λ-исчисление

λ-термы образуются из переменных $(x, y, z, \dots)$ с помощью двух операций: *применения:* (uv) и *абстракции:* $\lambda x.u$. При записи λ-термов мы придерживаемся следующих соглашений: скобки ассоциируются налево ("uvw" читается как $(uv)w$, а не как $u(vw)$); применение связывает сильнее, чем абстракция ("$\lambda x.uv$" значит $\lambda x.(uv)$, а не $(\lambda x.u)v$).

Абстракция $\lambda x.u$ *связывает* переменную x (в том же смысле, как в классической логике предикатов переменную связывают кванторы $\forall x$ и $\exists x$). Несвязанные переменные называются *свободными.* Связанные переменные можно переименовывать; термы, отличающиеся только именами связанных переменных, назовём α-равными ($u_1 =_\alpha u_2$), и такие термы мы будем отождествлять. Термы, не содержащие свободных переменных (т.е. термы, в которых все переменные связаны с помощью λ), будем называть *замкнутыми.*

Неформально термы следует мыслить как обозначения функциональных выражений. При этом применение (fu) следует понимать как действие функции на аргумент — $f(u)$. Абстракция же позволяет задать функцию с помощью выражения-терма: например, если в языке есть операциия сложения и умножения, можно записать функцию $\lambda x.(x \cdot x + x)$, т.е. $f \colon x \mapsto (x \cdot x + x)$. (Сама операция сложения понимается как функция двух аргументов, принимающая их последовательно: после принятия первого аргумента возвращается также функция. Формально, сложение надо писать в префиксном виде: «$(+2)2$» вместо «$2 + 2$». При этом, «$+2$» — это функция, прибавляющая 2 к своему аргументу. То же для умножения.)

Процесс вычисления значения λ-терма называется *β-редукцией:* $(\lambda x.u)v \to_\beta u[x := v]$. Здесь запись $u[x := v]$ означает результат замены всех свободных вхождений x в терм u на v; при этом, чтобы избежать коллизий, связанные переменные терма u при необходимости переименовываются. β-редукцию можно применять к произвольному подтерму данного терма, если этот подтерм имеет вид $(\lambda x.u)v$.

Два терма называются *β-эквивалентными* ($u =_\beta v$), если они приводятся с помощью β-редукций к одному и тому же терму w.

Отношение $=_\beta$ можно определить и по-другому, с помощью исчисления. Базовые эквивалентности суть следующие: $u_1 =_\beta u_2$, если они α-эквивалентны (в частности, $u =_\beta u$), и $(\lambda x.u)v =_\beta u[x := v]$.

Правила:

$$\frac{u_1 =_\beta u_2}{u_1 v =_\beta u_2 v} \qquad \frac{u_1 =_\beta u_2}{v u_1 =_\beta v u_2} \qquad \frac{u_1 =_\beta u_2}{\lambda x.u_1 =_\beta \lambda x.u_2} \qquad \frac{u_1 =_\beta u_2 \quad u_2 =_\beta u_3}{u_1 =_\beta u_3}$$

Правила читаются следующим образом: если мы уже установили эквивалентность(-сти) над чертой, то при помощи данного правила мы можем утверждать эквивалентность под чертой.

1. Докажите, что если $u \to_\beta v$, то множество свободных переменных терма u содержит множество свободных переменных терма v. Бывает ли это включение строгим?

Некоторым λ-термам можно приписать *типы.* Множество типов строится из *базовых типов* $p, q, r, \dots$ с помощью операции $\to$. Интуитивно типы можно воспринимать как множества. При этом $A \to B$ состоит из *отображений (функций)* из A в B. Эту интуицию можно формализовать с помощью теоремы о полноте относительно теоретико-множественной интерпретации.

Приписывание типов начинается с того, что каждой переменной присваивается некоторый тип (не обязательно базовый). При этом у всех вхождений переменной в терм тип будет один и тот же. Далее типы сложных термов определяются индуктивно: если x — переменная типа A, а терму u приписан тип B, то терму $\lambda x.u$ приписывается тип $A \to B$; если термам v и u приписаны типы $A \to B$ и A соответственно, то терму (vu) приписывается тип B. Если же типы термов v и u не согласованы так, как указано выше, терм (vu) остаётся без типа.

Как видно из определения, тип терма (и, в частности, приписан ли терму вообще какой-то тип) зависит от того, какие типы присвоены переменным. Терм u называется *типизуемым,* если существует такое присваивание типов переменным, что при нём u приписывается некий тип. Бывают нетипизуемые λ-термы, например, (xx).

2. Существуют ли такие замкнутые λ-термы u и v, что $u \to_\beta v$, при этом терм v типизуем, а u — нет?

3. Постройте замкнутые λ-термы следующих типов: **а)** $(p \to q) \to ((q \to r) \to (p \to r))$; **б)** $(p \to q) \to ((r \to p) \to (r \to q))$; **в)** $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$; **г)** $(p \to (q \to r)) \to (q \to (p \to r))$; **д)** $p \to ((p \to q) \to q)$; **е)** $((((p \to q) \to p) \to p) \to q) \to q$.

Нетипизуемые λ-термы доставляют ряд забавных примеров. Например, терм $(\lambda x.xx)(\lambda x.xx)$ редуцируется к самому себе (и, тем самым, процесс его редукций бесконечен). Терм $\mathbf{y} = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ является *комбинатором неподвижной точки:* для любого λ-терма g имеет место эквивалентность $g(\mathbf{y}g) =_\beta \mathbf{y}g$. Этот терм называется *комбинатором Карри.*

4. Докажите, что *комбинатор Тьюринга* $\mathbf{t} = (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$ также является комбинатором неподвижной точки.

5. **а)** Постройте такой λ-терм s, что $stu =_\beta ut$ для любых t и u. **б)** Постройте такой λ-терм s, что $st =_\beta ss$ для любого t.

6. Существует ли λ-терм v, для которого $\lambda x.v =_\beta v$, где x — переменная, не входящая в v?

7. Пусть $\mathbf{k} = \lambda x.\lambda y.x$ и $\mathbf{i} = \lambda z.z$. Добавим к исчислению, задающему отношение $=_\beta$, ещё одну базовую эквивалентность: $\mathbf{k} = \mathbf{i}$. Докажите, что полученная система будет *противоречивой,* т. е. для двух произвольных термов u и v в ней будет доказуема их эквивалентность.

Подсказка. Докажите, что в этой системе всякий терм будет эквивалентен комбинатору $\mathbf{i}$.