Problems

These problems will be discussed at seminar 2. Programs in Haskell are not required to compile. The exercises in Nielson have been renumbered. My references are to 1999 edition.

Problems

1 Define multiplication and exponentiation for the datatype `N` given that

```
data N = Zero | Suc N

add Zero m = m
add (Suc n) m = Suc (add n m)
```

2 Prove

> **Lemma.** Let `m` be any element in `N`. Then `add n (Suc m) = add (Suc n) m` for all `n` in `N`. ■

3 State an induction principle for

```
data List = Empty | Cons Integer List
```

4 Define the length of a `List` and composition of two `List`s by

```
len :: List -> Integer
len Empty = 0
len (Cons _ tail) = 1 + len tail
append :: List -> List -> List
append Empty ls = ls
append (Cons i ls) ls' = Cons i (append ls ls')
```

Prove that `len (append l1 l2) = len l1 + len l2` for all `l1` and `l2` in `List`.

5 Prove that $\mathcal{A}[\![a[y \mapsto a_0]]\!]\sigma = \mathcal{A}[\![a]\!](\sigma[y \mapsto \mathcal{A}[\![a_0]\!]\sigma])$. (Exercise 1.13.)

6 Define substitution for boolean expressions $b[y \mapsto a_0]$ and prove that $\mathcal{B}[\![b[y \mapsto a_0]]\!]\sigma = \mathcal{B}[\![b]\!](\sigma[y \mapsto \mathcal{A}[\![a_0]\!]\sigma])$. (Exercise 1.14.)

7 Define a natural semantics for binary numerals, **Num**, defined by the grammar $n ::= 0 \mid 1 \mid n\ 0 \mid n\ 1$. The formulae of the theory should be on the form $\langle n \rangle \to z$, where $n \in \mathbf{Num}$ and $z \in \mathbb{N}$ is a natural number.

8 With the semantics of the previous problem and $\mathcal{N}$ from the text book use structural induction and induction over the shape of derivation trees to prove that $\langle n \rangle \to z$ if and only if $\mathcal{N}[\![n]\!] = z$.