Hans de Nivelle, Patrick Maier

INFORMATIK

http://www.mpi-sb.mpg.de/~nivelle/teaching/intprooftools2003/main.html

Exercise 2.1 Which of the following pairs of terms are α-, β-, or η-equivalent?

1. $\lambda x\colon X\ p(x)$ and $\lambda y\colon X\ q(x)$,
2. $\lambda x\colon N\ \lambda y\colon N\ p(x,y)$ and $\lambda y\colon N\ \lambda x\colon N\ p(y,x)$.
3. $\lambda p\colon (X \to \mathrm{Form})\ \lambda x\colon X\ p(x)$ and $\lambda q\colon (X \to \mathrm{Form})\ \lambda y\colon X\ q(y)$,
4. $\lambda x\colon X\ f(x)$ and $(\lambda y\colon (X \to X)\ y) \cdot f$,
5. true and $(K\ (K\ \mathrm{true})) \cdot \mathrm{false}$
   in the context $K := \lambda x\colon \mathrm{Bool}\ \lambda y\colon \mathrm{Bool}\ x\colon (\mathrm{Bool} \to \mathrm{Bool})$.

Exercise 2.2 Substitute, then β-reduce:

1. $\lambda x\colon N\ p(x) \wedge q(x)$ for p in $\forall x\colon N\ p(x)$.
2. $\lambda x\colon N\ \exists y\ p(x,y)$ for p in $\forall x\colon N\ p(f(y))$.
3. $\lambda x, y\colon N\ x = y$ for p in $\forall x\colon N\ \exists y N\ p(x,y)$.

Exercise 2.3 Recall the `split` function from Exercise 1.3.

1. Completely specify `split` by means of the PVS type system. I. e., give a type for `split` such that there is exactly one function with that type.
2. Provide a recursive definition for `split` and type-check against the above type.
3. Prove in PVS that the above type is of cardinality at most 1, i. e., there is at most one function with that type.

Exercise 2.4 Binary trees and general trees.

1. Define an inductive datatype `Tree[T]` of binary trees over some parameter type `T`.
2. Define by recursion a function `size: [Tree[T] -> nat]` which computes the number of nodes in a tree.
3. Define by recursion a function `flatten: [Tree[T] -> list[T]]` which computes the list of all nodes that occur in the tree.

4. Prove that `length(flatten(t)) = size(t)`.
5. Define an inductive datatype `GenTree[T]` of general finitely branching trees over `T`. Redefine the functions `size` and `flatten` for `GenTree[T]` and prove again that `length(flatten(t)) = size(t)`.