CSCI 365 Problem Set 10: Lambda Calculus + Agda

Due Friday, 24 April 2026

Specification

To receive credit for this problem set:

- Complete exercises worth a total of at least 18 points.

The untyped λ-calculus

Exercise 1 (3 pts) Consider the Haskell data type

```
data Term where
  Var :: String -> Term
  Lam :: String -> Term -> Term
  App :: Term -> Term -> Term
```

which represents a naïve encoding of λ-calculus terms as Haskell values. Write a function

```
freeVars :: Term -> Set String
```

which computes the set of all *free variables* of a term.

- In any $\lambda x.\, t$, all occurrences of the variable x in the term t are *bound*.
- Any other variables are *free*.

For example, in the term

$$z\ (\lambda y.\, y\ (x\ (\lambda x.\, x\ y)))$$

the z and first x are free, whereas both ys and the second x are bound. As an example of what your function should do,

```
freeVars (App (Var "z") (Lam "y" (App (Var "y") (Var "x")))) = fromList ["z","x"].
```

The `Set` type can be found in the `Data.Set` module, which you can read about on Hackage: `https://hackage.haskell.org/package/containers/docs/Data-Set.html`. To import it into your `.hs` file, you'll want to put something like

```
import Data.Set (Set)
import qualified Data.Set as S
```

at the top. Then you can use the `Set` type and call functions from the library like `S.insert` and `S.union`.

Exercise 2 (2 pts) Ensure that your `freeVars` function takes $O(n \lg n)$ time or better, where n is the size (*i.e.* the total number of constructors) of the `Term`. Consult the `Data.Set` documentation for the asymptotic running time of relevant `Set` operations.

Note that some `Set` operations such as `union` are recorded as taking time $O(m \log(\frac{n}{m} + 1))$ which is somewhat unnecessarily precise. When $m = n$ this reduces to $O(n)$, and in any case it will always be no larger than $O(n)$.

Further Exploration

Exercise 3 (2 pts) Take a look at *An Unsolvable Problem of Elementary Number Theory* by Alonzo Church[1], available from `https://www.jstor.org/stable/2371045`. This was not the very first paper to introduce the λ-calculus, but it is one of the first where the λ-calculus is more or less recognizable as we use it today.

[1] Alonzo Church. American Journal of Mathematics, Vol. 58, No. 2 (Apr., 1936), pp. 345–363

Skim through the beginning of the paper and *either*:

- Read section 1 and footnote 3, and explain the significance of footnote 3. (Note also that Turing's paper introducing the Turing Machine—`https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf`—was published later in the same year, 1936.)
- *Or*, read section 2 and explain how Church's notation and operations I and II correspond to things we have discussed in class. (Church's operation III is called η-conversion, which we did not discuss.)

Agda

Complete exercises found in `PS10.agda`, linked from the course website.