Доклад на заседании научно-исследовательского семинара сектора логики Института философии РАН.

6 мая 2010 г.

ВВЕДЕНИЕ В КОМБИНАТОРНУЮ ЛОГИКУ ШЕЙНФИНКЕЛЯ-КАРРИ И λ-ИСЧИСЛЕНИЕ ЧЁРЧА

Историю комбинаторной логики и λ-исчисления принято отсчитывать от 7 декабря 1920 года - даты доклада М.И. Шейнфинкеля перед Математическим обществом Гёттингена. Сразу после доклада Д. Гильберт рекомендовал его к публикации, которая и была осуществлена четырьмя годами позже в известном математическом журнале «Mathematische Annalen» под названием «О строительных блоках математической логики». Помощь в подготовке публикации оказал Г. Беман. По некоторой информации он не только оказывал помощь в публикации, но и сам ее подготовил, так как М. Шейнфинкель не был в состоянии этого сделать.

Несмотря на широкую известность имени М. Шейнфинкеля, о самом авторе мы знаем на удивление мало – две статьи, одна сохранившаяся фотография и скупые биографические данные. Известно, что он родился 4 сентября то ли 1887, то ли 1889 года в городе Екатеринослав (ныне Днепропетровск на Украине). Изучал математику в Одессе в университете, который тогда назывался Новороссийским. Его руководителем был Самуил Осипович Шатуновский – известный российский математик, много внимания уделявший вопросам геометрии и ее оснований. Нет ничего удивительного в том, что в 1914 году М. Шейнфинкель приезжает в математическую Мекку того времени – в Гёттинген к Д. Гильберту. Там он сделал свой знаменитый доклад, там же, будучи ровесником, тесно общался с П. Бернайсом. В 1929 году ими была подготовлена совместная публикация, посвященная проблеме разрешимости одного частного класса формул исчисления предикатов. К тому 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 году 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.

Интересно отметить следующее совпадение. После знаменитого Математического Конгресса 1900 года, на котором Д. Гильберт сформулировал свои знаменитые 23 проблемы, он отошел от занятия чистой математикой и переключился на математическую физику. К. Рид в книге «Гильберт» пишет, что у него было два помощника. Один - математик, другой - физик. По тому, сколько времени он проводил с каждым из них, можно было судить о том, чем он занимается. Так вот после декабря 1920 года большую часть времени Д. Гильберт опять стал проводить с помощником по математике. Возможно, это чистая случайность, а возможно, что доклад М. Шейнфинкеля вновь пробудил у Д. Гильберта интерес к основаниям математики, ведь именно после этого он приступил к созданию теории доказательств.

В России с комбинаторной логикой и λ-исчислением знакомы довольно слабо. Историческое объяснение может заключаться в том, что Г. Беман, редактор статьи М. Шейнфинкеля, позже стал членом нацистской партии, и это по идеологическим соображениям помешало публикации ее перевода в СССР. В то же время с результатами М. Шейнфинкеля была знакома С.А. Яновская. Она даже напечатала статью на эту тему в энциклопедии. Результаты М. Шейнфинкеля были известны Кузнецову (из Кишенева). Большим энтузиастом изучения комбинаторной логики и λ-исчисления долгие годы был и ныне является Александр Сергеевич Кузичев, работающий на механико-математическом факультете МГУ. Ситуация начала исправляться лишь в последнее время, да и то лишь в основном на факультетах вычислительной математики.

Учитывая существующий во всем мире большой интерес к комбинаторной логике и λ-исчислению, попытаемся понять, в чем их изюминка, почему они действительно представляют интерес не только для математиков, но и для философов.

КОМБИНАТОРНАЯ ЛОГИКА

Начнем с комбинаторной логики.

Если спросить у логика или математика, что такое функция, ответ, скорее всего, будет следующим.

Функция F из множества A в множество B – это такое подмножество декартова произведения $A\times B$, что для всякого $a\in A$ существует единственный $b\in B$, для которых $<a,b>\in F$.

Такое теоретико-множественое определение функций стало входить в широкий обиход лишь в XX веке. До этого они понимались не как множества, а как определенные правила соответствия. Функции, сформулированные в виде правил, можно обнаружить уже на глиняных табличках Вавилона и древнеегипетских папирусах. С функциями, сформулированными в виде правил, имеют дело школьники, когда приступают к изучению арифметики, с ними имеем дело и мы, когда садимся за компьютер и запускаем какую-нибудь программу. Каждая компьютерная программа представляет собой некоторую комбинацию предписаний по вычислению тех или иных функций. Обязанности по выполнению этих предписаний возложены на «мозг» компьютера - его процессор.

Для того, чтобы это стало возможным, понятие функции как предписания должно было получить математически строго уточнение, что и было сделано в 20-е годы прошлого столетия с появлением комбинаторной логики и λ-исчисления.

Рассмотрим пример формулы, задающей правило вычисления арифметической функции.

$$f(x,y) = (x+2)\times(y+3)$$

Она представляет следующее правило: *взять результат сложения x с числом 2, и умножить его на результат сложения y с числом 3*. Это и будет результатом вычисления функции f в применении к аргументам x и y.

Рассмотрим другую формулу.

$$g(x,y) = (x\times y)+(3\times x)+(2\times y)+6$$

С теоретико-множественной точки зрения, функции $f(x,y)$ и $g(x,y)$ одинаковы, но, с точки зрения правил вычисления, они разные.

Нам известны такие свойства умножения, как коммутативность $x\times y = y\times x$ и дистрибутивность относительно сложения $(x+y)\times z = (x\times z)+(y\times z)$, которые позволяют посредством цепочки равенств преобразовать формулу $(x+2)\times(y+3)$ к виду $(x\times y)+(3\times x)+(2\times y)+6$. Это преобразование также можно рассматривать как результат выполнения некоторого вычислительного предписания, основанного на свойствах сложения и умножения, но уже в применении не к числам, а к самим формульным предписаниям.

Мы можем сформулировать предписание **I**, которое в применении к любому объекту x оставляет его без изменения, т.е. $\mathbf{I}(x) = x$. Смысл этого предписания таков, что ничто не мешает взять в качестве объекта x само предписание **I** и получить $\mathbf{I}(\mathbf{I}) = \mathbf{I}$. Это означает, что функции как предписания могут быть и самоприменимыми, чего не скажешь о теоретико-множественных функциях.

Важным свойством комбинаторной логики и λ-исчисления является сведение многоместных функций к одноместным.

Пусть дана двухместная функция $h(x,y)$. Подставляя вместо первого аргумента x конкретные значения $a, b, c, \ldots$, мы будем получать одноместные функции $h(a,y)$, $h(b,y)$, $h(c,y), \ldots$

Это можно представить в виде следующего соответствия:

$$\begin{array}{lll} a & \to & h(a,y) \\ b & \to & h(b,y) \\ c & \to & h(c,y) \\ \ldots & \ldots & \ldots \\ \mathrm{x} & \mapsto & h(x,y) \end{array}$$

Очевидно, что само это соответствие задается некоторой функцией, которую можно обозначить посредством h_x. Если рассматривать ее как предписание, то она просто подставляет в исходное предписание по вычислению функции $h(x,y)$ вместо аргумента x конкретные значения. В результате такой подстановки мы получаем предписания, задающие вычисления одноместных функций. Если теперь мы применим предписание $h(a,y)$ к аргументу i, то в результате получим $h(a,i)$, т.е. то же самое, что получили бы с самого начала, если бы применили исходное предписание $h(x,y)$ к паре аргументов $<a, i>$. Так и производится сведение многоместных функций к одноместным.

В теории множеств обоснованием такого сведения служит изоморфизм множеств

$$Z^{X \times Y} \text{ и } Z^{XY}$$

Идея сведения достаточно прозрачна, но заметен недостаток существующей символики для ее представления в языке. Заслуга создания такого языка как раз и принадлежит А. Чёрчу. К этому мы еще вернемся, а пока договоримся представлять одноместные функции $h(x)$ в виде $(h{*}x)$, где звездочка “*” обозначает *аппликацию* - применение функции к аргументу. Это позволит нам в свою очередь представить конечный результат сведения $h(x,y)$ к одноместным функциям посредством $((h'{*}x){*}y)$. Понятно, что h и h' - это разные функции. Первая из них двухместная, а вторая - одноместная, но они находятся в соответствии сведения первой ко второй. Далее, для простоты, когда это не будет вызывать недоразумений, мы будем использовать для них одинаковые обозначения.

Необходимо с самого начала обратить внимание на отличие *аппликации* от *композиции* функций, чтобы никогда их уже не путать. Композиция одноместных функций $\mathbf{f}(x)$ и $\mathbf{g}(y)$ всегда дает одноместную функцию $\mathbf{f}(\mathbf{g}(x))$ и потому не позволяет свести многоместные функции к одноместным. Последовательная аппликация, применение функции к аргументу, позволяет произвести такое сведение.

Вернемся к нашему первому школьному примеру, заменив обозначение операции сложения “+” на “А”, а операции умножения “×” - на “М”, и перейдя от инфиксной записи термов к префиксной.

$$f(x,y) = (x+2)\times(y+3) = M(A(x,2), A(y,3))$$

В результате сведения к одноместным функциям операция умножения $M(x,y)$ примет вид $((M{*}x){*}y)$, а операция сложения $A(x,y)$ будет выглядеть как $((A{*}x){*}y)$.

$$((f{*}x){*}y) = ((M{*}((A{*}x){*}2)){*}((A{*}y){*}3))$$

Если опустить скобки, приняв соглашение об их левой ассоциативности при восстановлении, равенство можно записать еще более просто.

$$f{*}x{*}y = M{*}(A{*}x{*}2){*}(A{*}y{*}3)$$

Поскольку, кроме аппликации у нас нет никаких других операций, то можно принять соглашение об опускании символа “*∗*”. Само заключение двух термов в скобки будет предполагать, что они связаны аппликацией. В результате этого мы получим очень компактную запись

$$fxy = M(Ax2)(Ay3)$$

Правая часть определения функции *f* является скобочной комбинацией двух переменных и четырех констант. Константы *M, 2* и *3* имеют по одному вхождению, а константа *A* - два.

На примере дистрибутивности умножения относительно сложения мы уже упоминали, что вычислительные предписания применимы не только по отношению к числам, но и к символьным данным - самим предписаниям. Допустим, у нас имеется предписание формульного преобразования ***Q***, которое выглядит следующим образом.

$$\boldsymbol{Q}uvwzxy = u(vxw)(vyz)$$

Тогда мы можем представить функцию *f* как предписание ***Q****MA23*. Результатом ее применения к *x* и *y* будет:

$$fxy = \boldsymbol{Q}MA23xy = M(Ax2)(Ay3)$$

М.И. Шейнфинкелю принадлежит заслуга выделения двух базисных формульных преобразований, получивших впоследствии название **комбинаторов**, с помощью которых можно определить любые другие формульные преобразования, подобные ***Q***. Его выступление на семинаре у Д. Гильберта в декабре 1920 г. и последовавшая за этим публикация «*О строительных блоках математической логики*» положили начало комбинаторной логике. Основные ее понятия – комбинатор и редукция. **Комбинатор** – это замкнутый терм языка, определяющий преобразование термов, именуемое **редукцией**.

В вышеприведенном примере $\boldsymbol{Q}uvwzxy = u(vxw)(vyz)$ **комбинатор** - это **Q**, а **редукция** - это переход от любого терма вида ***Q****UVWZXY* к терму вида *U(VXW)(VYZ).*

Определения базисных комбинаторов выглядят следующим образом.

$$\mathbf{K}xy = x$$

$$\mathbf{S}xyz = xz(yz)$$

Если восстановить опущенные скобки, они примут вид

$$((\mathbf{K}x)y) = x$$

$$(((\mathbf{S}x)y)z) = ((xz)(yz))$$

С их помощью можно определять другие комбинаторы. Например, комбинатор тождественного преобразования, который в применении к любому терму *x* оставляет его неизменным, можно определить как **SKK**.

$$\mathbf{SK}\mathbf{K}x = \mathbf{K}x(\mathbf{K}x) = x$$

В качестве другого примера можно привести комбинатор **K**(**SKK**), реализующий функцию проекции на второй аргумент.

$$\mathbf{K}(\mathbf{SKK})xy = \mathbf{SKK}y = \mathbf{K}y(\mathbf{K}y) = y$$

Несколько проще это же преобразование можно задать с помощью комбинатора **SK**.

$$\mathbf{SK}xy = \mathbf{K}y(xy) = y$$

Таким образом, одно и то же соотношение *вход-выход* можно представить в языке комбинаторной логики различными алгоритмическими преобразованиями.

В качестве дополнительной иллюстрации покажем, как средствами комбинаторной логики представить свойство коммутативности сложения.

$$x + y = y + x$$

Легко проверить, что комбинатор **S**(**S**(**KS**)(**S**(**KK**)**S**))(**KS**), который мы для краткости обозначим посредством **C**, осуществляет следующее преобразование.

$$\mathbf{C}xyz = xzy$$

Свойство коммутативности сложения в аппликативной записи имеет вид Axy = Ayx. Поскольку **C**Axy = Ayx, то же самое может быть представлено как Axy = **C**Axy. Так как переменные x и y играют лишь вспомогательную роль, они могут быть опущены. Результирующая запись, выражающая коммутативность операции A, выглядит очень просто.

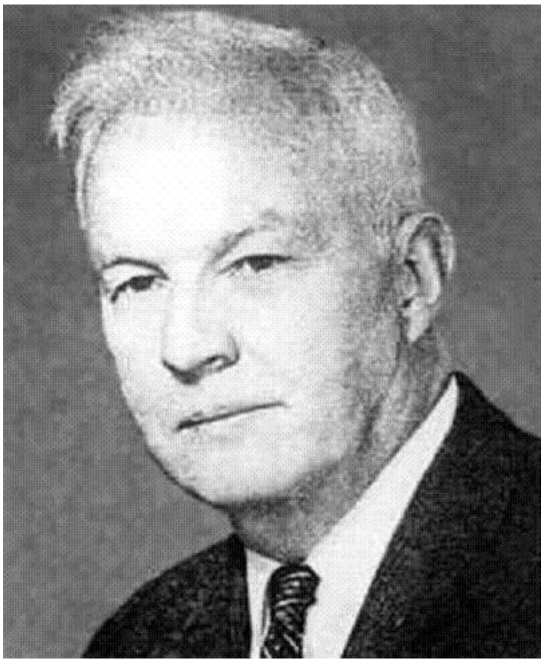
$$A = \mathbf{C}A$$

Никакие переменные не нужны, поскольку по правилам комбинаторной логики для любых двух термов X и Y из A = **C**A выводимо AXY = AYX.

$$A = \mathbf{C}A \quad \Rightarrow \quad AX = \mathbf{C}AX \quad \Rightarrow \quad AXY = \mathbf{C}AXY \quad \Rightarrow \quad AXY = AYX$$

Если подвести итог тому, что сделал М. Шейнфинкель, его основная заслуга заключается в выделении комбинаторов **K** и **S** и демонстрации того, как с их помощью выразить любые другие формульные преобразования, но сама комбинаторная логика как исчисление не была им построена, и никаких теорем в строгом смысле этого слова им доказано тоже не было.

Дальнейшим развитием комбинаторной логики, ее названием и современным видом мы обязаны Хаскеллу Карри. Справедливости ради необходимо сказать, что его заслуги в этом, за исключением приоритета, ничуть не меньше заслуг М. Шейнфинкеля. В течение нескольких десятилетий, когда существование комбинаторной логики фактически игнорировалось, он не только вопреки всему сохранял и развивал ее, но и воспитал целый ряд талантливых учеников, которые и помогли донести ее до нашего времени.



Хаскелл Карри родился 12 сентября 1900 года в городе Миллс в штате Массачусетс. Учился в Гарвардском университете, где сперва изучал медицину, потом физику, потом математику. В разные годы его руководителями были П. Бриджмен и Дж. Биркгоф. В конце концов, Карри заинтересовался логикой.

В 1922 году, прочитав «Principia Mathematica» Рассела и Уайтхеда, он обратил внимание на то, что формулировка правила подстановки, по сравнению с правилом modus ponens, очень запутанна. У него появилось желание разложить это правило на несколько более простых. Это и привело к идее *комбинаторов*. Так как в Гарварде никто не занимался логикой, Х. Карри отправился в 1927-1928 гг. на стажировку в Принстон и там, в библиотеке, обнаружил, что его собственные идеи уже предвосхищены в публикации М. Шейнфинкеля. Однако он понял, что М. Шенфинкель не довел свою работу до логического конца, и потому решил продолжить работу и написать на эту тему диссертацию.

Х. Карри уже знал, что М. Шейнфинкель болен и переехал в СССР. Тем не менее, он принимает решение отправиться в Гёттинген, чтобы пообщаться с людьми, которые лично знали М. Шейнфинкеля, - Г. Беманом и П. Бернайсом. В Гёттингене в течение года 1928-1929 Х. Карри работает над диссертацией. Его научным руководителем был Д. Гильберт, хотя практическое руководство осуществлял П. Бернайс. В это же время в течение полугода в Гёттингене находился и А. Чёрч, занимаясь сходными проблемами, но они с Х. Карри еще не были знакомы и потому ни разу не общались. Первая публикация Х. Карри на тему комбинаторной логики датируется 1929 годом. Он сумел представить комбинаторную логику в виде исчисления, близкого к тому, как это принято сейчас. Это позволило доказать ряд фундаментальных теорем и приступить к более глубокому изучению ее свойств.

Дадим определение комбинаторной логики, как это делают сейчас.

Исходные символы языка

1. x, y, z, … - переменные;
2. **K**, **S** – константы;
3.), (- скобки.

Термы

1. Всякая переменная есть терм;
2. **K** и **S** - термы;
3. Если X и Y - термы, то (XY) – терм;
4. Ничто другое термом не является.

Термы-переменные и термы-константы будем называть атомарными термами. Терм, составленный лишь из констант **K** и **S**, будем называть *комбинатором* и выделять в тексте жирным шрифтом. При опускании в термах скобок предполагается их ассоциация влево. Т.е. терм XYZU при восстановлении скобок примет вид (((XY)Z)U), а терм X(YZ)U - примет вид ((X(YZ))U).

Формулами комбинаторной логики являются выражения вида X = Y, где X и Y - термы.

Аксиомы

CA.1 X = X для атомарных термов X
CA.2 **K**XY = X
CA.3 **S**XYZ = XZ(YZ)

Правила вывода

CR.1 $\dfrac{X = Y,\ \ Y = Z}{X = Z}$

CR.2 $\dfrac{X = Y}{Y = X}$

CR.3 $\dfrac{X = Y}{XZ = YZ}$

CR.4 $\dfrac{X = Y}{ZX = ZY}$

Комбинаторная логика - это теория с единственным предикатом равенства, и все ее формулы имеют вид равенства двух термов. Смысл аксиом и правил вывода очевиден.

CA.1 - аксиома рефлексивности равенства, а правила CR.1 и CR.2 - его транзитивность и симметричность.

Оставшиеся два правила CR.3 и CR.4 - это конгруэнтность равенства.

Аксиомы CA.2 и CA.3 постулируют существование двух базисных преобразований.

Определение доказательства обычное.

Вместо того, чтобы заниматься доказательствами в аксиоматическом исчислении, можно поступить иначе. Легко показать, что формула U = W доказуема в комбинаторной логике, если и только если терм W можно получить из терма U путем цепочки последовательных замен входящих в них термов вида **K**XY на X, и термов вида **S**XYZ на XZ(YZ), или наоборот, т.е. X на **K**XY, а XZ(YZ) на **S**XYZ .

Как уже было сказано, М.И. Шейнфинкель выделил два базисных формульных преобразований, с помощью которых можно определить любые другие. Но это означает, что набор комбинаторов **K** и **S** в определенном смысле полон, т.е. для любого терма T, все переменные которого содержатся среди $x_1,\ldots,x_n$, существует такой выразимый с их помощью комбинатор **G**, что

$$(\mathbf{G}x_1\ldots x_n) = T$$

Смысл комбинаторной полноты заключается в том, что с помощью исходного набора комбинаторов реализуемо любое мыслимое формульное построение. Т.е. для любого терма T, являющегося произвольной скобочной комбинацией переменных $x_1,\ldots,x_n$ и произвольных комбинаторов, существует замкнутый терм **G**, кодирующий его построение.

Известен целый ряд алгоритмов нахождения таких комбинаторов **G**. Их называют алгоритмами функциональной абстракции. Приведем один из самых простых, хотя и очень неэффективный.

Функциональная абстракция

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

1. $[x].x \equiv \mathbf{SKK}$
2. $[x].y \equiv \mathbf{K}y$ если переменная *y* отлична от *x*
3. $[x].XY \equiv \mathbf{S}([x].X)([x].Y)$

Запись $[x_1,.., x_n].T$ означает результат последовательного применение данного алгоритма к *n* переменным, входящим в T.

$[x_n].T$

$[x_{n-1},x_n].T = [x_{n-1}].([x_n].T)$

. . . .

$[x_1,..,x_n].T = [x_1].([x_2,.., x_n].T)$

Будем говорить, что терм $[x_1,\ldots,x_n].T$ получен из терма T посредством функциональной абстракции относительно переменных $x_1,..., x_n$.

Попробуем понять смысл алгоритма функциональной абстракции.

Переход от терма T к замкнутому терму $[x_1,\ldots,x_n].T$, совершаемый согласно этому алгоритму, является своеобразным аналогом применения неограниченной аксиомы свертки в наивной теории множеств. Эта аксиома гласит, что для любого свойства $Q(x_1, \ldots,x_n)$ существует множество $\{<x_1,\ldots,x_n>: Q(x_1,\ldots,x_n)\}$, состоящее в точности из тех объектов, которые обладают данным свойством. Выражение $\{<x_1,\ldots,x_n>: Q(x_1,\ldots,x_n)\}$

является замкнутым термом теории множеств и может использоваться в рассуждениях, будучи самостоятельным объектом мысли. Точно так же и замкнутый терм $[x_1,\ldots,x_n].T$ может использоваться в рассуждениях, играя роль представителя структуры терма T.

В теории множеств от утверждения о принадлежности n-ки объектов $<t_1,\ldots,t_n>$ множеству $\{<x_1,\ldots,x_n>: Q(x_1,\ldots,x_n)\}$ можно совершить переход к утверждению об истинности $Q(t_1,\ldots,t_n)$, и наоборот.

$$<t_1,\ldots,t_n> \in \{<x_1,\ldots,x_n>: Q(x_1,\ldots,x_n)\} \leftrightarrow Q(t_1,\ldots,t_n)$$

Комбинаторным аналогом является переход посредством редукции от терма $([x_1, \ldots,x_n].T)Y_1\ldots Y_n$ к терму $T[Y_1/x_1,\ldots,Y_n/x_n]$.

Вспомним теперь, что в 1903 г. Б. Рассел обнаружил, что использование в наивной теории множеств неограниченной аксиомы свертки ведет к противоречиям, а поскольку математики того времени широко пользовались таким приемом построения математических объектов, открытие парадокса произвело на них сильнейшее впечатление. Принято считать, что это событие послужило началом кризиса оснований математики.

Алгоритм функциональной абстракции позволяет всякому терму языка сопоставить замкнутый терм, являющийся представителем его свойств. Никаких ограничений (!) на применение данного алгоритма не налагается, и ни к каким противоречиям (!) это не приводит. В существовании этого алгоритма и заключена особая притягательная сила комбинаторной логики.

М.И. Шейнфинкель получил еще один весьма интересный с философской точки зрения результат. Он показал, что можно ограничиться не двумя **K** и **S**, а всего одним специальным базисным комбинатором, через который легко определить все остальные, в том числе **K** и **S**. Если вспомнить, что любой конструктивный объект может быть закодирован словом в алфавите из двух букв, то этих двух букв и одного комбинатора оказывается достаточно для теоретического представления любых конструктивных объектов и любых конструктивных мыслимых построений.

И еще на чем следует остановиться, так это на необходимости отличать начальную мотивацию исследователей от того, к чему они, в конце концов, приходят. Если открыть оригинальную статью М. Шейнфинкеля, то можно прочитать, что он хотел избавиться от связанных переменных в формулах логики, поскольку они по своей сути не являются переменными в обычном смысле. Х. Карри, в свою очередь, хотел всего лишь упростить правило подстановки. В результате же они пришли к общей теории формульных преобразований.

Поскольку комбинаторная логика обладает свойством полноты, т.е. представления любых формульных преобразований, то нет ничего удивительного в том, что с ее помощью оказались представимы все эффективно вычислимые функции арифметики.

λ-ИСЧИСЛЕНИЕ ЧЁРЧА

Самым младшим из троицы, положившей начало новой области логико-математических исследований, был Алонзо Чёрч. Родился он 14 июня 1903 года в Вашингтоне, в 1924 году окончил Принстонский университет. А. Чёрч интересовался вопросами оснований математики. Его докторская диссертация 1927 года была посвящена системам, в которых аксиома выбора может быть ложной.

После написания диссертации А. Чёрч попытался построить бестиповую логическую систему, которая могла бы послужить в качестве новых оснований математики.

Подход, который развивал А. Чёрч, он сам выразил следующими словами.

«*В основании развиваемого исчисления лежит понятие функции в том виде, в каком оно встречается в различных разделах математики либо под этим же именем, либо под именами «операции», «преобразования» и пр. Изучение общих свойств функций независимо от их принадлежности конкретной области математики, является задачей логики или лежит на границе логики и математики. Такое изучение и есть исходная мотивация предлагаемого исчисления… Функция - это правило соответствия, с* *помощью которого, если нечто дано (как аргумент), то может быть получено другое (значение этой функции для данного аргумента)*»

В основу этой системы он хотел положить две операции - аппликацию (применение функции к аргументу) и функциональную абстракцию - специальную операцию, обратную по отношению к аппликации, и позволяющую из одних функций получать другие. Первая его публикация на эту тему датируется 1932 годом. К этому времени рядом с ним уже работали его ученики - С. Клини и Б. Россер. С. Клини занимался изучением самой системы А. Чёрча, а Б. Россер занялся изучением ее отношения к комбинаторной логике Х. Карри. В 1934 Клини и Россер показали противоречивость начальных систем Чёрча и Карри, которые включали не только те аксиомы и правила, которые включают в них сегодня, но и специальные логические аксимоы, выраженные в новых терминах. Так вот Клини и Россер в них непротиворечивые фрагменты, имеющие самостоятельное значение. Таким фрагментом в случае системы А. Чёрча оказалось чистое λ-исчисление, а в случае системы Карри - комбинаторная логика, формулировку которой мы привели ранее.

В 1936 году А. Черч средствами чистого λ-исчисления получил результат, который сделал его знаменитым. Он доказал существование неразрешимых проблем. Отсюда следовала неразрешимость арифметики и неразрешимость исчисления предикатов первого порядка.

Результат о λ-определимости арифметических функций подтолкнул А. Черча к мысли о том, что класс всех функций, которые вычислимы с интуитивной точки зрения, совпадает с классом функций, определимых в λ-исчислении. Это и есть знаменитый тезис Чёрча. В 1937 году А. Тьюринг доказал, что класс λ-определимых функций совпадает с классом функций, определимых в его собственном формализме, что явилось первым подтверждением тезиса Черча.

Как и в комбинаторной логике, основной операцией в λ-исчислении является аппликация – применение функции к аргументу. Отличие же заключается в том, что новые функции строятся не с помощью постулируемого набора базисных комбинаторов, а более непосредственно.

Рассмотрим следующий пример.

ПРЕДПИСАНИЕ [Сложить X с 5. Умножить получившийся результат на Y. Сложить получившийся результат с X, сложить с 2, и взять получившееся число в качестве результата выполнения предписания].

Более компактно это же предписание можно записать следующим образом:

$$((X + 5) * Y) + X + 2$$

Но в таком виде предписание еще не является функцией. Это всего лишь терм со свободными переменными, так как переменные в нем представляют лишь некоторые неопределенные объекты.

Точно так же не является высказыванием формула логики предикатов со свободными переменными **Любит**(X, Y). Чтобы превратить эту формулу в высказывание мы должны либо подставить в нее имена конкретных людей, либо связать переменные кванторами. Например,

Для **всякого** X **существует** такой Y, что **Любит**(X, Y).

Аналогичным образом, чтобы превратить терм в функцию, мы должны связать его переменные. Для этого А. Чёрч и ввел свой символ λ. Если применить его к переменным, входящим в наше предписание, то мы получим функцию

$$\lambda X \lambda Y.(((X + 5) * Y) + X + 2)$$

Эта операция связывания свободных переменных получила название **λ-абстракции**.

Какие операции с ней возможны?

Очевидно, что мы можем переименовывать ее связанные переменные. Например, замена переменной X на Z оставляет саму функцию неизменной.

$$\lambda Z \lambda Y.(((Z + 5) * Y) + Z + 2)$$

Но при этом необходимо соблюдать некоторые предосторожности, чтобы не возникло коллизий. Мы не можем переименовать X в Y, так как выражение

$$\lambda Y \lambda Y.(((Y + 5) * Y) + Y + 2)$$

представляет совсем другую функцию.

Переименование связанных переменных с соблюдением всех предосторожностей получило название **α-конверсии**.

Вторая, и самая главная, операция - это приложение функции к аргументам. Чтобы приложить функцию $\lambda X \lambda Y.(((X + 5) * Y) + X + 2)$ к аргументу (3+2) мы должны подставить его в тело функции вместо переменной X.

$$\lambda X \lambda Y.(((X + 5) * Y) + X + 2) \ \ (3+2) \ = \ \lambda Y.((((3+2) + 5) * Y) + (3+2) + 2)$$

Такая подстановка получила название **β-конверсии**.

Следует обратить внимание, что в результате применения нашей функции к (3+2) мы получили не конкретное число, а опять функцию, которую в свою очередь мы можем применить к новому аргументу, например, 9 и в результате получить

$$\lambda Y.((((3+2) + 5) * Y) + (3+2) + 2) \ \ 9 \ = \ ((((3+2)+5)*9)+(3+2)+2$$

Далее остается лишь применить обычные арифметические операции и получить 97.

Заметим, что при другой очередности связывания переменных мы получили бы другую функцию $\lambda Y \lambda X.(((X + 5) * Y) + X + 2)$. Результатом ее применения сначала к (3+2), а потом к 9 было бы другое число

$$\lambda Y \lambda X.(((X + 5) * Y) + X + 2) \ (3+2) \ = \ \lambda X.(((X + 5) * (3+2)) + X + 2)$$

$$\lambda X.(((X + 5) * (3+2)) + X + 2) \ \ 9 \ = \ (((9 + 5) * (3+2)) + 9 + 2) = 81$$

В λ-исчислении также принимается **правило ξ** - если для двух термов M и N доказано, что $M = N$, то и результаты применения к ним λ-абстракции равны, т.е. $\lambda x.M = \lambda x.N$.

Приведем теперь строгую формулировку чистого λ-исчисления.

Исходные символы

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

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 $(\lambda x Y)$ – терм;

4. Ничто другое термом не является.

Подстановка выглядит довольно сложно

1. $x[Z/x] \equiv Z$
2. $a[Z/x] \equiv a$ — a – атом, отличный от x
3. $(PQ)[Z/x] \equiv (P[Z/x]\ Q[Z/x])$
4. $(\lambda xP)[Z/x] \equiv (\lambda xP)$
5. $(\lambda yP)[Z/x] \equiv (\lambda yP)$ — $x \notin FV(P)$
6. $(\lambda yP)[Z/x] \equiv (\lambda yP[Z/x])$ — $x \in FV(P)$ и $y \notin FV(Z)$
7. $(\lambda yP)[Z/x] \equiv (\lambda zP[z/y][Z/x])$ — $x \in FV(P)$, $y \in FV(Z)$ и $z \notin FV(PZ)$

Сложная формулировка операции подстановки, необходимость различать свободные и связанные вхождения переменных, являются техническими недостатками λ-исчисления.

Формулами λ-исчисления являются выражения вида X = Y, где X и Y - λ-термы.

Аксиомы

1. $X = X$ - для атомарных термов X
2. $(\lambda xY) = (\lambda zY[z/x])$ - α - конверсия, $z \notin FV(Y) \cup BV(Y)$
3. $((\lambda xY)Z) = Y[Z/x]$ - β-конверсия

Смысл аксиом достаточно прост. Первая аксиома – это аксиома тождества.

Вторая аксиома позволяет осуществлять переименование связанных переменных. Вспомним, что в классической логике предикатов переименование связанных переменных является эквивалентным преобразованием формул.

Третья аксиома определяет приложение функции к аргументу, которое заключается в подстановке терма вместо переменной.

Правила вывода

1. $$\frac{X = Y}{Y = X}$$

2. $$\frac{X = Y}{(XZ) = (YZ)}$$

3. $$\frac{X = Y}{(ZX) = (ZY)}$$

4. $$\frac{X = Y,\ Y = Z}{X = Z}$$

5. $$\frac{X = Y}{(\lambda zX) = (\lambda zY)}$$ - правило ξ

Смысл правил вывода также очевиден. Это симметричность и транзитивность отношения конверсии плюс его конгруэнтность по отношению к аппликации и λ-абстракции.

Определение доказательства обычное.

Несмотря на то, что комбинаторная логика и λ-исчисление создавались независимо друг от друга, исходя из разных предпосылок, они оказались тесно связанными. Многие комбинаторы могут быть представлены замкнутыми термами λ-исчисления гораздо более просто, чем их оригиналы в комбинаторной логике. Так, например, комбинатор **K** может быть представлен замкнутым термом λ-исчисления (λxy.x). Проверим это.

$$(\lambda xy.x)XY = (\lambda y.X)Y = X = \mathbf{K}XY$$

Аналогично комбинатор **S** может быть представлен замкнутым термом λ-исчисления (λxyz.xz(yz)). Проверим это.

$$(\lambda xyz.xz(yz))XYZ = (\lambda yz.Xz(yz))YZ = (\lambda z.Xz(Yz))Z = XZ(YZ) = \mathbf{S}XYZ$$

В то же время алгоритм функциональной абстракции, о котором мы говорили в связи с комбинаторной логикой, является аналогом λ-абстракции. Функции перевода термов λ-исчисления в термы комбинаторной логики исходят именно из этого, когда переводят κ(λxY) в [x].κ(Y).

Но это не одно и то же.

Аналог правила ξ в комбинаторной логике не является допустимым правилом. Из доказуемости в ней равенства термов X = Y не следует доказуемость [x].X = [x].Y. Чтобы это имело место, необходимо добавить еще несколько не слишком естественных комбинаторных аксиом, дополнительно связывающих комбинаторы **K** и **S**, и выбрать другой алгоритм функциональной абстракции. Но это будет уже другая комбинаторная логика с другими свойствами.

Важным отличием комбинаторной логики от λ-исчисления является то, что она более интенсиональна, позволяя более тонко различать функции в зависимости от алгоритма их вычисления.

Как и в случае комбинаторной логики, переход от терма T к замкнутому терму $\lambda x_1 \ldots x_n.T$ является аналогом использования неограниченной аксиомы свертки в наивной теории множеств, но при этом не ведет ни к каким противоречиям.

Удивительным свойством чистого λ-исчисления является то, что в нем можно регулярным образом определить термы, которые могут представлять натуральные числа. Также в нем можно определить термы, которые представляют все эффективные функции на множестве натуральных чисел. И это при том, что никакие сущности в λ-исчислении, в отличие от комбинаторной логики, не постулируются. Постулируется лишь возможность переименовывать переменные и возможность подстановки. Более ничего.

ДЕФИНИЦИАЛЬНАЯ ЛОГИКА

В комбинаторной логике и λ-исчислении доказуемы следующие формулы:

$$([x_1,.., x_n].T)x_1\ldots x_n = T$$

$$(\lambda x_1\ldots x_n.T)x_1\ldots x_n = T$$

По своей логической форме они очень напоминают правило явных определений

$$\mathbf{G}(x_1,\ldots,x_n) =_{def} T$$

где **G** - это новый функциональный символ, вводимый явным определением.

Оказывается, если взять язык комбинаторной логики, исключив из него комбинаторы **K** и **S**, взять в качестве одного из его правил правило введения для любого терма T определения

$$\mathbf{D}x_1\ldots x_n =_{def} T$$

а в качестве другого правила взять обычное логическое правило устранения определений, т.е. замены $\mathbf{D}X_1\ldots X_n$ на $T[T_1/x_1,\ldots,T_n/x_n]$ чисто логическое исчисление, родственное комбинаторной логике и λ-исчислению, в котором также представимы и натуральные числа, и все вычислимые функции. Принципиальное отличие же его заключается в том, что мы уже имеем дело не с функциями и операцией аппликации, а просто с манипуляциями знаками.

Посредством правила введения определений мы получаем возможность посредством введения новой константы **D** запомнить структуру сложного знака T, а с помощью операции устранения определений наполнить эту структуру конкретными знаками. Все операции являются логическими. Мы не постулируем, как это имеет место в случае комбинаторной логики, никаких сущностей **K** и **S**, не занимаемся, как это имеет место в случае λ-исчисления, аксиоматизацией операции *аппликации* (приложения функции к аргументам), а просто оперируем с рядоположенными знаками путем запоминания их структур и делая эти структуры новыми объектами мысли.

Более подробно, с доказательством теорем, я рассматриваю это в своей диссертации.