М. И. Шейнфинкель и комбинаторная логика[1]

В. И. Шалак

abstract. In the article we review the famous paper of Moses Schönfinkel and it's influence on subsequent development of combinatory logic and λ-calculus.

Ключевые слова: Шейнфинкель, комбинаторная логика, лямбда-исчисление.

Прошло более 80 лет с момента публикации на немецком языке статьи Моисея Исаевича Шейнфинкеля "*О кирпичах математической логики*"[2]. Все это время в Советском Союзе она была труднодоступной. Остается лишь сожалеть, что люди, которым мы обязаны переводом на русский язык многих зарубежных монографий по логике и основаниям математики, в свое время не обратили на нее должного внимания. Возможно, именно по этой причине тематика комбинаторной логики и λ-исчисления не пользуется у нас той же популярностью, что и в остальном мире. Публикуя перевод статьи М. Шейнфинкеля, мы хотим восполнить этот пробел.

Несмотря на широкую известность имени М. Шейнфинкеля, о самом авторе мы знаем на удивление мало – две статьи, одна сохранившаяся фотография и скудные биографические данные. Известно, что он родился 4 сентября то ли 1887, то ли 1889 года в городе Екатеринославе (ныне Днепропетровск на Украине). Изучал математику в Одессе в университете, который тогда назывался Новороссийским. Его руководителем был

[1]Работа поддержана РГНФ. Грант № 08-03-00173а.

[2]*Schönfinkel M.* Über die Bausteine der mathematischen Logik // Mathematische Annalen. 1924. Bd. 92. S. 305–316.

Самуил Осипович Шатуновский – известный российский математик, много внимания уделявший вопросам геометрии и ее оснований. Нет ничего удивительного в том, что в 1914 г. М. Шейнфинкель приезжает в математическую Мекку того времени – в Геттинген к Д. Гильберту. Известность пришла к М. Шейнфинкелю благодаря докладу, который он сделал 7 декабря 1920 г. перед Математическим обществом Геттингена. Д. Гильберт рекомендовал доклад к публикации, которая и была осуществлена четырьмя годами позже в известном математическом журнале "Mathematische Annalen". Помощь в подготовке публикации оказал Г. Беман.

В Геттингене М. Шейнфинкель тесно общался с П. Бернайсом, который был его ровесником. В 1929 г. ими была подготовлена совместная публикация[3], посвященная проблеме разрешимости для одного частного класса формул исчисления предикатов. К этому времени М. Шейнфинкель уже вернулся в СССР. Последние годы жизни он провел в Москве, где страдал от нищеты, лечился от психического расстройства и умер в 1942 г. в госпитале. Ни точная дата смерти, ни место погребения неизвестны. Во время войны, чтобы хоть как-нибудь согреться, бывшие соседи М. Шейнфинкеля сожгли все его рукописи в печи.

Это все, что мы о нем знаем.

М. Шейнфинкель поставил перед собой задачу уменьшить число исходных понятий, используемых в логике, и избавиться от связанных переменных, поскольку они не имеют самостоятельного значения, а являются всего лишь "*знаками того, что определенные аргументные места и операторы принадлежат к одному типу*".

Если посредством определения $a|b = \neg a \vee \neg b$ ввести в язык классической логики новую связку, штрих Шеффера, то окажется, что ее одной достаточно для определения всех остальных связок.

$$\neg a = a|a, \quad a \vee b = (a|a)|(b|b)$$

М. Шейнфинкель замечает, что аналогичным образом можно определить обобщенный штрих Шеффера

[3] *Bernays P., Schönfinkel M.* Zum Entscheidungsproblem der mathematischen Logik // Mathematische Annalen. Bd. 99. S. 342–72.

$$f(x)|^x g(x) = (x)\neg(f(x) \ \& \ g(x)),$$

посредством которого выразимы не только связки, но и кванторы

$$\neg a = a|^x a, \quad a \vee b = (a|^y a)|^x(b|^y b),$$
$$(x)f(x) = (f(x)|^y f(x))|^x(f(x)|^y f(x)).$$

Далее он показывает, каким образом можно свести функции от нескольких аргументов к функциям одного аргумента.

Пусть дана функция $F(x, y)$. Для различных фиксированных значений аргумента x мы будем получать различные функции $G_x(y)$ от y. Если представить различные функции G как значения некоторой новой функции f, то $G = fx$, и исходная функция $F(x, y)$ может быть представлена в виде $(fx)y$. “*Таким образом, теперь fx представляет функцию, которая при подстановке значений для x дает не объекты основной области (как было со значением $F(x, y)$), а опять приводит к функции, аргументом которой является y. Иными словами, f есть функция, аргумент которой никак не ограничен, а ее значением вновь является некоторая функция*”. Функции могут быть не только значениями других функций, но и занимать аргументные места.

Записывая функцию $(((fx)y)z)$ в виде $fxyz$, а $((f(xy))z)$ в виде $f(xy)z$, мы будем по умолчанию предполагать левую ассоциацию скобок при их восстановлении.

М. Шейнфинкель вводит еще пять специальных функций, которые позднее получили название комбинаторов:

$\mathbf{I}x = x$ — функция тождества
$\mathbf{C}xy = x$ — константная функция
$\mathbf{T}xyz = xzy$ — функция перестановки
$\mathbf{Z}xyz = x(yz)$ — функция группировки
$\mathbf{S}xyz = xz(yz)$ — функция слияния

Они нужны для того, чтобы определять другие функции и представлять их свойства. Например, коммутативность сложения $x+y$, которое будем записывать как Axy, может быть представлена в виде $Axy = Ayx = \mathbf{T}Axy$, а после опускания x и y просто как $A = \mathbf{T}A$.

Эти функции не являются независимыми. Достаточно оставить лишь $\mathbf{C}$ и $\mathbf{S}$, так как остальные выразимы через них следующим образом:

$\mathbf{I} = \mathbf{SCC}$
$\mathbf{Z} = \mathbf{S(CS)C}$
$\mathbf{T} = \mathbf{S(ZZS)(CC)}$

Вместо обобщенного штриха Шеффера М. Шейнфинкель вводит функцию несовместимости $\mathbf{U}fg = fx|^x gx$. В левой части определения связанные переменные уже не фигурируют. С помощью $\mathbf{U}$ и функций $\mathbf{C}$ и $\mathbf{S}$ выразимы все логические формы. Например, формула логики предикатов второго порядка $(f)(Eg)(x)\neg(fx \ \& \ gx)$ может быть записана как $\mathbf{U}[\mathbf{S}(\mathbf{ZUU})\mathbf{U}]$ $[\mathbf{S}(\mathbf{ZUU})\mathbf{U}]$.

В последнем параграфе М. Шейнфинкель показывает, что достаточно даже не трех, а всего лишь одной функции $\mathbf{J}$, с помощью которой можно выразить все остальные. Она определяется следующим образом:

$\mathbf{JC} = \mathbf{U}$
$\mathbf{JS} = \mathbf{C}$
$\mathbf{J}x = \mathbf{S}$, где x отличен от $\mathbf{C}$ и $\mathbf{S}$.

Легко проверить, что имеет место $\mathbf{JJ} = \mathbf{S}$, $\quad \mathbf{J}(\mathbf{JJ}) = \mathbf{JS} = \mathbf{C}$, $\mathbf{J}[\mathbf{J}(\mathbf{JJ})] = \mathbf{JC} = \mathbf{U}$.

Возможно, это смутило даже его, так как далее он замечает, что в силу произвольного характера этой функции “*она едва ли имеет существенное значение*”.

Очевидно, что полученные М. Шейнфинкелем результаты имеют не только математическое, но и глубокое философское значение, так как относятся к глубинным основам математической деятельности. Возможно, это лишь случайное совпадение, но Д. Гильберт, который после знаменитого выступления на Математическом конгрессе 1900 г. основное внимание стал уделять вопросам физики и ее аксиоматизации, именно зимой 1920-1921 гг., когда и был сделан доклад М. Шейнфинкеля, вновь заинтересовался основаниями математики.

При всей ясности изложения результаты М. Шейнфинкеля были настолько необычны, что требовали глубокого осмысления. Он всего лишь заложил первый камень в исследования по комбинаторной логике, но не успел оформить их в законченную систему. Ее современным видом и самим термином мы обязаны Хаскеллу Бруксу Карри, который самостоятельно переоткрыл комбинаторную логику[4] и вдохнул в нее жизнь[5].

В одной небольшой статье невозможно изложить всю историю комбинаторной логики. Тем более что к ней были причастны люди с такими громкими именами, как Х.Б. Карри, А. Чёрч, С.К. Клини, Дж. Россер, А. Тьюринг, У. Куайн, Ф. Фитч. Поэтому мы приведем лишь современную формулировку комбинаторной логики и родственного ей λ-исчисления А. Чёрча, сформулируем полученные в них основные результаты и дадим доступную библиографию, которая позволит читателю более подробно ознакомиться с данным предметом и, возможно, заинтересоваться им с целью дальнейшего глубокого изучения.

1 Исчисление редукций чистой комбинаторной логики

1.1 Исходные символы

1. Var — множество переменных;
2. K, S — атомарные комбинаторы;
3.), (— скобки.

Все термы языка комбинаторной логики принадлежат одному типу.

1.2 Термы

1. Если $x \in Var$, то x — терм;
2. K и S — константные атомарные термы;
3. Если X и Y — термы, то (XY) — терм;
4. Ничто другое термом не является.

[4] *Curry H.B.* Grundlagen der kombinatorischen Logik // American Journal of Mathematics. 1930. Bd. 52. S. 509–536, 789–834.

[5] *Curry H.B., Feys R.* Combinatory Logic. Vol. 1. Amsterdam, 1958.

Терм вида (XY) называют аппликацией (применением) терма (функции) X к терму Y. В этом принципиальное отличие комбинаторной логики от современной теории категорий. Теория категорий строится на основе понятия композиции функций, а комбинаторная логика – на основе понятия применения функции к аргументу.

Термы, составленные лишь из константных термов K и S, будем называть *комбинаторами* и выделять в тексте жирным шрифтом. Поскольку при таком определении терма он может содержать много скобок, будем для облегчения восприятия по возможности опускать лишние скобки, предполагая их ассоциацию влево. Например, терм $(((XY)Z)U)$ после опускания скобок может быть записан просто как $XYZU$, а терм $((X(YZ))U)$ примет вид $X(YZ)U$.

В современной комбинаторной логике для обозначения комбинаторов М. Шейнфинкеля **I**, **C**, **T**, **Z**, **S** используют символы I, K, C, B, S.

В исчислении редукций комбинаторной логики изучают отношение редуцируемости между ее термами, которое обозначают посредством символа $\geq$. Это отношение можно рассматривать как аналог отношения выводимости между формулами классической логики.

1.3 Аксиомы

1. $X \geq X$ — для атомарных термов X
2. $\mathsf{K}XY \geq X$
3. $\mathsf{S}XYZ \geq XZ(YZ)$

1.4 Правила вывода

1. $X \geq Y \Rightarrow XZ \geq YZ$
2. $X \geq Y \Rightarrow ZX \geq ZY$
3. $X \geq Y, \;\; Y \geq Z \Rightarrow X \geq Z$

1.5 Определение доказательства. Последовательность редукций $< R_1, \ldots, R_n >$ называется доказательством, если для любого $i \leq n$ редукция R_i либо является аксиомой, либо получена из предшествующих редукций последовательности по одному из правил вывода. В этом случае редукция R_n называется доказуемой.

Иногда к приведенным выше трем правилам вывода добавляют еще одно

4. $X \geq Y \Rightarrow Y \geq X$

В этом случае говорят о *чистой комбинаторной логике* и вместо символа $\geq$ часто используют $=_*$.

1.6 Соглашения

1. Будем использовать символ $\equiv$ для обозначения графического равенства термов как языковых выражений.

2. Будем называть термы вида $\mathsf{K}XY$ и $\mathsf{S}XYZ$ *редексами*, а X и, соответственно, $XZ(YZ)$ – их *свертками.*

3. Будем говорить, что терм X находится в *нормальной форме*, если в него не имеет вхождения ни один редекс.

4. Посредством $T[Z_1/x_1, \ldots, Z_n/x_n]$ или $T[\overrightarrow{Z_n/x_n}]$ будем обозначать результат *одновременной подстановки* термов $Z_1, \ldots, Z_n$ в терм T вместо всех вхождений переменных $x_1, \ldots, x_n$, определяемый следующим образом:

   (a) $x_i[\overrightarrow{Z_n/x_n}] \equiv Z_i \quad 1 \leq i \leq n$

   (b) $y[\overrightarrow{Z_n/x_n}] \equiv y \quad y$ — атомарный терм, отличный от x_i, $1 \leq i \leq n$

   (c) $(XY)[\overrightarrow{Z_n/x_n}] \equiv (X[\overrightarrow{Z_n/x_n}])(Y[\overrightarrow{Z_n/x_n}])$

5. Посредством $FV(T)$ будем обозначать множество всех переменных, входящих в терм T.

Можно показать, что редукция $X \geq Y$ доказуема, е. и т. е. терм Y либо графически равен X, либо существует такая последовательность термов $< X_0, \ldots, X_n >$, что $X_0 \equiv X$, $X_n \equiv Y$ и для всякого $i > 0$ терм X_i получен из терма X_{i-1} путем единичной замены вхождения некоторого редекса на его свертку.

Исчисление редукций и сама комбинаторная логика обладают рядом важных свойств. Мы приведем их лишь для исчисления редукций, так как для комбинаторной логики они формулируются аналогичным образом.

1.7 Комбинаторная полнота. Для любого терма Т, все переменные которого содержатся среди $\{x_1, \ldots, x_n\}$, существует такой комбинатор $\mathbf{D}$, что доказуема редукция $\mathbf{D}x_1 \ldots x_n \geq T$.

Комбинаторная полнота является аналогом неограниченной свертки в наивной теории множеств. Благодаря этому свойству исчисление редукций обладает большими выразительными возможностями.

Известно несколько алгоритмов нахождения комбинатора $\mathbf{D}$. Приведем самый простой.

Пусть x – переменная, а T – терм. Обозначим посредством $[x].T$ терм, полученный в результате применения следующего алгоритма:

- $[x].x \equiv \mathsf{SKK}$
- $[x].X \equiv \mathsf{K}X$, если X не содержит вхождений переменной x
- $[x].XY \equiv \mathsf{S}([x].X)([x].Y)$

Определим терм $[x_1, \ldots, x_n].T$ с помощью рекурсии $[x_1, \ldots, x_n].T = [x_1].([x_2, \ldots, x_n].T)$. В этом случае будем говорить, что терм $[x_1, \ldots, x_n].T$ получен из терма T посредством функциональной абстракции относительно переменных $x_1, \ldots, x_n$. Терм $[x_1, \ldots, x_n].T$ и есть искомый комбинатор.

Можно показать, что для любых термов $Y_1, \ldots, Y_n$ доказуема редукция: $([x_1, \ldots, x_n].T)Y_1 \ldots Y_n \geq T[Y_1/x_1, \ldots, Y_n/x_n]$.

1.8 Свойство Чёрча–Россера. Если доказуемы редукции $X \geq Y$ и $X \geq Z$, то существует такой терм U, что доказуемы редукции $Y \geq U$ и $Z \geq U$.

Отсюда следует, что если термы Y и Z находятся в нормальной форме, то $Y \equiv Z$. Это позволяет представлять в исчислении редукций функциональные отношения. Из свойства Чёрча–Россера следует также, что исчисление редукций непротиворечиво в смысле нетривиальности.

1.9 Представимость рекурсивных функций. В исчислении редукций представимы все рекурсивные функции.

1.10 Стандартная форма доказательств. Если доказуема редукция $X \geq Y$, где Y находится в нормальной форме, то терм Y либо графически равен X, либо существует такая последовательность термов $< X_0, \ldots, X_n >$, что $X_0 \equiv X$, $X_n \equiv Y$ и для всякого $i > 0$ терм X_i получен из терма X_{i-1} путем замены самого левого вхождения редекса его сверткой.

Существование стандартных форм доказательств очень важно с точки зрения приложений. Если мы представили некоторую вычислимую функцию посредством терма исчисления редукций, то существует стандартный алгоритм ее вычисления.

Алгебраическими моделями комбинаторной логики являются комбинаторные алгебры, которые еще называют комбинаторно полными аппликативными структурами.

1.11 Структура $M =< X, * >$ называется *аппликативной*, если $*$ — бинарная операция на множестве X.

Очевидно, что это определение совпадает с определением группоида.

1.12 Комбинаторная алгебра – это аппликативная структура $M =< X, *, k, s >$ с двумя выделенными элементами k, s, удовлетворяющими равенствам

$$kxy = x,\ sxyz = xz(yz)^6.$$

"*Аксиомы комбинаторных алгебр порождены не алгебраическими соображениями, а анализом рекурсивных процессов. ... эти структуры – патологические с алгебраической точки зрения*"[7]. Нетривиальные комбинаторные алгебры некоммутативны, неассоциативны, неконечны и нерекурсивны.

Возникновение комбинаторной логики было связано с поиском новых оснований математики. Х. Карри надеялся найти их в обобщенной теории функциональности, которой по своей сути

[6] *Барендрегт Х.* Лямбда-исчисление. Его синтаксис и семантика. М.: Мир, 1985. С. 103.

[7] Там же. С. 105.

и должна была стать комбинаторная логика. Добавив к ней комбинаторы, соответствующие импликации и кванторам, он построил *иллативную комбинаторную логику*. В 1934 г. С.К. Клини и Дж. Россер показали, что в этой логике имеет место аналог парадокса Ришара. Проанализировав причины возникновения парадокса, Х. Карри пришел к выводу о несовместимости свойства комбинаторной полноты и свойства дедуктивной полноты, выражаемого хорошо известной теоремой дедукции. Это позволило ему сформулировать новый парадокс, получивший впоследствии имя *парадокса Карри*.

2 Чистое λ-исчисление Чёрча

Одновременно с Х. Карри поисками новых функциональных оснований математики занимался Алонзо Чёрч[8]. С его именем связано появление λ-исчисления. Если теоретико-множественное представление функций является экстенсиональным, то в λ-исчислении они трактуются интенсионально, как некоторые предписания. Привычное еще со школьной скамьи представление функций в виде формул как раз и является таким вычислительным предписанием.

2.1 Исходные символы

1. Var — множество переменных;
2. λ — оператор абстракции;
3.), (— скобки.

Все выражения языка чистого λ-исчисления принадлежат одному типу.

2.2 Термы

1. Если $x \in Var$, то x — терм;
2. Если X и Y — термы, то (XY) — терм;
3. Если $x \in Var$, Y — терм, то (λxY) — терм;

[8] *Church A.* A set of postulates for the foundation of logic // Annals of Math. 1932. Vol. 33. № 2. P. 346–366 and Annals of Math. 1933. Vol. 34. P. 839–864.

4. Ничто другое термом не является.

Оператор абстракции связывает переменные. Очевидным образом вводятся понятия свободных и связанных переменных, области действия оператора абстракции, подстановки терма вместо свободной переменной. На подстановку терма T вместо свободной переменной x в терм Y, обозначаемую посредством $Y[T/x]$, налагается обычное ограничение, что ни одна свободная переменная терма T после его подстановки не становится связанной. Операция подстановки всегда определена, поскольку в λ-исчислении возможно переименование связанных переменных по аналогии с тем, как это делается в исчислении предикатов.

В λ-исчислении изучается *отношение конверсии* между его термами, которое мы будем обозначать посредством символа $=$.

2.3 Аксиомы

1. $((\lambda xY)Z) = Y[Z/x]$ — β-конверсия
2. $X = X$ — для атомарных термов X

2.4 Правила вывода

1. $X = Y \Rightarrow Y = X$
2. $X = Y \Rightarrow XZ = YZ$
3. $X = Y \Rightarrow ZX = ZY$
4. $X = Y,\ Y = Z \Rightarrow X = Z$
5. $X = Y \Rightarrow (\lambda zX) = (\lambda zY)$ — правило ξ

Определение доказательства очевидно.

По аналогии с комбинаторной логикой мы будем называть λ-термы вида $((\lambda xY)Z)$ *редексами*, а λ-термы вида $Y[Z/x]$ — их *свертками*. Если понимать λ-терм вида (λxY) как некоторое вычислительное предписание, то λ-терм вида $Y[Z/x]$ является результатом применения этого предписания к терму Z.

λ-исчисление обладает свойством Чёрча–Россера, оно непротиворечиво, и для него также имеет место теорема о стандартной форме доказательств. В λ-исчислении так же, как и в комбинаторной логике, представимы все рекурсивные функции.

Будучи построены независимо друг от друга, комбинаторная логика и λ-исчисление тесно связаны между собой.

2.5 φ — функция перевода термов комбинаторной логики в термы λ-исчисления.

1. $\varphi(x) = x$, если x — переменная
2. $\varphi(\mathsf{K}) = (\lambda x(\lambda yx))$
3. $\varphi(\mathsf{S}) = (\lambda x(\lambda y(\lambda z((xz)(yz)))))$
4. $\varphi(XY) = (\varphi(X)\varphi(Y))$

2.6 ψ — функция перевода термов λ-исчисления в термы комбинаторной логики.

1. $\psi(x) = x \quad x$ — переменная
2. $\psi(XY) = (\psi(X)\psi(Y))$
3. $\psi(\lambda xY) = [x].\psi(Y)$

Имеет место следующая теорема.

2.7 Теорема. Если в чистой комбинаторной логике доказуемо $X =_* Y$, то в чистом λ-исчислении доказуемо $\varphi(X) = \varphi(Y)$.

Обратная теорема не имеет места, так как в комбинаторной логике не выполняется аналог ξ-правила $X =_* Y \Rightarrow [z]X =_* [z].Y$. Тем не менее, Х. Карри показал, что имеется конечный набор редукций, добавление которых в качестве новых аксиом приводит к тому, что если $X = Y$ доказуемо в чистом λ-исчислении, то $\psi(X) = \psi(Y)$ доказуемо в комбинаторной логике.

В 1936 г. А. Чёрч средствами чистого λ-исчисления получил результат, который сделал его знаменитым[9]. Он доказал существование неразрешимых проблем. Отсюда следовали неразрешимость арифметики и неразрешимость исчисления предикатов первого порядка.

Результат о λ-определимости арифметических функций позволил А. Чёрчу выдвинуть тезис, что класс всех функций, которые являются вычислимыми с интуитивной точки зрения, совпадает с классом функций, определимых в λ-исчислении. В 1937 г. А. Тьюринг доказал, что класс λ-определимых функций совпадает с классом функций, определимых в его собственном формализме.Это явилось первым подтверждением тезиса Чёрча.

Часто можно услышать, что λ-исчисление оказалось противоречивым и потому А. Чёрч перестал им заниматься. Это не верно. Так же, как и Х. Карри, А. Чёрч занимался поисками новых оснований математики и логики. Построение λ-исчисления как раз и преследовало эту цель. Как и Х. Карри, он добавил к нему специальные логические аксиомы, и это получившееся в результате исчисление оказалось противоречивым. Попытка построить бестиповую логику высших порядков, из которой можно было бы вывести всю математику, потерпела неудачу. Само же чистое λ-исчисление непротиворечиво, как непротиворечива и чистая комбинаторная логика. В отличие от Х. Карри, после обнаружения противоречий А. Чёрч действительно стал заниматься другими вопросами, и долгое время, почти три с лишним десятилетия, тематику комбинаторной логики и λ-исчисления разрабатывал лишь очень ограниченный круг ученых. Все изменилось в конце 50-х годов прошлого столетия в связи с развитием вычислительной техники и теоретического программирования.

Первым предложил использовать комбинаторы в программировании еще Х. Карри[10], но в то время это не получило должного отклика. Следующим был Ф. Фитч[11]. В конце 50-х и на-

---

[9] *Church A.* An unsolvable problem of elementary number theory // American Journal of Mathematics. 1936. Vol.58. P. 345–363.

[10] *Curry H.B.* The logic of program composition // In Applications Scientifiques de la Logique Mathematique, Actes du Deuxieme Colloque International de Logique Mathematique, Paris, 1952. P. 97–102. Gauthier-Villars, Paris, 1954.

[11] *Fitch F.B.* Representation of sequential circuits in combinatory logic //

чале 60-х годов Дж. Маккарти создает язык символьного программирования LISP, в котором есть много заимствований из λ-исчисления, в частности — механизм функциональной абстракции. Им был предложен новый способ организации программ, получивший впоследствии название *функционального программирования*[12]. В начале 60-х годов П. Ландин предлагает использовать λ-термы для кодирования структур известного языка программирования Algol-60[13]. В 1963 г. он же описывает абстрактную SECD-машину для редукции λ-термов, рассматриваемых как программы[14]. Дальнейший рост интереса к комбинаторной логике и λ-исчислению привел к получению многих важных теоретических и практических результатов, перечислить которые в одной небольшой статье не представляется возможным.

Выше мы говорили лишь о бестиповых исчислениях, но были разработаны и такие, в которых каждому терму сопоставлен его тип. Комбинаторная логика с типами оказалась тесно связанной с импликативной интуиционистской логикой. На это обратил внимание в 1934 г. Х. Карри[15] . В 1958 г. он отметил[16], что типам комбинаторов соответствуют доказуемые формулы интуиционистской логики.

Дальнейшие исследования[17] привели к новым результатам. В настоящее время эта тема активно разрабатывается, так как находит применение не только в теоретическом, но и практическом программировании, а само соответствие получило название *изоморфизма Карри–Говарда*.

Связь между типами базовых комбинаторов и формулами им-

Philosophy of Science. 1958. Vol. 25. P. 263–279.

[12] *McCarthy J.* A basis for a mathematical theory of computation // Proc. of 1961 Western Joint Computer Conference.

[13] *Landin P.J.* A correspondence between ALGOL 60 and Church's lambda notation // Communications of the ACM. 1965. Vol. 8. P. 89–101, 158–165.

[14] *Landin P. J.* The mechanical evaluation of expressions // The Computer Journal. 1964. Vol. 6. P. 308–320.

[15] *Curry H.* Functionality in Combinatory Logic // Proceedings of the National Academy of Sciences. 1934. Vol. 20. P. 584–590.

[16] *Curry H.B., Feys R.* Combinatory Logic. Vol. 1. Amsterdam, 1958.

[17] *Howard W.A* The formulae-as-types notion of construction // To H. B. Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism. Academic Press, London, 1980. P. 479–490. Manuscript circulated 1969.

пликативного вида позволила по-новому взглянуть на отношение между различными логиками и дать их стройную классификацию[18].

3 Логика дефинициальной дедукции[19]

Существует определенный перекос между большим интересом, который проявляют к комбинаторной логике и λ-исчислению представители computer science, и совершенно незначительным интересом со стороны логиков-философов. Этому есть две причины. Во-первых, при изложении этих исчислений, как правило, указывают на их предполагаемую функциональную интерпретацию, которая гораздо интереснее именно представителям computer science, а не логикам-философам. Во-вторых, они очень абстрактны и, в случае λ-исчисления, первоначально могут даже отпугнуть своей сложностью.

Оказывается, возможен третий чисто логический и гораздо более простой подход к представлению этих же структур. В его основе лежат две хорошо известные логические операции — аналоги введения определений и замены на их основе.

3.1 Исходные символы языка

1. Var — множество переменных;
2. $Const$ — множество констант, которое изначально является пустым;
3. $=_{def}$ — символ для введения определений;
4.), (— скобки.

3.2 Термы

1. Если $x \in Var$, то x — терм;
2. Если $c \in Const$, то c — терм;

[18] *Карпенко А.С.* Классификация пропозициональных логик // Логические исследования. Вып.4. М.: Наука, 1997. С. 107–133.

[19] *Шалак В.И.* Логический анализ дефинициальной дедукции // Логические исследования. Вып.15. М.: Наука, 2008.

3. Если X и Y — термы, то (XY) — терм;

4. Ничто другое термом не является.

Терм вида (XY) не предполагает никакой подразумеваемой интерпретации. Это не аппликация функции X к аргументу Y, а просто рядоположенность двух термов, которым могут соответствовать любые два объекта мысли.

Кроме термов, объектный язык содержит конструкции, называемые определениями.

3.3 Определения

1. Если T — терм и $FV(T) \subseteq \{x_1, \ldots, x_n\}$, то $Dx_1 \ldots x_n =_{def} T$ — определение, где D — новая константа.

2. Ничто другое определением не является.

Принятие определения и введение в язык новой константы D влечет за собой расширение множества правильно построенных термов. Примем дополнительные соглашения.

3.4 Соглашения

1. Пусть Δ — некоторое множество определений. Посредством $Const(\Delta)$ будем обозначать множество всех констант, введенных определениями Δ, т.е.

   $Const(\Delta) = \{D : (Dx_1 \ldots x_n =_{def} T) \in \Delta\}$.

2. Посредством $\mathrm{L}(\Delta)$ будем обозначать множество всех правильно построенных термов в языке с $\mathrm{Const} = \mathrm{Const}(\Delta)$.

3.5 Согласованное множество определений

1. Пустое множество определений является согласованным;

2. Если множество определений Δ согласовано, то множество определений $\Delta \cup \{Dx_1 \ldots x_n =_{def} T\}$, где $T \in L(\Delta)$, но $D \notin L(\Delta)$, также является согласованным;

3. Ничто другое согласованным множеством определений не является.

Смысл, который вкладывается в понятие согласованного множества определений, достаточно очевиден. Он заключается в том, что определения принимаются последовательно, и потому ранее принятые определения не могут содержать констант, которые будут введены в язык позже.

3.6 Правило замены по определению. Если $Dx_1 \ldots x_n =_{def} T$ – определение, а $X\{DZ_1 \ldots Z_n\}$ – терм, с выделенным вхождением терма $DZ_1 \ldots Z_n$, то $X\{T[\overrightarrow{Z_n/x_n}]\}$ есть результат замены $DZ_1 \ldots Z_n$ согласно определению на $T[\overrightarrow{Z_n/x_n}]$.

$$Dx_1 \ldots x_n =_{def} T, \quad X\{DZ_1 \ldots Z_n\} \Rightarrow X\{T[\overrightarrow{Z_n/x_n}]\}$$

3.7 Дефинициальной дедукцией (выводом) терма Y из согласованного множества определений Δ и терма $X \in L(\Delta)$ называется такая непустая конечная последовательность термов $< X_0, \ldots, X_n >$, что $X_0 \equiv X$, $X_n \equiv Y$ и для всякого $i > 0$ терм X_i получен из терма X_{i-1} по правилу замены.

Можно показать, что это исчисление в строго определенном смысле дефиницально эквивалентно исчислению редукций чистой комбинаторной логики. Отсюда следует, в частности, что в логике дефинициальной дедукции представимы все рекурсивные функции арифметики. Удивительно здесь то, что в этой логике изначально нет ни одной аксиомы и ни одного дескриптивного термина, как в комбинаторной логике. В этом смысле ее язык пуст и ничего кроме переменных и термов, построенных из них, не содержит. В то же время в этой логике нет связанных переменных, как в λ-исчислении, что делает ее гораздо более простой и понятной. Субъект дефинициальной логики всего лишь обладает способностью вводить определения и производить замену согласно им. Тем не менее, эти базовые логические операции, к которым мы привыкли относиться как к техническому приему, приводят к нетривиальным результатам.

4 Библиография

Выше уже упоминалось, что на русском языке очень мало публикаций, посвященных комбинаторной логике и λ-исчислению. Поэтому мы посчитали необходимым привести список работ, которые легко доступны и пригодны для начального ознакомления с предметом.

4.1 В 1987 г. в издательстве "Мир" была издана книга Э. Энгелера "*Метаматематика элементарной математики*". В ее третьей главе очень доступно излагаются основы комбинаторной логики и λ-исчисления.

4.2 В 1985 г. в издательстве "Мир" вышла книга Х. Барендрегта "*Ламбда-исчисление. Его синтаксис и семантика*". Ее по праву называют энциклопедией λ-исчисления. В книге подробнейшим образом излагаются λ-исчисление и комбинаторная логика, приводятся доказательства всех основных теорем, полученных в данной области. Большое количество упражнений позволяют использовать ее как превосходный учебник. Для тех, кто впервые открывает эту книгу, кажущаяся сложность изложения компенсируется логически строгими и полными доказательствами теорем. Автор сам заботится о том, чтобы облегчить ее чтение.

4.3 На механико-математическом факультете МГУ много лет исследованиями в области комбинаторной логики и оснований математики занимается А.Б. Кузичев. На его персональной странице в сети Интернет по адресу http://kuzichev.boom.ru можно найти много полезной информации.

4.4 Тем, кто не решится тратить свое время на изучение книги Х. Барендрегта, можно порекомендовать написанное им в соавторстве с Э. Барендсеном небольшое по объему "*Introduction to λ-calculus*", которое доступно по адресу http://www.cs.ru.nl/E.Barendsen/onderwijs/sl2/materiaal/lambda.pdf.

4.5 Хорошее изложение комбинаторной логики и λ-исчисления можно найти в "*Lecture Notes on the Lambda Calculus*" Петера Селинджера, которое также доступно для скачивания по адресу http://www.mscs.dal.ca/~selinger/papers/lambdanotes.pdf

4.6 Истории λ-исчисления и комбинаторной логики посвящена глава “*History of Lambda-calculus and Combinatory Logic*”, написанная Р. Хиндли и Ф. Кардоне для пятого тома *Handbook of the History of Logic.* Текст доступен по адресу http://www-maths.swan.ac.uk/staff/jrh/papers/JRHHislamWeb.pdf.

4.7 Еще одну работу по истории “*The Logic of Curry and Church*”, написанную также для пятого тома *Handbook of the History of Logic* Дж. Селдином, можно скачать по адресу http://people.uleth.ca/~Ejonathan.seldin/CCL.pdf.