» помещается в ту (свободную) ячейку хранения, на которую указывает «указатель». Если же основным результатом операции (анализа) является (числовое или нечисловое) значение соответствующего параметра β_r , то оно поступает в устройство управления для последующей обработки, включая выполнение команд (условных и безусловных) переходов по результатам анализа значения β_r [3, 6, 17, 18].

Легко доказывается (и проверяется), что для технологии ЧРФ структура универсального синтезатора-анализатора объектов (которую для удобства ссылок обозначим как S_U)), изображённая на рис. 1, фактически имитирует работу соответствующей м.Т. [3, 6, 17, 18]. Более того, S_U может использоваться в качестве модели для доказательства существования и эффективности различных аналитических, технологических и когнитивных алгоритмов [7, 19] — наподобие того, как м.Т. используется аналогичным образом для оценки возможностей различных алгоритмов обработки и преобразования информации, включая аналоговую [13, 20, 21]. Именно с помощью таких моделей была доказана теорема о неэквивалентности некоторых формально-технологических алгоритмических систем и систем ЧРФ [6, 22], подтверждающая соотношения (5).

Для понимания ряда следующих утверждений нам необходимо расширить понятийный аппарат ОФТ, чтобы с его помощью можно было работать с рядом других технологий, а также с так называемыми «неопределёнными» и «частично неопределёнными» операциями и технологиями.

ОПРЕДЕЛЕНИЕ 3. Операцией примитивной декомпозиции называется операция $F_D(x) \rightarrow \langle y, a \rangle$, позволяющая «отсоединить» элемент $a \in \mathbf{A}$ от конструкции x , если x — результат хотя бы однократного применения операции синтеза F_S к элементам из множества $\mathbf{A} \cup \mathbf{B}$:

$$F_D(x) \rightarrow \langle y, a \rangle, \text{ если } x \text{ получен в результате } F_S(y, a) \rightarrow x; \quad y, a \in \mathbf{A} \cup \mathbf{B}.$$

Как и раньше в примере (4), операции F_D могут иметь и не иметь параметры.

Иногда один тип операций примитивной декомпозиции оказывается недостаточным или неудобным для достижения полноты технологии. В этом случае оказывается полезным следующее определение.

ОПРЕДЕЛЕНИЕ 4. Конечное множество одноместных операций декомпозиции $F_{D1}, \dots, F_{Dn}$ в технологии $\mathbf{T}$ называется полным по декомпозиции, если они в совокупности позволяют разобрать любую конструкцию в данной $\mathbf{T}$ вплоть до уровня элементов базы.

Примером полного по декомпозиции множества технологических операций может служить множество $\{F_B\}$, состоящее из одной операции «случайной» декомпозиции. Операция случайной декомпозиции «разбирает» («разбивает») исходный объект y при каждом своём применении на случайное, но конечное число компонентов, не меньших, чем отдельные элементы базы $\mathbf{A}$:

$$F_B(y) \rightarrow \langle x_1, x_2, \dots, x_k \rangle; \quad x_1, x_2, \dots, x_k \in \mathbf{A} \cup \mathbf{B} \text{ [3].}$$

ОПРЕДЕЛЕНИЕ 5 (ОПРЕДЕЛЕНИЕ «НЕОПРЕДЕЛЕННОЙ», «НЕУСТАНОВЛЕННОЙ» или ПРОСТО «НЕИЗВЕСТНОЙ» ОПЕРАЦИИ). Операция $F \in \mathbf{F}$ называется неустановленной (неопределённой или просто «неизвестной») в технологии $\mathbf{T} = \langle \mathbf{B}, \mathbf{F} \rangle$, если мы не имеем никакой информации о её типе (то есть является ли она операцией типа синтеза, типа декомпозиции или операцией анализа), о её аргументности и характере используемых операндов (то есть о числе одновременно участвующих в ней объектов операции и их характере), о характере и числе результатов операции (вещественные объекты или информационные,

то есть параметры, их число и порядок следования) и т.д., за исключением того факта, что сама такая операция существует [3].

ОПРЕДЕЛЕНИЕ 6. Технология **T** называется частично неопределенной («частично неизвестной»), если **T** содержит неопределенные («неизвестные») операции и конструкции, а также некоторые «известные» операции, т. е. операции с известной функцией, известным числом и типами объектов операций (конструкций и/или параметров), известным порядком и типами результатов операции, известным способом управления выполнением этой операции [3].

Важное место во многих приводимых ниже утверждениях занимает условие отсутствия в составе операций исследуемых технологий *реальных* операций типа $F_k(x)$ — операций, позволяющих *реально* получать часть элемента x до бесконечности, в том числе если этот элемент — элемент базы (или его «часть»). Эти операции сознательно исключены из рассмотрения с тем, чтобы обеспечить стабильность множества элементов базы технологии на определенный конечный отрезок времени, в течение которого достигается полнота исследуемой технологии. После достижения полноты (с необходимой степенью достоверности) данное ограничение может быть снято. Во многих реальных технологиях на начальных этапах их развития операции типа F_k на самом деле отсутствуют в множестве **F** и могут появляться лишь на достаточно высоких ступенях развития соответствующих технологий благодаря изобретению (созданию) весьма сложных конструкций. Например: операции расщепления атомов — фундаментальных элементов базы фактически всех земных технологий — стали возможными для «осмысленной» практической реализации только после появления и развития ядерных технологий и изобретения соответствующих установок. Операции нецелочисленного деления с соответствующей системой записи любых дробных (нецелых) чисел в математике появились намного позже операций сложения, умножения и целочисленного деления целых чисел, и т. д. Таким образом, исключая *реальные* операции типа F_k из первоначального рассмотрения, мы тем самым предполагаем, что база технологии остается достаточное время устойчивой относительно входящих в нее операций и элементов и не меняется в ходе выполнения различных алгоритмов синтеза и анализа.

На сегодняшний день для S_U доказан ряд утверждений, одним из ключевых среди которых является следующее.

УТВЕРЖДЕНИЕ 2. Если в S_U с частично неопределенной («частично неизвестной») технологией

$$\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, F_m, P_E, \dots, P_k\} \rangle$$

имеются пустые ячейки хранения, операция P_E известна, а среди $F_1, \dots, P_k$ отсутствуют операции типа F_k и имеется неопределенная («неизвестная») операция синтеза, выполняемая над двумя элементами, и неопределенная («неизвестная») операция примитивной декомпозиции F_D , то существует алгоритм, который находит F_D , базу **A**, операцию синтеза и «формулирует» гипотезы относительно сущности и характеристик остальных операций.

Полное доказательство утверждения 2 (утв. 2) приведено в [3]. Оно основано на выдвижении ограниченного ряда гипотез о сущности тех или иных

неопределённых («неизвестных») операций в $\mathbf{T}$ с последующей их экспериментальной проверкой. Чтобы дать представление о технике доказательств такого рода утверждений, приведём его начало.

Доказательство. На первом (из трёх) этапе соответствующими гипотезами могут быть всего две:

- Γ^1) все операции в $\{F_1, \dots, F_m, \dots, P_k\}$, за исключением P_E — операции с одним операндом;
- Γ^2) все операции в $\{F_1, \dots, F_m, \dots, P_k\}$ — операции без параметров.

Для проверки гипотез Γ^1 и Γ^2 S_U должен последовательно применять операции $F_i \in \{F_1, \dots, F_m, \dots, P_k\}$ к различным элементам $b_j \in \mathbf{B}$. В соответствии с (4) возможны следующие результаты:

- a) $F_i(b_j) \rightarrow \langle b_j \rangle$ (ничего не произошло);
- b) $F_i(b_j) \rightarrow \langle x_1, x_2 \rangle$ (результатом являются 2 или больше элементов (конструкций), отличных от b_j);
- c) $F_i(b_j) \rightarrow \langle b_j, \beta_i \rangle$ (результатом является неизменяющаяся конструкция b_j и непустая конечная цепочка параметров $\beta_i = \beta_{i1}, \beta_{i2}, \dots, \beta_{in}; n \geq 1$);
- d) $F_i(b_j) \rightarrow \langle x_3, \beta_i \rangle$ (результатом являются новая конструкция (конструкция) x_3 и цепочка параметров β_i).

При этом возможны и особые случаи, например:

- e) $F_i(b_j) \rightarrow \langle \square \rangle$ (конструкция b_j исчезла);
- f) $F_i(b_j) \rightarrow S_U$ разрушен (фатальный результат).

Подобные итоги для простоты далее не рассматриваются.

Затем S_U сортирует результаты одного эксперимента путем сравнения b_j с результатами операции $F_i(b_j)$ с помощью операции P_E , и «формулирует» новые гипотезы относительно F_i .

Для случая а): $P_E(b_j, b_j) = 1$; $\Gamma_{1i}^1: F_i(b_j)$ не является операцией F_D .

Для случая б): $P_E(b_j, x_1) = 0$; $P_E(b_j, x_{12}) = 0$; (x_{12} — первый элемент в цепочке x_2); $P_E(b_j, x_{22}) = 0$; $\dots$; $P_E(b_j, x_{n2}) = 0$; (x_{n2} — последний элемент в цепочке x_2); $\Gamma_{1i}^2: F_i(b_j)$ — возможно, является F_D .

Для случая в): $P_E(b_j, b_j) = 1$; b_j и β_i несравнимы; $\Gamma_{1i}^3: F_i(b_j)$ не является F_D .

Для случая д): $P_E(b_j, x_{13}) = 0$; (x_{13} — первый элемент в цепочке x_3); $\dots$; $P_E(b_j, x_{m3}) = 0$; (x_{m3} — последний элемент в цепочке x_3); b_j и β_i несравнимы; $\Gamma_{1i}^4: F_i(b_j)$ — возможная F_D .

Чтобы различить случаи б) и д), достаточно иметь возможность детектировать появление параметров в результатах операции $F_i(b_j)$, что может сделать S_U со слегка модифицированным устройством управления.

Результаты одного эксперимента $F_i(b_j)$ размещаются в пустых ячейках хранения, а его «описание» (т. е. номер i выполненной операции, номер j опробованного элемента, номера ячеек хранения, в которых хранятся результаты эксперимента, номера возможных гипотез) записывается в свободных ячейках памяти микроконтроллера S_U .

Похожим образом проверяются другие неопределённые («неизвестные») операции и другие элементы из множества $\mathbf{B}$. В итоге на первом этапе оказывается возможным получить с достаточной степенью достоверности предположительный набор операций типа F_D и предположительный набор элементов базы $\mathbf{A}$, которые могут быть уточнены на последующих этапах.

На втором этапе предпринимается попытка найти операции синтеза (типа F_S) среди оставшихся непроверенными на причастность к F_D неопреде-

ленных операций. При этом используется несколько иной конечный набор первоначальных гипотез, да и текущие гипотезы на предмет совпадения с операциями типа F_S формулируются несколько иначе.

Наконец, после нахождения и проверки операций на соответствие F_S возможен поиск и формулировка предположений о сущности и числе используемых операндов остальными неопределёнными («неизвестными») операциями [3]. □

Методика, использованная в доказательстве утв. 2 для выделения операций синтеза, также оказывается пригодной для определения таких их важных характеристик, как число и типы объектов, над которыми операции выполняются. В результате оказывается, что условия утв. 2, относящиеся к выделению различных типов операций синтеза, которые работают не с двумя, а с большим числом исходных объектов, можно упростить.

УТВЕРЖДЕНИЕ 3. Если в S_U с частично неопределённой («частично неизвестной») технологией

$$\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, F_m, P_E, \dots, P_j\} \rangle$$

имеются пустые ячейки хранения, операция P_E известна, а среди $F_1, \dots, P_j$ отсутствуют операции типа F_k , имеются неопределённые операции синтеза, аналогичные F_S и выполняемые над любым конечным числом элементов, а также неопределённая операция F_D , то существует алгоритм, который находит F_D , базу $\mathbf{A}$, операции синтеза и «формулирует» гипотезы относительно сущности и характеристик остальных операций [3].

Отсутствие среди $F_1, \dots, P_j$ неопределённых операций синтеза может привести к следующему утверждению [3].

УТВЕРЖДЕНИЕ 4. Если в S_U с частично неопределённой технологией

$$\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, F_m, P_E, \dots, P_j\} \rangle$$

имеются пустые ячейки хранения, операция P_E известна, среди $F_1, \dots, P_j$ отсутствуют операции типа F_k и операции синтеза и имеется неопределённая операция F_D , то можно определить лишь базу $\mathbf{A}$ и операции, похожие на F_D .

Чтобы различать «случайные» (типа $\{F_B\}$) и «неслучайные», т.е. «стационарные» (приведённые в [19]) наборы полных по декомпозиции операций, введём небольшое уточнение.

ОПРЕДЕЛЕНИЕ 7. Множество операций декомпозиции называется множеством «неслучайных» (стационарных) операций, если результаты операций точно воспроизводимы любое число раз при точном повторении условий операций, и множеством «случайных» (нестационарных) операций декомпозиции, если результаты операций меняются от эксперимента к эксперименту при повторении условий проведения операций.

УТВЕРЖДЕНИЕ 5. Если в S_U с частично неопределённой технологией

$$\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, F_m, P_E, \dots, P_j\} \rangle$$

имеются пустые ячейки хранения, операция P_E является известной, а среди $F_1, \dots, P_j$ отсутствуют операции типа F_k и имеются неопределённые

операции синтеза, аналогичные F_S , выполняемые над конечным числом элементов, и конечное множество неопределенных, полных по декомпозиции одноместных стационарных операций, то существует алгоритм, который находит множество полных по декомпозиции операций, базу $\mathbf{A}$, операции синтеза и «формулирует» гипотезы относительно сущности и характеристик остальных операций [3].

Теперь в условиях утверждений 2–5 осталась лишь одна сложная для технической реализации операция — известная операция определения эквивалентности двух объектов, которые могут содержать в принципе миллионы и миллиарды элементов базы, — операция P_E . Можно ли ее заменить другими, более простыми, операциями? Ответ и на этот вопрос — положительный. В [3, 6] доказан ряд теорем, касающихся полноты технологий с различными наборами операций синтеза, декомпозиций и анализа. Один из этих наборов помимо операций синтеза и декомпозиции содержит еще одну очень своеобразную и неспецифицируемую (т.е. недетализируемую, нерасшифровываемую) операцию анализа, названную «случайным стационарным отображением». Операция действительно как бы отображает некоторое (возможно, неизвестное) свойство какой-либо конструкции на конечное число дискретных (числовых) значений. При этом термин «случайное» в названии этой операции анализа указывает как раз на тот очень важный аспект, что при самом первом её применении к любому новому объекту, к которому раньше такая операция анализа не применялась, невозможно никаким образом угадать (предсказать) то значение, которое «выдаст» данная операция. Зато при всех последующих применениях этой же операции к этому же объекту (или к полностью аналогичным объектам) результат будет таким же, то есть *стационарным*. Отсюда и название «случайная стационарная операция анализа». Обозначим операцию анализа с такими особенностями символами P_A . Естественно, в силу своей декларируемой неспецифицируемости операция P_A является примером операций, соответствующих «оракулам» в алгоритмических системах Тьюринга.

Введём также ещё одну достаточно простую аналитическую операцию: $P_0(x) \rightarrow \beta$; причем $\beta = 0$, если $x = \square$ (напомним, что символ $\square$ означает «пусто», т.е. x — пустой, несуществующий объект); и $\beta = 1$, если $x \neq \square$ (то есть x — непустой объект). Заметим, что операция P_0 также может быть неспецифицируемой.

В [3] доказывается, что справедливо следующее утверждение.

УТВЕРЖДЕНИЕ 6. Если в S_U с частично неопределенной технологией

$$\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, F_m, P_0, P_A, \dots, P_j\} \rangle$$

имеются пустые ячейки хранения; P_0, P_A известны, причем P_A — «случайная» стационарная операция анализа, выполняется для всех конструкций и имеет примерно одинаковые частоты появлений различных значений параметра на различных конструкциях; среди $F_1, \dots, P_j$ отсутствуют операции типа F_k и имеются неопределенные операции синтеза, аналогичные F_S , выполняемые над конечным числом элементов, и неопределенная операция F_D , то существует алгоритм, который с вероятностью, сколь угодно близкой к единице, находит операцию декомпозиции F_D , базу $\mathbf{A}$ (с чётким

разделением на различные элементы), операции синтеза и «формулирует» гипотезы относительно сущности и характеристик остальных операций.

Появление в утв. 6 вероятностных оценок связано с тем простым фактом, что операция P_A , согласно её определению, может давать минимум всего два различных значения анализируемого параметра, например, 0 и 1, однако вследствие своей непредсказуемости это значение не зависит от конкретной, впервые предъявляемой для такого анализа конструкции, то есть равновероятно для различных таких конструкций. Следовательно, применяя эту операцию к фиксированной последовательности конструкций, получаемых из исходной путем последовательного присоединения к ней различных элементов базы, мы получим некоторую последовательность ноликов и единиц, по которой в конце концов сможем идентифицировать и саму исходную конструкцию. Однако это не обязательно случится быстро, после фиксированного числа шагов алгоритма. Процесс в принципе может и затянуться, поэтому окончательный факт, устанавливающий конкретный состав структуры исходной конструкции (что необходимо для достижения полноты технологии **Т**), может растянуться во времени на неопределённый отрезок [3]. Тем не менее, по мере развития технологии **Т**, т.е. по мере «освоения» ею всё новых и всё более сложных конструкций-объектов, даже те ошибки, которые возникли раньше в силу неверной оценки из-за недостаточности наработанного экспериментального материала («опыта»), могут быть скорректированы.

В окружающем нас мире подавляющее число объектов являются двух- или трёхмерными. Соответствующие технологии имеет смысл так и назвать — двух- или трёхмерными. По этим причинам было бы интересно доказать теорему, аналогичную утв. 6, в которой в качестве известной операции (предиката) анализа выступает предикат определения формы конструкции типа P_S .

ОПРЕДЕЛЕНИЕ 8. Операцией типа P_S (определения формы конструкции) называется случайная стационарная операция анализа $P_S(x) \rightarrow \beta$, где параметр результата операции $\beta \in \mathbb{N}$ ($\mathbb{N}$ — множество натуральных чисел) принимает *конечное* число различных *конечных* (целых) числовых значений на конструкциях различной формы, причём если число элементов базы в конструкциях велико и формы соответствующих конструкций схожи, то значение параметра β для таких конструкций может быть одним и тем же. Этим мы, во-первых, гарантируем возможность конечного числа значений P_S , а во-вторых — избегаем излишней сложности реализации P_S при анализе различных конструкций с большим числом элементов базы, формы которых действительно могут быть схожи.

Слова «случайная стационарная» операция в определении 8 имеют тот же смысл, что и в случае операции типа P_A . То есть они могут говорить о том, что алгоритм формирования числового кода формы конструкции нам неизвестен, но для простых конструкций, в которых число элементов базы невелико, он чётко срабатывает всегда, когда соответствующая конструкция анализируется с помощью P_S .

Сформулируем теперь соответствующее утверждение.

УТВЕРЖДЕНИЕ 7. Если в S_U с частично неопределенной двух- или трёхмерной технологией

$$\mathbf{T} = \langle \mathbf{B}, \{F_1, \dots, F_m, P_S, \dots, P_j\} \rangle$$

имеются пустые ячейки хранения, операция P_S известна, имеются источники неизвестных конструкций $y_j \in \mathbf{B}$, различные элементы базы имеют различные значения по предикату P_S и «случайная» стационарная операция анализа $P_S(x)$ выполняется для всех конструкций, включая пустую, обеспечивая получение «изображения формы конструкции», а среди $F_1, \dots, P_j$ отсутствуют операции типа F_k и имеются неопределённые операции синтеза, аналогичные F_S , выполняемые над конечным числом элементов, и конечное множество неопределённых, полных по декомпозиции стационарных операций, то существует алгоритм, который находит с вероятностью, сколь угодно близкой к единице, множество полных по декомпозиции операций, базу $\mathbf{A}$ (с чётким разделением на различные элементы), операции синтеза и «формулирует» гипотезы относительно сущности и характеристик остальных операций.

Доказательство см. в [3].

Для исследования ряда вопросов, связанных с теорией познания, и в первую очередь — вопросов познания окружающего мира человеком, модель, представленная на рис. 1, не очень удобна — она неподвижна, тогда как человек в ходе изучения окружающего мира имеет возможность перемещаться в пространстве и часто именно так и поступает. Поэтому для большей адекватности была предложена другая модель, способная к подобного рода перемещениям. Простейший её вариант, опять же основанный на модели м.т. и способный перемещаться вдоль одномерного (для начала) массива ячеек хранения, представлен на рис. 2.

Отличие модели, представленной на рис. 2, от модели S_U заключается в том, что теперь одномерный массив ячеек хранения неподвижен, а сам универсальный синтезатор-анализатор объектов (который обозначим как S'_U) может перемещаться вдоль этого массива с помощью механизма сдвига и динамических операций (команд) из множества $\mathbf{G}$, в данном случае — команды сдвига на одну ячейку влево (G_L) и на одну ячейку вправо (G_R). Для полноты картины введена также ещё одна «динамическая» операция — оставаться на месте, то есть «ничего не делать» (G_0). Как и прежде, S'_U способен извлекать объекты (если они там есть) и помещать их в ту или иную технологическую ячейку для выполнения соответствующей операции, а затем

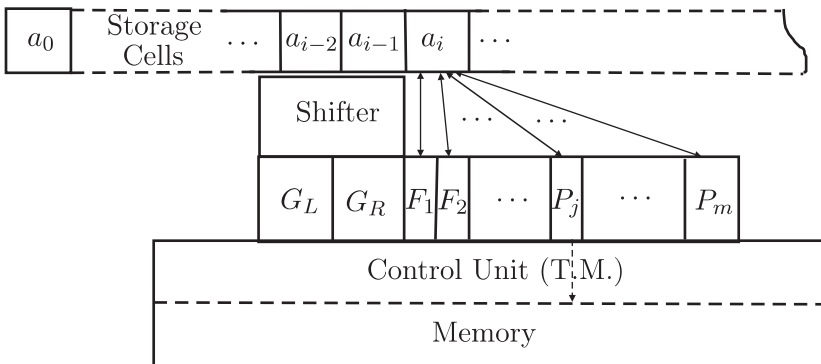


Рис. 2. Упрощённая структура познавателя
[Figure 2. A simplified structure of the cognizer]

возвращать полученный (полученные) объект-результат (объекты-результаты) в ту ячейку (ячейки), на которую(-ые) указывает указатель. В последнем случае, т.е. когда итогов (объектов) операции несколько, они размещаются в порядке, заданном описанием этой операции в виде (4). Возможно также извлечение объекта из ЯХ в любую ТЯ и возврат его обратно без выполнения соответствующей технологической операции. В этом случае сама ТЯ служит как бы буферной ячейкой хранения для извлекаемого объекта. В принципе, весь блок ТЯ может содержать даже специальную общую ЯХ для такого рода операций, используемую либо для последующего помещения объекта в соответствующую ТЯ, либо для возврата объекта обратно в текущую ЯХ. При этом между операциями извлечения и возврата объекта обратно могут выполняться любые другие операции S'_U .

Обозначим такую операцию «временного извлечения объекта» из некоторой ячейки a_j с последующим его возвратом обратно в эту же ЯХ символами M_F и M_B соответственно. Тогда запись типа $M_F M_B$ будет означать как раз то, что мы только что записали — извлечение возможного объекта из некоторой ячейки, на которую указывает указатель, и сразу же — возврат этого объекта обратно без каких-либо промежуточных операций. А запись типа $M_F P_A(a_j) M_B \rightarrow \beta$ будет означать анализ («измерение») содержимого ячейки a_j после временного извлечения из этой ячейки возможного объекта и перед его возвратом обратно.

Управление переходами из одного состояния S'_U в другое и выполнение тех или иных операций (команд) реализует микроконтроллер с памятью, вычислительные возможности которого соответствуют вычислительной мощности м.Т., а выполняемая в S'_U технология может быть практически полностью аналогична соответствующей технологии S_U . Ещё одной особенностью S'_U является возможность выполнения некоторых операций анализа непосредственно в ячейках линейного массива ячеек хранения, на что указывают односторонние стрелки на рис. 2.

Для модели S'_U легко доказывается следующее утверждение.

УТВЕРЖДЕНИЕ 8. *При идентичности содержимого ячеек хранения и идентичности реализуемых в S_U и S'_U операций оба устройства могут моделировать (имитировать) работу друг друга, т.е. их «алгоритмические возможности» (мощности) эквивалентны [3].*

Как и прежде, ряд операций соответствующих технологий, включая технологические операции, операции анализа и динамические ($G_L, G_R, G_0 \in \mathbf{G}$), могут быть неопределёнными («неизвестными»).

В доказательствах утверждений 2–7 важную роль играет наличие пустых ячеек хранения, в которых могут размещаться результаты проводимых в S'_U экспериментов при поиске тех или иных типов операций и тех или иных конструкций-объектов, включая элементы базы. Таким образом, чтобы можно было применить к S'_U приведённые выше утверждения (и соответствующие технологии), нам необходимо показать, что при некоторых условиях это возможно.

УТВЕРЖДЕНИЕ 9. *Если в S'_U с частично неопределённой технологией*

$$\mathbf{T}' = \langle \mathbf{B}, \{F_1, \dots, P_A, (P_S), \dots, P_j, G_L, G_R, G_0\} \rangle$$

операция анализа $P_A(P_S)$ известна; к группе неопределённых динамических

операций имеется доступ, независимый от остальных операций, а линейный массив ячеек хранения содержит непустые ячейки с конструкциями $b_i \in \mathbf{B}$, отличающимися по P_A (или P_S), и хотя бы один достаточно длинный конечный непрерывный массив пустых ячеек, причем значение по P_A (P_S) для пустых ячеек отличается от непустых; то существует реализуемый S'_U алгоритм, обеспечивающий нахождение операций G_L , G_R , G_0 , и формирующий в памяти S'_U адекватную (изоморфную) модель начального фрагмента массива ячеек хранения.

Доказательство. Для этого, как и раньше, используется конечная система фиксированных начальных гипотез относительно возможного устройства S'_U и окружающего его мира (касающаяся предполагаемой «сущности» линейного массива ячеек хранения). При этом возможны различные конкретные варианты реализации проверки этих гипотез, т.е. различные варианты реализации алгоритма, упоминаемого в самом утв. 9. Рассмотрим вариант, основанный на случайном выборе путей его продолжения.

В начальный момент времени t_0 в памяти S'_U фиксируются параметры (результат) операции анализа $P_A(x_i)$ (или $P_S(x_i)$) — в дальнейшем при упоминании P_A мы всегда будем иметь в виду и возможность использования вместо неё операции P_S) для некоторого объекта x_i (возможно, пустого) в некоторой ячейке хранения a_i . Поскольку какого-либо «внутреннего» представления о ячейке a_i (о её «имени», или номере, например), напротив которой находится S'_U (и которую он может «обозревать» с помощью P_A), а также о содержимом x_i этой ячейки в S'_U пока нет, операция P_A просто даёт в момент t_0 какое-то значение параметра β_0 , неизвестно к чему относящееся. Мы будем записывать это следующим образом:

$$P_A(t_0) \rightarrow \beta_0.$$

В следующий момент t_1 производится (по случайному выбору) одно из действий: G_L , G_R , G_0 и выполняется новый анализ результата операции $P_A(t_1)$. Возможны следующие варианты:

- а) $P_A(G_L(t_0)) = P_A(t_1) = P_A(t_0) = \beta_0$ (ничего не изменилось);
- б) $P_A(G_R(t_0)) = P_A(t_1) = P_A(t_0) = \beta_0$ (ничего не изменилось);
- с) $P_A(G_0(t_0)) = P_A(t_1) = P_A(t_0) = \beta_0$ (ничего не изменилось);
- д) $P_A(G_L(t_0)) = P_A(t_1) \rightarrow \beta_1 \neq P_A(t_0)$ (что-то изменилось);
- е) $P_A(G_R(t_0)) = P_A(t_1) \rightarrow \beta_1 \neq P_A(t_0)$ (что-то изменилось).

Как и ранее — в рассмотренном выше фрагменте доказательства утв. 2, а также в аналогичных утверждениях в [3] — «чудеса» и «фатальные исходы» (например, исчезновение объекта из ячейки хранения или разрушение S'_U) не рассматриваются.

В памяти S'_U фиксируется описание эксперимента в относительном времени — в виде $P_A(t_{i+1})$ равно (или не равно) $P_A(t_i)$. На этом этапе возможно формирование ложных гипотез о том, что движение влево или вправо ничего не меняет:

- для случая а) гипотезы Γ_1^a : $P_A(G_L(t_i)) = P_A(t_i)$;
- для случая б) гипотезы Γ_1^b : $P_A(G_R(t_i)) = P_A(t_i)$;
- для случая д) гипотезы Γ_1^d : $P_A(G_L(t_i)) \neq P_A(t_i)$;
- для случая е) гипотезы Γ_1^e : $P_A(G_R(t_i)) \neq P_A(t_i)$;

- тогда как гипотеза для случая **с**) $\Gamma_1^c: P_A(G_0(t_i)) = P_A(t_i)$ с очевидностью подтверждается.

В следующий момент времени t_2 вновь производится случайный выбор из $\{G_L, G_R, G_0\}$ и вновь фиксируются в памяти описание эксперимента и соответствующие гипотезы.

Рассмотрим ситуацию после n шагов, то есть после t_n . Если все ячейки, вдоль которых (случайным образом) двигался U'_S , заполнены конструкциями с одинаковым значением по P_A из $\mathbf{B} \cup \{\square\}$ (т. е. включая пустую конструкцию $\square$), то будут иметь место только случаи **а**), **б**) и **с**). Серия экспериментов должна быть продолжена до появления каких-либо изменений, т. е. до появления ситуаций **д**) или **е**), причем для выхода из ситуаций, когда в момент t_0 U'_S находился в середине длинного участка ячеек хранения, заполненных одинаковыми (по P_A) конструкциями, метод случайного выбора операций может быть заменен на более систематический поиск случаев **д**), **е**) — например, путем повторения выбранной операции $G_i \in G$ сначала m раз, а в случае неуспеха — с другой операцией $2m$ раз и т. д. — вплоть до появления ситуаций **д**), **е**). Ясно, что (по условиям утв. 9) рано или поздно такие ситуации появятся. Это будет означать, что некоторые (или все) ложные гипотезы не подтверждаются, т. е. будут опровергнуты. Действительно, если было, например, $\Gamma_j^a: P_A(G_L(t_i)) = P_A(t_i)$ и появилось $\Gamma_k^a: P_A(G_L(t_i)) \neq P_A(t_i)$, то Γ_j^a и Γ_k^a неверны (взаимно исключают друг друга). Если было $\Gamma_j^b: P_A(G_L(t_i)) = P_A(t_i)$ и появилось $\Gamma_k^b: P_A(G_L(t_i)) \neq P_A(t_i)$, то неверны Γ_j^b и Γ_k^b . В то же время все гипотезы вида Γ_j^c подтверждаются.

Появление ситуаций **д**) и **е**) возможно лишь при переходе S'_U между соседними ячейками a_k, a_{k+1} , содержащими разные по P_A конструкции x_k и x_{k+1} . Рассмотрим возможные продолжения. Предположим, первой возникла ситуация **д**):

$$P_A(G_L(t_i)) = P_A(t_{i+1}) \neq P_A(t_i).$$

Гипотезы **а**) и **д**) отбрасываются как ложные. S'_U проверяет остальные операции G_R, G_0 :

- ф**) $P_A(G_R(t_{i+1})) = P_A(t_{i+2}) \neq P_A(t_i)$; $\Gamma_{i+1}^e: P_A(G_R(t_i)) \neq P_A(t_i)$; Γ_j^b и Γ_{i+1}^e отбрасываются;
- г**) $P_A(G_0(t_{i+2})) = P_A(t_{i+2})$; Γ_{i+2}^c подтверждается.

Обратный порядок проверки (сначала G_0 , потом G_R) не меняет результата. Одновременно на основании описания экспериментов $i, i+1$ и, может быть, $i+2$, формируются новые гипотезы:

- $\Gamma_j^f: P_A(G_R(t_{i+1})) \rightarrow$ «возможны изменения»;
- $\Gamma_j^g: P_A(G_0(t_{i+2})) \rightarrow$ «невозможны изменения»;
- $\Gamma_j^h: P_A(G_L(t_i)) \rightarrow$ «возможны изменения»,
а также Γ_j^i :

$$P_A(G_R(G_L(t_i))) = P_A(t_{i+2}) = P_A(G_0(t_i)) \rightarrow \beta_i$$

или (при обратном порядке проверки G_0, G_R) Γ_j^k :

$$P_A(G_R(G_0(G_L(t_i)))) = P_A(t_{i+3}) = P_A(t_i) \rightarrow \beta_i.$$

Если первой возникла ситуация **е)**, то этот же алгоритм даёт те же результаты и те же гипотезы $\Gamma_j^f, \Gamma_j^g, \Gamma_j^h$, кроме двух последних, которые примут следующий вид:

- $\Gamma_j^m: P_A(G_L(G_R(t_i))) = P_A(t_{i+2}) = P_A(G_0(t_i)) \rightarrow \beta_i;$
- $\Gamma_j^n: P_A(G_L(G_0(G_R(t_i)))) = P_A(t_i) = P_A(G_0(t_i)) \rightarrow \beta_i.$

Заметим, что все гипотезы $\Gamma_j^f, \dots, \Gamma_j^n$ — истинные, в дальнейшем могут лишь подтверждаться, но не опровергаться. Одновременно гипотезы Γ_j^f, Γ_j^g дают направление поиска возможных других неоднородностей в содержимом массива ячеек хранения, так как позволяют исключить операцию G_0 из последующих динамических операций S'_U . Первые из этих операций заключаются в повторном сканировании перехода между ячейками a_k, a_{k+1} , содержащими неодинаковые по P_A конструкции. При этом всегда выбирается такая из G_L, G_R , которая «противоположна» последней из применявшихся динамических операций, относящихся к операциям с «возможными изменениями». Это гарантирует обязательное появление Γ_j^i или Γ_j^m , если они не появились раньше. Попутно может подтвердиться, что G_0 допускает свое исключение из гипотез Γ_j^k, Γ_j^n вследствие появления либо Γ_j^p :

$$P_A(G_L(G_0(G_R(t_i)))) = P_A(G_L(G_R(t_i))),$$

либо Γ_j^q :

$$P_A(G_R(G_0(G_L(t_i)))) = P_A(G_R(G_L(t_i)))$$

в зависимости от первоначальной ситуации **д)** или **е)** и порядка проверки G_0, G_L или G_R .

Значение («образ») $P_A(x_k)$ фиксируется в памяти S'_U как «начало отсчета», после чего в память с помощью любой из операций G_L или G_R копируются «образы» (по P_A) нескольких (например, n) последовательных ячеек хранения S'_U . С использованием «противоположной» динамической операции (G_R или G_L) проверяется при «противоположном» направлении движения содержимое по P_A всех n ячеек хранения, вплоть до ячейки a_k . Продолжая движение в том же направлении, но теперь уже на $2n$ ячеек, S'_U копирует в память «образы» (то есть значения по P_A) «противоположного» отрезка ячеек хранения, затем вновь «разворачивается» и вновь возвращается в ячейку a_k , проверяя верность записанных в памяти «образов» массива из $2n$ ячеек. В дальнейшем S'_U может постоянно расширять зону сканирования массива ячеек хранения, каждый раз увеличивая её, например, в 2 раза (или на фиксированное число ячеек). Особой ситуацией является достижение S'_U крайней левой ячейки a_0 . Для разрешения возникающей при этом проблемы (заданное число шагов влево еще не выполнено, а продолжать движение дальше влево нельзя) целесообразно использовать механизм типа «безусловного рефлекс», предусматривающий автоматический выбор «противоположной» динамической операции с одновременной пометкой ячейки памяти с кодом $P_A(x_0) = P_A(a_0)$ особым знаком типа «дальше нельзя». Введение этого механизма в устройство управления незначительно усложняет S'_U . Таким образом, по достижении a_0 в памяти S'_U гарантированно находятся «образы» соответствующих ячеек от a_0 до a_{k+1} включительно (а возможно, и дальше) в виде соответствующих значений операции $P_A(a_i)$, записанные в том порядке, в каком ячейки следуют друг за другом.

Теперь S'_U может заняться поиском пустых ячеек. Это можно сделать самыми разнообразными способами. Например, при выполнении различных экспериментов по поиску и верификации неопределённых технологических (и аналитических) операций можно сравнивать значение по P_A содержимого любой ячейки a_i до извлечения из неё какого-либо объекта x_i и после извлечения. Если значение по P_A в обоих случаях одинаково, то ячейка помечается как ячейка, имеющая одинаковые значение по P_A в обоих случаях, то есть будучи предположительно заполненной каким-то объектом и будучи предположительно пустой, что может в дальнейшем вызвать дополнительное исследование такого рода ситуации. Либо — с учётом рассмотренных выше возможностей S'_U извлекать из ЯХ и затем возвращать обратно без выполнения каких-либо технологических операций содержимое какой-либо ЯХ a_i — т.е. сначала выделяется «достаточно длинный» массив ЯХ, имеющих одинаковые значения по P_A . Пусть он начинается с ячейки j и продолжается до ячейки $j + r$. Затем ко всем этим ЯХ применяется гипотеза Γ_0^0 : содержимое ячеек $(a_j) = (a_{j+1}) = (a_{j+r}) = \square$. Гипотеза проверяется для каждой из этих ячеек a_{j+s} ($s \in \{0, \dots, r\}$) с помощью короткого алгоритма: $P_A(a_{j+s}) = M_F P_A(a_{j+s}) M_B$? Если равенство выполняется, то ячейка хранения может помечаться как предположительно пустая. Утверждение доказано. Отметим, что существуют и другие варианты доказательства этого утверждения. $\square$

Возможность получения в памяти S'_U адекватной модели фрагмента окружающего его (на данном этапе — линейного) мира позволяет перейти к этапу исследования этой частично неопределённой технологии $\mathbf{T}'$ на предмет её полноты и расшифровки сути некоторых неопределённых, но важных операций типа операций синтеза и декомпозиции. Для этого, как было сказано в условиях и во фрагменте доказательств утверждений 2–7, необходимы пустые ячейки хранения и ячейки с какими-либо неизвестными конструкциями, которые могут быть использованы в качестве $b_i \in \mathbf{B}$ в алгоритмах исследования возможностей частично неопределённых технологий в соответствии с упомянутыми утверждениями.

Из теории алгоритмов известно, что возможности (мощность) алгоритмических систем, реализуемых в м.Т. с одномерными, двухмерными (плоскими) и трёхмерными (объёмными) «лентами», одинаковы [23]. Этого же можно ожидать и для S'_U с двух- и трёхмерными массивами ЯХ. Ясно, что S'_U с трёхмерным массивом ячеек хранения очень сильно напоминает субъекта (или автомат), познающего окружающий его (или нас) трёхмерный мир с «встроенными» в него в различных местах различными физическими объектами и «пустым» (в смысле возможности перемещения в нём) пространством. Используя алгоритм перехода от трёхмерной «ленты» к двухмерной и далее — к одномерной (аналогичный, например, алгоритму, приведённому в [23]), нетрудно доказать следующее утверждение.

УТВЕРЖДЕНИЕ 10. *Если в S'_U с трёхмерным массивом ячеек хранения и частично неопределённой технологией*

$$\mathbf{T}' = \langle \mathbf{B}, \{F_1, \dots, P_A, (P_S), \dots, P_j, G_L^x, G_R^x, G_L^y, G_R^y, G_L^z, G_R^z, G_0\} \rangle$$

операция анализа $P_A(P_S)$ известна; имеются источники неизвестных конструкций $y_j \in B$, различные элементы базы имеют различные значения по предикату $P_A(P_S)$, а среди $F_1, \dots, P_j$ отсутствуют операции типа F_k и

имеются неопределенные операции синтеза, аналогичные F_S , выполняемые над конечным числом элементов, и конечное множество неопределенных, полных по декомпозиции стационарных операций, к группе неопределенных динамических операций перемещения на одну ячейку по каждой из трёх взаимно ортогональных координат X, Y и Z : $G_L^x, G_R^x, G_L^y, G_R^y, G_L^z, G_R^z, G_0$ имеется доступ, независимый от остальных операций, а трёхмерный массив ячеек хранения содержит непустые ячейки с конструкциями $b_i \in \mathbf{B}$, отличающимися по P_A (или P_S), и хотя бы один конечный непрерывный массив пустых ячеек, причем значение по P_A (или P_S) для пустых ячеек отличается от непустых, то существует реализуемый в S'_U алгоритм, обеспечивающий нахождение операций $G_L^x, G_R^x, G_L^y, G_R^y, G_L^z, G_R^z, G_0$, формирующий в памяти S'_U адекватную (изоморфную) модель начального фрагмента массива ячеек хранения, находит с вероятностью, сколь угодно близкой к единице, множество полных по декомпозиции операций, базу $\mathbf{A}$ (с четким разделением на различные элементы), операции синтеза, и «формулирует» гипотезы относительно сущности и характеристик остальных операций.

Существование такого алгоритма означает не что иное, как познаваемость соответствующего трёхмерного пространства с расположенными в нём физическими объектами (и фрагментами «пустого» пространства) с помощью автоматически действующего устройства, которое естественно назвать «познавателем».

Версии алгоритмов познания различных вариантов технологий для различных, свойственных в том числе и человеку, этапов и вариантов познания, рассмотрены выше. Согласно им, можно выделить следующие основные особенности всех таких алгоритмов:

- 1) на первом этапе познаватель просто «разбирается» со своими органами движения, выясняя, с помощью каких команд (операций) он может перемещаться в пространстве;
- 2) на втором этапе познаватель осваивает (с точки зрения «заполненности» различными объектами) близлежащее окружающее пространство, находя в нем «пустые» и непустые «ячейки»;
- 3) на третьем этапе — после формирования в памяти адекватной картины окружающего пространства — познаватель исследует возможности и особенности окружающей его технологии, «встроенной» в это пространство; согласно утверждениям 1–10, данный этап вполне алгоритмизуем; при этом достаточно эффективной основой многих алгоритмов познания оказывается метод «случайных проб и ошибок».

Нетрудно заметить, что аналогичные этапы освоения (познания — в контексте соответствующего уровня интеллекта) окружающего пространства при-сутствуют новорождённым практически всех высокоорганизованных животных, включая человека, что является дополнительным, пусть и косвенным, подтверждением корректности выбранного подхода.

ORCID

Сергей Михайлович Крылов: <http://orcid.org/0000-0002-7073-5569>

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с.

2. Новикова В. А., Андреева Е. Ю., Туйкина Д. К. *Искусственный интеллект и экспертные системы*, 2007, http://expro.ksu.ru/materials/ii_i_es/book.html.
3. Крылов С. М. *Формальная технология и эволюция*. Машиностроение-1: М., 2006. 384 с.
4. Крылов С. М. Математические основы научной метафизики // *Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки*, 2012. №1(26). С. 233–242. doi: [10.14498/vsgtu1043](https://doi.org/10.14498/vsgtu1043).
5. Крылов С. М. Перспективы метаматематических структур в науке // *Вестн. Сам. гос. техн. ун-та. Сер. Физ.-мат. науки*, 2013. №2(31). С. 101–110. doi: [10.14498/vsgtu1203](https://doi.org/10.14498/vsgtu1203).
6. Крылов С. М. *Метаматематические основы науки будущего*. Самара: СамГТУ, 2014. 247 с., <http://vt.samgtu.ru/images/pdf/krylov/metamath.pdf>.
7. Krylov S. M. Multifunctional Remote Laboratories for Real Experiments, Engineering Processes, and Manufacturing Methods // *International Journal of Online Engineering*, 2014. vol. 10, no. 5. pp. 29–30. doi: [10.3991/ijoe.v10i5.3726](https://doi.org/10.3991/ijoe.v10i5.3726).
8. Мальцев А. И. *Алгебраические системы*. М.: Наука, 1970. 392 с.
9. Hamming R. W. The Unreasonable Effectiveness of Mathematics // *The American Mathematical Monthly*, 1980. vol. 87, no. 2. pp. 81–90. doi: [10.2307/2321982](https://doi.org/10.2307/2321982).
10. *Алгоритмы в современной математике и ее приложениях*: Материалы международного симпозиума (Ургенч, Узбекистан, 16-22 сент. 1979 г.). Новосибирск: СО АН СССР, 1982. 364 с.
11. Tegmark M. *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality*. Random House Tower, New York: Knopf, 2014. 432 pp.
12. Успенский В. А. *Теорема Геделя о неполноте* / Популярные лекции по математике. М.: Наука, 1982. 112 с.
13. Мальцев А. И. *Алгоритмы и рекурсивные функции*. М.: Наука, 1965. 392 с.
14. Shoenfield J. R. *Degrees of Unsolvability* / North-Holland Mathematics Studies. vol. 2, 1971. viii+111 pp.
15. Manna Z. *Mathematical Theory of Computation* / McGraw-Hill Computer Science Series. New-York: McGraw-Hill Inc., 1974. x+448 pp.
16. Krylov S. M. Universal Programmable Completely Automated Factories-on-a-Chip / *Proc. of the COMS2004* (Aug. 29 – Sept. 2, 2004). Edmonton, Alberta, Canada, Washington: MANCEF, 2004. pp. 269–273.
17. Крылов С. М. Формальная технология и универсальные системы. I // *Кибернетика*, 1986. №4. С. 85–89.
18. Крылов С. М. Формальная технология и универсальные системы. II // *Кибернетика*, 1986. №5. С. 28–31.
19. Krylov S. M. Formal Technology and Cognitive Processes // *International Journal of General Systems*, 1996. vol. 24, no. 3. pp. 233–243. doi: [10.1080/03081079608945119](https://doi.org/10.1080/03081079608945119).
20. Shinichi Tamura, Kokichi Tanaka Note on Analog Memory Automata // *Information Sciences*, 1974. vol. 7. pp. 73–80. doi: [10.1016/0020-0255\(74\)90006-1](https://doi.org/10.1016/0020-0255(74)90006-1).
21. Крылов С. М. Модели универсальных дискретно-аналоговых машин на основе машины Тьюринга // *Электронное моделирование*, 1982. №3. С. 6–10.
22. Крылов С. М. Доказательство ограниченности действия тезиса Тьюринга–Черча на объектах с физическими свойствами // *Вестн. Оренбург. госуд. ун-та*, 2003. №3. С. 102–105, http://vestnik.osu.ru/2003_3/19.pdf.
23. Мартыненко Б. К. *Языки и трансляции*. СПб.: СПб. ун-т, 2013. 265 с.

Поступила в редакцию 20/IV/2015;
в окончательном варианте — 30/VI/2015;
принята в печать — 08/VIII/2015.

MSC: 03F50, 68T30

MATHEMATICAL FOUNDATIONS OF EPISTEMOLOGY BASED ON EXPERIMENTS

S. M. Krylov

Samara State Technical University,
 244, Molodogvardeyskaya st., Samara, 443100, Russian Federation.

Abstract

The paper deals with basic prerequisites for the development of epistemology, which uses information concerning real experiments in the real world (with real objects). Such experiments are conducted by “formal-technological” analogs of Turing Machines. These analogs are called “universal synthesizers-analyzers”. They can perform syntheses and analyses of various objects or constructions (obtained by conjunctions of finite number of smaller objects called basic elements) with the help of various algorithmic systems having some restrictions. Such algorithmic systems are called Formal Technologies. They have formal structures that are very similar to the formal structure of Maltsev’s algebraic systems. This formal closeness allows us, first, to set up a hypothesis concerning algorithmic basis of almost all surrounding physical processes, as understandable as well as till non-understandable ones, that partially explains the wide applicability of mathematics to the outer world; second, this closeness allows one to formulate and prove some theorems (called assertions) concerning features and peculiar properties of cognitive algorithms in one-, two- or three-dimensional surroundings for various formal technological systems, including a so called “acquired knowledge effectiveness theorem”. The theorem (assertion) can be applied to a very wide class of formal technologies which use an equality predicate for objects analyses. In the paper various cognitive algorithms are listed and proved. These algorithms have different sets of technological operations resembling syntheses and decompositions, as well as different sets of analytical operations including equality predicates, “random stationary mapping” operations (which use unknown algorithms to obtain stationary results, therefore these operations are very similar to oracles in Turing Machines), operations that define object shapes, and so on. The structure of automatic cognitive devices called “cognizers” is described.

Keywords: cognitive algorithms, physical processes with physical objects, General Formal Technology, General System Theory, object properties, object functionalities, universal object synthesizer-analyzer, universal cognizer.

© 2015 Samara State Technical University.

Please cite this article in press as:

Krylov S. M. Mathematical foundations of epistemology based on experiments, *Vestn. Samar. Gos. Tekhn. Univ., Ser. Fiz.-Mat. Nauki* [J. Samara State Tech. Univ., Ser. Phys. & Math. Sci.], 2015, vol. 19, no. 3, pp. 534–558. doi: [10.14498/vsgtu1427](http://dx.doi.org/10.14498/vsgtu1427). (In Russian)

Author Details:

Sergey M. Krylov (Dr. Techn. Sci; s_m_krylov@mail.ru), Professor, Dept. of Computer Science.

doi: <http://dx.doi.org/10.14498/vsgtu1427>

ORCID

Sergey M. Krylov: <http://orcid.org/0000-0002-7073-5569>

REFERENCES

1. Gavrilova T. A., Khoroshevskii V. F. *Bazy znaniy intellektual'nykh sistem* [Knowledge Base intelligent systems]. St. Petersburg, Piter, 2000, 384 pp. (In Russian)
2. Novikova V. A., Andreeva E. Iu., Tuikina D. K. *Iskusstvennyi intellekt i ekspertnye sistemy* [Artificial intelligence and expert systems], 2007 (In Russian), http://expro.ksu.ru/materials/ii_i_es/book.html.
3. Krylov S. M. *Formal'naiia tekhnologiia i evoliutsiia* [Formal technology and evolution]. Mashinostroenie-1, Moscow, 2006, 384 pp. (In Russian)
4. Krylov S. M. Mathematical foundations of metaphysics, *Vestn. Samar. Gos. Tekhn. Univ. Ser. Fiz.-Mat. Nauki* [J. Samara State Tech. Univ., Ser. Phys. & Math. Sci.], 2012, no. 1(26), pp. 233–242 (In Russian). doi: [10.14498/vsgtu1043](https://doi.org/10.14498/vsgtu1043).
5. Krylov S. M. Prospects of metamathematical structures in science, *Vestn. Samar. Gos. Tekhn. Univ. Ser. Fiz.-Mat. Nauki* [J. Samara State Tech. Univ., Ser. Phys. & Math. Sci.], 2013, no. 2(31), pp. 101–110 (In Russian). doi: [10.14498/vsgtu1203](https://doi.org/10.14498/vsgtu1203).
6. Krylov S. M. *Metamatematicheskie osnovy nauki budushchego* [Metamathematical foundations of the science of the future]. Samara, Samara State Technical Univ., 2014, 247 pp. (In Russian), <http://vt.samgtu.ru/images/pdf/krylov/metamath.pdf>.
7. Krylov S. M. Multifunctional Remote Laboratories for Real Experiments, Engineering Processes, and Manufacturing Methods, *International Journal of Online Engineering*, 2014, vol. 10, no. 5, pp. 29–30. doi: [10.3991/ijoe.v10i5.3726](https://doi.org/10.3991/ijoe.v10i5.3726).
8. Maltsev A. I. *Algebraic Systems*. Berlin, Springer-Verlag, 1973, xii+317 pp.
9. Hamming R. W. The Unreasonable Effectiveness of Mathematics, *The American Mathematical Monthly*, 1980, vol. 87, no. 2, pp. 81–90. doi: [10.2307/2321982](https://doi.org/10.2307/2321982).
10. *Algoritmy v sovremennoi matematike i ee prilozheniakh* [Algorithms in Modern Mathematics and Computer Science], Proceedings of the International Symposium. Novosibirsk, 1982, 364 pp. (In Russian)
11. Tegmark M. *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality*. Random House Tower, New York, Knopf, 2014, 432 pp.
12. Uspensky V. A. *Gödel's Incompleteness Theorem*, Little Mathematics Library. Moscow, Mir Publ., 1987, 104 pp.
13. Maltsev A. I. *Algorithms and recursive functions*. Groningen, Wolters-Noordhoff Publ., 1970, 372 pp.
14. Shoenfield J. R. *Degrees of Unsolvability*, North-Holland Mathematics Studies, vol. 2, 1971, viii+111 pp.
15. Manna Z. *Mathematical Theory of Computation*, McGraw-Hill Computer Science Series. New-York, McGraw-Hill Inc., 1974, x+448 pp.
16. Krylov S. M. Universal Programmable Completely Automated Factories-on-a-Chip, *Proc. of the COMS2004* (Aug. 29 – Sept. 2, 2004). Edmonton, Alberta, Canada, Washington, MANCEF, 2004, pp. 269–273.
17. Krylov S. M. Formal technology and universal systems. I, *Cybernetics*, 1986, vol. 22, no. 4, pp. 512–518. doi: [10.1007/bf01075083](https://doi.org/10.1007/bf01075083).
18. Krylov S. M. Formal technology and universal systems. II, *Cybernetics*, 1986, vol. 22, no. 5, pp. 567–572. doi: [10.1007/bf01068352](https://doi.org/10.1007/bf01068352).
19. Krylov S. M. Formal Technology and Cognitive Processes, *International Journal of General Systems*, 1996, vol. 24, no. 3, pp. 233–243. doi: [10.1080/03081079608945119](https://doi.org/10.1080/03081079608945119).
20. Shinichi Tamura, Kokichi Tanaka Note on Analog Memory Automata, *Information Sciences*, 1974, vol. 7, pp. 73–80. doi: [10.1016/0020-0255\(74\)90006-1](https://doi.org/10.1016/0020-0255(74)90006-1).

21. Krylov S. M. Models of universal discrete-analog machines based on Turing machine, *Elektronnoye modelirovaniye*, 1982, no. 3, pp. 6–10 (In Russian).
22. Krylov S. M. The proof of action boundedness for Turing–Church thesis on objects with physical properties, *Vestn. Orenburg. Gos. Univ.*, 2003, no. 3, pp. 102–105 (In Russian), http://vestnik.osu.ru/2003_3/19.pdf.
23. Martynenko B. K. *Iazyki i translyatsii* [Languages and Translation]. St. Petersburg, St. Petersburg State Univ., 2013, 265 pp. (in Russian)

Received 20/IV/2015;
received in revised form 30/VI/2015;
accepted 08/VIII/2015.