Chapter 1

Datatypes

Term variables in pure first-order logic range over all kinds of terms. In order to specify the domain of a term variable, we need to introduce a corresponding predicate symbol. For example, we use a predicate $Nat(x)$ to specify that x ranges over natural numbers, as in:

$$\forall x.Nat(x) \supset A$$
$$\exists x.Nat(x) \wedge A$$

This chapter develops first-order logic with datatypes which explicitly specifies the domain of each term variable inside the binder $\forall$ or $\exists$. For example, the above propositions can be concisely written as

$$\forall x \in \mathsf{nat}.A(x)$$
$$\exists x \in \mathsf{nat}.A(x)$$

where nat is a datatype for natural numbers and $A(x)$ indicates that proposition A contains term variable x.

Throughout the chapter, we adopt the new notation $A(x)$ to mean that proposition A contains term variable x. Then $A(t)$ stands for A in which every occurrence of x has been replaced by t; that is, we have $A(t) = [t/x]A$.

We introduce two new judgments τ type, to state that τ is a datatype, and $t \in \tau$, to state that term t is an element of datatype τ. Again we use natural deduction to explain the meaning of $t \in \tau$.

$$\begin{array}{rcl} \tau \text{ type} & \Leftrightarrow & \textit{τ is a datatype} \\ t \in \tau & \Leftrightarrow & \textit{term t has datatype τ} \end{array}$$

We use metavariables τ and σ for datatypes, and t and s for terms.

1.1 Natural numbers

Syntax:

$$\begin{array}{lrcl} \text{datatype} & \tau & ::= & \mathsf{nat} \\ \text{term} & t & ::= & \mathbf{0} \mid \mathbf{s}(t) \end{array}$$

Formation rule:

$$\frac{}{\mathsf{nat\ type}}\ \mathsf{natF}$$

Introduction rules:

$$\frac{}{\mathbf{0} \in \mathsf{nat}}\ \mathsf{natI}_0 \qquad \frac{t \in \mathsf{nat}}{\mathbf{s}(t) \in \mathsf{nat}}\ \mathsf{natI}_\mathsf{s}$$

The elimination rule is similar to the elimination rule for $\vee$ which analyzes two possibilities simultaneously.

$$\dfrac{t \in \mathsf{nat} \quad t_0 \in \tau \qquad \begin{array}{c}\overline{x \in \mathsf{nat}} \\ \vdots \\ t_s \in \tau\end{array}}{\mathbf{case}\ t\ \mathbf{of}\ \mathbf{0} \Rightarrow t_0 \mid \mathbf{s}(x) \Rightarrow t_s \in \tau}\ \mathsf{natE}$$

Using hypothetical judgments:

$$\dfrac{}{\Gamma \vdash \mathbf{0} \in \mathsf{nat}}\ \mathsf{natI_0} \qquad \dfrac{\Gamma \vdash t \in \mathsf{nat}}{\Gamma \vdash \mathbf{s}(t) \in \mathsf{nat}}\ \mathsf{natI_s} \qquad \dfrac{\Gamma \vdash t \in \mathsf{nat} \quad \Gamma \vdash t_0 \in \tau \quad \Gamma, x \in \mathsf{nat} \vdash t_s \in \tau}{\Gamma \vdash \mathbf{case}\ t\ \mathbf{of}\ \mathbf{0} \Rightarrow t_0 \mid \mathbf{s}(x) \Rightarrow t_s \in \tau}\ \mathsf{natE}$$

Local reductions of proofs:

$$\dfrac{\dfrac{}{\mathbf{0} \in \mathsf{nat}}\ \mathsf{natI_0} \quad \begin{array}{c}\mathcal{E} \\ t_0 \in \tau\end{array} \qquad \begin{array}{c}\overline{x \in \mathsf{nat}} \\ \vdots \\ t_s \in \tau\end{array}}{\mathbf{case}\ \mathbf{0}\ \mathbf{of}\ \mathbf{0} \Rightarrow t_0 \mid \mathbf{s}(x) \Rightarrow t_s \in \tau}\ \mathsf{natE} \quad \Longrightarrow_R \quad \begin{array}{c}\mathcal{E} \\ t_0 \in \tau\end{array}$$

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D} \\ t \in \mathsf{nat}\end{array}}{\mathbf{s}(t) \in \mathsf{nat}}\ \mathsf{natI_s} \quad \begin{array}{c}\mathcal{E} \\ t_0 \in \tau\end{array} \qquad \begin{array}{c}\overline{x \in \mathsf{nat}} \\ \vdots \\ t_s \in \tau\end{array}}{\mathbf{case}\ \mathbf{s}(t)\ \mathbf{of}\ \mathbf{0} \Rightarrow t_0 \mid \mathbf{s}(x) \Rightarrow t_s \in \tau}\ \mathsf{natE} \quad \Longrightarrow_R \quad \begin{array}{c}\mathcal{D} \\ t \in \mathsf{nat} \\ \vdots \\ [t/x]t_s \in \tau\end{array}$$

Local reductions of terms:

$$\begin{array}{rcl}\mathbf{case}\ \mathbf{0}\ \mathbf{of}\ \mathbf{0} \Rightarrow t_0 \mid \mathbf{s}(x) \Rightarrow t_s & \Longrightarrow_R & t_0 \\ \mathbf{case}\ \mathbf{s}(t)\ \mathbf{of}\ \mathbf{0} \Rightarrow t_0 \mid \mathbf{s}(x) \Rightarrow t_s & \Longrightarrow_R & [t/x]t_s\end{array}$$

1.2 Function types and product types

Syntax:

$$\begin{array}{lrcl}\text{datatype} & \tau & ::= & \cdots \mid \tau \to \tau \mid \tau \times \tau \\ \text{term} & t & ::= & \cdots \mid \lambda x \in \tau.\, t \mid t\ t \mid \langle t, t \rangle \mid \mathbf{fst}\ t \mid \mathbf{snd}\ t\end{array}$$

$\tau \to \tau'$ is called a function type; $\tau \times \tau'$ is called a product type.

Formation rules:

$$\dfrac{\tau\ \mathsf{type} \quad \sigma\ \mathsf{type}}{\tau \to \sigma\ \mathsf{type}}\ {\to}\mathsf{F} \qquad \dfrac{\tau\ \mathsf{type} \quad \sigma\ \mathsf{type}}{\tau \times \sigma\ \mathsf{type}}\ {\times}\mathsf{F}$$

Introduction and elimination rules:

$$\dfrac{\begin{array}{c}\overline{x \in \tau} \\ \vdots \\ t \in \sigma\end{array}}{\lambda x \in \tau.\, t \in \tau \to \sigma}\ {\to}\mathsf{I} \qquad \dfrac{t \in \tau \to \sigma \quad s \in \tau}{t\ s \in \sigma}\ {\to}\mathsf{E} \qquad \dfrac{t \in \tau \quad s \in \sigma}{\langle t, s \rangle \in \tau \times \sigma}\ {\times}\mathsf{I} \qquad \dfrac{t \in \tau \times \sigma}{\mathbf{fst}\ t \in \tau}\ {\times}\mathsf{E_L} \qquad \dfrac{t \in \tau \times \sigma}{\mathbf{snd}\ t \in \sigma}\ {\times}\mathsf{E_R}$$

Using hypothetical judgments:

$$\dfrac{\Gamma, x \in \tau \vdash t \in \sigma}{\Gamma \vdash \lambda x \in \tau.\, t \in \tau \to \sigma}\ {\to}\mathsf{I} \qquad \dfrac{\Gamma \vdash t \in \tau \to \sigma \quad \Gamma \vdash s \in \tau}{\Gamma \vdash t\ s \in \sigma}\ {\to}\mathsf{E}$$

$$\frac{\Gamma \vdash t \in \tau \quad \Gamma \vdash s \in \sigma}{\Gamma \vdash \langle t, s\rangle \in \tau \times \sigma}\ \times\mathsf{I} \qquad \frac{\Gamma \vdash t \in \tau \times \sigma}{\Gamma \vdash \mathbf{fst}\ t \in \tau}\ \times\mathsf{E_L} \qquad \frac{\Gamma \vdash t \in \tau \times \sigma}{\Gamma \vdash \mathbf{snd}\ t \in \sigma}\ \times\mathsf{E_R}$$

Local reductions of terms:

$$\begin{array}{rcl} (\lambda x \in \tau.\, t)\ s & \Longrightarrow_R & [s/x]t \\ \mathbf{fst}\ \langle t, s\rangle & \Longrightarrow_R & t \\ \mathbf{snd}\ \langle t, s\rangle & \Longrightarrow_R & s \end{array}$$

Examples

Predecessor:

$$\begin{array}{l} \mathit{pred} \in \mathsf{nat} \to \mathsf{nat} \\ \mathit{pred}\ =\ \lambda x \in \mathsf{nat}.\, \mathbf{case}\ x\ \mathbf{of}\ \mathbf{0} \Rightarrow \mathbf{0} \mid \mathbf{s}(y) \Rightarrow y \end{array}$$

Double:

$$\begin{array}{l} \mathit{double} \in \mathsf{nat} \to \mathsf{nat} \\ \mathit{double}\ =\ \lambda x \in \mathsf{nat}.\, \mathbf{case}\ x\ \mathbf{of}\ \mathbf{0} \Rightarrow \mathbf{0} \mid \mathbf{s}(y) \Rightarrow \mathbf{s}(\mathbf{s}(\mathit{double}\ y)) \end{array}$$

double makes a recursive call, which is not allowed. So we introduce a new term construct for *primitive recursion.*

1.3 Primitive recursion

The rule natE for primitive recursion on natural numbers is another elimination rule for nat:

$$\frac{\begin{array}{ccc} & \overline{x \in \mathsf{nat}} & \overline{f(x) \in \tau} \\ & & \vdots \\ t \in \mathsf{nat} \quad t_0 \in \tau & & t_s \in \tau \end{array}}{\mathbf{rec}\ f(t)\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s \in \tau}\ \mathsf{natE}$$

Using hypothetical judgment:

$$\frac{\Gamma \vdash t \in \mathsf{nat} \quad \Gamma \vdash t_0 \in \tau \quad \Gamma, x \in \mathsf{nat}, f(x) \in \tau \vdash t_s \in \tau}{\Gamma \vdash \mathbf{rec}\ f(t)\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s \in \tau}\ \mathsf{natE}$$

- t_0 is not allowed to make a recursive call to f.
- t_s is allowed to make a recursive call to f, but only with argument x.
- A recursive call in t_s is written as $f(x)$, which is not an application term but a variable in itself.
- We may think of $\mathbf{rec}\ f(t)\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s$ as a primitive recursive function f applied to t.
- Sometimes we write $\mathbf{rec}\ f(t)\ \mathbf{of} \left\{ \begin{array}{l} f(\mathbf{0}) \Rightarrow t_0 \\ f(\mathbf{s}(x)) \Rightarrow t_s \end{array} \right.$ for visual clarity.

Local reductions of terms:

$$\begin{array}{rcl} \mathbf{rec}\ f(\mathbf{0})\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s & \Longrightarrow_R & t_0 \\ \mathbf{rec}\ f(\mathbf{s}(t))\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s & \Longrightarrow_R & [\mathbf{rec}\ f(t)\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s / f(x)][t/x]t_s \end{array}$$

Note that a primitive recursion always terminates because the argument to f always decreases.

Examples

Double:

$$\begin{array}{rcl} \mathit{double}\ \mathbf{0} & = & \mathbf{0} \\ \mathit{double}\ \mathbf{s}(x) & = & \mathbf{s}(\mathbf{s}(\mathit{double}\ x)) \end{array}$$

$$\begin{array}{l} \mathit{double} \in \mathsf{nat} \to \mathsf{nat} \\ \mathit{double}\ =\ \lambda x \in \mathsf{nat}.\, \mathbf{rec}\ d(x)\ \mathbf{of}\ d(\mathbf{0}) \Rightarrow \mathbf{0} \mid d(\mathbf{s}(y)) \Rightarrow \mathbf{s}(\mathbf{s}(d(y))) \end{array}$$

$$\begin{array}{rcl} \mathit{double}\ \mathbf{s}(\mathbf{0}) & \Longrightarrow_R & \mathbf{rec}\ d(\mathbf{s}(\mathbf{0}))\ \mathbf{of}\ d(\mathbf{0}) \Rightarrow \mathbf{0} \mid d(\mathbf{s}(y)) \Rightarrow \mathbf{s}(\mathbf{s}(d(y))) \\ & \Longrightarrow_R & \mathbf{s}(\mathbf{s}(\mathbf{rec}\ d(\mathbf{0})\ \mathbf{of}\ d(\mathbf{0}) \Rightarrow \mathbf{0} \mid d(\mathbf{s}(y)) \Rightarrow \mathbf{s}(\mathbf{s}(d(y))))) \\ & \Longrightarrow_R & \mathbf{s}(\mathbf{s}(\mathbf{0})) \end{array}$$

Add:

$$\begin{array}{rcl} \mathit{plus}\ \mathbf{0}\ y & = & y \\ \mathit{plus}\ (\mathbf{s}(x))\ y & = & \mathbf{s}(\mathit{plus}\ x\ y) \end{array}$$

$$\begin{array}{l} \mathit{plus} \in \mathsf{nat} \to \mathsf{nat} \to \mathsf{nat} \\ \mathit{plus}\ =\ \lambda x \in \mathsf{nat}.\, \lambda y \in \mathsf{nat}.\, \mathbf{rec}\ p(x)\ \mathbf{of}\ p(\mathbf{0}) \Rightarrow y \mid p(\mathbf{s}(z)) \Rightarrow \mathbf{s}(p(z)) \end{array}$$

$$\begin{array}{rcl} \mathit{plus}\ \mathbf{s}(\mathbf{0})\ t & \Longrightarrow_R & (\lambda y \in \mathsf{nat}.\, \mathbf{rec}\ p(\mathbf{s}(\mathbf{0}))\ \mathbf{of}\ p(\mathbf{0}) \Rightarrow y \mid p(\mathbf{s}(z)) \Rightarrow \mathbf{s}(p(z)))\ t \\ & \Longrightarrow_R & \mathbf{rec}\ p(\mathbf{s}(\mathbf{0}))\ \mathbf{of}\ p(\mathbf{0}) \Rightarrow t \mid p(\mathbf{s}(z)) \Rightarrow \mathbf{s}(p(z)) \\ & \Longrightarrow_R & \mathbf{s}(\mathbf{rec}\ p(\mathbf{0})\ \mathbf{of}\ p(\mathbf{0}) \Rightarrow t \mid p(\mathbf{s}(z)) \Rightarrow \mathbf{s}(p(z))) \\ & \Longrightarrow_R & \mathbf{s}(t) \end{array}$$

An alternative definition is given as follows:

$$\begin{array}{l} \mathit{plus} \in \mathsf{nat} \to \mathsf{nat} \to \mathsf{nat} \\ \mathit{plus}\ =\ \lambda x \in \mathsf{nat}.\, \mathbf{rec}\ p(x)\ \mathbf{of}\ p(\mathbf{0}) \Rightarrow \lambda y \in \mathsf{nat}.\, y \mid p(\mathbf{s}(z)) \Rightarrow \lambda y \in \mathsf{nat}.\, \mathbf{s}(p(z)\ y) \end{array}$$

$$\begin{array}{rcl} \mathit{plus}\ \mathbf{s}(\mathbf{0})\ t & \Longrightarrow_R & \left(\mathbf{rec}\ p(\mathbf{s}(\mathbf{0}))\ \mathbf{of}\ \left\{\begin{array}{l} p(\mathbf{0}) \Rightarrow \lambda y \in \mathsf{nat}.\, y \\ p(\mathbf{s}(z)) \Rightarrow \lambda y \in \mathsf{nat}.\, \mathbf{s}(p(z)\ y) \end{array}\right.\right)\ t \\ & \Longrightarrow_R & \lambda y \in \mathsf{nat}.\, \mathbf{s}(\left(\mathbf{rec}\ p(\mathbf{0})\ \mathbf{of}\ \left\{\begin{array}{l} p(\mathbf{0}) \Rightarrow \lambda y \in \mathsf{nat}.\, y \\ p(\mathbf{s}(z)) \Rightarrow \lambda y \in \mathsf{nat}.\, \mathbf{s}(p(z)\ y) \end{array}\right.\right)\ y)\ t \\ & \Longrightarrow_R & \mathbf{s}(\left(\mathbf{rec}\ p(\mathbf{0})\ \mathbf{of}\ \left\{\begin{array}{l} p(\mathbf{0}) \Rightarrow \lambda y \in \mathsf{nat}.\, y \\ p(\mathbf{s}(z)) \Rightarrow \lambda y \in \mathsf{nat}.\, \mathbf{s}(p(z)\ y) \end{array}\right.\right)\ t) \\ & \Longrightarrow_R & \mathbf{s}((\lambda y \in \mathsf{nat}.\, y)\ t) \\ & \Longrightarrow_R & \mathbf{s}(t) \end{array}$$

1.4 Boolean values and lists

Boolean values

Syntax:

$$\begin{array}{llcl} \text{datatype} & \tau & ::= & \cdots \mid \mathsf{bool} \\ \text{term} & t & ::= & \cdots \mid \mathbf{true} \mid \mathbf{false} \end{array}$$

Introduction and elimination rules using hypothetical judgments:

$$\frac{}{\Gamma \vdash \mathbf{true} \in \mathsf{bool}}\ \mathsf{bool}_{\mathsf{t}} \qquad \frac{}{\Gamma \vdash \mathbf{false} \in \mathsf{bool}}\ \mathsf{bool}_{\mathsf{f}} \qquad \frac{\Gamma \vdash t \in \mathsf{bool} \quad \Gamma \vdash t_1 \in \tau \quad \Gamma \vdash t_2 \in \tau}{\Gamma \vdash \mathbf{if}\ t\ \mathbf{then}\ t_1\ \mathbf{else}\ t_2 \in \tau}\ \mathsf{boolE}$$

Local reductions of terms:

$$\begin{array}{lcl} \mathbf{if\ true\ then}\ t_1\ \mathbf{else}\ t_2 & \Longrightarrow_R & t_1 \\ \mathbf{if\ false\ then}\ t_1\ \mathbf{else}\ t_2 & \Longrightarrow_R & t_2 \end{array}$$

Examples:

$$
\begin{array}{l}
\mathit{and} \in \mathsf{bool} \to \mathsf{bool} \to \mathsf{bool} \\
\mathit{and} \;=\; \lambda x \in \mathsf{bool}.\, \lambda y \in \mathsf{bool}.\, \mathbf{if}\ x\ \mathbf{then}\ y\ \mathbf{else}\ \mathbf{false} \\
\mathit{or} \in \mathsf{bool} \to \mathsf{bool} \to \mathsf{bool} \\
\mathit{or} \;=\; \lambda x \in \mathsf{bool}.\, \lambda y \in \mathsf{bool}.\, \mathbf{if}\ x\ \mathbf{then}\ \mathbf{true}\ \mathbf{else}\ y \\
\mathit{not} \in \mathsf{bool} \to \mathsf{bool} \\
\mathit{not} \;=\; \lambda x \in \mathsf{bool}.\, \mathbf{if}\ x\ \mathbf{then}\ \mathbf{false}\ \mathbf{else}\ \mathbf{true}
\end{array}
$$

Lists

Syntax:

$$
\begin{array}{lrcl}
\text{datatype} & \tau & ::= & \cdots \mid \mathsf{list}\ \tau \\
\text{term} & t & ::= & \cdots \mid \mathbf{nil}^{\tau} \mid t :: t
\end{array}
$$

Formation rule:

$$
\frac{\tau\ \mathsf{type}}{\mathsf{list}\ \tau\ \mathsf{type}}\ \mathsf{list\,F}
$$

Introduction rules:

$$
\frac{}{\Gamma \vdash \mathbf{nil}^{\tau} \in \mathsf{list}\ \tau}\ \mathsf{list\,I_n} \qquad \frac{\Gamma \vdash t \in \tau \quad \Gamma \vdash s \in \mathsf{list}\ \tau}{\Gamma \vdash t :: s \in \mathsf{list}\ \tau}\ \mathsf{list\,I_c}
$$

The elimination rule is based on primitive recursion:

$$
\frac{\Gamma \vdash t \in \mathsf{list}\ \tau \quad \Gamma \vdash s_n \in \sigma \quad \Gamma, x \in \tau, l \in \mathsf{list}\ \tau, f(l) \in \sigma \vdash s_c \in \sigma}{\Gamma \vdash \mathbf{rec}\ f(t)\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c \in \sigma}\ \mathsf{list\,E}
$$

In the rule $\mathsf{list\,E}$, $f(l)$ is regarded as a variable.

Local reductions of terms:

$$
\begin{array}{lcl}
\mathbf{rec}\ f(\mathbf{nil}^{\tau})\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c & \Longrightarrow_R & s_n \\
\mathbf{rec}\ f(t :: t')\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c & \Longrightarrow_R & [\mathbf{rec}\ f(t')\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c / f(l)][t'/l][t/x]s_c
\end{array}
$$

Examples:

$$
\begin{array}{rcl}
\mathit{append}\ \mathbf{nil}^{\tau}\ t & = & t \\
\mathit{append}\ (x :: l)\ t & = & x :: (\mathit{append}\ l\ t)
\end{array}
$$

$$
\begin{array}{l}
\mathit{append} \in \mathsf{list}\ \tau \to \mathsf{list}\ \tau \to \mathsf{list}\ \tau \\
\mathit{append} \;=\; \lambda y \in \mathsf{list}\ \tau.\, \lambda z \in \mathsf{list}\ \tau.\, \mathbf{rec}\ f(y)\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow z \mid f(x :: l) \Rightarrow x :: f(l)
\end{array}
$$

$$
\begin{array}{rcl}
\mathit{length}\ \mathbf{nil}^{\tau} & = & \mathbf{0} \\
\mathit{length}\ (x :: l) & = & \mathbf{s}(\mathit{length}\ l)
\end{array}
$$

$$
\begin{array}{l}
\mathit{length} \in \mathsf{list}\ \tau \to \mathsf{nat} \\
\mathit{length} \;=\; \lambda y \in \mathsf{list}\ \tau.\, \mathbf{rec}\ f(y)\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow \mathbf{0} \mid f(x :: l) \Rightarrow \mathbf{s}(f(l))
\end{array}
$$

1.5 Predicates on terms

We wish to express properties of terms using predicates on terms. As an example, we consider a predicate LT such that $LT(m, n)$ means that m is less than n. We abbreviate $LT(m, n)$ as $m < n$.

$$
\text{proposition} \quad A \quad ::= \quad \cdots \mid m < n
$$

$$
\frac{m \in \mathsf{nat} \quad n \in \mathsf{nat}}{m < n\ \mathit{prop}}\ {<}\mathsf{F}
$$

Again we use natural deduction to define the judgment $m < n\ true$. By the rule $<\mathsf{F}$, a judgment $m < n\ true$ implicitly assumes that both m and n are of datatype nat.

Introduction rules:

$$\frac{}{\mathbf{0} < \mathbf{s}(n)\ true} <\mathsf{I}_0 \qquad \frac{m < n\ true}{\mathbf{s}(m) < \mathbf{s}(n)\ true} <\mathsf{I}_\mathsf{s}$$

In order to design elimination rules, we consider four possible cases of the judgment $m < n\ true$:

- $\mathbf{0} < \mathbf{0}\ true$ is impossible to prove. The corresponding elimination rule deduces any judgment $C\ true$.
- $\mathbf{s}(m) < \mathbf{0}\ true$ is impossible to prove. The corresponding elimination rule deduces any judgment $C\ true$.
- $\mathbf{0} < \mathbf{s}(n)\ true$ holds trivially by the rule $<\mathsf{I}_0$ whose premise is empty. Hence there is no corresponding elimination rule.
- $\mathbf{s}(m) < \mathbf{s}(n)\ true$ holds by the rule $<\mathsf{I}_\mathsf{s}$ whose premise is $m < n\ true$. Thus the corresponding elimination rule deduces $m < n\ true$.

We combine the first two cases to obtain a single elimination rule:

$$\frac{m < \mathbf{0}\ true}{C\ true} <\mathsf{E}_0 \qquad \frac{\mathbf{s}(m) < \mathbf{s}(n)\ true}{m < n\ true} <\mathsf{E}_\mathsf{s}$$

Note that the rules $<\mathsf{I}_\mathsf{s}$ and $<\mathsf{E}_\mathsf{s}$ have nothing to do with orthogonality of the system, since $<$ is not a connective but a predicate.

Using hypothetical judgments:

$$\frac{}{\Gamma \vdash \mathbf{0} < \mathbf{s}(n)\ true} <\mathsf{I}_0 \qquad \frac{\Gamma \vdash m < n\ true}{\Gamma \vdash \mathbf{s}(m) < \mathbf{s}(n)\ true} <\mathsf{I}_\mathsf{s}$$

$$\frac{\Gamma \vdash m < \mathbf{0}\ true}{\Gamma \vdash C\ true} <\mathsf{E}_0 \qquad \frac{\Gamma \vdash \mathbf{s}(m) < \mathbf{s}(n)\ true}{\Gamma \vdash m < n\ true} <\mathsf{E}_\mathsf{s}$$

Examples:

$$\dfrac{\dfrac{}{\mathbf{0} < \mathbf{s}(\mathbf{0})\ true} <\mathsf{I}_0}{\mathbf{s}(\mathbf{0}) < \mathbf{s}(\mathbf{s}(\mathbf{0}))\ true} <\mathsf{I}_\mathsf{s} \qquad \dfrac{\dfrac{\dfrac{}{m \in \mathsf{nat}, m < \mathbf{0}\ true \vdash m < \mathbf{0}\ true}\,\mathsf{Hyp}}{m \in \mathsf{nat}, m < \mathbf{0}\ true \vdash \bot\ true} <\mathsf{E}_0}{m \in \mathsf{nat} \vdash \neg(m < \mathbf{0})\ true} \supset\mathsf{I}$$

Equality

We consider another predicate $EQ(m, n)$ to mean that natural numbers m and n are equal. We abbreviate $EQ(m, n)$ as $m =_\mathsf{N} n$.

$$\text{proposition} \quad A \quad ::= \quad \cdots \mid m =_\mathsf{N} n$$

Formation rule:

$$\frac{m \in \mathsf{nat} \quad n \in \mathsf{nat}}{m =_\mathsf{N} n\ prop} =_\mathsf{N}\mathsf{F}$$

Introduction and elimination rules:

$$\frac{}{\mathbf{0} =_\mathsf{N} \mathbf{0}\ true} =_\mathsf{N}\mathsf{I}_0 \qquad \frac{m =_\mathsf{N} n\ true}{\mathbf{s}(m) =_\mathsf{N} \mathbf{s}(n)\ true} =_\mathsf{N}\mathsf{I}_\mathsf{s}$$

$$\frac{\mathbf{0} =_\mathsf{N} \mathbf{s}(n)\ true}{C\ true} =_\mathsf{N}\mathsf{E}_{0\mathsf{s}} \qquad \frac{\mathbf{s}(m) =_\mathsf{N} \mathbf{0}\ true}{C\ true} =_\mathsf{N}\mathsf{E}_{\mathsf{s}0} \qquad \frac{\mathbf{s}(m) =_\mathsf{N} \mathbf{s}(n)\ true}{m =_\mathsf{N} n\ true} =_\mathsf{N}\mathsf{E}_\mathsf{s}$$

There is no elimination rule for $\mathbf{0} =_\mathsf{N} \mathbf{0}\ true$ because the premise of the rule $=_\mathsf{N}\mathsf{I}_0$ is empty.

Using hypothetical judgments:

$$\frac{}{\Gamma \vdash \mathbf{0} =_{\mathsf{N}} \mathbf{0}\ true}\ =_{\mathsf{N}}\mathsf{I}_0 \qquad \frac{\Gamma \vdash m =_{\mathsf{N}} n\ true}{\Gamma \vdash \mathbf{s}(m) =_{\mathsf{N}} \mathbf{s}(n)\ true}\ =_{\mathsf{N}}\mathsf{I}_\mathsf{s}$$

$$\frac{\Gamma \vdash \mathbf{0} =_{\mathsf{N}} \mathbf{s}(n)\ true}{\Gamma \vdash C\ true}\ =_{\mathsf{N}}\mathsf{E}_{0\mathsf{s}} \qquad \frac{\Gamma \vdash \mathbf{s}(m) =_{\mathsf{N}} \mathbf{0}\ true}{\Gamma \vdash C\ true}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{s}0} \qquad \frac{\Gamma \vdash \mathbf{s}(m) =_{\mathsf{N}} \mathbf{s}(n)\ true}{\Gamma \vdash m =_{\mathsf{N}} n\ true}\ =_{\mathsf{N}}\mathsf{E}_\mathsf{s}$$

1.6 Proof terms for predicates

Syntax:

$$\begin{array}{rrcl} \text{proof term} & M & ::= & \cdots \mid \mathbf{ltI_0} \mid \mathbf{ltI_s}(M) \mid \mathbf{ltE_0}(M) \mid \mathbf{ltE_s}(M) \mid \\ & & & \mathbf{eqI_0} \mid \mathbf{eqI_s}(M) \mid \mathbf{eqE_{0s}}(M) \mid \mathbf{eqE_{s0}}(M) \mid \mathbf{eqE_s}(M) \end{array}$$

Proof terms for the predicate $LT(m, n)$:

$$\frac{}{\Gamma \vdash \mathbf{ltI_0} : \mathbf{0} < \mathbf{s}(n)}\ <\mathsf{I}_0 \qquad \frac{\Gamma \vdash M : m < n}{\Gamma \vdash \mathbf{ltI_s}(M) : \mathbf{s}(m) < \mathbf{s}(n)}\ <\mathsf{I}_\mathsf{s}$$

$$\frac{\Gamma \vdash M : m < \mathbf{0}}{\Gamma \vdash \mathbf{ltE_0}(M) : C}\ <\mathsf{E}_0 \qquad \frac{\Gamma \vdash M : \mathbf{s}(m) < \mathbf{s}(n)}{\Gamma \vdash \mathbf{ltE_s}(M) : m < n}\ <\mathsf{E}_\mathsf{s}$$

Proof terms for the predicate $EQ(m, n)$:

$$\frac{}{\Gamma \vdash \mathbf{eqI_0} : \mathbf{0} =_{\mathsf{N}} \mathbf{0}}\ =_{\mathsf{N}}\mathsf{I}_0 \qquad \frac{\Gamma \vdash M : m =_{\mathsf{N}} n}{\Gamma \vdash \mathbf{eqI_s}(M) : \mathbf{s}(m) =_{\mathsf{N}} \mathbf{s}(n)}\ =_{\mathsf{N}}\mathsf{I}_\mathsf{s}$$

$$\frac{\Gamma \vdash M : \mathbf{0} =_{\mathsf{N}} \mathbf{s}(n)}{\Gamma \vdash \mathbf{eqE_{0s}}(M) : C}\ =_{\mathsf{N}}\mathsf{E}_{0\mathsf{s}} \qquad \frac{\Gamma \vdash M : \mathbf{s}(m) =_{\mathsf{N}} \mathbf{0}}{\Gamma \vdash \mathbf{eqE_{s0}}(M) : C}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{s}0} \qquad \frac{\Gamma \vdash M : \mathbf{s}(m) =_{\mathsf{N}} \mathbf{s}(n)}{\Gamma \vdash \mathbf{eqE_s}(M) : m =_{\mathsf{N}} n}\ =_{\mathsf{N}}\mathsf{E}_\mathsf{s}$$

Examples:

$$\frac{\dfrac{}{\mathbf{ltI_0} : \mathbf{0} < \mathbf{s}(\mathbf{0})}\ <\mathsf{I}_0}{\mathbf{ltI_s}(\mathbf{ltI_0}) : \mathbf{s}(\mathbf{0}) < \mathbf{s}(\mathbf{s}(\mathbf{0}))}\ <\mathsf{I}_\mathsf{s} \qquad \frac{\dfrac{\dfrac{}{m \in \mathsf{nat}, z : m < \mathbf{0} \vdash z : m < \mathbf{0}}\ \mathsf{Hyp}}{m \in \mathsf{nat}, z : m < \mathbf{0} \vdash \mathbf{ltE_0}(z) : \bot}\ <\mathsf{E}_0}{m \in \mathsf{nat} \vdash \lambda z{:}m < \mathbf{0}.\, \mathbf{ltE_0}(z) : \neg(m < \mathbf{0})}\ \supset\mathsf{I}$$

1.7 Induction

In order to allow mathematical induction on natural numbers inside a proof, we introduce another elimination rule for nat:

$$\frac{\begin{array}{ccc} & \overline{x \in \mathsf{nat}} & \overline{A(x)\ true} \\ & & \vdots \\ t \in \mathsf{nat} \quad A(\mathbf{0})\ true & & A(\mathbf{s}(x))\ true \end{array}}{A(t)\ true}\ \mathsf{natE}_I$$

The second premise states that $A(x)\ true$ holds for $x = \mathbf{0}$, and corresponds to the base case in mathematical induction. The third premise states that an assumption of $A(x)\ true$ leads to a proof of $A(\mathbf{s}(x))\ true$, and corresponds to the inductive case in mathematical induction. Hence the second and third premises constitute a valid proof of $A(x)\ true$ for every natural number x. The first premise provides a specific natural number t to be substituted for x in $A(x)\ true$; hence t is not essential in completing a proof by mathematical induction. Often t is just a variable, in which case it is called an *induction variable*.

Using hypothetical judgments:

$$\frac{\Gamma \vdash t \in \mathsf{nat} \quad \Gamma \vdash A(\mathbf{0})\ \mathit{true} \quad \Gamma, x \in \mathsf{nat}, A(x)\ \mathit{true} \vdash A(\mathbf{s}(x))\ \mathit{true}}{\Gamma \vdash A(t)\ \mathit{true}}\ \mathsf{natE}_I$$

Proof term:

$$\frac{\Gamma \vdash t \in \mathsf{nat} \quad \Gamma \vdash M : A(\mathbf{0}) \quad \Gamma, x \in \mathsf{nat}, u(x) : A(x) \vdash N : A(\mathbf{s}(x))}{\Gamma \vdash \mathbf{ind}\ u(t)\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N : A(t)}\ \mathsf{natE}_I$$

If $\mathbf{ind}\ u(t)\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N$ does not use $u(x)$ in N, it degenerates to case analysis and is written as $\mathbf{case}\ t\ \mathbf{of}\ \mathbf{0} \Rightarrow M \mid \mathbf{s}(x) \Rightarrow N$ (which omits variable u):

$$\frac{\Gamma \vdash t \in \mathsf{nat} \quad \Gamma \vdash M : A(\mathbf{0}) \quad \Gamma, x \in \mathsf{nat} \vdash N : A(\mathbf{s}(x))}{\Gamma \vdash \mathbf{case}\ t\ \mathbf{of}\ \mathbf{0} \Rightarrow M \mid \mathbf{s}(x) \Rightarrow N : A(t)}\ \mathsf{natE}_I$$

Local reductions of proof terms:

$$\begin{array}{rcl} \mathbf{ind}\ u(\mathbf{0})\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N & \Longrightarrow_R & M \\ \mathbf{ind}\ u(\mathbf{s}(t))\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N & \Longrightarrow_R & [\mathbf{ind}\ u(t)\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N/u(x)][t/x]N \end{array}$$

Example

We let $A(x) = x < \mathbf{s}(x)$.

$$\frac{\Gamma \vdash t \in \mathsf{nat} \quad \dfrac{}{\Gamma \vdash \mathbf{0} < \mathbf{s}(\mathbf{0})\ \mathit{true}}\ <\mathsf{I}_0 \quad \dfrac{\dfrac{}{\Gamma, x \in \mathsf{nat}, x < \mathbf{s}(x)\ \mathit{true} \vdash x < \mathbf{s}(x)\ \mathit{true}}\ \mathsf{Hyp}}{\Gamma, x \in \mathsf{nat}, x < \mathbf{s}(x)\ \mathit{true} \vdash \mathbf{s}(x) < \mathbf{s}(\mathbf{s}(x))\ \mathit{true}}\ <\mathsf{I}_\mathsf{s}}{\Gamma \vdash t < \mathbf{s}(t)\ \mathit{true}}\ \mathsf{natE}_I$$

Proof term for $t < \mathbf{s}(t)\ \mathit{true}$:

$$\frac{\Gamma \vdash t \in \mathsf{nat} \quad \dfrac{}{\Gamma \vdash \mathbf{ltI_0} : \mathbf{0} < \mathbf{s}(\mathbf{0})}\ <\mathsf{I}_0 \quad \dfrac{\dfrac{}{\Gamma, x \in \mathsf{nat}, u(x) : x < \mathbf{s}(x) \vdash u(x) : x < \mathbf{s}(x)}\ \mathsf{Hyp}}{\Gamma, x \in \mathsf{nat}, u(x) : x < \mathbf{s}(x) \vdash \mathbf{ltI_s}(u(x)) : \mathbf{s}(x) < \mathbf{s}(\mathbf{s}(x))}\ <\mathsf{I}_\mathsf{s}}{\Gamma \vdash \mathbf{ind}\ u(t)\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow \mathbf{ltI_0} \mid u(\mathbf{s}(x)) \Rightarrow \mathbf{ltI_s}(u(x)) : t < \mathbf{s}(t)}\ \mathsf{natE}_I$$

1.8 First-order logic with datatypes

Formation rules:

$$\frac{\begin{array}{c}\overline{x \in \tau} \\ \vdots \\ A(x)\ \mathit{prop}\end{array}}{\forall x \in \tau . A(x)\ \mathit{prop}}\ \forall\mathsf{F} \qquad \frac{\begin{array}{c}\overline{x \in \tau} \\ \vdots \\ A(x)\ \mathit{prop}\end{array}}{\exists x \in \tau . A(x)\ \mathit{prop}}\ \exists\mathsf{F}$$

Introduction and elimination rules:

$$\frac{\begin{array}{c}\overline{x \in \tau} \\ \vdots \\ A(x)\ \mathit{true}\end{array}}{\forall x \in \tau . A(x)\ \mathit{true}}\ \forall\mathsf{I} \qquad \frac{\forall x \in \tau . A(x)\ \mathit{true} \quad t \in \tau}{A(t)\ \mathit{true}}\ \forall\mathsf{E}$$

$$\frac{t \in \tau \quad A(t)\ true}{\exists x{\in}\tau.A(x)\ true}\ \exists\mathsf{I} \qquad \frac{\exists x{\in}\tau.A(x)\ true \qquad \begin{array}{c}\overline{x \in \tau} \quad \overline{A(x)\ true}^{\,w} \\ \vdots \\ C\ true\end{array}}{C\ true}\ \exists\mathsf{E}^w$$

Using hypothetical judgments:

$$\frac{\Gamma, x \in \tau \vdash A(x)\ true}{\Gamma \vdash \forall x{\in}\tau.A(x)\ true}\ \forall\mathsf{I} \qquad \frac{\Gamma \vdash \forall x{\in}\tau.A(x)\ true \quad \Gamma \vdash t \in \tau}{\Gamma \vdash A(t)\ true}\ \forall\mathsf{E}$$

$$\frac{\Gamma \vdash t \in \tau \quad \Gamma \vdash A(t)\ true}{\Gamma \vdash \exists x{\in}\tau.A(x)\ true}\ \exists\mathsf{I} \qquad \frac{\Gamma \vdash \exists x{\in}\tau.A(x)\ true \quad \Gamma, x \in \tau, A(x)\ true \vdash C\ true}{\Gamma \vdash C\ true}\ \exists\mathsf{E}$$

Proof terms:

$$\text{proof term} \quad M \quad ::= \quad \cdots \mid \lambda x \in \tau.\, M \mid M\ t \mid \langle t, M\rangle \mid \mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ N$$

$$\frac{\begin{array}{c}\overline{x \in \tau} \\ \vdots \\ M : A(x)\end{array}}{\lambda x \in \tau.\, M : \forall x{\in}\tau.A(x)}\ \forall\mathsf{I} \qquad \frac{M : \forall x{\in}\tau.A(x) \quad t \in \tau}{M\ t : A(t)}\ \forall\mathsf{E}$$

$$\frac{t \in \tau \quad M : A(t)}{\langle t, M\rangle : \exists x{\in}\tau.A(x)}\ \exists\mathsf{I} \qquad \frac{M : \exists x{\in}\tau.A(x) \qquad \begin{array}{c}\overline{x \in \tau} \quad \overline{w : A(x)} \\ \vdots \\ N : C\end{array}}{\mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ N : C}\ \exists\mathsf{E}$$

Using hypothetical judgments:

$$\frac{\Gamma, x \in \tau \vdash M : A(x)}{\Gamma \vdash \lambda x \in \tau.\, M : \forall x{\in}\tau.A(x)}\ \forall\mathsf{I} \qquad \frac{\Gamma \vdash M : \forall x{\in}\tau.A(x) \quad \Gamma \vdash t \in \tau}{\Gamma \vdash M\ t : A(t)}\ \forall\mathsf{E}$$

$$\frac{\Gamma \vdash t \in \tau \quad \Gamma \vdash M : A(t)}{\Gamma \vdash \langle t, M\rangle : \exists x{\in}\tau.A(x)}\ \exists\mathsf{I} \qquad \frac{\Gamma \vdash M : \exists x{\in}\tau.A(x) \quad \Gamma, x \in \tau, w : A(x) \vdash N : C}{\Gamma \vdash \mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ N : C}\ \exists\mathsf{E}$$

Local reduction of proofs:

$$\frac{\begin{array}{c}\overline{x \in \tau} \\ \vdots \\ M : A(x)\end{array}}{\dfrac{\lambda x \in \tau.\, M : \forall x{\in}\tau.A(x)}{(\lambda x \in \tau.\, M)\ t : A(t)}\ \forall\mathsf{I} \quad t \in \tau}\ \exists\mathsf{E} \qquad \Longrightarrow_R \qquad \begin{array}{c}\overline{t \in \tau} \\ \vdots \\ [t/x]M : A(t)\end{array}$$

$$\frac{t \in \tau \quad M : A(t)}{\dfrac{\langle t, M\rangle : \exists x{\in}\tau.A(x)}{\mathsf{let}\ \langle x, w\rangle = \langle t, M\rangle\ \mathsf{in}\ N : C}\ \exists\mathsf{I} \qquad \begin{array}{c}\overline{x \in \tau} \quad \overline{w : A(x)} \\ \vdots \\ N : C\end{array}}\ \exists\mathsf{E} \qquad \Longrightarrow_R \qquad \begin{array}{c}\overline{t \in \tau} \quad \overline{[M/w]w : A(t)} \\ \vdots \\ [M/w][t/x]N : C\end{array}$$

Local reduction of proof terms:

$$\begin{array}{rcl}(\lambda x \in \tau.\, M)\ t & \Longrightarrow_R & [t/x]M \\ \mathsf{let}\ \langle x, w\rangle = \langle t, M\rangle\ \mathsf{in}\ N & \Longrightarrow_R & [M/w][t/x]N\end{array}$$

Local expansions of proof terms:

$$\begin{array}{lll} M : \forall x{\in}\tau.A & \Longrightarrow_E & \lambda x \in \tau.\, M\ x \qquad \textit{(x is not free in M)} \\ M : \exists x{\in}\tau.A & \Longrightarrow_E & \mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ \langle x, w\rangle \end{array}$$

1.9 Examples

1.9.1 $(\forall x{\in}\tau.A(x) \wedge B(x)) \supset \forall x{\in}\tau.A(x)$

A proof term of type $(\forall x{\in}\tau.A(x) \wedge B(x)) \supset \forall x{\in}\tau.A(x)$:

$$\lambda z{:}\forall x{\in}\tau.A(x) \wedge B(x).\ \lambda x \in \tau.\ \mathsf{fst}\ (z\ x)$$

1.9.2 $\exists x{\in}\tau.A(x) \vee B(x) \equiv (\exists x{\in}\tau.A(x)) \vee (\exists x{\in}\tau.B(x))$

A proof term of type $(\exists x{\in}\tau.A(x) \vee B(x)) \supset ((\exists x{\in}\tau.A(x)) \vee (\exists x{\in}\tau.B(x)))$:

$$\lambda z{:}\exists x{\in}\tau.A(x) \vee B(x).\ \mathsf{let}\ \langle x, w\rangle = z\ \mathsf{in}\ \mathsf{case}\ w\ \mathsf{of}\ \mathsf{inl}\ y_1 \Rightarrow \mathsf{inl}_{\exists x\in\tau.B(x)}\ \langle x, y_1\rangle \mid \mathsf{inr}\ y_2 \Rightarrow \mathsf{inr}_{\exists x\in\tau.A(x)}\ \langle x, y_2\rangle$$

That is, $\exists$ distributes over $\vee$.

A proof term of type $((\exists x{\in}\tau.A(x)) \vee (\exists x{\in}\tau.B(x))) \supset (\exists x{\in}\tau.A(x) \vee B(x))$:

$$\begin{array}{l} \lambda z{:}(\exists x{\in}\tau.A(x)) \vee (\exists x{\in}\tau.B(x)). \\ \quad \mathsf{case}\ z\ \mathsf{of}\ \mathsf{inl}\ y_1 \Rightarrow \mathsf{let}\ \langle x, w\rangle = y_1\ \mathsf{in}\ \langle x, \mathsf{inl}_{B(x)}\ w\rangle \mid \mathsf{inr}\ y_2 \Rightarrow \mathsf{let}\ \langle x, w\rangle = y_2\ \mathsf{in}\ \langle x, \mathsf{inr}_{A(x)}\ w\rangle \end{array}$$

1.9.3 $\forall x{\in}\mathsf{nat}.x =_{\mathsf{N}} x$

A mathematical proof of $\forall x{\in}\mathsf{nat}.x =_{\mathsf{N}} x\ \mathit{true}$:

Proof. By induction on x.

Base case $x = \mathbf{0}$:

$\mathbf{0} =_{\mathsf{N}} \mathbf{0}\ \mathit{true}$ from $\dfrac{}{\mathbf{0} =_{\mathsf{N}} \mathbf{0}\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{I}_0$

Inductive case $x = \mathbf{s}(x')$:

$x' =_{\mathsf{N}} x'\ \mathit{true}$ by induction hypothesis

$\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(x')\ \mathit{true}$ from $\dfrac{x' =_{\mathsf{N}} x'\ \mathit{true}}{\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(x')\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{I}_{\mathsf{s}}$

□

The specification for a proof term eqNat of type $\forall x{\in}\mathsf{nat}.x =_{\mathsf{N}} x$:

$$\begin{array}{llcl} \mathit{eqNat} & \mathbf{0} & = & \mathbf{eqI_0} \\ \mathit{eqNat} & \mathbf{s}(z) & = & \mathbf{eqI_s}(\mathit{eqNat}\ z) \end{array}$$

Proof term eqNat:

$$\mathit{eqNat}\ =\ \lambda x \in \mathsf{nat}.\, \mathbf{ind}\ u(x)\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow \mathbf{eqI_0} \mid u(\mathbf{s}(z)) \Rightarrow \mathbf{eqI_s}(u(z))$$

1.9.4 $\forall x \in \mathsf{nat}.\forall y \in \mathsf{nat}.\forall z \in \mathsf{nat}.x =_{\mathsf{N}} y \supset y =_{\mathsf{N}} z \supset x =_{\mathsf{N}} z$

A mathematical proof of $\forall x \in \mathsf{nat}.\forall y \in \mathsf{nat}.\forall z \in \mathsf{nat}.x =_{\mathsf{N}} y \supset y =_{\mathsf{N}} z \supset x =_{\mathsf{N}} z\ \mathit{true}$:

Proof. By induction on x. We consider subcases on y and z. In each case, we assume $x =_{\mathsf{N}} y\ \mathit{true}$ and $y =_{\mathsf{N}} z\ \mathit{true}$ to show $x =_{\mathsf{N}} z\ \mathit{true}$.

Base case $x = \mathbf{0}$. We need to show $\forall y \in \mathsf{nat}.\forall z \in \mathsf{nat}.\mathbf{0} =_{\mathsf{N}} y \supset y =_{\mathsf{N}} z \supset \mathbf{0} =_{\mathsf{N}} z\ \mathit{true}$:

Subcase $y = \mathbf{0}$:

Subcase $z = \mathbf{0}$. We need to show $\mathbf{0} =_{\mathsf{N}} \mathbf{0}\ \mathit{true}$:

$\mathbf{0} =_{\mathsf{N}} \mathbf{0}\ \mathit{true}$ from $\dfrac{}{\mathbf{0} =_{\mathsf{N}} \mathbf{0}\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{I}_{\mathbf{0}}$

Subcase $z = \mathbf{s}(z')$.

$\mathbf{0} =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}$ from the assumption $y =_{\mathsf{N}} z\ \mathit{true}$

$x =_{\mathsf{N}} z\ \mathit{true}$ from $\dfrac{\mathbf{0} =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}}{x =_{\mathsf{N}} z\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{0s}}$

Subcase $y = \mathbf{s}(y')$:

$\mathbf{0} =_{\mathsf{N}} \mathbf{s}(y')\ \mathit{true}$ from the assumption $x =_{\mathsf{N}} y\ \mathit{true}$

$x =_{\mathsf{N}} z\ \mathit{true}$ from $\dfrac{\mathbf{0} =_{\mathsf{N}} \mathbf{s}(y')\ \mathit{true}}{x =_{\mathsf{N}} z\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{0s}}$

Inductive case $x = \mathbf{s}(x')$. We need to show $\forall y \in \mathsf{nat}.\forall z \in \mathsf{nat}.\mathbf{s}(x') =_{\mathsf{N}} y \supset y =_{\mathsf{N}} z \supset \mathbf{s}(x') =_{\mathsf{N}} z\ \mathit{true}$:

$\forall y' \in \mathsf{nat}.\forall z' \in \mathsf{nat}.x' =_{\mathsf{N}} y' \supset y' =_{\mathsf{N}} z' \supset x' =_{\mathsf{N}} z'\ \mathit{true}$ by induction hypothesis

Subcase $y = \mathbf{0}$:

$\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}\ \mathit{true}$ from the assumption $x =_{\mathsf{N}} y\ \mathit{true}$

$x =_{\mathsf{N}} z\ \mathit{true}$ from $\dfrac{\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}\ \mathit{true}}{x =_{\mathsf{N}} z\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{s0}}$

Subcase $y = \mathbf{s}(y')$:

Subcase $z = \mathbf{0}$:

$\mathbf{s}(y') =_{\mathsf{N}} \mathbf{0}\ \mathit{true}$ from the assumption $y =_{\mathsf{N}} z\ \mathit{true}$

$x =_{\mathsf{N}} z\ \mathit{true}$ from $\dfrac{\mathbf{s}(y') =_{\mathsf{N}} \mathbf{0}\ \mathit{true}}{x =_{\mathsf{N}} z\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{s0}}$

Subcase $z = \mathbf{s}(z')$. We need to show $\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}$:

$\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y')\ \mathit{true}$ from the assumption $x =_{\mathsf{N}} y\ \mathit{true}$

$x' =_{\mathsf{N}} y'\ \mathit{true}$ from $\dfrac{\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y')\ \mathit{true}}{x' =_{\mathsf{N}} y'\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{s}}$

$\mathbf{s}(y') =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}$ from the assumption $y =_{\mathsf{N}} z\ \mathit{true}$

$y' =_{\mathsf{N}} z'\ \mathit{true}$ from $\dfrac{\mathbf{s}(y') =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}}{y' =_{\mathsf{N}} z'\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{E}_{\mathsf{s}}$

$x' =_{\mathsf{N}} z'\ \mathit{true}$ from $\forall y' \in \mathsf{nat}.\forall z' \in \mathsf{nat}.x' =_{\mathsf{N}} y' \supset y' =_{\mathsf{N}} z' \supset x' =_{\mathsf{N}} z'\ \mathit{true}, x' =_{\mathsf{N}} y'\ \mathit{true}, y' =_{\mathsf{N}} z'\ \mathit{true}$

$\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}$ from $\dfrac{x' =_{\mathsf{N}} z'\ \mathit{true}}{\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(z')\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{I}_{\mathsf{s}}$

□

The specification for a proof term trans of type $\forall x \in \mathsf{nat}.\forall y \in \mathsf{nat}.\forall z \in \mathsf{nat}.x =_{\mathsf{N}} y \supset y =_{\mathsf{N}} z \supset x =_{\mathsf{N}} z$:

trans	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$v : \mathbf{0} =_{\mathsf{N}} \mathbf{0}$	$w : \mathbf{0} =_{\mathsf{N}} \mathbf{0}$	$=$	$\mathbf{eqI_0}$
trans	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{s}(z')$	$v : \mathbf{0} =_{\mathsf{N}} \mathbf{0}$	$w : \mathbf{0} =_{\mathsf{N}} \mathbf{s}(z')$	$=$	$\mathbf{eqE_{0s}}(w)$
trans	$\mathbf{0}$	$\mathbf{s}(y')$	z	$v : \mathbf{0} =_{\mathsf{N}} \mathbf{s}(y')$	$w : \mathbf{s}(y') =_{\mathsf{N}} z$	$=$	$\mathbf{eqE_{0s}}(v)$
trans	$\mathbf{s}(x')$	$\mathbf{0}$	z	$v : \mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}$	$w : \mathbf{0} =_{\mathsf{N}} z$	$=$	$\mathbf{eqE_{s0}}(v)$
trans	$\mathbf{s}(x')$	$\mathbf{s}(y')$	$\mathbf{0}$	$v : \mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y')$	$w : \mathbf{s}(y') =_{\mathsf{N}} \mathbf{0}$	$=$	$\mathbf{eqE_{s0}}(w)$
trans	$\mathbf{s}(x')$	$\mathbf{s}(y')$	$\mathbf{s}(z')$	$v : \mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y')$	$w : \mathbf{s}(y') =_{\mathsf{N}} \mathbf{s}(z')$	$=$	$\mathbf{eqI_s}(\mathit{trans}\ x'\ y'\ z'\ \mathbf{eqE_s}(v)\ \mathbf{eqE_s}(w))$

Wrong definitions of proof term *trans*:

$$\lambda x \in \mathsf{nat}.\, \lambda y \in \mathsf{nat}.\, \lambda z \in \mathsf{nat}.$$
$$\mathbf{ind}\ u(x)\ \mathbf{of} \begin{cases} u(\mathbf{0}) \Rightarrow \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqI_0} \\ \mathbf{s}(z') \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{0s}}(w) \end{cases} \\ \mathbf{s}(y') \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{0s}}(v) \end{cases} \\ u(\mathbf{s}(x')) \Rightarrow \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{s0}}(v) \\ \mathbf{s}(y') \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{s0}}(w) \\ \mathbf{s}(z') \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \\ \qquad \mathbf{eqI_s}(u(x')\ y'\ z'\ \mathbf{eqE_s}(v)\ \mathbf{eqE_s}(w)) \end{cases} \end{cases} \end{cases}$$

$$\lambda x \in \mathsf{nat}.\, \lambda y \in \mathsf{nat}.\, \lambda z \in \mathsf{nat}.$$
$$\mathbf{ind}\ u(x)\ \mathbf{of} \begin{cases} u(\mathbf{0}) \Rightarrow \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}\mathbf{0} =_{\mathsf{N}} \mathbf{0}.\, \lambda w{:}\mathbf{0} =_{\mathsf{N}} \mathbf{0}.\, \mathbf{eqI_0} \\ \mathbf{s}(z') \Rightarrow \lambda v{:}\mathbf{0} =_{\mathsf{N}} \mathbf{0}.\, \lambda w{:}\mathbf{0} =_{\mathsf{N}} \mathbf{s}(z').\, \mathbf{eqE_{0s}}(w) \end{cases} \\ \mathbf{s}(y') \Rightarrow \lambda v{:}\mathbf{0} =_{\mathsf{N}} \mathbf{s}(y').\, \lambda w{:}\mathbf{s}(y') =_{\mathsf{N}} z.\, \mathbf{eqE_{0s}}(v) \end{cases} \\ u(\mathbf{s}(x')) \Rightarrow \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}.\, \lambda w{:}\mathbf{0} =_{\mathsf{N}} z.\, \mathbf{eqE_{s0}}(v) \\ \mathbf{s}(y') \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y').\, \lambda w{:}\mathbf{s}(y') =_{\mathsf{N}} \mathbf{0}.\, \mathbf{eqE_{s0}}(w) \\ \mathbf{s}(z') \Rightarrow \lambda v{:}\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y').\, \lambda w{:}\mathbf{s}(y') =_{\mathsf{N}} \mathbf{s}(z').\, \\ \qquad \mathbf{eqI_s}(u(x')\ y'\ z'\ \mathbf{eqE_s}(v)\ \mathbf{eqE_s}(w)) \end{cases} \end{cases} \end{cases}$$

$$\lambda x \in \mathsf{nat}.$$
$$\mathbf{ind}\ u(x)\ \mathbf{of} \begin{cases} u(\mathbf{0}) \Rightarrow \lambda y \in \mathsf{nat}.\, \lambda z \in \mathsf{nat}. \\ \qquad \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqI_0} \\ \mathbf{s}(z') \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{0s}}(w) \end{cases} \\ \mathbf{s}(y') \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{0s}}(v) \end{cases} \\ u(\mathbf{s}(x')) \Rightarrow \lambda y \in \mathsf{nat}.\, \lambda z \in \mathsf{nat}. \\ \qquad \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{s0}}(v) \\ \mathbf{s}(y') \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \mathbf{eqE_{s0}}(w) \\ \mathbf{s}(z') \Rightarrow \lambda v{:}x =_{\mathsf{N}} y.\, \lambda w{:}y =_{\mathsf{N}} z.\, \\ \qquad \mathbf{eqI_s}(u(x')\ y'\ z'\ \mathbf{eqE_s}(v)\ \mathbf{eqE_s}(w)) \end{cases} \end{cases} \end{cases}$$

A correct definition of proof term *trans*:

$$\lambda x \in \mathsf{nat}.$$
$$\mathbf{ind}\ u(x)\ \mathbf{of} \begin{cases} u(\mathbf{0}) \Rightarrow \lambda y \in \mathsf{nat}.\, \lambda z \in \mathsf{nat}. \\ \qquad \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}\mathbf{0} =_{\mathsf{N}} \mathbf{0}.\, \lambda w{:}\mathbf{0} =_{\mathsf{N}} \mathbf{0}.\, \mathbf{eqI_0} \\ \mathbf{s}(z') \Rightarrow \lambda v{:}\mathbf{0} =_{\mathsf{N}} \mathbf{0}.\, \lambda w{:}\mathbf{0} =_{\mathsf{N}} \mathbf{s}(z').\, \mathbf{eqE_{0s}}(w) \end{cases} \\ \mathbf{s}(y') \Rightarrow \lambda v{:}\mathbf{0} =_{\mathsf{N}} \mathbf{s}(y').\, \lambda w{:}\mathbf{s}(y') =_{\mathsf{N}} z.\, \mathbf{eqE_{0s}}(v) \end{cases} \\ u(\mathbf{s}(x')) \Rightarrow \lambda y \in \mathsf{nat}.\, \lambda z \in \mathsf{nat}. \\ \qquad \mathbf{case}\ y\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}.\, \lambda w{:}\mathbf{0} =_{\mathsf{N}} z.\, \mathbf{eqE_{s0}}(v) \\ \mathbf{s}(y') \Rightarrow \mathbf{case}\ z\ \mathbf{of} \begin{cases} \mathbf{0} \Rightarrow \lambda v{:}\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y').\, \lambda w{:}\mathbf{s}(y') =_{\mathsf{N}} \mathbf{0}.\, \mathbf{eqE_{s0}}(w) \\ \mathbf{s}(z') \Rightarrow \lambda v{:}\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(y').\, \lambda w{:}\mathbf{s}(y') =_{\mathsf{N}} \mathbf{s}(z').\, \\ \qquad \mathbf{eqI_s}(u(x')\ y'\ z'\ \mathbf{eqE_s}(v)\ \mathbf{eqE_s}(w)) \end{cases} \end{cases} \end{cases}$$

1.9.5 $\forall x \in \mathsf{nat}.\neg(x =_{\mathsf{N}} \mathbf{0}) \supset \exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} x\ \mathit{true}$

A mathematical proof of $\forall x \in \mathsf{nat}.\neg(x =_{\mathsf{N}} \mathbf{0}) \supset \exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} x\ \mathit{true}$:

Proof. By case analysis of x.

Case $x = \mathbf{0}$. We need to show $\neg(\mathbf{0} =_{\mathsf{N}} \mathbf{0}) \supset \exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} \mathbf{0}$:
$\neg(\mathbf{0} =_{\mathsf{N}} \mathbf{0})\ \mathit{true}$ assumption

$\exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} \mathbf{0}\ \mathit{true}$ from $\neg(\mathbf{0} =_{\mathsf{N}} \mathbf{0})\ \mathit{true}$ and $\dfrac{}{\mathbf{0} =_{\mathsf{N}} \mathbf{0}\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{I}_0$

Case $x = \mathbf{s}(x')$. We need to show $\neg(\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}) \supset \exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} \mathbf{s}(x')$:

$\neg(\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0})\ \mathit{true}$ assumption (which is not used)

$x' =_{\mathsf{N}} x'\ \mathit{true}$ from the proof of $\forall z \in \mathsf{nat}.z =_{\mathsf{N}} z\ \mathit{true}$ and $x' \in \mathsf{nat}$

$\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(x')\ \mathit{true}$ from $\dfrac{x' =_{\mathsf{N}} x'\ \mathit{true}}{\mathbf{s}(x') =_{\mathsf{N}} \mathbf{s}(x')\ \mathit{true}}\ =_{\mathsf{N}}\mathsf{I}_{\mathsf{s}}$

$\exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} \mathbf{s}(x')\ \mathit{true}$ from $\dfrac{x' \in \mathsf{nat} \quad \mathbf{s}(x') =_{\mathsf{N}} x\ \mathit{true}}{\exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} x\ \mathit{true}}\ \exists\mathsf{I}$

□

The specification for a proof term pred of type $\forall x \in \mathsf{nat}.\neg(x =_{\mathsf{N}} \mathbf{0}) \supset \exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} x\ \mathit{true}$:

$$\begin{array}{llll} \mathit{pred} & \mathbf{0} & v : \neg(\mathbf{0} =_{\mathsf{N}} \mathbf{0}) & = \quad \mathsf{abort}_{\exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} \mathbf{0}}\ (v\ \mathsf{natI}_0) \\ \mathit{pred} & \mathbf{s}(x') & v : \neg(\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}) & = \quad \langle x', \mathbf{eqI_s}(\mathit{eqNat}\ x')\rangle \end{array}$$

A wrong definition of pred:

$$\mathit{pred} \ = \ \lambda x \in \mathsf{nat}.\ \mathbf{case}\ x\ \mathbf{of}\ \left\{ \begin{array}{l} \mathbf{0} \Rightarrow \lambda v\!:\!\neg(x =_{\mathsf{N}} \mathbf{0}).\ \mathsf{abort}_{\exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} x}\ (v\ \mathsf{natI}_0) \\ \mathbf{s}(x') \Rightarrow \lambda v\!:\!\neg(x =_{\mathsf{N}} \mathbf{0}).\ \langle x', \mathbf{eqI_s}(\mathit{eqNat}\ x')\rangle \end{array} \right.$$

A corrected definition of pred:

$$\mathit{pred} \ = \ \lambda x \in \mathsf{nat}.\ \mathbf{case}\ x\ \mathbf{of}\ \left\{ \begin{array}{l} \mathbf{0} \Rightarrow \lambda v\!:\!\neg(\mathbf{0} =_{\mathsf{N}} \mathbf{0}).\ \mathsf{abort}_{\exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} \mathbf{0}}\ (v\ \mathsf{natI}_0) \\ \mathbf{s}(x') \Rightarrow \lambda v\!:\!\neg(\mathbf{s}(x') =_{\mathsf{N}} \mathbf{0}).\ \langle x', \mathbf{eqI_s}(\mathit{eqNat}\ x')\rangle \end{array} \right.$$