3.18 (7) Построение комбинатора, возвращающего n-е простое число (для нумералов Чёрча).

Замечание: Выражение Sub см. в билете 3.15. Построение Mod вынесено в отдельный билет, который мы не взяли, но понимать как мы его построили наверное стоит.

1. $Inc = \lambda nfx.f(nfx)$ $Inc\,\overline{n} = (\lambda nfx.f(nfx))\overline{n} = \lambda fx.f(\overline{n}fx) =$
   $= \lambda fx.f((\lambda gy.\underbrace{g(g(\ldots(g}_{n \text{ раз}}y))\ldots)fx) = \lambda fx.f(\underbrace{f(f(\ldots(f}_{n \text{ раз}}x))\ldots) = \overline{n+1}$

2. $False = \lambda xy.y$; $True = \lambda xy.x$

   $Not = \lambda p.p\,False\,True$ (если p, то выводим $False$, иначе $True$)

   $And = \lambda pq.pqp$ (если p, то выводим q, иначе - p)

3. $IsZero = \lambda n.n(\lambda x.False)True$

   $IsZero\,\overline{0} = \overline{0}(\lambda x.False)True = (\lambda fx.x)(\lambda x.False)True = True$

   $IsZero\,\overline{n+1} = (\lambda fx.f(\ldots))(\lambda x.False)True = (\lambda x.False)(\ldots) = False$

4. $GE = \lambda mn.IsZero(Sub\,n\,m)$ ($\geq$)

   $LT = \lambda mn.Not(GE\,m\,n)$ (less then, то есть $<$)

   $IsEqual = \lambda mn.And(GE\,m\,n)(GE\,n\,m)$

5. $Modfn = \lambda fmn.(LT\,m\,n)m(f(Sub\,m\,n)n)$

   Смысл: если $m < n$, то выводим m, иначе считаем остаток от деления $m - n$ на n

   $Mod = YModfn$ (Y - комбинатор неподвижной точки) - остаток от деления m на n

   $IsDivisible = \lambda nm.IsZero(Mod\,n\,m)$ (n кратно m)

6. Выразим терм $IsPrime$. Это индикатор того, что n не делится на числа $2, \ldots, n-1$. Выразим терм "число n не делится на числа $m, \ldots, n-1$"

   $NoDivsfn = \lambda fmn.(IsEqual\,n\,m)\overline{1}((IsDivisible\,n\,m)False(f(Inc\,m)))$

   $IsPrime = \lambda n.And(GE\;n\;\overline{2})((YNodivsfn)\overline{2})$ (добавили условие ≥ 2, так как наша вспомогательная функция не учитывала 0 и 1)

7. Перейдем к выражению искомого терма. Вспомогательный терм: идем по числам, рассмотрели k, нашли m простых

   $NthPrimefn = \lambda fkmn.(IsPrime\,k)((IsEqual\,n(Inc\,m))k(f(Inc\,k)(Inc\,m))(f(Inc\,k)m)$

   Смысл: если k - простое, то если нашли все числа, возвращаем k, иначе переходим к следующим m, k и вычисляем ту же функцию. Если k не простое, то просто переходим к следующему k.

   $NthPrime = \lambda n.(YNthPrimefn)\overline{0}\,\overline{0}\,n$