

Simply-typed lambda calculus; More types (Lectures 11-12)  
Section and Practice Problems

Week 7: Tue Mar 12–Fri Mar 15, 2019

---

## 1 Simply-typed lambda calculus

(a) Add appropriate type annotations to the following expressions, and state the type of the expression.

(i)  $\lambda a. a + 4$

(ii)  $\lambda f. 3 + f ()$

(iii)  $(\lambda x. x) (\lambda f. f (f 42))$

(b) For each of the following expressions, give a derivation showing that the expression is well typed.

(i)  $(\lambda f : \mathbf{int} \rightarrow \mathbf{int}. f 38) (\lambda a : \mathbf{int}. a + 4)$

(ii)  $\lambda g : (\mathbf{int} \rightarrow \mathbf{int}) \rightarrow (\mathbf{int} \rightarrow \mathbf{int}). g (\lambda c : \mathbf{int}. c + 1) 7$

(iii)  $\lambda f : \mathbf{int} \rightarrow \mathbf{int}. \lambda g : \mathbf{int} \rightarrow \mathbf{int}. \lambda x : \mathbf{int}. g (f x)$

## 2 Type soundness

(a) Recall the substitution lemma that we used in the proof of type soundness.

**Lemma (Substitution).** *If  $x : \tau' \vdash e : \tau$  and  $\vdash v : \tau'$  then  $\vdash e\{v/x\} : \tau$ .*

Using the definition of substitution given in Assignment 2, prove this lemma. You may assume that  $v$  does not have any free variables (i.e.,  $FV(v) = \emptyset$ ).

Remember to state what set you are performing induction on and what the property is that you are proving for every element in that set. If you are not sure what cases you need to consider, or what you are able to assume in each case of the inductive proof, we strongly suggest that you write down the inductive reasoning principle for the inductively defined set.

(b) Recall the context lemma that we used in the proof of type soundness.

**Lemma (Context).** *If  $\vdash E[e_0] : \tau$  and  $\vdash e_0 : \tau'$  and  $\vdash e_1 : \tau'$  then  $\vdash E[e_1] : \tau$ .*

Prove this lemma.

Remember to state what set you are performing induction on and what the property is that you are proving for every element in that set. If you are not sure what cases you need to consider, or what you are able to assume in each case of the inductive proof, we strongly suggest that you write down the inductive reasoning principle for the inductively defined set.

### 3 Products and Sums

For these questions, use the lambda calculus with products and sums (Lecture 13§1.1).

- (a) Write a program that constructs two values of type  $\mathbf{int} + (\mathbf{int} \rightarrow \mathbf{int})$ , one using left injection, and one using right injection.
- (b) Write a function that takes a value of type  $\mathbf{int} + (\mathbf{int} \rightarrow \mathbf{int})$  and if the value is an integer, it adds 7 to it, and if the value is a function it applies the function to 42.
- (c) Give a typing derivation for the following program.

$$\lambda p: (\mathbf{unit} \rightarrow \mathbf{int}) \times (\mathbf{int} \rightarrow \mathbf{int}). \lambda x: \mathbf{unit} + \mathbf{int}. \text{case } x \text{ of } \#1 p \mid \#2 p$$

- (d) Write a program that uses the term in part (c) above to produce the value 42.

### 4 Recursion

- (a) Use the  $\mu x. e$  expression to write a function that takes a natural number  $n$  and returns the sum of all even natural numbers less than or equal to  $n$ . (You can assume you have appropriate integer comparison operators, and also a modulus operator.)
- (b) Try executing your program by applying it to the number 5.
- (c) Give a typing derivation for the following program. What happens if you execute the program?

$$\mu p: (\mathbf{int} \rightarrow \mathbf{int}) \times (\mathbf{int} \rightarrow \mathbf{int}). (\lambda n: \mathbf{int}. n + 1, \#1 p)$$

### 5 References

- (a) Give a typing derivation for the following program.

```
let a: int ref = ref 4 in
let b: (int → int) ref = ref λx: int. x + 38 in
!b !a
```

- (b) Execute the program above for 4 small steps, to get configuration  $\langle e, \sigma \rangle$ . What is an appropriate  $\Sigma$  such that  $\emptyset, \Sigma \vdash e: \tau$  and  $\Sigma \vdash \sigma$ ?