

Programming Languages and Types

Exercise 12

Yi Dai

January 31, 2013

1 Simply-Typed λ -Calculus

1.1 Typing Derivations

1.2 Programming with Extensions

1. Define the subtraction function *sub* in the simply-typed λ -calculus extended with Peano numbers (**0** and **succ**) and fixed point operator **fix**.¹

```
fix ( $\lambda fsub : (\mathbf{Nat} \rightarrow \mathbf{Nat} \rightarrow \mathbf{Nat}) \rightarrow (\mathbf{Nat} \rightarrow \mathbf{Nat} \rightarrow \mathbf{Nat})$  .  
     $\lambda m : \mathbf{Nat} . \lambda n : \mathbf{Nat}$   
      if (iszero m)  
        then 0  
        else (succ (fsub (pred m) n)) )
```

2 System- \mathcal{F}

2.1 Parametric Polymorphism

1. Define the polymorphic function *I*.
2. Define the polymorphic function *K*.
3. Define the polymorphic function *S*.

¹I apologize for giving the wrong type $(\mathbf{Nat} \rightarrow \mathbf{Nat} \rightarrow \mathbf{Nat})$ to the parameter *fsub* during the exercise session.

2.2 Typing Church-Encodings

1. Church-numerals
2. Church-booleans