1 Просто типизированное лямбда-исчисление

Как мы уже убедились из парадокса Карри, лямбда-исчисление - это слишком мощная конструкция. Чтобы ее упорядочить, предпринимались многочисленные попытки. Сперва мы рассмотрим просто типизированное лямбда-исчисление.

Идея восходит к типизации теории множеств. Там, чтобы избежать парадоксов, можно предложить каждому множеству приписать некоторое значение (тип). Например, мы можем приписать пустому множеству тип 0, множеству, состоящему только из пустого - тип 1, и вообще, множеству, состоящему из типов $a_1, a_2, \dots$ тип $Upb_{Ord}\{a_1, a_2, \dots\}$. При этом некоторые

Здесь мы поступим похоже, только элементарных значений (и, соответственно, типов) у нас будет не одно, а счетное количество.

Определение 1.1. Тип — это:

- Элементарный тип — маленькая греческая буква $(\alpha, \beta, \dots)$
- Составной тип. Если τ и σ — некоторые типы, то запись вида $\tau \to \sigma$ — это также некоторый тип.

Существует два основных стиля типизации лямбда-исчисления — по Чёрчу и по Карри.

Теорема 1.1. О слабой нормализации. Пусть некоторый лямбда-терм имеет тип σ. Тогда существует конечная последовательность бета-редукций, приводящая к нормальной форме.

Лемма 1.2. Пусть A и B — лямбда-термы, а x — некоторая переменная. Тогда $h(A[x := B]) \leq max(h(A), h(B),)$.

Доказательство. □

Доказательство. Доказательство теоремы о слабой нормализации.

Мы покажем утверждение теоремы, доказав, что последовательность редукций, при которой редуцируется самый вложенный редекс максимальной степени (такой редекс $R = (\lambda x : \sigma.A^{\tau})B$ выражения $E = \dots R \dots$, что $h(R) = h(E)$, и $h(A) < h(R)$, как и $h(B) < h(R)$), приводит к нормальной форме.

Рассмотрим, что произойдет с количеством редексов высоты $h(R)$ в результате редукции R. Рассмотрим некоторый редекс P в новом выражении $F = \dots A[x := B] \dots$, получающемся из E путем редукции R. Он может:

- находиться целиком вне результата редукции редекса R, тогда он никак не изменяется при данной редукции;
- являться правой частью внешнего редекса $F = \dots (\lambda y : \theta.Q)A[x := B] \dots$. В этом случае, поскольку тип переменной x совпадает с типом терма B, изменения высоты данного редекса не произойдет;
- являться левой частью внешнего редекса $F = \dots A^{\theta}[x := B]Q^{\rho} \dots$. В этом случае неизбежно $A = \lambda y : \rho.C^{\theta}$ для некоторого C, и $h(A[x := B]Q) = h(\rho \to \theta) < h((\lambda x : \sigma.\lambda y : \rho.C)BQ) = h(\sigma \to \rho \to \theta)$, то есть, в этом случае появившийся редекс будет иметь высоту, меньшую $h(R)$;
- находиться целиком внутри результата редукции. Заметим, что из $R : \sigma \to \tau$ следует, что $h(R) = h(\sigma \to \tau) > h(\sigma)$, и по принципу построения последовательности редукций, $h(R) > h(A)$ и $h(R) > h(B)$. Тогда, воспользовавшись леммой, заключаем $h(A[x := B]) \leq max(h(A), h(B), h(\sigma)) < h(R)$.

Таким образом, поскольку сам редекс R будет разрушен, и никаких новых редексов данной или большей высоты не добавится, количество редексов высоты $h(R)$ уменьшится минимум на 1.

Рассмотрим функцию $m(E) = (h(E), n_E(h(E)))$, где $n_E(x)$ — количество редексов высоты x в формуле E. Если $n_E(h(E)) > 1$, то редукция самого вложенного редекса степени $h(E)$ уменьшит $n_E(h(E))$ минимум на 1, если же их остался один — то редукция устранит их совсем, не добавив новых. Значит, мы получим строго убывающую последовательность $m(E)$, ограниченную снизу: она прервется, когда в выражении не останется ни одного редекса. То есть, данная последовательность редукций приведет выражение к нормальной форме.

□

2 О классе функций, определимых в просто типизированном лямбда-исчислении

Определение 2.1. Назовем расширенным полиномом функцию

Лемма 2.1. Если в выражении X^ξ, находящемся в нормальной форме, подтерм T^τ не является свободной переменной выражения T, и $T \neq X$, то всегда найдется такой подтерм S^σ, что $h(\sigma) > h(\tau)$, причем $\sigma = \tau \to \rho$ или $\sigma = \rho \to \tau$.

Доказательство. Рассмотрим подтерм T. Возможны следующие варианты:

1. T — это некоторая переменная x (она обязана быть связанной по условию леммы). То есть T — часть выражения $S = \lambda x : \tau. \dots x \dots$. Тогда $S : \tau \to \rho$, и $h(\tau \to \rho) > h(\tau)$.

2. T — это некоторая абстракция $T = \lambda x : \sigma.P^\pi$. Тогда заметим, что по условию $T \neq X$. Значит, T входит в некоторое более общее выражение — либо в абстракцию $S^{\upsilon \to \tau} = \lambda y : \upsilon.T$, либо в применение $S^{\tau \to \upsilon}T$ (применение вида TA является редексом и потому невозможно).

3. T — это некоторое применение $T = S^\upsilon \to \tau Y$.

□

Теорема 2.2. При фиксированном типе для целых чисел $\nu = (\alpha \to \alpha) \to (\alpha \to \alpha)$ в типизированном исчислении по Чёрчу класс двуместных функций ограничен расширенными полиномами.

Доказательство. Рассмотрим некоторый лямбда-терм $Ra^\nu b^\nu f^{\alpha \to \alpha} : \alpha \to \alpha$, здесь a и b — переменные, содержащие чёрчевские нумералы (аргументы функции), f — первый аргумент чёрчевского нумерала.

Согласно свойству слабой нормализации, данный терм имеет нормальную форму N. Рассмотрим ее. Заметим, что если T^τ — подтерм N, то он обязан соответствовать одному из следующих вариантов:

Тип	Возможный вид терма
ν	a, b
$\alpha \to \alpha$	f, $aS_1^{\alpha \to \alpha}$, $bS_1^{\alpha \to \alpha}$, $\lambda x : \alpha.S_1^{\alpha \to \alpha}(S_2^{\alpha \to \alpha} \dots S_n^{\alpha \to \alpha}(y) \dots)$
α	f, связанная переменная $y : \alpha$, $S_1^{\alpha \to \alpha}(T^\alpha)$

Доказать это можно разбором случаев с использованием индукции и предыдущей леммы, для примера разберем случай $h(\tau) \geq 2$.

- Если $h(\tau) \geq 3$, то $T = a$ или $T = b$. Пусть это не так, и существуют такие P^π, что $P \neq a$, $P \neq b$ и $h(\pi) \geq 3$. Возьмем среди таких P подтерм с типом максимальной глубины. Однако, по лемме в нем неизбежно найдется такой S^σ, что $h(\sigma) > h(\pi)$, что противоречит максимальности $h(\pi)$.

- Если $h(\tau) = 2$, то τ имеет вид либо $\alpha \to (\alpha \to \alpha)$, либо $(\alpha \to \alpha) \to \alpha$. По лемме найдется такой S^σ, что $\sigma = \tau \to \rho$ или $\sigma = \rho \to \tau$. В любом из случаев не найдется такого ρ, что $\nu = \sigma$, то есть $S \neq a$ и $S \neq b$, что невозможно по предыдущему пункту.

Теперь рассмотрим весь терм N. Он имеет тип $\alpha \to \alpha$. Введем обозначение $\overline{m} = \lambda f : \alpha \to \alpha.\lambda x : \alpha.f^m x$. Покажем по индукции, что $N[a := \overline{m}, b := \overline{n}] =_\beta \lambda f x.f^{\overline{E(m,n)}}x$.

Т.о., в выражении возможны 3 только типа элементарных формул, типов α, $\alpha \to \alpha$ и $(\alpha \to \alpha) \to (\alpha \to \alpha)$.

Про вариант $(\alpha \to \alpha) \to (\alpha \to \alpha)$ мы уже разобрались, теперь рассмотрим $\alpha \to \alpha$.

Это может быть либо f, либо $aS^{\alpha \to \alpha}$, либо $bS^{\alpha \to \alpha}$, либо их комбинация: $\lambda x : \alpha.S_1^{\alpha \to \alpha}(S_2^{\alpha \to \alpha} \dots S_n^{\alpha \to \alpha}(z$

□

2.1 Типизируемость

3 Синтез типа и обитаемость типа

Прежде чем перейти к более сложным типовым системам, нам осталось ответить на важный вопрос о наличии эффективных процедур, позволяющих определить, существует ли лямбда-выражение, имеющее некоторый тип в некотором контексте: $\Gamma \vdash ? : \sigma$

Определение 3.1. Мы будем называть некоторый тип σ в контексте Γ *обитаемым*, если найдется такое выражение M, что $\Gamma \vdash M : \sigma$.

Теорема 3.1. Задача определения обитаемости типа — разрешима.

Доказательство. Для доказательства предоставим разрешающий алгоритм. □

4 Упорядоченные пары и алгебраические типы данных

Сперва попробуем расширить понятие типа экстенсивно: через добавление новых связок, не изменяя порядка исчисления. Естественные кандидаты здесь — конъюнкция и дизъюнкция, для которых изоморфизм Карри-Ховарда предлагает следующие аналоги:

Конструкция	Связка	Операции
Упорядоченная пара	$\alpha\&\beta$	$\pi_1 : \alpha\&\beta \to \alpha$
		$\pi_2 : \alpha\&\beta \to \beta$
		$\langle \alpha, \beta \rangle : \alpha \to \beta \to \alpha\&\beta$
Алгебраический тип	$\alpha \vee \beta$	$in_1 : \alpha \to \alpha \vee \beta$
		$in_2 : \beta \to \alpha \vee \beta$
		$case : (\alpha \to \gamma) \to (\beta \to \gamma) \to \alpha \vee \beta \to \gamma$

5 Исчисление 1-го порядка

Более радикальный путь усиления теории — рассмотрение исчислений 1-го и высших порядков. Подробно на теориях 1-го порядка мы останавливаться не будем, единственное, отметим, что такая теория будет требовать определение выражений двух сортов: предметных и логических. Аналогом с точки зрения изоморфизма Карри-Ховарда для логических значений будут типы, а предметными выражениями могут быть любые выражения над не-типовыми значениями: например, над строками, целыми числами и т.п.

Аналог предиката в данном случае — это функция, отображающая значение предметного множества в тип. Такой тип, зависящий от предметной переменной, называется *зависимым*.

Самый, видимо, известный пример подобного — шаблоны в C++. Если мы будем рассматривать значения, представимые в откомпилированном коде, в качестве элементов предметного множества, типы — как значения пропозициональных переменных, а шаблоны, параметризованные элементарными значениями — как предикаты, то мы как раз получим исчисление с зависимыми типами.

Например, шаблону

```
template <int a>
struct X { int v[a]; };
```

мы могли бы сопоставить тип: $X : \forall a$.

Хотя, конечно, система типов в C++ значительно сложнее и в приведенное исчисление 1-го порядка не помещается в точности: например, мы проигнорировали в формализации, что аргументы шаблонов имеют тип (и получается, что кванторы ограничены некоторым подмножеством предметного множества).

6 Исчисление 2-го порядка

Те же шаблоны из C++ позволяют задавать и более сильные операции: отображения из типов в типы. Такие типы (зависящие от других типов) в логике имеют аналогом предикаты, зависящие от других логических значений. Это уже — исчисление предикатов 2-го порядка. Изучение такого исчисления и соответствующей ему системы типов и будет нашей ближайшей целью.

6.1 Импликационный фрагмент интуиционистского исчисления второго порядка

6.2 Система F

Определение 6.1.