Функциональное и логическое программирование

Лекция 2. Лямбда исчисление

Учебные вопросы

- 2.1 Эквивалентность лямбда выражений. Экстенсиональность
- 2.2 Редукция
- 2.3 Теорема Чёрча-Россера
- 2.4 Комбинаторы

Эквивалентность лямбда выражений. Экстенсиональность

$$\frac{s \xrightarrow[\alpha]{} t \text{ или } s \xrightarrow[\beta]{} t \text{ или } s \xrightarrow[\eta]{} t}{s = t}$$

$$\frac{}{t = t}$$

$$\frac{s = t}{t = s}$$

$$\frac{s = t \text{ и } t = u}{s = u}$$

$$\frac{s = t}{s\ u = t\ u}$$

$$\frac{s = t}{u\ s = u\ t}$$

$$\frac{s = t}{\lambda x.\, s = \lambda x.\, t}$$

Эквивалентность лямбда выражений. Экстенсиональность

Мы уже упоминали ранее, что η-преобразование воплощает принцип *экстенсиональности*. В рамках общепринятых философских понятий два свойства называются *экстенсионально эквивалентными* (либо *коэкстенсивными*), если этими свойствами обладают в точности одни и те же объекты. В теории множеств принята аксиома экстенсиональности, согласно которой два множества совпадают, если они состоят из одних и тех же элементов. Аналогично, будем говорить, что две функции эквивалентны, если области их определения совпадают, а значения функций для всевозможных аргументов также одинаковы.

Введение η-преобразования делает наше понятие λ-эквивалентности экстенсиональным. В самом деле, пусть $f\ x$ и $g\ x$ равны для произвольного значения x; в частности, $f\ y = g\ y$, где переменная y выбирается так, чтобы она не была свободной как в f, так и в g. Согласно последнему из приведённых выше правил эквивалентности, $\lambda y.\ f\ y = \lambda y.\ g\ y$. Применив дважды η-преобразование, получаем, что $f = g$. С другой стороны, из экстенсиональности следует, что всевозможные η-преобразования не нарушают эквивалентности, поскольку согласно правилу β-редукции $(\lambda x.t\ x)\ y = t\ y$ для произвольного y, если переменная x не является свободной в терме t. На этом мы завершаем обсуждение сущности η-преобразования и его влияния на теорию в целом, чтобы уделить больше внимания более перспективному с точки зрения вычислимости β-преобразованию.

Редукция

Отношение λ-эквивалентности, как и следовало ожидать, является симметричным. Оно достаточно хорошо соответствует интуитивному понятию эквивалентности λ-термов, но с алгоритмической точки зрения более интересен его несимметричный аналог. Определим отношение *редукции* $\longrightarrow$ следующим образом:

$$\frac{s \underset{\alpha}{\longrightarrow} t \text{ или } s \underset{\beta}{\longrightarrow} t \text{ или } s \underset{\eta}{\longrightarrow} t}{s \longrightarrow t}$$

$$\frac{}{t \longrightarrow t}$$

$$\frac{s \longrightarrow t \text{ и } t \longrightarrow u}{s \longrightarrow u}$$

$$\frac{s \longrightarrow t}{s\ u \longrightarrow t\ u}$$

$$\frac{s \longrightarrow t}{u\ s \longrightarrow u\ t}$$

$$\frac{s \longrightarrow t}{\lambda x.\, s \longrightarrow \lambda x.\, t}$$

Редукция

В действительности слово «редукция» (в частности, термин β-редукция, которым иногда называют β-преобразования) не отражает точно сути происходящего, поскольку в процессе редукции терм может увеличиваться, например:

$$\begin{aligned}
(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x) &\longrightarrow (\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x) \\
&\longrightarrow (\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x)\,(\lambda x.\, x\, x\, x) \\
&\longrightarrow \ldots
\end{aligned}$$

Однако, несмотря на это редукция имеет прямое отношение к процедуре вычисления терма, в ходе которой последовательно вычисляются комбинации вида $f(x)$, где f — λ-абстракция. Если на некотором этапе оказывается, что не могут быть применены никакие правила редукции, кроме α-преобразований, то говорят, что терм имеет *нормальную форму*.

Редукция

$$
\begin{aligned}
& (\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \\
\longrightarrow\ & (\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \\
\longrightarrow\ & (\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \\
\longrightarrow\ & \cdots
\end{aligned}
$$

$$(\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \longrightarrow y$$

Редукция

Теорема 2.2 *Если справедливо* $s \longrightarrow t$, *где терм* t *имеет нормальную форму, то последовательность редукций, которая начинается с терма* s *и состоит в применении правил редукции к самому левому редексу, всегда завершается и приводит к терму в нормальной форме.*

Редукция

Пример 3.1. Рассмотрим терм $(\lambda x.\, xx)(\lambda x.\, xx)$, который сам является β-редексом, и применим к нему редукционное преобразование. $(\lambda x.\, xx)(\lambda x.\, xx) \to_\beta (\lambda x.\, xx)(\lambda x.\, xx) \to_\beta \ldots$. Имеем дело с бесконечной редукционной цепочкой.

Пример 3.2. Рассмотрим терм $\big(\lambda x.\, ((\lambda y.\, xy)u)\big)(\lambda v.\, v)$, в котором имеются два β-редекса: во-первых, сам терм является β-редексом и, во-вторых, β-редексом является его подтерм $((\lambda y.xy)u)$. Выбирая для свертки разные редексы, получим разные редукционные цепочки.

1. $(\lambda x.((\lambda y.xy)u))(\lambda v.v) \to_\beta (\lambda y.(\lambda v.v)y)u \to_\beta (\lambda v.v)u \to_\beta u$
2. $(\lambda x.((\lambda y.xy)u))(\lambda v.v) \to_\beta (\lambda x.(xu))(\lambda v.v) \to_\beta (\lambda v.v)u \to_\beta u$

Пример 3.3. Рассмотрим терм $(\lambda xy.y)\big((\lambda z.zz)(\lambda z.zz)\big)$. Здесь два редекса: внутренний — $(\lambda z.zz)(\lambda z.zz)$ и внешний — $\big(\lambda x.(\lambda y.y)\big)\big((\lambda z.zz)(\lambda z.zz)\big)$.
Выбрав внутренний редекс, получим его же. Таким образом, выбирая каждый раз внутренний редекс, получим бесконечную редукционную цепочку. Выбрав внешний редекс для свертывания, получим за один шаг терм без редексов. $\big(\lambda x.(\lambda y.y)\big)\big((\lambda z.zz)(\lambda z.zz)\big) \to_\beta (\lambda y.y)$

Редукция

Заметим, что во втором случае аргумент нам редуцировать не пришлось, реализовалось так называемое ленивое вычисление.

Аппликативный порядок редукций (АПР) предписывает всегда выбирать самый левый из внутренних редексов.

Нормальный порядок редукций (НПР) предписывает всегда выбирать самый левый из внешних редексов.

Напомним, что в рассмотренном выше примере 3 первый способ выбора редекса привел к зацикливанию, а второй за один шаг завершил вычисления.

Терм $\lambda x.(\lambda y.y)$ — классический пример функции, которая отбрасывает свой аргумент. Стратегия НПР откладывает вычисление аргумента до тех пор, пока он действительно не потребуется.

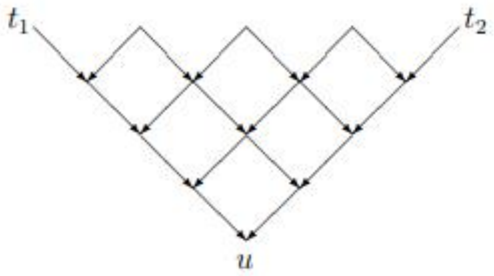
Теорема Чёрча-Россера

Теорема 2.3 *Если* $t \longrightarrow s_1$ *и* $t \longrightarrow s_2$, *то существует терм* u *такой, что* $s_1 \longrightarrow u$ *и* $s_2 \longrightarrow u$.

Теорема Чёрча-Россера

Corollary 2.4 *Если* $t_1 = t_2$ *то найдётся терм* u *такой, что* $t_1 \longrightarrow u$ *и* $t_2 \longrightarrow u$.

Доказательство: *Легко показать (при помощи структурной индукции), что отношение λ-равенства = представляет собой симметричное транзитивное замыкание отношения редукции. Дальнейшее следует по индукции согласно свойствам симметричного транзитивного замыкания. Приведённая ниже диаграмма может показаться читателям, не склонным к формальным построениям, более доходчивой:*



Мы полагаем, что $t_1 = t_2$, *т. е. существует некоторая последовательность редукций в обеих направлениях (зигзагообразная линия в верхней части рисунка), которая их объединяет. Теорема Чёрча-Россера позволяет нам заполнить недостающие участки на краях диаграммы, после чего требуемый результат достигается композицией этих редукций.*

Теорема Чёрча-Россера

Corollary 2.5 *Если* $t = t_1$ *и* $t = t_2$, *причём* t_1 *и* t_2 *имеют нормальную форму, то* $t_1 \equiv_\alpha t_2$, *т. е.* t_1 *и* t_2 *равны с точностью до* α*-преобразований.*

Доказательство: *Согласно изложенному выше, найдётся некоторый терм* u *такой, что* $t_1 \longrightarrow u$ *и* $t_2 \longrightarrow u$. *Но так как* t_1 *и* t_2 *уже имеют нормальную форму, последовательность редукций, приводящая к терму* u, *может состоять лишь из* α*-преобразований.* $\square$

Комбинаторы

$$
\begin{aligned}
I &= \lambda x.\, x \\
K &= \lambda x\, y.\, x \\
S &= \lambda f\, g\, x.\, (f\ x)(g\ x)
\end{aligned}
$$

Лемма 2.6 *Для произвольного λ-терма t, не содержащего λ-абстракций, найдётся терм u, который также не содержит λ-абстракций и представляет собой композицию S, K, I и переменных, причём $FV(u) = FV(t) - \{x\}$ и $u = \lambda x.\, t$, т. е. терм u λ-равен $\lambda x.\, t$.*

Доказательство: *Применим к терму t структурную индукцию. Согласно условию, он не может быть абстракцией, поэтому нам требуется рассмотреть лишь три случая.*

- *Если t представляет собой переменную, возможны два случая, из которых непосредственно следует требуемый вывод: при $t = x$ мы получаем $\lambda x.\, x = I$, иначе, например, при $t = y$, $\lambda x.\, y = K\ y$.*

- *Если t — константа c, то $\lambda x.\, c = K\ c$.*

Комбинаторы

- *Если* t *представляет собой комбинацию термов, например,* $s\ u$, *то согласно индуктивному предположению найдутся термы* s' *и* u', *которые не содержат* λ-*абстракций и для которых справедливы равенства* $s' = \lambda x.\, s$ *и* $u' = \lambda x.\, u$. *Из этого можно сделать вывод, что* $S\ s'\ u'$ *является искомым выражением. В самом деле,*

$$\begin{aligned} S\ s'\ u'\ x &= S\ (\lambda x.\, s)\ (\lambda x.\, u)\ x \\ &= ((\lambda x.\, s)\ x)((\lambda x.\, u)\ x) \\ &= s\ u \\ &= t \end{aligned}$$

Таким образом, применив η-*преобразование, мы получаем* $S\ \ s'\ \ u' = \lambda x.\, S\ \ s'\ \ u'\ \ x = \lambda x.\, t$, *поскольку согласно индуктивному предположению переменная* x *не является свободной в термах* s' *либо* u'.

Комбинаторы

Это примечательное утверждение может быть даже усилено, поскольку комбинатор I выражается через S и K. Отметим, что для произвольного A

$$\begin{aligned} S\ K\ A\ x &= (K\ x)(A\ x) \\ &= (\lambda y.\, x)(A\ x) \\ &= x \end{aligned}$$

Отсюда, применив η-преобразование, получаем, что $I = S\ K\ A$ для любых A. Однако, по причинам, которые станут яснее после знакомства с понятием типа, наиболее удобно положить $A = K$. Таким образом, $I = S\ K\ K$, что даёт нам возможность устранить все вхождения I в комбинаторные выражения.

Заметим, что приведённые выше доказательства имеют конструктивный характер, поскольку предлагают конкретные процедуры получения по заданному терму эквивалентного комбинаторного выражения. Процесс его построения идёт в направлении снизу вверх, и для каждой λ-абстракции, которая по построению имеет тело, свободное от λ-абстракций, применяются сверху вниз преобразования, изложенные в лемме.

Комбинаторы

Несмотря на то, что мы рассматриваем комбинаторы как некоторые термы λ-исчисления, на их основе можно сформулировать независимую теорию. Её построение начинается с определения формальных правил конструирования выражений, в которые не входит λ-абстракция, но входят комбинаторы. Далее вместо α, β и η-преобразований вводятся правила преобразования для выражений, включающих комбинаторы, например, $K\ x\ y \longrightarrow x$. Такая теория будет иметь множество аналогий в традиционном λ-исчислении, например, теорема Чёрча-Россера оказывается справедливой и для приведённого выше определения редукции. Кроме того, полностью устраняются сложности со связыванием переменных. Тем не менее, мы считаем полученный формализм не слишком интуитивным, поскольку комбинаторные выражения нередко бывают весьма неясными.

Комбинаторы

Определение	Название	Характеристика
$\mathbf{I} \equiv \lambda x.\, x$	тождественный	$\mathbf{I}X \to_\beta X$
$\mathbf{B} \equiv \lambda xyz.\, x(yz)$	композитор	$\mathbf{B}FGX \to_\beta F(GX)$
$\mathbf{C} \equiv \lambda xyz.\, xzy$	пермутатор	$\mathbf{C}FXY \to_\beta FYX$
$\mathbf{K} \equiv \lambda xy.\, x$	канцелятор	$\mathbf{K}XY \to_\beta X$
$\mathbf{S} \equiv \lambda xyz.\, xz(yz)$	коннектор	$\mathbf{S}FGX \to_\beta FX(GX)$
$\mathbf{W} \equiv \lambda xy.xyy$	дупликатор	$\mathbf{W}FX \to_\beta FXX$

Комбинаторы

Пример 4.1.
Приведём комбинатор $\mathbf{B}(\mathbf{SB}(\mathbf{KI}))(\mathbf{KI})(\mathbf{SB}(\mathbf{SB}(\mathbf{KI})))$ к β-нормальной форме:

$$
\begin{aligned}
&\mathbf{B}(\mathbf{SB}(\mathbf{KI}))(\mathbf{KI})(\mathbf{SB}(\mathbf{SB}(\mathbf{KI}))) \to_\beta \mathbf{SB}(\mathbf{KI})(\mathbf{KI}(\mathbf{SB}(\mathbf{SB}(\mathbf{KI}))))\\
&\to_\beta \mathbf{B}(\mathbf{KI}(\mathbf{SB}(\mathbf{SB}(\mathbf{KI}))))\Big(\mathbf{KI}(\mathbf{KI}(\mathbf{SB}(\mathbf{SB}(\mathbf{KI}))))\Big)\\
&\to_\beta \mathbf{BI}\Big(\mathbf{KI}(\mathbf{KI}(\mathbf{SB}(\mathbf{SB}(\mathbf{KI}))))\Big) \to_\beta \mathbf{BII}\\
&\equiv (\lambda xyz.\, x(yz))(\lambda x.\, x)(\lambda x.\, x) \to_\beta \lambda z.\, (\lambda x.\, x)((\lambda x.\, x)z)\\
&\to_\beta \lambda z.\, (\lambda x.\, x)z \to_\beta \lambda z.\, z \equiv \mathbf{I}
\end{aligned}
$$