

Erasure and Typability

Erasure

We can transform terms in λ_{\rightarrow} to terms of the untyped lambda-calculus simply by erasing type annotations on lambda-abstractions.

$$\begin{aligned} \text{erase}(x) &= x \\ \text{erase}(\lambda x:T_1. t_2) &= \lambda x. \text{erase}(t_2) \\ \text{erase}(t_1 t_2) &= \text{erase}(t_1) \text{erase}(t_2) \end{aligned}$$

Typability

Conversely, an untyped λ -term m is said to be *typable* if there is some term t in the simply typed lambda-calculus, some type T , and some context Γ such that $\text{erase}(t) = m$ and $\Gamma \vdash t : T$.

This process is called *type reconstruction* or *type inference*.

Typability

Conversely, an untyped λ -term m is said to be *typable* if there is some term t in the simply typed lambda-calculus, some type T , and some context Γ such that $\text{erase}(t) = m$ and $\Gamma \vdash t : T$.

This process is called *type reconstruction* or *type inference*.

Example: Is the term

$\lambda x. x x$

typable?