Демидов А.А., 2006

alex.demidov@mail.ru

Использование коллективов автоматов в построении активных интеллектуальных систем

(Теория интеллектуальных систем)

Какова должна быть степень формализации в биологии при изучении живых систем? Если взять для примера квантовую механику, то в её становлении можно выделить два этапа. Первый этап, когда Нильс Бор создал философию квантовой механики. Формул ещё нет, а если они и есть, то не совсем такие, как нужны, или совсем не такие. Второй этап – бурный расцвет, превращение в строгую область физики с большим количеством строгих формул. Но этот этап – всё же второй, он возможен лишь после первого этапа. Так вот, в биологии ещё не наступил первый этап.

И.Гельфанд [1]

Содержание

Даётся определение интеллекта и интеллектуальной системы, доказывается его непротиворечивость. Исследуются механизмы возникновения активности и интеллектуальности в поведении системы. Рассматриваются перспективы применения моделей коллективного поведения автоматов для изучения интеллектуальных систем в сравнении с моделями нейронных сетей на формальных нейронах. Предлагается иное определение алгоритма, доказывается его несводимость к машине Тьюринга. В новом формализме анализируется проблема алгоритмической неразрешимости и даётся иной взгляд на теоремы Гёделя о неполноте и на теорию множеств в целом; рассматривается гипотеза о топологической сути парадоксов квантовой физики.

Оглавление

Определения

Интеллектом будем называть модифицирующий себя алгоритм, способный на основе входных данных вычислять предсказуемый результат способом, не заложенным в исходный вариант алгоритма.

Алгоритм здесь понимается шире, нежели в принятых определениях алгоритма на основе машины Тьюринга или ассоциативного исчисления Маркова. Для дальнейшего рассмотрения будет необходимо определение ***алгоритма как конфигурации*** области некоторого пространства – вычислительной среды данного алгоритма. Модификация алгоритма означает изменение конфигурации данной области пространства, а процесс вычисления представляет собой взаимное изменение свойств самого алгоритма и некоторого объекта – сигнала при его прохождении через эту область пространства. Модификация как алгоритма, так и сигнала возможна только в процессе вычисления. Эти понятия будут уточнены в разделе «Алгоритм как конфигурация».

Системой будем называть вычислитель с заданным на нём алгоритмом. Или иными словами – произвольно выбираемую для рассмотрения область вычислительного пространства с наложенной на неё конфигурацией алгоритма, то есть сконфигурированную алгоритмом.

Интеллектуальной системой является «система с интеллектом», то есть вычислитель с заданным на нём самомодифицирующимся алгоритмом, способным вычислять предсказуемый результат. Системами искусственного интеллекта (системами ИИ) являются интеллектуальные системы, созданные человеком.

Предсказуемость результата подразумевает его неслучайность. Это означает, что аналогичный результат при тех же исходных данных вычисляется некоторым подмножеством нетождественных систем (не обязательно интеллектуальных) с высокой степенью вероятности. В качестве такого референтного подмножества могут рассматриваться версии самой системы во времени (с различными модификациями алгоритма), а также конкретная версия системы на различных входных данных – алгоритм можно назвать предсказуемым, если он преобразует кусочно-гладкую функцию входных данных в кусочно-гладкую функцию выходных данных, при условии, что функция выходных данных зависит от входных данных, то есть её значение изменяется при изменении входных данных.

Самомодификация и изначальная недетерминированность способа вычисления – суть одно и тоже требование, позволяющее избежать рекурсивности определения: если бы алгоритм вычисления (конфигурация области пространства) был детерминирован заранее, значит в силу теоремы Т. 1 он был задан некоторой сторонней интеллектуальной системой, которую мы и пытаемся определить.

Т. 1

Теорема: Алгоритм интеллектуальной системы (интеллект) может быть вычислен только этой или другой интеллектуальной системой.

Доказательство: Действительно, если исключать влияние некоего «демона» как исчезающе-малой вероятности, следовательно, конфигурация (алгоритм) рассматриваемой системы была задана какой-то другой системой, необязательно интеллектуальной. Однако, тогда встаёт вопрос – кто задал конфигурацию той другой системы? Единственный вариант, при котором исключается влияние «демона» – это объединение всех участвующих в конфигурировании систем в кольцо, когда конфигурация каждой системы задаётся следующей системой, а конфигурация последней системы вычисляется первой системой. Назовём это кольцо конфигурирующей системой.

Поскольку мы договорились исключать абсолютно случайное возникновение алгоритма интеллектуальной системы, то алгоритм конфигурирующей системы должен также быть в значительной степени детерминированным, неслучайным. Следовательно он не должен существенно зависеть от начальной конфигурации вычислительного пространства, которая никем не задаётся, то есть является случайной. Поэтому алгоритм конфигурирующей системы должен быть «эргодическим» (по аналогии с эргодической цепью Маркова) в том смысле, что бесконечная в пределе суперпозиция (самоприменение) данного алгоритма должна вычислять результат, независимый от начального варианта алгоритма, тогда конечная суперпозиция будет стремиться к данному результату с некоторой точностью.

Но такая конфигурирующая система после определённого времени её эволюции будет являться интеллектуальной по определению, причём она вычисляет и свой собственный алгоритм. **Теорема доказана.**

Введение

Целью данной работы изначально являлась потребность дать интеллектуальной системе определение. Вернее – показать, что такое определение можно дать в принципе, и на его основе строить логически обоснованные выводы. Можно считать, что всё, что я хотел сказать изначально, я уже сказал в разделе «Определения». И даже если всё последующее изложение будет признано не представляющим никакого интереса, всё равно эта работа будет иметь ценность – она разрывает тот порочный круг, когда мы пытаемся изучать и имитировать интеллект, не определяя, что он из себя представляет.

Ни понятия алгоритма, ни теорем, ни вычислительного пространства, ничего этого вообще не было в первоначальном замысле. И только когда я всё-таки собрался изложить свои мысли «на бумаге», все эти вещи возникли как необходимые и неотъемлемые части единой мозаики. Всё, что планировалось изначально – это каким-то образом определить интеллектуальную систему и показать, что применение нейронных сетей для её имитации неоправданно, ибо существует огромная масса биологических данных, не укладывающихся в нейросетевые модели.

Соответственно, первая часть работы как раз и является логическим продолжением первоначальной идеи и носит, в основном, описательный характер. В ней я покажу, что понятие активности в поведении системы относительно, и определяется исключительно той позицией, с которой мы рассматриваем эту систему – нельзя же превозносить систему только потому, что в ней имеется элементарный генератор сигнала! В какой-то мере это относится и к интеллектуальной системе – ничего не изменится, если сигнал в её обратных связях будет угасать, а потом возобновляться при следующем прохождении внешнего сигнала. Тем не менее, наличие самих обратных связей, причём сильных, необходимо.

Также в этой части я дам описание взаимодействия систем и покажу, что деление на системы в общем-то условно, что система возникает лишь в нашем сознании для группировки объектов реального мира. Затем я покажу, как из неинтеллектуальных систем возникает система интеллектуальная – это будет логически выведено из самого определения интеллектуальной системы. Далее будет доказано, что основные нейросетевые модели неспособны описать интеллектуальную систему, и что другая модель – коллективного поведения автоматов – может с этим прекрасно справиться.

Вторая часть работы, начинающаяся разделом «Алгоритм как конфигурация», значительно более сложна и спорна (однако, и значительно более интересна). В ней даётся математическое обоснование модели, и устанавливаются параллели с некоторыми физическими теориями. Там же даётся другая трактовка теорем Гёделя о неполноте – на определённом этапе своих рассуждений о вычислительном пространстве я вдруг обнаружил, что они противоречат этим теоремам. Соответственно, некоторые поползновения в сферу интересов физики явились следствием этих же рассуждений.

Вообще-то я сомневался, стоит ли включать всё это в текст, чтобы не компрометировать основное изложение, но всё-таки результаты мне показались очень интересными и достойными, по крайней мере, конструктивной критики. Поэтому я заранее прошу специалистов в данных областях отнестись к этим выводам с нисхождением, если в них содержатся очевидные ошибки, и не распространять негативное мнение на всю работу в целом. Я спешил поскорее завершить изложение, чтобы иметь возможность получить отзывы на эту работу, а времени, которое я мог выделить на неё, катастрофически не хватало. Поэтому, если данная работа в какой-то своей части действительно представляет интерес, я был бы в огромной степени признателен любым конструктивным отзывам на неё, в том числе и негативным.

В данной работе я никак не затрагиваю высокоуровневые системы ИИ, основанные на явных правилах обработки структурированной информации, поскольку такой подход хотя и результативен, но системы абсолютно не гибки в обучении. Использование моделей высокоуровневого ИИ мало что может дать в понимании биологических систем, так же как и в построении сложных самообучающихся систем ИИ. А первостепенной задачей любой модели является именно изучение с её помощью реальных явлений, а уж только потом – их имитация.

Данная работа в моём представлении – лишь начало, повод к диалогу со специалистами в различных областях математики, биологии, физики и искусственного интеллекта. Приводимые доказательства похоже верны, но часто недостаточно формальны и нуждаются в дальнейшей проработке. Однако, я надеюсь на то, что данная работа будет интересной в той степени, чтобы получить дальнейшее развитие.

Модель активной системы

Изобразим модель функционирования любой системы (в общем понимании этого термина) в её окружении. Рассматриваемую систему принято называть управляющей, а её окружение – управляемой системой. Под **сигналом** в данной модели понимается любое воздействие одной системы на другую.

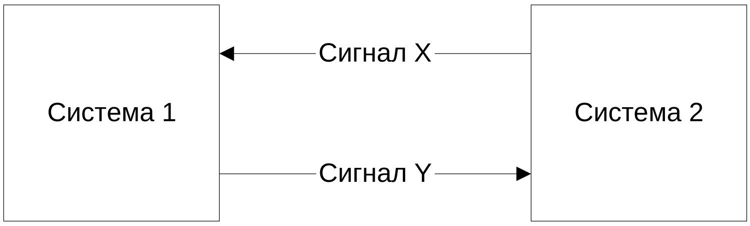

Рис. 1

Если рассматриваемая система является управляющей «Системой 1», то «Система 2» будет по отношению к ней управляемой. Однако, справедливо и обратное – «Система 2» будет управляющей по отношению к «Системе 1», если смотреть с другой точки зрения. Выбор управляющей и управляемой системы – лишь вопрос удобства рассмотрения.

Этот «принцип относительности» имеет и более глубокое следствие, принципиально важное в дальнейшем рассмотрении – ответный сигнал системы можно рассматривать либо как функцию входных сигналов: $\mathbf{Y = F_1(X)}$, либо как активное действие управляющей системы на управляемую, влекущее изменение управляемой системы, о чём сигнализирует последующее изменение сигнала на входе управляющей системы: $\mathbf{X = F_2(Y)}$. При этом сами функции $\mathbf{F_1}$ и $\mathbf{F_2}$ могут изменяться сигналами, что часто учитывается введением понятия внутренних **состояний** системы: $\mathbf{(Y, S_1) = F_1(X, S_1); (X, S_2) = F_2(X, S_2)}$, что эквивалентно записи: $\mathbf{(Y, F_1) = F_1(X); (X, F_2) = F_2(X)}$.

Отдельно следует рассмотреть случай, когда система не может сама породить «Сигнал Y» иначе, как на основе входящего «Сигнала X». Самостоятельное порождение сигнала возможно в двух случаях: если система имеет в своём составе некоторый генератор сигнала, либо если она содержит петли обратной связи, по которым непрерывно циркулирует сигнал. Однако, генератор сигнала не может действовать иначе, как на основе циркуляции сигнала, ибо конфигурация сама по себе не способна породить сигнал без процесса вычисления. Соответственно, более строго – система не способна породить сигнал, если не имеет в своём составе обратных связей с постоянной циркуляцией сигнала.

Такие системы без циркуляции сигнала, тем не менее, могут условно рассматриваться как активные – несмотря на то, что они функционируют лишь в момент прохождения сквозь них внешнего сигнала, однако при постоянной циркуляции сигналов по пути «Сигнал X» – «Система 1» – «Сигнал Y» – «Система 2» поведение таких систем будет неотличимо от поведения систем с внутренней циркуляцией сигнала – процесс вычисления будет непрерывным.

Необходимо отметить, что наличие в составе системы обратных связей является необходимым, но не достаточным признаком интеллектуальной системы. Действительно, модификация алгоритма (конфигурации подпространства) возможна только в процессе вычисления при прохождении сигнала через данную конфигурацию. Следовательно, наличие обратных связей есть необходимое условие самомодификации и существования интеллектуальной системы. С другой стороны, процесс вычисления может и не приводить к изменению конфигурации данного подпространства, следовательно, признак не является достаточным.

Соответственно, ***любую систему можно рассматривать*** одновременно и как пассивную функцию внешних и внутренних переменных, и ***как активно действующий субъект***. Этот простой, но фундаментальный вывод лежит в основе всех дальнейших рассуждений.

Взаимодействие систем

Если системы взаимодействуют друг с другом, то говорят, что они **связаны** друг с другом. Если следовать определению системы как сконфигурированной алгоритмом области вычислительного пространства, то взаимодействие систем представимо в виде конкурентного изменения конфигурации их окружения:

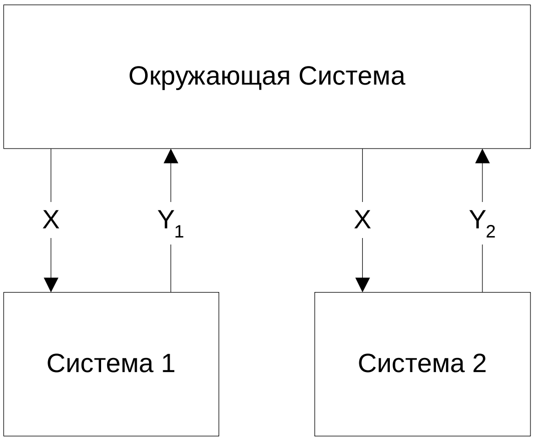

Рис. 2

Таким образом, алгоритм «Системы 1» и алгоритм «Системы 2» одновременно меняют конфигурацию (то есть алгоритм – по определению) «Окружающей системы» с тем, чтобы достичь своих результатов. Изменение конфигурации происходит только посредством сигналов в процессе вычисления. Как было определено выше, сигналы – это любые воздействия одной системы на другую: $\mathbf{X = F(Y_1 + Y_2)}$.

В силу введённого «принципа относительности» с точки зрения «Окружающей системы» такое влияние других систем будет эквивалентно ответам этих систем на действия самой же «Окружающей системы»: $\mathbf{(Y_1, Y_2) = F_1(X) + F_2(X)}$. При вырожденности «Окружающей системы» схема, изображённая на Рис. 2, переходит в схему на Рис. 1, то есть две системы будут непосредственно влиять друг на друга.

Рассмотрим теперь ситуацию с точки зрения «Системы 1». Для неё «Окружающая система» и «Система 2» в целом будут являться окружением, то есть для «Системы 1» нет никакой разницы – различаем ли мы логические подсистемы в её окружении или нет. Действительно, по определению системы – это свободно выбираемая область вычислительного пространства, следовательно, деление на подсистемы носит произвольный характер и не может влиять на свойства самого пространства.

Таким образом, любая система может быть произвольным образом разбита на составляющие подсистемы, причём каждая из них может рассматриваться изолированно, при этом ***все остальные подсистемы, связанные с данной, будут являться для неё частью окружения***. Это второй фундаментальный вывод, который будет использован в дальнейшем рассмотрении.

Рассмотрим также следующую схему взаимодействия систем, которая нам пригодится в дальнейшем.

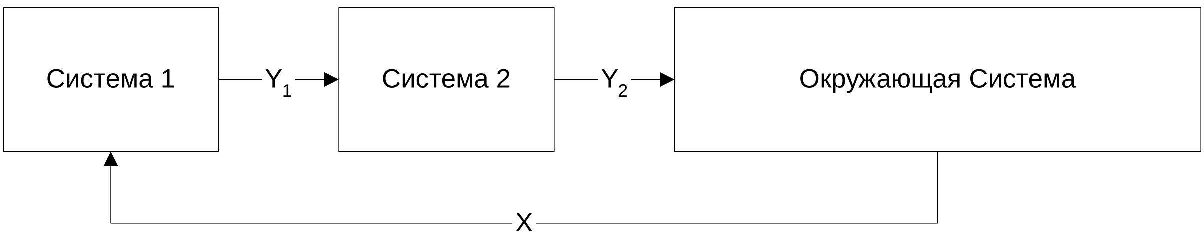

Рис. 3

Здесь системы связаны последовательно, в отличие от параллельного их соединения на Рис. 2. В таком случае алгоритм «Системы 1» меняет алгоритм «Системы 2», которая в свою очередь меняет алгоритм «Окружающей системы». Также можно рассуждать и наоборот – что это «Окружающая система» меняет алгоритм «Системы 1». Заметим, что при такой связи нам уже проще воспринимать наш «принцип относительности», ибо системы соединены в кольцо и визуально равноправны.

Последовательное взаимодействие систем будет описываться другим уравнением, нежели параллельное: $\mathbf{Y_2 = F_2(F_1(X))}$, где $\mathbf{F_1(X) = Y_1}$. Или наоборот: $\mathbf{X = F(F_2(Y_1))}$. Таким образом, влияние «Системы 1» и «Системы 2» на «Окружающую систему» не суммируется, как при параллельном соединении, а происходит его суперпозиция – модуляция влияния «Системы 1» алгоритмом «Системы 2».

Важно отметить тот факт, что при прочих равных условиях ***последовательная связь обладает меньшей «пропускной способностью» нежели параллельная***, так как в этом случае мы имеем только один сигнал вместо двух, ***но она характеризуется большим «динамическим диапазоном»***, поскольку суперпозиция является более кардинальным преобразованием, нежели простое сложение. Любые связи нескольких систем друг с другом можно свести к последовательным и параллельным.

Возникновение интеллекта

Вернёмся к Рис. 1. Пусть «Система 1» и «Система 2» не являются интеллектуальными, то есть не меняют свой собственный алгоритм, пусть тем не менее, «Система 1» меняет алгоритм «Системы 2», иными словами – пусть функция «Системы 2» зависит от входного «Сигнала Y»: $\mathbf{(X, F_2) = F_2(Y)}$. Пусть также существует некоторое подмножество систем, подобных «Системе 1» и «Системе 2», попарно взаимодействующих между собой, но никак не влияющих на рассматриваемые системы.

Тогда, полная система, состоящая из двух неинтеллектуальных систем: «Системы 1» и «Системы 2», будет являться интеллектуальной по определению – она не взаимодействует ни с какими окружающими системами, тем не менее меняет свой собственный алгоритм!

Таким образом, ***правило произвольного разбиения на подсистемы для интеллектуальной системы неприменимо*** в том смысле, что при разбиении на подсистемы свойство интеллектуальности может не сохраняться. Это третий фундаментальный вывод, который будет использоваться в дальнейшем.

Остановимся на причинах возникновения интеллекта у системы более подробно. **В-первых**, очевидно, что если бы «Система 2» не меняла свой алгоритм (конфигурацию) под воздействием «Системы 1», то интеллектуальной системы при объединении не получилось – при объединении двух статичных конфигураций получилась бы также статичная конфигурация. Следовательно, необходимым условием возникновения интеллекта является восприимчивость системы к проходящему через неё сигналу, то есть изменчивость функции **F**. Такие системы с изменяемой конфигурацией будем называть **динамическими системами** или системами с памятью. Для описания поведения динамических систем используется язык дифференциальных уравнений, определяющий следующее состояние системы через её предыдущее состояние. Как мы увидим в теореме Т. 9 в разделе «Математика интеллекта», эти динамические системы должны также быть нелинейными.

Во-вторых, необходимым условием интеллектуальности является наличие сильных обратных связей внутри системы. Действительно, как было определено, изменение конфигурации вычислительного пространства происходит только посредством сигналов в процессе вычисления. Соответственно, чтобы система имела возможность менять свой собственный алгоритм, в пределах области вычислительного пространства, занимаемого системой, какое-то время должен присутствовать её собственный сигнал, иначе система не будет являться интеллектуальной по определению. При этом не требуется, чтобы такой сигнал циркулировал в системе постоянно – как мы убедились в разделе «Модель активной системы», нет никакой разницы между порождённым и наведённым сигналом. Однако важно, чтобы сила обратных связей была достаточной для компенсации дестабилизирующего эффекта внешних воздействий. Иначе нарушается «эргодичность» алгоритма данной системы, установленная в ходе доказательства теоремы Т. 1, и в этом случае нельзя будет говорить о том, что система меняет свой алгоритм сама – его в значительно большей степени будет менять какая-то внешняя система.

В-третьих, необходимым условием интеллекта является неслучайность, определённая устойчивость результата, признаком чего служит наличие других систем, демонстрирующих поведение, аналогичное поведению рассматриваемой системы. Иными словами, понятие интеллекта в определённой степени является относительным – оно определяется путём сопоставления поведения различных систем или эволюции одной и той же системы во времени. Когда один человек говорит ерунду, его порой признают сумасшедшим, если же его взгляды разделяют многие или они подкреплены доказательством – гением. Инертность динамической системы относительно внешних воздействий определяется отрицательными, а лабильность относительно внутреннего сигнала – положительными обратными связями [[2]], поэтому важен баланс внутренних положительных и замыкаемых внешним окружением отрицательных обратных связей. В дальнейшем рассмотрении мы всегда будем подразумевать наличие референтного подмножества систем, демонстрирующих аналогичное поведение, если не будет оговорено иное.

Совокупность этих трёх условий является также и достаточным условием возникновения интеллекта детерминированных (не вероятностных) систем, поскольку мы не делали никаких дополнительных допущений при рассмотрении модели, а добавление в неё новых систем не может изменить ситуации (при условии аналогичных действий на всём подмножестве подобных систем) – как было показано, дополнительные системы будут являться частью окружения как с точки зрения «Системы 1», так и с точки зрения «Системы 2».

Однако, добавление в систему влияющих на неё стохастических (вероятностных) систем не гарантирует сохранение интеллекта в объединённой системе – можно представить случай, когда поведение полной системы при таком добавлении станет случайным и не предсказуемым никакой другой системой (например, когда шум полностью искажает сигнал системы). Однако, представить какой-либо критерий, по которому можно было бы судить – потеряет ли система интеллект или нет, не представляется возможным, поэтому в случае добавления в интеллектуальную систему стохастических систем исследование предсказуемости поведения объединённой системы необходимо проводить отдельно. В дальнейшем мы будем рассматривать только детерминированные системы, если не будет оговорено иное.

Таким образом, ***интеллектуальная система возникает на определённом этапе при объединении взаимовлияющих неинтеллектуальных систем в том случае, когда среди объединяемых присутствуют нелинейные динамические системы и в объединённой системе появляются сильные обратные связи, проходящие через эти системы***. Возникнув однажды, интеллект системы сохраняется при добавлении в неё любого числа взаимосвязанных детерминированных систем. При разрыве обратных связей интеллект системы исчезает.

Ограничения моделей нейронных сетей

Теория нейронных сетей строится на понятии формального нейрона как пассивного нелинейного сумматора входных сигналов. Формальные нейроны могут объединяться в сети как с обратными связями, так и без обратных связей – персептроны. Для нейронной сети задаётся тот или иной способ обучения, заключающийся в изменении весов связей (синапсов) между формальными нейронами.

Способы обучения подразделяются на два основных вида: обучение с учителем, когда некоторая сторонняя система сообщает нейронной сети «верность» или «ошибочность» полученного результата, детерминизируя тем самым поведение нейронной сети в соответствии со своим, и обучение без учителя, когда сеть обучается самостоятельно, хотя в этом случае она сможет выступать лишь кластеризатором входных сигналов по некоторым признакам – по сути той же цели служат статистические методы обработки данных.

Всё вышесказанное можно свести в таблицу, классифицирующую моделей нейронных сетей по типам связей и обучения (приводится на основе классификации, данной в [3]):

Тип связей \\ Тип обучения	**Без учителя**	**С учителем**
Без обратных связей	Соревновательные сети, карты Кохонена (сжатие данных, выделение признаков)	Многослойные персептроны (аппроксимация функций, классификация)
С обратными связями	Сеть Хопфилда (ассоциативная память, кластеризация данных, оптимизация)	Рекуррентные аппроксиматоры (предсказание временных рядов, обучение в режиме on-line)

Таблица 1

Рассмотрим нейросетевые модели с точки зрения модели активной интеллектуальной системы. Можно выявить следующие основные ограничения, сдерживающие, на мой взгляд, развитие в сфере изучения интеллектуальных систем.

Во-первых, принципиальным ограничением нейросетевых моделей является определение формального нейрона как базового элемента нейронной сети. Между тем, формальный нейрон не только не является интеллектуальной системой, но даже не является системой динамической! Кроме того, он может рассматриваться как активная система лишь условно, так как не содержит в себе обратных связей, что влечёт формирование одностороннего взгляда – он понимается как пассивный объект $\mathbf{Y = F_1(X)}$ (см. схему на Рис. 1), второй же вариант, когда нейрон понимается как активно действующий субъект $\mathbf{X = F_2(Y)}$, упускается из вида.

Во-вторых, не все модели нейронных сетей являются интеллектуальными. Так, сети без учителя и обратных связей вообще никогда не демонстрируют интеллекта (представляется, что обратные связи в них всё же присутствуют между соседними нейронами при изменении весов, тогда для них в точности справедливы замечания, высказываемые в адрес сетей Хопфилда). Персептроны имеют обратные связи только в процессе обучения (они замыкаются системой учителя), и без учителя немедленно теряют интеллект. Но если мы уже имеем постоянно задействованную обучающую систему, дающую верные результаты, зачем в таком случае нам нужна нейронная сеть? А ведь персептроны являются наиболее распространённой, в определённом смысле – типовой, моделью нейронной сети!

Далее, сети Хопфилда (сети с обратными связями и без учителя) уже являются интеллектуальными, однако они «имеют» своё собственное понятие цели поведения, которое мы не в силах изменить и можем только использовать в той мере, в которой оно совпадает с целями наших задач. И только рекуррентные аппроксиматоры – сети с учителем и обратными связями являются и интеллектуальными, и управляемыми – они не теряют интеллект, хотя могут терять цель (то есть вычислять результат, предсказуемый другим, чем требуется, подмножеством систем) если их не мотивировать. Но математическая теория и применение сетей такого типа отнюдь не столь просты, как в случае персептрона, поэтому встаёт вопрос о целесообразности применения именно нейросетевой модели в сравнении с остальными.

В-третьих, модель формального нейрона вступает в противоречие с целым рядом наблюдаемых фактов деятельности реальных биологических интеллектуальных систем, таких как пейсмекерная активность, объёмная проводимость, непрямая химическая синаптическая передача, большое разнообразие медиаторов (см., например, [[4]]), поэтому учёт экспериментальных данных биологии в теории нейронных сетей существенно затруднён.

Все эти проблемы приводят к серьёзным логическим противоречиям при попытках применить нейронные сети к исследованию и имитации реальных биологических систем. Большинство нейросетевых моделей принципиально узки и неприменимы для представления интеллектуальных систем. Кроме того, понятие формального нейрона как базового элемента заводит в логический тупик, когда каждый элемент системы является пассивным, в то время как вся система в целом должна обладать активным целенаправленным интеллектуальным поведением по отношению к своему окружению. Именно поэтому применение нейросетевых моделей имеет значительный успех лишь в распознавании образов, когда система по сути является лишь функцией входных сигналов: $\mathbf{Y = F_1(X)}$. Второй же вариант, когда система рассматривается как активно действующий субъект $\mathbf{X = F_2(Y)}$ не может найти должного применения.

Преимущества моделей коллективов автоматов

Доказана формальная эквивалентность моделей нейронных сетей и гладких конечных автоматов [[5]], поэтому формально модель коллективного поведения автоматов не уступает по мощности модели нейронных сетей. Однако, в ходе данного доказательства рассматриваются только рекуррентные нейронные сети – то есть сети с обратными связями, для других же нейросетевых моделей такое доказательство неверно. Как мы уже показали, только рекуррентные нейронные сети и способны имитировать интеллектуальные системы, остальные нейросетевые модели для этого слишком узки. В настоящий момент существуют только две разумные альтернативы – рекуррентные сети или коллективы автоматов. Выбор подходящей модели – лишь вопрос удобства рассмотрения.

Как я покажу в этом разделе, описание интеллектуальной системы в модели коллективного поведения автоматов значительно превосходит по простоте, силе и выразительности модель рекуррентных сетей. В модели коллективного поведения базовым элементом является автомат, представимый некоторой функцией аргументов и состояний $\mathbf{(Y, S) = F(X, S)}$. По сути он является совокупностью многих функций $\mathbf{F_1(X)\ldots F_N(X)}$, а выбор конкретной функции определяется состоянием автомата. Состояние же, в свою очередь, зависит от предыстории автомата – аргументов, которые подавались ему в прошлом. При определённой конструкции автомат уже представляет собой элементарную интеллектуальную систему, поэтому декомпозиция общей системы не приводит к таким катастрофическим последствиям, когда понятие интеллекта и даже активности вдруг мистическим образом исчезает на каком-то из этапов такой декомпозиции.

Кроме того, этот подход как нельзя лучше соответствует нашим представлениям о реальных биологических и социальных системах, где каждая клетка или индивид понимается как независимый и активно действующий субъект, иными словами – более или менее сложный автомат. Поэтому появляется возможность использования модели коллективного поведения для изучения и имитации любых биологических систем – одноклеточных организмов, колоний микроорганизмов, простых организмов типа вольвокса или губок, нервной системы моллюсков и насекомых, колоний насекомых, мозга животных и человека, социальных систем.

Исследования на всех уровнях данной иерархии, будучи проводимыми с применением одной модели, способны в конце концов дать нам ключ к разгадке интеллекта, а в итоге – и человеческого разума. В данной работе я привожу необходимые общие условия возникновения интеллекта, однако биология способна дать нам конкретные сведения о строении реальных интеллектуальных систем, что несомненно ускорит наш прогресс в понимании и имитации таких систем.

Сама идея, лежащая в основе данной работы, имеет свои корни в биологии, в основном – в трёх статьях о поведении одноклеточных организмов [[6]], колоний микроорганизмов [[7]] и в частности – коллективных амёб [[8]]. Как оказалось, сходный подход уже рассматривался ранее и в нейробиологии [[9], [10]], но, к сожалению, он не нашёл широкого распространения, по моему мнению – в силу недостаточной формализации выводов, когда для функциональной системы взаимосодействующих компонентов постулировалась решающая роль результата как системообразующего фактора, но при этом ни понятие системы, ни механизм анализа сложного результата, ни способ коммуникации взаимосодействующих компонентов не имели строгого определения и объяснения. Отметив прогрессивность данного подхода, я не буду к нему апеллировать в дальнейшем изложении, ибо у нас обратная задача – не расширить, а сузить модель, отделив необходимые для интеллекта факты от сопутствующих артефактов. Отправным пунктом лучше сделать работы [Error: Reference source not found] и [Error: Reference source not found, Error: Reference source not found], соблюдая естественную последовательность событий.

Так, в первой части работы [Error: Reference source not found] рассматриваются выразительные примеры поведения одноклеточных, показывается их способность к ориентировке, анализу различного рода сигналов, пластичности поведения, но что более поразительно – к увязыванию различных сигналов между собой. Изначально индифферентный для организма раздражитель может приобрести такое же сигнальное значение, как и непосредственный биотический агент, если в течение некоторого времени подавать оба раздражителя вместе. То есть, уже одноклеточный организм способен различать несколько типов сигналов и образовывать некое примитивное подобие кратковременных «условных рефлексов».

Одноклеточные организмы оказываются также способны к обработке и более сложных сигналов. В той же работе приводятся описания экспериментов, в которых дафния обучалась проходить через изогнутую трубку при своём движении к свету. После того, как трубку убирали, зигзагообразность её движения сохранялась ещё в течение некоторого времени. Это возможно только в том случае, если дафния способна удержать в своей памяти достаточное количество бит информации – либо она должна запоминать яркость или угол падения света, при котором надо делать петлю, либо каким-то образом считать свои «шаги» до поворота. Кроме того, ей ещё нужно более или менее детально запомнить и саму траекторию петли – когда следует делать обратный разворот к свету.

Также очень интересным является описание пластичности поведения одноклеточных. Они способны различать сигналы одного и того же типа – стремясь к слабым раздражителям, но избегая сильных. Амёба демонстрирует свою способность привыкать к сигналам, подавляя реакцию на монотонно действующий раздражитель. Однако, она тут же оживляется при любом изменении этого сигнала. Все эти факты однозначно указывают на то, что уже на уровне клетки существует и память, и обработка информации. Поэтому, бездоказательно отбрасывать гипотезу активного участия нейрона в когнитивных процессах представляется неразумным.

Далее, в работах [Error: Reference source not found] и [Error: Reference source not found, с. 212] рассматриваются явления самоорганизации и дифференцировки в колониях микроорганизмов, вскрываются тончайшие сигнальные механизмы взаимодействия клеток в популяции. Одним из наиболее ярких примеров является объединение разобщённых одноклеточных амёб «Dictyostelium Discoideum» в многоклеточный мигрирующий псевдоплазмодий. Подчинённые волнообразному процессу распространения цАМФ, амёбы собираются в единый «суперорганизм» из десятка тысяч клеток, который трансформируется, передвигается и функционирует как единое целое, при том, что никаких других средств коммуникации, кроме химического сигнала, у него нет. Этот суперорганизм передвигается в поисках питательной среды быстрее каждой отдельной амёбы, и если так и не находит пищи, происходит дальнейшая трансформация – часть клеток погибает в результате апоптоза и образует ножку, а остальные взбираются по ней и образуют плодовое тело со спорами.

Социальные связи

Теперь обратимся к нейрону.

Модель коллективного поведения автоматов способна легко объяснить такие явления, как объёмное проведение сигналов в головном мозге, пейсмекерную активность нейронов, большое разнообразие медиаторов, и многие другие (если не все) известные факты строения и взаимодействия нервных клеток. Каждый нейрон в ней рассматривается как отдельный организм, и его поведение можно предсказывать на основе поведения одноклеточных организмов (эта точка зрения близка, но не эквивалентна и не основана на высказываемой в). Он обладает своей собственной памятью, анализирует входящие сигналы и проявляет активность, направленную на изменение окружающей его среды.

Нейрон, также как и одноклеточный организм, может анализировать различные виды входных сигналов, но если для одноклеточного таковыми являются химические, тактильные или световые сигналы, то для нейрона – практически только химические, именно поэтому и нужны различные медиаторы – чтобы сигнализировать о разных свойствах среды, окружающей нейрон. Несмотря на то, что один и тот же медиатор может выполнять различную роль в разных отделах мозга, каждый отдельный нейрон этого не ощущает в силу значительной локальности его рецептивной области, и для него играет роль само разнообразие входных сигналов.

Нейрон, как и одноклеточный организм, способен связывать эти сигналы между собой, образуя подобие условных рефлексов (о таких «рефлексах» у одноклеточных можно посмотреть в [Error: Reference source not found]). Для нейрона неважно, как пришёл к нему сигнал – через синапс или путём объёмной передачи, и более того – нейрон может сам проявлять активность, если его не устраивает, например, отсутствие каких-либо сигналов от окружающей его среды, в том числе гормональных сигналов организма и сигналов от других нейронов.

Тренировка синапсов в такой модели может рассматриваться наряду с системой регулирования кровотока как система минимизации расхода энергии – она может быть эквивалентна тренировке мышц атлета и быть направленной на минимизацию энергии, расходуемой нейроном – тот же результат достигается меньшими усилиями. Однако многие факты говорят в пользу того, что синапсы всё же принимают участие в образовании памяти, поэтому, скорее всего, они реализуют обе функции. Тем не менее, явление нейрональной пластичности и самой возможности быстрой обработки информации можно объяснить только наличием памяти у самого нейрона.

Потенциал действия всегда имеет силовой, а не информационный характер – он направлен на изменение окружающей среды нейрона. Если инфузория-туфелька способна двигать «ножками», то нейрон не имеет такой возможности (но он этого не знает), поэтому реализует своё движение, генерируя электрический импульс. То, что этот импульс не приводит к движению самого нейрона, не играет никакой роли – входные сигналы нейрона изменяются в соответствии с его воздействием на свою окружающую среду, поэтому для него это абсолютно эквивалентно реальному движению инфузории-туфельки. Форма импульсов потенциала действия в этом случае не играет особой роли, важна лишь их интенсивность в данный момент времени.

В то же время медиаторы, выделившиеся во входных синапсах или при объёмной передаче, для данного нейрона будут являться входными информационными сигналами. Их форма опять же не играет никакой роли – важна лишь интенсивность и динамика во времени. Нейрон, как и инфузория-туфелька, должен реагировать на динамику входящих сигналов, соизмеряя таким образом скорость своего «движения». Вообще, такое рассмотрение нейрона как активной системы в значительной степени относительно, как мы уже показали это в разделах «Модель активной системы» и «Взаимодействие систем». Но нам проще будет понимать процесс, если мы будем придерживаться именно такой точки зрения.

Ансамбли нейронов способны анализировать большее количество и разнообразие сигналов. Ансамбль можно рассматривать с позиции колоний микроорганизмов или простых организмов типа вольвокса, губок или других, имеющих небольшое количество клеток одной или разных специализаций. Ансамбль нейронов эквивалентен одному, более развитому нейрону, понимающему большее количество входных сигналов и формирующему более сложный сигнал изменения своей окружающей среды, нежели каждый отдельный нейрон.

Поведение и целеполагание организма как целого определяется, но не имеет ничего общего с целеполаганием отдельного нейрона или ансамбля. Например, хотя наше чувство голода и связано с поведением отдельных нейронов, они при этом могут быть совсем не голодны – учитывая тот объём ресурсов, которые организм направляет на содержание нервной системы. Каждый отдельный нейрон живёт своей жизнью и изменяет только свою собственную окружающую среду, организм же реализует волю коллектива нейронов и изменяет свою окружающую среду – сам себя, окружающие вещи, природу, социальные отношения и тому подобное. Отдельный нейрон обо всём этом не имеет ни малейшего понятия.

Из вышесказанного ясно, что автоматные модели подходят для описания реальных процессов гораздо более, нежели модель рекуррентной нейросети. Рассмотрим модель коллективного поведения автоматов более подробно. Основы этой теории были заложены в работах советских математиков – М.Л.Цетлина, Д.А.Поспелова и В.И.Варшавского [Error: Reference source not found, [11]]. Центральным понятием в ней является понятие элементарного автомата, принципиальная схема которого может быть представлена, например, в следующем виде.

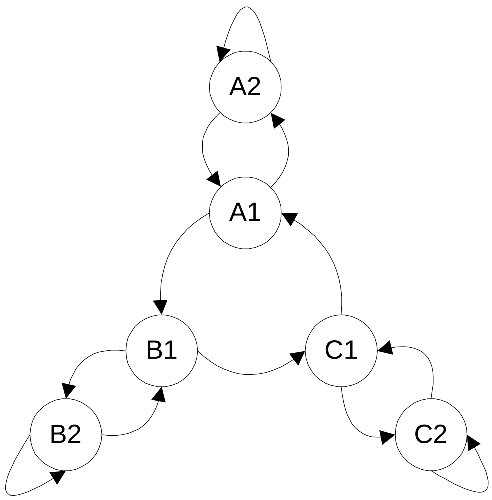

Рис. 4

На данном рисунке представлена не полная модель автомата, а только схема его памяти – состояний и переходов между этими состояниями. Кроме того, возможны различные способы построения автоматов, с другими схемами переходов и состояний. И даже при данной схеме построения возможны вариации как в количестве, так и в глубине ветвей. Рассмотрим эту схему более детально.

Каждый автомат характеризуется множеством его состояний. Состояние автомата определяет, какое действие он будет выполнять в зависимости от его входных сигналов. Одни и те же входные сигналы при разных состояниях приведут к разным действиям автомата. Также как и различные входные сигналы приведут к разным действиям, даже если автомат находится в одном и том же состоянии.

Недостаток – плоская модель, недостаток динамики.

Ещё – про обучение. У Цетлина в больших коллективах с этим плохо.

Автомат – функция с параметром.

Оценка вычислительной мощности, объёма памяти одноклеточных (угловые изменения в движении как мера информации, или сравнение с автоматами).

Алгоритм как конфигурация

В силу того, что всё изложение строится на введённом в самом начале понятии алгоритма как конфигурации вычислительного пространства, необходимо дать более подробное и конкретное его определение, а также показать, что оно применимо для описания интеллектуальных систем. В разделе «Определения» было дано следующее понятие алгоритма.

«Алгоритм есть конфигурация области некоторого пространства – вычислительной среды данного алгоритма. Модификация алгоритма означает изменение конфигурации данной области пространства, а процесс вычисления представляет собой взаимное изменение свойств самого алгоритма и некоторого объекта – сигнала при его прохождении через эту область пространства. Модификация как алгоритма, так и сигнала возможна только в процессе вычисления».

Затем, в разделе «Модель активной системы», было уточнено понятие сигнала как любого воздействия одной системы на другую. На самом деле сигнал тоже представляет собой конфигурацию пространства, то есть – алгоритм! Таким образом, ***в нашей модели нет ничего, кроме вычислительного пространства, алгоритма и процесса вычисления, представляющего собой движение***.

Все системы представляют собой конфигурации вычислительного пространства, и сигналы, которыми обмениваются системы, также являются конфигурациями (то есть алгоритмами) вычислительного пространства. Свойство движения не является прерогативой сигналов, ибо движение относительно – можно рассматривать движение сигнала относительно системы, или наоборот, движение системы относительно сигнала. Важно то, что ни сигнал, ни система не порождают процесс вычисления сами по себе, то есть статичная конфигурация не способна измениться самостоятельно. И только когда в одной области пространства присутствуют разнонаправленные движения, возникает процесс вычисления и изменение конфигураций, то есть – алгоритмов.

Более того, системы могут взаимодействовать непосредственно, «интерферируя» друг с другом при относительном движении одной конфигурации сквозь другую. Тогда каждая их них будет являться «сигналом» для других. При этом системы могут взаимодействовать не целиком, а частично, и проходить друг через друга как угодно – по вертикали, диагонали, окружности или более сложной траектории, если это позволяет размерность вычислительного пространства и сам процесс вычисления.

Далее, как видно из определения, ***система – это исключительно логическое понятие, применяемое для группировки объектов некоторой области пространства для удобства рассмотрения***. Поэтому мы с полным правом можем говорить не о взаимодействии систем, а о взаимодействии алгоритмов как «интерференции» конфигураций пространства. При этом алгоритм, как и система, не является строго определённой областью пространства, мы можем лишь условно обозначить границы конфигураций, основываясь на относительном движении областей вычислительного пространства. Необходимо осознать этот факт, несмотря на то, что он противоречит обыденному пониманию алгоритма, диктуемом нам последовательностью нашего логического мышления.

Ещё одно серьёзное предубеждение, которое нам необходимо преодолеть – это разделение в обыденном сознании понятий алгоритма и переменных, функции и её аргументов. В нашей модели и функции, и аргументы суть одно – конфигурации вычислительного пространства, алгоритмы. При взаимодействии конфигураций меняются не только аргументы функции, но и сама эта функция. Например, число **4** есть алгоритм. Следовательно, можно рассмотреть взаимодействие двух алгоритмов, представляющих два числа **4**, при этом они оба получат возможно другое, но вполне определённое значение! ***Функции и аргументы суть одно и то же – алгоритмы, конфигурации пространства. В нашей модели алгоритм и его переменные не существуют раздельно, это одно целостное понятие.***

С точки зрения нашей модели все объекты реального мира по определению представляют собой алгоритмы, по которым работает наш мир. Причём, как уже было сказано, алгоритм в нашей модели не является чётко очерченной областью пространства – его границы условны, и возможным ориентиром может служить движение какой-то области как целого, однако эта область при взаимодействии может разбиться на разные области с разными траекториями или же наоборот, объединиться с другими областями, продолжая движение по одной траектории. Поэтому говорить о взаимодействии алгоритмов как неделимого целого не имеет особого смысла.

Попытаемся дать более точное геометрическое толкование этих понятий, не претендующее, однако, на значительную физическую строгость (в частности, понятие координат, времени и расстояния, а скорее всего – и самого пространства в квантовом мире следует определять иначе; на некоторые выводы может натолкнуть зависимость угла полного поворота от спина частицы, который не равен в общем случае 360 градусов [[12]], а также проявление вихревого магнитного поля как релятивистского эффекта линейного электрического при повороте осей – поля перпендикулярны уже в трех измерениях, но линейное движение во времени воспринимается как круговое в пространстве. Однако, в любом случае, эти понятия используются в нашей модели не очень широко – только для определения области взаимодействия алгоритмов, поэтому она не должна пострадать при таком переопределении).

Вычислительное пространство есть нечто, имеющее *протяжённость* и способное принимать форму, или иными словами – способное иметь локальную неоднородность, локальные свойства. Пространство характеризуется только его связностью, при этом размерность пространства в нашей модели особой роли не играет, однако в случае одномерного пространства движущиеся друг относительно друга конфигурации обязательно провзаимодействуют, а в многомерном случае – могут и разминуться. Поэтому в общем случае вычислительное пространство полагается многомерным.

Конфигурация есть совокупность отличий двух локальных областей пространства друг от друга. Конфигурация сама по себе не имеет никакого смысла – она также не порождает вычисления, как и «ровная» поверхность. Когда мы представляем пространство некоторой поверхностью, то её форму мы подсознательно воспринимаем относительно привычного нам евклидова пространства. «Ровная» поверхность является полноправной конфигурацией, но в рассуждениях мы, как правило, упускаем её из вида, так как влияние такой ровной конфигурации на все области другой, движущейся сквозь неё конфигурации, одинаково и меняет их свойства одинаковым образом.

Процесс вычисления есть *движение* одной конфигурации (локальной неоднородности) относительно другой. Изменение конфигурации также представляет собой движение, когда соседние области пространства движутся по разным *траекториям*, то есть меняют свои свойства различным образом. Говорить о движении конфигурации самой по себе бессмысленно – любая её скорость эквивалентна покою, ибо не порождает изменения данной конфигурации. Разумной аналогией движения и взаимодействия конфигураций следует считать распространение и интерференцию волновых пакетов.

Протяжённость характеризует число степеней свободы пространства, когда два объекта могут иметь одни и те же свойства (или не иметь никаких других свойств), но тем не менее быть различимы только на основании их положения в пространстве. В этом случае можно сказать, что они всё же отличаются одним из свойств – у них разные координаты. В нашей модели все свойства конфигурации можно описать координатами её элементарных областей (точек) – пространственными и «временными» координатами. При этом пространственные координаты независимы, а все «временные», или собственно свойства – зависимы в том смысле, что если две точки обладают одинаковыми пространственными координатами, то их «временные» координаты изменяются, порождая процесс вычисления.

Движение есть изменение свойств объекта, в нашей модели – изменение «временных» координат некоторой области вычислительного пространства. Понятие движения нельзя исключить из модели, представив конфигурацию застывшей областью пространства-времени, ибо такое исключение унесёт с собой и процесс вычисления, и конфигурацию пространства как таковую. Поэтому время не может выступать в качестве пространственной координаты, скорее вместо него следует рассматривать скорость как одну, и возможно – единственную, из «временных» координат. Скорость – это вектор, обладающий величиной и направлением, она задаёт шаг и направление распространения движения.

Траектория есть воображаемая область пространства, совокупность координат, которые принимает объект в процессе своего движения. Нельзя сказать, что траектория и есть сам объект в пространстве-времени, ибо в таком случае исчезает само понятие движения. Если объект был в некоторой точке пространства, но изменил положение, это значит, что его там нет. То же самое и во времени – если объект присутствовал в каком-то времени, но изменил положение, это значит, что его там уже нет. Однако, совокупность координат, принимаемых объектом, мысленно можно себе представить в виде его траектории – статичной поверхности в пространстве-времени (то есть – вычислительном пространстве).

Геометрическое представление алгоритма было выбрано не случайно – оно, на мой взгляд, как никакое другое способно объединить математику и физику нашего реального мира. Такое представление алгоритма, в общем говоря, не ново – существует понятие клеточных автоматов [[13], [14]], однако наша модель не эквивалентна клеточным автоматам в силу совершенно иного подхода к процессу вычисления: если в клеточных автоматах для вычисления используется признак соседства точек самой конфигурации, то в нашем случае вычисление происходит только при относительном движении и взаимной «интерференции» как минимум двух конфигураций, кроме того, движение в ней не рождается в процессе вычисления, а является полным эквивалентом самого вычисления – без движения нет вычисления. Тем не менее, с учётом ограничений на движение, модель клеточных автоматов даёт адекватное визуальное представление о взаимодействии алгоритмов.

Полезной аналогией для рассмотрения нашей модели может служить стеклянная призма. Это алгоритм, который умеет, кроме всего прочего, преломлять или отражать свет под определённым углом, причём свет разной длины волны преобразуется по-разному. Луч света – это тоже алгоритм, он может провзаимодействовать со всей или частью призмы как целиком, так и частично – когда одна часть луча пройдёт сквозь призму, а другая – мимо. Если мы попробуем стукнуть по призме, то почувствуем боль, то есть наша система при взаимодействии с призмой получила ответ – призма есть твёрдый предмет. Призму можно разбить на части, после чего она может сохранить, либо потерять свои свойства, если осколки слишком мелкие. Таким образом, один и тот же алгоритм призмы ведёт себя по-разному в зависимости от проходящего сигнала и может терять свои свойства при декомпозиции на подсистемы.

Ещё одной полезной аналогией может служить лист бумаги. Характеристикой этого алгоритма будет являться шероховатость, изогнутость, наличие текста и тому подобное. При взаимном движении двух листов, плотно прижатых друг к другу, они мнутся, истираются, смазывается текст – то есть происходит процесс вычисления, изменяющий оба алгоритма. Содержание текста в этом случае не играет особой роли просто потому, что эта конфигурация взаимодействует с неподходящим с нашей точки зрения алгоритмом, тем не менее свет рассеивается на буквах иначе, истирание в местах краски может быть более сильным – то есть текст в общем случае также влияет на взаимодействие.

Если продолжать аналогию, то движение сигнала внутри системы можно представить как лист, перегнутый пополам, где каждая его половина скользит относительно другой без внешнего участия. Если при этом происходит истирание листа, то это уже будет динамической системой, при некоторых определённых условиях она будет интеллектуальной. Однако, следует предостеречь от мысли, что лист может согнуться пополам самостоятельно – его способна изогнуть только внешняя система, в данном случае – наша рука. Но уж если половинки стали двигаться друг относительно друга, то они могут продолжать это движение уже без внешнего участия. Это движение может продолжаться или затухнуть, для нас это будет означать, что внутренние связи системы в каком-то месте системы не замкнуты и сигнал способен покинуть пределы системы.

Здесь мы проскочили один важнейший момент, а именно – текст на листе бумаги может представлять собой некоторый алгоритм, написанный нами, например – алгоритм сортировки стопки книг. Процесс истирания верного алгоритма будет очень мало отличаться от истирания алгоритма, содержащего ошибку. Значительно менее, чем от истирания зачириканного ручкой листа. Возникает резонный вопрос – а эквивалентен ли сам алгоритм его записи на материальном носителе?!

Нет, не эквивалентен. И одновременно – да, эквивалентен, но только по отношению к определённой, предельно узкой группе сигналов. То есть две записи того же алгоритма, сделанные на бумаге и, например, на жёстком диске компьютера будут эквивалентны при условии, что текст на бумаге взаимодействует со световым лучом, нашим глазом и мозгом, а жёсткий диск взаимодействует с компьютером, монитором, нашим глазом и мозгом. В обоих случаях в определённой области нашего мозга возникнет примерно одна и та же конфигурация, при прохождении через которую будут меняться и другие мыслительные процессы.

У другого человека обе эти записи также вызовут примерно одинаковую, но отличную от нашей, конфигурацию в определённой, но отличной от нашей, области его мозга. Однако, если мы оба будем выполнять предписываемые алгоритмом действия, то отсортируем стопку книг одинаково. Следовательно, алгоритм на бумаге и алгоритм на жёстком диске компьютера преобразуются в одну и ту же конфигурацию вычислительного пространства при условии их взаимодействия с другими, подходящими алгоритмами вычислительного пространства.

Сможем ли мы должным образом отсортировать стопку книг, не имея представления, по какому критерию их нужно отсортировать? Очевидно, нет. А если алгоритм придумали мы сами, записали его на листке и занесли в компьютер, а потом компьютер сломался, листок сгорел, а сам алгоритм мы забыли? Очевидно, что мы утратили алгоритм насовсем, такая конфигурация просто исчезла из нашего пространства, и мы также не сможем отсортировать эту стопку книг в нужном порядке (если не восстановим алгоритм на основе других алгоритмов в нашем сознании).

Текст, лист бумаги, световой луч и всех нас можно рассматривать как множество систем, также и жёсткий диск, компьютер, монитор всех нас – как ещё одно множество систем. Все эти системы из обоих множеств будут иметь эквивалентную конфигурацию для сигнала «отсортировать книги». То есть именно ***физически*** эквивалентную по отношению к определённой группе сигналов конфигурацию реального пространства, проходя через которую звуковая волна, изображение или другой сигнал «отсортировать книги» приведёт к тому, что конфигурация стопки книг будет изменена.

То есть, если две системы преобразуют определённую группу алгоритмов одним и тем же образом, то их конфигурация должна быть эквивалентна относительно этой группы алгоритмов. Это не значит, что сигнал «отсортировать книги» пойдёт точно по той же траектории во всех случаях, но, тем не менее, это значит, что конечная траектория этого сигнала будет примерно одной и той же во всех случаях. И даже более того – для каждого отдельного индивида определённый вид сигнала «отсортировать книги» пойдёт именно по той же самой траектории в ***физическом*** пространстве! Конечно, это верно только в идеальном случае, когда такая конфигурация в мозге только одна и она не успела сильно измениться под влиянием других, но принципиальный вывод – именно таков. Причём он действительно подтверждается реальными исследованиями ЭЭГ головного мозга (см., например, [[15]]).

Таким образом, алгоритм – исключительно материальное понятие, и без материального носителя он не существует. При этом, если разные области вычислительного пространства преобразуют один и тот же алгоритм одинаковым образом, то они имеют в идеале одинаковую, а в общем случае – эквивалентную конфигурацию в той своей подобласти, с которой взаимодействует данный алгоритм. Интеллектуальная система сама строит необходимую конфигурацию из других доступных ей конфигураций с тем, чтобы преобразовать определённый тип сигналов нужным ей образом, причём эта конфигурация – исключительно материальное понятие. Поэтому можно утверждать, что ***интеллект и даже человеческий разум – познаваем и имитируем***.

Как будет видно из нижеследующей теоремы Т. 2, введённое нами геометрическое понятие алгоритма в общем случае шире формализма машины Тьюринга. Однако, как будет показано в теореме Т. 3, интеллектуальная система в определённых случаях всё же представима в виде модифицированной машины Тьюринга. Доказательство будем строить с использованием теорем раздела «Алгоритмически неразрешимые проблемы», нарушая последовательность следования теорем. Однако, доказываемая никак не используются в привлекаемых теоремах, поэтому никакого противоречия здесь возникнуть не может.

Т. 2

Теорема: Геометрическое представление алгоритма как конфигурации вычислительного пространства шире формализма машины Тьюринга с бесконечной лентой.

Доказательство: Действительно, мы не вводили никаких ограничений на вид конфигурации, следовательно, она может представлять и любую машину Тьюринга. С другой стороны, использование машины Тьюринга с бесконечной лентой приводит к алгоритмически неразрешимым проблемам, которые, в соответствии с теоремой Т. 6, могут быть разрешены в геометрической (конфигурационной) модели алгоритма. Следовательно, тезис Чёрча-Тьюринга неверен, а модель машины Тьюринга не полна в сравнении с геометрической моделью алгоритма. **Теорема доказана.**

Следствие: Не все алгоритмы представимы машиной Тьюринга с бесконечной лентой.

Если машина Тьюринга не способна вычислить некоторые алгоритмы, то возможна ли вообще имитация интеллекта современными вычислительными средствами? Думаю, что на этот вопрос можно ответить утвердительно, но для доказательства нам потребуется **машина Тьюринга с кольцевой лентой** – такая машина Тьюринга, лента которой имеет произвольно задаваемую длину, но при этом не имеет концов (то есть она замкнута в кольцо). С точки зрения ЭВМ это предполагает замкнутость произвольно выбираемых адресных пространств, когда после самого последнего машинного слова в этом адресном пространстве следующим опять идёт самое первое.

При кажущейся простоте такого дополнения машины, оно, тем не менее, носит принципиальный характер. А именно – оно подразумевает отказ от понятия **результата** как фиксированного ответа на некоторой ограниченной области ленты. Вся кольцевая лента целиком и является одновременно и алгоритмом, и результатом его работы! В кольцевой модели не существует проблемы останова, как и самой команды завершения вычислений, а результат представляет собой динамическое понятие – он может быть статически определённым, если содержимое ленты перестало меняться во времени, либо динамически определённым, если содержимое периодически меняется во времени (но оно статично в пространстве-времени). В конечной машине Тьюринга возможен либо первый, либо второй вариант.

Такое дополнение машины Тьюринга необходимо для описания самой интеллектуальной системы, но не для действий, выполняемых ею над некоторым внешним сигналом. Действительно, поскольку интеллект по определению – результат суперпозиции своего же алгоритма, то весь алгоритм целиком и является результатом вычисления. При этом обратные связи, по которым циркулирует сигнал и идёт процесс вычисления, в идеальном случае являются замкнутыми, когда любая произвольно взятая область этого сигнала, последовательно и непрерывно пробегая по всей системе, возвращаются в исходную позицию.

Однако, для внешних сигналов машина Тьюринга с кольцевой лентой будет неотличима от машины с бесконечной лентой – информация на ленте будет меняться обратными связями, и внешний сигнал будет обрабатываться так же, как и обычной машиной. При этом внешний сигнал может полностью пойти по обратным связям и стать частью интеллектуальной системы (также он может и покинуть её, если связи не замкнуты), но говорить при этом об отсутствии результата нельзя, ибо алгоритм интеллектуальной системы (впрочем, как и любой другой алгоритм) всегда представляет собой результат каких-либо предыдущих вычислений.

Кроме того, доказательство мы будем строить в **дискретном пространстве** – таком вычислительном пространстве, каждая произвольно выбираемая конечная область которого может принимать только счётное число состояний. В континуальном пространстве говорить о машине Тьюринга не представляется возможным, так как её алфавит можно задать только на счётном множестве. Однако, такое ограничение не является для нас решающим, поскольку физическое пространство, в котором функционируют реальные интеллектуальные системы, по современным представлениям также является дискретным, в котором значения всех физических величин квантованы. При этом бесконечных энергий мы в реальных интеллектуальных системах (как и вообще в мире) не наблюдаем, поэтому будем считать, что число состояний любой конечной области такого пространства также и конечно.

Т. 3

Теорема: Алгоритм интеллектуальной системы в дискретном вычислительном пространстве представим машиной Тьюринга с кольцевой лентой.

Доказательство: В ходе доказательства теоремы Т. 1 было показано, что интеллектуальная система должна представлять собой как минимум одно кольцо алгоритмов. Следовательно, она может занимать ограниченную область **S** вычислительного пространства. Представим эту кольцевую область **S** в виде полного тора (сплошного «бублика») соответствующей размерности, мысленно разделим его на отдельные сектора в соответствии с нашим представлением о количестве алгоритмов, составляющих данное кольцо. Каждый такой сектор из **S** назовём точкой.

Поскольку вычислительное пространство дискретно, а в ограниченной области **S** число его возможных состояний конечно, то мы можем выбрать такой конечный алфавит **A**, буквы которого будут соответствовать любым допустимым состояниям точек в области **S**. Промаркируем все точки из **S** буквами алфавита **A**, назначая букву каждой точке в зависимости от её текущего состояния. В общем итоге у нас получится **N** точек, которые будут промаркированы **M** символами, также могут остаться и незадействованные буквы алфавита **A**, если какие-то из возможных состояний в данный момент не заняты ни одной из точек.

На всём множестве точек зададим отношение следования в соответствии с порядком этих точек в исходном алгоритмическом кольце. Это отношение будет однозначным и транзитивно замкнутым, когда каждая точка имеет только одну предшествующую и только одну следующую точку, при этом первая следует за самой последней. Упорядоченный таким образом набор точек назовём лентой.

Зададим таблицу переходов с количеством строк и столбцов, равным полному количеству букв в алфавите **A**, обозначив это количество букв **K = |A|**, будем иметь таблицу размером **K*K**. Пусть символы алфавита **A** представляют собой строчные буквы латинского алфавита: **a**, **b**, **c** и т.д., обозначим состояния знаком **q** с индексом, соответствующим букве предшествующей точки: $\mathbf{q_a}$, $\mathbf{q_b}$, $\mathbf{q_c}$ и т.д. В каждую ячейку таблицы занесём инструкцию замены символа, инструкцию перехода в соответствующее новому символу состояние, а также – инструкцию перевода головки на один шаг вправо.

Символ \\ Состояние	$\mathbf{q_a}$	$\mathbf{q_b}$	$\mathbf{q_c}$	$\mathbf{q_d}$	**…**
a	b, q_b, $\rightarrow$	a, q_a, $\rightarrow$	b, q_b, $\rightarrow$	a, q_a, $\rightarrow$	…
b	c, q_c, $\rightarrow$	c, q_c, $\rightarrow$	d, q_d, $\rightarrow$	c, q_c, $\rightarrow$	…
c	d, q_d, $\rightarrow$	b, q_b, $\rightarrow$	a, q_a, $\rightarrow$	a, q_a, $\rightarrow$	…
d	a, q_a, $\rightarrow$	c, q_c, $\rightarrow$	b, q_b, $\rightarrow$	c, q_c, $\rightarrow$	…
…	…	…	…	…	…

Таблица 2

Тогда новый символ в текущей позиции на ленте будет полностью определяться старым символом в этой позиции и символом в предшествующей позиции, вычисленном на предыдущем шаге. Очевидно, такая машина Тьюринга будет адекватно описывать взаимодействие и преобразование алгоритмов в кольце по любым правилам. При этом мы всегда можем задать таблицу переходов таким образом, чтобы в независимости от исходного дальнейшее состояние ленты стремилось к какому-либо определённому виду. Продемонстрируем это утверждение.

Введём грамматику над алфавитом **A**, где в левой части правил поставим все допустимые **N**-буквенные слова в алфавите **A**, а в правой – все **N**-буквенные слова, в которые возможен переход. Число таких правил будет большим, но конечным: $\mathbf{K^N}$. При этом обеспечим её полноту – если слово встречается в правой части какого-то правила, оно должно быть представлено также и в левой части этого или другого правила. Однако, не будем придерживаться обратного принципа – часть слов не будем включать в правые части правил, продублировав остальные слова для заполнения всех $\mathbf{K^N}$ правил. Например:

aaaa → aaab

aaab → aaac

aaac → abcd

aaad → abcd

…

abcd → abcd

…

Тогда требование эргодичности, выдвинутое в теореме Т. 1, будет удовлетворено, причём такая грамматика без терминального символа будет соответствовать некоторой машине Тьюринга с кольцевой лентой (мы всегда можем обозначить слова нашей грамматики буквами какого-то другого алфавита **B** и задать машину Тьюринга с одним состоянием и лентой из единственной ячейки) и будет представлять интеллектуальную систему. **Теорема доказана.**

Итак, я обещал определить более конкретно понятия, связанные с алгоритмом. В нашей модели это: вычислительное пространство, его конфигурация (алгоритм), и процесс вычисления.

Формальное определение: вычислительное пространство есть полугруппа отображений $\langle \mathbf{R}, \circ \rangle$, заданная на множестве всех алгоритмов **R** и замкнутая относительно операции суперпозиции «$\circ$». Алгоритм $\mathbf{r_i} \in \mathbf{R}$ есть такое определённое на всём множестве **R** однозначное отображение $\mathbf{r_i}: \mathbf{R} \rightarrow \mathbf{R_i}$ этого множества в некоторое его подмножество $\mathbf{R_i} \subseteq \mathbf{R}$, что для любых $\mathbf{r_a}, \mathbf{r_b} \in \mathbf{R}$ справедливо равенство: $\mathbf{r_b(r_a) = r_a \circ r_b}$, где $\mathbf{r_b(r_a)}$ – функциональная запись отображения $\mathbf{r_b}$ от аргумента $\mathbf{r_a}$, а суперпозиция записывается в прямом[1] порядке [16, 17]. Процесс вычисления не является понятием математической, но только физической модели, где он вводит понятие времени и расстояния в применение операций суперпозиции и отображения.

Трактовать данное определение проще всего, рассматривая отображение как определённую структуру, сопоставляющую одним элементам некоторого множества другие его элементы, при этом в зависимости от способа упорядочения исходного множества может меняться график данного отображения, но не само отображение как структура. Например, можно задать отображение множества **F = {c b a f e d}** в себя как множество упорядоченных пар **f = {(c c), (b c), (a f), (f d), (e a), (d e)}**, при этом элементами множества **F** могут являться произвольные объекты, в том числе – и сами отображения $\mathbf{f_1 \ldots f_n}$. Заметим, что **R** по определению – множество всех алгоритмов, но не множество всех возможных на множестве **R** отображений – здесь мы предполагаем наличие некоторых ограничений на отображения, включаемые в множество алгоритмов **R**.

Важно различать физическое вычислительное пространство и его математическую абстракцию **R**, ибо **R** по определению – полная совокупность, бесконечное множество всех возможных конфигураций, в реальном же вычислительном пространстве некоторые конфигурации могут и не присутствовать в каком-то месте или в какой-то момент времени, однако они всегда выводимы из существующих конфигураций, которые, в свою очередь, могут иметь произвольное число дубликатов. Как мы увидим позже, **R** является метрическим топологическим пространством, однако точки этого пространства (алгоритмы) не соответствуют прямо точкам физического пространства (координатам).

Понятия «алгоритм» и «конфигурация», а также «пространство» и «вычислительное пространство» в нашей модели абсолютно эквивалентны и используются в паре только для того, чтобы разгрузить текст от повторений. В формальной модели также эквивалентны термины «система», «сигнал» и «алгоритм». Они используются в зависимости от преимущества ассоциаций, и всегда однозначно могут быть заменены друг другом.

Алгоритмы я буду обозначать строчными буквами латинского алфавита, либо как $\mathbf{r_a, r_b, r_i \in R}$, где индекс «**i**» обозначает любую конфигурацию из **R**, либо как $\mathbf{s, t, u \in R}$, подразумевая что $\mathbf{s \in S \subset R}$, где **S**, **T**, $\mathbf{U \subset R}$ – подмножества конфигураций, объединяемых по какому-либо критерию. Операцию суперпозиции наряду с $\mathbf{t = s \circ u}$ я буду обозначать также как **t = u(s)**, поскольку аргументами отображения являются сами алгоритмы, что в случае множеств аналогично записи $\mathbf{u: S \to T}$.

Также в дальнейшем будет употребляться термин **вычисление**. Алгоритм $\mathbf{u \in R}$ может вычислить алгоритм $\mathbf{t \in R}$ в том случае, когда в вычислительном пространстве **R** существует такая конфигурация **s**, которую **u** преобразует в **t**, то есть **t = u(s)**.

T. 4

Теорема: Предложенная формальная модель $\mathbf{\langle R, \circ \rangle}$ адекватно описывает введённую выше физическую модель взаимодействия алгоритмов.

Доказательство: Введём в физическом пространстве систему координат, каждую элементарную ячейку в этих координатах назовём точкой. Каждая такая точка пространства будет обладать определённым набором физических свойств, причём любая точка кроме координат будет характеризоваться только этими её свойствами и ничем более. Следовательно, все явления в физическом пространстве будут определяться исключительно свойствами его точек. Обозначим множество всех возможных состояний – наборов этих свойств – как **P**, тогда мы с уверенностью сможем сказать, что каждая точка физического пространства находится в каком-то состоянии $\mathbf{p_i \in P}$.

Определим множество **Q** всех комбинаций элементов множества **P**: если $\mathbf{p_1, p_2 \in P}$, то все произвольные наборы $\mathbf{\langle p_1 \rangle, \langle p_2 \rangle, \langle p_1\ p_1 \rangle, \langle p_1\ p_2 \rangle, \langle p_2\ p_1 \rangle, \langle p_2\ p_2 \rangle, \langle p_1\ p_1\ p_1 \ldots \rangle, \langle p_1\ p_2\ p_1 \ldots \rangle, \langle p_2\ p_1\ p_2 \ldots \rangle \in Q}$. Тогда мы сможем утверждать, что любая произвольно взятая область физического пространства находится в

[1] Часто применяют обратный порядок записи суперпозиции, при котором запись $\mathbf{g \circ f(x)}$ означает **g(f(x))**, т.е. отображения в композиции пишутся в порядке, обратном тому, в каком они применяются. Мы же будем везде использовать прямой порядок $\mathbf{f \circ g}$, естественный для других математических операций – сложения и умножения, возведения в степень, полагая, что $\mathbf{f \circ g(x) = g(f(x))}$.

каком-то состоянии $\mathbf{q_i} \in \mathbf{Q}$ относительно данной системы координат. Привязка к координатам важна из-за анизотропии сконфигурированного пространства: одна и та же область может одновременно находиться в состоянии $<\mathbf{p_1}\ \mathbf{p_2}>$ в одной системе координат, и в состоянии $<\mathbf{p_2}\ \mathbf{p_1}>$ в другой, с обращённым направлением осей – если стальной клинок нагреть, а потом резко охладить, это не то же самое, как если его сперва охладить, а потом нагреть.

Рассмотрим теперь движение некоторого пробного тела в физическом пространстве. В каждый момент времени такое пробное тело испытывает влияние какой-то одной определённой области пространства. Это влияние выражается в том, что состояние данного тела меняется при движении сквозь эту область пространства. Следовательно, мы можем рассматривать каждую область физического пространства в состоянии $\mathbf{q_i} \in \mathbf{Q}$ как некоторое отображение $\mathbf{r_i} \in \mathbf{R}$, осуществляющее преобразование предыдущего состояния пробного тела в его следующее состояние. Такое отображение, очевидно, будет определено на множестве всех возможных состояний пробного тела.

Но пробное тело, в свою очередь, представляет собой конфигурацию некоторой области пространства, поэтому оно не может иметь никаких других состояний, нежели из множества $\mathbf{Q}$. Следовательно, любое отображение $\mathbf{r_i} \in \mathbf{R}$ ставит в соответствие некоторые состояния из $\mathbf{Q}$ другим состояниям из $\mathbf{Q}$. При этом, поскольку каждая область пространства характеризуется только её состоянием, то любые две области, находящиеся в одинаковом состоянии, будут влиять на одно и то же пробное тело одинаково, а для двух областей, находящихся в разном состоянии, всегда найдётся такое пробное тело, для которого влияние этих областей будет различным. В противном случае говорить об эквивалентности или различии физических состояний не имело бы никакого смысла. Поэтому каждому состоянию $\mathbf{q_i} \in \mathbf{Q}$ взаимно однозначно соответствует единственное отображение $\mathbf{r_i} \in \mathbf{R}$.

И действительно, допустим существование двух различных областей $\mathbf{q_1}, \mathbf{q_2} \in \mathbf{Q}$, таких что $\mathbf{q_1} \neq \mathbf{q_2}$, которым соответствует одно и то же отображение $\mathbf{r} \in \mathbf{R}$. Тогда каково бы ни было пробное тело $\mathbf{q_t} \in \mathbf{Q}$, взаимодействие с данными конфигурациями будет для него неразличимо – в результате в любом случае получится тело $\mathbf{r(q_t)}$. Пусть также существует некоторая конфигурация $\mathbf{q_a} \in \mathbf{Q}$ с отображением $\mathbf{r_a} \in \mathbf{R}$, при взаимодействии с которой исходные области преобразуются в $\mathbf{q_3} = \mathbf{r_a(q_1)}$ и $\mathbf{q_4} = \mathbf{r_a(q_2)}$, и пусть им соответствуют отображения $\mathbf{r_3}, \mathbf{r_4} \in \mathbf{R}$, причём $\mathbf{r_3} \neq \mathbf{r_4}$. Но тогда мы можем сконструировать пробное тело, состоящее из двух областей – $\mathbf{q_a}$ и $\mathbf{q_t}$, для которого взаимодействие с исходными конфигурациями будет различным, что противоречит нашему выводу об их неразличимости. Следовательно $\mathbf{r_3} = \mathbf{r_4}$, а неразличимые конфигурации будут оставаться неразличимыми при любых преобразованиях $\mathbf{r_i} \in \mathbf{R}$. Но тогда никакое преобразование из $\mathbf{q_1}$ в $\mathbf{q_2}$ и обратно не будет возможным, а всё множество $\mathbf{Q}$ разобьётся на непересекающиеся и никак неразличимые классы с числом элементов $|\mathbf{R}|$. Поскольку мы не можем физически различить конфигурации, в том числе анализом составляющих их точек $\mathbf{p_i} \in \mathbf{P}$, то должны считать их эквивалентными, а исходное предположение $\mathbf{q_1} \neq \mathbf{q_2}$ – неверным.

Таким образом, в силу однозначного соответствия, всякое отображение $\mathbf{r_i} \in \mathbf{R}$ можно рассматривать определённым на множестве $\mathbf{R}$. То есть, каждое состояние $\mathbf{q_i} \in \mathbf{Q}$ области физического пространства задаёт какое-то своё, отличное от других, отображение $\mathbf{r_i}: \mathbf{R^n} \rightarrow \mathbf{R_i^m}$, определённое на множестве всех отображений $\mathbf{R}$, где $\mathbf{n}$ и $\mathbf{m}$ – число аргументов и результатов (для случая взаимодействия нескольких конфигураций). Причём, поскольку множество $\mathbf{Q}$ по определению замкнуто относительно любых взаимодействий в физическом пространстве, также замкнутым относительно любых своих отображений будет и множество $\mathbf{R}$ – мы не можем получить конфигурацию, не представимую в физическом пространстве.

Рассмотрим теперь случай взаимодействия нескольких тел. Каждое из них будет изменяемо остальными телами, участвующими во взаимодействии. Однако, тело на самом деле не может рассматриваться как отдельная область, «парящая» в окружающем пространстве, а только как некоторая конфигурация этого же пространства. Следовательно, с точки зрения каждого из участвующего во взаимодействии тел все остальные могут быть заменены единственным телом, причём оно будет описываться каким-то определённым состоянием $\mathbf{q_s} \in \mathbf{Q}$. Поэтому любое $\mathbf{n}$-арное $\mathbf{m}$-значное отображение $\mathbf{r_i}: \mathbf{R^n} \rightarrow \mathbf{R_i^m}$ в вычислительном пространстве представимо бинарным однозначным отображением $\mathbf{r_s}: \mathbf{R} \rightarrow \mathbf{R_s}$.

Ну а требование $\mathbf{r_b(r_a)} = \mathbf{r_a} \circ \mathbf{r_b}$ есть ни что иное, как стандартное определение операции суперпозиции: если пробное тело $\mathbf{q_t} \in \mathbf{Q}$ последовательно взаимодействует с конфигурацией $\mathbf{r_a}$ а затем – с $\mathbf{r_b}$, то отображение $\mathbf{q_y = r_b(r_a(q_t))}$ есть ни что иное, как суперпозиция $\mathbf{q_y = [r_a \circ r_b](q_t)}$, а с учётом того, что в нашем случае аргументами и результатами являются сами отображения, заменяя $\mathbf{q_t}$ и $\mathbf{q_y}$ на $\mathbf{r_t}$ и $\mathbf{r_y}$, окончательно получим: $\mathbf{r_y = r_t \circ r_a \circ r_b = r_t \circ [r_a \circ r_b] = r_t \circ [r_b(r_a)]}$.

При всей своей внешней простоте, это требование является исключительно важным свойством модели – оно соответствует возможности рассматривать любую область $\mathbf{q_s = \langle p_a\ p_b\ \dots\ p_n \rangle}$ как совокупность её точек $\mathbf{\langle p_a \rangle \dots \langle p_n \rangle \in Q}$, когда взаимодействие точечного пробного тела $\mathbf{q_t}$ с данной областью будет эквивалентно последовательности его взаимодействий с точками данной области: $\mathbf{r_t \circ r_s = r_t \circ r_a \circ \dots \circ r_n}$. Если по числу измерений физического пространства составить **N**-мерный тензор из отображений $\mathbf{r_a \dots r_n}$, разместив их соответственно координатам исходных точек, то такая «кристаллическая решётка» будет наглядно представлять вычислительную модель исходного тела (но что удивительно – любая точка так же может рассматриваться как суперпозиция некоторых тел).

Что ещё более важно, равенство $\mathbf{r_b(r_a) = r_a \circ r_b}$ позволяет поставить в соответствие алгоритмам элементы некоторого равномощного упорядоченного множества **V** (назовём их **собственными числами**) и получать функцию суперпозиции $\mathbf{r_c = r_a \circ r_b}$ с помощью простого отображения $\mathbf{v_c = r_b(v_a)}$, где $\mathbf{v_a, v_c \in V}$ – собственные числа алгоритмов $\mathbf{r_a}$ и $\mathbf{r_c}$. При этом всегда можно подобрать такое соответствие чисел **V** и отображений **R**, чтобы равенство $\mathbf{v_c = r_i(v_a)}$ выполнялось для любых $\mathbf{r_i \in R}$, то есть чтобы собственное число любого алгоритма не зависело от выполняемого преобразования $\mathbf{r_i}$ и всегда однозначно соответствовало своему алгоритму!

Действительно, каждому алгоритму однозначно соответствует некоторая конфигурация $\mathbf{q_i \in Q}$, которая состоит из упорядоченного набора точек $\mathbf{p_i \in P}$. Но тогда такие упорядоченные наборы $\mathbf{q_i}$ и можно объявить собственными числами: поскольку каждое состояние $\mathbf{p_i}$ задаёт значения не более чем счётного множества свойств, каждое из которых допускает измерение и выразимо числом, то состояния $\mathbf{p_i}$ можно представить векторами, которые допускают покомпонентное сравнение, а следовательно, допускают сравнение и наборы $\mathbf{q_i}$. Формальное доказательство, правда только для счётных множеств, можно найти в [[18]] в разделе «Нумерации и операции. Главные универсальные функции». **Теорема доказана.**

Вот такую, мощнейшую, но полную противоречий модель поведения алгоритмов мы и будем использовать в дальнейшем. Не оперируя координатами, она, тем не менее, способна в точности описать расположение и все свойства объектов в пределах произвольной области реального пространства. Координаты здесь используются исключительно для проекции воспринимаемого нами физического пространства на пространство алгоритмов **R** и обратно.

С точки зрения алгоритма совершенно неважно, что представляет собой заключающая его область физического пространства, она может быть как связной, так и разобщённой на разделённые расстоянием части – нагрев нашего клинка мы можем производить на Земле, а охлаждение – на Луне, при условии, что между этими стадиями состояние клинка не будет меняться. И тем не менее, определив некоторую область физического пространства, мы будем точно знать, что она будет из себя представлять после ряда преобразований – какие объекты в ней появятся, какие уйдут, и как изменятся свойства оставшихся. В том числе мы можем получить и координаты объектов в пределах рассматриваемой области, а если такой областью является вся Вселенная – то и координаты любых её объектов.

Однако, при отсутствии дальнейших уточнений, данная модель противоречива – хорошо известно, что мощность множества всех функций всегда больше мощности множества их аргументов, у нас же функции и аргументы – это одно множество **R**. Кроме того, в отсутствие ограничений, модель допускает возникновение тел из вакуума $\mathbf{q_0}$ при $\mathbf{f(q_0) \neq q_0}$, внезапное их исчезновение при $\mathbf{f(g) = q_0}$ и $\mathbf{g(f) = q_0}$, также в ней нарушается принцип локальности взаимодействия, поскольку $\mathbf{f(g \circ h) \neq f(g) \circ f(h)}$. Этими противоречиями мы займёмся чуть позже, а пока хотелось бы рассмотреть один удобный способ записи преобразований, базирующийся на подстановках [[19], [20]].

Допустим, что все отображения $\mathbf{r_i \in R}$ можно перенумеровать натуральными числами (как мы увидим позднее, такое допущение не лишено смысла). Тогда, в силу однозначности, любое отображение будет ставить в соответствие своему прообразу единственный образ, что можно записать в виде: **(3 1 2 …)**, где позиция числа соответствует аргументу, а само число – результату отображения. Этот подход несколько отличается от данного в работе [Error: Reference source not found], но только по форме – мы опускаем первую строку, всегда представляющую собой последовательность **(1 2 3 …)**. В таком случае суперпозиция двух отображений $\mathbf{r_a} = (3\ 1\ 2\ \ldots)$ и $\mathbf{r_b} = (2\ 3\ 1\ \ldots)$ будет вычисляться следующим образом: оба вектора записываются один под другим, причём в случае $\mathbf{r_a} \circ \mathbf{r_b}$, то есть $\mathbf{r_b(r_a)}$, в верхнюю строку попадает $\mathbf{r_a}$. Затем в третьей строке для каждого столбца записываем значение из второй строки, находящееся в позиции, указываемой текущим значением из первой строки. То есть, используем первую строку как индекс для выбора значения из второй строки. Например:

$\mathbf{r_a}$	$= (3\ 1\ 2\ \ldots)$	$\mathbf{r_b}$	$= (2\ 3\ 1\ \ldots)$
$\mathbf{r_b}$	$= (2\ 3\ 1\ \ldots)$	$\mathbf{r_a}$	$= (3\ 1\ 2\ \ldots)$
$\mathbf{r_a} \circ \mathbf{r_b}$	$= (1\ 2\ 3\ \ldots)$	$\mathbf{r_b} \circ \mathbf{r_a}$	$= (1\ 2\ 3\ \ldots)$

Аналогичные преобразования для $\mathbf{r_b} = \mathbf{x^2}$ и $\mathbf{r_a} = \mathbf{2x}$ можно представить следующим образом (значения взяты по модулю **5** как $((\mathbf{x} - 1)\ \mathbf{mod}\ 5) + 1$, чтобы иметь возможность использовать конечные векторы):

$\mathbf{r_a} = 2\mathbf{x}$	$= (2\ 4\ 1\ 3\ 5)$	$\mathbf{r_b} = \mathbf{x}^2$	$= (1\ 4\ 4\ 1\ 5)$
$\mathbf{r_b} = \mathbf{x}^2$	$= (1\ 4\ 4\ 1\ 5)$	$\mathbf{r_a} = 2\mathbf{x}$	$= (2\ 4\ 1\ 3\ 5)$
$(2\mathbf{x})^2$	$= (4\ 1\ 1\ 4\ 5)$	$2\mathbf{x}^2$	$= (2\ 3\ 3\ 2\ 5)$

Нетрудно убедиться, что таким образом можно верно вычислить суперпозицию любых отображений. Следовательно, ***все отображения можно рассматривать как векторы в бесконечномерном пространстве, где операторами преобразования являются сами векторы***. Здесь, кстати, становится очевидным преимущество собственных чисел, введённых в теореме Т. 4 – вместо трудоемкого вычисления суперпозиции $\mathbf{r_a} \circ \mathbf{r_b}$ достаточно было бы выполнить преобразование всего над одним числом – взять компонент вектора $\mathbf{r_b}$ с индексом $\mathbf{v_a}$. Причём, как было доказано, ***собственное число уникально и всегда взаимно однозначно соответствует своему алгоритму, оно всегда одно и то же для любых выполняемых над алгоритмом преобразований***.

Но проблема в том, что во множестве всех функций **F**, определённых на некотором множестве аргументов, для каждой функции всегда образуется замкнутое относительно её суперпозиции подмножество, мощности не выше множества её аргументов. То есть, взяв, например, функцию **sin(x)** и вычисляя её суперпозицию, мы породим множество функций, мощности не выше $|\mathbf{X}|$, и не сможем вычислить ещё как минимум $|\mathbf{F}| - |\mathbf{X}|$ функций. Таким образом, практически любая отдельно взятая функция может подойти на роль алгоритма и породить множество **R** целиком, но вот которой из них оперирует Природа? Отыскать иголку в стоге сена – весьма нетривиальная задача, но этим мы как раз и собираемся заняться прямо сейчас.

Алгоритмически неразрешимые проблемы

Как уже ясно из предыдущего изложения, исключительный интерес для нас представляют алгоритмически неразрешимые проблемы, то есть такие математически строго сформулированные задачи, для которых математически же строго доказано, что алгоритма их решения не существует в принципе. К таким проблемам относятся: проблема самоприменимости (проблема останова), проблема эквивалентности алгоритмов, десятая проблема Гильберта о диофантовых уравнениях и ряд других в различных областях математики. «Брадобрей бреет тех и только тех, кто не бреется сам. Должен ли он брить сам себя?»

Суть этих проблем выражается первой и второй теоремами Гёделя о неполноте, согласно которым во всякой достаточно богатой непротиворечивой теории первого порядка (например, во всякой непротиворечивой теории, достаточно сильной для определения натуральных чисел), существует такая замкнутая формула **F**, что ни сама эта формула **F**, ни её отрицание $\neg\mathbf{F}$ не являются выводимыми в этой теории. В частности, формула **F**, утверждающая непротиворечивость этой теории, не является выводимой в ней. То есть, согласно этим теоремам, всякая достаточно сильная непротиворечивая формальная аксиоматическая теория неполна, что часто ассоциируют с предположением о том, что никакой компьютер не в состоянии воспроизвести человеческий разум.

Из определения $\langle\mathbf{R}, \circ\rangle$ следует, что вычислительное пространство **R** замкнуто относительно любых своих алгоритмов. То есть взаимодействие любых конфигураций этого пространства способно породить только конфигурации этого же пространства. Однако, из этого определения отнюдь не следует, что для любых двух конфигураций $\mathbf{t} \in \mathbf{T} \subset \mathbf{R}$ и $\mathbf{s} \in \mathbf{S} \subset \mathbf{R}$ всегда существует третья конфигурация $\mathbf{u} \in \mathbf{R}$, переводящая конфигурацию **t** в **s**. То есть алгоритм $\mathbf{u}: \mathbf{T} \to \mathbf{S}$ в общем случае может не существовать. Иными словами, вычислительное пространство **R** в общем случае может не являться **автоморфным**.

Теоремы Гёделя о неполноте как раз и утверждают, что любое достаточно сложное пространство **R** не является автоморфным, и не может быть сделано автоморфным добавлением произвольного числа алгоритмов. Согласно первой теореме Гёделя, при добавлении недостающего алгоритма $\mathbf{u}: \mathbf{T} \to \mathbf{S}$ мы получим ситуацию, когда уже для этого алгоритма **u** отсутствует необходимый алгоритм вывода. Мы докажем, что в общем случае это неверно. Но прежде нам необходимо поточнее определиться с понятиями непротиворечивости и полноты применительно к вычислительному пространству.

Интерпретация формальной системы – это метод трактовки формул системы, посредством которого устанавливается взаимосвязь данной формальной системы и содержательной теории, для описания которой она используется. Интерпретация, в которой истинны все выводимые формулы данной системы, есть **модель**. Если формальная система имеет модель, то она является содержательно непротиворечивой, а следовательно – и формально непротиворечивой.

Непротиворечивость или совместимость – свойство формальной системы, состоящее в том, что в рамках неё нельзя вывести противоречие, поскольку не каждая формула этой системы выводима в ней. Иными словами, в рамках непротиворечивой формальной системы не существует такой формулы **F**, что **F** и $\neg\mathbf{F}$ обе выводимы – манипулируя с принятыми символами по правилам совместимой системы, мы никогда не получим два логически противоречивых высказывания.

Полнота или адекватность – свойство формальной системы, состоящее в том, что в рамках неё можно вывести по крайней мере все истинные в заданной интерпретации формулы, представимые по правилам образования данной формальной системы. Иными словами, в рамках полной формальной системы выводима хотя бы одна из формул **F** или $\neg\mathbf{F}$. В отличие от условия непротиворечивости, согласно которому только такие-то и такие формулы являются выводимыми, условие полноты утверждает, что выводимы, по крайней мере, все такие-то и такие формулы [[21], [22]].

В силу взаимно однозначного соответствия состояний физического пространства $\mathbf{q_i} \in \mathbf{Q}$ и отображений $\mathbf{r_i} \in \mathbf{R}$, доказанного в ходе теоремы Т. 4, формальная система $\langle \mathbf{R}, \circ \rangle$ имеет модель в данном выше определении – это реальное физическое пространство. При этом все существующие алгоритмы представляют истинные формулы, а все отсутствующие алгоритмы (алгоритмически неразрешимые проблемы) – ложные формулы. Образно говоря, Природа не ищет смысла, в ней всё сущее – правда, а ничто иное – не выводимо.

Следовательно, в силу существования модели, формальная система $\langle \mathbf{R}, \circ \rangle$ является непротиворечивой. Нам необходимо доказать, что при этом она может быть также и полной в том смысле, что $\langle \mathbf{R}, \circ \rangle$ может являться симметрической полугруппой, а **R** – автоморфным пространством ***всех возможных на нём отображений***. В противном случае нам придётся признать, что в реальном мире существуют объекты, недоступные пониманию и воссозданию из определённого набора других объектов – если мы не имеем в своём распоряжении всего бесконечного множества фундаментальных алгоритмов. Для начала докажем вспомогательную, но важную для нас теорему.

Т. 5

Теорема: В любом неавтоморфном пространстве **R** всегда существует подпространство $\mathbf{T} \subset \mathbf{R}$, замкнутое относительно любых алгоритмов $\mathbf{r_i} \in \mathbf{R}$. Иными словами – в вычислительном пространстве **R** с отсутствующим алгоритмом всегда образуется такое подпространство конфигураций **T**, никакие выходы из которого не возможны в принципе, с какими бы конфигурациями из всего пространства **R** его конфигурации ни взаимодействовали.

Доказательство: Действительно, пусть во всём вычислительном пространстве **R** отсутствует алгоритм $\mathbf{u} \in \mathbf{R}$, переводящий конфигурацию $\mathbf{t} \in \mathbf{T} \subset \mathbf{R}$ в конфигурацию $\mathbf{s} \in \mathbf{S} \subset \mathbf{R}$, где $\mathbf{S} = \mathbf{R} \setminus \mathbf{T}$.

~~кроме единицы: если~~ $\cancel{\mathbf{1} \in \mathbf{R}}$~~, то допустимо, чтобы~~ $\cancel{\mathbf{1} \in \mathbf{T}}$ ~~и~~ $\cancel{\mathbf{1} \in \mathbf{S}}$ ~~– это~~

Допустим, что существует какая-либо цепочка алгоритмов, отличных от **u**, переводящая конфигурацию **t** в **s**. Но тогда эта цепочка может быть названа алгоритмом **u**, что противоречит условию, по которому такого алгоритма не существует. Следовательно, никакой такой цепочки не существует, и пространство **R** действительно содержит замкнутое подпространство **T**, никакие выходы из которого невозможны.

Кроме того, замкнутое подпространство **T** не способно само или при взаимодействии с остальным подпространством **S** сконструировать такую конфигурацию, с помощью которой его собственные алгоритмы смогли бы вычислить алгоритмы подпространства **S**. Если допустить такую возможность, это будет означать, что некоторая цепочка алгоритмов из **T** и **S** способна привести к переходу из **T** в **S**, но, как мы уже доказали, такой цепочки не существует. **Теорема доказана.**

Следствие: Если в вычислительном пространстве **R** отсутствует также алгоритм обратного преобразования конфигурации **s** в **t**, такое пространство разбивается на два замкнутых относительно любых алгоритмов из **R** подпространства **S** и **T**, никакие переходы между которыми невозможны.

А теперь, возвращаясь к теме, докажем, что теорема Гёделя в общем случае несправедлива.

Т. 6

Теорема: Существуют такие сложные по Гёделю вычислительные пространства, для которых теорема Гёделя о неполноте является неверной.

Доказательство: Введём на вычислительном пространстве **R** соответственно его размерности систему координат, каждую элементарную ячейку в этих координатах назовём точкой.

Тогда отсутствие какого-либо алгоритма в неавтоморфном пространстве с топологической точки зрения будет являться «выколотой точкой». Иными словами, в силу теоремы Т. 5 пространство **R** будет иметь некоторую топологическую **связность**[2], большую минимальной [[23]].

Но тогда мы всегда сможем дополнить это пространство **R** недостающей точкой, спроецировав его вместе с такой точкой на пространство **R+** с меньшей связностью (так, сфера $\mathbf{P_0}$ гомеоморфна проективной комплексной плоскости $\mathbf{N_1}$ с выколотой точкой, которая в свою очередь гомеоморфна обычной плоскости $\mathbf{P_1}$, дополненной бесконечно удалёнными точками. Двумерные поверхности рассматриваются здесь только для примера, наша модель не ограничивает размерность вычислительного пространства).

Таким же образом мы можем добавлять и остальные недостающие точки, пока не достигнем автоморфного пространства **R0** наименьшей связности. То есть простым добавлением недостающих алгоритмов любое неавтоморфное вычислительное пространство **R** можно привести к автоморфному пространству **R0**, в котором в силу теоремы Т. 5 будут присутствовать все необходимые алгоритмы для преобразования любой конфигурации из **R0** в любую другую конфигурацию из **R0**.

А поскольку мы не делали никаких предположении о содержимом алгоритмов, они могут представлять в том числе и сложную по Гёделю логику второго порядка, как и всё наше знание, количество правил которого конечно. Следовательно связность такого пространства также будет конечной, и мы сможем преобразовать его в автоморфное пространство за конечное число шагов. **Теорема доказана.**

Следствие: Евклидово или любое другое многосвязное вычислительное пространство не является автоморфным, то есть оно алгоритмически неразрешимо и непознаваемо. В частности, противоречивым является диагональный метод доказательства Кантора.

Учитывая то, что определение алгоритма $\mathbf{r_i}$**:** $\mathbf{R} \rightarrow \mathbf{R_i}$ при отсутствии ограничений на $\mathbf{R_i}$ приводит к понятию множества всех подмножеств (однако, понятия пустого множества в нём нет), зададимся

---

[2] Говоря о связности, я буду подразумевать **порядок топологической связности** – инвариантную характеристику ограниченной поверхности, гомеоморфной данной. По существующей классификации все топологические поверхности подразделяются на классы: $\mathbf{P_0}$ (или **S**) – сфера, $\mathbf{P_1 \ldots P_K}$ – сфера с **K** дырами (или ручками), $\mathbf{N_1 \ldots N_K}$ – сфера с **K** дырами, «заклеенными» листами Мебиуса, при этом два листа Мебиуса можно представить одной ручкой (но хотя бы один лист Мебиуса должен сохраняться). Поверхности **P** являются ориентируемыми, а все поверхности **N** – односторонними, неориентируемыми.

вопросом, а непротиворечивым ли является само определение $\langle \mathbf{R}, \circ \rangle$ в том смысле, что добавление новых алгоритмов не приводит к появлению новых «выколотых точек»? Можно доказать, что оставаясь в евклидовом пространстве добавление новых алгоритмов приводит к неограниченному росту количества новых дыр, и что в автоморфном пространстве такое добавление не приводит к их появлению. Мне представляется, что противоречия здесь нет, ибо чтобы преобразовать, скажем, евклидово пространство $\mathbf{P}_1$ в риманово $\mathbf{P}_0$, мы должны добавить точки вне бесконечного евклидова пространства, выйдя за пределы этого пространства.

Т. 7

Теорема: Автоморфное вычислительное пространство **R0** остается автоморфным относительно любых своих алгоритмов. Любое неавтоморфное вычислительное пространство **R** имеет потенциально бесконечную связность – в этом смысле риманово, евклидово, гиперболическое и все иные неавтоморфные вычислительные пространства эквивалентны.

Доказательство: Суть проблемы на интуитивном уровне можно выразить следующим образом. Пусть мы имеем евклидово вычислительное пространство **R** с тремя базовыми алгоритмами, назовём их «**+0**», «**+1**» и «**+2**». По определению алгоритм есть отображение множества **R** на себя, само принадлежащее этому множеству **R**. Тогда мы можем задать каждое отображение в виде таблицы, она же будет являться графиком данного отображения. Пусть исходные три отображения задаются следующими графиками (будем внизу изображать аргумент, а слева – результат отображения, как это принято на координатной плоскости X-Y).

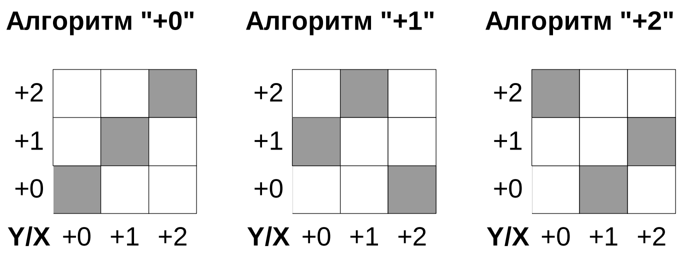

Рис. 5

Однако, возможны ещё двадцать четыре дополнительных отображения между алгоритмами «**+0**», «**+1**» и «**+2**», отсутствующие среди приведённых графиков! Приведём их все, имея ввиду, что вертикальные графики невозможны по определению однозначного отображения. ***То есть наличие нескольких закрашенных ячеек в одном столбце невозможно по определению.***

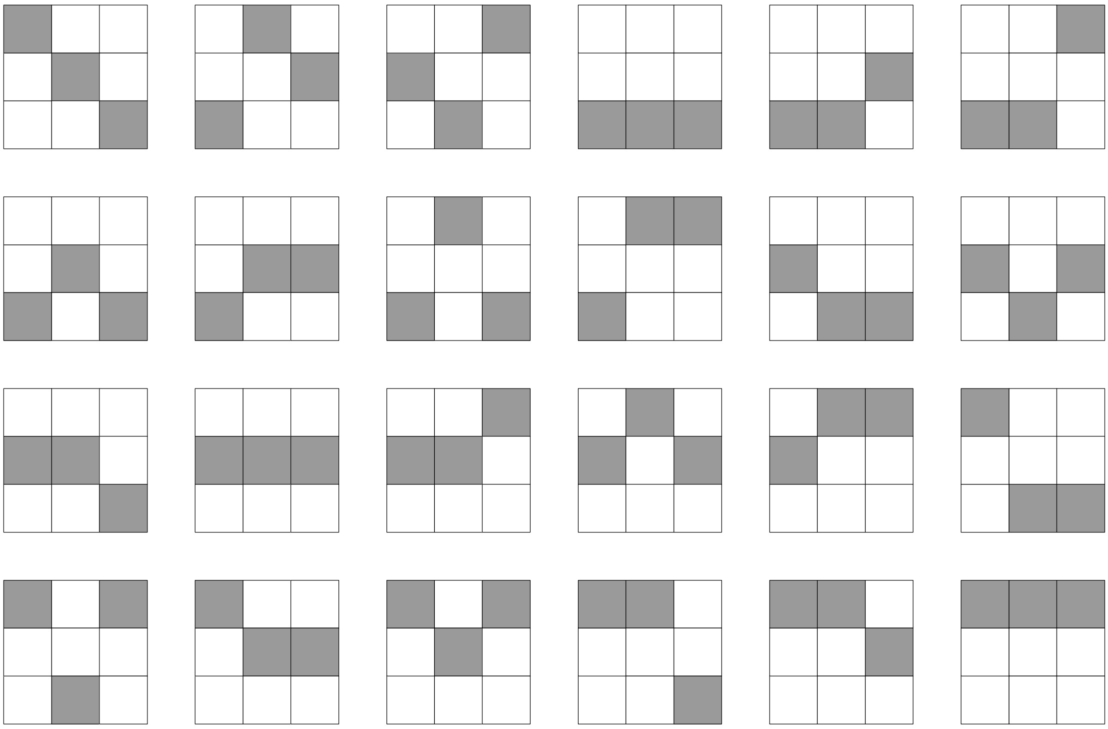

Рис. 6

Получается, что есть ещё двадцать четыре алгоритма, которые мы не можем задать, а значит вычислить, в евклидовом пространстве. Конечно, мы их можем добавить в вычислительное пространство **R**, но тогда мы будем иметь таблицы уже из $\mathbf{27^2 = 729}$ клеток, и число невычислимых конфигураций возрастет ещё более – отношение вычислимых алгоритмов к всем возможным (назовём это отношение **полнотой**) убывает как $\mathbf{N / N^N}$, и уже при десяти вычислимых алгоритмах мы будем иметь как минимум миллиард невычислимых!

~~Как мы показали в теореме Т. 5 связность вычислительного пространства **R** (то есть число «выколотых точек») растёт пропорционально числу невычислимых в нём алгоритмов, следовательно, связность любого неавтоморфного пространства **R** как множества всех возможных алгоритмов будет бесконечной,~~ а мощность множества невычислимых алгоритмов в нём будет всегда больше мощности множества вычислимых алгоритмов. Уже физическое евклидово пространство двух алгоритмов имеет ещё два невычислимых! То есть оно уже должно распасться на два независимых автоморфных подпространства из одного алгоритма или увеличить свою связность до бесконечности.

Значит ли это, что существование неавтоморфных вычислительных пространств невозможно? Нет, поскольку реальное вычислительное пространство в отличие от его математической абстракции **R** имеет фактор времени. И какое-то время оно может существовать, но в конечном итоге такое пространство распадётся на множество замкнутых подпространств, каждое из которых будет автоморфным.

Теперь рассмотрим алгоритмы в римановом вычислительном пространстве **R**. Зададим графики отображений в виде таблиц на сфере, аналогично тому, как мы это делали в евклидовом пространстве – таблица образуется перпендикулярными прямыми (на сфере – окружностями максимального радиуса) и занимает всё пространство. Для наглядности изобразим таблицу для пяти алгоритмов – она будет иметь пять строк и пять столбцов. На рисунке таблица дана лишь схематично, однако такого изображения достаточно, чтобы представить существо дела.

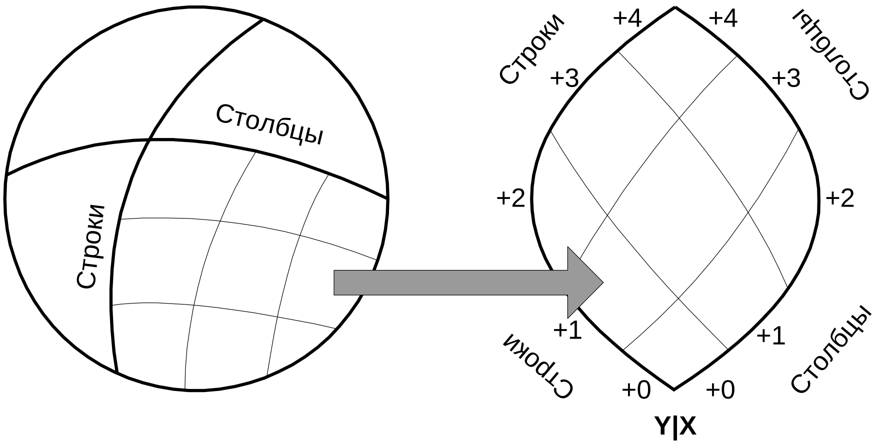

Рис. 7

Развернув нашу таблицу как показано на Рис. 7 мы увидим, что на ней строки переходят в столбцы и наоборот! Следовательно, ***мы уже не можем ставить несколько закрашенных ячеек даже в одной строке***. В противном случае такое отображение будет ставить в соответствие одному аргументу два или более результатов. Таким образом, все графики, изображённые на Рис. 6 кроме первых трёх будут недопустимы по определению отображения. Дадим ещё одно схематичное представление таблицы на сфере, верное, тем не менее, в рассматриваемой далее области.

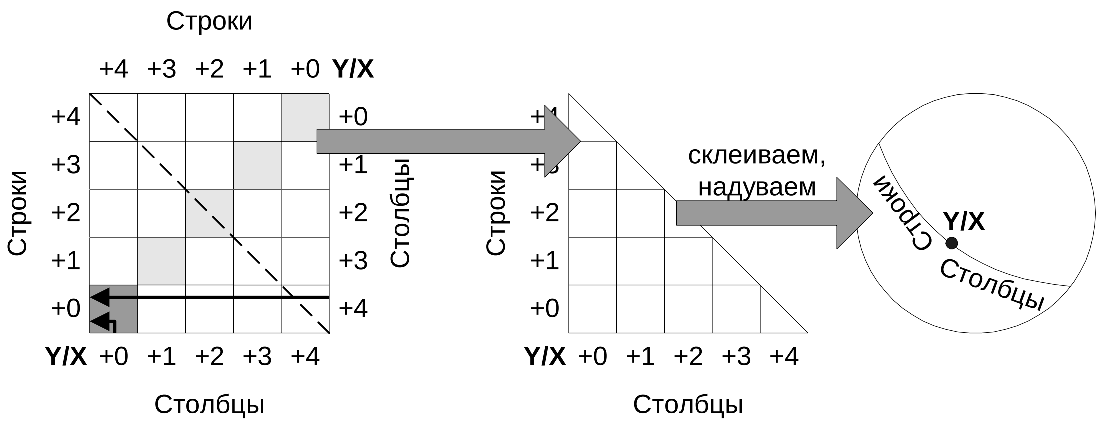

Рис. 8

Так как столбцы таблицы в римановом пространстве соответствуют строкам, то задавая какую-то одну строку в таблице графика отображения мы тем самым одновременно задаём и соответствующий столбец в этой же таблице. То есть мы одновременно строим два отображения – «мир» и «антимир», развёрнутых друг относительно друга на девяносто градусов. Задавая преобразование «**+0**» $\longrightarrow$ «**+0**», мы тем самым задаём и преобразование «**+4**» $\longrightarrow$ «**+0**» – как это показано на Рис. 8. Как можно видеть, оставшиеся три первых графика на Рис. 6 как раз и являются такими развёрнутыми графиками алгоритмов «**+0**», «**+1**» и «**+2**», изображённых на Рис. 5!

Однако, что ещё более важно, закрашенная клетка на Рис. 8 задаёт на самом деле не два, а сразу четыре преобразования: также задаются «**+4**» $\longrightarrow$ «**+4**» и «**+0**» $\longrightarrow$ «**+4**» (стрелки можно продолжить и на верхний ряд строк). На первый взгляд получается, что любое отображение в римановом вычислительном пространстве ставит в соответствие одному аргументу два результата. Значит ли это, что понятие однозначного отображения на сфере вообще не имеет смысла? Нет не значит, ***причём есть даже три различных способа трактовать эту ситуацию!***

Во-первых, можно сказать, что мы уже не можем использовать целые вещественные числа для представления алгоритмов. Поскольку каждая клетка таблицы ставит в соответствие каждым двум аргументам ровно два результата (а не множество, если будем закрашивать строки или столбцы), то для обозначения алгоритмов мы можем применить целые комплексные числа. Условимся считать прямоугольную стрелку ответственной за вещественную, а проходящую напрямую стрелку – за мнимую часть комплексного числа, тогда обозначенные на Рис. 8 преобразования алгоритмов можно записать в виде: «**+0**» **→** «**+0+4i**» и «**+4**» **→** «**+4+0i**» – эти преобразования единственны в силу симметричности любого графика отображения на сфере относительно диагонали. Таким образом, после введения комплексных чисел для представления алгоритмов отображение опять стало однозначным.

Но как по этим правилам представить, скажем, отображение алгоритма «**+1+4i**»? Так как комплексное число можно разложить на вещественную и мнимую составляющие, то его отображением будет являться композиция отображений вещественной «**x**» и мнимой «**y**» части: **z = x + iy**. Так, если следовать преобразованию на Рис. 8, то при отображении алгоритма «**+1+4i**» мы получим две составляющие: «**+1+3i**» – для вещественной, и «**+0+4i**» – для мнимой части исходного алгоритма. Тогда композиция этих составляющих будет равна: «**+1+3i**» **+ i *** «**+0+4i**» **=** «**-3 + 3i**». В этом случае график отображения должен быть определён также и на отрицательных коэффициентах, что не меняет существа дела – на том же Рис. 8 мы можем заменить индексацию столбцов и строк с **+0…+4** на **–2…+2**.

Во-вторых, мы можем заимствовать из физики идею квантовой суперпозиции состояний системы, согласно которой эта система может находиться как в чистых, факторизованных состояниях, так и в некоторой их суперпозиции – запутанном состоянии. Для представления алгоритмов необходимо использоваться векторы в гильбертовом пространстве размерности, равной числу возможных состояний системы. В нашем случае системой является каждый отдельный алгоритм, а его возможными состояниями – все алгоритмы вычислительного пространства **R**, число которых всегда бесконечно.

Тогда преобразования алгоритмов на Рис. 8 можно записать в виде: **|**«**+0**»**⟩ →** $\mathbf{C_1}$ **|**«**+0**»**⟩ +** $\mathbf{C_2}$ **|**«**+4**»**⟩** и **|**«**+4**»**⟩ →** $\mathbf{C_1}$ **|**«**+4**»**⟩ +** $\mathbf{C_2}$ **|**«**+0**»**⟩**, где $\mathbf{C_1 \ldots C_N \ldots}$ – амплитуды вероятностей того, что алгоритм находится в соответствующем состоянии (в данном конкретном случае все остальные компоненты векторов состояний будут нулевые). При таком преобразовании каждый алгоритм теряет детерминированность, как бы «размывается», представляя одновременно множество алгоритмов в большей или меньшей степени. Точно таким же образом размывается и его график отображения – вместо одной чистой линии мы имеем в данном случае две, а в общем – множество «менее чётких» линий: в каждой клетке будет стоять не единица, как раньше, а комплексное число – амплитуда вероятности для прямоугольной стрелки (вещественная часть амплитуды) и для стрелки, проходящей напрямую (мнимая часть).

То есть, график отображения становится аналогом матрицы амплитуд в квантовой механике с той лишь разницей, что матричные операции в нашем случае не были введены. Однако, учитывая то, что матрица амплитуд является линейным оператором на множестве векторов состояний, а значит – отображением, график отображения алгоритмов всегда можно преобразовать к соответствующему матричному виду. Следовательно, наша ***алгоритмическая модель эквивалентна формализму квантовой механики*** в предположении справедливости механизма квантовой суперпозиции.

В-третьих, и в главных, мы можем объявить два предыдущих случая всего лишь ***проекцией*** риманова пространства на плоское евклидово пространство! На самом деле площадь меньше.

Следовательно, в автоморфном вычислительном пространстве **R0** количество всех возможных графиков отображений в точности соответствует числу алгоритмов, а отношение вычислимых алгоритмов к всем возможным **N / N** не убывает, а всегда равно единице.

ln(n^n / n!) = n, вернее n^n / n! = e^(n – ln(sqrt(2*pi*n))) = e^(n / sqrt(2*pi*n))

Cn m = N! / [M! * (N – M)!]

Конец 1.Set.pdf: ΣCn m = 2^n, суммирование по m = 0..n

Рассмотрим также заслуживающее особого внимания одностороннее проективное пространство $\mathbf{N_1}$ – по порядку связности оно является ближайшим соседом сферы – топологического пространства $\mathbf{P_0}$ (см., например, [Error: Reference source not found]). Так как на проективной плоскости любые параллельные прямые пересекаются (в бесконечности), следовательно, так же как и на сфере, перпендикулярные прямые должны переходить в параллельные, то есть ситуация с взаимным переходом строк и столбцов друг в друга также имеет место. Однако, на проективной плоскости отсутствует поворот графиков на 90 градусов, то есть, задавая отображение «мир» мы уже не задаем «антимир», а дублируем то же самое отображение!

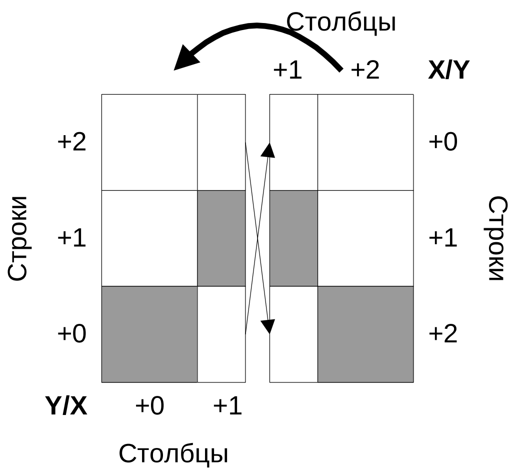

Рис. 9

Итак, автоморфное пространство **R0** является возможным, непротиворечивым и полным (**N / N = 1**), неориентируемое проективное пространство имеет полноту $\mathbf{N / (N^2 – N)}$ и непротиворечиво только при двух алгоритмах, евклидово пространство имеет полноту $\mathbf{N / N^N}$ и противоречиво всегда, кроме случая единственного алгоритма, когда оно становится автоморфным.

Теорема доказана.

Следствие: Все множества в автоморфном вычислительном пространстве являются счётными, а само автоморфное пространство – дискретным ***(квантованным)***. Действительно, в евклидовом пространстве множество функций имеет мощность алеф, следующую за мощностью множества аргументов, мы же показали, что в автоморфном пространстве мощности обоих множеств совпадают.

Геометрия автоморфного пространства

В силу того, что выводы теоремы Т. 7 представляют самостоятельную ценность, мне представляется интересным остановиться на следствиях более подробно.

Таким образом, неавтоморфное вычислительное пространство **R** всегда разбивается на подпространства, преобразования конфигураций между которыми невозможны в одну или обе стороны. Подпространства не разделены «территориально», поэтому взаимодействие между ними всё же возможно. Реальное динамическое пространство, если в нём сохраняется количество движения, всегда придёт к состоянию с полностью независимыми, но не изолированными друг от друга подпространствами.

Действительно, если преобразование невозможно только в сторону **S**, но возможно в сторону **T**, то почти все конфигурации подпространства **S**, взаимодействуя с такой «чёрной дырой», рано или поздно будут преобразованы в конфигурации подпространства **T**, возможно кроме самой «чёрной дыры» – той совокупности конфигураций, которая необходима и достаточна для такого преобразования – если она не принадлежит **S** и не способна преобразовать сама себя. Весь этот процесс «территориально» будет происходить в одном и том же пространстве **R**, при этом для **S** он будет представляться уменьшением, а для **T** – увеличением потенциального «разнообразия», то есть числа возможных состояний пространства. Таким образом, в итоге мы будем иметь либо одно, либо два подпространства, замкнутых относительно любых конфигураций из **R**.

При взаимодействии двух конфигураций **s** и **t**, принадлежащих к разным таким подпространствам **S** и **T**, конфигурация **s** может быть преобразована только в конфигурацию из своего подпространства **S**, так же как и конфигурация **t** может быть преобразована только в конфигурацию из **T**. При этом для конфигурации **s** её взаимодействие с **t** будет эквивалентно взаимодействию с какой-то «своей» конфигурацией из **S**, так же и **t** будет полагать, что взаимодействует с какой-то знакомой конфигурацией из своего подпространства **T**. Однако, в общем случае будет существовать множество таких конфигураций из **T**, взаимодействие с которыми для **s** будет эквивалентно взаимодействию с единственной конфигурацией из **S**, тоже самое и для **t** – различные конфигурации из **S** будут для неё неразличимы. То есть подпространства **S** и **T** в общем случае не будут гомоморфны.

«Чёрная дыра» здесь и в математическом, и в физическом смысле в кавычках, ибо те чёрные дыры, которые предсказывает физика, могут и не являться однонаправленными алгоритмами в определении нашей модели. Но у меня просто нет никаких других подходящих ассоциаций для этого понятия. Ассоциация не вполне верна потому, что, как я уже сказал, системы разных подпространств могут взаимодействовать между собой, но они этого не поймут – для них это будет взаимодействием с одной или несколькими системами своих подпространств. В физическом пространстве в качестве «чёрной дыры» скорее мог бы рассматриваться процесс необратимого преобразования вещества в излучение или какой-то другой вид материи, если он имеет место быть в глобальном масштабе.

В ходе доказательства теоремы Т. 6 мы предполагали, что существует некий «демон», вставляющий недостающие точки, то есть задающий отсутствовавшие конфигурации пространства **R**. Однако, на самом деле такого демона мы не знаем, следовательно неавтоморфное пространство **R** не способно само стать автоморфным пространством **R0** никакими способами – это мы доказали в теореме Т. 5 (далее мы будем в основном употреблять обозначение **R**, говоря о вычислительном пространстве в общем виде, оно в том числе может быть и автоморфным).

В разделе «» я покажу, что мысль и идея с точки зрения нашей модели материальны, поэтому вопрос – кроется ли причина неразрешимости существующих теорий в нашей математике или же в физике нашего реального пространства – представляется отнюдь не праздным. Если наш мир евклидов, то мы в принципе не сможем его познать никогда, в принципе не сможем построить теории, объясняющей все факты реального мира. Физические параллели нашей модели я проведу в разделе «Интеллект в реальном мире», а пока попробуем рассмотреть вопрос с точки зрения математики.

Прежде всего вопрос – почему и каким образом даже сейчас у нас существуют непротиворечивые теории, типа булевой алгебры? Булеву алгебру можно рассматривать как своё пространство «**R**», хотя на самом деле правильней рассматривать её как подпространство единого вычислительного пространства **R**, потому что это глобальное пространство может накладывать ограничения на теорию, являясь, например, евклидовым.

Итак, допустим, что наше реальное глобальное пространство – евклидово. Как же в нём существует непротиворечивая булева алгебра? Ответ в том, что набор её алгоритмов ограничен относительно всех возможных алгоритмов из **R**. Можно представить идеальную модель, поясняющую ситуацию. Пусть $\mathbf{s_1}$ (как и $\mathbf{s_2}$) – конфигурация подпространства **L** пространства **R** ($\mathbf{s_1}, \mathbf{s_2} \in \mathbf{L} \subset \mathbf{S} \subset \mathbf{R}$), преобразующая конфигурацию $\mathbf{s_2}$ в конфигурацию $\mathbf{s_1}$, пусть $\mathbf{s_2}$ в свою очередь обладает тем же свойством, то есть $\mathbf{s_1} = \mathbf{s_1(s_2)}$ и $\mathbf{s_2} = \mathbf{s_2(s_1)}$. Тогда при любых взаимодействиях этих двух конфигураций между собой будут получаться конфигурации либо $\mathbf{s_1}$, либо $\mathbf{s_2}$. То есть подпространство **L** будет локально автоморфным.

Однако, при любом взаимодействии $\mathbf{s_1}$ или $\mathbf{s_2}$ с другими алгоритмами из полного пространства **R** конфигурации $\mathbf{s_1}$ или $\mathbf{s_2}$ может и не получиться, мы выйдем в пространство **R**, которое евклидово и неавтоморфно. Заметим, что такой выход возможен только в то подпространство **S** пространства **R**, которое является «своим» для $\mathbf{s_1}$ и $\mathbf{s_2}$, то есть в недоступную область **T** у конфигурации выхода нет. Этим и отличается локальная автоморфность – она может быть нарушена при взаимодействии с другими алгоритмами.

Думаю, что на поставленный вопрос мы ответили. Точно таким же образом некоторая формальная система может быть противоречивой даже в автоморфном вычислительном пространстве. Она противоречива только потому, что неполна – в ней не хватает какого-то необходимого правила. Причём именно правила, а не постулата, в том смысле, что данное правило должно выводиться из других правил этой теории.

Так, в математике, очевидно, евклидова числовая ось, это приводит к понятию бесконечности. И если плюс и минус бесконечность ещё как-то участвуют в операциях, то уж произведение нуля на бесконечность – никак. Это **NaN**, «not a number», «не число». Оно и является той «чёрной дырой» в математике, при взаимодействии с которой все наши правила уходят в никуда (кстати, в булевой алгебре понятия бесконечности как раз и нет). В соответствии с выводами теоремы Т. 7, математика в автоморфном пространстве будет непротиворечивой. Только вот если наше физическое подпространство не является автоморфным и в нём существует реальная «чёрная дыра», такая математика не будет способна иметь дело с этим фактом реального мира.

Существование алгоритмически неразрешимой проблемы означало бы, что имея предмет, мы были не в состоянии выяснить какие-то из его свойств.

В автоморфном пространстве всё не столь мрачно. С точки зрения любой системы в этом пространстве оно и является полным вычислительным пространством **R**. Даже если и существует объемлющее пространство, любое взаимодействие с его системами будет эквивалентно взаимодействию с какими-то «своими» системами. На уровне абстракции нашей модели автоморфное подпространство ничем не отличается от объемлющего автоморфного пространства, кроме количества состояний – то есть разнообразия возможных алгоритмов. При этом мы не сможем обнаружить отсутствующие конфигурации (это означало бы, что мы вычислили алгоритм – «выколотую точку»). Для нас конфигурация в любой точке вычислительного пространства будет плавно изменяться от своего наименьшего до наибольшего значения и обратно, просто в объемлющем пространстве эти рамки будут шире. Реальные же законы подпространства будут определяться алгоритмами, возможными в нём, но это уже находится за рамками нашей модели (но не за рамками современной физики – см. «Интеллект в реальном мире»).

Таким образом, в автоморфном подпространстве мы можем просто забыть о существовании других подпространств, а наше собственное пространство мы будем познать в состоянии. В автоморфном пространстве на вопрос: «Может ли брадобрей брить сам себя?» – я могу дать конкретный ответ: «Не знаю». Надо только придумать правило связи этого ответа с другими понятиями.

А сколько всего существует подпространств, то есть скольких алгоритмов мы не досчитываемся? Для ответа на этот вопрос предлагаю рассмотреть ситуацию с точки зрения бота, живущего в предложенном выше пространстве $\mathbf{L = \{s_1\ s_2\}}$. Пусть мы ему дали сосчитать наше число **5**, и он сосчитал: 1, 2, 1, 2, 1. Таким образом, число недостающих алгоритмов для него будет равно одному. Или двум…

Интеллект в реальном мире

~~Как уже было определено в предыдущем разделе, с точки зрения нашей модели в реальном мире вычислительным пространством является реальное пространство. Алгоритмом является материя, представляющая собой искривление пространства. Процессом вычисления является изменение координат материи в процессе её движения в искривленном пространстве.~~

С точки зрения нашей модели в реальном мире вычислительным пространством является реальное пространство, алгоритмом является материя, представляющая собой искривление пространства, а процессом вычисления является изменение координат материи в процессе её движения в искривленном пространстве. Наша модель не претендует на полное описание физики нашего мира, поскольку мы не вводим быстродействия вычислительного пространства и правил преобразования конфигураций. Однако, такая задача и не ставилась, тем более, что в современной физике есть свои модели, с которыми мы проведём параллели в разделе «».

Понятие алгоритма как кривизны пространства очень тесно коррелирует с понятием симметрии в Стандартной модели, понимающей поле как компенсирующий эффект, восстанавливающий симметрию (инвариантность) при локальном калибровочном преобразовании (теорема Нётер, [[24]]). В Струнных теориях соответствие ещё более точно – нужно только колебание струны назвать алгоритмом. Представляется, что в рамках этих теорий возможно объединение математического и физического понятия алгоритма, в таком случае наша модель может стать вспомогательной на пути такого объединения.

В частности, построив автоморфную математику, может оказаться возможным принципиально устранить расходимости в физических теориях, в силу теоремы Т. 7 неперенормируемые теории попросту не будут возникать – они живут только в нашей евклидовой математике, а в автоморфном физическом пространстве они отсутствуют так же, как отсутствуют множества с мощностью большей алеф-0. В пользу того, что реальное пространство – автоморфно, говорит тот факт, что оно дискретно (квантуемо) и множество его точек счётно. Тогда на вопрос «почему природа так добра, что использует только перенормируемые взаимодействия?» [Error: Reference source not found, с. 96] можно дать ответ – потому что она автоморфна! В этом отношении дальнейшее развитие множеств в автоморфном пространстве выглядит весьма многообещающе.

С этой позиции важными также выглядят наши выводы о связи алгоритмической неразрешимости с глобальной кривизной (то есть связностью) реального пространства – по сути обнаруживается удивительная связь строения микромира и макромира. Также интересно описание «чёрных дыр», как алгоритма, необратимо преобразующего одни конфигурации пространства в другие, а также сам вид этих «чёрных дыр», более похожих на энтропийные процессы в случае их необратимости, нежели на предсказываемые гравитационные чёрные дыры.

локальная автоморфность - сверпроводники

Интересно заключение о возможном сосуществовании и взаимной изолированности разных миров, когда они существуют бок о бок, но не распознают друг друга. С этой точки зрения универсальность Струнных теорий не является недостатком, ибо конкретный вид законов определяется набором возможных алгоритмов (диапазонов амплитуд и частот) в каждом конкретном мире. Как было сказано ранее, никаких разрывов в этих диапазонах мы обнаружить не сможем, они просто будут для нас ограничены конечным набором значений. Все эти заключения были сделаны нами ранее, начиная с раздела «Алгоритм как конфигурация».

Как следует из названия данного раздела, здесь я попытаюсь сформулировать некоторые характерные черты, позволяющие выделить интеллектуальные системы реального мира среди всех прочих. Поскольку наше определение интеллектуальной системы дано строго, не следует удивляться наличию других, небиологических интеллектуальных систем, несмотря на то, что зачастую это будет противоречить нашему «здравому смыслу». Тем более, что на поверку такие небиологические системы по степени интеллекта могут оказаться «глупее» любого, самого примитивного микроорганизма.

Как было установлено в разделе «Возникновение интеллекта», необходимым и достаточным (в отсутствие стохастических систем) условием возникновения интеллектуальной системы является наличие в её составе нелинейных динамических систем с сильными обратными связями, охватывающими эти системы.

Отсюда немедленно следует, что ***обнаружить следы интеллекта в рамках линейных теорий, в том числе в рамках Квантовой механики не представляется возможным***, несмотря на то, что именно с принципиальной случайностью квантовых процессов ряд специалистов связывает надежды на появление интеллекта (см., например, [[25]]). Я думаю, мы уже определились, в рамках какой теории стоит искать появление интеллекта с точки зрения нашей модели – в теории нелинейных динамических систем (в качестве введения можно посмотреть [[26]]).

Под динамической системой в физике понимается любой объект или процесс, для которого однозначно определено понятие состояния как совокупности некоторых величин в данный момент времени и задан закон, который описывает изменение (эволюцию) начального состояния с течением времени. Этот закон позволяет по начальному состоянию прогнозировать будущее состояние динамической системы, его называют законом эволюции. Математическая модель динамической системы считается заданной, если введены параметры (координаты) системы, определяющие однозначно её состояние, и указан закон эволюции [[27]].

Если наш реальный мир неавтоморфен, то существует поверхность меньшей связности, воспринимаемая в нашем неавтоморфном пространстве как точка. Однако, как мне кажется, я не должен был здесь ошибиться в своих построениях. Косвенно это подтверждает тот факт, что мы способны различать античастицы, но с другой стороны – всё вокруг превращается в фотоны…

либо наше представление, либо мир – кривые.

Степень интеллекта

Система взаимодействия трех тел

Вирус и геном являются интеллектом, но не являются интеллектуальной системой – это «интеллект в чистом виде». – являются, но «не в том пространстве».

Математика интеллекта

Для того, чтобы перекинуть мостик от теории к её приложениям, попытаемся по возможности строго обобщить предыдущие выводы и сформулировать дополнительные принципы построения интеллектуальных систем.

Прежде всего, соотнесем понятия алгоритма и функции. Алгоритм не является тождественным понятию функции, ибо функция в математике является отображением $\mathbf{F: X \to Y}$ некоторого множества аргументов $\mathbf{X}$ на множество результатов $\mathbf{Y}$, причём само это отображение $\mathbf{F}$ рассматривается как нечто внешнее по отношению к $\mathbf{X}$ и $\mathbf{Y}$, то есть не входит в общем случае в множества $\mathbf{X}$ или $\mathbf{Y}$. Алгоритм же, как конфигурация вычислительного пространства, представляет собой совокупность самой функции и её аргументов. Следовательно, алгоритм может рассматриваться как функция $\mathbf{R: R \to R}$, отображающая множество $\mathbf{R}$ само в себя, причём эта функция сама принадлежит этому множеству $\mathbf{R}$.

Налицо несоответствие понятий алгоритма и функции. И тем не менее, нам весьма желательно получить функциональное представление алгоритма, чтобы иметь возможность привлечь к решению аппарат функционального анализа. Между тем, данное несоответствие понятий уже в какой-то мере решено в теории автоматов. Как я уже упоминал в разделе «Модель активной системы», оно обходится введением понятия внутренних состояний системы: $\mathbf{(Y, S) = F(X, S)}$, что является суррогатным представлением записи: $\mathbf{(Y, F) = F(X)}$. Однако, возможность такого представления ещё необходимо доказать.

Т. 8

Теорема: Любой алгоритм вычислительного пространства $\mathbf{R}$ представим в виде некоторой функции от аргументов и параметров.

Доказательство: Приведём формальное определение функции: отображение $\mathbf{F: X \to Y}$ множества $\mathbf{X}$ в множество $\mathbf{Y}$ есть подмножество $\mathbf{F \subseteq X \times Y}$ прямого произведения множеств $\mathbf{X}$ и $\mathbf{Y}$, такое что для любого элемента $\mathbf{x \in X}$ существует единственный элемент $\mathbf{y \in Y}$, такой что $\mathbf{(x, y) \in F}$. Тогда функцию с параметрами можно определить следующим образом: $\mathbf{F: X \times S \to Y \times T}$, где $\mathbf{S}$ и $\mathbf{T}$ – отдельные множества входных и выходных параметров, отличные от самого отображения $\mathbf{F}$, и множеств входных $\mathbf{X}$ и выходных $\mathbf{Y}$ аргументов, при этом $\mathbf{F \subseteq X \times S \times Y \times T}$ где $\mathbf{T \subseteq S}$.

Для доказательства нам необходимо показать, во-первых, что для любых множеств $\mathbf{X}$ и $\mathbf{Y}$, где $\mathbf{Y \subseteq X}$, существует такая функция $\mathbf{F: X \times S \to Y \times T}$, которая при фиксированных значениях параметра $\mathbf{S}$ способна служить отображением $\mathbf{F: X \to Y}$, и во-вторых, что эта функция $\mathbf{F}$ при изменении параметра $\mathbf{S}$ способна отобразить множество $\mathbf{X}$ в множество $\mathbf{Y}$ произвольным образом, то есть так составить пары $\mathbf{(x, y)}$, чтобы для любого $\mathbf{x \in X}$ и любого $\mathbf{y \in Y}$ всегда нашёлся бы такой $\mathbf{s \in S}$, чтобы $\mathbf{F(s): x \to y}$.

Доказательство первого утверждения очевидно – так как параметр $\mathbf{S}$ не меняется, то множество $\mathbf{S}$ будет состоять из одного элемента, а прямое произведение $\mathbf{X \times S}$ будет равномощно самому множеству $\mathbf{X}$. Поэтому мы всегда можем выбрать такую функцию $\mathbf{G: X \times S \to X_1}$, которая задаёт биективное (взаимно однозначное) отображение множества $\mathbf{X}$ самого в себя. А определив функцию $\mathbf{H: X_1 \to Y}$, мы тем самым определяем и функцию $\mathbf{F: X \to Y}$, где $\mathbf{F = G \circ H}$ (суперпозиция функций $\mathbf{H}$ и $\mathbf{G}$).

Доказательство второго утверждения будем строить с использованием понятия графика отображения $\mathbf{F}$ как множества упорядоченных пар $\mathbf{\{(x, y) \mid x \in X, y = F(x)\}}$ – для наглядности можно ориентироваться на Рис. 5. Введём на вычислительном пространстве $\mathbf{R}$ соответственно его размерности систему координат, каждую элементарную ячейку в этих координатах назовём точкой $\mathbf{x \in X}$, где $\mathbf{X}$ – биективное отображение множества $\mathbf{R}$. Выберем равномощное множество $\mathbf{S}$ таким образом, чтобы его элементы состояли из координат точек в пространстве $\mathbf{R}$, в таком случае получим $\mathbf{S = X}$.

Тогда мы сможем задать операцию параллельного переноса начала системы координат $\mathbf{G: X \times s_i \rightarrow X_i}$, вычитая из координат элементов множества **X** значение какого-либо из элементов $\mathbf{s_i}$ множества **S** (при этом вычитание будем производить «с загибом» – если координата вышла за допустимый нижний предел **min**, то её будем вычислять относительно допустимого верхнего предела **max** по формуле: $\mathbf{x_i = max - (min - (x - s_i)) + 1 = max - min + 1 + x - s_i}$). Посредством такого преобразования каждая точка множества **X** может быть отображена в начало координат в множестве $\mathbf{X_i}$ соответствующим выбором элемента $\mathbf{s_i}$, в силу того, что **S = X**. При этом все остальные точки будут сдвинуты параллельным переносом аналогичным образом так, что все углы и расстояния будут сохранены.

Если задана функция $\mathbf{H: X \rightarrow Y}$, то можно построить такую функцию $\mathbf{F: X \rightarrow Y}$, где $\mathbf{F = G \circ H}$, которая будет реализовывать параллельный перенос и потом уже осуществлять отображение элементов во множество **Y**, то есть $\mathbf{F: X \times s_i \rightarrow Y}$. Для графика отображения это будет означать его смещение как целого по горизонтали (вернее, по любым осям **X**) относительно начала системы координат в зависимости от параметра $\mathbf{s_i}$ – как это изображено на Рис. 5. А поскольку любая точка может быть сдвинута на произвольное расстояние, то такая функция будет заполнять всё координатное пространство в своей области определения плотно и действительно способна отобразить множество **X** в множество **Y** произвольным образом, то есть при подходящей области своего определения функция от аргументов и параметров способна представить любой алгоритм из **R**.

При этом выходной параметр $\mathbf{t \in T}$ определяет значение $\mathbf{s_i}$, которое будет взято для параллельного переноса на следующей итерации вычисления функции $\mathbf{F: X \times S \rightarrow Y \times T}$. Следовательно, тот способ, которым функция вычисляет значение этого параметра, оказывает существенное влияние на поведение графика отображения во времени. Так, если $\mathbf{S \setminus T \neq \varnothing}$, то это будет означать появление «чёрной дыры» как и в случае $\mathbf{X \setminus Y \neq \varnothing}$, когда аргумент может быть преобразован только в ограниченное подмножество результатов **Y**, соответственно – в ограниченное подмножество конфигураций из **R**. Также если функция **F** отображает любой элемент $\mathbf{s_i \in S}$ в элемент $\mathbf{t_i \in T}$, такой что $\mathbf{t_i = s_i}$, то такая функция будет описывать единственный алгоритм, неизменный при любых взаимодействиях.

Фактически выходной параметр $\mathbf{t \in T}$ динамически задаёт график отображения $\mathbf{F: X \rightarrow Y}$ во времени, то есть правило преобразования алгоритмов при взаимодействиях, и какой бы ни была исходная функция $\mathbf{F: X \rightarrow Y}$, параллельный перенос посредством параметров **S** и **T** способен как угодно изменить график функции $\mathbf{F: X \times S \rightarrow Y \times T}$ в пределах области её определения, сдвигая каждое следующее значение **X** по сравнению с предыдущим. Таким образом, само отображение **F**, входной и выходной параметры **S** и **T** задают некоторый алгоритм, конфигурацию вычислительного пространства **R**, осуществляющий отображение $\mathbf{F: X \rightarrow Y}$ и изменяемый в процессе такого отображения. **Теорема доказана.**

Следствие: Все существующие алгоритмы нейронных сетей неэффективны. Действительно, каждый исходящий синапс формального нейрона можно представить как функцию **F** суммированного входного параметра $\mathbf{x \in X}$ и внутреннего состояния синапса $\mathbf{s \in S}$, при этом такая функция должна быть непрерывной по теореме Колмогорова о представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного [[28], Error: Reference source not found]. Непрерывность функции S(t)

Теорема о квадрате: 1.Set.pdf

для бесконечных множеств M = M * M

Зададимся вопросом, любой ли функцией представим алгоритм интеллектуальной системы? Можно показать, что не любая функция способна описать интеллект, в частности никакая линейная функция не способна этого сделать. Докажем это утверждение.

Т. 9

Теорема: Алгоритм интеллектуальной системы (интеллект) не представим в виде линейной функции.

Доказательство: Действительно, необходимость эргодичности алгоритма интеллектуальной системы была установлена в ходе доказательства теоремы Т. 1, в противном случае такая система не сможет сама модифицировать свой алгоритм и не будет являться интеллектуальной по определению.

Легко показать, что линейная функция не удовлетворяет критерию эргодичности. Действительно, любая линейная функция задаётся уравнением: **y(x) = A * x + B**, где **A** и **B** – константы. Суперпозиция **y(y(x)) = A * (A * x + B) + B** при раскрытии скобок даёт **y(y(x)) =** $\mathbf{A^2}$ *** x + A * B + B = C * x + D**, то есть снова линейную функцию. Следовательно, суперпозиция линейных функций всегда будет зависеть от параметра **x**.

В частности, алгоритм интеллектуальной системы не может представлять из себя функцию **y(x) = Const**, ибо значение **Const** полностью определяется случайной начальной конфигурацией вычислительного пространства системы, таким образом функция однозначно зависит от аргумента **x**, когда под этим аргументом понимается сама функция **y**. Требование предсказуемости результата в этом как и в предыдущем случае не соблюдается, ибо множество референтных систем пусто. **Теорема доказана.**

Так какими же тогда функциями описывается интеллект? Как вообще представить себе функцию, которая должна вычислять некоторый результат, абсолютно независимый тем не менее ни от аргументов функции, ни даже от самой этой функции?! Противоречие можно преодолеть, сделав предположение о том, что не всякая случайная конфигурация способна породить интеллектуальную систему, а только такая, которая удовлетворяет некоторым начальным условиям, в противном случае интеллектуальная система просто не возникает. При этом класс таких функций должен быть достаточно широк, чтобы вероятность появления нужной конфигурации не стремилась к нулю. Выражаясь более строго – множество необходимых функций **Z** должно быть сопоставимо по мощности с множеством всех возможных функций **R**, то есть: $|\mathbf{Z}| / |\mathbf{R}| > \mathbf{0}$.

То, что такие функции в принципе существуют, мы доказали теоремой Т. 8. Осталось только определить вид этих функций. На самом деле у меня есть предположение о виде необходимых для интеллекта функций, поэтому я сразу дам нужное определение, а потом докажу, что такие функции нам подходят. Итак, сформулируем требования к такой функции.

Обозначим как **Z** множество конфигураций, при которых интеллектуальная способна возникнуть, и как **Y** – множество конфигураций, являющихся результатом. Очевидно, что **Y** и **Z** являются подмножествами множества **R** всех возможных конфигураций вычислительного пространства: $\mathbf{Y} \subset \mathbf{R}$ и $\mathbf{Z} \subset \mathbf{R}$, а пересечение множеств **Y** и **Z** – непусто: $\mathbf{Y} \cap \mathbf{Z} \neq \varnothing$. Если **Y** не является собственным подмножеством **Z**, то есть: $\mathbf{Y} \setminus \mathbf{Z} \neq \varnothing$, то это означает, что интеллектуальная система может исчезнуть в определённый момент своей эволюции. Тогда:

Хаотическая функция без аттрактора – это такая функция, определённая на всём множестве **Z**, суперпозиция которой плотно покрывает всё пространство **Z** при любом начальном значении её аргумента.

Хаотическая функция с аттрактором – это такая функция, определённая на всём множестве **Z**, суперпозиция которой плотно покрывает пространство **Z**, пока не достигнет результата – конфигурации, принадлежащей множеству **Y**. Множество **Y** замкнуто относительно данной функции, то есть любое применение этой функции к аргументу из множества **Y** даёт результат, принадлежащий множеству **Y**.

На самом деле хаотическая функция без аттрактора является частным случаем хаотической функции с аттрактором, у которой множества **Y** и **Z** совпадают. Поэтому обе разновидности можно объединить в один класс **хаотических функций**. Теперь покажем, что такие функции способны описать интеллект.

Т. 10

Теорема: Алгоритм интеллектуальной системы (интеллект) представим в виде хаотической функции.

Доказательство: Легко видеть, что какова бы ни была хаотическая функция, бесконечная суперпозиция данной функции на множестве **Z** всегда приведёт к результату **Y**, поскольку суперпозиция хаотической функции покрывает всё пространство **Z** плотно. Иными словами – какова бы ни была начальная конфигурация вычислительного пространства, принадлежащая множеству **Z**, в процессе своей эволюции она всегда приведёт к конфигурации, принадлежащей множеству **Y**, следовательно, хаотическая функция с аттрактором является «эргодической» в определённом нами в теореме Т. 1 смысле.

Далее, необходимо доказать, что класс хаотических функций **Z** сопоставим по мощности с множеством всех возможных функций **R**.

Заславский - От маятника до турбулентности и хаоса, с.177

Теорема доказана.

«Синергетические методы управления сложными системами» - «Размерность аттрактора -- цели исходной системы -- обычно существенно меньше размерности её фазового пространства. Отсюда вытекает идеология процессов обработки информации и управления в сложных нелинейных динамических системах: для этого необходимо, чтобы указанные процессы включали, по меньшей мере, две фазы: во-первых, фазу расширения и, во-вторых, фазу сжатия пространства состояний. Эти фазы реализуются с помощью соответствующей совокупности нелинейных положительных и отрицательных обратных связей. При этом в фазе расширения в системе формируется подмножество различных альтернатив поведения для её взаимодействия с внешней средой или другими системами. В фазе сжатия система сжимает область притяжения аттракторов, ранее построенных, в один из желаемых аттракторов -- цель системы.»

С точки зрения практической реализации полезно определить, как влияет размер аттрактора на интеллектуальность системы.

Важно найти функции, определённые на всём множестве Z или его большом подмножестве.

Однако, применение хаотической функции без аттрактора не подходит для представления интеллектуальной системы, ибо <…>.

Вывод операции суперпозиции для взаимодействия и состояния из частей: рассмотрим область A*B*C, тогда её состояние после взаимодействия можно записать либо как: A*B(C) * C(B*A), либо как A(B*C) * [C*B](A), очевидно, что в силу ассоциативности «*» - это суперпозиция. Порядок берётся относительно вычисляющего алгоритма.

Точечный аттрактор – уничтожение внутреннего сигнала.

Бесконечный аттрактор – странное «решение».

ru.wikipedia.org/wiki/Многообразие: Обычно в определениях дополнительно предполагается, что многообразие либо паракомпактно (это эквивалентно метризуемости), либо, что ещё сильнее, имеет счётную базу (это эквивалентно тому, что многообразие вкладывается в Евклидово пространство конечной размерности). – Someone: неверно

Док-во неметризуемости: lib.mexmat.ru/forum/viewtopic.php?t=5631 (пр-во Минковского есть T_0, то же следует из КМ – отсутствие изоморфизма и интегралы по путям).

Пространство это способ выражения неких фундаметальных свойств взаимодействия на строгом математическом языке. (Котофеич)

Если время - полноценная координата, почему бы той статической 4D Вселенной не поиметь такую конфигурацию, чтобы в 3D с законами получился полный бардак? Ну скомкали при рождении эту Вселенную неверно и получили, что в одном месте мировые линии расположены одним образом, в другом - совершенно иначе.

Если пространство-время - это некоторая физическая сущность, то из чего состит эта сущность?

Любое пространство - подпорка для ума, оно должно следовать из законов взаимодействия объектов. В математике вот именно так и делают.

Я не говорю, что 4D - это абстракция, а 3D - реальность. Нет, я хочу сказать, что и 4D, и 3D - оба абстракции. И противоречия здесь нет, поскольку возможность геометризации физики говорит о том, что геометрия = законы взаимодействия. Это равенство можно трактовать либо как геометрия -> законы взаимодействия, либо как законы взаимодействия -> геометрия.

Напротив, если мы встанем на позицию, что кроме материи и движения есть такая дополнительная физическая сущность как пространство, то фактически признаем существование эфира - если эта сущность из чего-то состоит, то можно принять её за абсолютную систему отсчёта и измерять абсолютную скорость относительно данной АСО.

Есть другой вариант - отказаться от взаимодействия и рассматривать только искривлённое пространство. Этот вариант тоже непротиворечив и к эфиру не приводит, но материя и движение мне как-то больше нравится. На самом деле это два математически совершенно эквивалентных описания. Но вот в статическом варианте вероятности КМ как-то совсем неважно выглядят - получается, что все результаты измерения были заложены заранее и изучать их смысла никакого нет. Тогда непонятно, почему эти результаты всегда соответствуют законам КМ - если вероятности заложены заранее, с тем же успехом они могли бы и нарушать эти законы...

Пригожин, Стенгерс: теорема о минимуме производства энтропии (на роль целевой функции).

Открытый проект

Заключение

Разум – это интеллект, осознающий сам себя.

Литература

1 Цетлин М.Л. «Исследования по теории автоматов и моделированию биологических систем», М: Наука, 1959

2 Винер Н. «Кибернетика, или управление и связь в животном и машине», М: Наука, 1983

3 Ежов А.А., Шумский С.А. «Нейрокомпьютинг и его применения в экономике и бизнесе», М: МИФИ, 1998

4 Николс Дж.Г., Мартин А.Р., Валлас Б.Дж., Фукс П.А. «От нейрона к мозгу», М: Едиториал УРСС, 2003

5 Горбань А.Н., Дунин-Барковский В.Л., Миркес Е.М. и др. «Нейроинформатика», Новосибирск: Наука. Сиб. предприятие РАН, 1998

6 Лурия А.Р. «Лекции по общей психологии», Питер, 2004

7 Олескин А.В., Ботвинко И.В., Цавкелова Е.А. «Колониальная организация и межклеточная коммуникация у микроорганизмов», Микробиология, 2000, Т.69, №3

8 Пригожин И., Стенгерс И. «Порядок из хаоса: новый диалог человека с природой», М: Прогресс, 1986

9 Анохин П.К. «Системный анализ интегративной деятельности нейрона», Успехи физиол. наук, 1974, Т.5, №2

10 Анохин П.К. «Принципы системной организации функций», М: Наука, 1975

11 Варшавский В.И., Поспелов Д.А. «Оркестр играет без дирижера», М: Наука, 1984

12 Девис П. «Суперсила», М: Мир, 1989

13 Фон Нейман Дж. «Теория самовоспроизводящихся автоматов», М: Мир, 1971

14 Тоффоли Т., Марголус Н. «Машины клеточных автоматов» , М: Мир, 1991

15 Каплан А.Я. «Нестационарность ЭЭГ: методологический и экспериментальный анализ», Успехи физиол. наук, 1998, Т.29, №3

16 Белоусов А.И., Ткачев С.Б. «Дискретная математика», МГТУ им. Баумана, 2003

17 Хаусдорф Ф. «Теория множеств», М: Едиториал УРСС, 2006

18 Верещагин Н.К., Шень А. «Вычислимые функции», М: МЦНМО, 2002

19 Ольшанский А.Ю. «Умножение симметрий и преобразований», СОЖ, 1996, №5

20 Сушкевич А.К. «Теория обобщенных групп», Харьков-Киев: ДНТВУ, 1937

21 Клини С.К. «Введение в метаматематику», М: ИЛ, 1957

22 Верещагин Н.К., Шень А. «Языки и исчисления», М: МЦНМО, 2002

23 Болтянский В.Г., Ефремович В.А. «Наглядная топология», М: Наука, 1983

24 Пескин М.Е., Шредер Д.В. «Введение в квантовую теорию поля», Москва-Ижевск: РХД, 2001

25 Дойч Д. «Структура реальности», Москва-Ижевск: РХД, 2001

26 Заславский Г.М., Сагдеев Р.З. «Введение в нелинейную физику: от маятника до турбулентности и хаоса», М: Наука, 1988

27 Анищенко В.С. «Динамические системы», СОЖ, 1997, №11

28 Колмогоров А.Н. «О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного», Докл. АН СССР, 1957, Т.114, №5