Н. К. Косовский, А. В. Тишков

ГРАДУИРУЕМЫЕ ЛОГИЧЕСКИЕ ЗНАЧЕНИЯ ДЛЯ ПРЕДСТАВЛЕНИЯ ЗНАНИЙ

(РЕЗУЛЬТАТЫ ДОЛОЖЕНЫ 6 МАЯ 1996 Г.)

Описаны секвенциальные исчисления для содержащих сравнения предложений логик Поста, а также изучены общие первоначальные свойства этих исчислений, предназначенных для представления знаний в персональном компьютере.

§1. Предикатные логики Поста со сравнениями

Предлагается использовать целые числа в качестве логических значений для градуируемого представления знаний. Представляемые в информационных системах знания удобно градуировать, например, по степени первоочередности их использования. Для этих целей предлагается использовать логики Поста [1], дополненные сравнениями логических значений предикатных формул. Формулируются секвенциальные исчисления для таких логик. Статья является дальнейшим развитием [2]. Предлагаемые логики способны на практике заменить нечеткие и непрерывные логики [3, 4], поскольку константу 1 можно интерпретировать (не меняя самих логик) как очень малую единицу измерения степени истинности.

Предлагаемые исчисления удобно применять при анализе аксиоматических систем, присвоив разным предикатам разные уровни очевидности (истинности). Тогда в рамках одной аксиоматической теории можно соединить несколько, например, геометрию без предиката параллельности с евклидовой геометрией. Эти две аксиоматические системы выделяются на основе неотрицательных логических значений утверждений (евклидова геометрия) и максимального положительного логического значения утверждений (геометрия без предиката параллельности).

Аналогичным образом можно использовать различные логические значения для предикатов различных наук, соединив несколько наук в единую иерархическую систему. Например, можно предложить, чтобы логические значения у специфических предикатов наук возрастали бы в следующей последовательности: социология, психология, философия, другие гуманитарные науки, естественные науки, начинающиеся биологией и заканчивающиеся химией, физикой, математикой и, наконец, математической логикой.

Для представления знаний предлагается использовать только числа, легко представимые в персональном компьютере (целые числа типа integer). Фиксируем положительное целое число I. Атомарная формула исчисления SP_I, соответствующая логике Поста P_I, представляет собой логическую константу или атомарную формулу классического исчисления предикатов. Логические значения атомарных формул и констант – целые числа из сегмента $[-\lfloor I/2\rfloor, \lceil I/2\rceil - 1]$. Скажем, что этот сегмент является *сегментом логических значений* для SP_I.

Логические значения конъюнкции и дизъюнкции определяются соответственно как минимум и максимум логических значений аргументов. Операция отрицания Лукасевича соответствует одноместному минусу при нечетном Iи дополнительно уменьшается на единицу при четном I. Материальная импликация $(x \Rightarrow y)$ имеет логическое значение $\max(-[x], [y])$, где $[x]$, $[y]$ - логические значения x и y соответственно. Традиционным образом можно определить и эквивалентность.

Логическое значение выражения $\exists x A$ равно $\max\limits_x[A(x)]$. Логическое значение выражения $\forall h A$ равно $\min\limits_x[A(x)]$.

Все логические значения, большие нуля, являются истинными или достоверными (большей или меньшей степени (или уровня) достоверности). Все логические значения, меньшие нуля, являются ложными. При нечетном Iлогическое значение, равное нулю, называется парадоксальным. Парадоксальное (нулевое) логическое значение имеют любые парадоксы, например, парадокс лжеца. При четном Iнуль является минимально истинным логическим значением.

Представляют интерес атомарные формулы, которые могут быть более или менее достоверными. Примерами могут быть использованы условие “x не меньше y” (логическое значение равно разности аргументов либо ближайшее из чисел $-\lfloor I/2\rfloor$и $\lceil I/2\rceil - 1$, если разность выходит за пределы логических значений для SP_I) и “x больше y” (логическое значение равно разности аргументов, уменьшенной на единицу или, опять же, ближайшее из чисел $-\lfloor I/2\rfloor$и $\lceil I/2\rceil - 1$, если разность, уменьшенная на единицу, выходит за пределы логических значений для SP_I). Таким образом удается различить предикаты, совпадающие при традиционной нечеткой интерпретации.

Предикатные формулы исчисления SP_Iопределяются как обычно с использованием отрицания, конъюнкции, дизъюнкции, четырехместной связки
$if(B \leqslant A) then\, C\, else\, D\, fi$, и кванторов, начиная с атомарных формул исчисления SP_I. Условное выражение интерпретируется как в языках программирования.

Пусть знак сравнения одного из следующих видов $\leq$, $<$, $=$, $\in$ используется между предикатными формулами (последний знак – между предикатной формулой и конечным множеством предикатных формул). Такую формулу будем называть логическим сравнением исчисления SP_Iили кратко – *сравнением*.

Логические значения сравнений определяются обычным образом (в классической двузначной логике).

Предикатная формула со сравнениями исчисления SP_I определяется так же, как предикатная формула SP_I, но атомарными формулами для нее служат сравнения. Логические связки и кванторы интерпретируются как в классической двузначной логике.

§2. Аксиомы и правила вывода секвенциального исчисления SP_I

Секвенция представляет собой слово, состоящее из знака секвенции $\rightarrow$ и конечного списка предикатных формул со сравнениями (членов секвенции).

Формульным образом секвенции является дизъюнкция интерпретаций предикатных формул со сравнениями. Интерпретация секвенции, содержащей только знак $\rightarrow$, ложна.

Для определения того, является ли секвенция аксиомой SP_I, вычеркнем из членов секвенции все неатомарные предикатные формулы со сравнениями, далее из оставшихся сравнений вычеркнем содержащие бинарные логические связки, кванторы и знаки $=$, $\in$. После этого строится система линейных неравенств. Нечетное число логических отрицаний перед формулой заменяется на одно отрицание, четное – удаляется. Логические отрицания заменяются унарными минусами при нечетном Iи дополнительно уменьшаются на единицу при четном I. Исходная секвенция является аксиомой, если и только если эта система несовместна в целых числах из сегмента $[-\lfloor I/2\rfloor, \lceil I/2\rceil - 1]$.

Например, секвенция $\to (A \leqslant B)(B < A)$ является аксиомой.

Понятие вывода в SP_I, выводимой секвенции в SP_I определяется как обычно. Предикатная формула A называется выводимой в SP_I, если в SP_Iвыводима секвенция $\to (0 \leqslant A)$.

Для всех Iисчисления SP_Iимеют одни и те же правила вывода.

Формулируя правила, будем пользоваться прописными курсивными буквами для обозначения предикатных формул, прописными полужирными буквами – для обозначения предикатных формул со сравнениями и буквой Γ для обозначения пустого списка или конечного (быть может пустого) списка формул со сравнениями.

• Правило для перестановки членов секвенции. • Правила для отрицания внутри сравнения.

$$\frac{\to \Gamma\ (B < A)}{\to \Gamma\ (\neg A < \neg B)}\ (\neg{<}\neg), \qquad \frac{\to \Gamma\ (\neg B < A)}{\to \Gamma\ (\neg A < B)}\ (\neg{<}),$$

$$\frac{\to \Gamma\ (A < \neg B)}{\to \Gamma\ (B < \neg A)}\ ({<}\neg),$$

и аналогичные правила с заменой $<$ на $\leqslant$.

• Правила для отрицания вне сравнения.

$$\frac{\to \Gamma\ (B < A)}{\to \Gamma\ \neg(A \leqslant B)}\ \neg(\leqslant), \quad \frac{\to \Gamma\ (B \leqslant A)}{\to \Gamma\ \neg(A < B)}\ \neg(<), \quad \frac{\to \Gamma\ \mathbf{A}}{\to \Gamma\ \neg\neg\mathbf{A}}\ (\neg\neg).$$

• Правила для конъюнкции и дизъюнкции внутри сравнения.

$$\frac{\to \Gamma\ (A < C)\ (B < C)}{\to \Gamma\ ((A \& B) < C)}\ (\& <), \qquad \frac{\begin{array}{l}\to \Gamma\ (C < A)\\ \to \Gamma\ (C < B)\end{array}}{\to \Gamma\ (C < (A \& B))}\ (< \&),$$

$$\frac{\begin{array}{l}\to \Gamma\ (A < C)\\ \to \Gamma\ (B < C)\end{array}}{\to \Gamma\ ((A \vee B) < C)}\ (\vee <), \qquad \frac{\to \Gamma\ (C < A)\ (C < B)}{\to \Gamma\ (C < (A \vee B))}\ (< \vee),$$

и аналогичные правила с заменой $<$ на $\leqslant$, а так же правила

• Правила для конъюнкции и дизъюнкции вне сравнений.

$$\frac{\begin{array}{l}\to \Gamma\ \mathbf{А}\\ \to \Gamma\ \mathbf{Б}\end{array}}{\to \Gamma\ \mathbf{А}\&\mathbf{Б}}\ (\&), \qquad \frac{\to \Gamma\ \mathbf{АБ}}{\to \Gamma\ \mathbf{А}\vee\mathbf{Б}}\ (\vee),$$

$$\frac{\to \Gamma\ \neg\mathbf{А}\neg\mathbf{Б}}{\to \Gamma\ \neg(\mathbf{А}\&\mathbf{Б})}\ (\neg\&), \qquad \frac{\begin{array}{l}\to \Gamma\ \neg\mathbf{А}\\ \to \Gamma\ \neg\mathbf{Б}\end{array}}{\to \Gamma\ \neg(\mathbf{А}\vee\mathbf{Б})}\ (\neg\vee).$$

• Правила для условного выражения.

$$\frac{\begin{array}{l}\to \Gamma\ (B \leqslant A)\ (D \leqslant E)\\ \to \Gamma\ (A < B)\ (C \leqslant E)\end{array}}{\to \Gamma\ (if(B \leqslant A)then\,C\,else\,D\,fi \leqslant E)}\quad (if\ \ \leqslant),$$

$$\frac{\begin{array}{l}\to \Gamma\ (B \leqslant A)\ (E \leqslant D)\\ \to \Gamma\ (A < B)\ (E \leqslant C)\end{array}}{\to \Gamma\ (E \leqslant if(B \leqslant A)then\,C\,else\,D\,fi}\quad (\leqslant\ \ if),$$

$$\frac{\begin{array}{l}\to \Gamma\ (B \leqslant A)\ (D < E)\\ \to \Gamma\ (A < B)\ (C < E)\end{array}}{\to \Gamma\ (if(B \leqslant A)then\,C\,else\,D\,fi < E)}\quad (if\ \ <),$$

$$\frac{\begin{array}{l}\to \Gamma\ (B \leqslant A)\ (E < D)\\ \to \Gamma\ (A < B)\ (E < C)\end{array}}{\to \Gamma\ (E < if(B \leqslant A)then\,C\,else\,D\,fi)}\quad (<\ \ if).$$

• Правила для равенства и неравенства.

$$\frac{\begin{array}{l}\to \Gamma\ (A \leqslant B)\\ \to \Gamma\ (B \leqslant A)\end{array}}{\to \Gamma\ (A = B)}\ (=), \qquad \frac{\to \Gamma\ (B < A)(A < B)}{\to \Gamma\ \neg(A = B)}\ \neg(=).$$

• Правила для принадлежности и непринадлежности.

$$\frac{\to \Gamma\ (X_0 = X_1)(X_0 = X_2)\ldots(X_0 = X_m)}{\to \Gamma\ (X_0 \in \{X_1, X_2, \ldots, X_m\})}\quad (\in),$$

$$\frac{\begin{array}{l}\to \Gamma\ \neg(X_0 = X_1)\\ \to \Gamma\ \neg(X_0 = X_2)\\ \ldots\\ \to \Gamma\ \neg(X_0 = X_m)\end{array}}{\to \Gamma\ \neg(X_0 \in \{X_1, X_2, \ldots, X_m\})}\quad \neg(\in).$$

• Кванторные правила внутри сравнения.

$$\frac{\to \Gamma\ ([A]^x_T < B)\ (\forall x A < B)}{\to \Gamma\ (\forall x A < B)}\ (\forall <), \qquad \frac{\to \Gamma\ (B < [A]^x_y)}{\to \Gamma\ (B < \forall x A)}\ (< \forall),$$

$$\frac{\to \Gamma\ ([A]^x_y < B)}{\to \Gamma\ (\exists x A < B)}\ (\exists < B), \qquad \frac{\to \Gamma\ (B < [A]^x_T)\ (B < \exists x A)}{\to \Gamma\ (B < \exists x A)}\ (< \exists),$$

и аналогичные правила с заменой $<$ на $\leqslant$.

• Кванторные правила вне сравнений.

$$\frac{\to \Gamma\ [\mathbf{A}]^x_y}{\to \Gamma\ \forall x \mathbf{A}}\ (\forall), \qquad \frac{\to \Gamma\ [\mathbf{A}]^x_T\ \exists x \mathbf{A}}{\to \Gamma\ \exists x \mathbf{A}}\ (\exists),$$

$$\frac{\to \Gamma\ \exists x \neg \mathbf{A}}{\to \Gamma\ \neg(\forall x \mathbf{A})}\ \neg(\forall), \qquad \frac{\to \Gamma\ \forall x \neg \mathbf{A}}{\to \Gamma\ \neg(\exists x \mathbf{A})}\ \neg(\exists).$$

Кванторные правила имеют стандартные ограничения: предметная переменная y и терм T свободны для подстановки в A и $\mathbf{A}$ вместо свободного вхождения x и, кроме этого, переменная y не входит свободно в заключения правил $(\exists \leqslant)$, $(\exists <)$, $(\leqslant \forall)$, $(< \forall)$ и $(\forall)$.

Исчисление предназначено для вывода секвенций, обладающих свойством чистоты переменных, т.е. тех секвенций, которые не имеют свободных и связанных вхождений какой-либо одной переменной.

§3. Основные свойства исчислений SP_I

Лемма 1. *Для любого конечного $I > 1$ в предложенной интерпретации секвенций все аксиомы являются истинными и заключения правил этого исчисления являются истинными, если посылки истинны.*

Идея доказательства. Аксиомы истинны, поскольку их определение практически совпадает с понятием истинности интерпретации.

Возьмем, например, правило

$$\frac{\to \Gamma\ (A < C)\ (B < C)}{\to \Gamma\ ((A \& B) < C)}\ (\& <).$$

Пусть секвенция $\rightarrow \Gamma$ невыводима. Тогда если посылка истинна, то $A < C$, или $B < C$, и, следовательно, минимум из A, B меньше C. Для других правил утверждение леммы доказывается аналогично.

Лемма 2 (о допустимости правила утончения). *Правило* $\frac{\rightarrow \Gamma}{\rightarrow \Gamma \Phi}$ *допустимо.*

Идея доказательства. Доказательство аналогично традиционному доказательству для логики первого порядка.

Формулу Φ можно добавить ко всем аксиомам в выводе $\rightarrow \Gamma$, предварительно преобразовав вывод в вывод с чистыми переменными. В результате получится вывод $\rightarrow \Gamma\Phi$.

Теорема 1. *Все правила* SP_I *обратимы.*

Доказательство. Обратимость правил $(\prec \exists)$ $(\forall \prec)$, где $\prec$ означает либо $<$, либо $\leqslant$, а также правила $(\exists)$ следует из допустимости правила утончения.

Правила $(\neg \prec)$ и $(\prec \neg)$ обратны сами себе.

Допустимость остальных обратных правил докажем индукцией по длине вывода посылки.

База индукции. Посылка – аксиома.

Для правил

$$\frac{\rightarrow \Gamma\ (\neg A \prec \neg B)}{\rightarrow \Gamma\ (B \prec A)} \quad (\neg \prec \neg)^{-}.$$

база индукции очевидна, поскольку неравенства $-A \prec -B$ и $B \prec A$ при нечетном Iи неравенства $-A-1 \prec -B-1$ и $B \prec A$ при четном Iсовпадают.

Рассмотрим оставшиеся правила удаления связок, правила $(\exists \prec)^{-}$, $(\prec \forall)^{-}$ и $(\forall)^{-}$ удаления кванторов, могут быть представлены в следующем виде:

$$\frac{\rightarrow \Gamma\Phi}{\rightarrow \Gamma\Psi} \quad ,$$

где Ψ – это список формул, а формула Φ содержит квантор или связку, которая должна быть удалена. Если посылка этого правила – аксиома, то по определению аксиомы $\rightarrow \Gamma$ – тоже аксиома, а значит и $\rightarrow \Gamma\Psi$ также является аксиомой.

Индукционный переход. Для каждого обратного правила проводим следующее рассуждение. Если на последнем шаге было применено соответствующее прямое правило, то оно применялось к секвенции, выводимость которой требуется установить. Если на последнем шаге вывода было применено какое-либо другое правило, то допустимость обратного правила следует из индукционного предположения. *Теорема доказана.*

Теорема 2. *Пусть A – формула классической логики первого порядка, не содержащая логических констант. Если секвенция $\to (0 \leqslant A)$ выводима в SP_I при некотором $I > 1$, то формула A доказуема в классической двузначной логике первого порядка.*

Идея доказательства. Вывод формулы $(0 \leqslant A)$ в SP_I заменим на вывод в секвенциальном исчислении классической логики, которое содержит правила для конъюнкции, дизъюнкции, отрицания и кванторов (в том числе правила для тех же связок и кванторов под отрицание) и два правила перестановки членов секвенции (в антецеденте и сукцеденте). Правила для связок и кванторов получаются из соответствующих правил для SP_I. Сначала отбрасывается $\to \Gamma$. Затем выражения вида $(A \leqslant B)(A \leqslant D)$ и $(A \leqslant C)(B \leqslant C)$ заменяются на выражения вида $\Gamma A \to \Delta BD$ и $\Gamma AB \to \Delta C$ соответственно, а выражения вида $(A \leqslant B)$ – на $\Gamma A \to \Delta B$, где Γ и Δ обозначают списки формул. При этом среди A, B, C, D удаляются формулы, не участвующие в записи главной формулы с вводимой логической связкой или квантором.

Например, правило $(\leqslant \vee)$ соответствует правилу

$$\frac{\Gamma \to \Delta AB}{\Gamma \to \Delta(A \vee B)} \quad (\to \vee),$$

а правило $(\forall \leqslant)$ – правилу

$$\frac{\Gamma\ [A]^x_T\ \forall x A \to \Delta}{\Gamma\ \forall x A \to \Delta} \quad (\forall \to).$$

Аксиомы будут только вида $\Gamma P \to \Delta P$, $\Gamma P \neg P \to \Delta$, $\Gamma \to \Delta P \neg P$. Можно доказать, что в выводе $\to (0 \leqslant A)$ не содержится ни одной аксиомы, для которой неразрешимость системы неравенств из определения аксиомы зависит от I(например, секвенция $\to (A \leqslant B)(B \leqslant C)$ для $I = 2$ является аксиомой, а для $I = 3$ – нет).

Имея вывод A в SP_Iмы можем получить вывод в только что описанном исчислении. Это может быть доказано индукцией по длине исходного вывода.

§4. Теорема о сложности задачи распознавания аксиом

Теорема 3 (о распознавании аксиом). *Задача распознавания аксиом разрешается алгорифмом класса $LIN-SPACE$ с полиномиальной временной сложностью.*

Доказательство. В системе неравенств, соответствующей изучаемой секвенции, добавим для каждого неравенства $x_{i1} \prec x_{i2}$ неравенство $-x_{i2} \prec -x_{i1}$. Кроме этого, для каждой переменной x запишем по два неравенства $x < \lceil I/2 \rceil$ и $-\lfloor I/2 \rfloor - 1 < x$. Очевидно, что получится эквивалентная исходной система и длина записи увеличится не более, чем линейно. В итоге получаем систему **C**

$$\begin{cases} x_{11} \prec_1 x_{12} \\ -x_{12} \prec_1 -x_{11} \\ x_{21} \prec_2 x_{22} \\ -x_{22} \prec_2 -x_{21} \\ \dots \\ x_{n1} \prec_n x_{n2} \\ -x_{n2} \prec_n -x_{n1}, \end{cases}$$

где для каждого i выполняется $\prec_i \in \{<, \leqslant\}$.

Система **C**

1. Система **C** содержит цикл $x_1 \prec_{i_1} x_2 \prec_{i_2} \dots \prec_{i_r} x_1$, при некотором $r \geqslant 1$. При этом хотя бы одно из неравенств строгое.
2. Пусть **C** содержит цикл $x_1 \leqslant x_2 \leqslant \dots \leqslant x_r \leqslant x_1$.
   - 2.1а. При четном I, $r \geqslant 2$ и x_i $(2 \leqslant i \leqslant r)$ равном $-x_1$.
   - 2.1б. При нечетном I, $r \geqslant 2$ и x_i $(2 \leqslant i \leqslant r)$ равном нулю или $-x_1$ система, полученная заменой всех элементов цикла нулем, несовместна.
   - 2.2. Если такой цикл содержит только одну константу и не содержит x и $-x$ ни для какого x, то система, полученная заменой всех элементов цикла этой константой, несовместна.
   - 2.3. Если такой цикл не содержит ни одной константы и ни одной пары x и $-x$, то система, полученная заменой всех элементов цикла на новую переменную, несовместна.

3. Система содержит цепочку $x_1 \prec_{i_1} x_2 \prec_{i_2} \ldots \prec_{i_{r-1}} x_r$, где x_1 и x_r и только они являются константами, и $x_r < x_1$.
4. **C** содержит цепочку $x_1 \prec_{i_1} x_2 \prec_{i_2} \ldots \prec_{i_{r-1}} x_r$, где x_i попарно различны, x_1 и x_r и только они являются константами, и длина цепочки, измеряемая в количестве знаков ‘ $<$ ‘, не меньше $x_r - x_1$.

Условия 2.1б, 2.2 и 2.3 позволяют перейти к более простой эквивалентной системе.

Поэтому далее считаем, что система не содержит циклов из условий бц.1б, 2.2 и 2.3.

Любое из условий 2.1а, 1, 3, 4 очевидным образом влечет несовместность **C**: первый случай невозможен по определению отрицания в SP_I с четным I, во втором имеем $x_i \leqslant x_j$ и при этом $x_i > x_j$, во третьем – равенство двух различных констант, и, наконец, в четвертом – ситуацию типа $2 < x < 3$ для целого x.

Пусть, напротив, ни одно из условий 1 – 4 не выполняется. Тогда **C** имеет решение. Действительно, поскольку циклов нет, переменные и константы системы представляют собой частично упорядоченную структуру, порожденную строгими и нестрогими неравенствами. При этом расстояние между любой парой констант достаточно, чтобы разместить цепочку целочисленных переменных, лежащую между этими константами со знаком строгого неравенства. Начиная, например, с переменных, наибольших, относительно нашего частичного порядка, нетрудно подобрать для них значения, не нарушающие этот порядок. Эти значения будут решением.

Представим систему **C** в виде ориентированного графа. Переменные и константы рассматриваются как вершины (x и $-x$ – разные вершины), а каждому неравенству $x \prec y$ – дугу из y в x. Дуги для строгих и нестрогих неравенств будем различать.

Производится поиск цикла обходом графа в глубину, используя стек и помечая встречающиеся вершины и дуги. Если дойдем до помеченной вершины, то цикл найден. Процесс заканчивается, когда нет пути из текущей вершины все вершины пройдены. Дополнительная память будет расходоваться только на стек и пометки поэтому, алгоритм использует только линейную память. Число шагов пропорционально числу вершин, умноженному на число дуг, что, очевидно, не превосходит некоторого полинома от длины записи графа. Этот алгорифм поиска цикла эффективнее по времени и памяти алгорифма поиска сильно связных компонент ориентированного графа, предложенного в [5, гл. 5], который также можно использовать для поиска циклов.

Если цикл найден, то, возвращаясь по входящим дугам, проверяем условия 1 – 2.3. В случаях 1 и 2.1а алгорифм закончит работу – система несовместна, в остальных случаях соответствующие замены уничтожат цикл.

Процедуру поиска цикла, возможно, придется повторить несколько раз, с каждым разом уничтожая какой-нибудь цикл, но не более, чем число вершин в графе. Это дает полиномиальную верхнюю оценку на число шагов алгоритма.

Если ни один из циклов не определил несовместность системы, или ни одного цикла не нашлось, то остается проверить условия 3 и 4 уже для ациклического графа.

Для случая 3 достаточно совершить обход всех путей, ведущих из каждой вершины-константы, сравнивая встречаемые константы с первоначальной.

Условие 4 предполагает поиск пути между вершинами-константами, содержащего наибольшее количество дуг, соответствующих строгим неравенствам. Алгоритм обход графа в ширину начинает работу с константы A и помечает вершины и дуги. Если в одну из текущих вершин входят не менее двух дуг, то удаляем те, которые заканчивают путь с меньшим числом дуг для строгих неравенств. Этот путь можно определить, возвращаясь по очереди по входящим дугам всеми помеченными путями. После удаления начинаем обход сначала с вершины A. Таким образом, обход в ширину, возможно, придется проделать несколько раз, но не более, чем число дуг в графе. Это показывает, что данная процедура полиномиальна по времени, а память расходуется только на список текущих вершин и два числа для нахождения наибольшего пути. Последний из этих обходов приведет из A в B требуемым путем. Вернувшись по входящим дугам, сравниваем количество дуг для строгих неравенств ц разностью A и B.

Поскольку этот алгоритм потребуется применить $O(m^2)$ раз, где m – число различных констант в графе, машина Тьюринга полиномиальна по временной сложности. *Теорема доказана.*

§5. Теорема о сложности процедуры вывода

Теорема 4. *Для каждого натурального числа I при $I > 1$ алгорифм проверки выводимости пропозициональных утверждений SP_I существует, принадлежит классу PSPACE и использует 2^{Cn} шагов машины Тьюринга при некотором C, где n – длина секвенции.*

Доказательство. Идея поиска вывода секвенции заключается в удалении бинарных логических связок из формул этой секвенции применением правил, обратных к правилам для этих связок (эти правила допустимы согласно теореме 1), и дальнейшем распознавании аксиом в получившихся секвенциях.

В дереве вывода каждый внутренний узел помечен секвенцией, являющейся заключением правила введения какой-то связки, а его сыновья помечены посылками этого правила. Корень помечен исходной секвенцией; листья помечены секвенциями, не содержащими бинарных логических связок. Внутренний узел дерева соответствует некоторой вводимой связке. Исходная секвенция выводима, если пометки всех листьев дерева – аксиомы.

Описание машины Тьюринга. Машина Тьюринга будет трехленточной.

На первой ленте машины Тьюринга записывается исходная секвенция.

Вторая лента отводится под регистры, по два на каждую связку. В первый регистр каждой связки будет занесено число посылок в правиле, вводящем эту связку. Во втором регистре будет лежать номер посылки правила, выводимой в данный момент. Регистровые пары упорядочиваются в соответствии со скобочной структурой исходной секвенции.

Третья лента будет содержать текущую секвенцию.

Алгорифм обходит все листья дерева, возвращаясь после каждого листа обратно к корню. Второй регистр каждого внутреннего узла указывает номер посылки правила, которую требуется вывести. Имея на третьей ленте секвенцию – пометку этого узла, применяется обратное правило и тем самым достигается продвижение на шаг по пути к листу.

Дойдя до листа, запускаем алгорифм распознавания аксиом, Если он выдает отрицательный результат, то исходная секвенция невыводима. Иначе переустанавливаем переменные регистры так, чтобы они указывали на путь к следующему листу. Сравнение со вторым регистром внутреннего узла при этом покажет, все ли посылки правила для этого узла выведены.

Если удастся обойти все листья и доказать, что их пометки – аксиомы, то исходная секвенция выводима.

Доказательство корректности алгорифма.

Доказательство проведем индукцией по общему количеству связок во всех формулах исходной секвенции.

База индукции. Если исходная секвенция не содержит связок, то корректность данного алгорифма эквивалентна корректности алгорифма распознавания аксиом.

Индукционный переход. Пусть мы выводим секвенцию у которой $\odot$ – самая внешняя связка. Выводимость нашей секвенции эквивалентна одновременной выводимости всех посылок правила введения $\odot$. Время работы машины Тьюринга можно разбить на три состояния: участки вывода посылок; в случае более чем одной посылки увеличение переменного регистра связки $\odot$ в промежутках между этими участками (если алгорифм до этой стадии дошел, т.е. все предшествующие посылки правила оказались выводимыми); и уничтожение регистровой пары связки $\odot$ (опять же, если ранее не выяснилось, что секвенция невыводима). Очевидно, что работа алгорифма на каждой из посылок и работа алгорифма проверяющего выводимость только посылки правила проделывается одна и та же. Легко убедиться, что алгорифм корректно переходит от одной посылки правила к другой. Теперь применяем индукционное предположение (посылки имеют по крайней мере на одну связку меньше) и заключаем, что на каждой из посылок правила машина работает верно, а значит работает верно и в целом. *Корректность алгорифма доказана.*

Верхняя оценка памяти, используемой машиной Тьюринга.

Для доказательства теоремы осталось показать, что максимальный используемый объем памяти всех трех лент не превышает $O(n)$, где n – длина входной формулы. Пусть $l(x)$ обозначает длину x.

Посчитаем расходуемую память для каждой ленты.

Первая лента содержит исходную секвенцию и остается неизменной.

Вторая лента потребует по два регистра на каждое вхождение логической связки в секвенцию: длина одного регистра равна длине записи числа посылок правила, которое будет применено для введения этой связки в выводе секвенции. Верхняя оценка памяти для второй ленты – $O(n)$, где n – длина входной формулы.

Третья лента содержит текущую секвенцию. Вложенная конъюнкция в левой части сравнения и вложенная дизъюнкция в правой части сравнения порождает повторения формулы C в правилах $(\& <)$, $(\& \leqslant)$, $(< \vee)$, $(\leqslant \vee)$. Остальные связки либо не допускают вложения, либо запись заключений правил для них больше записи каждой посылки. Таким образом, третья лента потребует памяти, не большей чем полиномиальной от длины исходной формулы.

Поскольку задача распознавания аксиом решается линейной по памяти машиной Тьюринга по теореме 3, использование алгорифма распознавания аксиом как подпрограммы не ухудшит полиномиальную верхнюю оценку на используемую память.

Поскольку более, чем двухпосылочное правило $\neg(\in)$ применяется только на самом внешнем уровне вложенности формул, общее число секвенций в листьях дерева вывода не превосходит 2^{Cn} при некотором C. По теореме 3 алгорифм распознавания аксиом полиномиален по времени, поэтому верхняя оценка числа шагов алгорифма выводимости может быть получена с использованием неравенств:

$$2^{Cn}p(n) \leqslant 2^{Cn+C_1 \log n} \leqslant 2^{C_2 n}$$

при некоторых C_1 и S_2. *Теорема доказана.*

Работа частично поддержана грантовым центром при Новосибирском госуниверситете, грантом РГНФ No. 97-05-12048 и федеральной целевой программой “Государственная поддержка интеграции высшего образования и фундаментальной науки на 1997–2000 годы” по комплексному проекту ”Развитие и поддержка системы Учебно-научных центров СПбГУ”. Регистрационный No. 326.53.

Литература

1. E. L. Post, *Introduction to a general theory of elementary proposition.* — Amer. J. Math. **43, No 3** (1921), 163–185 .
2. Н. К. Косовский, *Уровневые логики.* — Зап. научн. семин. ПОМИ **220** (1995), 72–82 .

3. Л. И. Волгин, В. И. Левин, *Непрерывные логики, теория и применение.* Таллинн (1991) .
4. L. A. Zadeh, *Fuzzy logic, neural network and soft computing.* — Comm. ACM **37** (1994), 77–89 .
5. А. Ахо, Дж. Хопкрофт, Дж. Ульман, *Построение и анализ вычислительных алгоритмов.* Мир, М. (1978) .

Kossovski N. K., Tishkov A. V. Gradable logical values for knowlege representation.

The paper is devoted to the description of sequent calculuses with the possibility of comparison of Post's logics' sentences and to study their main properties. These calculuses may be applied for computer representation of knowedge.

С.-Петербургский
государственный университет

Поступило 19 февраля 1997 г.