УДК 681.3

О ВОЗМОЖНОСТЯХ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

С.В. Листровой, канд. техн. наук, А.И. Тимочко, канд. техн. наук, А.Ю. Гуль, канд. техн. наук

Харьковский институт Военно-Воздушных Сил им. Ивана Кожедуба

В работе показана принципиальная возможность построения интеллектуальных вычислительных систем, позволяющих строить непротиворечивые теории на основе конструктивных объектов. Предложен алгоритм, дающий возможность полностью формализовать процесс построения непротиворечивых теорий.

* * *

У роботі показана принципова можливість побудови інтелектуальних обчислювальних систем, що дозволяють будувати несуперечливі теорії на основі конструктивних об'єктів. Запропоновано алгоритм, що дає можливість цілком формалізувати процес побудови несуперечливих теорій.

* * *

In work the basic opportunity of construction of intellectual computing systems allowing to build the not inconsistent theory on the basis of constructive objects is shown. The algorithm giving an opportunity completely to formalize process of construction of the inconsistent theories is offered.

Постановка проблемы. Вопрос о перспективах передачи интеллектуальных функций человека вычислительной машине является предметом широкой дискуссии между ученными различных направлений. Сейчас вычислительные системы используются для решения четко формализованных задач. Американский специалист по искусственному интеллекту, Г. Саймон в 1960 г. предсказывал, что за 25 лет можно построить вычислительные системы, на которые можно возложить все функции выполняемые человеком в организационных процессах. Известный профессор Массачусетского технологического института М. Минский заявлял, что «при жизни нашего поколения останется лишь немного интеллектуальных задач, которые будут не под силу машинам, проблема создания «искусственного интеллекта» будет в основном решена».

Анализ известных достижений. Однако, судя по современным публикациям об интеллектуальных системах [1,3], большинство ученых стало скептически относится к возможности создания «искусственного интеллекта». При этом в качестве основных аргументов выдвигаются результаты полученные в теории алгоритмов, а именно: показано существование алгоритмически не разрешимых проблем, например проблемы «останова» для машины Тьюринга, теоремы Райса для вычислимых функций и др., а также доказательстве теоремы Геделя о существовании условий, при которых любые дедуктики неразрешимы [2]. Таким образом, складывается впечатление принципиальной невозможности построения «искусственного интеллекта» на основе вычислительных систем.

Цель работы. В данной работе сделана попытка снова перейти на оптимистическую ноту в вопросе создания «искусственного интеллекта».

Постановка задачи

Прежде чем обсуждать проблему, нужно определиться с тем, что мы будем вкладывать в понятие «интеллектуальная вычислительная система». Один из известных подходов заключается в том, что вам предлагается в процессе общения с системой, например, по телефону, выяснить: вы говорите с человеком или вычислительной системой. Если вам не удается установить, что с вами общается вычислительная система, то такую систему нужно признать «интеллектуальной вычислительной системой». Однако такой подход с точки зрения построения «интеллектуальных вычислительных систем» не конструктивен. Поэтому постараемся поэтапно конкретизировать данное понятие.

Пусть имеется вычислительная система (ВС) в которую в виде программ введена некоторая теория. Предположим, в рамках данной теории известно некоторое подмножество M истинных высказываний в виде теорем. Если ВС сможет выдать нам новое истинное высказывание $1 \notin M$, то будем называть такую ВС интеллектуальной, а способность ВС перечислять истинные высказывания $\{l_i\}$, принадлежащие и не принадлежащие M, интеллектуальной способностью ВС.

ВС, обладающую интеллектуальной способностью, отнесем к интеллектуальным системам 1-го уровня. Предположим, что в ВС вводится некоторое подмножество показателей качества и ВС перечисляет истинные высказывания, удовлетворяющие требуемым значениям введенных показателей качества. Такую ВС будем называть интеллектуальной системой 2-го уровня. Если на основе ВС второго уровня построить самовоспроизводящую ВС, то такую ВС назовем интеллектуальной системой 3-го уровня, а если в процессе самовоспроизведения ВС сама выбирает показатели качества и их значения, то такую ВС будем называть интеллектуальной системой 4-го уровня. Рассмотрим возможность построения ВС различных уровней.

Анализ подходов к решению задачи

В математике истинность высказываний и справедливость теорем основывается на понятии «доказательства». Хотя термин «доказательство» является едва ли не самым главным в математике, оно не имеет точного определения. Понятие доказательства во всей своей полноте принадлежит математике не более чем психологии. Ведь «доказательство» это просто рассуждение, убеждающие нас настолько, что с его помощью мы готовы убеждать других. Тем не менее, если теория не противоречива, то метод полной математической индукции и метод рассуждения от противного являются общепринятыми средствами доказательства теорем. Для большей определенности будем полагать, что доказательство – это определение с помощью алгоритма А того, является ли высказывание истинным или ложным. Тогда процесс перечисления истинных высказываний, можно рассматривать как процесс построения некоторой теории T, состоящей из некоторого подмножества истинных высказываний, полученных в рамках некоторого алфавита L, на основе которого формируются слова, образующие высказывания. Алфавит L состоит из букв $\{l_i\}$ которые по определенным правилам R^0 объединяются в слова, а слова по правилам R^1 объединяются в высказывания. Система высказываний, объединенная правилами R^2, образует теорию T. Фактически четверка $<A, R^0, R^1, R^2>$ образует дедуктику над алфавитом L, которая эквивалентна тройке $<Д, D, \delta>$ в теореме Геделя о противоречивости дедуктики, где Д- алфавит доказательств, D-подмножества, элементы которого являются доказательствами, δ-функция выделения доказанного. Рассмотрим следующие теоремы Геделя о противоречивости дедуктики:

Теорема 1. Если T – перечислимое множество, то для фундаментальной пары $<L,T>$ можно построить полную и непротиворечивую дедуктику.

Теорема 2. Если посредством фундаментальной пары $<L,T>$ выразима принадлежность хотя бы к одному не перечислимому множеству натуральных чисел, для $<L,T>$ не может существовать полной непротиворечивой дедуктики.

Как следует из теоремы 2, появление бесконечных множеств является индикатором противоречивости теории. Под противоречивостью понимается наличие утверждений в рамках одной теории, которые нельзя ни доказать, ни опровергнуть. Следует отметить, что теория бесконечных множеств, основателем которой явился Кантор, и развитая в дальнейшем в работах Бореля, Шварца, Лебега, Лузина, Урысона, Цермело, Френкеля, Брауэра, Гильберта

[4], изобилует противоречиями и парадоксами. В бесконечных множествах понятиями истинно и ложно нужно пользоваться очень осторожно. Брауэр более радикален и считает, что в суждениях, полученных с привлечением актуальной бесконечности, понятие истинно или ложно нельзя использовать. Из теоремы 1 следует, что для конечных объектов, относящихся к конструктивным, дедуктики полны и разрешимы. В работах Колмогорова [3] введены (Б,К) комплексы на основе понятия конструктивных объектов. При этом под конструктивным объектом понимаются объекты, о которых можно мыслить, не привлекая абстракции «актуальной бесконечности», т.е. объект может быть предъявлен целиком. В место термина конструктивный объект Колмогоров часто употреблял термин «состояние» алгоритмического процесса. Все конструктивные объекты могут быть заданы конечными графами, что, собственно, и представляют (Б,К) комплексы Колмогорова [3].

Формализация и решение задачи

В общем случае конструктивный объект состоит из конечного множества базовых элементов $L = \{l_i\}$ мощности W, из которого на основе правил R^p он может строиться, состоя из r базовых элементов, где $r \leq W$. В принципе сам базовый элемент может быть сложно устроен и сам состоять из $\{l_i^/\}$ объединенных между собой по правилам R^{p-1}. При этом будем полагать, что правила R^p и R^{p-1} и отношения, определенные внутри этих правил, строго определены и не вносят никаких неопределенностей в процесс формирования конструктивных объектов. Сложность устройства базовых элементов может быть сколь угодно большой, но конечной, т. е. можно выделить первичное множество $\{l_i^0\}$ базовых элементов мощности W^0 из которых по правилам $R=(R^0, R^1,R^2,\ldots, R^n)$ может быть построен произвольный конструктивный объект. Если для построения конструктивного объекта привлекаются правила $R^0,R^1,R^2,\ldots,R^k$, то будем говорить, что построенный конструктивный объект имеет сложность k. Каждому базовому элементу $\{l_i\}$ поставим в соответствие $m+1$ весовую характеристику определяемую по правилам R_ν, которые тоже должны быть конечны, строго определены, и должны позволять определять весовые характеристики порожденных конструктивных объектов любой сложности k. Таким образом, конструктивный объект сложности k тоже характеризуется $m+1$ весовой характеристикой. Одну весовую характеристику будем считать основной, а остальные m вспомогательными, и, в общем случае, на них могут налагаться ограничения. Обозначим множество всех объектов, которое можно построить на основе базовых элементов $\{l_i\}$, через Ω. Тогда подмножество объектов $\Omega^/\in\Omega$, которое можно построить, используя правила R и R_ν удовлетворяющие ограничениям на вспомогательные m характеристик, будем называть подмножеством конструктивных объектов, удовлетворяющих свойству ν. Построим граф G^1, вершины которого соответствуют базовым элементам $\{l_i^0\}$, которые соединены в соответствии с правилами R^0. Если рассматривать базовые элементы как буквы некоторого алфавита L, тогда сами объекты, построенные по правилам R^0, можно рассматривать как слова алфавита L. Далее, построив множество всех слов $\{l_i^1\}$ на основе графа G^0, и рассматривая их как новое базовое множество элементов $\{l_i^1\}$, построим, аналогично используя правила R^1, граф G^2, в котором вершинам соответствует множество возможных высказываний $\{l_i^2\}$. Используя множество высказываний опять как базовые элементы и правила R^2, можно построить граф G^3, в котором множества $\{l_i^3\}$ образуют множества теорий. Продолжая процесс построения мы можем получать последовательность графов

$$G^0,\ G^1,\ G^2,G^3,G^4,\ldots,G^n, \qquad (1),$$

соответствующих теориям различной сложности. В данной последовательности графов, граф G^0 соответствует алфавиту, граф G^1 словарю некоторой предметной области,а графы G^i при $i \geq 2$ теориям различной сложности.

Построим алгоритм *B*, позволяющий формировать графы G^i из исходных базовых элементов. В общем случае объект может быть произвольным, но должно быть определено конечное множество элементов $\Omega=\{\omega_l\}$ или подмножества $L_i \in \Omega$ и правила *R*, позволяющие формировать объекты из исходных элементов или подмножеств L_i, принадлежащие множеству Ω. Пусть задано некоторое разбиение множества Ω на семейства подмножеств $\{L_i\}$ такое, что $\bigcup_i L_i = \Omega$ (L_i описывают интересующие нас объекты), состоящие из базовых элементов $\{l_i\}$ таких, что $\bigcup_i l_i = \Omega$, и правило *R*, позволяющее из базовых элементов определять весовые характеристики произвольных объединений $L_k \bigcup L_p \in \Omega$, характеризующие свойства $\{v\}$ и требуется определить объект с интересующим нас свойством $v^* \in \{v\}$. Представим множество всех возможных объединений подмножеств L_i в виде графа $D_{\square}$ (рис.1) с параллельно ярусной структурой, состоящей из *n* горизонтальных линеек с вершинами *1,2,…n* и n ярусами,

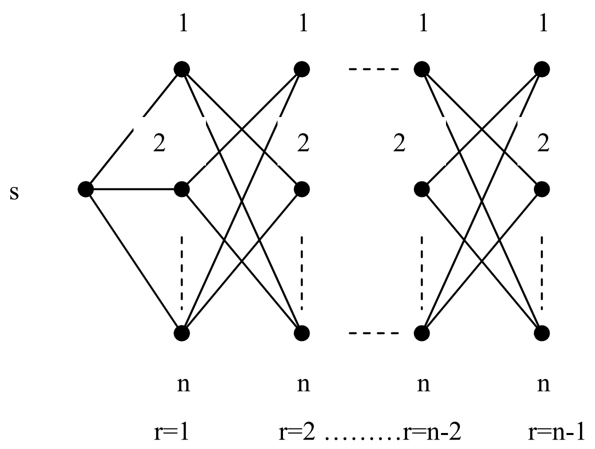

Рис. 1. Граф с параллельно ярусной структурой

каждый из которых содержит все вершины графа $D_{\square}$. При этом каждой вершине графа $D_{\square}$ поставим в соответствие базовый элемент l_i.

В графе $D_{\square}$ произвольная вершина i может быть достигнута путями рангов $r=1,\ r=2, \ldots r=n-1$, а произвольному пути μ_{st}, удовлетворяющему правилам построения *R* и проходящему через вершины $(j,p,\ldots k,t)$, соответствует объединение базовых элементов $(l_j\ U\ l_p\ U \ldots U\ l_k\ U\ l_t)$, определяющих некоторый объект $L_i \in \Omega$. Длина этого пути $d(\mu_{st})$ определяется по правилам, принадлежащим множеству *R*. Следовательно, множество всех путей $m_{si}(r)$ в графе $D_{\square}$, удовлетворяющих правилам *R*, определяют область допустимых решений исходной задачи по выделению объекта с интересующим нас свойством $v^* \in \{v\}$. В качестве исходной вершины в графе $D_{\square}$ будем использовать фиктивную вершину *S*, которую в некоторых случаях удобно отождествлять с нулевым или исходным состоянием системы. Это приводит к тому, что максимальный ранг пути в графе $D_{\square}$ становится равным *n*, а добавление вершины *S* к базовым элементам системы не изменяет их свойств, определяемых правилами *R*.

Множество сложных систем, обладающих различными свойствами $\{F_i\}$, может быть отображено с помощью некоторого подмножества графов $\{G_i\}$. Рассмотрим произвольный *n*-вершинный граф $G(V,E) \in \{G_i\}$, который описывает состояние системы $F \in \{F_i\}$ с конечным числом состояний *n*. Вершины $\{i\} \in V$ графа $G(V,E)$ соответствуют возможным состояниям системы, пути в графе $G(V,E)$, определяемые последовательностью прохождения вершин $\{v_i\}$ и ребер $\{(i,j)\}$, характеризуют возможный порядок достижения состояния $i=p$ из некоторого исходного состояния *s*. Важной характеристикой пути является ранг пути *r* – число ребер, образующих путь. В графе $G(V,E)$ максимальное значение ранга $r=n-1$, и, в общем случае, ранг произвольного пути μ_{sp} характеризует сумму начального состояния, конечного состояния и числа состояний предшествования, через которые может быть достигнуто состояние *p* из некоторого исходного состояния *s*. Тогда множества путей $m_{sj}^r\ ;\ j=(\overline{1,n})$ определяют способы достижения состояния j. Поскольку мы установили взаимно однозначное соответствие базовых элементов $\{l_i\}$ и множества $\{v_i\} \in V$ вершин графа $G(V,E)$, то объектам $\{L_j\}$ будут соответствовать все множество объектов Ω, которое можно породить на множестве V, используя правила R. Каждый объект может характеризоваться $m+1$ весовой характеристикой, где m – это некоторые второстепенные характеристики объекта, на которые в общем случае могут быть наложены ограничения на то, что они не должны превышать некоторых величин $\{b_i\};\ i=(1,2\ldots m)$, и имеется один определяющий показатель качества объекта. Правила P определения весовых характеристик объектов естественно должны определятся в соответствии с правилами формирования самих объектов и $P \in R$. Таким образом, пути μ_{sj}^{r} в графе $D_{\square}$ соответствует объект L_j который может быть построен из r базовых элементов $\{v_j\}$, включая элемент j, на основе правил R, а множества путей $m_{sj}^{r}\ ;\ j=(\overline{1,n})$ определяют множество объектов L_j, которые можно построить из r базовых элементов $\{v_j\}$, включая элемент j.

Введем обобщенную процедуру A_0 для формирования путей в графе $D_{\square}$, позволяющие перечислять все объекты множества $\{L_j\}$, и $A_0^{/}$ – для определения объектов $\{L_j\}$ с интересующим нас свойством, при этом рассмотрение начнем со случая, когда определяющей характеристикой объектов $\{L_j\}$ является одна весовая характеристика объекта.

Процедура A_0

Шаг 1.Формируем в графе $D_{\square}$ и з вершины S в множества путей $m_{sj}^{r=1}$ все возможные пути ранга $r=1$, удовлетворяющие правилу R.

Шаг 2. На основе путей текущего ранга r формируем все возможные пути ранга $r:= r+1$ в подмножествах $m_{sj}^{r:=r+1}$, удовлетворяющие правилу R.

Шаг 3. Проверяем, пусты ли подмножества $m_{sj}^{r:=r}$ текущего ранга r. Если нет, то переходим к выполнению шага 2, иначе процедура заканчивает работу, поскольку все объекты $L_j \in \Omega$ перечислены.

Процедура A_0 при переходе от произвольного ранга r к рангу $r+1$ позволяет строить из объектов L_j, содержащих r базовых элементов v_j, объекты L_j, содержащие $r+1$ базовый элемент v_j

Обобщенная процедура $A_0^{/}$

Шаг 1. Из вершины S строятся пути ранга r=1, удовлетворяющие правилам R, и, в соответствии с правилами R, определяются их весовые характеристики.

Шаг 2. На основе путей текущего ранга r строятся все возможные пути следующего ранга $r= r\ +1$, удовлетворяющие правилам R на основе следующего рекуррентного соотношения

$$\mu_{SP}^{r+1} = \min_{d_j(m_{sj}^{r} \cup (j,p))} \begin{pmatrix} (max)\{ m_{sj}^{r} \bigcup (j,p)\}\ ; \\ j=(\overline{1,n})\ ;\ p=(\overline{1,n}); \\ j \neq p \end{pmatrix} \qquad (2)$$

где весовые характеристики d_l объектов определяются в соответствии с правилами $P \in R$.

Шаг 3. Проверяем, пусты ли подмножества m_{sj}^{r} текущего ранга r. Если нет, то переходим к выполнению шага 2, иначе процедура заканчивает работу, и из полученных локальных экстремумов на всех рангах выбирается глобальный экстремум.

В зависимости от интересующих свойств объектов и правил R может возникнуть необходимость в работе процедуры $A_0^{/}$ до достижения некоторого конкретного значения ранга $r=k$. Используя процедуры A_0 и $A_0^{/}$ можно построить алгоритм B порождения графов последовательности (1).

Алгоритм *B*

Шаг 1. На основе процедуры A_0 перечисляем все возможные объекты и используя правила R^0 удаляем из полученного множества объекты не являющиеся словами в рассматриваемом языке и далее, используя правила $R^n \in R$, можно удалить и слова которые не принадлежат той предметной области, для которой будет строиться теория. Т.е. мы получим словарь Q^n, состоящий из множества слов $\{q_i\}$.

Шаг 2. Устанавливаем взаимно однозначное соответствие между множеством вершин графа $D_{□}$ и множеством слов $\{q_i\}$. Для этого можно перенумеровать все слова в Q^n,и первой вершине графа $D_{□}$ поставить в соответствие слово q_1, второй вершине графа $D_{□}$ слово q_2, и т.д.

Шаг 3. Используя процедуру $A_0^{/}$ строим все возможные высказывания, удовлетворяющие интересующему нас свойству ν, т.е. получаем теорию, обозначенную в последовательности (1) как G^2.

Ясно, что используя алгоритм B можно построить теорию любой сложности k, если существуют правила R^k, которые можно использовать для выделения базовых высказываний, необходимых для создания теории данной сложности k. Например, если существуют правила R^3 для выделения базовых высказываний в теории G^2, то опять установив взаимно однозначное соответствие между базовыми высказываниями теории G^2 и вершинами графа $D_{□}$и, применив к ним процедуру $A_0^{/}$, мы получим теорию G^3.

Таким образом, на основе алгоритма B вычислительная система может построить непротиворечивую теорию, позволяющую в рамках рассматриваемой предметной области перечислять истинные высказывания. Мы предположили, что для определения истинности высказываний в синтезируемых теориях G^i существует алгоритм A. Множеству слов U, полученных из начального алфавита $\{l_i^0\}$, можно поставить множество векторов $\{X_1,X_2,\dots, Xn\}$, в которых X_i=1, если буквы из $\{l_i^0\}$ присутствует в слове $u_p \in U$ и объединены в слово в соответствии с правилами объединения R^0, и $X_i=0$ в противном случае. Тогда для описания всего множества слов можно использовать булеву функцию

$$F(x) = \sum_{r=1}^{k} P_r(C_n^r),$$

где

$$P_r(C_n^r) = \sum_{j=1}^{j!} C_{rj} S_{rj};$$

$$S_{rj} = \underbrace{(X_p X_k \dots X_m)}_{r};$$

$$k = \frac{n!}{r!(n-r)!};$$

$P_k(C_n^r)$-функция, содержащая сумму всех j перестановок r переменных образующих сочетание C_n^r. В случае использования правил R^0 коэффициенты C_{rj} могут принимать значения, равное 1 и 0. Если C_{rj}=0, то это значит, что слово u_p в виде j-й перестановки из C_n^r сочетания букв и правила R^0 не может быть построено в алфавите $\{l_i^0\}$, и C_{rj}=1 в противном случае. Таким образом, множество всех наборов переменных, на которых F(x) принимает значение «истинно», будет соответствовать словам алфавита $\{l_i^0\}$, а наборы переменных, на которых $F(x)$ принимает значение «ложно», соответствуют словам, не принадлежащим алфавиту. Аналогичным образом с помощью булевой функции $F(x)$ можно описать все истинные высказывания при выполнении правил R^1. Известно, что проблема определения истинности высказываний в теории формальных языков может быть сведена к решению задач «выполнимость» и «3 выполнимость» [5,6]. Задача «выполнимость» может быть легко преобразована в «3 выполнимость» [6], путем введения дополнительных фиктивных переменных. Рассмотрим булеву функцию $f(x_1,x_2,\dots,x_n)$ в конъюктивной форме записи

$$f(x_1,x_2,\dots,x_n) = (x_1^{\sigma 11} \vee x_2^{\sigma 12} \vee \dots \vee x_n^{\sigma 1n}) \wedge \dots$$
$$\wedge (x_1^{\sigma m1} \vee x_2^{\sigma m2} \vee \dots \vee x_n^{\sigma mn}),$$

где

$$x_i^{\sigma} = \begin{cases} x_i, -при\ \sigma = 1 \\ \overline{x_i} - при\ \sigma = 0 \end{cases}$$

Операции $\vee,\wedge$ являются булевыми и моделируют простейшие логические высказывания: $\vee$- «ИЛИ»; $\wedge$-«И». Для любого двоичного набора $x = (x_1, x_2, .., x_n)$ функция принимает одно из двух возможных значений: единицу или нуль. Задача «выполнимость» заключается в ответе на вопрос: существует ли набор значений переменных $x = (x_1, x_2, .., x_n)$, обращающий функцию f в единицу.

Произвольная задача выполнимость может рассматриваться как задача k-выполнимость на наборе из k переменных, где k определяется дизъюнктом с максимальным числом переменных, и при этом те переменные, которые отсутствуют в некоторых дизъюнкциях, будем обозначать 0^*, который не инвертируется при переходе от X_i=1 к X_i=0. Как показано в [6], если дизъюнкт C_t содержит k-переменных $C_t=(x_1Vx_2V\ldots Vx_k)$, k>3, то возможна замена C_t на k-2 дизъюнкта

$$(x_1Vx_2Vu_1)(\overline{u_1}\ Vx_3Vu_2)(\overline{u_2}\ Vx_4Vu_3)(\overline{u_{k-3}}\ Vx_{k-1}Vx_k),$$

где u_1, u_2, u_{k-3} **-** новые переменные, причем набор новых дизъюнктов выполним тогда и только тогда, когда выполним дизъюнкт C_t. После сведения задачи k-выполнимость к 3-выполнимость число дизъюнктов возрастет в (k-2) раза, а число переменных в (k-3) раза и, следовательно, при решении её как 3-выполнимость [6] на наборе из $m(k-3)$ переменных мы получим временную сложность решения задачи k-выполнимость не превышающую $\boldsymbol{O(26m(k-2)+m\log_2 3+3m)}$ (где m – число скобок в конъюнктивном представлении логической функции). В работе [7] предложен алгоритм решения задачи «3 выполнимость» с временной сложностью, не превышающей $\boldsymbol{O(26m)}$ при ее решение на наборе из трех переменных. Итак, мы имеем алгоритм A, позволяющий за линейное время определять истинность высказываний в рамках некоторого алфавита L и, следовательно, с помощью алгоритмов B и A мы можем построить непротиворечивую теорию, при этом наращивать сложность теории мы можем только в рамках алфавита базовых элементов. Возникает вопрос: если мы имеем две непротиворечивые T_1 и T_2 теории, каждая со своим алфавитом, можно ли создать новую теорию T_3, объединяющую обе, т. е. такую, что T_1 и T_2 являются частными случаями T_3. Естественно рассматривать возможность объединения только тех теорий, которые имеют общие базовые элементы алфавита. Если $L_1 \cap L_2 = \varnothing$, то такие теории существуют независимо, и в рамках этих теорий не может существовать высказываний, которые справедливы в рамках одной теории T_1 и не справедливы в рамках другой теории T_2, поскольку предметная область у них разная. Например, «теория кодирования» и «теория построения механизмов». Таким образом, именно пересечение алфавитов $L_1 \cap L_2$ с одной стороны дают принципиальную возможность объединять теории, и это же пересечение дает возможность для возникновения противоречивых высказываний. Например, в геометрии Евклида справедливо высказывание «параллельные прямые не пересекаются», а в геометрии Лобачевского справедливо противоположное высказывание «параллельные прямые пересекаются». Введение определенных правил R позволяют объединить эти две теории и показать, что они не противоречивы и что геометрия Евклида является частным случаем геометрии Лобачевского. Таким образом, $|L_1 \cap L_2|$ можно использовать как оценку сверху возможной противоречивости теории T_3, т.е. числа высказываний, для которых необходимо ввести правила R, согласующие эти две теории. Правила R будем называть согласующими для теорий T_1 и T_2, если их введение позволяет исключить все противоречивые высказывания в рамках объединенной теории T_3. Относительно алфавита L_1 теория T_1 не противоречива и, соответственно, относительно алфавита L_2 теория T_2 тоже не противоречива. Выделим высказывание $1 \in L_1 \cap L_2$ о возможности построения некоторого объекта P, которое истинно в T_1 и ложно в T_2. Тогда, если мы попытаемся проанализировать возможность существования объекта P в рамках алфавита $L_1 \cup L_2$, используя метод рассуждения от противного в рамках обоих теорий, мы сможем получить высказывание вида: объект P существует (т.е. может быть построен), если P не существует (т.е. не может быть построен) и, вследствие противоречивости этого высказывания, будет сделан вывод о том, что объект P не существует. Именно таким образом доказывается проблема «останова», т.е. не возможности построения машины Тьюринга (T), решающей проблему «останова», где сначала для произвольных данных α на основе метода рассуждения от противного получают высказывание, что существует машина T, которая останавливается, если T не останавливается, и, следовательно, делается вывод о невозможности построения машины T, решающей проблему останова. Отсюда в силу тезиса Тьюринга: «Всякий алгоритм может быть реализован машиной Тьюринга» делается вывод об отсутствии алгоритма решения данной проблемы, т.е. признается существование алгоритмически неразрешимой проблемы. Любопытно типичное истолкование полученного результата [5] «невозможность построения машин T решающей проблему «останова» в общем случае, не исключает возможность ее решения в частных случаях». Таким образом, для множества частных случаев, которые никак не определяются, а рассматриваются как опыт построения программ, следует обобщение о том, что проблемы «останова» для них не существует, т.е. мы пришли к противоречию. Возникает вопрос: высказывание «проблема «останова» неразрешима» истинно или ложно? Для того чтобы однозначно ответить на этот вопрос, следует признать, что понятие истинности или ложности относительно к исходным данным, т.е. к исходному базовому алфавиту. По всей видимости, нельзя говорить об абсолютности истинности или ложности высказываний в отрыве от входных данных базового алфавита и способа их кодировки, в рамках которых данное высказывание определено. Так высказывание 1 о возможности построения объекта P относительно алфавита L_1 истинно, а по отношению к алфавиту $L_1 \cup L_2$ оно не определено. Если мы с помощью правил R^3 (по умолчанию предполагая, что они нас не выводят за рамки понятий конструктивных объектов) доопределим все противоречивые высказывания в виде аксиом принимаемых в теориях T_1 и T_2 как истинные, то тогда получим непротиворечивую теорию T_3 в конечном алфавите $L_1 \cup L_2$. Вопрос о существовании таких правил зависит от свойств базовых объектов, определяющих исходный алфавит и, следовательно, выходит за рамки формальной разработки самих теорий.

Выводы

Итак, мы рассмотрели два подхода к созданию непротиворечивых теорий. Один, основанный на постепенном усложнении теории в соответствии с последовательностью (1), построенной в рамках алфавита заданных базовых элементов, на основе предложенного алгоритма B. Второй путь связан с возможностью объединения не противоречивых теорий, имеющих общее подмножество базовых элементов алфавита и требующий введения правил согласования, устраняющих противоречия, причем сами правила выходят за рамки процесса формализации построения самой теории, поскольку определяются свойствами базовых элементов, являющихся исходными данными для формирования самой теории. При этом возникает довольно сложный вопрос алгоритмической разрешимости проблемы, которая, как показано, является относительным по отношению исходным данным. Оба пути имеют право на свое существование, но второй путь, по всей видимости, приемлем для создания метатеорий и не может быть автоматизирован полностью, поскольку на этапе создания правил, устраняющих противоречие, нужно вмешательство человека. Первый путь фактически полностью формализован, и вычислительная система будет только перечислять истинные высказывания, а человек будет являться их пользователем. На основе первого подхода для каждой предметной области может быть легко построен язык и словарь терминов и понятий, удобный для пользователя. При этом создаваемые языки будут такими, какие мы используем в повседневной жизни, но несколько ограниченные в возможностях формирования высказываний в рамках предметной области. Сам процесс создания таких языков может быть автоматизирован за счет создания соответствующего программного обеспечения на основе уже имеющегося опыта построения языков программирования, и с помощью его легко корректироваться и расширяться в зависимости от потребностей пользователя и в общем случае не требует разработки новых технологий построения вычислительных систем, хотя для эффективной реализации алгоритма В, как показано в работах [8-10], могут быть применены вычислительные системы циклического типа.

Литература

1. Юдин Д.Б., Юдин А.Д. Число и мысль // Выпуск 8, –М., Издательство «Знание», 1985. - 189 с.
2. Успенский В.А. Теорема Геделя о неполноте. – М.: Наука. Главная редакция физико-математической литературы, 1982. - 111 с.
3. Успенский В.А., Семенов А.Л. Теория алгоритмов: основные открытия и приложения – М.: Наука. Главная редакция физико-математической литературы, 1987. – 288 с.
4. Виленкин Н.Я. В поисках бесконечности.// АН СССР, Серия «Наука и технический прогресс», издательство «Наука», 1983. – 160 с.
5. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера – М.: Энергия, 1980. – 342 с.
6. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985.
7. Листровой С.В. Метод решения задачи 3 выполнимость.// Электрон. моделирование № 6. – К.: 2001. С. 66-76.
8. Листровой С.В. Архитектура параллельных вычислительных систем циклического типа. //Электрон. моделирование 1992, №2, том 14, с. 28.
9. Листровой С.В. Голубничий Д.Ю. Листровая Е.С. Метод решения задач целочисленного линейного программирования с булевыми переменными на основе рангового подхода// Электрон. моделирование, т.20, №6, 1998, с. 14-32.
10. S.V. Listrovoy, D.Yu. Golubnichiy and E.S. Listrovaya. Solution Method on the Basis of Rank Approach for integer Linear Programming Problems with Boolean Variables// Engineering Simulation, 1999, Vol.16, pp. 707-725.

Поступила в редакцию 26.09.03

Рецензент: д-р техн. наук, профессор Сироджа И.Б., Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», г. Харьков