

Practice Problems – Type Systems

Here is the simply typed lambda calculus, extended with integers, and its type system:

$$\begin{aligned} e &::= v \mid x \mid e \ e \\ v &::= n \mid \lambda x : t . e \\ t &::= int \mid t \rightarrow t \\ A &::= \cdot \mid x : t, A \end{aligned}$$

$$\begin{array}{c} \text{INT} \\ \hline A \vdash n : int \\ \text{VAR} \\ \hline \frac{x \in \text{dom}(A)}{A \vdash x : A(x)} \\ \text{LAM} \\ \hline \frac{x : t, A \vdash e : t'}{A \vdash \lambda x : t . e : t \rightarrow t'} \\ \text{APP} \\ \hline \frac{A \vdash e_1 : t \rightarrow t' \quad A \vdash e_2 : t}{A \vdash e_1 \ e_2 : t'} \end{array}$$

1. Draw derivations showing that the following typing judgments hold:

- (a) $\cdot \vdash 42 : int$
- (b) $y : int \vdash \lambda x : int . y : int \rightarrow int$
- (c) $\cdot \vdash \lambda x : int . \lambda y : int . x : int \rightarrow int \rightarrow int$
- (d) $+ : int \rightarrow int \rightarrow int \vdash (\lambda f : int \rightarrow int . f \ 42) (\lambda x : int . + \ x \ 3) : int$

To save writing effort, you can write i instead of int .

- 2. Give a simply typed lambda calculus term that is type-incorrect, and yet it does not get stuck at run time. Your term must not be typable by the trivial operation of changing type annotations on parameters. For example, $(\lambda x : int \rightarrow int . x) \ 3$ is not a valid answer.
- 3. Finally, consider type inference for the simply-typed lambda calculus:

$$\begin{aligned} e &::= v \mid x \mid e \ e \\ v &::= n \mid \lambda x . e \\ t &::= \alpha \mid int \mid t \rightarrow t \\ A &::= \cdot \mid x : t, A \end{aligned}$$

$$\begin{array}{c} \text{INT} \\ \hline A \vdash n : int \\ \text{VAR} \\ \hline \frac{x \in \text{dom}(A)}{A \vdash x : A(x)} \\ \text{LAM} \\ \hline \frac{x : \alpha, A \vdash e : t' \quad \alpha \text{ fresh}}{A \vdash \lambda x . e : \alpha \rightarrow t'} \\ \text{APP} \\ \hline \frac{A \vdash e_1 : t \quad A \vdash e_2 : t' \quad t = t' \rightarrow \beta \quad \beta \text{ fresh}}{A \vdash e_1 \ e_2 : \beta} \end{array}$$

Draw derivations showing type inference applied to the following terms in the empty type environment; write down a solution to the associated constraints; and write down the fully resolved type of the term. We've done the first one as an example.

$$\begin{array}{c} \text{(a) } \lambda x . x \\ \frac{x : \alpha \vdash x : \alpha}{\cdot \vdash \lambda x . x : \alpha \rightarrow \alpha} \quad \cdot \vdash 42 : int \quad (\alpha \rightarrow \alpha) = (int \rightarrow \beta) \\ \hline \cdot \vdash (\lambda x . x) \ 42 : \beta \end{array}$$

Solution: $\alpha = \beta = int$.

Type: int .

- (b) $\lambda x . \lambda y . x$
- (c) $\lambda x . \lambda y . x \ y$
- (d) $(\lambda x . \lambda y . x) \ 3 \ 42$