Chapter 4

First-Order Logic and Type Theory

In the first chapter we developed the logic of pure propositions without reference to data types such as natural numbers. In the second chapter we explained the computational interpretation of proofs. Later in this chapter we will introduce data types and ways to compute with them using primitive recursion. Together, these will allows us to reason about data and programs manipulating data. In other words, we will be able to prove our programs correct with respect to their expected behavior on data. The principal means for will be induction, introduced towards the end of this chapter. There are several ways to employ the machinery we will develop. For example, we can execute proofs directly, using their interpretation as programs. Or we can *extract* functions, ignoring some proof objects that have are irrelevant with respect to the data our programs return. That is, we can contract proofs to programs. Or we can simply write our programs and use the logical machinery we have developed to prove them correct.

In practice, there are situations in which each of them is appropriate. However, we note that in practice we rarely formally prove our programs to be correct. This is because there is no mechanical procedure to establish if a given programs satisfies its specification. Moreover, we often have to deal with input or output, with mutable state or concurrency, or with complex systems where the specification itself could be as difficult to develop as the implementation. Instead, we typically convince ourselves that central parts of our program and the critical algorithms are correct. Even if proofs are never formalized, this chapter will help you in reasoning about programs and their correctness.

There is another way in which the material of this chapter is directly relevant to computing practice. In the absence of practical methods for verifying full correctness, we can be less ambitious by limiting ourselves to program properties that can indeed be mechanically verified. The most pervasive application of this idea in programming is the idea of *type systems*. By checking the type

correctness of a program we fall far short of verifying it, but we establish a kind of consistency statement. Since languages satisfy (or are supposed to satisfy) type preservation, we know that, if a result is returned, it is a value of the right type. Moreover, during the execution of a program (modeled here by reduction), all intermediate states are well-typed which prevents certain absurd situations, such as adding a natural number to a function. This is often summarized in the slogan that "*well-typed programs cannot go wrong*". Well-typed programs are *safe* in this respect. In terms of machine language, assuming a correct compiler, this guards against irrecoverable faults such as jumping to an address that does not contain valid code, or attempting to write to inaccessible memory location.

There is some room for exploring the continuum between types, as present in current programming languages, and full specifications, the domain of *type theory*. By presenting these elements in a unified framework, we have the basis for such an exploration.

We begin this chapter with a discussion of the universal and existential quantifiers, followed by a number of examples of inductive reasoning with data types.

4.1 Quantification

In this section, we introduce universal and existential quantification. As usual, we follow the method of using introduction and elimination rules to explain the meaning of the connectives. An important aspect of the treatment of quantifiers is that it should be completely independent of the domain of quantification. We want to capture what is true of all quantifiers, rather than those applying to natural numbers or integers or rationals or lists or other type of data. We will therefore quantify over objects of an unspecified (arbitrary) type τ. Whatever we derive, will of course also hold for specific domain (for example, $\tau = \mathsf{nat}$). The basic judgment connecting objects t to types τ is $t : \tau$. We also have a judgment that τ is a valid type, written as $\tau\ \mathit{type}$. We will refer to these judgments here, but not define any specific instances until later in the chapter when discussing data types.

First, universal quantification, written as $\forall x{:}\tau.\ A(x)$. Here x is a bound variable and can therefore be renamed as discussed in the preceding chapter. When we write $A(x)$ we mean an arbitrary proposition which may depend on x. We will als say that A is *predicate* on elements of type τ.

For the quantification to be well-formed, the body must be well-formed under the assumption that a is an object of type τ, and τ must be a valid type, two new judgments we consider in this chapter.

$$\frac{\tau\ \mathit{type} \quad \begin{array}{c}\overline{a : \tau}\\ \vdots \\ A(a)\ \mathit{prop}\end{array}}{\forall x{:}\tau.\ A(x)\ \mathit{prop}}\ \forall F^a$$

For the introduction rule we require that $A(a)$ be true for arbitrary a. In other words, the premise contains a *parametric judgment*, explained in more detail below.

$$\dfrac{\begin{array}{c}\overline{a:\tau}\\ \vdots \\ A(a)\ \mathit{true}\end{array}}{\forall x{:}\tau.\ A(x)\ \mathit{true}}\ \forall I^a$$

It is important that a be a new parameter, not used outside of its scope, which is the derivation between the new hypothesis $a : \tau$ and the conclusion $A(a)$ *true*. In particular, it may not occur in $\forall x{:}\tau.\ A(x)$.

If we think of this as the defining property of universal quantification, then a verification of $\forall x{:}\tau.\ A(x)$ describes a construction by which an arbitrary $t : \tau$ can be transformed into a proof of $A(t)$ *true*.

$$\dfrac{\forall x{:}\tau.\ A(x)\ \mathit{true} \quad t:\tau}{A(t)\ \mathit{true}}\ \forall E$$

We must verify that $t : \tau$ so that $A(t)$ is a well-formed proposition.

The local reduction uses the following *substitution principle for parametric judgments*:

$$\text{If}\quad \begin{array}{c}\overline{a:\tau}\\ \mathcal{D}\\ J(a)\end{array}\quad\text{and}\quad \begin{array}{c}\mathcal{E}\\ t:\tau\end{array}\quad\text{then}\quad \begin{array}{c}\dfrac{\mathcal{E}}{t:\tau}\\ [t/a]\mathcal{D}\\ J(t)\end{array}$$

The right hand side is constructed by systematically substituting t for a in $\mathcal{D}$ and the judgments occurring in it. As usual, this substitution must be *capture avoiding* to be meaningful. It is the substitution into the judgments themselves which distinguishes substitution for parameters from substitution for hypotheses.

The local reduction for universal quantification then exploits this substitution principle.

$$\dfrac{\dfrac{\begin{array}{c}\overline{a:\tau}\\ \mathcal{D}\\ A(a)\ \mathit{true}\end{array}}{\forall x{:}\tau.\ A(x)\ \mathit{true}}\ \forall I^a \qquad \begin{array}{c}\mathcal{E}\\ t:\tau\end{array}}{A(t)\ \mathit{true}}\ \forall E \qquad \Longrightarrow_R \qquad \begin{array}{c}\dfrac{\mathcal{E}}{t:\tau}\\ [t/a]\mathcal{D}\\ A(t)\ \mathit{true}\end{array}$$

The local expansion introduces a parameter which we can use the eliminate

the universal quantifier.

$$\begin{array}{c} \mathcal{D} \\ \forall x{:}\tau.\ A(x)\ true \end{array} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \forall x{:}\tau.\ A(x)\ true\end{array} \quad \overline{a:\tau}}{A(a)\ true}\ \forall E}{\forall x{:}\tau.\ A(x)\ true}\ \forall I^a$$

As a simple example, consider the proof that universal quantifiers distribute over conjunction.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\forall x{:}\tau.\ A(x)) \wedge (\forall x{:}\tau.\ B(x))\ true}\ u}{\forall x{:}\tau.\ A(x)\ true}\ \wedge E_L \quad \overline{a:\tau}}{A(a)\ true}\ \forall E}{\forall x{:}\tau.\ A(x)\ true}\ \forall I^a \quad \dfrac{\dfrac{\dfrac{\dfrac{}{(\forall x{:}\tau.\ A(x)) \wedge (\forall x{:}\tau.\ B(x))\ true}\ u}{\forall x{:}\tau.\ B(x)\ true}\ \wedge E_R \quad \overline{b:\tau}}{B(b)\ true}\ \forall E}{\forall x{:}\tau.\ B(x)\ true}\ \forall I^b}{(\forall x{:}\tau.\ A(x)) \wedge (\forall x{:}\tau.\ B(x))\ true}\ \wedge I}{(\forall x{:}\tau.\ A(x) \wedge B(x)) \supset (\forall x{:}\tau.\ A(x)) \wedge (\forall x{:}\tau.\ B(x))\ true}\ \supset I^u$$

The existential quantifier is more difficult to specify, although the introduction rule seems innocuous enough.

$$\dfrac{t:\tau \quad A(t)\ true}{\exists x{:}\tau.\ A(x)\ true}\ \exists I$$

The elimination rules creates some difficulties. We cannot write

$$\dfrac{\exists x{:}\tau.\ A(x)\ true}{A(t)\ true}\ \exists E?$$

because we do not know for which t is is the case that $A(t)$ holds. It is easy to see that local soundness would fail with this rule, because we would prove $\exists x{:}\tau.\ A(x)$ with one witness t and then eliminate the quantifier using another object t'.

The best we can do is to assume that $A(a)$ is true for some new parameter a. The scope of this assumption is limited to the proof of some conclusion $C\ true$ which does not mention a (which must be new).

$$\dfrac{\exists x{:}\tau.\ A(x)\ true \qquad \begin{array}{c}\overline{a:\tau} \quad \overline{A(a)\ true}\ u\\ \vdots \\ C\ true\end{array}}{C\ true}\ \exists E^{a,u}$$

Here, the scope of the hypotheses a and u is the deduction on the right, indicated by the vertical dots. In particular, C may not depend on a. We use this crucially in the local reduction.

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ t:\tau\end{array}\quad\begin{array}{c}\mathcal{E}\\ A(t)\ true\end{array}}{\exists x{:}\tau.\ A(x)\ true}\exists I \qquad \begin{array}{c}\overline{a:\tau}\\ \vdots\\ C\ true\end{array}\quad\begin{array}{c}\overline{A(a)\ true}\ u\\ \mathcal{F}\end{array}}{C\ true}\exists E^{a,u} \qquad \Longrightarrow_R \qquad \dfrac{\begin{array}{c}\mathcal{D}\\ t:\tau\end{array}\quad\dfrac{\begin{array}{c}\mathcal{E}\\ A(t)\ true\end{array}}{\begin{array}{c}[t/a]\mathcal{F}\\ C\ true\end{array}}\ u}{}$$

The reduction requires two substitutions, one for a parameter a and one for a hypothesis u.

The local expansion is patterned after the disjunction.

$$\begin{array}{c}\mathcal{D}\\ \exists x{:}\tau.\ A(x)\ true\end{array} \quad \Longrightarrow_E \quad \dfrac{\begin{array}{c}\mathcal{D}\\ \exists x{:}\tau.\ A(x)\ true\end{array}\quad\dfrac{\overline{a:\tau}\quad\overline{A(a)\ true}\ u}{\exists x{:}\tau.\ A(x)\ true}\exists I}{\exists x{:}\tau.\ A(x)\ true}\exists E^{a,u}$$

As an example of quantifiers we show the equivalence of $\forall x{:}\tau.\ A(x) \supset C$ and $(\exists x{:}\tau.\ A(x)) \supset C$, where C does not depend on x. Generally, in our propositions, any possibly dependence on a bound variable is indicated by writing a general *predicate* $A(x_1, \ldots, x_n)$. We do not make explicit when such propositions are well-formed, although appropriate rules for explicit A could be given.

When looking at a proof, the static representation on the page is an inadequate image for the dynamics of proof construction. As we did earlier, we give two examples where we show the various stages of proof construction.

$$\begin{array}{c}\vdots\\ ((\exists x{:}\tau.\ A(x)) \supset C) \supset \forall x{:}\tau.\ (A(x) \supset C)\ true\end{array}$$

The first three steps can be taken without hesitation, because we can always apply implication and universal introduction from the bottom up without possibly missing a proof.

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\overline{(\exists x{:}\tau.\ A(x)) \supset C\ true}\ u\\ \vdots\\ C\ true\end{array}}{A(a) \supset C\ true}\supset I^w}{\forall x{:}\tau.\ A(x) \supset C\ true}\forall I^a \qquad \begin{array}{c}\overline{a:\tau}\\ \overline{A(a)\ true}\ w\end{array}}{((\exists x{:}\tau.\ A(x)) \supset C) \supset \forall x{:}\tau.\ (A(x) \supset C)\ true}\supset I^u$$

At this point the conclusion is atomic, so we must apply an elimination to an assumption if we follow the strategy of *introductions bottom-up* and *eliminations top-down*. The only possibility is implication elimination, since $a : \tau$ and

$A(a)$ *true* are atomic. This gives us a new subgoal.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a:\tau} \quad \dfrac{}{A(a)\ true}\,w}{\vdots}}{\dfrac{}{(\exists x{:}\tau.\ A(x)) \supset C\ true}\,u \qquad \exists x{:}\tau.\ A(x)}\supset E}{\dfrac{C\ true}{A(a)\supset C\ true}\supset I^w}}{\dfrac{}{\forall x{:}\tau.\ A(x)\supset C\ true}\forall I^a}}{((\exists x{:}\tau.\ A(x))\supset C)\supset \forall x{:}\tau.\ (A(x)\supset C)\ true}\supset I^u$$

At this point it is easy to see how to complete the proof with an existential introduction.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\exists x{:}\tau.\ A(x))\supset C\ true}\,u \quad \dfrac{\dfrac{}{a:\tau} \quad \dfrac{}{A(a)\ true}\,w}{\exists x{:}\tau.\ A(x)}\exists I}{C\ true}\supset E}{A(a)\supset C\ true}\supset I^w}{\forall x{:}\tau.\ A(x)\supset C\ true}\forall I^a}{((\exists x{:}\tau.\ A(x))\supset C)\supset \forall x{:}\tau.\ (A(x)\supset C)\ true}\supset I^u$$

We now consider the reverse implication.

$$\dfrac{}{(\forall x{:}\tau.\ (A(x)\supset C))\supset((\exists x{:}\tau.\ A(x))\supset C)\ true}$$

From the initial goal, we can blindly carry out two implication introductions, bottom-up, which yields the following situation.

$$\dfrac{\dfrac{\dfrac{}{\exists x{:}\tau.\ A(x)\ true}\,w \quad \dfrac{}{\forall x{:}\tau.\ A(x)\supset C\ true}\,u}{\vdots}}{(\forall x{:}\tau.\ (A(x)\supset C))\supset((\exists x{:}\tau.\ A(x))\supset C)\ true}\supset I^u$$

No we have two choices: existential elimination applied to w or universal elimination applied to u. However, we have not introduced any terms, so only the

existential elimination can go forward.

$$\dfrac{\dfrac{\dfrac{}{\exists x{:}\tau.\ A(x)\ true}\,w \qquad \dfrac{\dfrac{\dfrac{}{\forall x{:}\tau.\ A(x)\supset C\ true}\,u \quad \dfrac{}{a:\tau}}{A(a)\supset C\ true}\,\forall E \quad \dfrac{}{A(a)\ true}\,v}{C\ true}\,\supset E \\ \vdots \\ C\ true}{\dfrac{C\ true}{(\exists x{:}\tau.\ A(x))\supset C\ true}\,\supset I^{w}}\,\exists E^{a,v}}{(\forall x{:}\tau.\ (A(x)\supset C))\supset((\exists x{:}\tau.\ A(x))\supset C)\ true}\,\supset I^{u}$$

At this point we need to apply another elimination rule to an assumption. We don't have much to work with, so we try universal elimination.

$$\dfrac{\dfrac{\dfrac{}{\exists x{:}\tau.\ A(x)\ true}\,w \qquad \dfrac{\dfrac{}{\forall x{:}\tau.\ A(x)\supset C\ true}\,u \quad \dfrac{}{a:\tau}}{A(a)\supset C\ true}\,\forall E \quad \dfrac{}{A(a)\ true}\,v \\ \vdots \\ C\ true}{\dfrac{C\ true}{(\exists x{:}\tau.\ A(x))\supset C\ true}\,\supset I^{w}}\,\exists E^{a,v}}{(\forall x{:}\tau.\ (A(x)\supset C))\supset((\exists x{:}\tau.\ A(x))\supset C)\ true}\,\supset I^{u}$$

Now we can fill the gap with an implication elimination.

$$\dfrac{\dfrac{\dfrac{}{\exists x{:}\tau.\ A(x)\ true}\,w \qquad \dfrac{\dfrac{\dfrac{}{\forall x{:}\tau.\ A(x)\supset C\ true}\,u \quad \dfrac{}{a:\tau}}{A(a)\supset C\ true}\,\forall E \quad \dfrac{}{A(a)\ true}\,v}{C\ true}\,\supset E}{\dfrac{C\ true}{(\exists x{:}\tau.\ A(x))\supset C\ true}\,\supset I^{w}}\,\exists E^{a,v}}{(\forall x{:}\tau.\ (A(x)\supset C))\supset((\exists x{:}\tau.\ A(x))\supset C)\ true}\,\supset I^{u}$$

In order to formalize the proof search strategy, we use the judgments A has a verification ($A\uparrow$) and A may be used ($A\downarrow$) as we did in the propositional case. Universal quantification is straightforward:

$$\dfrac{\dfrac{}{a:\tau}\\ \vdots \\ A(a)\uparrow}{\forall x{:}\tau.\ A(x)\uparrow}\,\forall I^{a} \qquad \dfrac{\forall x{:}\tau.\ A(x)\downarrow \quad t:\tau}{A(t)\downarrow}\,\forall E$$

We do not assign a direction to the judgment for typing objects, $t : \tau$.

Verifications for the existential elimination are patterned after the disjunction: we translate a usable $\exists x{:}\tau.\ A(x)$ into a usable $A(a)$ with a limited scope, both in the verification of some C.

$$\frac{t : \tau \quad A(t)\uparrow}{\exists x{:}\tau.\ A(x)\uparrow}\ \exists I \qquad \frac{\exists x{:}\tau.\ A(x)\downarrow \qquad \begin{array}{c}\overline{a:\tau} \quad \overline{A(a)\downarrow}^{\,u}\\ \vdots \\ C\uparrow\end{array}}{C\uparrow}\ \exists E^{a,u}$$

As before, the fact that every true proposition has a verification is a kind of global version of the local soundness and completeness properties. If we take this for granted (since we do not prove it until later), then we can use this to demonstrate that certain propositions are not true, parametrically.

For example, we show that $(\exists x{:}\tau.\ A(x)) \supset (\forall x{:}\tau.\ A(x))$ is not true in general. After the first two steps of constructing a verification, we arrive at

$$\frac{\dfrac{\dfrac{\overline{\exists x{:}\tau.\ A(x)\downarrow}^{\,u} \quad \overline{a:\tau}}{\vdots}}{A(a)\uparrow}}{\forall x{:}\tau.\ A(x)\uparrow}\ \forall I^a}{(\exists x{:}\tau.\ A(x)) \supset (\forall x{:}\tau.\ A(x))\uparrow}\ \supset I^u$$

At this point we can only apply existential elimination, which leads to

$$\frac{\dfrac{\dfrac{\overline{\exists x{:}\tau.\ A(x)\downarrow}^{\,u} \qquad \begin{array}{c}\overline{b:\tau} \quad \overline{A(b)\downarrow}^{\,v} \quad \overline{a:\tau}\\ \vdots \\ A(a)\uparrow\end{array}}{A(a)\uparrow}\ \exists E^{b,v}}{\forall x{:}\tau.\ A(x)\uparrow}\ \forall I^a}{(\exists x{:}\tau.\ A(x)) \supset (\forall x{:}\tau.\ A(x))\uparrow}\ \supset I^u$$

We cannot close the gap, because a and b are different parameters. We can only apply existential elimination to assumption u again. But this only creates $c : \tau$ and $A(c)\downarrow$ for some new c, so have made no progress. No matter how often we apply existential elimination, since the parameter introduced must be new, we can never prove $A(a)$.