Разрешимость одно-переменных итеративных пропозициональных исчислений

Г. В. Боков

В работе рассматриваются одно-переменные итеративные пропозициональные исчисления, представляющие собой конечные множества пропозициональных формул от одной переменной вместе с операцией *modus ponens* и операцией суперпозиции, заданной множеством операций Мальцева. Для таких исчислений будет проведено сведение проблемы выводимости формул к проблеме вывода слов в линейных канонических системах. В частности, будет показано, что все одно-переменные итеративные пропозициональные исчисления разрешимы.

Ключевые слова: итеративные пропозициональные исчисления, проблема выводимости, линейные канонические системы.

Введение

Пропозициональное исчисление как пара — конечное множество пропозициональных формул в некоторой сигнатуре и множество операций над этими формулами — является неотъемлемой частью многих логических систем. Одной из центральных проблем, возникающей в этой связи, — это проблема разрешимости таких исчислений. Впервые данная проблема была поставлена Тарским [13] в 1946 году.

Существует много подходов к заданию пропозициональных исчислений, каждый из которых определяется своим множеством операций над формулами. Как правило, выбор таких операций зависит от специфики задач, для решения которых предназначены эти исчисления.

Отметим лишь классический подход, в котором операциями над формулами выступают операция *modus ponens* (из формул A и $A \to B$

выводима формула B) и операция *подстановки* (из формулы $A(x)$ выводима формула $A(B)$ для любой формулы B). Первое неразрешимое пропозициональное исчисление для классического подхода было построено Линиалом и Постом в 1949 году [11]. В 1975 году Хигис и Синглетари [10] доказали существование неразрешимого пропозиционального исчисления от трех переменных. Чуть позже Хигис [9] построил неразрешимое пропозициональное исчисление от двух переменных. Завершающее построение для классических исчислений было сделано Глэдстоуном [8] в 1979 году. Он доказал разрешимость всех пропозициональных исчислений от одной переменной.

В данной работе будет рассмотрен ослабленный вариант классического подхода, в котором операция подстановки заменена на операцию *слабой подстановки* (из формул $A(x)$ и B выводима формула $A(B)$). Как отмечает Циткин в [7], впервые данную операцию ввел в рассмотрение Кузнецов в 1965 году [3]. В [4, 5] данная операция и операция замены эквивалентным использовались для определения выразимости формул в той или иной логике относительной некоторой системы формул. Поскольку слабую подстановку можно представить в виде конечной последовательности операций Мальцева [2], то в [1] данную операцию было принято назвать операцией суперпозиции формул, а исчисления, в которых вместо операции подстановки используется операция суперпозиции, — итеративными, ввиду их схожести с итеративными алгебрами Поста, введенными Мальцевым [6].

В данной работе будет доказано, что любое одно-переменное итеративное пропозициональное исчисления разрешимо. Для этого будут определены линейные канонические системы, представляющие собой простейший вариант канонических систем Поста [12] с разрешимой проблемой вывода слов. Стоит отметить, что для доказательства разрешимости одно-переменных классических пропозициональных исчислений Глэдстоун в [8] использовал **L**-системы, которые являются частным случаем линейных канонических систем.

Определения и основные результаты

Язык пропозициональных исчислений состоит из счетного множества пропозициональных переменных $\mathcal{V}$ и конечного множества логических связок Σ, которое будем называть сигнатурой. Буквами x, y, p

будем обозначать переменные. Как правило, логические связки унарные или бинарные, например, $\neg$, $\vee$, $\wedge$ или $\to$.

Пропозициональные формулы или Σ-*формулы* строятся из логических связок Σ и переменных $\mathcal{V}$ обычным образом. Например, следующие обозначения

$$x, \quad \neg A, \quad (A \vee B), \quad (A \wedge B), \quad (A \to B)$$

являются формулами в сигнатуре $\{\neg,\ \vee,\ \wedge,\ \to\}$. Заглавные буквы A, B, C будут использоваться для обозначения формул. Далее условимся опускать внешние скобки, а также скобки, однозначно восстанавливаемые из частичного порядка логических связок. Формулу, зависящую от одной переменной $p \in \mathcal{V}$ будем называть p-*формулой*.

Итеративное пропозициональное исчисление P над множеством логических связок Σ это пара, состоящая из конечного множества Σ-формул P, называемых *аксиомами*, и двух правил вывода:

1) *modus ponens*

$$A, A \to B \vdash B;$$

2) *суперпозиция* (совокупность операций Мальцева)

$$A(x), B \vdash A(B).$$

Если P состоит из p-формул, то исчисление P будем называть *одно-переменным*.

Обозначим через $[P]$ множество выводимых (или доказуемых) формул исчисления P. *Вывод* в P из аксиом с помощью правил вывода определяется обычным образом. Выводимость формулы A из P будем обозначать через $P \vdash A$.

Исчисление $\mathcal{P}$ будем называть *разрешимым*, если существует алгоритм, который по произвольной формуле A отвечает на вопрос: $P \vdash A$? Основным результатом данной работы является следующая теорема.

Теорема 1. *Любое одно-переменное итеративное пропозициональное исчисление разрешимо.*

Доказательство основного результата

Прежде, чем преступить к доказательству теоремы 1, мы определим понятие линейной канонической системы.

Линейные канонические системы

В данном разделе мы рассмотрим частные случаи канонических систем Поста [12], которые будут полезны в двух аспектах. С одной стороны, проблема вывода слов для данных систем алгоритмически разрешима, с другой — операции данных систем позволяют эффективно имитировать логический вывод в пропозициональных исчислениях.

Определение. *Линейная каноническая система* над алфавитом $\mathcal{A}$ — это конечное множество L слов в алфавите $\mathcal{A}$ вместе с конечным множеством $\mathcal{R}$ *правил вывода* вида (C, V, α), где C и V — это конечные множества слов в алфавите $\mathcal{A}$, которые будем называть *посылками* правила, и α — это слово в алфавите $\mathcal{A}$, которое будем называть *заключением* правила.

Для правила (C, V, α) посылки C будем называть *константными*, а посылки V — *переменными*.

Отметим, что линейные канонические системы являются более широким классом, чем **L**-системы, которые рассматривались в работе Глэдстоуна [8]. В частности, правила **L**-систем не допускают константных посылок.

Рассмотрим произвольную линейную канонические систему L с правилами $\mathcal{R}$. *Выводом* в системе L будем называть всякую последовательность слов $\lambda_1, \ldots, \lambda_n$, каждый элемент λ_i которой либо принадлежит L, либо для него существует правило $(C, V, \alpha) \in \mathcal{R}$, удовлетворяющее следующим условиям:

1) $\lambda_i = \alpha\xi$, для некоторого слова $\xi \in \mathcal{A}^*$;
2) для каждого слова $\zeta \in C$ найдется такой индекс $j < i$, что $\zeta = \lambda_j$;
3) для каждого слова $\zeta \in V$ найдется такой индекс $j < i$, что $\zeta\xi = \lambda_j$.

При этом слово λ_n будем называть *выводимым* в системе L. Факт выводимости слова α в системе L будем обозначать через $L \vdash_{\mathcal{R}} \alpha$. Множество всех выводимых в системе L слов будем обозначать через $[L]_{\mathcal{R}}$, либо просто через $[L]$, когда множество правил фиксировано.

Для произвольного множества слов M обозначим для краткости

$$M\xi := \{\alpha\xi \mid \alpha \in M\}.$$

В 1979 году Глэдстоун [8] доказал, что проблема выводимости слов α в **L**-системах алгоритмически разрешима. Далее мы покажем, что доказательство Глэдстоуна проходит и для линейных канонические систем. Для этого введем аналог позитивных **L**-систем.

Правило вывода (C, V, α) будем называть *позитивным*, если длины его переменных посылок меньше длины его заключения, то есть для любого слова $\beta \in V$ выполнено $|\beta| < |\alpha|$. Линейную каноническую систему, у которой все правила вывода являются позитивными, назовем *позитивным*. Следующая лемма показывает, что можно ограничиться только рассмотрением позитивных линейных канонических систем.

Лемма 1. *Для любой линейной канонической системы L с конечным множеством правил $\mathcal{R}$ существует конечное множество позитивных правил $\mathcal{R}'$, для которого*

$$[L]_{\mathcal{R}} = [L]_{\mathcal{R}'}.$$

Доказательство. Если множество правил $\mathcal{R}$ пустое, то оно является позитивным и утверждение леммы верно. Поэтому будем предполагать, что $\mathcal{R}$ содержит хотя бы одно правило.

Обозначим через N максимальную длину заключений правил из $\mathcal{R}$. Если алфавит системы L состоит из бесконечного числа символов, то ограничимся рассмотрением его конечного подмножества, содержащего только символы, встречающиеся в L. Множество правил $\mathcal{R}'$ будем строить индуктивно. Положим $\mathcal{R}_0 = \varnothing$.

Предположим, что мы уже построили $\mathcal{R}_n$ для $n \geqslant 0$. Тогда $\mathcal{R}_{n+1}$ получается из $\mathcal{R}_n$ добавлением всевозможных позитивных правил (C', V', α'), удовлетворяющих следующим условиям:

1) $|\alpha'| \leqslant N$;
2) существует такое правило $(C, V, \alpha) \in \mathcal{R}$, что $C = C'$, $V\xi \subseteq [L \cup V']_{\mathcal{R}_n}$ и $\alpha\xi = \alpha'$ для некоторого слово $\xi \in \mathcal{A}^*$.

Поскольку существует только конечное множество позитивных правил (C', V', α'), для которых $|\alpha'| \leqslant N$, то наступит момент, когда процесс добавления новых правил закончится, то есть $\mathcal{R}_{m+1} = \mathcal{R}_m$ для некоторого $m \geqslant 1$. Положим в этом случае $\mathcal{R}' := \mathcal{R}_m$.

Остается показать, что выполнено равенство

$$[L]_{\mathcal{R}} = [L]_{\mathcal{R}'}.$$

По построению, применение любого правила из $\mathcal{R}'$ можно заменить применением соответствующего правила из $\mathcal{R}$, поэтому включение

$$[L]_{\mathcal{R}} \supseteq [L]_{\mathcal{R}'}$$

очевидно. Докажем обратное включение. Для этого достаточно доказать включение

$$[L]_{\mathcal{R}\cup\mathcal{R}'} \subseteq [L]_{\mathcal{R}'}.$$

Пусть $\gamma \in [L]_{\mathcal{R}\cup\mathcal{R}'}$. Рассмотрим минимальный вывод слова γ в L с помощью правил $\mathcal{R} \cup \mathcal{R}'$. Без ограничения общности можно считать, что данный вывод не содержит применений правил из $\mathcal{R}$, кроме последнего вывода финальной формулы. Обозначим это правило через (C, V, α).

Пусть $\gamma = \alpha\xi\zeta$ для некоторых $\xi, \zeta \in \mathcal{A}^*$ таких, что $|\alpha\xi| = N$, тогда по предположению выполнено включение $V\xi\zeta \subseteq [L]_{\mathcal{R}'}$. Докажем индукцией по $m = |\zeta|$, что $\gamma \in [L]_{\mathcal{R}'}$.

Если $m = 0$, то утверждение следует из определения множества $\mathcal{R}'$. Пусть утверждение верно для всех $m' < m$, докажем его для m. Положим

$$V' := \{\beta \mid \beta\zeta \text{ участвует в выводе элементов из } M\xi\zeta \text{ и } |\beta| < N\},$$

тогда правило $(C, V', \alpha\xi)$ по определению принадлежит $\mathcal{R}'$. Следовательно, $\gamma \in [L]_{\mathcal{R}'}$.

Лемма доказана.

Лемма 2. *Существует алгоритм, который по произвольному слову α и произвольной линейной канонической системе L с правилами вывода $\mathcal{R}$ отвечает на вопрос, выводимо ли слово α из L с помощью правил $\mathcal{R}$, то есть $L \vdash_{\mathcal{R}} \alpha$?*

Доказательство. По Лемме 1 существует такое конечное множество позитивных правил $\mathcal{R}'$, что выводимость $L \vdash_{\mathcal{R}} \alpha$ равносильна выводимости $L \vdash_{\mathcal{R}'} \alpha$. Поскольку, после применения позитивного правила вывода строго возрастает длина слова, то для доказательства того, что слово α не выводимо из L, достаточно проверить только слова, выводимые с помощью применения не более чем $|\alpha|$ правил из $\mathcal{R}'$. Следовательно, данный процесс разрешим. Лемма доказана.

Далее мы покажем, как можно закодировать словами формулы произвольного итеративного исчисления от одной переменной так, чтобы выводимость формул в данном исчислении была равносильна выводимости кодов соответствующих формул в некоторой линейной канонической системе.

Сведение вывода формул к выводу слов

Для начала заметим, что операция суперпозиции для двух p-формул A и B естественным образом определяет операцию конкатенации:

$$A \cdot B := A[B].$$

Нетрудно убедиться, что данная операция является ассоциативной, то есть

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C.$$

Длиной формулы A будем называть число символов в ней и обозначать как $|A|$. Длина формулы, состоящей из одной переменной, равна 1. Такие формула будем называть *тривиальными*.

Формулу от одной переменной A назовем *неприводимой*, если ее нельзя приставить в виде конкатенации двух нетривиальных формул одной переменной, то есть

$$A \neq B \cdot C$$

ни для каких нетривиальных формул одной переменной B и C.

Для таким образом определенной операции конкатенации формул верна следующая тривиальная лемма.

Лемма 3. *Любая формула одной переменной единственным образом представляется в виде конкатенации неприводимых формул.*

Теперь перейдем к кодированию формул словами. В качестве алфавита $\mathcal{A}$ рассмотрим множество всех неприводимых p-формул. Отметим, что данной алфавит является бесконечным, но, как это будет видно дальше, нам понадобится не весь алфавит, а лишь та его конечная часть, которая определяется неприводимыми формулами, соответствующими аксиомам некоторого одно-переменного итеративного исчисления.

Согласно Лемме 3 каждую формулу от одной переменной можно однозначно разложить в конкатенацию неприводимых формул. Поэтому *кодом p-формулы* A будем называть слово $\overline{A} \in \mathcal{A}^*$, соответствующее ее разложению на неприводимые формулы, то есть $\overline{A} = A$ для неприводимой формулы A и $\overline{A \cdot B} = \overline{A} \cdot \overline{B}$. Пустое слово ε является кодом формулы p.

Рассмотрим произвольное одно-переменное итеративное пропозициональное исчисление P. Обозначим через N максимальную длину аксиомы исчисления P. Сопоставим исчислению P следующую линейную каноническую систему L_P:

$$L_P := \{\overline{A} \mid A \in P\}.$$

Алфавитом системы L_P выступает множество $\mathcal{A}$ всех неприводимых формул, длина которых не превосходит N. В качестве правил вывода $\mathcal{R}_P$ системы L_P рассмотрим следующие две группы правил:

1) *Правила, кодирующие операцию суперпозиции*:

$$\big(\{\overline{A}\}, \{\varepsilon\}, \overline{A}\big)$$

для любой p-формулы A такой, что $|A| \leqslant N$;

2) *Правила, кодирующие операцию modus ponens*:

$$\big(\varnothing, \{\overline{A}, \overline{A \to B}\}, \overline{B}\big)$$

для всех p-формул A и B таких, что $|A \to B| \leqslant N$

Лемма 4. *Если* $P \vdash A$, *то* $L_P \vdash_{\mathcal{R}_P} \overline{A}$.

Доказательство. Доказывать будем индукцией по длине вывода. Если $A \in P$, то $\overline{A} \in L_P$ и, следовательно, $L_P \vdash_{\mathcal{R}_P} \overline{A}$.

Если $A = B \cdot C$, где B — неприводимая формула, и $P \vdash B, C$, то по предположению индукции $L_P \vdash_{\mathcal{R}_P} \overline{B}, \overline{C}$ и $|B| \leqslant N$. Тогда $(\overline{B}, \{\varepsilon\}, \overline{B}) \in \mathcal{R}_P$ и, следовательно, $L_P \vdash_{\mathcal{R}_P} \overline{A}$.

Пусть $P \vdash B, B \to A$, тогда согласно Лемме 3 существуют такие неприводимые формулы B' и A', что $B = B' \cdot C$ и $A = A' \cdot C$. Очевидно, что $B' \to A'$ — неприводимая формула. По предположению индукции $L_P \vdash_{\mathcal{R}_P} \overline{B}, \overline{B \to A}$ и $|B' \to A'| \leqslant N$, поэтому $(\varnothing, \{\overline{B'}, \overline{B' \to A'}\}, \overline{A'}) \in \mathcal{R}_P$ и, следовательно, $L_P \vdash_{\mathcal{R}_P} \overline{A}$.

Лемма доказана.

Лемма 5. *Если* $L_P \vdash_{\mathcal{R}} \overline{A}$, *то* $P \vdash A$.

Доказательство. Если код $\overline{A}$ формулы A выводим в линейной системе L_P с помощью правил $\mathcal{R}_P$, то рассмотрим вывод этого кода. Каждое слово в этом выводе является кодом некоторой p-формулы. Нетрудно убедиться, что применение правил системы L_P к кодам формул равносильно применению соответствующих операций исчисления P к этим формулам. Следовательно, вывод слова $\overline{A}$ в системе L_P однозначно определяет вывод формулы A в исчислении P. Лемма доказана.

Доказательство теоремы 1

Рассмотрим произвольное одно-переменное итеративное пропозициональное исчисление P. Согласно Леммам 4 и 5 разрешимость исчисления P равносильна разрешимости линейной канонической системы L_P. Поскольку, система L_P разрешима по Лемме 2, то тем самым мы доказали, что исчисление P также разрешимо. Теорема 1 доказана.

Заключение и дальнейшие исследования

В данной работе доказано, что, как и для классического случая, всякое одно-переменное итеративное пропозициональное исчисление разрешимо. Недавно автором был установлен факт существования неразрешимого итеративного пропозиционального исчисления от трех переменных. Естественно возникает вопрос о границе разрешимых и неразрешимых исчислений. В частности, остается открытым вопрос о существовании неразрешимого итеративного исчисления от двух переменных.

Список литературы

[1] Боков Г. В. Итеративные пропозициональные исчисления // Интеллектуальные системы. Теория и приложения. — 2014. Т. 18, вып. 4. — С. 99–106.

[2] Кудрявцев В. Б. Функциональные системы. — М.: Изд-во Моск. ун-та, 1982.

[3] Кузнецов А. В. Аналоги «штриха Шеффера» в конструктивной логике // ДАН СССР. — 1965. Т. 160, № 2. — С. 274–277.

[4] Кузнецов А. В. О функциональной выразимости в суперинтуиционистских логиках // Матем. исследования. — 1971. Т. 6, № 4. — С. 75–122.

[5] Кузнецов А. В. О средствах для обнаружения невыводимости или невыразимости // Логический вывод. — М.: Наука, 1979. — С. 5–33.

[6] Мальцев А. И. Итеративные алгебры и многообразия Поста // Алгебра и логика. — 1966. Т. 5, № 2. — С. 5–24.

[7] Citkin A. A mind of a non-countable set of ideas // Logic and Logical Philosophy. — 2008. Vol. 17. — P. 23–39.

[8] Gladstone M. D. The decidability of one-variable propositional calculi // Notre Dame Journal of Formal Logic. — 1979. Vol. 20, no. 2. — P. 438–450.

[9] Hughes C. E. Two Variable Implicational Calculi of Prescribed Many-One Degrees of Unsolvability // Journal of Symbolic Logic. — 1976. Vol. 41, no. 1. — P. 39–44.

[10] Hughes C. E., Singletary W. E. Triadic partial implicational propositional calculi // Zeitschrift für mathematische Logik und Grundlagen der Mathematik. — 1975. Vol. 21. — P. 21–28.

[11] Linial S., Post E. L. Recursive unsolvability of the deducibility, Tarski's comleteness, and independence of axioms problems of the propositional calculus // Bulletin of the American Mathematical Society. — 1949. Vol. 55. — P. 50.

[12] Post E. L. Formal reduction of the general combinatorial decision problem // Am. J. of Mathematics. — 1943. Vol. 65. — P. 197–215.

[13] Sinaceur H. Address at the Princeton University bicentennial conference on problems of mathematics (December 17–19, 1946), by Alfred Tarski // Bulletin of Symbolic Logic. — 2000. Vol. 6, no. 1. — P. 1–44.