DOI: 10.51790/2712-9942-2020-1-2-6

В-КОМПЬЮТЕРЫ: САМОРАЗВИТИЕ**Г. Е. Деев^а, С. В. Ермаков^б***Обнинский институт атомной энергетики, Национальный исследовательский ядерный университет «МИФИ», г. Обнинск, Российская Федерация,**^а georgdeo@mail.ru, ^б ermakov@iate.obninsk.ru*

Аннотация: показано, что В-компьютеры обладают неограниченным потенциалом развития, реализуемым на уровне разнообразных цифровых устройств, интегрируемых В-компьютером. На основании этого факта делается вывод о том, что решение задачи создания искусственного интеллекта в полной мере может быть осуществлено на базе В-компьютеров.

Ключевые слова: В-компьютер, искусственный интеллект (ИИ), шаблон заготовки, абстрактное вычислительное устройство (АВУ), В-схема.

Благодарности: авторы выражают благодарность А. В. Ламкову за обсуждение работы. Работа выполнена при поддержке гранта РФФИ 20-07-00862.

Для цитирования: Деев Г. Е., Ермаков С. В. В-компьютеры: саморазвитие. *Успехи кибернетики*. 2020;1(2):47–55. DOI: 10.51790/2712-9942-2020-1-2-6.

B-COMPUTERS: SELF-EVOLUTION**Georgiy E. Deev^a, Sergey V. Ermakov^b***Obninsk Institute for Nuclear Power Engineering, National Research Nuclear University MEPhI, Obninsk, Russian Federation**^a georgdeo@mail.ru, ^b ermakov@iate.obninsk.ru*

Abstract: the study demonstrates that the evolution of the B-computer can be unlimited as it can integrate a range of other digital devices. With this fact, we claimed that AI can be implemented with B-computers.

Keywords: B-computer, artificial intelligence (AI), PCB layout template, abstract computer (AC), B-circuit.

Acknowledgements: I wish to show my appreciation to A.V. Lamkov for reviewing this paper. This study is supported by RFBR grants 20-07-00862.

Cite this article: Deev G. E., Ermakov S. V. B-Computers: Self-Evolution. *Russian Journal of Cybernetics*. 2020;1(2):47–55. DOI: 10.51790/2712-9942-2020-1-2-6.

Предварительное неформальное пояснение

В работе упоминаются «А-компьютеры» и «В-компьютеры» (буквы «А» и «В» — латинские). Под А-компьютерами мы имеем в виду повсеместно используемые компьютеры и родственные им устройства типа всевозможных гаджетов. В связке с А-компьютерами идут В-компьютеры. А-компьютеры имеют исторический приоритет, они первичны, что и оттеняется использованием буквы «А» в их названии. По отношению к ним В-компьютеры вторичны, что также оттеняется использованием в их названии буквы «В». А-компьютеры и В-компьютеры объединяют и одновременно противопоставляют родственные принципы построения абстрактных схем. Принцип А, служащий для построения схем А-компьютеров, гласит: *по одной и той же линии могут передаваться различные информационные сигналы*. Но, понятно, слишком много различных информационных сигналов по одной и той же линии передаваться не может, т.к. в этом случае возникает трудноразрешимая проблема различения сигналов, приводящая к ненадежным вычислениям. Это послужило основанием Дж. фон Нейману для формулировки тезиса о том, что наилучшей системой счисления, пригодной для создания компьютеров, является двоичная система счисления. И действительно, в этом случае вычисления максимально надежны. С тех пор вся компьютерная индустрия двоичная. Вычисления на А-компьютерах, будучи максимально надежными в двоичной системе счисления, тем не менее не избавлены от сбоев.

Принципу А противопоставляется двойственный ему принцип В, гласящий: *по одной и той же линии не могут передаваться различные информационные сигналы*. Как видно, оба принципа, будучи

антитезой друг другу, дополняют один другой так, что третьего не дано. Таким образом, в силу принципа В, между линией и информационным сигналом имеет место взаимно однозначное соответствие и путаницы сигналов в линии никогда не происходит. Этим обеспечивается надежность вычислений при любом количестве сигналов. Поэтому тезис Неймана к В-компьютерам неприменим. Следовательно, В-компьютеры можно создавать для любой системы счисления, не испытывая принципиальных трудностей. Одной из особенностей В-компьютеров является простота базиса, на основе которого можно строить как сами В-компьютеры, так и содержащиеся в них В-устройства. Этот конструктивный базис состоит из двух элементов: из линии, по которой передаются информационные сигналы, и из элемента &, управляющего направлением движения информационного сигнала. Базис можно изобразить записью: $\{ |, \& \}$; такой набор конструктивных элементов удовлетворяет принципу максимальной простоты. Следовательно, В-схемы В-компьютеров максимально просты по своему конструктивному исполнению.

Всякий объект находится на той или иной стадии развития. Под *стадией развития* объекта понимается его *конструктивное состояние*, а также набор функциональных взаимодействий объекта с окружающей средой, который мы будем называть *набором функций объекта*. Под *развитием* понимается переход объекта от одной стадии развития к другой. Это подразумевает как возможное конструктивное изменение самого объекта, так и изменение набора связанных с ним функций. В связи с понятием развития сразу же возникает необходимость в характеристике этого развития, будет ли развитие регрессивным или прогрессивным. Как сказано выше, каждая стадия развития характеризуется двумя компонентами: конструктивными особенностями объекта и его функциональным набором. Вполне естественно предполагать, что функциональный набор объекта зависит от его конструкции, так что набор функций, характеризующих стадию развития, есть своего рода функция его конструктивного совершенства, что отражается записью $\{f_1, f_2, \dots\} = \Psi(k)$, где k — коэффициент, характеризующий конструктивное развитие. Будем считать его меняющимся в пределах $0 \leq k < \infty$ (нет предела совершенству). Итак, стадия развития объекта — это пара $\langle k, \{f_1, f_2, \dots\} \rangle$, состоящая из коэффициента конструктивного развития k и набора функций $\{f_1, f_2, \dots\}$. Это своеобразный *вектор развития*. Не будем, однако, далее углублять векторную аналогию. Как регресс, так и прогресс характеризуются поведением пары $\langle k, \{f_1, f_2, \dots\} \rangle$. Оба фактора, входящие в эту пару, могут определяться внешним воздействием, тогда мы говорим об индуцированном развитии. Но может быть и так, что поведение пары $\langle k, \{f_1, f_2, \dots\} \rangle$ определяется внутренними причинами, присущими объекту, при слабом или вообще отсутствующем внешнем воздействии. Тогда мы говорим о *саморазвитии*.

С понятием саморазвития тесно связано понятие *искусственного интеллекта*, (ИИ), столь популярное в последнее время. Под *искусственным интеллектом* понимается *искусственно созданный объект, обладающий способностью к самостоятельной интеллектуальной деятельности*. Таким образом, первой отличительной чертой искусственного интеллекта является его искусственное происхождение, т.е. он должен быть создан кем-то другим, способным к творческой деятельности, например, человеком творящим или каким-либо еще мыслящим существом (МС), способным к созданию творений. По этой причине муха не является искусственным интеллектом, несмотря на то, что она проявляет признаки интеллекта. Не является по той причине, что ни человек, ни какое-либо другое МС не являются ее создателями. С другой стороны, машинка, катающаяся по полу и управляемая мальчиком с помощью рычажков на пульте управления, также не является ИИ, несмотря на кажущуюся разумность ее поведения. Кажущаяся разумность — это результат работы интеллекта мальчика, а не самой машины. Аналогично знаменитые японские роботы не являются примерами ИИ 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. В конечном счете, поведение робота определяется человеком, как и в случае с машинкой. Все роботы управляются А-компьютерами, атрибутом которых являются программы, создаваемые человеком. Поэтому поведение роботов целиком и полностью зависит от человека. Следовательно, до тех пор, пока они управляются А-компьютерами, никакой «бунт машин» не возможен. Просто-напросто потому, что у этих машин нет собственного интеллекта.

Другое дело — В-компьютеры. В-компьютеры обладают неограниченным потенциалом развития. Наличие потенциала развития допускает возможность саморазвития и в конечном итоге превращения его в интеллект. Этот потенциал может быть использован человеком в своих целях, а может быть использован самим компьютером для собственного развития. Приобретя интеллект, В-компьютер может вступить в конкурентную интеллектуальную игру с человеком и даже выиграть ее. Не обязательно, однако, дело должно закончиться «войной», вполне возможен дружеский симбиоз человека и машины.

Синтез полностью определенных устройств на шаблонах заготовок

Заготовками мы называем абстрактные платы, на которых заранее подготовлены полуфабрикаты для будущих схем устройств. Заготовки удовлетворяют требованию максимальной простоты. На заготовках имеются линии для передачи информационных сигналов, а также множество элементов &. Больше ничего нет. Если в заготовке все линии параллельны, то в такой заготовке варьирование связей невозможно. Варьирование связей возможно только в заготовке, где есть перекрещивающиеся линии. Если есть перекрещивающиеся линии, то их соединение, хотя и возможно, но не обязательно. Наличие соединения будем изображать жирной точкой в перекрестии, как показано на рис. 1.

Рис. 1. Соединения нет (слева), соединение есть (справа)

Таким образом, в абстрактных В-схемах выполнение операции соединения сводится к простейшей операции — к простановке точки.

Рис. 2. Пример варианта соединения

На рис. 2 показан один из возможных вариантов соединения пяти перекрещивающихся линий. Всего возможно 2^6 таких вариантов. Этим числом оценивается потенциал развития (потенциал возможностей) данной конфигурации линий.

На следующей странице приведен пример типичной заготовки. Слева изображена «сырая» заготовка, на которой нет никаких соединений; справа — она же, но на ней выбран один из возможных вариантов соединений. Очевидно, что вариантов соединений невообразимо много (2^{728}) и каждому варианту соединений соответствует свое вычислительное устройство. Не все из них представляют в настоящий момент какой-либо интерес, но некоторые, несомненно, представляют. Именно таковым является устройство, получившееся в результате изображенного варианта соединений. Это устройство есть не что иное, как умножитель на три в четверичной системе счисления. Оно обозначается символом: $\bar{x} \cdot 3|\bar{q}$, где $\bar{x}$ — числоид, записанный на сетке $Gr^0 = \dots \bar{2} \bar{1} \bar{0}$, представляющий, в частности, натуральные числа; $\bar{q}$ — состояние умножителя, $\bar{q} \in \{0, 1, 2\}$. Как видно, после выбора варианта соединений на заготовке могут оставаться «лишние» линии. Не играя какой-либо роли по отношению к выбранному варианту соединений, они могут нести новые информационные сигналы и оказаться полезными для других целей. Числом 2^{728} выражается «скрытый» потенциал развития, потенциал возможностей, связанный с заготовками такого типа. Если учесть, что типов заготовок может быть также невообразимо много, то можно представить себе, каков потенциал развития, связанный с В-компьютерами. Символьная атрибутика, присутствующая на рис. 3, необходима нам для понимания функционирования устройства. Само устройство, на современной стадии его развития, в ней не нуждается. Среди множества режимов работы устройства есть режимы, осмысленные с нашей точки зрения. Поэтому имеет смысл снабдить некоторые элементы схемы понятными нам информационными сигналами. Так,

символами 0, 1, 2, 3 снабжены некоторые линии устройства. Группы из четырех линий образуют вход и выход устройств, группы из трех линий — управляющие сигналы. Группа элементов &, у которых на первом (слева) месте стоит 0, образует состояние $\bar{q} = 0$; аналогично, состояния $\bar{q} = 1$ и $\bar{q} = 2$.

Рис. 3. Заготовка без соединений (слева) и заготовка с соединениями (справа)

Состояние $\bar{q} = 0$ будем называть начальным состоянием. Устройство выполняет умножение на три, будучи поставленным при $t = 0$ именно в начальное состояние. Если его ставить при $t = 0$ в другие состояния, то оно вычисляет другие функции.

Рассмотрим пример вычисления этим устройством.

Пример 1. Пусть $\bar{x} = \dots 00322$ в четверичной системе счисления. За вычислением следим с помощью *развертки во времени процесса вычисления*.

Информация в развертке читается справа налево. Так же справа налево растут номера разрядов в числоидах. В первой строке показаны моменты времени, в которые функционирует устройство. В роли таких моментов выступают целые неотрицательные числа. Во второй строке стоят цифры числа, поданного на вход. В разряде 0, соответствующем моменту $t = 0$, стоит цифра $x_0 = 2$. Аналогично во всех остальных разрядах.

6	5	4	3	2	1	0	t
...	0	0	0	3	2	2	x_t
&00	&00	&00	&20	&13	&12	&02	$\&qx_t$
...	0	0	2	2	3	2	y_t

В строке $\&qx_t$ стоят элементы &, которые срабатывают в момент t . В строке y_t стоят поразрядно расположенные цифры результата, причем соответствие между моментом времени t и номером разряда, как и для входа, сохраняется. Отметим, что устройство является абстрактом, т.е. идеальным объектом, в котором все идеально срабатывает, все элементы & имеют одну и ту же задержку; выходной сигнал выдается в тот же момент времени, что и поступивший входной. Реально выходной сигнал всегда выдается с некоторым запаздыванием относительно входного сигнала. Но инженеры, создающие вычислительные устройства, всегда стремятся это запаздывание уменьшить, насколько это возможно. Математики этот процесс доводят до конца, сводя запаздывание к нулю. Ясно, что это идеализация, но она не портит математического содержания вычислительного процесса, а, скорее, именно она его представляет.

Понятно, что одной конструкции устройства недостаточно, чтобы стал ясен принцип его работы. Требуется еще дать закон функционирования устройства, что обеспечивается его уравнениями функционирования. Они имеют вид:

$$\begin{cases} \bar{q}_{t+1} = \theta \bar{q}_t x_t = \langle 3 \cdot x_t + \bar{q}_t \rangle, \\ y_t = \sigma \bar{q}_t x_t = \langle 3 \cdot x_t + \bar{q}_t \rangle, \end{cases} \quad (1)$$

где $\bar{q} \in Q_3 = \{0, 1, 2\}$, $x, y \in Z_{(4)} = \{0, 1, 2, 3\}$, θ — функция переходов, σ — функция выходов. В (1) используются обозначения: $\langle \bar{\alpha} \rangle = \langle \dots 00\alpha_r \dots \alpha_1 \alpha_0 \rangle = \alpha_0$ и $\langle \bar{\alpha} \rangle = \langle \dots 00\alpha_r \dots \alpha_1 \alpha_0 \rangle = \dots 00\alpha_r \dots \alpha_1$.

Читаем развертку: при $t = 0$ устройство находится в начальном состоянии $\bar{q} = 0$, что обеспечивается поступлением управляющего сигнала по линии-0 входа-С. К срабатыванию готовы все четыре элемента & начального состояния: &00, &01, &02, &03. Но сработает только один из них, а именно тот, который соответствует поступившей на вход цифре, которой является цифра $x_0 = 2$. Поэтому сработает элемент &02, что и отмечено в развертке. Согласно схеме устройства на выходе появится цифра 2, и будет передан управляющий сигнал для момента времени $t = 1$ на элементы & состояния 1, т.е. на элементы &10, &11, &12, &13. В этот же момент на вход поступает цифра $x_1 = 2$; далее все происходит аналогично тому, как это имело место при $t = 0$. В итоге на выходе сформируется результат, число $\bar{y} = \dots 002232$.

Запишем результат работы устройства в виде: $\dots 00322 \cdot 3|0 = \dots 002232$.

Читаем: если поставить устройство в состояние $\bar{q} = 0$ как начальное (это обозначено нулем справа от вертикальной черты) и подать на вход число $\bar{x} = \dots 00322$, то устройство выдаст результат $\bar{y} = \dots 2232$, являющийся произведением числа $\bar{x} = \dots 00322$ на три: $\dots 00322 \cdot 3 = \dots 2232$. Результат проверяется школьным умножением «столбиком» (в четверичной системе счисления).

Обратим внимание, что результат выдается при $t = 2$, т.е. за 3 (а не за 4) такта. Это объясняется тем, что старшие две цифры результата выдаются устройством одновременно. Это явление обобщается при рассмотрении всех В-устройств, что приводит к выводу о том, что *в В-устройствах вычисление заканчивается в момент поступления старшей цифры из входной информации*. Это справедливо только для В-устройств. Аналогично можно разобрать работу любого из 2^{728} устройств, которые можно создать на шаблоне заготовки, представленной на рис. 3. Если учесть, что шаблонов типа приведенного и других, значительно более сложных, невообразимо много, то, повторим, ясно, насколько велик потенциал развития В-устройств.

В разобранном примере предполагалось, что отображение, реализуемое устройством, *полностью задано*, т.е. известны или уравнения функционирования устройства (1), или его табличное задание (не приводилось), или его В-схема (не приводилась). Требовалось на шаблоне, выполнив нужные соединения, получить его шаблонную В-схему.

Иначе обстоит дело в следующем разделе, где отображение задано фрагментарно.

Синтез устройств по фрагментам отображений

Мы не будем описывать *алгоритм синтеза абстрактных вычислительных устройств (АВУ) по фрагментам отображений*, а лишь приведем на примере результат его работы. В дальнейшем вместо «алгоритм синтеза» будем говорить просто «алгоритм».

Обозначим символом ${}_p A_q$ абстрактное вычислительное устройство (АВУ) с предвидением на p шагов и с последствием на q шагов, т.е. устройство, функционирование которого описывается уравнениями:

$$\begin{cases} \bar{q}_{t+1} = \theta \bar{q}_t x_{t+p} \dots x_{t+1} x_t x_{t-1} \dots x_{t-q}, \\ y_t = \sigma \bar{q}_t x_{t+p} \dots x_{t+1} x_t x_{t-1} \dots x_{t-q}, \end{cases} \quad (t \in Z = \{\dots 2, 1, 0, -1, -2, \dots\}) \quad (2)$$

где $x \in Z_k = \{0, 1, \dots, k-1\}$, $y \in Z_l = \{0, 1, \dots, l-1\}$ — множества цифр в k -ичной и l -ичной системах счисления, соответственно; $k, l \geq 2$, натуральные; $p, q \geq 0$. Символом ${}_p A_q$ обозначим *множество* всех АВУ с предвидением на p шагов и последствием на q шагов: ${}_p A_q = \{{}_p A_q\}$. Множество всех АВУ есть множество $\mathcal{A} = \bigcup_p \bigcup_q {}_p A_q$. Абстрактные автоматы образуют множество ${}_0 A_0$.

Фрагментом отображения называется множество соответствий типа $\bar{x} \rightarrow \bar{y}$, где $\bar{x}$ берется из области определения отображения, а $\bar{y}$ — из области значений. Количество соответствий во фрагменте может быть любым, кроме нуля.

Фрагмент отображения называется *репрезентативным*, если он позволяет полностью найти отображение.

Пусть числовой объект $\bar{x} = 1300122203 \underset{0}{1}$ записан в четверичной системе счисления. Это значит, что все его цифры принадлежат множеству цифр $Z_{(4)} = \{0, 1, 2, 3\}$. Очевидно, что номера разрядов растут справа налево, начиная с нулевого разряда, отмеченного нулем снизу. Пусть, аналогично, объект $\bar{y} = 4403212103 \underset{0}{4}$ записан в пятеричной системе счисления, $Z_{(5)} = \{0, 1, 2, 3, 4\}$, и пусть отображение таково, что между объектами $\bar{x}$ и $\bar{y}$ имеет место соответствие, которое мы запишем на разрядной сетке:

$$\begin{array}{cccccccccccc|c}
 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & t \\
 \hline
 1 & 3 & 0 & 0 & 1 & 2 & 2 & 2 & 0 & 3 & 1 & x_t \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 4 & 4 & 0 & 3 & 2 & 1 & 2 & 1 & 0 & 3 & 4 & y_t
 \end{array} \quad (3)$$

Спрашивается, существуют ли АВУ из класса ${}_p\mathcal{A}_q$, ($p, q \geq 0$), реализующие соответствие числовых объектов $\bar{x} \rightarrow \bar{y}$?

Алгоритм начинает поиск АВУ с класса ${}_0\mathcal{A}_0$, т.е. вначале ставится вопрос, существует ли абстрактный автомат, реализующий указанное соответствие (напомним, что во множестве всех АВУ абстрактные автоматы заполняют класс ${}_0\mathcal{A}_0$).

В результате применения алгоритма выясняется, что существуют два минимальных автомата A_1 и A_2 с двумя состояниями, для которых соответствие (3) имеет место. Эти автоматы задаются таблицами:

$$\begin{array}{l}
 A_1: \quad \begin{array}{c|cc} & x & \\ \hline q \backslash & 0 & 1 \\ \hline 0 & 0, 0 & 0, 3 \\ 1 & 1, 4 & 1, 2 \\ 2 & 1, 1 & 0, 2 \\ 3 & 0, 4 & 0, 3 \end{array} \quad \quad \quad A_2: \quad \begin{array}{c|cc} & x & \\ \hline q \backslash & 0 & 1 \\ \hline 0 & 1, 3 & 0, 0 \\ 1 & 1, 4 & 0, 2 \\ 2 & 1, 1 & 0, 2 \\ 3 & 0, 4 & 1, 3 \end{array}
 \end{array} \quad (4)$$

В таблицах $x \in Z_{(4)} = \{0, 1, 2, 3\}$, q — состояние автомата, $\bar{q} \in Q = \{0, 1\}$, Q — множество состояний автомата. На пересечении строки- x и столбца- q стоит пара (θ, σ) , где $\theta = \theta(q, x)$ — значение функции переходов на паре (q, x) , а $\sigma = \sigma(q, x)$ — значение функции выходов на паре (q, x) .

Таким образом, согласно алгоритму существует два автоматных отображения f_1 и f_2 , реализуемых, соответственно, автоматами A_1 и A_2 , для каждого из которых имеет место, в частности, данное соответствие $\bar{x} \rightarrow \bar{y}$.

Проверим, что это действительно так. Для этого пропустим объект $\bar{x}$ через автоматы A_1 и A_2 и убедимся, что в обоих случаях на выходе будет $\bar{y}$. Изобразим развертки во времени вычисления автоматами при прохождении через них объекта $\bar{x}$.

Через A_1 :

$$\begin{array}{cccccccccccc|c}
 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & t \\
 \hline
 & 1 & 3 & 0 & 0 & 1 & 2 & 2 & 2 & 0 & 3 & 1 & x_t \\
 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & q_t \\
 & 4 & 4 & 0 & 3 & 2 & 1 & 2 & 1 & 0 & 3 & 4 & y_t
 \end{array} \quad (5)$$

Через A_2 :

11	10	9	8	7	6	5	4	3	2	1	0	t
	1	3	0	0	1	2	2	2	0	3	1	x_t
1	0	0	1	0	1	0	1	0	1	1	0	q_t
4	4	0	3	2	1	2	1	0	3	4		y_t

(6)

Символ t обозначает автоматное время и одновременно для строк x_t и y_t номер разряда; начальное состояние автоматов в обоих случаях $q = 0$.

Результаты работы автоматов прочитываются в нижних строках разверток (5) и (6), в обоих случаях совпадают и оказываются равными объекту $\bar{y}$, как и должно быть.

Тот факт, что алгоритм выделил в классе ${}_0\mathcal{A}_0$ два автомата A_1 и A_2 , означает, что информации, заложенной во фрагменте соответствия $\bar{x} \rightarrow \bar{y}$, недостаточно для однозначного выделения автомата. Чтобы однозначно найти автомат в классе ${}_0\mathcal{A}_0$, дополним соответствие $\bar{x} \rightarrow \bar{y}$ в 11-ом разряде и перейдем к новому соответствию:

11	10	9	8	7	6	5	4	3	2	1	0	t
0	1	3	0	0	1	2	2	2	0	3	1	x_t
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
3	4	4	0	3	2	1	2	1	0	3	4	y_t

(7)

Пропустим объект $\bar{x}_1 = 01300122203 \underset{0}{1}$ через автоматы A_1 и A_2 и посмотрим, что получится на их выходах. При прохождении $\bar{x}_1$ через A_1 развертка (5) пополнится лишь 12-ым тактом:

12	11	10	...		...	1	0	t
	0	1				3	1	x_t
0	1	0				1	0	q_t
3	4					3	4	y_t

(8)

Как видно, в 11-ом такте, в полном соответствии с (7), получился результат $y = 3$, т.е. автомат A_1 реализует соответствие $\bar{x}_1 \rightarrow \bar{y}_1$, где $\bar{y}_1 = 34403212103 \underset{0}{4}$.

При прохождении $\bar{x}_1$ через A_2 будем иметь аналогично:

12	11	10	...		...	1	0	t
	0	1				3	1	x_t
0	1	0				1	0	q_t
0	4					4	4	y_t

(9)

Но на этот раз при $t = 11$ автомат A_2 вырабатывает сигнал $y'_{11} = 0$, отличный от требуемой цифры 3:

$$\bar{x}_1 \xrightarrow{A_2} \bar{y}'_1 \neq \bar{y}_1 \quad (10)$$

Схема (10) констатирует, что автомат A_2 не реализует соответствие $\bar{x}_1 \rightarrow \bar{y}_1$, и, таким образом, существует лишь один автомат, A_1 (рис. 4), реализующий соответствие $\bar{x}_1 \rightarrow \bar{y}_1$, а значит, в предположении, что соответствие $\bar{x}_1 \rightarrow \bar{y}_1$ – репрезентативное, существует лишь одно автоматное отображение f_1 , для которого $\bar{x}_1 \rightarrow \bar{y}_1$. Так по одному соответствию можно найти все автоматное отображение.

Если бы мы при $t=11$ потребовали, скажем, соответствия цифр: $\begin{matrix} 0 & 0 & 0 \\ \downarrow & \downarrow & \downarrow \\ 1 & 2 & 4 \end{matrix}$ (а также еще некоторых), то мы не нашли бы ни одного автомата, реализующего соответствие $\bar{x} \rightarrow \bar{y}$ с

указанными соответствиями в 11-ом разряде. В таком случае алгоритм перешел бы к поиску АВУ в классах ${}_1A_0$, ${}_0A_1$ и т.д.

Факт построения автомата A_1 по соответствию $\bar{x}_1 \rightarrow \bar{y}_1$ можно интерпретировать как процесс обучения, с которым синтез A_1 по соответствию $\bar{x}_1 \rightarrow \bar{y}_1$ имеет большое сходство.

Действительно, в мозгу школьника при обучении его, скажем, операции сложения после выполнения им *конечного* числа примеров формируется нечто, позволяющее ему в дальнейшем решить *любой* пример на выполнение операции сложения.

Здесь мы имеем аналогию. Через внешний конструктор, в котором содержится алгоритм синтеза по фрагментам отображений, пропускается *конечное* число соответствий (в рассмотренном примере одно: $\bar{x}_1 \rightarrow \bar{y}_1$), и в результате формируется структура (автомат A_1), позволяющая найти *любое* другое соответствие автоматного отображения f_1 .

Конечно, мы не утверждаем, что в мозгу происходит буквально то же самое, мы лишь подчеркиваем аналогию. Более того, мы не будем стремиться моделировать природу, не только по незнанию того, как она это делает, но и потому, что количественного эффекта при достижении одной и той же цели мы чаще всего добиваемся на путях, отличных от тех, которым следует природа.

Рис. 4. Каноническая В-схема автомата A_1 (слева) и ее двойник (справа), выполненный на шаблоне заготовки

Всего на такого типа шаблоне заготовки возможно 2^{234} различных вариантов соединений. Показан один из них.

Эффект снижения роли программ в В-компьютерах

Рассмотрим типичный линейный кусок программы, исполняемый на А-компьютере (см. рис. 5). Программа содержит блоки, в которых происходят вычисления функций. Вычисление происходит программно. После завершения вычисления в блоке происходит передача управления в следующий блок, в котором происходит аналогичное. Обращаем внимание на то, что в каждом блоке вычисление происходит *программно*.

Рис. 5. Линейная часть программы А-компьютера

Сравним эту ситуацию с аналогичной в В-компьютере. Изобразим эту же часть программы, но выполненную В-компьютером. Отличие состоит в том, что вычисление в блоках происходит не

программно, а с помощью устройств.

Рис. 6. Та же линейная часть программы, но выполненная В-компьютером

Сразу же обратим внимание на то, что эта часть программы будет выполнена В-компьютером быстрее, чем А-компьютером, поскольку вычисление функции устройством происходит быстрее, чем программное вычисление. Но не только на это обстоятельство обратим внимание, но также и на то, что значительная часть программной составляющей исчезла. Это именно то, что мы называли снижением роли программ в В-компьютерах. В перспективе можно вести речь о почти полном нивелировании роли программ в В-компьютерах. А что останется? Останется та часть программ, которая связана с управленческими функциями, то, что на рисунках обозначено как «передача управления». Но и это в перспективе может быть реализовано аппаратно, с помощью устройств.

Заключение

Приведенный краткий анализ достаточно убедительно показывает, что В-компьютеры имеют возможность неограниченно наращивать свою «железную массу», но не за счет однотипных ящиков, нагромождаемых в пределах определенного объема, а за счет синтеза разнообразных вычислительных устройств и последующей их интеграции в компьютер. Разнообразие вычислительных устройств, интегрированных в В-компьютер, свидетельствует о «развитости» В-компьютера, о его способности решать самые разные и сложные вычислительные задачи. Приведенные примеры показывают, что синтез вычислительных устройств, встраиваемых в В-компьютер, может быть осуществлен самим компьютером внутри самого себя, поскольку дело сводится к простановке точек на шаблонах заготовок — простейшей из всех возможных операций. Решение задачи о создании искусственного интеллекта может быть получено путем использования В-компьютеров, поскольку именно В-компьютеры позволяют избавиться от влияния человеческого фактора, проявляющего себя через программы, влияние которых в В-компьютерах имеют тенденцию к нивелированию.

ЛИТЕРАТУРА

1. Деев Г. Е., Ермаков С. В., Король Н. А., Перегуда А. И., Старков С. О. Компьютеры, ведущие вычисления в k -ичной системе счисления. *Супервычисления и математическое моделирование*. XVI Международная конференция. 3 – 7 октября 2016 года, г. Саров.
2. Деев Г. Е., Ламков А. В. *Абстрактные вычислительные устройства*. В 2 т. Т. 1. М.: Энергоатомиздат; 2004. 548 с.
3. Деев Г. Е. *Абстрактные вычислительные устройства*. В 2 т. Т. 2. М.: Энергоатомиздат; 2007. 332 с.
4. Деев Г. Е. *Теория вычислительных устройств*. Санкт-Петербург: Лань; 2019. 452 с.
5. Минский М. *Вычисления и автоматы*. М.: Мир; 1971. 362 с.
6. Rich E. *Artificial Intelligence*. New York: McGraw-Hill; 1983. 411 p.
7. Эндрю А. *Искусственный интеллект*. М.: Мир; 1985. 264 с.
8. *Вычислительные машины и мышление* / Под ред. Э. Фейгенбаума и Дж. Фельдмана. М.: Мир; 1967. 552 с.
9. Тьюринг А. *Может ли машина мыслить?* М.: Физматгиз; 1960. 112 с.
10. Рафаэл Б. *Думающий компьютер*. М.: Мир; 1979. 408 с.
11. Дрейфус Х. *Чего не могут вычислительные машины: критика искусственного разума*. М.: Прогресс; 1978. 336 с.
12. Тимофеев А. В. *Роботы и искусственный интеллект*. М.: Наука; 1978. 192 с.