ФИВТ МФТИ, весна 2014. Специальности ПМФ и ПМИ.

Краткие заметки по курсу
Математическая логика и теория алгоритмов.
Часть третья: лямбда-исчисление.
(А.Е. Ромащенко).

Краткий конспект для студентов, слушавших лекции курса и посещавших семинары на факультете ИВТ Физтеха.

1 Введение и мотивировки

На лекции мы обсуждали

- *прямую польскую нотацию* для логических, арифметических и алгебраических выражений;
- *лямбда-обозначение* для описания функций;
- *каррирование (карринг)* как способ описания функции нескольких переменных.

2 Основные определения

Мы дали определение

- λ-терма,
- области действия квантора в λ-терме,
- свободные и связанные переменные,
- комбинаторы (замкнутые λ-термы).

Мы договорились о стандартном правиле сокращения записи: мы обычно опускаем внешние скобки λ-терма; кроме того, мы пишем

- $A\,B\,C$ вместо $(A\,B)C$,
- $\lambda x.AB$ вместо $\lambda x.(AB)$ и
- $\lambda xy.A$ вместо $\lambda x.(\lambda y.A)$.

Далее мы ввели основные операции на λ-термах:

- α-конверсию,
- β-редукцию,
- η-редукцию.

Каждое из этих трёх преобразований определялось индукцивно (индукция по построению терма).

Мы договорились называть *нормальной формой* такой λ-терм, к которому нельзя применить ни одну из редукций (даже если предварительно применить одну или несколько α-конверсий).

Мы определили *равенство* λ-термов: термы A и B считаются равными, если существует такая последовательность λ-термов $C_0, C_1, \ldots, C_n$, где терм C_0 совпадает с A, терм C_n совпадает с B, и для каждого $i = 0, \ldots, n-1$ либо терм C_i можно преобразовать в C_{i+1} с помощью α-конверсий, β-редукций и η-редукций, либо, наоборот, терм C_{i+1} можно преобразовать в C_i с помощью α-конверсий, β-редукций и η-редукций. Другими словами, термы A и B считаются равными, если A можно преобразовать в B, совершив последовательность операций, каждая их которых является либо α-конверсией, либо β- или η-редукцией, либо преобразованием, обратным к β- или η-редукции.

3 О приведении к нормальной форме

Некоторые λ-термы сами не являются нормальными формами, однако их можно *привести к нормальной форме* (превратить в терм, являющийся нормальной формой, применив последовательность α-конверсий, β-редукций и η-редукций). Бывают и термы, которые невозможно привести к нормальной форме.

Теорема Чёрча–Россера. Если λ-терм A с помощью последовательности α-конверсий, β-редукций и η-редукций можно преобразовать в терм B и в терм C, то существует такой терм D, что B и C в свою очередь можно преобразовать в D с помощью α-конверсий, β-редукций и η-редукций.
[без доказательства]

Следствие 1. Если λ-терм A равен некоторому λ-терму B, являющемуся нормальной формой, то A можно преобразовать в B с помощью последовательности α-конверсий, β-редукций и η-редукций (не применяя операции, обратные к редукциям).

Следствие 2. Если λ-терм A равен двум λ-термам B и C в нормальной форме, то B и C можно преобразовать друг в друга последовательностью α-конверсий. (Другими словами, нормальная форма для λ-терма A определена однозначно с точностью до переименования связанных переменных.)

4 Кодирование логических операций и структуры данных в λ-исчислении

Мы договорились кодировать *истину* и *ложь* комбинаторами $\lambda xy.x$ и $\lambda xy.y$ соответственно. Мы нашли комбинаторы, представляющие в λ-исчислении операции *конъюнкции*, *дизъюнкции* и *отрицания*.

Мы условились кодировать *упорядоченную пару* комбинаторов $\langle A, B\rangle$ в виде комбинатора $\lambda f.fAB$. Мы построили комбинатор **Pair** такой, что

$$\mathbf{Pair}\,A\,B = \lambda f.fAB.$$

Также мы построили комбинаторы **First** и **Second** такие, что

$$\mathbf{First}\,(\lambda f.fAB) = A$$

и

$$\mathbf{Second}\,(\lambda f.fAB) = B.$$

5 Арифметика в λ-исчислении

Мы кодируем в λ-исчислении натуральные числа с помощью *нумералов Чёрча*:

$$\underline{0} = \lambda fx.x,\ \underline{1} = \lambda fx.fx,\ \underline{2} = \lambda fx.f(fx),\ \underline{3} = \lambda fx.f(f(f(x))), \ldots$$

Мы построили комбинаторы, представляющие в λ-исчислении функции *прибавления единицы*, *сложения*, *умножения*, *возведения в степень*.

С помощью «трюка Клини» мы построили комбинатор представляющий функцию *вычитания единицы*. Из неё мы получили комбиантор, представляющий функцию *модифицированного вычитания*

$$\mathsf{sub}\,n\,m = \begin{cases} n-m, & \text{если } n \geq m \\ 0, & \text{иначе.} \end{cases}$$

6 Рекурсивное программирование

Мы построили комбинатор Y (*комбинатор неподвижной точки*) такой, что для любого комбинатора F для уравнения

$$X = F\,X$$

терм $X = Y\,F$ является решением. Другими словами, для любого F выполнено равенство

$$Y\,F = F(Y\,F).$$

На лекциях и семинарах с помощью комбинатора неподвижной точки мы научились строить комбинаторы, представляющие различные арифметические функции с «рекурсивным» определением — факториал, n-ое число Фибоначчи, и т.п.

7 Представление произвольных функций в λ-исчислении

Говорят, что комбинатор F представляет в λ-исчислении (частичную) функцию $f : \mathbb{N}^k \to \mathbb{N}$, если для любых чисел $n_1, \ldots, n_k$ комбинатор $(F\, \underline{n_1}\, \underline{n_2} \ldots \underline{n_k})$ равен нумералу Чёрча для числа $f(n_1, \ldots, n_k)$, если значение $f(n_1, \ldots, n_k)$ определено, и не имеет нормальной формы в противном случае.

Теорема. Все вычислимые функции (и только они) представимы в λ-исчислении. [без доказательства]

Список литературы

[1] Барендрегт Х. Лямбда-исчисление. Его синтаксис и семантика. Пер. с англ. М.: Мир, 1985.

[2] eter Selinger, Lecture notes on the lambda calculus. `http://arxiv.org/abs/0804.3434`

[3] chim Jung, A short introduction to the Lambda Calculus. `http://www.cs.bham.ac.uk/ axj/pub/papers/lambda-calculus.pdf`

[4] enk Barendregt, Erik Barendsen, Introduction to Lambda Calculus. `https://files.nyu.edu/cb125/public/Lambda/barendregt.94.pdf`

[5] awrence Paulson. Foundations of Functional Programming. `http://www.cl.cam.ac.uk/ lp15/papers/Notes/Founds-FP.pdf`