Вычислительная метамодель реальности и проблема истины[1]

А.М. Анисов

abstract. A computational interpretation of philosophical notions of reality and truth is provided along with a non-standard computational treatment of logical connectives and quantors. First-order language statements are compared to sets whereupon the computablilty of these statements comes to the problem of computability of the corresponding sets.

В ряде предшествующих работ проблема истины и реальности обсуждалась применительно к моделям, в которых реальность была представлена теоретико-множественным образом. Точнее, это не были обычные регулярные структуры, которые можно мыслить как построенные с помощью стандартных теоретико-множественных операций исходя из пустого множества. Объекты и предикаты реальности моделировались иррегулярными атомами, свойства которых не подлежали контролю со стороны модельного познающего субъекта ([3], [4], [5]). Несмотря на некоторую экзотичность таких моделей, они всецело оставались в рамках статического подхода к представлению реальности. Более того, применительно к ним опять возникал уже давно поставленный в философии теории множеств вопрос: в каком смысле множества могут обладать реальным существованием? И если еще можно смириться с реальным существованием множеств объектов, соответствующим сингулярным предикатам (допустим, согласиться с реальным существованием множеств людей, футбольных мячей и планет), то как быть с реальным существованием упорядоченных пар, троек, четверок и т.д., образующих основу предикатов-отношений?

В теории множеств упорядоченная пара $< a, b >$ объектов a и b стандартно определяется множеством множеств $< a, b > =$

[1]Работа поддержана РГНФ. Грант № 04-03-00344.

$\{\{a\},\{a,b\}\}$. Полагая $< a,b,c > = < a,< b,c >>$, $< a,b,c,d > = < a,< b,< c,d >>>$ и, вообще, $< a_1,a_2,\dots a_n > = < a_1,< a_2,\dots,< a_{n-1},a_n >> \dots >$, получаем быстро растущую конструкцию вложенных друг в друга множеств. Скажем, очередь из трех человек a,b,c должна быть представлена следующим множеством множеств:

$< a,b,c > = < a,< b,c >> = \{\{a\},\{a,\{\{b\},\{b,c\}\}\}\}$.

Затруднительно придать такой конструкции статус реальной, т.е. существующей независимо от субъекта, сущности, хотя очереди, вне сомнений, существуют как раз реально.

Поскольку речь идет не о недостатках какой-то конкретной модели или серии моделей, а об ограниченности возможностей теоретико-множественного моделирования в целом, уместнее в этом плане говорить о *теоретико-множественной метамодели.* Естественно, подразумевается, что данная метамодель не единственна. Альтернативой теоретико-множественной метамодели могла бы стать *вычислительная метамодель.* В первом приближении идея вычислимости предпочтительнее теоретико-множественного подхода как в отношении возможности преодоления статики, так и в смысле возможности построения более реалистичной онтологии, не предполагающей реального существования иерархии множеств.

В условиях широчайшего распространения вычислительного моделирования, особенно с применением компьютеров, призыв к построению вычислительной метамодели может показаться неуместным: не ломится ли автор в давно открытую дверь? Однако напомним, в какой связи возникает идея этой метамодели. *Теория вычислимости до сих пор не стала альтернативой теории множеств как фундаментального основания философии и науки.* Последние по сей день либо пребывают в плену статики, либо впадают в иррационализм, когда обнаруживают неустранимую нестабильность универсума. Автор лично неоднократно сталкивался с искренним недоумением известных и не очень известных философов и физиков, когда предлагал вычислительную модель времени — они-то знают, что «в действительности» все существует «в едином пространстве-времени», все «мировые линии» которого раз и навсегда прописаны. Мною был даже предложен термин *парменидовская наука* для описания практически полного торжества в ориентированной на науку философии и в точном естествознании восходящей к элеатам концепции неподвижного универсума, само время которого полностью статично и лишено даже намека на становление ([2]).

Сама по себе идея рассмотрения природы как особого рода вычислительного процесса или совокупности вычислительных процессов не нова. Уже давно мы слышим призывы считать законы природы алгоритмами, высказывания о том, что роль теорий с успехом способны взять на себя компьютерные программы и т.п. Однако, как правило, эти высказывания — либо не более чем декларации ([10]), либо за ними скрываются пока еще мало продуктивные попытки ввести вычислительный язык в физику ([6]). Редким удачным исключением является, например, книга известного ученого и инженера Х. Хармута ([9]), но в ней обсуждается проблема перехода от континуальных пространств к дискретным, а не идея физической вычислимости как таковая. И все же здесь затрагиваются некоторые философские вопросы, в частности, проблемы дискретного времени.

В философии вычислительные концепции фактически находятся в стадии начальной разработки. В нашей стране, кроме упомянутой вычислительной модели времени ([1]), можно назвать предложенное В.И. Шалаком альтернативное классическому понятие вычислительного следования[2] и попытку А.А. Крушинского трактовать рассуждения древних китайцев как вычисления ([8] и другие работы). Примечательно, что оба специалиста скептически относятся к понятию истины, усматривая в нем выражение унаследованной от Платона и Аристотеля статичности и созерцательности в отношении к миру, несовместимых с ориентированными на умения вычислительными действиями. Разделяя позицию коллег в плане отрицания гордой *Истины*, тем не менее хотелось бы высказаться в защиту куда более скромной *истины*. Я полагаю, что *изменчивость универсума не является препятствием для принятия некоторых высказываний в качестве истинных.* Аргументации в пользу этого тезиса и будет посвящена данная работа.

Пусть U непустое множество, $V \subset U$ (включая случай $V = U$) и $f_V(x_1, x_2, \dots, x_n)$ — тотальная или частичная функция из мно-

[2] См. статьи В.И. Шалака в данном сборнике.

жества V^n в V, C_V — компьютер[3], обрабатывающий данные из V, и $\pi_{C_V}(x_1, x_2, \dots, x_n)$ — программа, выполняемая компьютером C_V. Если программа $\pi_{C_V}(x_1, x_2, \dots, x_n)$ нормально останавливается на входе $(x_1, x_2, \dots, x_n)$, пишем $\pi_{C_V}(x_1, x_2, \dots, x_n) \downarrow$. В противном случае (учитывая как возможность неограниченно продолжающихся вычислений, так и возможность аварийного останова или *авоста*) пишем $\pi_{C_V}(x_1, x_2, \dots, x_n) \uparrow$. Условимся, что только в случае $\pi_{C_V}(x_1, x_2, \dots, x_n) \downarrow$ существует, и при том единственное, $y \in V$, которое называется *результатом выполнения* программы π_{C_V} *на входе* $(x_1, x_2, \dots, x_n)$. Тогда применяем запись $\pi_{C_V}(x_1, x_2, \dots, x_n) \downarrow y$. Если же $\pi_{C_V}(x_1, x_2, \dots, x_n) \uparrow$, считаем, что результат выполнения программы π_{C_V} на входе $(x_1, x_2, \dots, x_n)$ *отсутствует*.

Введем достаточно близкое к стандартному уточнение понятия вычислимости применительно к функциям. Функция $f_V(x_1, x_2, \dots, x_n)$ *вычислима* на компьютере C_V, если существует программа π_{C_V} такая, что $\forall x_1 \forall x_2 \dots \forall x_n \forall y (f_V(x_1, x_2, \dots, x_n) = y \iff \pi_{C_V}(x_1, x_2, \dots, x_n) \downarrow y)$. (В дальнейшем, если конкретные возможности компьютера C_V не существенны, упоминание о нем можем опускать.)

Идея представления универсума в виде совокупности вычислительных процессов порождает проблему, которую нелегко сформулировать точным образом. Проблема в следующем. В каком смысле эти процессы, возможно, между собой никак не связанные, относятся к одному и тому же универсуму, формируют его, делают существующим? Теоретико-множественный универсум можно представлять как конструирующийся при помощи соответствующих операций из постулированно существующих пустого множества и бесконечного множества. Тем самым эти операции придают ему требуемое единство. Какие же операции придают связность совокупности вычислительных процессов? Взаимодействия процессов, их взаимное влияние? Но какая связь может быть, например, между процессами формирования звезды, рождения человека и становления государствен-

[3]Вычислительное устройство естественного или искусственного происхождения, реальное или идеальное, физическое или ментальное — на данном этапе это не столь существенно; важно лишь помнить, что все эти возможности допустимы, так что идея компьютера берется в предельно общем виде

ности? Хотя все это процессы порождения чего-то, они относятся к столь разным областям реальности, что никакая конкретная связь между ними невозможна. Остается искать связь абстрактную, не физическую, а логическую. Но и она интуитивно не очень-то просматривается. Тем не менее, мы рискнем предложить аргумент в пользу существования такой логической связи. Утверждая, что выполняются процессы π_1 и π_2, мы фактически как-то соотносим их между собой, как-то объединяем. Не допустить ли тогда существование некоторого абстрактного объединяющего процесса? Одним из возможных способов формально выразить данный аргумент является принятие следующего *постулата локального объединения.*

$(\forall V \subset U)(\forall W \subset U)((f_V$ *вычислима* $\&$ f_W *вычислима*$) \Longrightarrow f_{V \cup W}$ *вычислима*$)$.

Можно было бы обобщить постулат локального объединения. Пусть $\{V_{i \in I}\}$ — произвольное семейство множеств такое, что $I \subset U$ и $(\forall i \in I)(V_i \subset U)$. Тогда *постулат тотального объединения* утверждает, что $(\forall i \in I)$ f_{V_i} *вычислима* $\Longrightarrow f_{\cup\{V_{i\in I}\}}$ *вычислима.* Очевидно, что из второго постулата следует первый (достаточно взять двухэлементное $I \subset U$, если $|U| > 1$).

Дополнительными, хотя и косвенными, аргументами в защиту постулатов объединения могут послужить следующие факты. При вычислениях на заводских компьютерах, объединенных в сеть, достаточно реализовать возможность посылки терминального сигнала всем компьютерам любым из них, который первым закончит вычисление (пример реализации постулата локального объединения). В наиболее общей теории пространств — топологии — топологическое пространство τ может быть представлено семейством множеств, для которого принимается аксиома о том, что объединение любого семейства множеств из τ принадлежит τ (аналог постулата тотального объединения). Тем не менее в данной работе принимается лишь постулат локального объединения, поскольку только он понадобится в дальнейшем.

Те же соображения наводят на мысль о постулировании обязательного существования пересекающихся процессов. Если функции f_V и f_W вычислимы и при этом $V \cap W \neq \varnothing$, то интуитивно функция $f_{V \cap W}$ тоже должна быть вычислимой. В самом деле, если исполняются вычисляющие эти функции программы π_{C_V}

и π_{C_W}, то при $< x_1, x_2, \ldots, x_n >\in V \cap W$ будем в конце концов иметь $\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow$ и $\pi_{C_W}(x_1, x_2, \ldots, x_n) \downarrow$. Но это по сути и означает вычислимость функции $f_{V \cap W}$. Остается формально обеспечить наличие соответствующих объектов, приняв следующий *постулат локального пересечения.*

$(\forall V \subset U)(\forall W \subset U)((f_V$ *вычислима* & f_W *вычислима* & $V \cap W \neq \varnothing) \Longrightarrow f_{V \cap W}$ *вычислима*).

Требование непустоты пересечения нетрудно обосновать. Поскольку никакая конкретная модель вычислимости в обсуждаемой метатеории задана не была, возможно сужение области вычислимых функций на некоторую подобласть непустых тотальных (т.е. всюду определенных) функций $f^n : U^n \longmapsto U$. Тогда при $V \cap W = \varnothing$ функция $f_{V \cap W}$ не попадет в область вычислимым функций.

Разумеется, нельзя в общем случае требовать, чтобы для каждого непустого подмножества M множества V (соответственно W) функция f_M была бы вычислима, если функции f_V и f_W вычислимы. Например, классическая модель вычислимости на натуральных числах дает счетное множество вычислимых функций, тогда как число непустых подмножеств натурального ряда несчетно.

Также можно сформулировать обобщение только что принятой аксиомы до *постулата тотального пересечения.* Пусть $\{V_{i \in I}\}$ — произвольное семейство множеств такое, что $I \subset U$, $(\forall i \in I)(V_i \subset U)$ и $\cap\{V_{i \in I}\} \neq \varnothing$. Тогда $(\forall i \in I)$ f_{V_i} *вычислима* $\Longrightarrow f_{\cap\{V_{i \in I}\}}$ *вычислима.* Опять очевидно, что из постулата тотального пересечения вытекает постулат локального пересечения.

И вновь наличествует топологическая аналогия. Топологическое пространство τ представимо семейством множеств, для которого принимается аксиома о том, что пересечение любого семейства множеств из τ принадлежит τ. Тем не менее и здесь мы ограничимся принятием лишь локальной формы постулата пересечения.

Обратим внимание, что функция $f_V(x_1, x_2, \ldots, x_n)$ может быть частичной в двояком смысле. Во-первых, при $< x_1, x_2, \ldots, x_n > \in V^n$ значение функции $f_V(x_1, x_2, \ldots, x_n)$ может быть не определено. Во-вторых, при $V \neq U$ аргумент функции $f_V(x_1, x_2, \ldots, x_n)$

может быть неопределен. Это случится, если $< x_1, x_2, \dots, x_n > \notin V^n$. В этой ситуации значение функции $f_V(x_1, x_2, \dots, x_n)$ также будет неопределенным. Эти два случая позволяют трактовать запись $\pi_{C_V}(x_1, x_2, \dots, x_n) \uparrow$ при вычислениии функции $f_V(x_1, x_2, \dots, x_n)$ просто как отсутствие результата вычисления, оставляя открытым вопрос, явилось ли отсутствие результата следствием неограниченности вычисления во времени или тем, что оно закончилось аварийно.

Пусть L — какой-нибудь язык предикатов первого порядка, не содержащий функциональных символов[4], так что любой терм t — это либо индивидная переменная, либо индивидная константа. Пусть, далее, $P(x)$ сингулярный предикатный символ из L. В стандартной семантике А.Тарского функция интерпретации I сопоставит этому символу подмножество универсума: $I(P(x)) = V$, $V \subset U$. В обычной теории вычислимости, базирующейся на понятии машины с неограниченными регистрами (МНР)[5], которая манипулирует натуральными числами, вычислимость сингулярного предиката V определяется следующим образом. Предикат V *вычислим*, если существует МНР-программа π такая, что $\pi(x) \downarrow 1$ при $x \in V$ и $\pi(x) \downarrow 0$ при $x \notin V$. Но в наши планы отнюдь не входит намерение объявить числа частью реальности. Более того, нет вообще никаких оснований искать в реальности особые объекты «истина» и «ложь», даже если не отождествлять истину с 1, а ложь с 0. Истина и ложь входят в мир вместе с субъектом. Как тогда они могут иметь отношение к реальности?

Поэтапно отвечая на данный вопрос, вначале сопоставим предикату V функцию $f_V(x)$ такую, что $f_V(x) = y$ в случае $x \in V$ и $f_V(x)$ не определена в противном случае. Предикат V *вычислим*, если вычислима функция $f_V(x)$. Иными словами, вычислимость сингулярного предиката V означает, что существует программа $\pi_{C_V}(x)$, для которой верно $\pi_{C_V}(x) \downarrow y$, если $x \in V$, и $\pi_{C_V}(x) \uparrow$, если $x \notin V$. Предикатный символ $P(x)$ имеет *вычислительную*

---

[4]Это ограничение не приводит к потере общности, поскольку любой функциональный символ $f(x_1, x_2, \dots, x_n)$ может быть заменен предикатным символом $F(x_1, x_2, \dots, x_n, x_{n+1})$.

[5]См. [7]. В частности, в гл. 3 показано, что МНР-вычислимость эквивалентна вычислимости по Тьюрингу, а также другим известным классическим теориям вычислимости.

интерпретацию в смысле I, если предикат $I(P(x))$ вычислим.

Предложенная конструкция без проблем обобщается на случай многоместных предикатов. Пусть $I(P(x_1, x_2, \ldots, x_n)) = V^n$, $V^n \subset U^n$. Сопоставим предикату V^n функцию $f_V(x_1, x_2, \ldots, x_n)$ такую, что $f_V(x_1, x_2, \ldots, x_n) = y$ в случае $< x_1, x_2, \ldots, x_n > \in V^n$ и $f_V(x_1, x_2, \ldots, x_n)$ не определена в противном случае. Предикат V^n *вычислим*, если вычислима функция $f_V(x_1, x_2, \ldots, x_n)$, т.е. существует программа $\pi_{C_V}(x_1, x_2, \ldots, x_n)$, для которой верно $\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow y$, если $< x_1, x_2, \ldots, x_n > \in V^n$, и $\pi_{C_V}(x_1, x_2, \ldots, x_n) \uparrow$, если $< x_1, x_2, \ldots, x_n > \notin V^n$. Предикатный символ $P(x_1, x_2, \ldots, x_n)$ имеет *вычислительную интерпретацию в смысле* I, если предикат $I(P(x_1, x_2, \ldots, x_n))$ вычислим.

Если $\forall x_1 \forall x_2 \ldots \forall x_n \exists y \exists z (f_V(x_1, x_2, \ldots, x_n) = y \iff g_V(x_1, x_2, \ldots, x_n) = z)$, то пишем $f_V(x_1, x_2, \ldots, x_n) \equiv g_V(x_1, x_2, \ldots, x_n)$ или, короче, $f_V \equiv g_V$. Аналогичным образом для программ: если $\forall x_1 \forall x_2 \ldots \forall x_n \exists y \exists z (\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow y \iff \tau_{C_V}(x_1, x_2, \ldots, x_n) \downarrow z)$, то по определению $\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow y \equiv \tau_{C_V}(x_1, x_2, \ldots, x_n) \downarrow z$ или $\pi_{C_V} \equiv \tau_{C_V}$.

ТЕОРЕМА 1. *Отношение $\equiv$ является отношением эквивалентности соответственно на функциях и на программах. Кроме того, если функция f_V и соответствующая ей программа π_{C_V} вычисляют предикат V^n и при этом $f_V \equiv g_V$ и $\pi_{C_V} \equiv \tau_{C_V}$, то g_V и τ_{C_V} тоже вычисляют V^n.*

Доказательство. Доказательство очевидно. Q.E.D.

Из данной теоремы вытекает, что формально все равно, какую из эквивалентных функций или программ взять. В любом случае важно лишь то, имеет место $\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow$ или $\pi_{C_V}(x_1, x_2, \ldots, x_n) \uparrow$. Если реализовался первый вариант, то конкретный результат вычисления программы π_{C_V} на входе $(x_1, x_2, \ldots, x_n)$ не существен. Однако с философской точки зрения в случае реализации второго варианта при выборе эквивалентных программ необходимо учитывать, что невозможность получить результат вычисления может быть обусловлена двумя типами причин. Одно дело неограниченность процесса вычислений, другое — авост. Вряд ли в природе при невозможности получить результат какого-то конкретного процесса он будет продолжаться неограниченно долго. Более реалистично предположить, что такой процесс все-таки завершится, но аварийно. Например, природное вычисление предиката *Ключ x подошел к Замку y* может успешно завершиться за конечное число шагов, но может и не иметь благополучного завершения. В последнем случае вовсе не обязательно до бесконечности предпринимать все новые и новые безуспешные попытки открыть замок. В реальности такой процесс завершится авостом, чем бы он ни был вызван — поломкой ключа или замка, потерей терпения или чем-то иным. Хотя исключать априори реальное существование конкретных бесконечных процессов мы бы не решились, не говоря уже о возможных бесконечных процессах, связанных с универсумом в целом.

На данном этапе исследования установлена цепочка переходов *предикатный символ* $P(x_1, x_2, \ldots, x_n) \longmapsto$ *предикат* $I(P(x_1, x_2, \ldots, x_n)) = V^n \longmapsto$ *функция* $f_V(x_1, x_2, \ldots, x_n) \rightsquigarrow$ *программа* $\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow$. Последний переход $\rightsquigarrow$ осуществим не всегда — ведь функция $f_V(x_1, x_2, \ldots, x_n)$ не обязательно будет вычислимой на компьютере C_V. Но если она вычислима, понятия вычислимой выполнимости и вычислимой истины определяются почти очевидным образом. Пусть v — функция приписывания значений свободным переменным языка L: $v : Var \mapsto V$. Выражение $P(x_1, x_2, \ldots, x_n)$ *вычислимо выполнено* на компьютере C_V при интерпретации I и приписывании v, если $\pi_{C_V}(v(x_1), v(x_2), \ldots, v(x_n)) \downarrow$. Пусть $c_1, c_2, \ldots, c_n$ — индивидные константы из L. Высказывание $P(c_1, c_2, \ldots, c_n)$ *вычислимо истинно* на компьютере C_V при интерпретации I, если $\pi_{C_V}(I(c_1), I(c_2), \ldots, I(c_n)) \downarrow$.

В стандартной семантике не выполнимость $P(x_1, x_2, \ldots, x_n)$ и не истинность (ложность) $P(c_1, c_2, \ldots, c_n)$ определялись условиями $< v(x_1), v(x_2), \ldots, v(x_n) > \notin V^n$ и $< I(c_1), I(c_2), \ldots, I(c_n) > \notin V^n$ соответственно. Но механически переносить эту идею на вычислимость нельзя. Ведь замена $\pi_{C_V}(v(x_1), v(x_2), \ldots, v(x_n)) \downarrow$ на $\pi_{C_V}(v(x_1), v(x_2), \ldots, v(x_n)) \uparrow$, а $\pi_{C_V}(I(c_1), I(c_2), \ldots, I(c_n)) \downarrow$ на $\pi_{C_V}(I(c_1), I(c_2), \ldots, I(c_n)) \uparrow$ приведет к многочисленным нелепостям практического и теоретического характера.

Например, попробуем вычислять функцию деления $x/y = z$, т.е. трехместный предикат $/(x, y, z)$. Что случится, если $y = 0$?

Калькулятор Windows в ответ на введение $x = 1$ и $y = 0$ при применении операции / сообщил: «Деление на нуль запрещено». Можно ли сделать вывод, что это же сообщение последует при любых допустимых значениях переменной x, если $y = 0$? Однако при $x = 0$, $y = 0$ последовало: «Значение не определено». Чтобы прояснить вопрос, в системе Free Pascal была откомпилирована (компилятор PPC386) следующая простая программа.

```
var x, y, z : real;
begin
  Read(x, y);
   z := x / y;
  Write(z);
end.
```

В ответ на ввод 1 и 0 ничего не произошло! Программа либо зависла, либо продолжала выполняться как ни в чем не бывало, не выдавая никаких сообщений и явно не собираясь когда-либо заканчивать вычисления. Зато ввод значений 0 и 0 привел к распечатке результата: —0.0000000000000000E+000.

Или тот же пример с ключом и замком. Разве из того факта, что ключ x не открыл замок y, однозначно вытекает отрицание предиката *Ключ x подошел к Замку y* и выполнимость $\neg$(*Ключ x подошел к Замку y*)? А если замок тугой и не поддается, но ключ все-таки от него? А если ключ крутили не в ту сторону? Могли быть и многие другие причины неудачи. Так что и здесь отсутствие результата не означает, что выполнено отрицание искомого предиката.

Что касается теоретических аргументов, то главное теоретическое возражение против трактовки отрицания как отсутствия результата вычисления состоит в том, что факт получения результата в принципе может быть зафиксирован, тогда как продолжающееся вычисление, вообще говоря, оставляет открытым вопрос, закончится оно нормально или нет. Те же компьютерные вычисления могут продолжаться неприемлемо долго, но их искусственное прерывание не означает фиксации наступления события $\pi_{C_V}(x_1, x_2, \ldots, x_n) \uparrow$. В теории классической вычислимости, как хорошо известно, проблема останова не разрешима. В логике доказано существование алгоритма перебора всех теорем исчисления предикатов, но если некоторая формула A не появилась на шаге n выполнения такого алгоритма, то в общем не исключено, что она появится потом на каком-то шаге m. Не наступивший до сих пор конец света не ведет к однозначному выводу, что мир будет существовать вечно. И так далее. Так что корректное введение отрицания на такой неопределенной по самой своей сути основе, как незавершенность вычисления, невозможно.

Теоретически, выходом из создавшегося затруднительного положения было бы такое определение отрицания, которое базировалось бы на нормально завершающихся вычислениях, т.е. на конструкциях типа $\pi_{C_V}(x_1, x_2, \ldots, x_n) \downarrow$. Алгебра логики подсказывает решение, связанное с операцией *дополнения* $\setminus$ множеств на выбранном заранее универсуме. Поскольку универсум U фиксирован, вычислимость $\neg P(x)$ можно свести к проблеме вычислимости предиката $U \setminus V$, где $V = I(P(x))$. Или, в общем виде, вычислимость $P(x_1, x_2, \ldots, x_n)$ свести к вычислимости предиката $U^n \setminus V^n$, где $V^n = I(P(x_1, x_2, \ldots, x_n))$. Однако есть определенные сомнения на счет философской приемлемости такого решения.

Во-первых, нередко получается так, что универсумы рассуждений весьма обширны, тогда как исходные предикаты имеют сравнительно небольшой объем. Тогда подвергнутый отрицанию такой исходный предикат внезапно увеличивается до непомерных размеров. Проанализируем в качестве примера следующий ряд утверждений: «*Все люди смертны. Все греки люди. Сократ грек. Следовательно, Сократ смертен. Но Наполеон не грек. . .*». По контексту в качестве универсума здесь надо взять, как мнимум, множество смертных существ. Но тогда высказывание $\neg$*Грек(Наполеон)* должно быть проинтерпретировано на всем огромном множестве смертных, за исключением греков. Обычно считается, что подобные вещи не создают особых логических проблем, однако проблемы с вычислимостью появляются. Представьте себе вычислительный процесс, который перебирает всех смертных не греков, включая бактерии, растения, грибы и животных, чтобы устновить, что Наполеон не грек.

Во-вторых, для практики и эмпирической науки характерно использование многослойных или многосортных универсумов. Простой подсчет баранов в стаде уже сочетает такие сорта, как бараны и числа. Многосортная логика предикатов первого порядка, как известно, сводима к обычной односортной, так что и здесь проблем обычно не видят. Семантически сводимость обеспечивается объединением слоев в единый универсум с введением соотвествующих слоям предикатов. Однако операция отрицания вновь выглядит довольно нелепо. К примеру, выделив в стаде белых баранов, получим с использованием отрицания высказывательную форму *Это не белый*, которая может быть представлена в классическом исчислении предикатов формулой $\neg$*Белый*(x), интерпретация которой потребует смешения баранов и чисел, ибо не былыми окажутся не только некоторые бараны, но и все числа. Вычисление так понимаемого отрицания еще более усложнит проблему.

Отчасти аналогичными затруднениями вызвана трактовка отрицания как *неудачи* в логическом программировании и в языке декларативного программирования Пролог. Проблеме отрицания в логическом программировании и в Прологе посвящена обширная литература[6], которую мы разбирать не будем. В самом общем виде отрицание $\neg P$ здесь понимается как неудача в выводе P: если вывести P не удалось, считается выведенным $\neg P$. В терминах наших примеров, чтобы вычислить истинностное значение высказывания $\neg$*Грек(Наполеон)*, надо проверять высказывание *Грек(Наполеон)*, перебирая множество греков Г (все-таки не весь универсум!). Если перебор завершится неудачей, разрешено сделать вывод $\neg$*Грек(Наполеон)*. Для нас проблема в том, что перебирающий множество $V = I(P(x))$ вычислительный процесс может «не видеть» объекты из дополнения $U \backslash V$. Скажем, если есть механизм развития болезни $\pi_{C_\Gamma}(x) \downarrow$, поражающий греков, и только их, то в отношении Наполеона такой процесс не установит ничего определенного — он попросту не будет выполнен: $\pi_{C_\Gamma}(\textit{Наполеон}) \uparrow$. Таким образом, трактовка отрицания как неудачи вновь возвращает нас к уже отвергнутому варианту применения конструкций вида $\pi_{C_V}(x_1, x_2, \ldots, x_n) \uparrow$ в связи с отрицанием.

[6]Отметим книги: *Хоггер К.* Введение в логическое программирование. М., 1988; *Ковальски Р.* Логика в решении проблем. М., 1990; *Клоксин У., Меллиш К.* Программирование на языке Пролог. М., 1987; *Стерлинг Л., Шапиро Э.* Искусство программирования на языке Пролог. М., 1990.

Напрашивается следующий, как нам представляется, естественный путь избавления от возникших проблем с отрицанием. С исходным дескриптивным предикатом P надо соотнести *релевантную* для него часть R универсума U. В некотором отношении такая часть сама ведет себя подобно универсуму, так что R можно назвать *релевантным универсумом* для P. Поясним основную идею на примере рассуждения: «*Насекомое — это животное. Следовательно, крупное насекомое — это крупное животное*». По биологическим классификациям насекомые действительно являются животными, так что посылка истинна. Но как быть с заключением? Один из квалифицированных логиков, к которому обратился автор, согласился с этим рассуждением. Синтаксис языка в самом деле принуждает к тому, чтобы считать два вхождения слова «крупное» вхождением одного и того же предиката. Однако смысл первого вхождения данного слова отличается от его второго вхождения. Поэтому надо было бы написать «*крупное*$_1$ *насекомое — это крупное*$_2$ *животное*». Затем предикат $\textit{Крупное}_1(x)$ определить на множестве насекомых R_1, получив $I(\textit{Крупное}_1(x)) \subset R_1$, а $\textit{Крупное}_2(x)$ определить на множестве животных R_2, получив $I(\textit{Крупное}_2(x)) \subset R_2$ и не возражая против $R_2 = U$. Множества R_1 и R_2 будут соответствующими исходным дескриптивным предикатам $\textit{Крупное}_1(x)$ и $\textit{Крупное}_2(x)$ релевантными универсумами. В этом случае следования, конечно, нет, в должном соответствии со смыслом рассуждения.

Предлагаемый подход к отрицанию состоит в ограничении операции дополнения соответствующим релевантным универсумом. Отрицанию $\neg\textit{Крупное}_1(x)$ в модели сопоставляется множество $R_1 \backslash I(\textit{Крупное}_1(x))$, а отрицанию $\neg\textit{Крупное}_2(x)$ — множество $R_2 \backslash I(\textit{Крупное}_2(x))$. Такой подход не только позволяет избавиться от призрака отрицания как неудачи, но и решает как проблему соразмерности утверждаемого и отрицаемого, так и проблему предотвращения смешения сортов при отрицании в многосортном универсуме.

Теперь определения вычислимой выполнимости и вычислимой истины на отрицаниях атомарных предикатов получаются из соответствующих определений для атомарных предикатов. Достаточно при построении семантических значений отрицаний заменить предикат V^n на предикат $R^n \backslash V^n$, только и всего. При этом случай $R^n = U^n$, вообще говоря, не исключается. Однако следует явным образом исключить возможность пустоты релевантных универсумов. Ведь каждый релевантный универсум с философской точки зрения есть именно универсум, а универсумы должны быть непусты (иначе о чем рассуждать?). Кроме того, как нельзя в одном рассуждении по-разному интерпретировать предикатный символ P, так и нельзя по тем же причинам в одном рассуждении приписывать предикатному символу P разные релевантные универсумы. Отсюда необходимость принятия следующего неформального постулата.

Для всякого атомарного предикатного символа P^n *и релевантного для* P^n *универсума* R^n *имеет место непустота и единственность* R^n.

Но мы не решаемся в общем случае требовать, чтобы для всякого атомарного предиката P^n *существовал* релевантный универсум R^n. Хотя бы из-за вышеописанных трудностей с отрицанием некоторых разновидностей пустых предикатов. Но применительно к конкретным предметным областям и наборам исходных дескриптивных предикатных символов такое существование возможно.

Свойства получившегося вычислительного отрицания на атомарных предикатах в существенных отношениях не совпадают с классическим логическим отрицанием. В дальнейшем при обсуждении свойств логических операций будем использовать сингулярные предикаты, если их местность не важна. Зафиксировав функцию интерпретации I, вместо длинного «предикатный символ $P(x)$ имеет вычислительную интерпретацию в смысле I» будем писать «$P(x)$ вычислимо». Будем использовать символ $\vdash$ для принятия метаутверждения и символ $\dashv$ для отбрасывания метаутверждения.

ТЕОРЕМА 2. $\vdash$ *(a). $P(x)$ и $\neg P(x)$ не могут быть вместе вычислимо выполнены;*

$\vdash$ *(b). $P(c)$ и $\neg P(c)$ не могут быть вместе вычислимо истинны;*

$\vdash$ *(c). $P(t)$ вычислимо (выполнено, истинно)* $\Longleftrightarrow$ $\neg\neg P(t)$ *вычислимо (выполнено, истинно);*

$\dashv$ *(d). $P(x)$ вычислимо* $\Longrightarrow$ $\neg P(x)$ *вычислимо;*

$\dashv$ *(e). $\neg P(x)$ вычислимо $\Longrightarrow$ $P(x)$ вычислимо;*

Доказательство. Ограничимся пунктами (b)–(d).

(b). Пусть $P(c)$ вычислимо истинно. Тогда $\pi_{C_V}(I(c))\downarrow$. Отсюда $I(c) \in V$. Если бы $\neg P(c)$ было вычислимо истинно, имели бы $\pi_{C_{R\backslash V}}(I(c))\downarrow$ и, следовательно, $I(c) \in R\backslash V$, что противоречиво. Пусть $\neg P(c)$ вычислимо истинно. Имеем $\pi_{C_{R\backslash V}}(I(c))\downarrow$ и $I(c) \in R\backslash V$, что исключает вычислимую истинность $P(c)$.

(c). Если $I(P(t)) = V$, то $I(\neg P(t)) = R\backslash V$. Положим $W = R\backslash V$. Тогда $I(\neg\neg P(t)) = R\backslash W = R\backslash(R\backslash V) = V$.

(d). Возьмем в качестве компьютера C МНР-машину. Тогда на множестве формул классического исчисления предикатов предикат *Теорема*(x) вычислим, но предикат $\neg$*Теорема*(x) не вычислим. Q.E.D.

Обратим внимание на то, что в построенном фундаменте вычислительной онтологии и семантики понятие *лжи* до сих пор не появилось, хотя онтология и семантика отрицания были определены. Речь шла о вычислимой выполнимости и вычислимой истине независимо от того, применялись эти понятия к предикатному символу $P(t)$ или к $\neg P(t)$. Невольно возникает мысль, что *без понятия лжи вообще можно обойтись*! Такая идея хорошо согласуется с нововременным философским принципом классической науки о том, что природа не может лгать. В логическом плане идею истины как единственного денотата определенных языковых выражений развивает С.А. Павлов[7]. Суть в том, что некоторые выражения обозначают истину, тогда как остальные не обозначают истину. Но отсюда, как показывает С.А. Павлов, вовсе не обязательно делать вывод, что эти остальные обозначают ложь. Другой вопрос, что в его логической метатеории формально ложь можно сделать единственным денотатом. Наш подход в этом пункте принципиально отличается. Не видно, как в рамках предлагаемой метатеории вычислимости формально можно было бы вычислительную ложь сделать исходной по отношению к вычислительной истине, тем более сделать ложь единственным денотатом, полностью устранив истину.

[7]См. статью С.А. Павлова в данном сборнике.

Не только производность, вторичность лжи по отношению к истине, но и некоторую искусственность и избыточность понятия лжи мы сейчас, учитывая уже сказанное, попытаемся продемонстрировать. Что может означать утверждение о *ложности* высказывания $P(c)$ при интерпретации I в вычислительном смысле? Единственно возможный ответ — никакой вычислительный процесс никогда нормально не остановится на входе $I(c)$: $\neg\exists\pi(\pi(I(P(c))\downarrow)$. Вытекает ли отсюда, что существуют множество R и процесс π такие, что $\pi(R\backslash I(P(c))\downarrow$? Рассуждая абстрактно, ни откуда не следует, что такое множество и такой процесс существуют. Тогда каков реальный смысл утверждения о ложности $P(c)$? Ведь получается, что с ним ни прямо, ни косвенно нельзя связать никакого процесса!

Рассмотрим следующий пример утверждения: *Ложно, что Жанна д'Арк — ведьма*, или, в логической форме, *Ведьма*(*Жанна д'Арк*) *ложно*. Чтобы напрямую в этом убедиться, необходимо запустить *все возможные* процессы (что само по себе абсурдно). И сидеть ждать? Как долго? А если некоторые процессы никогда не завершатся? Ясно, что прямой путь ведет в никуда. Остается косвенное решение: обосновать истинность $\neg$*Ведьма*(*Жанна д'Арк*), подобрав релевантное R, а затем найдя подходящий процесс π, для которых имело бы место $\pi(R\backslash I($*Ведьма*(*Жанна д'Арк*)$)\downarrow$. Но что взять за R? Ложность рассматриваемого высказывания предполагает, что $I($*Ведьма*(*Жанна д'Арк*)$) = \varnothing$. Если положить $R = U$, то $I(\neg$*Ведьма*(*Жанна д'Арк*)$) = U\backslash\varnothing = U$, и *любой* нормально завершившийся процесс (например, передачи данных, деления клетки, слияния фирм, борьбы с врагом и т.д., до бесконечности) делает это высказывание истинным, что бессмысленно. Ясно, что R должно быть существенно сужено. Как сужено? Раз ведьмы могут встречаться только среди женщин, не взять ли $R = \{$x: Женщина(x)$\}$? Тогда $I(\neg$*Ведьма*(*Жанна д'Арк*)$) = R\backslash\varnothing = R = \{$x: Женщина(x)$\}$, и любая успешная проверка на свойство быть женщиной будет означать ложность высказывания *Ведьма*(*Жанна д'Арк*), что вновь бессмысленно.

Такая же аргументация может быть применена по отношению к любым пустым высказываниям, в действительности не имеющим отношения к реальности. Ясно, что причиной систематических провалов в определении ложности подобных высказываний является отсутствие у них истинного аналога в реальности R. Отсюда вывод: *прежде чем какое-то высказывание* A *получит статус ложного, необходимо уметь обосновать истинность* $\neg A$. Этими соображениями мотивировано принятие следующего определения:

Высказывание A *вычислимо ложно* $\rightleftharpoons$ Высказывание $\neg A$ *вычислимо истинно.*

Проще говоря, фраза «*высказывание* A *вычислимо ложно*» не может означать ничего иного, как сокращение выражения «*высказывание* $\neg A$ *вычислимо истинно*». Значит, вычислимая истина иногда (а именно в случае $\neg A$) детерминирует вычислимую ложь, но никогда не наоборот. Следовательно, истина первична, а ложь производна, вторична. Без истины не обойтись, а ложь избыточна и о ней вообще можно было бы не упоминать. А раз так, то истина естественна, а ложь искусственна.

Расширим функцию интерпретации I, дополнив область ее определения не атомарными формулами. В результате каждой формуле A языка L будет сопоставлено определенное подмножество $I^n(A)$ декартова произведения универсума U^n. Это позволит трактовать формулы как представления вычислимых или не вычислимых множеств. Такая метаоперация выводит нас за границы стандартной семантики А.Тарского.

Реализация указанного замысла наталкивается на следующую проблему. Пусть для формул $P(x)$ и $Q(x, y)$ значения $I(P(x))$ и $I(Q(x, y))$ даны. Как и полагается, $I(P(x)) \subset U$ и $I(Q(x, y)) \subset U^2$. Предположим, формула A имеет вид $(P(x)\ \&\ Q(x, y))$. Тогда какое множество сопоставить A? Естественно было бы конъюнкции сопоставить пересечение, но пересечение $I(P(x)) \cap I(Q(x, y))$ заведомо пусто. Получится, к примеру, что простейшее утверждение «Сократ — человек и Сократ старше Платона» заранее обречено на пустоту. Формальных затруднений не возникает при понимании дизъюнкции как объединения: достаточно положить $I(P(x) \vee Q(x, y)) = I(P(x)) \cup I(Q(x, y))$. Но как вычислять такое смешанное множество, сколько параметров должно быть у соответствующей программы? Хотя многие компьютерные программы и команды допускают ввод разного числа параметров, логически это не естественно, коль скоро предполагается, что каждая программа вычисляет какую-то частичную или тотальную функцию.

Мы предлагаем простое решение возникшей проблемы. Пусть n — наибольшая местность, которую имеют атомарные предикаты из формулы A. Назовем число n *степенью формулы* A. Предположим, что интерпретация I определена для всех атомарных предикатов из A. Если входящий в A атомарный предикат $P(t_1, t_2, \ldots, t_m)$ таков, что $m < n$ (где n — степень формулы A), то, не меняя саму формулу A, изменим интерпретацию данного предикатного символа следующим образом. Вместо $I(P(t_1, t_2, \ldots, t_m)) = V_1 \times V_2 \times \ldots \times V_m \subset U^m$ положим $I^n(P(t_1, t_2, \ldots, t_m)) = V_1 \times V_2 \times \ldots \times V_m \times V_{m+1} \times V_{m+2} \times \ldots \times V_{m+(n-m)} \subset U^n$, причем, по определению, $V_m = V_{m+i}$ для всех i, $m < i \leq n$. Если же $m = n$, то полагаем $I^n(P(t_1, t_2, \ldots, t_m)) = I(P(t_1, t_2, \ldots, t_m))$.

Представляется, что такое расширение интерпретации не меняет ее содержательной сути. Зато теперь каждый атомарный предикатный символ из A, независимо от своей местности, интерпретируется как подмножество декартова произведения U^n, где n — степень формулы A. Отметим, что нет препятствий для того, чтобы с помощью той же техники определить более высокую степень интерпретации атомарных предикатов из A: для $k > n$ точно таким же способом, как и для n, определим $I^k(P(t_1, t_2, \ldots, t_m)) \subset U^k$. Данный прием *повышения степени интерпретации* атомарных предикатных символов потребуется в дальнейшем.

Перейдем к вычислительной интерпретации логических связок. Допустим, формула A имеет степень n, содержит k атомарных предикатных символов (ясно, что $k > 0$) и интерпретация $I^n(A)$ определена. Тем самым определена и интерпретация $I^n(P_i(t_1, t_2, \ldots, t_m))$ каждого атомарного предикатного символа $P_i(t_1, t_2, \ldots, t_m)$ из A, где $1 \leq i \leq k$. Допустим также, что для каждого атомарного предикатного символа $P_i(t_1, t_2, \ldots, t_m)$ из A существует релевантная для $I^n(P_i(t_1, t_2, \ldots, t_m))$ часть универсума R_i^n, так что выполняется включение $I^n(P_i(t_1, t_2, \ldots, t_m)) \subset R_i^n$. Определим релевантную часть универсума R_A^n для всей формулы A: $R_A^n = \cup\{R_{i \leq k}^n\}$. Иными словами, релевантная часть универсума для формулы A получается в результате объединения релевантных частей универсума ее атомарных предикатов. Поскольку в силу третьей аксиомы каждое R_i^n из $\{R_{i\leq k}^n\}$ непусто, R_A^n тем более не пусто. После этого, естественно, полагаем $I^n(\neg A) = R_A^n \backslash I^n(A)$. По определению, формула $\neg A$ *вычислимо выполнена* (*истинна*), если (1) предикат $I^n(\neg A)$ вычислим, и (2) $I^n(\neg A) \neq \varnothing$.

Речь здесь и в дальнейшем идет о *выполнимости*, если значение формулы зависит от функции приписывания значений свободным индивидным переменным, и об *истинности*, если значение формулы от приписываний не зависит, например, если свободных индивидных переменных в ней нет. В двух этих конкретных пунктах имеется полное соответствие с семантикой А.Тарского.

Предположим, что функции интерпретации I^k и I^m определены для формул A и B. Лишь в этом случае имеет смысл рассматривать интерпретацию формул вида $(A\&B)$ и $(A\vee B)$. Перейдем к вычислительной характеризации логических операций конъюнкции & и дизъюнкции $\vee$. Предварительно вычислим степень n формул $(A\&B)$ и $(A\vee B)$ (ясно, что она будет одинаковой для этих формул и, к тому же, $n = k$, если $k \geq m$, или $n = m$, если $m \geq k$) и переопределим с помощью приема повышения степени интерпретации $I^k(A)$ и $I^m(B)$, получив в результате $I^n(A)$ и $I^n(B)$. Формула $A\&B$ *вычислимо выполнена* (*истинна*), если (1) предикат $I^n(A\&B) = I^n(A) \cap I^n(B)$ вычислим, и (2) $I^n(A\&B) \neq \varnothing$. Формула $A \vee B$ *вычислимо выполнена* (*истинна*), если (1) предикат $I^n(A\vee B) = I^n(A) \cup I^n(B)$ вычислим, и (2) $I^n(A \vee B) \neq \varnothing$.

Будем опускать индексы степеней интерпретации и писать I вместо I^n и R вместо R^n, памятуя о всегда имеющейся возможности уравнения степеней интерпретации всех подформул формулы A с помощью приема повышения степени. Вообще, от любого *конечного* набора формул $\{A_1, A_2, \ldots, A_k\}$, для которого определены интерпретации $\{I^{n_1}(A_1), I^{n_2}(A_2), \ldots, I^{n_k}(A_k)\}$, всегда можно перейти к унифицированному по степени интерпретации набору $\{I^n(A_1), I^n(A_2), \ldots, I^n(A_k)\}$, где $n \geq n_i$ $(1 \leq i \leq k)$. Разумеется, это не так в случае бесконечных множеств формул.

ТЕОРЕМА 3.

$\dashv$ *(a). Если $I(A)$ и $I(B)$ вычислимы, то $I(A\&B)$ вычислим.*

$\dashv$ *(b). Если $I(A\&B)$ вычислим, то $I(A)$ и $I(B)$ вычислимы.*

$\vdash$ *(c). Если $I(A)$ вычислим, $I(B)$ вычислим и $\exists x(x \in I(A\&B))$, то $I(A\&B)$ вычислим.*

$\vdash$ *(d). Если $I(A)$ и $I(B)$ вычислимы, то $I(A \vee B)$ вычислим.*

Доказательство.

(a). При $I(A\&B) = \varnothing$ предикат $I(A\&B)$ в общем случае не является вычислимым (см. обсуждение постулата локального пересечения).

(b). Возьмем в качестве компьютера C МНР-машину. Тогда на множестве формул классического исчисления предикатов предикат *Теорема*(x) вычислим, а предикат *Выполнима*(x) не вычислим. Положим $I(A) =$ *Теорема*(x) и $I(B) =$ Выполнима(x). Тогда $I(A) \cap I(B) =$ *Теорема*$(x) \cap$ Выполнима$(x) =$ *Теорема*$(x) = I(A\&B)$. Отсюда предикат $I(A\&B)$ вычислим, а предикат $I(B)$ не вычислим.

(c). Следует из постулата локального пересечения.

(d). Следует из постулата локального объединения. Q.E.D.

Если считать импликацию $(A \to B)$ самостоятельной логической операцией, то не вполне понятно, какой вычислительный смысл эта операция может иметь. Наверное, недаром импликация отсутствует в большинстве языков программирования. Проще поступить обычным образом: положить, по определению, $(A \to B) \rightleftharpoons (\neg A \vee B)$. В этом случае интерпретация формулы $(A \to B)$ предполагает существование предикатов $I(\neg A)$ и $I(B)$. Тогда $I(A \to B) = I(\neg A) \cup I(B)$.

Остается рассмотреть выполнимость и истинностные условия для формул с кванторами. В каком случае считать выполненными и истинными формулы вида $\forall x A(x)$ и $\exists x A(x)$? Первый шаг очевиден: предположим, что предикат $I(A(x))$ определен. В случае с квантором общности второй шаг должен состоять в предположении, что определено релевантное (в нашем смысле) отрицание предиката $I(A(x))$, т.е. предикат $R \backslash I(A(x))$. Тогда на третьем шаге получаем: формула $\forall x A(x)$ *вычислимо выполнена* (*истинна*), если (1) предикат R вычислим и (2) $R \backslash I(A(x)) = \varnothing$

(или, что то же самое, $R = I(A(x))$). Отсюда вытекает, что следует принять по определению: $I(\forall x A(x)) = R$, если $I(A(x)) = R$, и $I(\forall x A(x)) = \varnothing$, если $I(A(x)) \neq R$.

Второй шаг в случае с квантором существования совершенно иной. Здесь вообще не требуется отнесения к релевантному универсуму R. Вместо этого достаточно потребовать, чтобы выполнялись требования вычислимости и непустоты: формула $\exists x A(x)$ *вычислимо выполнена* (*истинна*), если (1) предикат $I(A(x))$ вычислим и (2) $I(A(x)) \neq \varnothing$. Тогда логично установить $I(\exists x A(x)) = I(A(x))$, если $I(A(x)) \neq \varnothing$, и $I(\exists x A(x)) = \varnothing$, если $I(A(x)) = \varnothing$. Отсюда принимаем по определению $I(\exists x A(x)) = I(A(x))$.

Таким образом, квантор общности действует как переключатель, дающий либо R, либо $\varnothing$ в зависимости от значения $I(A(x))$, тогда как квантор существования сохраняет значение $I(A(x))$ и в этом смысле ничего не меняет.

В итоге был представлен метод *вычислительной интерпретации* формул классического первопорядкового языка, сводимый к двум этапам: на первом определялось множество $I(A)$, а на втором ставился вопрос о вычислимости множества $I(A)$. При этом отнюдь не всякой правильно построенной формуле была гарантирована вычислительная интерпретация из-за проблем с релевантным отрицанием на первом этапе и из-за невычислимости некоторых множеств на втором этапе. Тем не менее, верно следующее общее утверждение.

ТЕОРЕМА 4. *Формула A вычислимо выполнена (истинна) $\Longleftrightarrow$ предикат $I(A)$ вычислим и $I(A) \neq \varnothing$.*

Доказательство. Доказательство осуществляется индукцией по числу логических связок и кванторов в формуле A. Q.E.D.

Подробное изучение свойств введенных логических операций оставим на будущее. Обсудим лишь некоторые частные случаи. Начнем с формулы $(P(x) \vee \neg P(x))$. Если релевантный универсум для P не определен, говорить не о чем. Если же имеется соотвествующее $R \subset U$ и $I(P(x)) = V$, где $V \subset R$, то $I(\neg P(x)) = R \backslash V$. Тогда формуле $(P(x) \vee \neg P(x))$ сопоставляется множество $I(P(x) \vee \neg P(x)) = I(P(x)) \cup I(\neg P(x)) = V \cup R \backslash V = R$. Если $R =$

U, то формула $(P(x) \vee \neg P(x))$ выполнена при каждом приписывании v, т.е. классически истинна. Но вычислимо истинна она будет лишь в том случае, если множество U вычислимо, что не гарантировано. Но если $R \neq U$, классически $(P(x) \vee \neg P(x))$ уже не будет истинной. Зато, учитывая $V \cup R \backslash V = R$ и $R \neq \varnothing$, при условии вычислимости множества R, формула $(P(x) \vee \neg P(x))$ будет вычислимо истинна: $(P(x) \vee \neg P(x))$ вычислимо истинна $\Longleftrightarrow R$ вычислим. Более того, при тех же условиях будет вычислимо истинна и формула $\forall x(P(x) \vee \neg P(x))$, поскольку предикат R вычислим и $R \backslash I(P(x) \vee \neg P(x)) = \varnothing$. Непустота и вычислимость релевантного универсума R обеспечат вычислимую истинность и формулы $\exists x(P(x) \vee \neg P(x))$. Однако невычислимость R все разрушит.

Формула $(P(x) \mathbin{\&} \neg P(x))$ ни при каких условиях не может стать вычислимо истинной. Действительно, $I(P(x) \mathbin{\&} \neg P(x)) = I(P(x)) \cap I(\neg P(x)) = V \cap R \backslash V = \varnothing$. Но вычислимая истинность ее отрицания вновь зависит от вычислимости R. Поскольку $R_{(P(x) \,\&\, \neg P(x))} = \cup\{R_{P(x)}\} = R$, имеем $I(\neg(P(x) \mathbin{\&} \neg P(x))) = R \backslash I(P(x) \mathbin{\&} \neg P(x)) = R \backslash \varnothing = R \neq \varnothing$. Отсюда $\neg(P(x) \mathbin{\&} \neg P(x))$ вычислимо истинна $\Longleftrightarrow R$ вычислим. И снова вычислимость R обеспечит вычислимую истинность формул $\forall x \neg(P(x) \mathbin{\&} \neg P(x))$ и $\exists x \neg(P(x) \mathbin{\&} \neg P(x))$.

Предложенное введение в теорию вычислимой истины является дальнейшим развитием концепции истины как трехместного предиката, обсуждавшейся в упомянутых работах автора. Истина является не характеристикой высказывания A самого по себе, и не определяется (как у А.Тарского) сочетанием «высказывание — модель», т.е. парой $< A, I(A) >$. Интерпретация I лишь придает высказыванию A *смысл.* Необходима третья компонента, позволяющая соотнести высказывание A вместе с вложенным в него смыслом $I(A)$ с *реальностью* $\mathbb{C}$. Тем самым получаем трехкомпонентную структуру $< A, I(A), \mathbb{C} >$: *высказывание A с приписанным смыслом $I(A)$* **истинно** (*или* **не истинно**) *по отношению к реальности* $\mathbb{C}$. В предыдущих работах автора по проблеме истины и реальности последняя была описана теоретико-множественным образом. Здесь реальность $\mathbb{C}$ представлена вычислительными устройствами (компьютерами) и реализуемыми на них вычислительными процессами, тогда как смысловые характеристики оставались теоретико-множественными. В какой мере плодотворно представление изменяющейся реальности в виде вычислимого универсума — покажет будущее.

Литература

[1] *Анисов А.М.* Время и компьютер. Негеометрический образ времени. М., 1991.

[2] *Анисов А.М.* Темпоральный универсум и его познание. М., 2000.

[3] *Анисов А.М.* Проблема реальности в семантической теории истины // Труды научно-исследовательского семинара Логического центра Института философии РАН. Вып. XVII. М., 2004.

[4] *Анисов А.М.* Определение понятия реальной истины в теории множеств с атомами // Логические исследования. Вып. 11. М., 2004.

[5] *Анисов А.М.* Понятие реальности и логика // Логические исследования. Вып. 12. М., 2005.

[6] *Беркович С.Я.* Клеточные автоматы как модель реальности: Поиски новых представлений физических и информационных процессов. М., 1993.

[7] *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. М., 1983.

[8] *Крушинский А.А.* Логика «И цзина»: дедукция в древнем Китае. М., 1999.

[9] *Хармут Х.* Применение методов теории информации в физике. М., 1989.

[10] *Poundstone W.* The Recursive Universe. Cosmic Complexity and the Limits of Scientific Knowledge. N.Y., 1985.