Term Formation.

$$\frac{}{\mathbf{0} \in \mathbf{nat}}\,\mathbf{nat}I_0 \qquad \frac{n \in \mathbf{nat}}{\mathbf{s}(n) \in \mathbf{nat}}\,\mathbf{nat}I_s$$

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

$$\frac{}{\Gamma \vdash \mathbf{true} \in \mathbf{bool}}\,\mathbf{bool}I_1 \qquad \frac{}{\Gamma \vdash \mathbf{false} \in \mathbf{bool}}\,\mathbf{bool}I_0$$

$$\frac{\Gamma \vdash t \in \mathbf{bool} \qquad \Gamma \vdash s_1 \in \tau \qquad \Gamma \vdash s_0 \in \tau}{\Gamma \vdash \mathbf{if}\ t\ \mathbf{then}\ s_1\ \mathbf{else}\ s_0 \in \tau}\,\mathbf{bool}E$$

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

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

Reductions.

$$\begin{array}{rcl}
(\mathbf{rec}\ \mathbf{0}\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s) & \Longrightarrow & t_0 \\
(\mathbf{rec}\ \mathbf{s}(n)\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s) & \Longrightarrow & \\
\multicolumn{3}{r}{[(\mathbf{rec}\ n\ \mathbf{of}\ f(\mathbf{0}) \Rightarrow t_0 \mid f(\mathbf{s}(x)) \Rightarrow t_s)/f(x)]\,[n/x]\,t_s} \\
\mathbf{if}\,\mathbf{true}\ \mathbf{then}\ s_1\ \mathbf{else}\ s_0 & \Longrightarrow & s_1 \\
\mathbf{if}\,\mathbf{false}\ \mathbf{then}\ s_1\ \mathbf{else}\ s_0 & \Longrightarrow & s_0 \\
(\mathbf{rec}\,\mathbf{nil}\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c) & \Longrightarrow & s_n \\
(\mathbf{rec}\ (h :: t)\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c) & \Longrightarrow & \\
\multicolumn{3}{r}{[(\mathbf{rec}\,t\ \mathbf{of}\ f(\mathbf{nil}) \Rightarrow s_n \mid f(x :: l) \Rightarrow s_c)/f(l)]\,[h/x]\,[t/l]\,s_c}
\end{array}$$

1.9 Predicates on Data Types

In the preceding sections we have introduced the concept of a type which is determined by its elements. Examples were natural numbers, Booleans, and lists. In the next chapter we will explicitly quantify over elements of types. For example, we may assert that every natural number is either even or odd. Or we may claim that any two numbers possess a greatest common divisor. In order to formulate such statements we need some basic propositions concerned with data types. In this section we will define such predicates, following our usual methodology of using introduction and elimination rules to define the meaning of propositions.

We begin with $n < m$, the less-than relation between natural numbers. We have the following formation rule:

$$\frac{\Gamma \vdash m \in \mathbf{nat} \quad \Gamma \vdash n \in \mathbf{nat}}{\Gamma \vdash m < n \; \mathit{prop}} {<}F$$

Note that this formation rule for propositions relies on the judgment $t \in \tau$. Consequently, we have to permit a hypothetical judgment, in case n or m mention variables declared with their type, such as $x \in \mathbf{nat}$. Thus, in general, the question whether A *prop* may now depend on assumptions of the form $x \in \tau$.

This has a consequence for the judgment A *true*. As before, we now must allow assumptions of the form B *true*, but in addition we must permit assumptions of the form $x \in \tau$. We still call the collection of such assumptions a *context* and continue to denote it with Γ.

$$\frac{}{\Gamma \vdash \mathbf{0} < \mathbf{s}(n) \; \mathit{true}} {<}I_0 \qquad \frac{\Gamma \vdash m < n \; \mathit{true}}{\Gamma \vdash \mathbf{s}(m) < \mathbf{s}(n) \; \mathit{true}} {<}I_s$$

The second rule exhibits a new phenomenon: the relation '$<$' whose meaning we are trying to define appears in the premise as well as in the conclusion. In effect, we have not really introduced '$<$', since it already occurs. However, such a definition is still justified, since the conclusion defines the meaning of $\mathbf{s}(m) < \cdot$ in terms of $m < \cdot$. We refer to this relation as *inductively defined*. Actually we have already seen a similar phenomenon in the second "introduction" rule for **nat**:

$$\frac{\Gamma \vdash n \in \mathbf{nat}}{\Gamma \vdash \mathbf{s}(n) \in \mathbf{nat}} \mathbf{nat}I_s$$

The type **nat** we are trying to define already occurs in the premise! So it may be better to think of this rule as a formation rule for the successor operation on natural numbers, rather than an introduction rule for natural numbers.

Returning to the less-than relation, we have to derive the elimination rules. What can we conclude from $\Gamma \vdash m < n$ *true*? Since there are two introduction rules, we could try our previous approach and distinguish cases for the proof of that judgment. This, however, is somewhat awkward in this case—we postpone discussion of this option until later. Instead of distinguishing cases for the proof of the judgment, we distinguish cases for m and n. In each case, we analyse how the resulting judgment could be proven and write out the corresponding elimination rule. First, if n is zero, then the judgment can never have a normal proof, since no introduction rule applies. Therefore we are justified in concluding anything, as in the elimination rule for falsehood.

$$\frac{\Gamma \vdash m < \mathbf{0} \; \mathit{true}}{\Gamma \vdash C \; \mathit{true}} {<}E_0$$

If the $m = \mathbf{0}$ and $n = \mathbf{s}(n')$, then it could be inferred only by the first introduction rule ${<}I_0$. This yields no information, since there are no premises to this rule. This is just as in the case of the true proposition $\top$.

The last remaining possibility is that both $m = \mathbf{s}(m')$ and $n = \mathbf{s}(n')$. In that case we now that $m' < n'$, because ${<}I_s$ is the only rule that could have been applied.

$$\frac{\Gamma \vdash \mathbf{s}(m') < \mathbf{s}(n') \; \mathit{true}}{\Gamma \vdash m' < n' \; \mathit{true}}\, {<}E_s$$

We summarize the formation, introduction, and elimination rules.

$$\frac{\Gamma \vdash n \in \mathbf{nat} \quad \Gamma \vdash m \in \mathbf{nat}}{\Gamma \vdash n < m \; \mathit{prop}}\, {<}F$$

$$\frac{}{\Gamma \vdash \mathbf{0} < \mathbf{s}(n) \; \mathit{true}}\, {<}I_0 \qquad\qquad \frac{\Gamma \vdash m < n \; \mathit{true}}{\Gamma \vdash \mathbf{s}(m) < \mathbf{s}(n) \; \mathit{true}}\, {<}I_s$$

$$\frac{\Gamma \vdash m < \mathbf{0} \; \mathit{true}}{\Gamma \vdash C \; \mathit{true}}\, {<}E_0$$

$$\textit{no rule for } \mathbf{0} < \mathbf{s}(n') \qquad\qquad \frac{\Gamma \vdash \mathbf{s}(m') < \mathbf{s}(n') \; \mathit{true}}{\Gamma \vdash m' < n' \; \mathit{true}}\, {<}E_s$$

Now we can prove some simple relations between natural numbers. For example:

$$\frac{\dfrac{}{\cdot \vdash \mathbf{0} < \mathbf{s}(\mathbf{0}) \; \mathit{true}}\, {<}I_0}{\cdot \vdash \mathbf{0} < \mathbf{s}(\mathbf{s}(\mathbf{0})) \; \mathit{true}}\, {<}I_s$$

We can also establish some simple parametric properties of natural numbers.

$$\frac{\dfrac{\dfrac{}{m \in \mathbf{nat}, m < 0 \; \mathit{true} \vdash m < 0 \; \mathit{true}}\, u}{m \in \mathbf{nat}, m < 0 \; \mathit{true} \vdash \bot \; \mathit{true}}\, {<}E_0}{m \in \mathbf{nat} \vdash \neg(m < 0) \; \mathit{true}}\, {\supset}I^u$$

In the application of the ${<}E_0$ rule, we chose $C = \bot$ in order to complete the proof of $\neg(m < \mathbf{0})$. Even slightly more complicated properties, such as $m < \mathbf{s}(m)$ require a proof by induction and are therefore postponed until Section 1.10.

We introduce one further relation between natural numbers, namely equality.

We write $m =_N n$. Otherwise we follow the blueprint of the less-than relation.

$$\frac{\Gamma \vdash m \in \mathbf{nat} \qquad \Gamma \vdash n \in \mathbf{nat}}{\Gamma \vdash m =_N n \; \mathit{prop}} =_N F$$

$$\frac{}{\Gamma \vdash \mathbf{0} =_N \mathbf{0} \; \mathit{true}} =_N I_0 \qquad \frac{\Gamma \vdash m =_N n \; \mathit{true}}{\Gamma \vdash \mathbf{s}(m) =_N \mathbf{s}(n) \; \mathit{true}} =_N I_s$$

$$\mathit{no} =_N E_{00} \; \mathit{elimination\ rule} \qquad \frac{\Gamma \vdash \mathbf{0} =_N \mathbf{s}(n) \; \mathit{true}}{\Gamma \vdash C \; \mathit{true}} =_N E_{0s}$$

$$\frac{\Gamma \vdash \mathbf{s}(m) =_N \mathbf{0} \; \mathit{true}}{\Gamma \vdash C \; \mathit{true}} =_N E_{s0} \qquad \frac{\Gamma \vdash \mathbf{s}(m) =_N \mathbf{s}(n) \; \mathit{true}}{\Gamma \vdash m =_N n \; \mathit{true}} =_N E_{ss}$$

Note the difference between the *function*

$$\mathit{eq} \in \mathbf{nat} \rightarrow \mathbf{nat} \rightarrow \mathbf{bool}$$

and the *proposition*

$$m =_N n$$

The equality function provides a computation on natural numbers, always returning **true** or **false**. The proposition $m =_N n$ requires *proof.* Using induction, we can later verify a relationship between these two notions, namely that $\mathit{eq}\, n\, m$ reduces to **true** if $m =_N n$ is true, and $\mathit{eq}\, n\, m$ reduces to **false** if $\neg(m =_N n)$.

1.10 Induction

Now that we have introduced the basic propositions regarding order and equality, we can consider induction as a reasoning principle. So far, we have considered the following elimination rule for natural numbers:

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

This rule can be applied if we can derive $t \in \mathbf{nat}$ from our assumptions and we are trying to construct a term $s \in \tau$. But how do we use a variable or term $t \in \mathbf{nat}$ if the judgment we are trying to prove has the form $M : A$, that is, if we are trying the prove the truth of a proposition? The answer is induction. This is actually very similar to primitive recursion. The only complication is that the proposition A we are trying to prove may depend on t. We indicate this by writing $A(x)$ to mean the proposition A with one or more occurrences of a variable x. $A(t)$ is our notation for the result of substituting t for x in A. We could also write $[t/x]A$, but this is more difficult to read. Informally, induction says that in order to prove $A(t)$ *true* for arbitrary t we have to prove $A(\mathbf{0})$ *true* (the base case), and that for every $x \in \mathbf{nat}$, if $A(x)$ *true* then $A(\mathbf{s}(x))$ *true.*

Formally this becomes:

$$\frac{\Gamma \vdash t \in \mathbf{nat} \qquad \Gamma \vdash A(\mathbf{0})\ \mathit{true} \qquad \Gamma, x \in \mathbf{nat}, A(x)\ \mathit{true} \vdash A(\mathbf{s}(x))\ \mathit{true}}{\Gamma \vdash A(t)\ \mathit{true}}\ \mathbf{nat}E'$$

Here, $A(x)$ is called the *induction predicate.* If t is a variable (which is frequently the case) it is called the *induction variable.* With this rule, we can now prove some more interesting properties. As a simple example we show that $m < \mathbf{s}(m)$ *true* for any natural number m. Here we use $\mathcal{D}$ to stand for the derivation of the third premise in order to overcome the typesetting difficulties.

$$\mathcal{D} = \frac{\dfrac{}{m \in \mathbf{nat}, x \in \mathbf{nat}, x < \mathbf{s}(x)\ \mathit{true} \vdash x < \mathbf{s}(x)\ \mathit{true}}}{m \in \mathbf{nat}, x \in \mathbf{nat}, x < \mathbf{s}(x)\ \mathit{true} \vdash \mathbf{s}(x) < \mathbf{s}(\mathbf{s}(x))}\ {<}I_s$$

$$\frac{\dfrac{}{m \in \mathbf{nat} \vdash m \in \mathbf{nat}} \qquad \dfrac{}{m \in \mathbf{nat} \vdash \mathbf{0} < \mathbf{s}(\mathbf{0})}\ {<}I_0 \qquad \mathcal{D}}{m \in \mathbf{nat} \vdash m < \mathbf{s}(m)}\ \mathbf{nat}E'$$

The property $A(x)$ appearing in the induction principle is $A(x) = x < \mathbf{s}(x)$. So the final conclusion is $A(m) = m < \mathbf{s}(m)$. In the second premise we have to prove $A(\mathbf{0}) = \mathbf{0} < \mathbf{s}(\mathbf{0})$ which follows directly by an introduction rule.

Despite the presence of the induction rule, there are other properties we cannot yet prove easily since the logic does not have quantifiers. An example is the decidability of equality: For any natural numbers m and n, either $m =_N n$ or $\neg(m =_N n)$. This is an example of the practical limitations of *quantifier-free induction*, that is, induction where the induction predicate does not contain any quantifiers.

The topic of this chapter is the interpretation of constructive proofs as programs. So what is the computational meaning of induction? It actually corresponds very closely to primitive recursion.

$$\frac{\Gamma \vdash t \in \mathbf{nat} \qquad \Gamma \vdash M : A(\mathbf{0}) \qquad \Gamma, x \in \mathbf{nat}, u(x){:}A(x) \vdash N : A(\mathbf{s}(x))}{\Gamma \vdash \mathbf{ind}\ t\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N : A(t)}\ \mathbf{nat}E'$$

Here, $u(x)$ is just the notation for a variable which may occur in N. Note that u cannot occur in M or in N in any other form. The reduction rules are precisely the same as for primitive recursion.

$$\begin{array}{l}
(\mathbf{ind}\ \mathbf{0}\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N) \quad \Longrightarrow \quad M \\
(\mathbf{ind}\ \mathbf{s}(n)\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N) \quad \Longrightarrow \\
{[(\mathbf{ind}\ n\ \mathbf{of}\ u(\mathbf{0}) \Rightarrow M \mid u(\mathbf{s}(x)) \Rightarrow N)/u(n)]}\ [n/x]N
\end{array}$$

We see that primitive recursion and induction are almost identical. The only difference is that primitive recursion returns an element of a type, while induction generates a proof of a proposition. Thus one could say that they are related by an extension of the Curry-Howard correspondence. However, not every type τ can be naturally interpreted as a proposition (which proposition, for example, is expressed by **nat**?), so we no longer speak of an isomorphism.

We close this section by the version of the rules for the basic relations between natural numbers that carry proof terms. This annotation of the rules is straightforward.

$$\frac{\Gamma \vdash n \in \mathbf{nat} \quad \Gamma \vdash m \in \mathbf{nat}}{\Gamma \vdash n < m \ \mathit{prop}} {<}F$$

$$\frac{}{\Gamma \vdash \mathbf{lt}_0 : \mathbf{0} < \mathbf{s}(n)} {<}I_0 \qquad \frac{\Gamma \vdash M : m < n}{\Gamma \vdash \mathbf{lt}_s(M) : \mathbf{s}(m) < \mathbf{s}(n)} {<}I_s$$

$$\frac{\Gamma \vdash M : m < \mathbf{0}}{\Gamma \vdash \mathbf{ltE}_0(M) : C} {<}E_0$$

$$\textit{no rule for } \mathbf{0} < \mathbf{s}(n') \qquad \frac{\Gamma \vdash M : \mathbf{s}(m') < \mathbf{s}(n')}{\Gamma \vdash \mathbf{ltE}_s(M) : m' < n'} {<}E_s$$

$$\frac{\Gamma \vdash m \in \mathbf{nat} \qquad \Gamma \vdash n \in \mathbf{nat}}{\Gamma \vdash m =_N n \ \mathit{prop}} =_N F$$

$$\frac{}{\Gamma \vdash \mathbf{eq}_0 : \mathbf{0} =_N \mathbf{0}} =_N I_0 \qquad \frac{\Gamma \vdash M : m =_N n}{\Gamma \vdash \mathbf{eq}_s(M) : \mathbf{s}(m) =_N \mathbf{s}(n)} =_N I_s$$

$$\textit{no } =_N E_{00} \textit{ elimination rule} \qquad \frac{\Gamma \vdash M : \mathbf{0} =_N \mathbf{s}(n)}{\Gamma \vdash \mathbf{eqE}_{0s}(M) : C} =_N E_{0s}$$

$$\frac{\Gamma \vdash M : \mathbf{s}(m) =_N \mathbf{0}}{\Gamma \vdash \mathbf{eqE}_{s0}(M) : C} =_N E_{s0} \qquad \frac{\Gamma \vdash M : \mathbf{s}(m) =_N \mathbf{s}(n)}{\Gamma \vdash \mathbf{eqE}_{ss}(M) : m =_N n} =_N E_{ss}$$