Recall the static semantics (typing rules) of the simply-typed λ-calculus with natural numbers:

\[
\begin{align*}
\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} & \quad \frac{n \in \mathbb{N}}{\Gamma \vdash n : \text{nat}} \\
\frac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x : \tau. e : \tau \rightarrow \tau'} & \quad \frac{\Gamma \vdash e_1 : \tau_2 \rightarrow \tau \quad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash e_1(e_2) : \tau}
\end{align*}
\]

We will consider two versions of this language, a natural one and an unnatural one. The natural one is as defined above. The unnatural one is the same, except with natural numbers \(n\) (and their corresponding typing rule) removed. Note: The type \(\text{nat}\) is present in both the natural and unnatural languages; in the unnatural one, however, there are no values of type \(\text{nat}\).

1 The Natural Language

The following problems concern the natural version of the language.

Warmup

Prove the well-known substitution lemma, which is needed in the proof of type preservation:

Suppose \(\Gamma, x : \tau \vdash e' : \tau'\). Then, for all \(e\), \(\Gamma \vdash e : \tau\) implies \(\Gamma \vdash e'[e/x] : \tau'\).

1.1 Inhabitation (1 points)

Prove that all types are inhabited, that is:

For all types \(\tau\), there exists a closed term \(e\) of type \(\tau\).

1.2 Anti-Substitution (5 points)

Prove or refute the converse of substitution, which we'll call anti-substitution. That is:

Suppose that, for all \(e\), \(\Gamma \vdash e : \tau\) implies \(\Gamma \vdash e'[e/x] : \tau'\). Then, \(\Gamma, x : \tau \vdash e' : \tau'\).

Your answer should make use of the inhabitation property proved above.
2 The Unnatural Language

The following problems concern the unnatural version of the language.

2.1 Uninhabitation (1 point)

It is clear that there are no closed values of type nat in this language. What is less immediately obvious is that there are no closed terms of type nat. For instance, there might be some term of the form $e_1(e_2)$ with type nat. Prove that this is not the case, i.e., that the type nat is in fact uninhabited. Your proof should be a one-liner.

2.2 Uninhabitation, Statically (5 points)

Prove uninhabitation of nat, but this time without mentioning the dynamic semantics (reduction relation) of the $\lambda$-calculus.

2.3 Anti-Substitution Revisited (3 points)

Prove or refute anti-substitution, as stated in Problem 1.2. Your answer should make use of the uninhabitation of nat.