DOI: 10.51790/2712-9942-2021-2-4-11

В-КОМПЬЮТЕРЫ: ГЕНЕЗИС, РАЗВИТИЕ, ПРИЛОЖЕНИЯ

Г. Е. Деев, С. В. Ермаков
Обнинский институт атомной энергетики, Национальный исследовательский ядерный университет «МИФИ», г. Обнинск, Российская Федерация,
georgdeo@mail.ru, ermakov@iate.obninsk.ru

Аннотация: показан генезис В-компьютера как результата решения проблемы обозначения для чисел. Первым В-компьютером, появляющимся при решении этой проблемы, является автомат сдвига. Далее в результате применения метода последовательного синтеза появляются все остальные В-компьютеры. Показано, что В-компьютеры обладают неограниченным потенциалом развития. Принципиальных ограничений такого развития не существует. Это приводит к перманентному обогащению вычислительных возможностей В-компьютера. На основании этого факта делается вывод о том, что в полной мере решение таких задач, как создание суперкомпьютера, создание искусственного интеллекта может быть осуществлено на базе В-компьютеров.

Ключевые слова: В-компьютер, мыслящее существо, искусственный интеллект, шаблон заготовки, абстрактное вычислительное устройство, В-схема.

Благодарности: работа выполнена при поддержке гранта РФФИ 20-07-00862.

Для цитирования: Деев Г. Е., Ермаков С. В. В-компьютеры: генезис, развитие, приложения. *Успехи кибернетики*. 2021;2(4):90–106. DOI: 10.51790/2712-9942-2021-2-4-11.

B-COMPUTERS: GENESIS, EVOLUTION, AND APPLICATIONS

G. E. Deev, S. V. Ermakov
Obninsk Institute for Nuclear Power Engineering, National Nuclear Research University MEPhI, Obninsk, Russian Federation,
georgdeo@mail.ru, ermakov@iate.obninsk.ru

Abstract: this study presents the B-computer genesis arising from the number designation problem. The first B-computer that emerges as we solve this problem is shift automation. Other B-computers are generated through sequential synthesis. It is shown that the B-computers have an unlimited potential for evolution. There are no basic restrictions to it. The B-computer capabilities keep expanding. With this fact, we concluded that B-computers are suitable for building a supercomputer and creating artificial intelligence.

Keywords: B-computer, thinking creature, artificial intelligence, template, abstract computer, B-layout.

Acknowledgements: this work was supported by RFBR Grant 20-07-00862.

Cite this article: Deev G. E., Ermakov S. V. B-Computers: Genesis, Evolution, and Applications. *Russian Journal of Cybernetics*. 2021;2(4):90–106. DOI: 10.51790/2712-9942-2021-2-4-11.

Предварительное пояснение

В-компьютеры — это объекты, собираемые из абстрактных вычислительных устройств (АВУ) (использованная в названии «В-компьютеры» буква «В» латинская). В свою очередь, АВУ — это обобщение понятия абстрактного автомата, распространенное на все возможные устройства, вычисляющие математические функции. Одним из важнейших требований, предъявляемых к АВУ, является требование их минимальности, сводящееся, в случае автоматов, к минимальности количества состояний; в случае общего АВУ — к требованию, родственному требованию минимальности состояний. При выполнении условия минимальности получается новое устройство, не помнящее своего происхождения и с этого момента функционирующее как самостоятельный независимый объект. Часто используемым АВУ могут даваться имена, отражающие их функциональное назначение. Например, «автомат сложения», «умножитель на три» и т.д. АВУ задаются таблицами, формулами, В-схемами и в «железе». Наличие материальных объектов, представляющих АВУ, позволяет вести их полноценное исследование. АВУ — это объект, с необходимостью возникающий при решении принципиальных и прикладных задач. К изначальным принципиальным задачам относятся задачи, связанные с понятием числа, и задачи, связанные с решением важнейшей *проблемы обозначений для чисел*. Фундаментальное и глубинное *понятие* числа само по себе никак не используется в приложениях и, в частности, в вычислениях до тех пор, *пока для него не будет разработано обозначение*. Без обозначений число оказывается ускользающим от нас, его невозможно применять на практике. Без обозначений невозможно существование никаких вычислительных устройств. Лишь тогда, когда число получает обозначение, лишь с этого момента оно оказывается включенным в практическую и научную деятельность. Проблема обозначений для чисел является важнейшей как с теоретической, так и с прикладной точек зрения. Ее решение естественным образом, с необходимостью, приводит к появлению первого вычислительного устройства — *автомата сдвига.* В свою очередь, появление автомата сдвига приводит к порождению всех остальных вычислительных устройств. Таким образом, автомат сдвига является прародителем всех вычислительных устройств, используемых в настоящее время. Помимо автомата сдвига в каждой системе счисления существуют другие автоматы, которые могут быть взяты в качестве первичных автоматов. Каждый из них порождает свое дерево вычислительных устройств, отличных от тех, которые порождает автомат сдвига, которые, однако, в настоящее время не находят себе применения. Возможно, это наш скрытый потенциал при решении проблем будущего. Основными инструментами, служащими для построения вторичных АВУ по отношению к автомату сдвига, являются *метод последовательного синтеза* и его развитие — *метод сетевого синтеза*. Эти методы на языке устройств реализуют основные инструменты построения функций — *суперпозицию и обобщенную суперпозицию.* Применение этих методов принципиально может продолжаться до бесконечности, однако на практике возникает непреодолимое препятствие, связанное со *сложностью вычислений.* Никакие вычислительные машины не могут помочь преодолеть это препятствие. Судя по всему, это означает наличие предела нашим знаниям, основанным на понятии числа. Как известно, Д. Гильберт в свое время сказал знаменитую красивую фразу «Wir müßen wissen und wir werden wissen» («Мы должны знать и мы будем знать»). Однако наличие предела, связанного со сложностью вычислений, заставляет нас в этом усомниться. Наши знания могут постоянно возрастать, но они никогда не преодолеют некую невидимую границу. Исследуя АВУ и их вычислительные возможности, мы узнаем немало нового о природе чисел и о природе вычислений.

Для практической реализации В-компьютеров нет принципиальных препятствий. Они могут быть созданы в любой технологии и для любой системы счисления, а не только для двоичной. Производства, создающие В-компьютеры, должны быть проще, чем аналогичные производства современных компьютеров.

Проблема обозначений для натуральных чисел

Вычисления проводятся не над числами, а над их представителями — обозначениями для чисел. Этим обусловлена важность решения проблемы обозначений для чисел. При этом первое вычислительное устройство — автомат сдвига — появляется естественным образом в результате решения проблемы обозначений для натуральных чисел.

Насколько велик произвол в создании обозначений для чисел? Отвечая на этот вопрос, надо учитывать, что обозначения для чисел создает человек, создает для себя, используя свои ограниченные возможности. Учет ограниченности этих возможностей необходим, так как он дает представление об ареале доступного человеку, но вместе с тем он показывает, что ограниченность эта не чрезмерна и что такое мыслящее существо, как человек, может решить проблему обозначений для чисел. Однако могло бы быть и так, что ограничения настолько велики, что человек не смог бы решить проблему обозначений для чисел и такой инструмент научных исследований, как вычислительное устройство, был бы ему недоступен. Понятие числа оказалось бы для него бесполезным. Кроме того, выяснилось, что ограничения мыслительных возможностей человека таковы, что они с необходимостью приводят к позиционной системе счисления.

Проследим за созданием какой-либо конкретной системы обозначений.

Пусть процесс порождения чисел запущен. Этот процесс носит индуктивный характер и выражается записью: $n \to A(n)$, где $A(n)$ — мыслительное действие, состоящее в порождении непосредственно следующего числа по отношению к числу n (Пеано для $A(n)$ использует термин *последователь*).

Пусть
число «ноль» обозначено символом: 0,
число «один» обозначено символом: 1,
число «два» обозначено символом: 2,
число «три» обозначено символом: 3.

В этом процессе на каждом шаге происходит изобретение имени и символа для каждого вновь порожденного числа. При этом число как *количественная идея* первично, а имя и символ вторичны. Как соотносится этот процесс с возможностями человека? Этот процесс требует: 1) ***изобретения*** и ***запоминания имени*** вновь порожденной количественной идеи, т.е имени числа; 2) ***изобретения*** и ***запоминания символа*** для каждого числа; 3) чтобы в результате изобретения порождались новые имена и символы, не встречавшиеся на предыдущих шагах; для этого нужно каждый символ, который предлагается изобретательской мыслью, ***сравнивать*** на предмет отличия его от всех ранее порожденных символов. Но чисел много, количество их стремится к бесконечности, поэтому процесс сравнения становится невообразимо большим. Все сказанное требует (1) бесконечной памяти, (2) сколь угодно большой скорости передачи сигналов (это необходимо, чтобы осуществить процедуру сравнения друг с другом сколь угодно большого количества объектов за конечное время). Эти требования не соответствуют ни возможностям человека (у него конечная память), ни возможностям природы (в природе, насколько известно, скорость передачи сигналов ограничена). Поэтому *описанный процесс изобретения индивидуальных обозначений для каждого вновь порожденного натурального числа должен быть по необходимости* ***прерван*** *на некотором шаге*. Мы можем изобрести лишь конечное число различных знаков для нескольких первых натуральных чисел. Назовем это множество знаков *множеством цифр*. Для остальных чисел описанный процесс индивидуального изобретательства знаков должен быть прерван. На каком шаге будет прерван процесс изобретательства знаков, принципиального значения не имеет. Все зависит от объема памяти мыслящего существа. У разных существ она может быть различной, но в любом случае конечной.

На этом этапе мы фиксируем следующие два момента:

1. *Процесс изобретательства обозначений и запоминания их должен быть остановлен после проведения нескольких начальных шагов, причем количество начальных шагов принципиального значения не имеет*. (Произвол в **количестве** первых изобретенных знаков. В этот момент рождается многообразие будущих систем счисления.)

2. *Значки для первых чисел могут изобретаться произвольно, при условии, что они различны*. (Здесь имеется ограниченный произвол в **начертании** знаков.)

Итак, на начальном этапе процесса изобретательства обозначений для чисел возможен произвол в количестве первых изобретенных знаков и в их начертании. Мы остановимся на приведенном выше шаге: самое большое число, для которого мы изобрели индивидуальный знак, – это число три.

Далее нужно решить проблему обозначений всех чисел, порождаемых вслед за числом три. Индивидуальное изобретательство знаков, проводимое на начальном этапе, в общем-то, простой по замыслу и тривиальный по видимости (но не по сути), процесс: *придумывай и запоминай*. Но если на «придумывание» нет принципиальных ограничений, то на запоминание ограничения есть. Ограниченность памяти является тем препятствием, которое стоит на пути присвоения числам представляющих их знаков. Раз так, то делаем необходимый вывод: нужно учитывать ограниченность памяти в дальнейшем процессе присвоения числам знаков. Ограниченность участия памяти имеет место как на начальной фазе процесса присвоения знаков числам, так и в дальнейшем. На начальном этапе изобретательство и запоминание **необходимы**. На последующем этапе процесса порождения обозначений для чисел память понадобится по необходимости, в ограниченном объеме и не в таком, как раньше, «тривиальном» виде. Если раньше требовалась «тривиальная» память (т.е. появился знак — запоминай), то на новой фазе порождения обозначений запоминанию должно подвергаться нечто иное: с одной стороны, что-то компактное, обязательно конечное, а с другой стороны, эквивалентное в некотором смысле бесконечности, заменяющее бесконечность. Такое возможно, и оно соответствует метасистемному переходу. Нам известны случаи, когда бесконечные структуры оказывались эквивалентными конечным структурам. Процесс порождения обозначений должен быть передан чему-то, напоминающему механизм. Этот конечный механизм должен быть *создан* в результате *мыслительной* деятельности. Таким образом, окажется, что работа мысли заменит бесконечную память. Отсюда можно прийти к идее: *мысль нам дана для того, чтобы возместить отсутствие бесконечной памяти. Таким образом, мы* *мыслим, в частности, потому, что не обладаем бесконечной памятью*. (Могут быть и другие причины, приводящие нас к необходимости мыслить, в том числе и при наличии бесконечной памяти.)

Процесс создания обозначений должен быть алгоритмизован.

После этих размышлений вернемся к уже созданным обозначениям: 0, 1, 2, 3 и продолжим процесс создания обозначений для чисел. Согласно процедуре $n \to A(n)$ в мыслительном аппарате мыслящего существа (МАМС) рождается новая количественная идея (число) $A(3)$. Таким образом, число, следующее за числом 3, *как идея*, порождено. Для него требуется создать обозначение. Новые простые знаки типа 0, 1, 2, 3 изобретать нельзя. Следовательно, новый знак должен быть сложным и представлять собой комбинацию из уже изобретенных простых знаков. Так как мы владеем понятиями чисел 2, 3, то мы можем составлять комбинации из двух или трех знаков. (Если бы мы не владели понятиями чисел 2, 3, то составление комбинаций из двух и трех знаков было бы невозможно. Так достигнутая стадия развития обеспечивает выполнимость следующих шагов.) Это будут комбинации типа ba, cba.

Начнем с двухзначных комбинаций ba.

Вопрос: что ставить на место a и что ставить на место b? Чтобы определиться с ответом, надо сформулировать требования, которые должны выполняться при такой простановке.

Требование 1: комбинация ba должна быть новым знаком по сравнению с уже существующими.

Но любая простановка знаков даст *новый* знак. Например, $ba = 32$. Как выбрать среди возможных комбинаций знаков ту, которую нужно взять в качестве обозначения для вновь порожденной количественной идеи? Пока нет никаких предпочтений. Поэтому примем произвольно написанную комбинацию 32 как одну из равновозможных в качестве обозначения для появившегося числа. Таким образом, у нас появились числа, обозначаемые символами: 0, 1, 2, 3, 32. Количественной идее (числу), изображаемой сложным знаком 32, мы даем имя «четыре» (мы даем знакомое имя для облегчения ориентировки; можно было бы дать любое другое имя). С появлением нового числа процесс порождения чисел $n \to A(n)$ может быть продолжен и мы получаем новое число $A(32)$, которое также изображаем комбинацией типа ba, удовлетворяющей требованию 1. Например, изобразим новое число символом 12, так что будем иметь: $A(32)=12$. И т.д. Когда исчерпаем возможности двухсимвольных обозначений, перейдем к трехсимвольным, потом четырехсимвольным (этот термин после появления числа 32, «четыре», становится осмысленным) и т.д. Новые обозначения будут появляться. Но этот вариант введения обозначений предъявляет чрезмерные требования к памяти. Действительно, в этом способе обозначений нет, по существу, никакой системы (точнее, имеется минимальная), ибо происходит случайная простановка символов 0, 1, 2, 3 вместо букв $a, b, \ldots$. В итоге каждую новую комбинацию символов приходится *запоминать*, что при растущем неограниченно количестве чисел невозможно. Поэтому мы должны сформулировать

требование 2: простановка символов 0, 1, 2, 3 вместо букв $a, b, \ldots$ в комбинациях типа ba, cba, $\ldots$ не должна быть случайной, ибо случайная простановка символов требует бесконечной памяти.

Следствие из требований 1 и 2 состоит в том, что *система обозначений для чисел, следующих за уже порожденными числами 0, 1, 2, 3, должна описываться конечным образом (тогда память справится), но так, чтобы она могла быть применена сколь угодно много раз в согласии с бесконечной процедурой $n \to A(n)$ (тогда обозначениями потенциально будут охвачены все натуральные числа).*

Такую систему обозначений будем называть *регулярной*.

Регулярная система обозначений должна содержать в себе некоторое *конечное описание, которое можно повторить неограниченное число раз*. Костяком такого конечного описания должен быть конечный порядок. Этот конечный порядок мы должны извлечь из уже имеющегося у нас материала. Есть ли у нас такой порядок? Есть. Это *естественный* порядок порождения первых чисел:

$$0 \prec 1 \prec 2 \prec 3. \tag{1}$$

Этот порядок лежит в основе понятия отношения непосредственного следования, являясь его «затравочной» частью. Но он не определен для элементов 0 и 3. У элемента 0 нет непосредственно предшествующего элемента, а у элемента 3 нет непосредственно следующего. Мы доопределяем описанный порядок также естественным образом, *естественным потому, что он единственно возможен*:

$$0 \prec 1 \prec 2 \prec 3 \prec 0. \tag{2}$$

Единственность его следует из того, что простановка за элементом 3 любого другого числа (1 или 2) приводит к противоречию, к нарушению хотя бы одного из свойств, характеризующих отношение непосредственного следования.

Выясним, возможно ли его использование при решении проблемы обозначений.

Прежде всего, отметим, что с помощью двух знаков *ba* невозможно изобразить все множество натуральных чисел, потому что с помощью двух знаков можно изобразить лишь конечное множество чисел. Это можно обнаружить экспериментально. Точно так же невозможно изобразить все числа с помощью трех, четырех и т.д. знаков. *Следовательно, для изображения множества всех натуральных чисел требуются комбинации из бесконечного числа простых знаков*. Отметим, что каждый такой знак будет занимать определенное *место (позицию)* в пространстве, что два разных знака не могут находиться на одном и том же месте, что должна быть *система прочтения знаков* в зависимости от мест (т.е. опять должен быть *порядок,* но на этот раз во множестве мест).

Таким образом, для представления *всех* натуральных чисел требуется использование бесконечно местных записей.

Кроме того, как уже выяснено ранее, на места в этих записях ставить цифры случайным образом нельзя. Их нужно ставить каким-то регулярным способом.

Множество мест, на которых будут стоять цифры, изобразим следующим образом

$$\ldots\_\ldots\_\,\_\,\_\,\_\,. \tag{3}$$

Это один из простейших и удобных способов расположения мест в пространстве. Возможны и другие способы, например, по вертикали (как у инков) или логарифмической спирали и т.д., важно, чтобы была система прочтения знаков в зависимости от мест. Каждая черточка «–» символизирует собою место для цифры. Первое многоточие, встречающееся в (3) при движении слева направо, означает, что слева располагается бесконечное количество мест, которые мы не изображаем ввиду невозможности этого сделать. Второе многоточие между черточками означает, что здесь должно стоять некоторое конечное количество мест, которое мы также не изображаем, но по другой причине, а именно, по той, что количество этих мест при общем рассмотрении вопроса не определено. Места упорядочиваются *естественным* образом справа налево: действительно, учет происходящего на местах справа налево нам доступен даже при нашей конечной памяти. Происходящее же слева, в бесконечности, нам недоступно. Поэтому мы и говорим о *естественной упорядоченности мест справа налево*.

Итак, любое число должно быть представлено бесконечно местной записью.

После порождения числа 0 и обозначения его цифрой 0, мы цифру 0 ставим на все места справа налево вплоть до бесконечности. Будем считать, что именно эта знаковая строка представляет число 0:

$$\ldots\underline{0}\ldots\underline{000}. \tag{4}$$

Таким образом, количественная идея $A\{\ \}$, сформировавшаяся при обозрении пустого множества, имеет два обозначения:

$$A\{\ \} = 0 \text{ и } A\{\ \} = \ldots\underline{0}\ldots\underline{000}. \tag{5}$$

В этом месте мы отступаем от требования взаимной однозначности между объектом и обозначением для него. Как видно, *это отступление необходимо и обусловлено конечностью памяти мыслящего существа (МС)*. При бесконечной памяти МС требование взаимной однозначности между идеей и знаком могло бы быть реализовано «без труда». Конечность памяти МС заставляет нас отступить от требования взаимной однозначности между объектом и обозначением для него. Это обстоятельство с практической точки зрения может оказаться удобным, поскольку a'priori понятно, что в одних случаях может оказаться целесообразным использовать одно обозначение, а в других — другое. В частности, уже сейчас мы воспользуемся этой возможностью. Дело в том, что введенные нами новые объекты — места — целесообразно отличать друг от друга не только по их пространственному расположению, но и по именам. Это можно сделать с помощью порождаемых обозначений для порождаемых чисел. В частности, назовем самое правое в (5) место словосочетанием «место ноль» и, воспользовавшись появившимся удобством, обозначим его простейшим из двух обозначением для нуля:

$$\cdots\ \overline{0}\,. \tag{6}$$

На место с номером 0 оператор порождения чисел A может ставить символы 0, 1, 2, 3, а остальные места при этом заполняются символом 0. В результате простановки символа 1 получится запись:

$$\ldots\underline{0}\ldots\underline{0}\,\underset{0}{\underline{1}}, \tag{7}$$

что означает

$$(\ldots 0\ldots 00) = \ldots\underline{0}\ldots\underline{0}\,\underset{0}{\underline{1}}\,. \tag{8}$$

Это еще один вариант записи равенства: $A\{0\} = 1$.

Введем упрощенное обозначение для результата действия оператора A на множество: $A\{0,\ldots,n\}\overset{\geq 1}{=}A(n)$. Таким образом, мы имеем три варианта записи того, что последователь A вслед за числом 0 порождает число 1:

$$\{0\} = 1, \qquad A(0) = 1, \qquad A(\ldots\underline{0}\ldots\underline{0}\,\underset{0}{\underline{0}}) = \ldots\underline{0}\ldots\underline{0}\,\underset{0}{\underline{1}}\,. \tag{9}$$

Число 1, появившееся вслед за числом 0, используется в упрощенном варианте, как и число 0, для нумерации мест:

$$\ldots\underline{\ }\ldots\underset{1}{\underline{\ }}\,\underset{0}{\underline{\ }}\,, \tag{10}$$

так что помимо «места ноль» у нас появилось «место один». (В дальнейшем каждое вновь порожденное число нумерует соответствующее место.)
Третий вариант в (9), с участием мест, может быть записан так:

$$A(\ldots\underline{0}\ldots\underline{0}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{0}}) = \ldots\underline{0}\ldots\underline{0}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{1}}\,. \tag{11}$$

Понятно, что это чересчур подробная запись, и потому мы будем использовать ее лишь на начальной стадии решения проблемы обозначений для чисел, а также в случае необходимости. В основном же будут использоваться записи типа $A(0) = 1$.

Далее генератор чисел, т.е. последователь A, породит еще два числа: 2 и 3 согласно равенствам: $A(1) = 2$, $A(2) = 3$. Порожденные числа будут использоваться для нумерации мест, и для них равенства типа (11) будут иметь вид:

$$(\ldots\underline{0}\ldots\underline{0}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{1}}) = \ldots\underline{0}\ldots\underline{0}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{2}}\,. \tag{12}$$

$$(\ldots\underline{0}\ldots\underset{2}{\underline{0}}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{2}}) = \ldots\underline{0}\ldots\underset{2}{\underline{0}}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{3}}\,. \tag{13}$$

Число 2, появившееся в (12), используется в (13) для нумерации мест (появилось «место два»); аналогично используются для нумерации мест все появляющиеся в последующем числа. Их мы писать не будем, но сейчас обратим внимание на то, что с помощью двух мест: «места ноль» и «места один» — можно изобразить 16 чисел, которыми можно пронумеровать 16 мест; а с помощью трех мест: «места ноль», «места один», «места два» — можно изобразить 64 числа и ими пронумеровать 64 места и т.д. Другими словами, два процесса: *процесс порождения чисел* и *процесс нумерации мест* непротиворечивы. В том смысле, что не может случиться так, чтобы для нумерации мест не хватило уже порожденных чисел.

Что писать вслед за соотношением (13)? Согласно (2), вместо цифры 3, стоящей на месте 0, надо ставить 0. Если при этом не произвести никаких изменений на всех остальных местах, то мы вернемся к уже встречавшемуся обозначению для числа 0, что неправильно. Следовательно, изменения надо произвести на других местах. Мы уже ранее сказали, что места естественным образом упорядочены справа налево (это вытекает из конечности памяти МС). Поэтому следующим по порядку местом для включения его в работу должно быть «место один». Согласно (2), на месте с номером 0 нужно ставить цифру 0, а на месте с номером 1 вместо цифры 0 надо поставить цифру 1. Появившееся изображение для числа будет новым по сравнению со всеми ранее встречавшимися и в то же время согласованным с формирующимся регулярным процессом обозначения чисел. Получим:

$$(A(\ldots\underline{0}\ldots\underset{2}{\underline{0}}\,\underset{1}{\underline{0}}\,\underset{0}{\underline{3}}) = \ldots\underline{0}\ldots\underset{2}{\underline{0}}\,\underset{1}{\underline{1}}\,\underset{0}{\underline{0}}\,. \qquad (14)$$

В общем, понятно: на «месте ноль» надо последовательно согласно порядку, определяемому последовательностью (2), менять цифры. Если на «месте ноль» происходит переход от цифры 3 к цифре 0, то на «месте один», также согласно порядку (2), нужно совершить переход к следующей цифре. Если при этом на «месте один» происходит переход от цифры 3 к цифре 0, то на «месте два», также согласно порядку (2), нужно совершить переход к следующей цифре. Если при этом на «месте два» ... и т.д.

Повторяющиеся фразы являются не чем иным, как упомянутым ранее *конечным описанием процедуры*, которая должна применяться *бесконечное число раз*. Той самой процедуры, о которой шла речь ранее.

Эта процедура возникла с *необходимостью* в результате приведенных рассуждений, каждый шаг которых обоснован и вытекает из предыдущих. Теперь уместно отметить, что описанная процедура никак не касается порождения чисел как количественных идей, а работает только на множестве многоместных *обозначений* для чисел. Она работает не в ареале идей, а в ареале обозначений. Поэтому она может быть отделена от оператора A, имеющего идейный характер, и объявлена оператором, действующим на множестве обозначений и порождающим из каждого существующего обозначения для натурального числа обозначение для следующего числа.

Сформулируем это в виде *вывода (для выбранного множества цифр)*:
для оператора-последователя A(n), единственным образом порождающего число как идею, существует и соответствует ему единственный оператор-последователь S(n), единственным образом порождающий многоместное обозначение для числа.

Результатом действия оператора-последователя $S(n)$ является выписанный ниже столбец обозначений для нескольких первых натуральных чисел. Столбцовая запись выбрана для усиления зрительного впечатления.

$$\ldots 0000;$$
$$\ldots 0001;$$
$$\ldots 0002;$$
$$\ldots 0003;$$
$$\ldots 0010;$$
$$\ldots 0011;$$
$$\ldots 0012;$$
$$\ldots\ldots\ldots$$

В связи с многоместными обозначениями для чисел установилась терминология для мест, занимаемых цифрами (см. рис. 1):

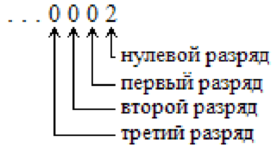

Рис. 1. *Разрядная сетка*

Позиции, занимаемые цифрами, называются *разрядами*, а совокупность разрядов — *разрядной сеткой*. Это одна из возможных реализаций упомянутого выше требования, что *для изображения натуральных чисел требуется бесконечно местная разрядная сетка*.

Таким образом, на поставленный в начале параграфа вопрос можно ответить так: если учесть ограничения, предъявляемые к мыслительным возможностям МС, то произвол в выборе обозначений для чисел не так уж и велик: для представления *всех* натуральных чисел мы с необходимостью приходим к позиционной системе представления чисел.

Продолжим работу теперь уже с парой операторов $A(n)$ и $S(n)$.

Автоматное решение проблемы обозначений для чисел

В предыдущем разделе проблема обозначений для натуральных чисел решена в принципиальном плане. Здесь мы ее четко оформим и алгоритмизуем.

1. Для обозначения чисел используется конечное множество цифр $Z_4 = \{0,1,2,3\}$(в общем случае $Z_k = \{0,\ldots,k-1\}$, $k \geq 2$; в дальнейшем эту оговорку делать не будем).

2. Создается шаблон, состоящий из бесконечного количества мест. Каждое место в шаблоне изображается черточкой, под черточкой пишется номер места:

$$\ldots\ \_\ \ldots \underset{1}{\_}\,\underset{0}{\_}\ . \tag{15}$$

В роли номеров выступают порождаемые числа. Процесс нумерации мест обгоняет процесс порождения чисел, следовательно, процесс нумерации и процесс порождения чисел взаимно согласованы, непротиворечивы.

3. Процесс обозначения чисел начинается с того, что на все места ставится цифра 0. Получается символический объект:

$$\ldots\underline{0}\ldots\underset{1}{\underline{0}}\,\underset{0}{\underline{0}}\,\underline{0}, \tag{16}$$

обозначающий число 0.

4. Назовем *циклическим цифровым порядком* последовательность цифр

$$0 \prec 1 \prec 2 \prec 3 \prec 0. \tag{17}$$

Если цифры x и $x\prime$ связаны записью: $x \prec x\prime$, то будем говорить, что $x\prime$ *непосредственно следует за* x. (Отношение непосредственного следования « $\prec$ » нетранзитивно.)

5. Для обозначения чисел на места шаблона ставятся цифры из $Z_4 = \{0,1,2,3\}$. Пусть $\frac{x}{i}$ обозначает цифру $_{x \in Z_4}$, которая ставится на место с номером i. Тогда обозначение для произвольного натурального числа будет иметь вид:

$$\ldots\frac{0}{n\prime}\frac{x}{n}\ldots\frac{x}{2}\frac{x}{1}\frac{x}{0}, \tag{18}$$

где n' — следующее за n число. На месте n' и всех последующих стоят нули.

6. Подобно тому, как оператор A из числа n порождает следующее за ним число, оператор S, порождающий обозначения для чисел, из уже обозначенного числа порождает обозначение для следующего за ним числа. Пусть (18) — обозначение для некоторого числа. Оператор S действует следующим образом.

А0) Цифру $\frac{x}{0}$, согласно циклическому цифровому порядку (17),
заменяет на цифру $\frac{x\prime}{0}$.
Если $x' \neq 0$, то действие оператора S на этом заканчивается.
Если $x' = 0$, то действие оператора S переходит на цифру $\frac{x}{1}$.

А1) Цифру $\frac{x}{1}$, согласно циклическому цифровому порядку (17),
заменяет на цифру $\frac{x\prime}{1}$.
Если $x' \neq 0$, то действие оператора S на этом заканчивается.
Если $x' = 0$, то действие оператора S переходит на цифру $\frac{x}{2}$.

И т.д.

На этом описание действия оператора S заканчивается. Как видим, *описание действия оператора S* ***конечно****, но применяется* ***бесконечное*** *число раз*, т.е. мы удовлетворили требованию о создании процедуры, имеющей конечное описание (его, следовательно, МС с конечной памятью способно воспринять и запомнить), но которая может применяться (потенциально) бесконечное число раз. Каждое натуральное число получает свое обозначение. В системе цифр Z_4 разные числа получают разные обозначения. При переходе в другую систему счисления $Z_k = \{0, \ldots, k-1\}$, $k \geq 2$ числа также получат разные обозначения. Поэтому, вообще говоря, число имеет столько различных обозначений, сколько имеется систем счисления, определяющий параметр которых (т.е. $(k-1)$) не превосходит величины этого числа. Например, пусть число $\bar{E} = 7$. Тогда возможные обозначения для числа 7 в различных системах счисления будут такие:

$$\begin{aligned} [k=2] &\to 7_2 = \ldots 00111; \quad Z_2 = \{0,1\}. \\ [k=3] &\to 7_3 = \ldots 00021; \quad Z_3 = \{0,1,2\}. \\ [k=4] &\to 7_4 = \ldots 00013; \quad Z_4 = \{0,1,2,3\}. \\ [k=5] &\to 7_5 = \ldots 00012; \quad Z_5 = \{0,1,2,3,4\}. \\ [k=6] &\to 7_6 = \ldots 00011; \quad Z_6 = \{0,1,2,3,4,5\}. \\ [k=7] &\to 7_7 = \ldots 00010; \quad Z_7 = \{0,1,2,3,4,5,6\}. \\ [k=8] &\to 7_8 = \ldots 00007; \quad Z_8 = \{0,1,2,3,4,5,6,7\}. \\ [k=9] &\to 7_9 = \ldots 00007; \quad Z_9 = \{0,1,2,3,4,5,6,7,8\}. \\ [k=10] &\to 7_{10} = \ldots 00007; \quad Z_{10} = \{0,1,2,3,4,5,6,7,8,9\}. \end{aligned} \tag{19}$$

Хорошо видно, что число $\bar{x}= 7$ имеет 7 различных обозначений, т.е. количество различных обозначений для числа 7 равно количеству различных систем счисления, определяющий параметр которых (т.е. $(k-1)$) не превосходит 7 (это системы $Z_2, \ldots, Z_8$); для всех систем счисления, для которых $k \geq 8$, число 7 имеет одно и то же обозначение (это явление будем называть *автонотацией*). В этом примере мы не писали под цифрами номера мест, на которых они стоят. В силу небольшого количества знаков в изображении числа 7 номера мест легко восстанавливаются в уме; кроме того, так делается всегда: во избежание громоздкости в записях обозначений для чисел номера мест, на которые ставятся цифры, не пишут.

Обозначения для чисел являются достаточно содержательными представителями чисел, настолько содержательными, что после непродолжительного привыкания к ним их отождествляют с самими числами. Мы также, говоря далее о числах, будем иметь в виду обозначения для них. Тогда действие оператора A рождения чисел можно представить как действие оператора S, дающего обозначение для вновь порожденного числа, и мы можем записать (приняв для краткости $\frac{x}{i} \overset{\text{об}}{=} x_i$):

$$S\,(\ldots 0x_n \ldots x_2x_1x_0) = \begin{cases} \ldots 0x_n \ldots x_2x_1x_0', \text{ если } x_0' \neq 0; \\ \ldots 0x_n \ldots x_2x_1'0\,, \text{ если } x_1' \neq 0; \\ \ldots 0x_n \ldots x_2'0\;0\,, \text{ если } x_2' \neq 0; \\ \ldots\ldots\ldots\ldots\ldots \end{cases} \tag{20}$$

В формуле (20) предполагается, что результатом действия является первый реализованный случай при движении сверху вниз в правой части.

Формула (20) дает описание действий, которые надо выполнить над данным числом $\bar{x} = \ldots 0x_n \ldots x_2x_1x_0$, чтобы получить следующее за ним число Мы видим, что *позиционная система для обозначений чисел появляется с необходимостью*. Мы промоделировали с помощью МС процесс появления этой системы. Однако у (20) есть особенность, состоящая в том, что она требует наличия наблюдателя, находящегося извне по отношению к (20) и выполняющего действия, согласовываясь с (20). Наблюдатель должен: 1) посмотреть на цифру x_0; 2) пользуясь циклическим цифровым порядком (17), найти цифру x_0'; 3) выяснить, отлична ли она от 0; если отлична, то принять $\ldots 0x_n \ldots x_2x_1x_0'$ за следующее число и на этом закончить работу; если нет, то перейти к рассмотрению цифры x_1 и т.д. Нам бы хотелось другого: отделить функцию S от наблюдателя. Роль наблюдателя желательно свести к минимуму: чтобы он только подавал на вход функции S число, после этого в работу функции S не вмешивался и после работы функции S получал бы на выходе число, идущее в ряду чисел вслед за поданным.

Другими словами, мы хотим, чтобы функция S, если использовать терминологию теории систем, представляла собою самостоятельно функционирующую систему со входом и выходом или, если использовать аппаратную терминологию, была бы устройством по переработке информации, т.е. вычисляющим устройством.

Появление автомата сдвига $S\,\bar{x}|\bar{q}$

Проследим за работой функции S еще раз.

При переходе к следующему числу на «месте ноль» обязательно происходит замена цифры x_0 на цифру x_0' согласно циклическому цифровому порядку (3), §3.

Эту замену мы изобразим как соответствие:

$$\begin{array}{ccc} x_0 & \to & x_0' \\ \hline 0 & \to & 1 \\ 1 & \to & 2 \\ 2 & \to & 3 \\ 3 & \to & 0 \end{array} \tag{21}$$

Если была реализована последняя строка этого соответствия, то происходит переход к следующему «месту один» и выполняется такая же замена. Если же была реализована одна из первых трех строк, то на следующем «месте один» замена реализуется согласно другому соответствию:

$$\begin{array}{ccc} x_1 & \to & x_1 \\ \hline 0 & \to & 0 \\ 1 & \to & 1 \\ 2 & \to & 2 \\ 3 & \to & 3 \end{array} \tag{22}$$

Аналогичные действия при порождении следующего числа осуществляются и на других местах. Мы видим, что при работе функции S происходит переход от одного варианта соответствия к другому. Отметим это подробно, более информативно переписав соответствия, указывая на этот раз в каждой строке соответствия: а) какой вариант соответствия (21) или (22) реализуется при осуществлении замены цифры числа на данном месте числа, б) по какой строке соответствия осуществляется замена, в) к какому варианту соответствия нужно переходить на следующем месте порождаемого числа:

$$\begin{array}{cccccc} & x & \to & x' & & \\ \hline (1) & 0 & \to & 1 & (2) & \\ (1) & 1 & \to & 2 & (2) & \\ (1) & 2 & \to & 3 & (2) & * \\ (1) & 3 & \to & 0 & (1) & \\ \uparrow & & \uparrow & & \uparrow & \\ \text{а)} & & \text{б)} & & \text{в)} & \end{array} \qquad \begin{array}{cccccc} & x & \to & x & & \\ \hline (2) & 0 & \to & 0 & (2) & \\ (2) & 1 & \to & 1 & (2) & ** \\ (2) & 2 & \to & 2 & (2) & \\ (2) & 3 & \to & 3 & (2) & \\ \uparrow & & \uparrow & & \uparrow & \\ \text{а)} & & \text{б)} & & \text{в)} & \end{array} \tag{23}$$

Прочитаем, например, строчку, отмеченную знаком $*$:

«а) на некотором месте шаблона числа замена цифры, стоящей на этом месте, происходит согласно соответствию (21);

б) цифра 2 заменяется на цифру 3;

в) на следующем месте шаблона числа замена происходит согласно соответствию (22).»

Аналогично прочитаем строчку, отмеченную знаком $**$:

«а) на некотором месте шаблона числа замена цифры, стоящей на этом месте, происходит согласно соответствию (22);

б) цифра 1 заменяется на цифру 1;

в) на следующем месте шаблона числа замена происходит согласно соответствию (22).»

Как видно из (23), при переходе к следующему месту шаблона почти всегда замена цифр происходит по соответствию (22). Лишь в одном случае нарушается это наблюдение — в случае замены цифры 3 на цифру 0: в этом случае при переходе на следующее место шаблона числа нужно снова пользоваться соответствием (21). Информация о том, по какой схеме соответствия происходит замена цифры, есть, по существу, информация о том, в каком состоянии находится вычислительное устройство S: то ли в состоянии, когда срабатывает соответствие (21), то ли в состоянии, когда срабатывает соответствие (22). С этого момента введем термин «состояние вычислительного устройства» как рабочий термин при описании работы вычислительных устройств и дадим ему обозначение «q».

Вычислительное устройство S имеет, как уже сказано, два состояния. Эти два разных состояния мы будем различать, нумеруя их различными числами. Выберем, например, два числа: 0 и 1, так что в связи с q мы можем иметь два равенства: $q = 0$ или $q = 1$. Для упрощения изложения удобно столбцы цифр в соответствиях (21) и (22), стоящие слева от стрелок, называть входными (вход), а справа от стрелок — выходными (выход). Будем также считать, что работа устройства S в режиме соответствия (21) есть работа в состоянии q =1, а работа устройства S в режиме (22) есть работа в состоянии $q = 0$. Это позволит нам переписать (23) с использованием символов состояний, которые мы теперь будем писать рядом с цифрами, отделив их от цифр запятыми:

$$
\begin{array}{ccc}
q,x & \to & x',q \\
\hline
1{,}0 & \to & 1{,}0 \\
1{,}1 & \to & 2{,}0 \\
1{,}2 & \to & 3{,}0 \\
1{,}3 & \to & 0{,}1^*
\end{array}
\qquad\qquad
\begin{array}{ccc}
q,x & \to & x,q \\
\hline
0{,}0 & \to & 0{,}0 \\
0{,}1 & \to & 1{,}0 \\
0{,}2 & \to & 2{,}0^{**} \\
0{,}3 & \to & 3{,}0
\end{array}
\qquad (24)
$$

Прочитаем строку, отмеченную символом $*$:

«если устройство S находилось в состоянии 1 и на вход его поступила цифра 3, то на выходе его появляется цифра 0 и устройство переходит в состояние 1».

Аналогично читается строка $**$:

«если устройство S находилось в состоянии 0 и на вход его поступила цифра 2, то на выходе его появляется цифра 2 и устройство переходит в состояние 0».

Аналогично читаются все остальные строки (24).

Столбцы (24) дают принципиальную математическую основу вычислительного устройства S.

Будем называть далее вычислительное устройство S абстрактным вычислительным устройством S или коротко — АВУ S.

Введем часто употребляемый термин: наряду со словосочетанием «место в шаблоне числа» будем использовать также словосочетание «разряд числа».

Как мы видим, АВУ S при своей работе по порождению обозначения для нового числа осуществляет переходы от одного места шаблона числа к следующему месту. Этот переход совершается дискретно, и мы можем ввести дискретное время t со значениями $0, 1, 2, 3, \ldots$ и говорить, что для АВУ S переход во времени от одного момента к следующему соответствует переходу от одного разряда числа к следующему. Мы будем по тому или иному закону увязывать моменты времени с номерами разрядов числа. В связи с появлением времени t удобна иная интерпретация взаимодействия АВУ S с преобразуемым числом: если раньше мы говорили о перемещении АВУ S от одного места шаблона числа к следующему, то теперь будем считать АВУ S неподвижным, а число подвижным и поступающим поразрядно от младших разрядов к старшим на вход АВУ S.

Теперь совершим еще одно преобразование соответствий (24). Запишем их в виде единой таблицы. Эта таблица по содержанию совпадает со столбцами соответствий (24), но более удобна для использования и четче оттеняет роль состояний АВУ S.

Покажем, как пользоваться таблицей. Напомним, что если АВУ S находится в состоянии $q = 0$, то поступающие на вход АВУ S цифры после работы АВУ не меняются, т.е. на выходе АВУ появляется то, что поступило на вход, а если АВУ S находится в состоянии $q = 1$, то поступающие на вход цифры меняются согласно циклическому цифровому порядку (23).

Пусть, например, АВУ S в некоторый момент времени находилось в состоянии $q = 0$ и на вход АВУ S поступила цифра 2. Тогда на пересечении строки, отмеченной цифрой 2, и на пересечении столбца, отмеченного состоянием $q = 0$, мы найдем, что в следующий момент времени АВУ S окажется в состоянии $q = 0$ и на выходе его появится цифра 2, которая ставится на место шаблона числа, увязанное с этим моментом времени.

Аналогично, пусть АВУ S в некоторый момент времени находилось в состоянии $q = 1$ и на вход АВУ S поступила цифра 3. Тогда на пересечении строки, отмеченной цифрой 3, и на пересечении столбца, отмеченного состоянием $q = 1$, мы найдем, что в следующий момент времени АВУ S окажется в состоянии $q = 1$ и на выходе его появится цифра 0, которая ставится на место шаблона числа, увязанное с этим моментом времени.

Таблица 1

АВУ S

x \ q	0	1
0	0, 0	0, 1
1	0, 1	0, 2
2	0, 2	0, 3
3	0, 3	1, 0

Работу АВУ S по преобразованию числа $\bar{x}$ в следующее за ним число $\bar{x}'$удобно проследить с помощью так называемой *развертки во времени работы АВУ S* (далее –развертка). Развертка представляет собою многострочную структуру, достаточно легко воспринимаемую и позволяющую без лишних слов следить за работой АВУ S. Покажем это на примере. Пусть на вход АВУ S поступает число $\bar{x} = \ldots 0001233$. Требуется с помощью АВУ S найти следующее за ним число $\bar{x}'$. Изобразим развертку:

$$\begin{array}{ccccccccccc|c}
5 & & 4 & & 3 & & 2 & & 1 & & 0 & t \\
\hline
 & & 0 & & 1 & & 2 & & 3 & & 3 & x \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
0 & \leftarrow & 0 & \leftarrow & 0 & \leftarrow & 1 & \leftarrow & 1 & \leftarrow & 1 & q \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
 & & 0 & & 1 & & 3 & & 0 & & 0 & x'
\end{array}$$

Читаем содержимое развертки.

При $t = 0$ на вход АВУ S поступает цифра $x_0 = 3$ числа $\bar{x}$ (цифра этого разряда соответствует моменту $t = 0$). Устройство S в этот момент времени находится в состоянии q=1. По таблице 1 находим, что устройство S в следующий момент времени $t = 1$ переходит в состояние $q = 1$ (на развертке показано стрелочкой влево) и вырабатывает выходной сигнал $x' = 0$ (на развертке показано стрелочкой вниз).

При $t = 1$ ситуация повторяется.

При $t = 2$ на вход АВУ S поступает цифра $x_2 = 2$ числа $\bar{x}$ (цифра этого разряда соответствует моменту $t = 2$). Устройство S в этот момент времени находится в состоянии $q = 1$. По таблице 1 находим, что устройство S в следующий момент времени переходит в состояние $q = 0$ (на развертке показано стрелочкой влево) и вырабатывает выходной сигнал $x'_2 = 3$ (на развертке показано стрелочкой вниз).

При $t = 3$ на вход АВУ S поступает цифра $x_3 = 1$ числа $\bar{x}$ (цифра этого разряда соответствует моменту $t = 3$). Устройство S в этот момент времени находится в состоянии $q = 0$. По таблице 1 находим, что устройство S в следующий момент времени переходит в состояние $q = 0$ (на развертке показано стрелочкой влево) и вырабатывает выходной сигнал $x'_3 = 1$ (на развертке показано стрелочкой вниз).

И т.д.

На выходе АВУ S получится число $\bar{x}'= \ldots 001300$. Таким образом, АВУ S вычислило число $\bar{x}'$, следующее за числом $\bar{x} = \ldots 001233$ (напоминаем, что у нас было намерение отделить наблюдателя от АВУ S и следить за работой S как бы со стороны; по этой причине мы подчеркиваем самостоятельность работы АВУ). Запишем результат в виде равенства:

$$S(\ldots 001233) = \ldots 1300. \tag{25}$$

Итак, мы с необходимостью получили АВУ S, вычисляющее по каждому числу $\bar{x}$ следующее за ним число $\bar{x}'$. В-схема АВУ S показана на рис. 2 [1].

$S\ \bar{x}|\bar{q}$ — это полное обозначение для АВУ S, где S — символ вычисляемой функции, $\bar{x}$ — число на входе (аргумент), $\bar{q}$ — состояние, в которое ставится АВУ S в начальный момент времени (начальное состояние).

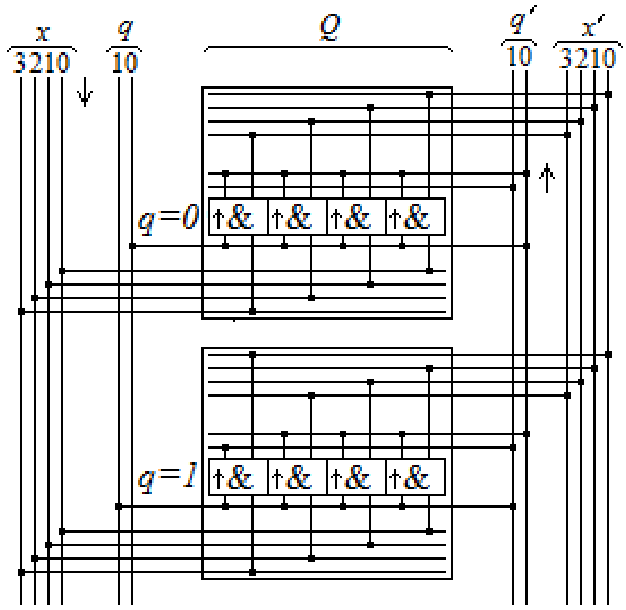

Рис. 2. *В-схема АВУ* $S\bar{x}|\bar{q}$

Устройство работает независимо от наблюдателя. Роль наблюдателя сводится только к подаче числа на вход устройства и к считыванию результата. Устройство, работу которого мы разобрали, функционирует в четверичной системе счисления с набором цифр $Z_4 = \{0,1,2,3\}$. Но аналогичное устройство может быть создано для любой другой системы счисления, причем изменения, которые должны быть сделаны в устройстве *S*, заданном таблицей 1, совершенно очевидны. Устройство *S* осуществляет сдвиг в последовательности натуральных чисел (точнее, в последовательности обозначений для них) от произвольного числа к следующему за ним числу. Поэтому оно имеет другое название: «устройство сдвига». Про него говорят, что устройство сдвига реализует функцию сдвига.

Сделаем еще одно усовершенствование. Оно существенно как с принципиальной, так и с практической точек зрения. А именно, в числе $\bar{x} = ...000x_n...x_2x_1x_0$ заменим бесконечное число нулей $...000$ на значок $\overleftarrow{0}$, в результате чего для числа $\bar{x}$ будем иметь другую запись:

$$\bar{x} = \overleftarrow{0}\ x_n...x_2x_1x_0\ . \tag{26}$$

Понятно, что, заменив бесконечное число нулей $...000$ на значок $\overleftarrow{0}$, мы совершили актуализацию бесконечности:

$$Act(...000) = \overleftarrow{0}\ . \tag{27}$$

Символ $\overleftarrow{0}$ информативно полезен: он показывает не только то, что влево идет бесконечное число нулей, но и то, что номера разрядов в записи числа растут влево. Это сразу наводит на мысль ввести не только символ $\overleftarrow{0}$, но и символ $\vec{0}$, который показывал бы, что в записи числа разряды растут не влево, а вправо, и что вправо идет, начиная с некоторого места, бесконечное число нулей. Это делает две возможные записи числа равноправными:

$$\bar{x} = \overleftarrow{0}\ x_n...x_2x_1x_0 = x_0x_1x_2...x_n\vec{0}. \tag{28}$$

Символ $\overleftarrow{0}$ целесообразно ввести во множество цифр Z_4:

$$Z_4 = \left\{\overleftarrow{0}\ ,0,1,2,3\right\}. \tag{29}$$

А принимая во внимание развертку во времени работы АВУ *S*, мы понимаем, что должно быть внесено также небольшое дополнение в таблицу 1.

Появление символа $\overleftarrow{0}$ повлияет также на развертку, упростив ее и обогатив дополнительным содержанием. В чем суть достигнутого упрощения и дополнительного содержания? Во-первых, развертка становится конечной, а не потенциально бесконечной, как раньше. Во-вторых, мы получили

Таблица 2

АВУ $S\bar{x}|\bar{q}$

x \ q	0	1
$\overleftarrow{0}$	0, $\overleftarrow{0}$	0, 1
0	0, 0	0, 1
1	0, 1	0, 2
2	0, 2	0, 3
3	0, 3	1, 0

дополнительный информативный сигнал: *появление* $\overleftarrow{0}$ *в последней строке развертки означает конец работы АВУ S*. Таким образом, мы получили признак, по которому можно судить о конце работы устройства. Так в АВУ *S* решается проблема останова.

$$
\begin{array}{ccccccccccc|c}
5 & & 4 & & 3 & & 2 & & 1 & & 0 & t \\
\hline
 & & \overleftarrow{0} & & 1 & & 2 & & 3 & & 3 & x \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
0 & \leftarrow & 0 & \leftarrow & 0 & \leftarrow & 1 & \leftarrow & 1 & \leftarrow & 1 & q \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
 & & \overleftarrow{0} & & 1 & & 3 & & 0 & & 0 & y
\end{array}
$$

И, наконец, бросив ретроспективный взгляд на таблицу 2, мы легко приходим к выводу, что первую и вторую строчки таблицы (не считая «шапки» таблицы) можно объединить. В результате получается итоговая таблица 3 для устройства *S*.

В этой таблице стрелочки над нулями поставлены в скобки, что соответствует тому, что на вход устройства *S* нули поступают не всегда со стрелочками. В случае, если на вход поступил нолик со стрелочкой, скобки нужно удалить.

Таблица 3

АВУ $S\bar{x}|\bar{q}$

x \ q	0	1
$\overset{(\leftarrow)}{0}$	0, $\overset{(\leftarrow)}{0}$	0, 1
1	0, 1	0, 2
2	0, 2	0, 3
3	0, 3	1, 0

АВУ *S*, задаваемое таблицей 3, *полностью решает проблему обозначений для натуральных чисел* и представляет множество натуральных чисел в следующих обозначениях:

$$N_0 = \left\{ \overleftarrow{0}\ ,\ \overleftarrow{0}\,1,\ \overleftarrow{0}\,2,\ \overleftarrow{0}\,3,\ \overleftarrow{0}\,10,\ \overleftarrow{0}\,12,\ \overleftarrow{0}\,13,\ \overleftarrow{0}\,20,\ \overleftarrow{0}\,21,... \right\}. \tag{30}$$

Отметим, что АВУ *S, будучи генератором натуральных чисел, предъявляемых нам в их обозначениях, отправляясь от* $\overleftarrow{0}$, *последовательно порождает свою область определения — множество* N_0. Областью значений АВУ *S* при этом оказывается множество $N_0 \backslash \{\overleftarrow{0}\}$, т.е. множество всех натуральных чисел без нуля. Действительно, при действии АВУ *S* на любое натуральное число $\overleftarrow{0}$ никогда не получается. АВУ *S*, будучи применено к любому числу упорядоченной последовательности (30), совершает переход по этому ряду чисел на один шаг вправо и тем самым выполняет операцию «сдвиг». Поэтому АВУ *S* называют *автоматом сдвига*, каноническая B-схема автомата сдвига $S\ \bar{x}|\bar{q}$ изображена на рис. 2.

Порождение автоматом сдвига $S\ \bar{x}|\bar{q}$ других вычислительных устройств

В основе порождения других вычислительных устройств лежит *метод последовательного синтеза* устройств, который перерастает в *метод сетевого синтеза*. Оба метода моделируют историческое развитие математики и являются аппаратной реализацией построения функций посредством суперпозиции. Созданное таким образом многообразие функций вполне удовлетворяет практическим потребностям математики. A'priori функции, взятые «с потолка», могут не входить в это многообразие, являясь по отношению к этому многообразию его идеальными периферийными объектами или прародителями других семейств функций, могущих быть специально рассмотренными. Дело в том, что автоматов типа $S\ \bar{x}|\bar{q}$, способных порождать аналогичные многообразия функций, существует 20^{20}, где 10 — основание *k*-ичной системы счисления. Эти многообразия могут быть рассмотрены наподобие рассмотрения функций, порождаемых автоматом сдвига, но в любом случае это будут В-автоматы, возможно, перспективные, но в настоящее время находящиеся за пределами как теоретических, так и практических рассмотрений.

Характерным признаком этого подхода является его «демократичность», состоящая в том, что ни одна система счисления не выделяется на фоне других, все системы равноправны. Схемная реализация этих устройств также обладает этой особенностью. Такой схемной реализацией является *В-схемная* реализация, одинаково применимая для всех систем счисления. Поэтому В-индустрия является компьютерной индустрией, допускающей потенциально неограниченное развитие не только по такому параметру, как система счисления, но и по многообразию потенциальных логик, со своими особыми арифметиками и математиками.

В результате применения метода последовательного синтеза получаются поэтапно семейство автоматов сдвига на целую константу, семейство сумматоров и вычитателей, семейство умножителей и делителей, семейство экспоненциаторов и т.д. Принципиально ясная схема построения устройств наталкивается на трудно преодолимую преграду, заключающуюся в сложности устройств высокого порядка. Тем не менее то, что удается получить, служит содержательной основой для построения В-компьютера с многообразными возможностями.

В-компьютеры и искусственный интеллект

В-компьютеры в полной мере приспособлены к созданию искусственного интеллекта. Существует два подхода к пониманию искусственного интеллекта. Согласно первому подходу под *искусственным интеллектом понимается искусственно созданный объект, обладающий способностью к самостоятельной интеллектуальной деятельности.* Будем называть это понятие *искусственным интеллектом в* ***строгом*** *смысле.* Таким образом, первой отличительной чертой искусственного интеллекта является его искусственное происхождение, т.е. он должен быть создан кем-то другим, способным к достаточно высокой творческой деятельности, Роботы не являются примерами ИИ в строгом смысле по той причине, что, хотя они и созданы искусственно, но собственным интеллектом не обладают; они ведут себя, подчиняясь приказам, исходящим от встроенных в них компьютеров, в которых, в свою очередь, содержатся программы, написанные человеком. Эти программы определяют поведение роботов и их зависимость от человека.

Согласно второму подходу под *искусственным интеллектом понимается искусственно созданный объект, обладающий способностью к разумной поведенческой деятельности в пределах запрограммированной компетенции.* Будем называть это понятие *искусственным интеллектом в* ***слабом*** *смысле.* Все известные роботы реализуют искусственный интеллект в слабом смысле. В конечном счете поведение такого робота определяется человеком. Все роботы управляются компьютерами, атрибутом которых являются программы, создаваемые человеком. Поэтому поведение роботов целиком и полностью зависит от человека. Следовательно, эти роботы могли бы быть названы *программируемыми роботами с разумным поведением*, а не искусственным интеллектом в строгом смысле. До тех пор, пока они управляются программируемыми компьютерами, искусственным интеллектом в строгом смысле они названы быть не могут. В этих условиях никакой «бунт машин» невозможен. Просто-напросто потому, что у этих машин нет собственного интеллекта и они полностью подчиняются человеку. Обычные вычислительные программы являются такими же роботами, как и те объекты, которые привычно называются роботами, потому что вычислительные программы ведут себя разумно в процессе вычислений, но их разумное поведение, в отличие от поведения роботов, не обнаруживается внешним образом. Еще одним примером роботов являются банкоматы, проявляющие искусственный интеллект в слабом смысле. Они демонстрируют довольно сложное поведение в пределах своей компетенции, обусловленной управляющими программами. Примеры *программируемых роботов с разумным поведением* можно приводить до бесконечности.

Итак, *искусственный интеллект в слабом смысле* используется повсеместно. Именно он, искусственный интеллект в слабом смысле, обычно принимается в быту за полноценный искусственный интеллект. Мы, однако, будем отличать искусственный интеллект в строгом смысле от искусственного интеллекта в слабом смысле, понимая, что подлинный искусственный интеллект есть ИИ в строгом смысле.

Поскольку В-компьютеры обладают способностью неограниченно пополняться вычислительными устройствами различного целевого назначения и тем самым расширять свои возможности, то они вполне приспособлены к созданию искусственного интеллекта как в строгом, так и в слабом смыслах. При этом, поскольку в В-компьютерах вместо программ используются устройства, то управленческие функции, осуществляемые в них над искусственным интеллектом, будут выполняться с большей надежностью и с большей скоростью.

В-компьютеры и нейронные сети

В-компьютеры обладают еще одним уникальным свойством: они могут внутри себя содержать собственное производство по созданию вычислительных устройств. Это обусловлено возможностью включать в конструкцию компьютера блок заготовок, на которых создаются вычислительные устройства. В дальнейшем созданное устройство встраивается в общую конструкцию компьютера и принимает участие в его работе на общих основаниях наряду с другими, уже существующими устройствами. Объем заготовок не имеет ограничений, вследствие чего компьютер имеет возможность неограниченно развиваться. Заготовки имеют предварительную классификацию и согласно этой классификации возможно создание устройств, принадлежащих тем или иным классам. Заготовка является аналогом полуфабриката и превращается в реальное изделие после дополнительной работы над ним. На заготовках создается вычислительное устройство, реализующее нужное отображение. Информацию об отображении первоначально мы имеем фрагментарно, и по имеющемуся фрагменту отображения создается устройство, реализующее отображение в целом. Ситуация здесь напоминает ситуацию с нейронными сетями, где также создается структура в результате процедуры, называемой обучением. Эта процедура по фрагментам отображения создает структуру (нейронную сеть), реализующую, как предполагается, отображение в целом. Почему предполагается? Потому что нейронная сеть представляет собою «черный ящик», а что находится в нем, неизвестно. Лишь по его работе делается суждение о том, что структура правильная. Так, например, на химфаке МГУ была создана нейронная сеть, расшифровывающая устройство сложных молекул, при этом создатели нейронной сети комментировали: «Пока она работает правильно. Но как она это делает, мы не знаем». Этот комментарий подчеркивает, что нейронная сеть – это черный ящик. В отличие от этого, устройство, синтезированное на заготовке, нам полностью известно, его схему мы можем посмотреть и изучить. Это устройство черным ящиком для нас не является, следовательно, и отображение, им реализуемое, нам может быть достаточно хорошо известно.

Обратим также внимание на тот факт, что синтезированное устройство k-ичное, не обязательно двоичное.

Таким образом, В-компьютеры могут содержать в себе отсеки, заменяющие работу нейронных сетей.

Зачем иметь в В-компьютере отсек с заготовками? В качестве примера, объясняющего причину этого, можно привести пример двух конкурирующих компьютеров и один из них – это В-компьютер с заготовками. Пусть В-компьютер имеет возможность получить информацию об отображении (естественно, частичную) компьютера-конкурента. Тогда он сможет по имеющимся у него фрагментам синтезировать устройство компьютера-конкурента и оказаться в результате способным предвидеть действия конкурента. Понятно, что это большое конкурентное преимущество.

Другие причины, по которым желательно иметь отсек с заготовками, могут быть связаны с оптимизацией работы исходного базового компьютера.

ЛИТЕРАТУРА

1. Деев Г. Е., Рахов Э. В. Устройство для сложения числа с константой. А. с. СССР № 1278836, кл. G06 F7/50, бюл. № 47, 1986.
2. Деев Г. Е. *Абстрактные вычислительные устройства*. Т. 2. М.: Энергоатомиздат; 2007. 332 с.
3. Деев Г. Е. *Теория вычислительных устройств*. Санкт-Петербург: Лань; 2019. 452 с.
4. Лорьер Ж.-Л. *Системы искусственного интеллекта*. Москва: Мир; 1991.
5. Тейз А., Грибомон П. и др. *Логический подход к искусственному интеллекту*. Москва: Мир; 1991.
6. Нильсон Н. *Искусственный интеллект*. Москва: Мир; 1973.
7. Арбиб М. *Метафорический мозг*. Москва: Мир; 1976.
8. Артоболевский И., Кобринский А. *Знакомьтесь — роботы*. Москва; Молодая гвардия; 1980.
9. Эндрю А. *Искусственный интеллект*. Москва: Мир; 1985.
10. Минский М. На пути к созданию искусственного разума. *Вычислительные машины и мышление* / Под ред. Э. Фейгенбаума и Дж. Фельдмана. Москва: Мир; 1967. С. 402–458.
11. Лем С. *Сумма технологий*. Москва: Мир; 1968.