Моисей Шейнфинкель и Комбинаторный язык.

С. Тропанец

Аннотация. В настоящее время имя Моисея Шейнфинкеля стало широко известным благодаря новым приложениям его фундаментальных идей по упрощению формального языка в функциональном программировании, хотя сами эти идеи были получены в совершенно другом контексте, именно, в контексте программы Гильберта. Шейнфинкель был одним из самых активных членов знаменитой команды Гильберта, пытавшейся обосновать классическую математику в 20-х годах XX столетия. Статья состоит из двух частей. В первой части изложена биография Шейнфинкеля. Во второй части излагаются основные идеи Шейнфинкеля, а также доказывается дополнение Бемана об однозначности расшифровки формул комбинаторного языка, обоснование которого автор не нашел в литературе.

1. Моисей Шейнфинкель

Моисей Ильич (или Исаевич) Шейнфинкель (4 сентября 1889-го года, Днепропетровск (или Одесса) -- 1942 год, Москва) – выпускник Одесского университета, который в то время назывался Новороссийским. Сегодня практически любой логик и многие программисты знают эту фамилию и замечательные идеи с которыми она связана. Наибольшее влияние на Шейнфинкеля, как и на многих других видных одесских математиков, оказал С. И. Шатуновский.

М. Шейнфинкель. Одесский период.

БИЛЕТЪ
для постояннаго ношенія при себѣ и предъявленія при входѣ
въ зданіе Университета.

№ 258.
Шейнфинкель
Моисей
Эльевичъ
Осень 1909/10 уч. года.
Проректоръ
Секретарь по студ. дѣламъ

СТУДЕНТЪ
ИМПЕРАТОРСКАГО
Новороссійскаго Университета
Физико-Математическаго
ФАКУЛЬТЕТА
Матем. отд.
Весна 1909/10 учеб. года.
Секретарь по студ. дѣламъ

Фото студенческого билета.

После окончания университета, Шейнфинкель уезжает на стажировку в Германию, в Геттингенский университет, где вступает в логико-математическую школу Гильберта. Представители этой школы (среди которых были Бернайс, фон Нейман, Сколем, Аккерман, Эрбран, Генцен, Беман и др.) назывались формалистами. Перед членами группы стояли три фундаментальные проблемы, сформулированные Гильбертом, решение которых означало бы выход из кризиса математики конца XIX – начала XX века: аксиоматизация математики (доказательство полноты предложенных к тому времени аксиоматических систем для множеств и чисел), доказательство непротиворечивости этой аксиоматизации, поиск единого способа, позволяющего решить любую задачу о множествах, а следовательно и о любых математических объектах, в положительную или отрицательную сторону. Последняя проблема получила название проблемы разрешимости для логики первого порядка. Существование такого разрешающего способа вытекает уже из положительного решения первой проблемы. Но, поскольку, первая проблема представлялась чрезвычайно трудной, а вера формалистов, во всяком случае, вера Гильберта, в ее положительное решение была очень сильной, формалисты искали этот способ и назависимо от нее.

Во время своего десятилетнего пребывания в Геттингене с 1914 по 1924 годы, Шейнфинкель атакует указанные проблемы и получает замечательные результаты. Он создает новый упрощенный язык первого порядка, в котором нет связанных переменных, но есть пять базовых комбинаторов: I, C, T, Z, S и U, дает, совместно с Бернайсом, разрешающий алгоритм для проверки на универсальную истинность формул первого порядка, которые в нормальной форме имеют либо только кванторы всеобщности, либо только кванторы существования, либо в которых все кванторы всеобщности предшествуют всем кванторам существования, аксиоматизирует импликативные тавтологии[1]

Шейнфинкель в Германии.

1 Аксиоматизация Шейнфинкеля импликативных тавтологий стала известной благодаря Бернайсу.

В итоге этих исследований Шейнфинкель публикует две статьи. В статье "О структурных блоках математической логики" ([14]), которая была опубликована в Mathematische Annalen (одном из ведущих математических журналов Германии и мира того времени) в 1924 году благодаря редакционной помощи Бемана, впервые излагается комбинаторная логика и связанные с ней результаты. Именно эта статья впоследствии вдохновит его заокеанского коллегу Хаскела Карри, который придаст его идеям завершенный вид, и именно эту статью затем, как одну из самых важных из опубликованных когда-либо статей по математической логике, в 1967 году отберет первый историк математической логики, французский математик Жан ван Хейенорт[2] в свою, ставшую настольной, книгу по истории математической логики "От Фреге к Геделю: книга источников по математической логике" ([10]). Комбинаторная логика позже станет, на ряду с лямбда исчислением, основанием функционального программирования. Идеи комбинаторной логики найдут применение также в теории категорий.

Разрешающий алгоритм Шейнфинкель публикует совместно с Бернайсом в статье 1929 года "К проблеме разрешимости в математической логике"все в том же Mathematische Annalen([2]) и, опять-таки, работу по оформлению результата полностью выполняет его коллега. Примерно в это же время,независимо от Бернайса и Шейнфинкеля, Фрэнк Пламптон Рамсей[3] докажет этот же результат, только другим способом, и опубликует на последнем году своей жизни в статье "Об одной проблеме формальной логики" 1930 года ([13]) (знаменитая теорема Рамсея, которая как впоследствии выяснилось, была лишь вершиной огромного айсберга интереснейших свойств графов и которая привела к известной сегодня теории Рамсея, доказывается в этой статье лишь как вспомогательная лемма). Немного позже Алонзо Черч и, независимо от него, Алан Тьюринг поставят точку этим попыткам, доказав неразрешимость проблемы разрешимости в общем случае, т. е. что не существует алгоритма, выясняющего универсальную истинность любой формулы первого порядка ([6], [7], [16]). Найденный в результате разрешимый класс универсально истинных формул сегодня известен как класс Бернайса-Шейнфинкеля или (реже) Бернайса-Шейнфинкеля-Рамсея.

После окончания стажировки, Шейнфинкель возвращается в Советскую Россию, но теперь уже не в Одессу, а в Москву. По свидетельству Александра Кузичева, «в Москве жил он очень уединённо, скромно (можно сказать, в бедности), был в стороне от столичной математической жизни, возможно, в силу чрезвычайной застенчивости. С.А. Яновская вспоминала, что он иногда приходил на семинары по логике в МГУ, но никак о себе не заявлял, ни с кем не контактировал, тихонько приходил, слушал и тихонько уходил».

После того, как Карри узнал о работе Шейнфинкеля, он пытался связаться с ним. Карри в своих заметках пишет: «Эта статья содержит большую часть из того, что я сделал». Однако было поздно, так как в 1927 году Шейнфинкеля объявляют душевно больным и помещают в санаторий. Карри продолжал работу над комбинаторной логикой уже с Бернайсом в Германии.

Шейнфинкель провел свои последние годы в подавленной войной Москве в ужасной нищете, поддерживаемый несколькими друзьями. Болезнь сразила его окончательно в 1942 году. Пытаясь согреться, его соседи впоследствии сожгут все его рукописи, большинство из которых не были опубликованы ни устно, ни письменно. В них вполне могли содержаться формализации логики, натуральных чисел или множеств на комбинаторном языке.

---

2 Жан ван Хейенорт (1912 – 1986) – французский логик, историк математической логики, деятель французского и американского троцкистского движения, личный секретать Льва Троцкого в Америке.

3 Фрэнк Пламптон Рамсей (1903 – 1930) – британский математик, философ и экономист, переводчик логико-философского трактата Витгенштейна на английский язык.

2. Комбинаторный язык

Главной целью программы Гильберта являлось доказательство непротиворечивости формальных аксиоматических систем математики финитными методами. Аксиомы логики и математики записанные на формальном языке воспринимались как конечные последовательности знаков формального языка, а правила вывода – как правила преобразования одних конечных последовательностей знаков в другие. Задача обоснования непротиворечивости, таким образом, сводилась к доказательству того, что правила вывода, будучи применены к аксиомам, никогда не приведут к двум формулам вида A и $\neg A$. Финитная концепция требовала, чтобы в таком доказательстве использовались только самые простые и явные методы и понятия, такие как понятие знака, конечной последовательности знаков, перестановки знаков, конкатенации последовательностей и т. д. Например, мы рассуждаем финитно, когда доказываем, что при определенной начальной позиции шахматной игры черные не смогут избежать мат. В таких рассуждениях нам не приходится использовать бесконечные множества. Для более детального знакомства с финитной концепцией Гильберта читателю следует обратиться к соответствующей литературе (см., например, [17]), хотя такое знакомство и не является необходимым для понимания последующего материала.

В свете вышесказанного, для доказательства непротиворечивости формальных систем, важно чтобы сами эти системы были как можно более просты и состояли только из самых базовых категорий. Например, можно минимизировать количество аксиом и правил вывода, упростить аксиомы и правила вывода. Можно, более того, упростить лежащий в основе формальный язык и тогда, возможно, упростится и все остальное. Как раз упрощение формального языка, поиск самых базовых, существенных категорий формального языка и составляли цель работы Шейнфинкеля, как чрезвычайно важный этап подготовки к доказательству непротиворечивости математики. Как уже упоминалось, результаты Шейнфинкеля в этом направлении были изложены Беманом в статье "Структурные блоки математической логики". Ниже, мы подробно рассматриваем эти результаты.

Как хорошо известно, Язык Первого Порядка (ЯПП), выделенный Левенгеймом и Сколемом, имеет следующие базовые символы:

переменные	$x_1, x_2, x_3, \ldots$
предикатные буквы	$R_1^1, R_2^1, \ldots, R_i^j, \ldots$
логические символы	$\neg, \vee, \wedge, \rightarrow, \Leftrightarrow, \forall, \exists$
скобки и запятая	(,), , .

Сегодня в ЯПП для удобства включают также символы операций $f_1^1, f_2^1, \ldots, f_i^j$ ... и констант $c_1, c_2,$ ..., чего мы с той же целью делать не будем. Язык Второго Порядка (ЯВП) получается из ЯПП добавлением предикатных переменных $X_1^1, X_2^1, \ldots, X_i^j, \ldots$. Через $\neg, \vee, \forall$ можно выразить остальные пропозициональные связки и квантор $\exists$:

$F \wedge G$ эквивалентно $\neg(\neg F \vee \neg G)$

$F \rightarrow G$ эквивалентно $\neg F \vee G$

$F \Leftrightarrow G$ эквивалентно $(F \rightarrow G) \wedge (G \rightarrow F)$

$\exists x_i(F)$ эквивалентно $\neg(\forall x_i \neg(F))$

Шеффер, однако, нашел иную пропозициональную связку |, штрих Шеффера, выражающую несовместимость:

$F \mid G$ эквивалентно $\neg(F \wedge G)$, с помощью которой можно выразить остальные:

$$\neg F \text{ эквивалентно } F|F$$
$$F \vee G \text{ эквивалентно } (F|F)|(G|G)$$

Таким образом, вместо пропозициональных связок $\neg, \vee, \wedge, \rightarrow, \Leftrightarrow$ можно обойтись одним лишь штрихом Шеффера.

Первая идея Шейнфинкеля состояла в обобщении штриха Шеффера. Он вводит обобщенный штрих | следующим образом:

$$F|^{x_i}G \text{ эквивалентно } \forall x_i \neg(F \wedge G)$$

Тогда обобщенный штрих заменяет обычный штрих и квантор всеобщности, являясь, таким образом, единственным логическим знаком формального языка:

$$\neg F \text{ эквивалентно } F|^{x_i}F$$
$$F \vee G \text{ эквивалентно } (F|^{x_i}F)|^{x_j}(G|^{x_i}G)$$
$$\forall x_i(F) \text{ эквивалентно } (F|^{x_j}F)|^{x_i}(G|^{x_j}G)$$

Дальнейшее нововведение Шейнфинкеля состояло в элиминировании многоместных предикатных букв. Каждый предикат $R(x, y)$ можно рассматривать как предикатную функцию $R_f(x, y)$, определенную так:

$$R_f(x, y)=1, \textit{если}\, R(x, y)$$
$$R_f(x, y)=0, \textit{если не}\, R(x, y)$$

И так далее для предикатов большей местности. Более того, каждую многоместную функцию можно заменить на одноместную, если позволить всем функциям иметь в качестве аргументов другие функции и даже самих себя. Именно, функцию $f(x, y): M * M \rightarrow A$ заменяем на функцию $f'(x): M \rightarrow (M \rightarrow A)$ такую, что:

$$(f'(x))(y)=f(x, y) .$$

Очевидно, что везде где использовалась f теперь, с несущественными изменениями, можно использовать f'. Аналогично, $f(x, \ldots, z): M^n \rightarrow A$ можно заменить на $f'(x): M \rightarrow (M \rightarrow \ldots \rightarrow (M \rightarrow A) \ldots)$ такую, что $(\ldots((f'(x))(y))\ldots)(z)=f(x, \ldots, z)$. Это сведение получило название "каррирования", хотя сам Карри не раз подчеркивал авторство Шейнфинкеля. (Впрочем, применение такого сведения можно найти уже у Фреге). Таким образом, двухместный предикат $R(x, y)$ в новом языке можно выразить через предикатную функцию $(R_1^2(x_1))(x_2)$, трехместный $R(x, y, z)$ через $((R_1^3(x_1))(x_2))(x_3)$, n-местный предикат $R(x, \ldots, z)$ через

$$(\ldots((R_1^n(x_1))(x_2))\ldots(x_{n-1}))(x_n) .$$

Легко переопределить формулы нового ЯВП с одноместными символами предикатных функций.

Далее, Шейнфинкель вводит комбинатор U:

$$(U(F))(G) \text{ эквивалентно } F(x_i)|^{x_i}G(x_i) ,$$

Здесь $F(x_i)$ и $G(x_i)$ – выражения вида $((((R_j^{n+1}(x_{i_1}))(x_{i_2}))...)(x_{i_n}))(x_i)$, где R – $n+1$ – местный символ предикатной функции. Т. е., последняя переменная x_i в обоих выражениях элиминируется за счет введения комбинатора U. Одной только связки U не хватит, чтобы записать все формулы ЯВП (и элиминировать, таким образом, связанные переменные). С этой целью, Шейнфинкель вводит знаменитые комбинаторы I, C, T, Z и S, которых, вместе с U, достаточно для перевода любой формулы ЯВП с элиминацией связанных переменных. При этом определение комбинатора U обобщается так, что F, G и x_i могут быть произвольными корректными выражения составлеными из переменных и комбинаторов. Каждый комбинатор, выполняя свою уникальную роль, "помогает" в процессе элиминирования связанных переменных.

2.1. Комбинаторы.

1) Комбинатор I по определению не изменяет своего аргумента:

$$I(x)=x \text{ для любого } x.$$

В частности, $I(I)=I$. Комбинатор I позволяет представить любое выражение F как значение некоторой функции, именно, функции I :

$$F=I(F) .$$

2) Комбинатор C по константе возвращает постоянную функцию равную этой константе:

$$(C(a))(x)=a \text{ для любого } x.$$

Если некоторое выражение не зависит от x, то с помощью C это выражение можно преобразовать в костантную функцию от аргумента x. Это дает возможность применить U так как его применение требует чтобы оба аргумента заканчивались общей переменной связанной обобщенным штрихом. Например, пусть F и G не содержат x, тогда формулу $F|^x G$ можно представить как $(C(F))(x)|^x(C(F))(x)$, а затем уже можно ввести U: $(U(C(F)))(C(F))$.

3) Комбинатор T изменяет порядок применения функций:

$$((T(f))(x))(y)=(f(y))(x)$$

Он помогает вынести нужную переменную наружу обоих выражений чтобы затем можно было бы ввести U.

4) Комбинатор Z переставляет скобки:

$$((Z(f))(g))(x)=f(g(x))$$

Играет ту же роль, что и T.

5) Комбинатор S:

$$((S(f))(g))(x)=(f(x))(g(x)) .$$

Кроме того, что, как T и Z, S позволяет выносить переменные наружу, он вместе с тем сокращает количество вхождений переменных, которые выносит.

Шейнфинкель, вводя комбинаторы, понимает их как функции нового типа, обобщающие обычные функции в смысле анализа или теории множеств и имеющие самих себя в качестве аргументов. Равенства в определениях комбинаторов он сначала понимает не как синтаксические правила перевода с одного языка на другой, а как свойства этих новых объектов. Затем, однако, без специальных оговорок, он работает с указанными равенствами как с правилами перевода и сокращения в синтаксическом смысле. (В силу этого, он не доказывает возможность таких обобщенных функций – этим вопросом занимался в последствии Дана Скотт (см., например, [15])).

Проиллюстрируем перевод следующим примером самого Шейнфинкеля ([14]). Переведем формулу второго порядка $\forall X_1^1(\neg\forall X_2^1(\neg\forall x_1\neg(X_1^1(x_1)\wedge X_2^1(x_1))))$ на комбинаторный язык:

$$\forall X_1^1\neg\forall X_2^1\neg\forall x_1\neg(X_1^1(x_1)\wedge X_2^1(x_1)) > \forall X_1^1\neg\forall X_2^1\neg(X_1^1(x_1)|^{x_1}X_2^1(x_1)) >$$

$$\forall X_1^1\neg\forall X_2^1\neg((X_1^1(x_1)|^{x_1}X_2^1(x_1))\wedge(X_1^1(x_1)|^{x_1}X_2^1(x_1))) >$$

$$\forall X_1^1\neg((X_1^1(x_1)|^{x_1}X_2^1(x_1))|^{X_2^1}(X_1^1(x_1)|^{x_1}X_2^1(x_1))) >$$

$$\forall X_1^1\neg(((X_1^1(x_1)|^{x_1}X_2^1(x_1))|^{X_2^1}(X_1^1(x_1)|^{x_1}X_2^1(x_1)))\wedge((X_1^1(x_1)|^{x_1}X_2^1(x_1))|^{X_2^1}(X_1^1(x_1)|^{x_1}X_2^1(x_1)))) >$$

$$((X_1^1(x_1)|^{x_1}X_2^1(x_1))|^{X_2^1}(X_1^1(x_1)|^{x_1}X_2^1(x_1)))|^{X_1^1}((X_1^1(x_1)|^{x_1}X_2^1(x_1))|^{X_2^1}(X_1^1(x_1)|^{x_1}X_2^1(x_1))) >$$

(после того, как формула переписана на язык одного только обобщенного штриха Шеффера, вводим комбинаторы)

$$(((U(X_1^1))(X_2^1)|^{X_2^1}(U(X_1^1))(X_2^1))|^{X_1^1}((U(X_1^1))(X_2^1)|^{X_2^1}(U(X_1^1))(X_2^1))) >$$

$$(U(U(X_1^1)))(U(X_1^1))|^{X_1^1}(U(U(X_1^1)))(U(X_1^1)) >$$

$$(((Z(U))(U))(X_1^1))(U(X_1^1))|^{X_1^1}(((Z(U))(U))(X_1^1))(U(X_1^1)) >$$

$$((S(((Z(U))(U))))(U))(X_1^1)|^{X_1^1}((S(((Z(U))(U))))(U))(X_1^1) >$$

$$(U((S(((Z(U))(U))))(U)))((S(((Z(U))(U))))(U))$$

Последняя формула и является переводом на язык комбинаторов.

2.2. Алгоритм перевода

Шейнфинкель в своей статье ограничивается вышеприведенным примером и не формулирует точный алгоритм перевода, считая задачу уточнения несложной. Можно использовать, например, нижеследующий корректный алгоритм перевода. Перед тем как приводить алгоритм перевода, построим сначала алгоритм *П*, который выносит переменную x_i из формулы F, составленной из комбинаторов, переменных и предикатных символов, т. е. преобразует F в формулу вида $D(x_i)$, где выражение D уже не содержит вхождений переменной x_i, при условии, что x_i входит в F (причем x_i может означать как

переменную так и предикатную переменную).
Алгоритм *П*:

Шаг 1. Фиксируем самое левое вхождение переменной x_i в F .
Шаг 2. Применять следующие преобразования 1) - 4) к вхождению x_i выбранному на предыдущем шаге пока они применимы (после каждого применения любого из преобразований 1) - 4) указанное вхождение переменной x_i передвигается вправо; на каждом шаге лишь одно из преобразований 1) –4) применимо):

1) если вхождение переменной x_i в F расположено так, что F можно представить как $...((D(x_i))(E))...$ (E и D – некоторые выражения), то F можно преобразовать к $...(((S(F))(C(G)))(x_i))...$:

$$...((D(x_i))(E))... > ...((D(x_i))((C(E))(x_i)))... > ...(((S(D))(C(E)))(x_i))...$$

2) если вхождение переменной x_i в F расположено так, что F можно представить как $...(D(E(x_i)))...$ (E и D – некоторые выражения), то F можно преобразовать к $...(((Z(D))(E))(x_i))...$:

$$...(D(E(x_i)))... > ...(((Z(D))(E))(x_i))...$$

3) если вхождение переменной x_i в F расположено так, что F можно представить как $...(D(x_i(E)))...$ (E и D – некоторые выражения), то F можно преобразовать к $...(((T(Z(D)))(E))(x_i))...$:

$$...(D(x_i(E)))... > ...(((Z(D))(x_i))(E))... > ...(((Z(D))(x_i))(E))... >$$
$$...(((T(Z(D)))(E))(x_i))...$$

4) если вхождение переменной x_i в F расположено так, что F можно представить как $...((x_i(D))(E))...$ (E и D -- некоторые выражения), то F можно преобразовать к $...(((T(I))(E))(x_i(D)))...$:

$$...((x_i(D))(E))... > ...((I(x_i(D)))(E))... > ...(((T(I))(E))(x_i(D)))... ,$$

и затем уже применить к последнему выражению преобразование 3).

Через конечное число применений преобразований 1) - 4) мы придем к выражению вида $D(x_i)$, т. е. вынесем самое левое вхождение x_i вперед. Если в D еще остались вхождения переменной x_i , то с помощью аналогичной процедуры можно вынести из D самое левое из таких вхождений и получить $(D_1(x_i))(x_i)$. В конце концов мы вынесем все вхождения x_i и получим

$$(...((D_k(x_i))(x_i))...)(x_i) .$$

Шаг 3. "Сокращаем" вхождения x_i оставив только одно из них впереди:

$$(...(((D_k(x_i))(x_i))(x_i))...)(x_i) > (...(((D_k(x_i))(I(x_i)))(x_i))...)(x_i) >$$

$$(...((((S(D_k))(I))(x_i))(x_i))...)(x_i) > (...((((S(D_k))(I))(x_i))(I(x_i)))...)(x_i) >$$

$(\ldots(((S((S(D_k))(I)))(I))(x_i))\ldots)(x_i) > \ldots$

в конце этого процесса мы придем к формуле вида $D(x_i)$, где D уже не содержит вхождений x_i, что и требовалось.

Теперь, опишем алгоритм перевода. Алгоритм элиминирует связанные переменные из подформул вида $F|^{x_i}G$, предварительно вынося x_i в F и G с помощью алгоритма *П*.
Пусть A – произвольная формула записанная с помощью обощенного штриха Шеффера.

Шаг 1. Находим первую слева подформулу вида $F|^{x_i}G$, где F и G уже не содержат символ |, т. е. составлены только из переменных, символов предикатов, предикатных переменных и комбинаторов. Если x_i входит в F и G, то переходим к шагу 2. Если x_i не входит ни в F ни в G, то переходим к шагу 3. Если x_i входит только в одно из выражений F или G, то переходим к шагу 4.

Шаг 2. Элиминируем | и переменную x_i в подформуле $F|^{x_i}G$, найденной на шаге 1, следующим образом. Из F и G выносим переменную x_i применяя к ним алгоритм *П*. Таким образом, получим соответственно $D_1(x_i)$ и $D_2(x_i)$. Подформулу $F|^{x_i}G$ в формуле заменяем на $D_1(x_i)|^{x_i}D_2(x_i)$ и затем уже можно ввести комбинатор U, заменив $D_1(x_i)|^{x_i}D_2(x_i)$ на $(U(D_1))(D_2)$.

Шаг 3. Элиминируем | и переменную x_i в подформуле $F|^{x_i}G$, найденной на шаге 1, с помощью комбинатора C :

$$F|^{x_i}G > (C(F))(x_i)|^{x_i}(C(G))(x_i) > (U(C(F)))(C(G)) \;.$$

Шаг 4. Элиминируем | и переменную x_i в подформуле $F|^{x_i}G$, найденной на предыдущем шаге, следующим образом. Для определенности, предположим, что x_i входит в F. Сначала применим алгоритм *П* к F. В результате получим $D(x_i)$, где x_i не входит в D. Далее преобразовываем формулу так:

$$F|^{x_i}G > D(x_i)|^{x_i}(C(G))(x_i) > (U(D))(C(G)) \;.$$

Шаг 5. Связанная переменная x_i и обощенный штрих элиминированы из подформулы $F|^{x_i}G$. Если в формуле все еще присутствуют связанные переменными, то продолжаем элиминацию возвращаясь к шагу 1.

Но и на этом Шейнфинкель не завершает упрощения и сводит все комбинаторы к трем из них: U, C и S. Рассуждая интуитивно, т. е. интерпретируя комбинаторы как обощенные функции, он проводит следующие редукции:

$$I(x)=x=(C(x))(C(x))=((S(C))(C))(x) \text{, т. е. } I=(S(C))(C)$$

$$((Z(f))(g))(x)=f(g(x))=((C(f))(x))(g(x))=((S(C(f)))(g))(x)=((((C(S))(f))(C(f)))(g))(x)$$
$$=((((S(C(S)))(C))(f))(g))(x) \text{, т. е. } Z=(S(C(S)))(C)$$

$$((T(f))(y))(x)=(f(x))(y)=(f(x))((C(y))(x))=((S(f))(C(y)))(x) =$$
$$(((Z(S(f)))(C))(y))(x)=(((((Z(Z))(S))(f))(C))(y))(x)=(((((Z(Z))(S))(f))((C(C))(f)))(y))(x)$$
$$=((((S((Z(Z))(S)))(C(C)))(f))(y))(x) \text{, т. е. } T=(S((Z(Z))(S)))(C(C)) \;.$$

Таким образом, весь ЯПП сведен к всего лишь трем символам *U*, *C*, *S* и символам предикатов (предложения ЯВП можно выразить используя только *U*, *C* и *S*).

В дальнейшем, Карри продолжал развитие идей Шейнфинкеля, пытаясь формализовать логику и множества на комбинаторном языке. Вместо логического комбинатора *U* Карри использовал комбинатор импликации ([9]). Такая формализация однако наткнулась на значительные трудности, на что указывали парадоксы Клини-Россера ([11]) и Карри ([8]), означающие противоречивость первоначальных систем Карри и Черча. Системы удалось спасти посредством типизации, что, хоть и усложнило конструкцию, дало возможность формализовать интуиционистский ($\wedge, \forall$) – фрагмент предикатной логики первого порядка. Лишь недавно были построены непротиворечивые комбинаторные системы, в которых можно формализовать всю классическую логику первого порядка. Формализация натуральных чисел и множеств является предметом статей Бандера ([3], [4]).

Под влиянием работы Шейнфинкеля, Бернайс и Куайн построили другие варианты исчисления предикатов без переменных, используя комбинаторы сходные с теми, которые ввел Шейнфинкель, однако без возможности аппликации комбинаторов друг к другу (см., например, статью Куайна [12], а также [10], стр. 357). Работа Куайна получила применения в области автоматического вывода.

За дальнейшими подробностями истории развития комбинаторной логики вплоть до современности отсылаем читателя к статье [5].

2.3. Однозначность расшифровки

Хейнрих Беман (Behmann), редактор статьи Шейнфинкеля, в конце статьи сделал три добавления. В первом из них, Беман замечает, что из формулы комбинаторного языка всегда можно вынести все вхождения комбинатора *U* наружу и, затем, сократить, оставив только одно такое вхождение. Это можно сделать, например, аналогично тому, как выносились и сокращались вхождения переменных в алгоритме перевода. Таким образом, каждое предложение ЯВП можно перевести в формулу комбинаторного языка вида *H(U)*, где выражение *H* не содержит свободных переменных и составлено только из *C* и *S*. Если рассматривать только переводы такого вида и опускать единственное вхождение комбинатора *U* в конце, то, таким образом, любое предложение ЯВП можно записать только с помощью *C* и *S*.

Второе замечание касалось элиминации скобок и, как впоследствии обнаружил Боскович, оказалось ошибочным ([9], стр. 184).

Шейнфинкель опускает доказательство того, что при любой дешифровке формулы комбинаторного языка на ЯПП или ЯВП получаются, по крайней мере, эквивалентные формулы. Беман, в своем третьем замечании, утверждает без доказательства, что хотя формуле ЯВП могут соответствовать несколько различных переводов на комбинаторный язык, каждой формуле комбинаторного языка соответствует точно одна формула ЯВП с обобщенным штрихом Шеффера с точностью до переименования связанных переменных. Автор данной статьи не нашел в литературе обоснований этого факта и далее предлагает такое обоснование.

Сначала, дадим строгое определение перевода. Общим языком назовем ЯВП в который добавили комбинаторы *C*, *S* и *U*. Формулой такого языка будет, например, такое выражение: $S((C(x_1))(x_2))|^{x_3}(U(X_1^1))(x_1)$. Нетрудно дать точное определение формулы общего языка. Пусть A_1 и A_2 - формулы общего языка. Говорят, что A_1 редуцируется к A_2 , если A_2 может быть получена из A_1 посредством замены подвыражения вида $(C(D))(E)$ на *D* или подвыражения вида $((S(D))(E))(H)$ на $(D(H))(E(H))$ или подвыражения вида $(U(D))(E)$ на $D(x_i)|^{x_i}E(x_i)$. Т. е., редукция соответствует элиминированию одного из комбинаторов в соответствии с его определением. Для удобства, будем считать каждую формулу редуцирующейся к самой себе. Тот факт, что A_1 редуцируется к A_2 обозначают посредством записи: $A_1 > A_2$, которую называют редукцией. Будем говорить, что A_1 сводится к A_2, если существует последовательность редукций $A_1 = B_1 > \ldots > B_n = A_2$, в которой все связанные переменные, вводимые при элиминировании вхождений комбинатора *U*, отличны от переменных, которые входят в A_1, и никакая из вводимых переменных не вводится более одного раза. Такую последовательность назовем выводом. Переводом формулы *A* общего языка на ЯВП назовем формулу *B* ЯВП для которой существует вывод $A_1 = A > A_2 > \ldots > A_n = B$. Не все формулы общего языка имеют перевод на ЯВП (например, $S(U(x_1))$).

Покажем теперь, что если формула *A* общего языка имеет хотя бы один перевод, то все ее переводы отличаются друг от друга переименованием связанных переменных. Пусть B_1 и B_2 – переводы *A*, соответствующие следующим выводам:

$$A > C_1 > \ldots > C_l = B_1$$

$$A > D_1 > \ldots > D_s = B_2$$

Покажем, что для любых C_i и D_j ($i < l, j < s$) существуют формулы D' и D'' общего языка к которым редуцируются соответственно C_i и D_j и которые отличаются друг от друга только наименованием связанных переменных. Отсюда будет следовать требуемое утверждение, поскольку при $i = l - 1$ и $j = s - 1$ будем иметь $C_{l-1} > D'$ и $D_{s-1} > D$ (C_{l-1} и D_{s-1} сводятся к D' и D'' за один шаг, так как содержат по одному вхождению комбинатора), т. е. D' совпадает с B_1 и B_2 с точностью до переименования связанных переменных. Нам будет удобнее доказать индукцией по *n* следующую переформулировку: для любого $n \leq max(l, s)$ и для любых $i \leq n$ и $j \leq n$, из того что $i < l$ и $j < s$ следует, что для C_i и C_j существуют D' и D'' с указанными выше свойствами. Пусть $n = 1$, т. е. $i = j = 1$. Если C_1 совпадает с D_1, то утверждение очевидно. Пусть C_1 отлично от D_1. Тогда в редукциях $A > C_1$ и $A > D_1$ элиминировались разные вхождения комбинаторов. Пусть, например, C_1 получена элиминированием вхождения комбинатора *C*, а D_1 – вхождением комбинатора *S*:

$$A = \ldots((C(F))(G))\ldots > \ldots(F)\ldots \ = \ C_1$$

$$A = \ldots(((S(E))(R))(H))\ldots > \ldots((E(H))(R(H)))\ldots \ = \ D_1$$

Могут представиться следующие 3 случая:

1) Подвыражение $(((S(E))(R))(H))$ входит либо в *F* либо в *G*;
2) Подвыражение $((C(F))(G))$ входит либо в *E* либо в *R* либо в *H*;
3) Подвыражения $((C(F))(G))$ и $(((S(E))(R))(H))$ не пересекаются, и, в этом случае, *A* можно представить в виде $\ldots((C(F))(G))\ldots(((S(E))(R))(H))\ldots$ или $\ldots(((S(E))(R))(H))\ldots((C(F))(G))\ldots$.

В первом случае, предположив, например, что $(((S(E))(R))(H))$ входит в *F*, можно построить следующие выводы:

$$A = \ldots((C(F))(G))\ldots = \ldots(C(\ldots(((S(E))(R))(H))\ldots)(G))\ldots \ > \\ \ldots(\ldots(((S(E))(R))(H))\ldots)\ldots \ = \ C_1 \ > \ \ldots(\ldots((E(H))(R(H)))\ldots)\ldots = D'$$

$$A = \ldots((C(F))(G))\ldots = \ldots(C(\ldots(((S(E))(R))(H))\ldots)(G))\ldots \ > \\ \ldots(C(\ldots((E(H))(R(H)))\ldots)(G))\ldots > \ldots(\ldots((E(H))(R(H)))\ldots)\ldots = D''\ ,$$

и утверждение доказано. Аналогично рассматривается случай, когда $(((S(E))(R))(H))$ входит в G. Пусть теперь имеет место случай 2) и $((C(F))(G))$ входит, например, в H. Тогда можно построить следующие выводы:

$$A = \ldots(((S(E))(R))(\ldots((C(F))(G))\ldots))\ldots > \ldots(((S(E))(R))(\ldots(F)\ldots))\ldots \; = \; C_1 \; >$$
$$\ldots((E(\ldots(F)\ldots))(R(\ldots(F)\ldots)))\ldots = D'$$

$$A = \ldots(((S(E))(R))(\ldots((C(F))(G))\ldots))\ldots > \ldots((E(\ldots((C(F))(G))\ldots))(R(\ldots((C(F))(G))\ldots)))\ldots \; =$$
$$D_1 \; >$$
$$\ldots((E(\ldots(F)\ldots))(R(\ldots((C(F))(G))\ldots)))\ldots > \ldots((E(\ldots(F)\ldots))(R(\ldots(F)\ldots)))\ldots \; = D''$$

Аналогично рассматриваются случаи, когда $((C(F))(G))$ входит в E или R.

Случай 3) еще более прост и рассматривается аналогично. Таким образом, в случае, когда C_1 получена элиминированием вхождения комбинатора C, а D_1 – вхождением комбинатора S, база индукции доказана. Докажем базу индукции для случая, когда обе C_1 и D_1 получены элиминацией вхождений (разных) комбинатора U:

$$A = \ldots((U(F))(G))\ldots > \ldots(F(x_i)|^{x_i} G(x_i))\ldots = C_1$$

$$A = \ldots((U(E))(R))\ldots > \ldots(E(x_j)|^{x_j} R(x_j))\ldots = D_1$$

Рассуждения будут аналогичными предыдущему случаю, за исключением необходимости в контроле связанных переменных. Могут представиться следующие 3 случая:

1) Подвыражение $((U(E))(R))$ входит либо в F либо в G;
2) Подвыражение $((U(F))(G))$ входит либо в E либо в R;
3) Подвыражения $((U(F))(G))$ и $((U(E))(R))$ не пересекаются, и, в этом случае, A можно представить в виде $\ldots((U(F))(G))\ldots((U(E))(R))\ldots$ или $\ldots((U(E))(R))\ldots((U(F))(G))\ldots$.

В первом случае, предположив, например, что $((U(E))(R))$ входит в F, можно построить следующие выводы:

$$A = \ldots((U(F))(G))\ldots = \ldots((U(\ldots((U(E))(R))\ldots))(G))\ldots \; >$$
$$\ldots((\ldots((U(E))(R))\ldots)(x_{i_1})|^{x_{i_1}} G(x_{i_1}))\ldots = C_1 \; > \; \ldots((\ldots(E(x_{i_2})|^{x_{i_2}} R(x_{i_2}))\ldots)(x_{i_1})|^{x_{i_1}} G(x_{i_1}))\ldots = D'$$

$$A = \ldots((U(F))(G))\ldots = \ldots((U(\ldots((U(E))(R))\ldots))(G))\ldots \; >$$
$$\ldots((U(\ldots(E(x_{j_1})|^{x_{j_1}} R(x_{j_1}))\ldots))(G))\ldots = D_1 > \ldots((\ldots(E(x_{j_1})|^{x_{j_1}} R(x_{j_1}))\ldots)(x_{j_2})|^{x_{j_2}} G(x_{j_2}))\ldots = D''$$

и утверждение доказано. Аналогично рассматривается случай, когда $((U(E))(R))$ входит в G. Случай 2) рассматривается симметрично. В случае 3) можно построить следующие выводы:

$$A = \ldots((U(F))(G))\ldots((U(E))(R))\ldots > \ldots(F(x_{i_1})|^{x_{i_1}} G(x_{i_1}))\ldots((U(E))(R))\ldots = C_1 \; >$$
$$\ldots(F(x_{i_1})|^{x_{i_1}} G(x_{i_1}))\ldots(E(x_{i_2})|^{x_{i_2}} R(x_{i_2}))\ldots = D'$$

$$A = \ldots((U(F))(G))\ldots((U(E))(R))\ldots > \ldots((U(F))(G))\ldots(E(x_{j_1})|^{x_{j_1}} R(x_{j_1}))\ldots = C_1 \; >$$
$$\ldots(F(x_{j_2})|^{x_{j_2}} G(x_{j_2}))\ldots(E(x_{j_1})|^{x_{j_1}} R(x_{j_1}))\ldots = D'',$$

и база снова доказана.

Аналогично рассматриваются случаи, когда C_1 и D_1 получены элиминацией соответственно вхождений (разных) комбинаторов *C* и *C*, *C* и *U*, *S* и *S*, или *S* и *U*.
Предположим теперь, что наше утверждение справедливо для любых *i* и *j* меньших $n \leq max(l,s)$. Покажем, что тогда утверждение справедливо также и для любых *i* и *j* меньших либо равных $n \leq max(l,s)$. Рассмотрим

$$A > C_1 > \ldots > C_i$$

$$A > D_1 > \ldots > D_j\ ,$$

где $i,j \leq n$. Если $i,j < n$, то утверждение верно в силу предположения индукции. Предположим, одно из *i* и *j* равно *n*, а другое меньше *n*. Пусть, например, *i* = *n*, *j* < *n*. По предположению индукции, существуют такие *D'* и *D''*, отличающиеся наименованием связанных переменных, что C_{i-1} сводится к *D'*, а D_j сводится к *D''*:

$$C_{n-1} > W_1 > \ldots > W_p = D'$$

$$D_j > Q_1 > \ldots > Q_t = D''\ .$$

C_n получена из C_{n-1} элиминированием одного вхождения комбинатора. Как мы покажем ниже, это не мешает провести с C_n редукции аналогичные тем, что вели от C_{n-1} к *D'*. Нам понадобится несколько определений. Пусть задана последовательность редукций. Определим понятия предка и потомка для вхождений комбинаторов. Пусть в заданной последовательности встречается редукция вида:

$$\ldots(((S(E))(R))(H))\ldots > \ldots((E(H))(R(H)))\ldots$$

тогда вхождение некоторого комбинатора в подвыражение *H* формулы слева назовем непосредственным предком двух соответствующих копий этого вхождения в подвыражениях *H* в формуле справа, а эти последние вхождения – непосредственными потомками вхождения в формулу слева. Аналогично, вхождение некоторого комбинатора в подвыражение *R* формулы слева назовем непосредственным предком соответствующей копии этого вхождения в подвыражении *R* в формуле справа, а это последнее вхождение – непосредственным потомком вхождения в формулу слева и т. д. Самое левое вхождение комбинатора *S* в подформулу ((*S*(*E*))(*R*))(*H*) формулы слева по определению не является предком никакого вхождения. Подобным же образом определяется понятие предка в редукциях в которых элиминируются другие комбинаторы. Таким образом, определены два бинарных отношения между вхождениями комбинаторов в последовательность редукций – "быть непосредственным предком" и "быть непосредственным потомком". Их рефлексивные и транзитивные замыкания назовем соответственно отношениями "быть предком" и "быть потомком". Например, в последовательности:

$$((S((C(C))(S)))(S))((U(X_1^1))(X_2^1)) > (((C(C))(S))((U(X_1^1))(X_2^1)))(S((U(X_1^1))(X_2^1))) \ >$$

$$(((C(C))(S))((U(X_1^1))(X_2^1)))(S(X_1^1(x_1)|^{x_1}X_2^1(x_1)))$$

самое левое вхождение комбинатора *C* во второй формуле является непосредственным потомком самого левого вхождения комбинатора *C* в первой формуле, самое левое вхождение комбинатора *C* в третьей формуле является потомком самого левого вхождения комбинатора *C* в первой формуле, самое левое вхождение комбинатора *S* в первой формуле не имеет

потомков и т. д.
Докажем сначала важный технический результат.

Лемма. Пусть имеется вывод:

$$(*) \qquad A_1 > A_2 > ... > A_e \ .$$

Если B_1 получается из A_1 элиминированием вхождения комбинатора *П* ($П \in \{C, S, U\}$), т. е. $A_1 > B_1$ – редукция, и формула A_e не содержит предков элиминируемого вхождения комбинатора *П* в A_1, которое, для удобства, назовем главным вхождением, то B_1 также можно свести к A_e, т. е. существует вывод

$$(**) \qquad B_1 > B_2 > ... > B_e \ ,$$

в котором $B_e = A_e$.

Мы покажем, как строится указанный вывод $B_1 > B_2 > ... > B_e$, в случае, когда *П* = *C*, т. е. A_1 имеет вид $...((C(F))(G))...$, а B_1 – $...(F)...$ (остальные случаи разбираются аналогично). Каждая редукция вывода (**) определяется в зависимости от соответствующей редукции вывода (*):

редукция $B_{k-1} > B_k$ (*k*>1) вывода (**) строится из редукции $A_{k-1} > A_k$ заменой в обоих частях всех подвыражений вида $((C(X))(Y))$, в которых самое левое вхождение комбинатора *C* является потомком главного вхождения *C* в A_1 , на (*X*).
(Заметим, что из определения потомка следует, что все потомки главного вхождения *C* входят в формулы вывода (*) как самые левые вхождения комбинатора *C* в подвыражении вида $((C(X))(Y))$.)

Покажем, что на каждом шаге получается корректная редукция $B_{k-1} > B_k$. Могут представиться следующие возможности:

1) Если в редукции $A_{k-1} > A_k$, A_k получается элиминацией вхождения комбинатора, не входящего ни в какое подвыражение вида $((C(X))(Y))$, в котором самое левое вхождение комбинатора *C* является потомком главного вхождения, то B_k можно получить из B_{k-1} элиминацией того же вхождения комбинатора и, таким образом, $B_{k-1} > B_k$ – корректная редукция.
2) Если в редукции $A_{k-1} > A_k$, A_k получается элиминацией вхождения комбинатора, входящего в подвыражение $((C(X))(Y))$, в котором самое левое вхождение комбинатора *C* является потомком главного вхождения, точнее, входящего в *X* так, что редукция имеет вид $...((C(X))(Y))... > ...((C(X'))(Y))...$, то редукция $B_{k-1} > B_k$ имеет вид $...(X)... > ...(X')...$, и, таким образом, является корректной.
3) Если в первой редукции $A_{k-1} > A_k$, A_k получается элиминацией вхождения комбинатора входящего в подвыражение $((C(X))(Y))$, в котором самое левое вхождение комбинатора *C* является потомком главного вхождения, точнее, входящего в *Y* так, что редукция имеет вид $...((C(X))(Y))... > ...((C(X))(Y'))...$, то редукция $B_{k-1} > B_k$ имеет вид $...(X)... > ...(X)...$, что является тривиальной редукцией и, значит, корректной.
4) Если в первой редукции $A_{k-1} > A_k$, A_k получается элиминацией вхождения комбинатора *C* из подвыражения $((C(X))(Y))$, в котором самое левое вхождение комбинатора *C* является потомком главного вхождения, так, что редукция имеет вид $...((C(X))(Y))... > ...(X)...$, то редукция $B_{k-1} > B_k$ имеет вид $...(X)... > ...(X)...$, т. е. снова корректна.

Рассмотрим теперь последнюю редукцию $B_{e-1} > B_e$ в последовательности (**), которая получена из $A_{e-1} > A_e$. Поскольку A_e не содержит потомков главного вхождения комбинатора *C*, то, по построению, B_e совпадает с A_e. Таким образом, последовательность (**) является выводом от B_1 к A_e, что и требовалось.

Вернемся теперь к основному доказательству. Предположим, что C_n получена из C_{n-1} элиминированием вхождения комбинатора *C*, так, что $C_{n-1} = ...((C(F))(G))...$, а $C_n = ...(F)...$. Продолжим последовательность

$$C_{n-1} = ...((C(F))(G))... > W_1 > ... > W_p = D'$$

последовательно элиминировав всех наследников указанного вхождения комбинатора *C* в C_{n-1}:

$$(*) \quad C_{n-1} = ...((C(F))(G))... > W_1 > ... > W_p = D' > W_{p+1} > ... > W_{p+z} = D .$$

Если мы докажем, что C_n также сводится к *D*, то шаг индукции будет доказан, поскольку *D″* можно свести к *D**, отличающейся от *D* лишь наименованием связанных переменных, аналогично тому, как *D′* сводилось к *D* в (*). Но C_n действительно сводится к *D* на основании леммы. Случаи, когда C_n получается из C_{n-1} посредством элиминации вхождения комбинаторов *S* и *U* рассматриваются аналогично.

Теперь перейдем к случаю, когда оба *i* и *j* равны *n*. По предположению индукции, существуют такие *D′* и *D″*, отличающиеся друг от друга лишь наименованием связанных переменных, что C_{n-1} сводится к *D′*, а D_{n-1} сводится к *D″*. Пусть C_n получается из C_{n-1} элиминарованием вхождения комбинатора *П* ($\Pi \in \{C, S, U\}$), а D_n из D_{n-1} – элиминированием вхождения комбинатора *Г* ($\Gamma \in \{C, S, U\}$). Тогда *D′* можно свести к некоторой *D*, элиминировав всех потомков главного вхождения комбинатора *П* в C_{n-1}, а *D″* к некоторому *D**, элиминировав теже вхождения. При этом *D* и *D** будут отличаться лишь наименованием связанных переменных. Далее, *D** можно свести к некоторой $\tilde{D}*$, элиминировав всех потомков главного вхождения комбинатора *Г* в D_{n-1}, а *D* к некоторому $\tilde{D}$, элиминировав теже вхождения. Аналогично, $\tilde{D}$ и $\tilde{D}*$ будут отличаться лишь наименованием связанных переменных. Наконец, на основании леммы, можно построить выводы от C_n к $\tilde{D}$ и от D_n к $\tilde{D}*$ что и заканчивает доказательство.

Литература

[1] H. Barendregt, M. Bunder, W. Dekkers. "Systems of illative combinatory logic complete for first-order propositional and predicate calculus". J. Symb. Logic, 58, 1993, pp. 769–888

[2] P. Bernays, M. Schonfinkel. "Zum Entscheidungsproblem der mathematischen Logik". Math. Ann. 99, 1929, pp. 342-372

[3] M. W. Bunder. "Set theory based on combinatory logic". Ph.D. Thesis, 1969

[4] M. W. Bunder. "Propositional and predicate calculus based on combinatory logic". Notre Dame J. Formal Logic, 15 (1974), pp. 25–32

[5] F. Cardone and J. R. Hindley. "History of Lambda-Calculus and Combinatory logic". In

Vol. 5 of the Handbook of the History of Logic.

[6] A. Church. "An unsolvable problem of elementary number theory", American Journal of Mathematics, 58 (1936), pp 345–363

[7] A. Church. "A note on the Entscheidungsproblem". J. Symb. Logic, 1, 1936, pp 40–41.

[8] H. B. Curry. "The inconsistency of certain formal logics". J. Symb. Logic , 7 (1942), pp. 115–117

[9] H. B. Curry, R. Feys. "Combinatory logic". 1, North-Holland, 1958

[10] J. van Heijenoort. "From Frege to Godel". Cambridge, Harvard University Press, 1967.

[11] S. C. Kleene, J. B. Rosser. "The inconsistency of certain formal logics". Ann. of Math. (2), 36 (1935), pp. 630–636

[12] W. V. Quine. "Variables Explained Away". Proceedings of the American Philosophical Society, Vol. 104, No. 3., Jun. 15, 1960, pp. 343-347.

[13] F. P. Ramsey. "On a problem in formal logic". Proc. London Math. Soc., Vol. 30, 1930, pp. 264–286

[14] M. Schonfinkel. "Uber die Bausteine de mathematischen Logik". Math. Ann., 92, 1924, pp. 305–316

[15] D. S. Scott. "Outline of a mathematical theory of computation". In Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems, pp. 169–176.

[16] A. Turing. "On computable numbers, with an application to the Entscheidungsproblem". Proc. London Math. Soc., Series 2, 42, 1937, pp 230–265.

[17] R. Zach. "Hilbert's program then and now". In Dale Jacquette, ed., Philosophy of Logic. Handbook of the Philosophy of Science, vol. 5., Elsevier, Amsterdam, 2006, pp. 411-447.