Type theory and proof assistants

answers

1. \[ \lambda x : a \to b \to c. \lambda y : b. \lambda z : a. xzy \]

(This term corresponds to the proof

\[
\begin{align*}
\frac{a \to b \to c \quad [a^z]}{b \to c \quad E \to} & \quad \frac{c \quad E \to}{a \to c \quad I[z] \to} \\
\frac{b \to a \to c \quad I[y] \to}{(a \to b \to c) \to b \to a \to c \quad I[x] \to}
\end{align*}
\]

but this proof is not part of the answer.)

2. \[
\begin{align*}
\Gamma \vdash x : a \to b \to c & \quad \Gamma \vdash z : a \\
\Gamma \vdash xz : b \to c & \quad \Gamma \vdash y : b \\
\Gamma \vdash xzy : c \\
x : a \to b \to c, y : b, z : a & \vdash (\lambda z : a. xzy) : a \to c
\end{align*}
\]

where we used the abbreviation \( \Gamma := x : a \to b \to c, y : b, z : a \).

3. \[
\begin{align*}
\frac{[A^v] \quad I[y] \to \quad [A^x]}{A \to A \quad E \to} & \quad \frac{A \quad E \to}{A \to A \quad I[x] \to} \\
\frac{[A^x] \quad I[x] \to}{A \to A \quad I[x] \to}
\end{align*}
\]

This corresponds to the reduction

\[ \lambda x : A. (\lambda y : A. y) x \to_\beta \lambda x : A. x \]

4. \[ \Pi A : \star. \text{list } A \ 0 \]

In Coq notation this is

1
forall A : Set, list A O

5. forall P : tree -> Prop,
P leaf -
(forall t1 : tree, P t1 - forall t2 : tree, P t2 -
P (node t1 t2)) -
forall t : tree, P t

6.

7.

(This term corresponds to the proof

but this proof is not part of the answer.)

8.

2 := λa : *. λz : a. λs : a - a, s (sz)

9.

system judgments

\begin{align*}
    \lambda & \rightarrow 1, 2, 4 \\
    \lambda P & \rightarrow 1, 2, 4, 5 \\
    \lambda 2 & \rightarrow 1, 2, 3, 4 \\
\end{align*}

10. Inductive even : nat -> Prop :=
    | even_0 : even 0
    | even_SS : forall n : nat, even n - even (S (S n)).
11. \[
\Phi_{\text{even}}(X) := \{0\} \cup \{n + 2 \mid n \in X\}
\]
\(\Phi_{\text{even}}\) is order-preserving means that \(\Phi_{\text{even}}(X) \subseteq \Phi_{\text{even}}(Y)\) when \(X \subseteq Y\).

12. Take for \(L\) the lattice from the previous exercise and for \(\Phi\)

\[
\Phi(X) = N \setminus X
\]

\[
H = \{X \mid X \subseteq N \setminus X\} = \emptyset
\]

\[
\bigvee_{L} H = \bigvee_{L} \{\emptyset\} = \emptyset
\]

13. Type checking: given \(\Gamma, M\) and \(A\), determine whether \(\Gamma \vdash M : A\) is a derivable judgment.

Type synthesis: given \(\Gamma\) and \(M\), determine whether an \(A\) exists such that \(\Gamma \vdash M : A\) is a derivable judgment, and if so, find one.

Type inhabitation: given \(\Gamma\) and \(A\), determine whether an \(M\) exists such that \(\Gamma \vdash M : A\) is a derivable judgment, and if so, find one.

The first two are decidable for \(\lambda P\), while the last is not decidable.

14. If the last step of the derivation was a \(\lambda\) rule

\[
\begin{array}{c}
\Gamma, x : B, y : C \vdash N : \ldots \quad \ldots \\
\hline
\Gamma, x : B \vdash (\lambda y : C. N) : \ldots
\end{array}
\]

one would like to use the induction hypothesis for the derivation of \(\Gamma, x : B, y : C \vdash N : \ldots\) to obtain a derivation

\[
\begin{array}{c}
\ldots \\
\hline
\Gamma, y : C[x := P] \vdash N[x := P] : \ldots \quad \ldots \\
\hline
\Gamma \vdash (\lambda y : C. N)[x := P] : \ldots
\end{array}
\]

However, this does not work, because there \(x\) is not the last variable in the context.

The way to solve this problem is to use induction loading and instead prove

\[
\begin{aligned}
\Gamma, x : B, \Delta \vdash M : A & \quad \text{and} \quad \Gamma \vdash P : B \\
\text{then} \quad \Gamma, \Delta[x := P] \vdash M[x := P] : A[x := P]
\end{aligned}
\]

The lemma of the exercise is the special case of this where \(\Delta\) is the empty context.

15. There are two cases:
• If $A = a$, then we know that $M[x := N]\vec{P}$ is strongly normalizing, and hence $M$ and the terms in $\vec{P}$ are also strongly normalizing. This means that there are only finitely many reduction steps possible in $(\lambda x. M) N \vec{P}$ which do not contract the redex. Once we contract the redex, we get a reduct of $M[x := N] \vec{P}$ (by applying the reductions of $M$, $N$ and $\vec{P}$ that we already did to $M[x := N] \vec{P}$) and we are in a reduct of a strongly normalizing term. Which cannot have infinitely many reductions.

• If $A = B \rightarrow C$ then we know that

$$\forall P^\prime \in \llbracket B \rrbracket. M[x := N] \vec{P} P^\prime \in \llbracket C \rrbracket$$

and need to show that

$$\forall P^\prime \in \llbracket B \rrbracket. (\lambda x. M) N \vec{P} P^\prime \in \llbracket C \rrbracket$$

But that immediately follows from the induction hypothesis, because we are doing induction on the structure of the type.

(Note that we do not need the induction hypothesis for $B$.)