Programming Paradigms Fall 2023

Nikolai Kudasov and Khaled Ismaeel

August 31, 2023

Week 1. Problem set

1. For each of the following λ-terms write down an α-equivalent term without shadowed variables (i.e. where different variables have distinct names):
   (a) $\lambda x.(\lambda x.x)\ x$
   (b) $\lambda a.(\lambda b.a\ b)\ a$
   (c) $\lambda d.d\ (\lambda d.d)$
   (d) $\lambda z.(\lambda z.z)\ z$
   (e) $(\lambda x.\lambda y.y)\ z\ x$
2. Write down evaluation sequence for the following λ-terms. Each step of the evaluation must correspond to a single β-reduction or an α-conversion. You may introduce aliases for subterms.
   (a) $(\lambda x.\lambda y.y)\ y\ z$
   (b) $(\lambda x.\lambda y.y)\ z\ (\lambda z.y)\ w$
   (c) $(\lambda b.\lambda x.\lambda y.b\ y\ x)\ (\lambda x.\lambda y.x)$
   (d) $(\lambda s.\lambda z.s\ (s\ (s\ z)))\ (\lambda b.\lambda x.\lambda y.b\ y\ x)\ (\lambda x.\lambda y.y)$
   (e) $(\lambda s.\lambda z.s\ (s\ z))\ (\lambda s.\lambda z.s\ (s\ z))\ (\lambda b.\lambda y.\lambda x.b\ x\ y)\ (\lambda y.\lambda x.x)$
3. Recall that with Church booleans we have the following encoding:
$$\mathsf{tru} = \lambda t.\lambda f.t$$
$$\mathsf{fls} = \lambda t.\lambda f.f$$
   (a) Using only bare λ-calculus (variables, λ-abstraction and application), write down a λ-term for logical NAND (nand) of two Church booleans. You may **not** use aliases.
   (b) Verify your implementation of nand by writing down evaluation sequence for the term nand fls tru. You must expand this term and then evaluate **without** aliases.
4. Recall that with Church numerals we have the following encoding:
$$\mathsf{c}_0 = \lambda s.\lambda z.z$$
$$\mathsf{c}_1 = \lambda s.\lambda z.sz$$
$$\mathsf{c}_2 = \lambda s.\lambda z.s\ (s\ z)$$
$$\mathsf{c}_3 = \lambda s.\lambda z.s\ (s\ (s\ z))$$
$$\dots$$
   (a) Using only bare λ-calculus (variables, λ-abstraction and application), write down a single λ-term for each of the following functions on natural numbers. You may **not** use aliases.
      i. $n \mapsto 2(n+1)$
      ii. $n \mapsto (n+1)^2$
      iii. $n \mapsto 2^{n+2}$
      iv. $n \mapsto 2^{2^n}$
   (b) Verify each your implementations of the functions above by writing down evaluation sequence for each of them, when applied to c_2. You may use aliases.