**Неподвижная точка**

Неподвижной точкой функции называется такой аргумент для которого . Т.е. значение функции в точке снова равно .

**Теорема:** *о неподвижной точке.* В λ-исчислении каждый терм имеет неподвижную точку, т.е. существует такое , что .

Доказательство: рассмотрим следующий комбинатор: Y = и применим его к :

Заметим, что выражение в скобках, в самой правой части равенства выше,

Отсюда:

,

А значит, читая равенство в обратном направлении, получим:

,

т.е. – неподвижная точка для . Таким образом, существование комбинатор доказывает, что для каждого терма можно найти неподвижную точку .

Практической пользой теоремы о неподвижной точке является возможность введения при помощи комбинатора рекурсивный определений.

Рекурсивное определение объекта выглядит следующим образом:

Т.е. при определении объекта в теле определения встречается сам этот объект. Наиболее часто в программировании рекурсия используется для определения рекурсивный функций (рекурсивными могут быть не только функции, см. упр. 2), которые являются альтернативой циклам при определении итеративных алгоритмов.

У объектов (термов) в λ-исчислении нет имён, поэтому возникает вопрос, как при помощи неё моделировать рекурсивные функции? Ответ на этот вопрос даёт следующее следствие из теоремы о неподвижной точке.

Следствие: *про* рекурсивные *определения*. Пусть где , тогда .

Доказательство: из данного видим, что – неподвижная точка для . Значит – допустимое решение уравнения , относительно неизвестного объекта , т.е. .

Следствие теоремы о неподвижной точке даёт нам инструмент для определения всякого . Для этого достаточно выполнить следующую абстракцию:

здесь очевидно, что , т.к. все свободные вхождения в были абстрагированы и заменены на связанную переменную . Поэтому выполняются условия следствия из теоремы о неподвижной точке и следовательно:

)

Рассмотрим пример:

Функция факториала:

factorial n = if (n == 0) then 1 else n \* factorial (n - 1)

Абстрагируем в правом части определения factorial выражения:

factorial = (λF. λn. if (n == 0) then 1 else n \* F(n - 1))factorial

По теореме о неподвижной точке получаем:

factorial = Y (λF. λn. if (n == 0) then 1 else n \* F(n - 1))

Посмотрим работу полученного определения на практике:

factorial 3 =

= Y (λF. λn. if (n == 0) then 1 else n \* F(n - 1)) 3 =

= Y E 3 = E (Y E) 3 =

= (λF. λn. if (n == 0) then 1 else n \* F(n - 1))**(Y E)** 3 =

= (λn. if (n == 0) then 1 else n \* **(Y E)** (n - 1)) **3** =

= if (**3** == 0) then 1 else **3** \* (Y E) (**3** - 1) =

= if **false** then 1 else 3 \* (Y E) (3 - 1) =

= 3 \* (Y E) (**3 - 1**) =

= 3 \* (Y E) 2 =

= 3 \* (E (Y E) 2) = … (аналогично предыдущему шагу) …

= 3 \* (if 2 == 0 then 1 else 2 \* (Y E) (n - 1)) =

= 3 \* (2 \* (Y E)(n - 1)) = …

= 3 \* 2 \* 1 \* (if 0 == 0 then 1 else 0 \* (Y E) (0 - 1)) =

= 3 \* 2 \* 1 \* 1 = 6

Получили действительно факториал для 3.

**Упражнения:**

1. Решите следующие уравнения (найдите g)
   1. g = (M N) g, где g ∉ FV(M N).
   2. g = (λx. x M) g, где g ∉ FV(M).
   3. g = λx. x M g, где g ∉ FV(M).
   4. g = λx. λy.x y (g x)
2. Будем понимать запись a1:a2:…:an:nil как список [a1, a2, …, an]. Здесь nil – пустой список. Пусть дано определение бесконечного списка f = 1:f. Дайте не рекурсивное определение бесконечного списка f.
3. Предполагая наличие в языке: чисел, λ-абстракций, условных выражений и логический значений, операций над числами (сложение, умножение, вычитание сравнение и т.д.), определите функцию fib n, которая вычисляет n-ое число Фибоначчи. Дайте не рекурсивное определение для функции fib.
4. Пользуясь полученным на предыдущем шаге определением, вычислите fib 4.