Chapter 1

Proof Terms

This chapter presents an alternative formulation of propositional logic using the principle called the *Curry-Howard isomorphism* [**?**]. As a principle connecting logic and programming languages, it states that propositions in logic correspond to types in programming languages (*propositions-as-types* correspondence) and that proofs in logic correspond to programs in programming languages (*proofs-as-programs* correspondence). Thus, by applying the Curry-Howard isomorphism to a formulation of logic, we systematically derive a formulation of a corresponding programming language. In the case of propositional logic, we obtain a basic definition of the simply-typed λ-calculus.

1.1 Proof terms

The basic idea behind the Curry-Howard isomorphism is to represent a proof $\mathcal{D}$ of a truth judgment $A\ true$ as a *proof term* M of *type* A:

$$\overset{\mathcal{D}}{A\ true} \quad \Longleftrightarrow \quad M : A$$

That is, a *typing judgment* $M : A$ expresses that a proof term M of type A is a (concise) representation of a proof of $A\ true$. When $M : A$ holds, we say that proof term M typechecks with type A. Note that A can be interpreted both as a proposition and as a type, depending on the context in which it is used.

Under the correspondence between proofs and proof terms shown above, each inference rule for deducing truth judgments is translated to a corresponding *typing rule* for deducing typing judgments; by convention, a typing rule is given the same name as the inference rule from which it is derived:

$$\frac{\cdots}{A\ true}\ R \quad \Longleftrightarrow \quad \frac{\cdots}{M : A}\ R$$

Thus the typing rules for proof terms constitute another natural deduction system, in which an introduction rule assigns to a proof term a type involving a particular connective whereas an elimination rule uses such a proof term in its premise.

We may choose any syntax for proof terms as long as each proof term of type A provides all necessary information to extract a corresponding proof of $A\ true$. Below we design proof terms according to the syntax for the simply-typed λ-calculus so as to emphasize the close connection between logic and type theory. We use metavariables M, N, $\cdots$ for terms. Figure 1.1 shows all the typing rules for proof terms for propositional logic where the set of proof terms is inductively defined as follows:

$$\begin{array}{lrcl} \text{proof term} & M & ::= & (M, M) \mid \mathsf{fst}\ M \mid \mathsf{snd}\ M \mid \lambda x{:}A.\, M \mid M\ M \mid \\ & & & \mathsf{inl}_A\ M \mid \mathsf{inr}_A\ M \mid \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow M \mid \mathsf{inr}\ x \Rightarrow M \mid \langle\rangle \mid \mathsf{abort}_A\ M \end{array}$$

Conjunction

Consider an application of the rule $\wedge\mathsf{I}$ in which a proof $\mathcal{D}$ of $A \wedge B\ \mathit{true}$ is constructed from a proof $\mathcal{D}_A$ of $A\ \mathit{true}$ and a proof $\mathcal{D}_B$ of $B\ \mathit{true}$. If proof terms M and N represent $\mathcal{D}_A$ and $\mathcal{D}_B$, respectively, we use a *product term* (M, N) of type $A \wedge B$ to represent $\mathcal{D}$. Thus the rule $\wedge\mathsf{I}$ is translated to the following typing rule (of the same name):

$$\frac{A\ \mathit{true} \quad B\ \mathit{true}}{A \wedge B\ \mathit{true}}\ \wedge\mathsf{I} \qquad \Longleftrightarrow \qquad \frac{M : A \quad N : B}{(M, N) : A \wedge B}\ \wedge\mathsf{I}$$

We use *projection terms* $\mathsf{fst}\ M$ and $\mathsf{snd}\ M$ in translating the rule $\wedge\mathsf{E_L}$ and $\wedge\mathsf{E_R}$; fst and snd stand for 'first projection' and 'second projection,' respectively:

$$\frac{A \wedge B\ \mathit{true}}{A\ \mathit{true}}\ \wedge\mathsf{E_L} \qquad \Longleftrightarrow \qquad \frac{M : A \wedge B}{\mathsf{fst}\ M : A}\ \wedge\mathsf{E_L} \qquad\qquad \frac{A \wedge B\ \mathit{true}}{B\ \mathit{true}}\ \wedge\mathsf{E_R} \qquad \Longleftrightarrow \qquad \frac{M : A \wedge B}{\mathsf{snd}\ M : B}\ \wedge\mathsf{E_R}$$

Implication

Suppose that we wish to convert to a proof term a proof $\mathcal{D}$ of $A \supset B\ \mathit{true}$ that applies the rule $\supset\mathsf{I}$ to a hypothetical proof $\mathcal{E}$ of $B\ \mathit{true}$:

$$\mathcal{D}\left\{\begin{array}{l} \mathcal{E}\left\{\begin{array}{c} \overline{A\ \mathit{true}}^{\,x} \\ \vdots \\ B\ \mathit{true} \end{array}\right. \\ \hline A \supset B\ \mathit{true}\ \supset\mathsf{I}^x \end{array}\right.$$

In order to build a proof term M representing $\mathcal{E}$, we first need to assign a proof term to the hypothesis $\overline{A\ \mathit{true}}^{\,x}$. Since $A\ \mathit{true}$ is just a hypothesis without a concrete proof, its corresponding proof term is also unknown. Hence we represent $\overline{A\ \mathit{true}}^{\,x}$ as a *variable* x, for which we can later substitute another proof term (like we substitute a concrete proof of $A\ \mathit{true}$ for the hypothesis $\overline{A\ \mathit{true}}^{\,x}$):

$$\overline{A\ \mathit{true}}^{\,x} \qquad \Longleftrightarrow \qquad \overline{x : A}$$

If M represents $\mathcal{E}$, we use a *λ-abstraction* $\lambda x\!:\!A.\, M$ to represent $\mathcal{D}$:

$$\begin{array}{c} \overline{A\ \mathit{true}}^{\,x} \\ \vdots \\ B\ \mathit{true} \\ \hline A \supset B\ \mathit{true}\ \supset\mathsf{I}^x \end{array} \qquad \Longleftrightarrow \qquad \begin{array}{c} \overline{x : A} \\ \vdots \\ M : B \\ \hline \lambda x\!:\!A.\, M : A \supset B\ \supset\mathsf{I} \end{array}$$

We say that variable x is bound in the λ-abstraction $\lambda x\!:\!A.\, M$. Note that we may rename x to another variable without changing the meaning of $\lambda x\!:\!A.\, M$. For example, both $\lambda x\!:\!A.\, (x, x)$ and $\lambda y\!:\!A.\, (y, y)$ represent the same proof, since using a different label for the same hypothesis does not alter the structure of the proof. (Renaming a bound variable in a λ-abstraction is commonly called *α-conversion*.)

Similarly to the rule $\supset\mathsf{I}$ in propositional logic, the typing rule $\supset\mathsf{I}$ restricts the scope of the hypothesis $\overline{x : A}$ to its premise. As a result, the hypothesis $\overline{x : A}$ is discharged when the rule $\supset\mathsf{I}$ is applied, and variable x in $\lambda x\!:\!A.\, M$ can be assigned type A only if it appears within M. For example, $\lambda x\!:\!A.\, x$ has type $A \supset A$, but $(\lambda x\!:\!A.\, x, x)$ cannot be assigned a type and fails to typecheck. Also the hypothesis $\overline{x : A}$ may be used not just once but as many times as necessary. Hence proof term M in $\lambda x\!:\!A.\, M$ may contain any number of occurrences of variable x, as illustrated below:

$$\frac{\dfrac{\overline{x : B} \quad \overline{y : A}\ \textit{(not used in the proof)}}{\lambda y\!:\!A.\, x : A \supset B}\ \supset\mathsf{I}}{\lambda x\!:\!B.\, \lambda y\!:\!A.\, x : B \supset (A \supset B)}\ \supset\mathsf{I} \qquad \frac{\overline{x : A}}{\lambda x\!:\!A.\, x : A \supset A}\ \supset\mathsf{I} \qquad \frac{\dfrac{\overline{x : A} \quad \overline{x : A}}{(x, x) : A \wedge A}\ \wedge\mathsf{I}}{\lambda x\!:\!A.\, (x, x) : A \supset (A \wedge A)}\ \supset\mathsf{I}$$

(See Page **??** for proofs of corresponding truth judgments.)

As a proof term corresponding to the rule $\supset\mathsf{E}$, we use a *λ-application* $M\ N$:

$$\frac{A \supset B\ \mathit{true} \quad A\ \mathit{true}}{B\ \mathit{true}}\ \supset\mathsf{E} \quad\Longleftrightarrow\quad \frac{M : A \supset B \quad N : A}{M\ N : B}\ \supset\mathsf{E}$$

The following example uses the rule $\supset\mathsf{E}$ to typecheck $\lambda x\!:\!A \supset B.\, \lambda y\!:\!A.\, x\ y$:

$$\dfrac{\dfrac{\dfrac{\overline{x : A \supset B} \quad \overline{y : A}}{x\ y : B}\ \supset\mathsf{E}}{\lambda y\!:\!A.\, x\ y : A \supset B}\ \supset\mathsf{I}}{\lambda x\!:\!A \supset B.\, \lambda y\!:\!A.\, x\ y : (A \supset B) \supset (A \supset B)}\ \supset\mathsf{I}$$

Disjunction

As proof terms corresponding to the rule $\vee\mathsf{I_L}$ and $\vee\mathsf{I_R}$, we use *injection terms* $\mathsf{inl}_A\ M$ and $\mathsf{inr}_A\ M$; inl and inr stand for 'injection left' and 'injection right,' respectively:

$$\frac{A\ \mathit{true}}{A \vee B\ \mathit{true}}\ \vee\mathsf{I_L} \quad\Longleftrightarrow\quad \frac{M : A}{\mathsf{inl}_B\ M : A \vee B}\ \vee\mathsf{I_L} \qquad \frac{B\ \mathit{true}}{A \vee B\ \mathit{true}}\ \vee\mathsf{I_R} \quad\Longleftrightarrow\quad \frac{M : B}{\mathsf{inr}_A\ M : A \vee B}\ \vee\mathsf{I_R}$$

We annotate an injection term $\mathsf{inl}_A\ M$ or $\mathsf{inr}_A\ M$ with a type A so that whenever M typechecks, the whole injection term also typechecks with a unique type.

For the elimination rule $\vee\mathsf{E}$, we use a *case term* $\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N'$; as with the rule $\supset\mathsf{I}$, we represent hypotheses $\overline{A\ \mathit{true}}^{\,x}$ and $\overline{B\ \mathit{true}}^{\,y}$ in the premise as variables x and y:

$$\dfrac{\begin{array}{ccc} & \overline{A\ \mathit{true}}^{\,x} & \overline{B\ \mathit{true}}^{\,y} \\ & \vdots & \vdots \\ A \vee B\ \mathit{true} & C\ \mathit{true} & C\ \mathit{true} \end{array}}{C\ \mathit{true}}\ \vee\mathsf{E}^{x,y} \quad\Longleftrightarrow\quad \dfrac{\begin{array}{ccc} & \overline{x : A} & \overline{y : B} \\ & \vdots & \vdots \\ M : A \vee B & N : C & N' : C \end{array}}{\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' : C}\ \vee\mathsf{E}$$

Variables x and y are bound in the case term $\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N'$, and remain valid only within N and N', respectively. As an example, here is a proof term of type $(A \vee B) \supset (B \vee A)$:

$$\dfrac{\dfrac{\overline{x : A \vee B} \quad \dfrac{\overline{y : A}}{\mathsf{inr}_B\ y : B \vee A}\ \vee\mathsf{I_R} \quad \dfrac{\overline{z : B}}{\mathsf{inl}_A\ z : B \vee A}\ \vee\mathsf{I_L}}{\mathsf{case}\ x\ \mathsf{of}\ \mathsf{inl}\ y \Rightarrow \mathsf{inr}_B\ y \mid \mathsf{inr}\ z \Rightarrow \mathsf{inl}_A\ z : B \vee A}\ \vee\mathsf{E}}{\lambda x\!:\!A \vee B.\, \mathsf{case}\ x\ \mathsf{of}\ \mathsf{inl}\ y \Rightarrow \mathsf{inr}_B\ y \mid \mathsf{inr}\ z \Rightarrow \mathsf{inl}_A\ z : (A \vee B) \supset (B \vee A)}\ \supset\mathsf{I}$$

Truth and falsehood

We use a *unit term* $\langle\rangle$ as a proof term for $\top\ \mathit{true}$:

$$\frac{}{\top\ \mathit{true}}\ \top\mathsf{I} \quad\Longleftrightarrow\quad \frac{}{\langle\rangle : \top}\ \top\mathsf{I}$$

Just like there is no logical content in $\top\ \mathit{true}$, a unit term carries no useful information. As truth $\top$ has no elimination rule, there is no more rule for $\langle\rangle$.

Since falsehood $\bot$ has no introduction rule, there is no proof term for type $\bot$. For the elimination rule $\bot\mathsf{E}$, we use an *abort term* $\mathsf{abort}_C\ M$:

$$\frac{\bot\ \mathit{true}}{C\ \mathit{true}}\ \bot\mathsf{E} \quad\Longleftrightarrow\quad \frac{M : \bot}{\mathsf{abort}_C\ M : C}\ \bot\mathsf{E}$$

We annotate an abort term with a type C so that an unambiguous type can be assigned when M has type $\bot$.

$$\frac{M : A \quad N : B}{(M, N) : A \wedge B}\ \wedge\mathsf{I} \qquad \frac{M : A \wedge B}{\mathsf{fst}\ M : A}\ \wedge\mathsf{E_L} \qquad \frac{M : A \wedge B}{\mathsf{snd}\ M : B}\ \wedge\mathsf{E_R} \qquad \frac{\begin{array}{c}\overline{x : A}\\ \vdots \\ M : B\end{array}}{\lambda x{:}A.\, M : A \supset B}\ \supset\mathsf{I} \qquad \frac{M : A \supset B \quad N : A}{M\ N : B}\ \supset\mathsf{E}$$

$$\frac{M : A}{\mathsf{inl}_B\ M : A \vee B}\ \vee\mathsf{I_L} \qquad \frac{M : B}{\mathsf{inr}_A\ M : A \vee B}\ \vee\mathsf{I_R} \qquad \frac{M : A \vee B \quad \begin{array}{c}\overline{x : A}\\ \vdots \\ N : C\end{array} \quad \begin{array}{c}\overline{y : B}\\ \vdots \\ N' : C\end{array}}{\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' : C}\ \vee\mathsf{E}$$

$$\frac{}{\langle\rangle : \top}\ \top\mathsf{I} \qquad \frac{M : \bot}{\mathsf{abort}_C\ M : C}\ \bot\mathsf{E}$$

Figure 1.1: Typing rules for proof terms for propositional logic

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}\ \mathsf{Hyp} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x{:}A.\, M : A \supset B}\ \supset\mathsf{I} \qquad \frac{\Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B}\ \supset\mathsf{E}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M, N) : A \wedge B}\ \wedge\mathsf{I} \qquad \frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash \mathsf{fst}\ M : A}\ \wedge\mathsf{E_L} \qquad \frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash \mathsf{snd}\ M : B}\ \wedge\mathsf{E_R}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{inl}_B\ M : A \vee B}\ \vee\mathsf{I_L} \qquad \frac{\Gamma \vdash M : B}{\Gamma \vdash \mathsf{inr}_A\ M : A \vee B}\ \vee\mathsf{I_R} \qquad \frac{\Gamma \vdash M : A \vee B \quad \Gamma, x : A \vdash N : C \quad \Gamma, y : B \vdash N' : C}{\Gamma \vdash \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' : C}\ \vee\mathsf{E}$$

$$\frac{}{\Gamma \vdash \langle\rangle : \top}\ \top\mathsf{I} \qquad \frac{\Gamma \vdash M : \bot}{\Gamma \vdash \mathsf{abort}_C\ M : C}\ \bot\mathsf{E}$$

Figure 1.2: Typing rules using hypothetical judgments

1.2 Type system

Since hypothetical judgments are just a syntactic tool for displaying hypothetical proofs, it is straightforward to extend the translation in Section 1.1 to hypothetical judgments. We continue to use the same set of proof terms to represent hypothetical proofs, but use a new typing judgment with an entailment relation $\vdash$:

$$\overset{\mathcal{D}}{A_1\ \mathit{true}, \cdots, A_n\ \mathit{true} \vdash C\ \mathit{true}} \quad \Longleftrightarrow \quad x_1 : A_1, \cdots, x_n : A_n \vdash M : C$$

The new typing judgment chooses a fresh variable x_i to represent each hypothesis $\overline{A_i\ \mathit{true}}$. Note that the new typing judgment itself is an example of a hypothetical judgment such that antecedents are (typing) judgments of the form $x_i : A_i$ and the succedent is a (typing) judgment of the form $M : C$. For the sake of simplicity, we maintain the invariant that all variables in antecedents are distinct.

Figure 1.2 shows a system of typing rules, or a *type system*, based on the new typing judgment. An antecedent $x : A$ is called a *type binding* because it binds variable x to type A. Γ denotes a collection of type bindings, and is called a *typing context*. We assume that the exchange rule is built into the typing judgment (*i.e.*, we may reorder type bindings as we like). The other two structural properties are stated as follows:

Proposition 1.1 (Structural properties).
(Weakening) *If* $\Gamma \vdash M : C$, *then* $\Gamma, x : A \vdash M : C$.
(Contraction) *If* $\Gamma, x : A, x : A \vdash M : C$, *then* $\Gamma, x : A \vdash M : C$.

Proof. By induction on the structure of the proof of $\Gamma \vdash M : C$ and $\Gamma, x : A, x : A \vdash M : C$. □

Alternatively the proof of Proposition 1.1 may proceed by induction on the structure of proof term M. This is because the type system in Figure 1.2 is *syntax-directed*: the *syntactic* form of proof term M decides, or *directs*, a unique typing rule necessary for deducing a typing judgment $\Gamma \vdash M : C$. Hence, for example, if M is a λ-abstraction $\lambda y{:}C_1.\, M'$, then $\Gamma \vdash M : C$ is provable only by applying the rule $\supset$I, from which we conclude $\Gamma, y : C_1 \vdash M' : C_2$ (the premise of the rule $\supset$I) and $C = C_1 \supset C_2$. As an illustration, we give a proof of the weakening property for the case $M = \lambda y{:}C_1.\, M'$:

Case $M = \lambda y{:}C_1.\, M'$

$\Gamma, y : C_1 \vdash M' : C_2$ and $C = C_1 \supset C_2$	by the rule $\supset$I with $\Gamma \vdash M : C$
$\Gamma, x : A, y : C_1 \vdash M' : C_2$	by induction hypothesis on M'
$\Gamma, x : A \vdash \lambda y{:}C_1.\, M' : C_1 \supset C_2$	by the rule $\supset$I
$\Gamma, x : A \vdash M : C$	from $M = \lambda y{:}C_1.\, M'$ and $C = C_1 \supset C_2$

In essence, the entire proof of a typing judgment $\Gamma \vdash M : C$ can be reconstructed by analyzing proof term M, which implies that analyzing the structure of the proof of $\Gamma \vdash M : C$ is equivalent to analyzing the structure of proof term M.

As a special case of a hypothetical judgment, the typing judgment in Figure 1.2 satisfies the two general properties of hypothetical judgments: reflexivity and substitution principle. Reflexivity follows from the rule Hyp. For the substitution principle, we need an operation on proof terms that corresponds to $[\mathcal{D}/\overline{A\ true}]\mathcal{E}$, *i.e.*, a substitution of a proof $\mathcal{D}$ for a hypothesis $\overline{A\ true}$ in a hypothetical proof $\mathcal{E}$. Suppose that proof terms M and N represent proofs $\mathcal{D}$ and $\mathcal{E}$, respectively, and that we use a variable x to represent hypothesis $\overline{A\ true}$. Then $[\mathcal{D}/\overline{A\ true}]\mathcal{E}$ is literally translated to $[M/x]N$, which is our notation for substituting M for x in N. We define $[M/x]N$ inductively on the structure of N, where we assume $x \neq y$ and $x \neq z$:

$$
\begin{array}{rcl}
[M/x]x & = & M \\
[M/x]y & = & y \\
[M/x]\lambda x{:}A.\, N & = & \lambda x{:}A.\, N \\
[M/x]\lambda y{:}A.\, N & = & \lambda y{:}A.\, [M/x]N \\
[M/x](N_1\ N_2) & = & ([M/x]N_1)\ ([M/x]N_2) \\
[M/x](N_1, N_2) & = & ([M/x]N_1, [M/x]N_2) \\
[M/x]\mathsf{fst}\ N & = & \mathsf{fst}\ [M/x]N \\
[M/x]\mathsf{snd}\ N & = & \mathsf{snd}\ [M/x]N \\
[M/x]\mathsf{inl}_B\ N & = & \mathsf{inl}_B\ [M/x]N \\
[M/x]\mathsf{inr}_A\ N & = & \mathsf{inr}_A\ [M/x]N \\
[M/x]\mathsf{case}\ N\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N_1 \mid \mathsf{inr}\ x \Rightarrow N_2 & = & \mathsf{case}\ [M/x]N\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N_1 \mid \mathsf{inr}\ x \Rightarrow N_2 \\
[M/x]\mathsf{case}\ N\ \mathsf{of}\ \mathsf{inl}\ y \Rightarrow N_1 \mid \mathsf{inr}\ x \Rightarrow N_2 & = & \mathsf{case}\ [M/x]N\ \mathsf{of}\ \mathsf{inl}\ y \Rightarrow [M/x]N_1 \mid \mathsf{inr}\ x \Rightarrow N_2 \\
[M/x]\mathsf{case}\ N\ \mathsf{of}\ \mathsf{inl}\ y \Rightarrow N_1 \mid \mathsf{inr}\ z \Rightarrow N_2 & = & \mathsf{case}\ [M/x]N\ \mathsf{of}\ \mathsf{inl}\ y \Rightarrow [M/x]N_1 \mid \mathsf{inr}\ z \Rightarrow [M/x]N_2 \\
[M/x]\langle\rangle & = & \langle\rangle \\
[M/x]\mathsf{abort}_C\ N & = & \mathsf{abort}_C\ [M/x]N
\end{array}
$$

In the case of $[M/x]\lambda y{:}A.\, N$, we assume that y is not a *free variable* of M, where a free variable of M is a variable that is not bound in λ-abstractions or case terms within M. If y happens to be a free variable of M, we say that a *variable capture* occurs: y, a free variable before the substitution, turns into a bound variable after the substitution. For example, $\lambda y{:}A.\, (x, y)$ and $\lambda z{:}A.\, (x, z)$ represent the same proof, so $[y/x]\lambda y{:}A.\, (x, y)$ must be equivalent to $[y/x]\lambda z{:}A.\, (x, z) = \lambda z{:}A.\, (y, z)$, which still recognizes y as a free variable. A variable capture, however, occurs in $[y/x]\lambda y{:}A.\, (x, y)$ to yield $\lambda y{:}A.\, (y, y)$, in which y is used only as a bound variable. Thus, if a variable capture occurs in $[M/x]\lambda y{:}A.\, N$, we need to rename y to a different variable. A similar restriction applies to substitutions into case terms.

Theorem 1.2 (Substitution). *If* $\Gamma \vdash M : A$ *and* $\Gamma, x : A \vdash N : C$, *then* $\Gamma \vdash [M/x]N : C$.

Proof. By induction on the structure of the proof of $\Gamma, x : A \vdash N : C$. We may also use induction on the structure of proof term N.

We consider three cases Hyp, $\supset$I, and $\supset$E. In the case $\supset$I, we rename y as necessary so as to avoid variable captures.

Case $\dfrac{y : C \in \Gamma}{\Gamma, x : A \vdash y : C}\ \mathsf{Hyp}$ where $N = y$

$\Gamma \vdash y : C$ by the rule Hyp with $y : C \in \Gamma$

$\Gamma \vdash [M/x]y : C$ from $[M/x]y = y$

Case $\dfrac{}{\Gamma, x : A \vdash x : C}\ \mathsf{Hyp}$ where $N = x$ and $A = C$

$\Gamma \vdash M : C$ from the assumption $\Gamma \vdash M : A$

$\Gamma \vdash [M/x]x : C$ from $[M/x]x = M$

Case $\dfrac{\Gamma, x : A, y : C_1 \vdash N' : C_2}{\Gamma, x : A \vdash \lambda y{:}C_1.\, N' : C_1 \supset C_2}\ \supset\mathsf{I}$ where $N = \lambda y{:}C_1.\, N'$ and $C = C_1 \supset C_2$

$\Gamma, y : C_1 \vdash M : A$ by weakening $\Gamma \vdash M : A$

$\Gamma, y : C_1 \vdash [M/x]N' : C_2$ by IH on $\Gamma, x : A, y : C_1 \vdash N' : C_2$ with $\Gamma, y : C_1 \vdash M : A$

$\Gamma \vdash \lambda y{:}C_1.\, [M/x]N' : C_1 \supset C_2$ by the rule $\supset$I

$\Gamma \vdash [M/x]\lambda y{:}C_1.\, N' : C_1 \supset C_2$ from $\lambda y{:}C_1.\, [M/x]N' = [M/x]\lambda y{:}C_1.\, N'$

Case $\dfrac{\Gamma, x : A \vdash N_1 : C' \supset C \quad \Gamma, x : A \vdash N_2 : C'}{\Gamma, x : A \vdash N_1\ N_2 : C}\ \supset\mathsf{E}$ where $N = N_1\ N_2$

$\Gamma \vdash [M/x]N_1 : C' \supset C$ by IH on $\Gamma, x : A \vdash N_1 : C' \supset C$ with $\Gamma \vdash M : A$

$\Gamma \vdash [M/x]N_2 : C'$ by IH on $\Gamma, x : A \vdash N_2 : C'$ with $\Gamma \vdash M : A$

$\Gamma \vdash [M/x]N_1\ [M/x]N_2 : C$ by the rule $\supset$E

$\Gamma \vdash [M/x](N_1\ N_2) : C$ from $[M/x]N_1\ [M/x]N_2 = [M/x](N_1\ N_2)$

□

1.3 β-reductions and η-expansions

We have seen in Section **??** that a local reduction removes a detour in a proof of $A\ \mathit{true}$ to yield a reduced proof of the same judgment. Since a proof of $A\ \mathit{true}$ can be represented as a proof term of type A under the Curry-Howard isomorphism, a local reduction is translated to a reduction of a proof term to another proof term of the same type. We refer to such a reduction of a proof term as a *β-reduction*; we write $M \Longrightarrow_\beta N$ for a β-reduction of M to N.

It is easy to derive β-reductions of proof terms from local reductions of proofs. For example, we obtain a β-reduction of $\mathsf{fst}\ (M, N)$ to M as follows:

$$\dfrac{\dfrac{M : A \quad N : B}{(M, N) : A \wedge B}\ \wedge\mathsf{I}}{\mathsf{fst}\ (M, N) : A}\ \wedge\mathsf{E_L} \qquad \Longrightarrow_\beta \qquad M : A$$

The following diagram explains how to obtain a β-reduction from a local reduction removing a detour in which the rule $\supset$I is immediately followed by the rule $\supset$E:

$$\dfrac{\dfrac{\begin{array}{c}\overline{x : A}\\ \vdots\\ M : B\end{array}}{\lambda x{:}A.\, M : A \to B}\ \supset\mathsf{I} \qquad N : A}{(\lambda x{:}A.\, M)\ N : B}\ \supset\mathsf{E} \qquad \Longrightarrow_\beta \qquad \begin{array}{c}[N/x]x : A\\ \vdots\\ [N/x]M : B\end{array}$$

The same β-reduction from a proof using hypothetical judgments is obtained as follows:

$$\begin{array}{ccc}
\dfrac{\dfrac{\overset{\mathcal{D}}{\Gamma, A\ true \vdash B\ true}}{\Gamma \vdash A \supset B\ true}\supset\mathsf{I} \quad \dfrac{\overset{\mathcal{E}}{}}{\Gamma \vdash A\ true}}{\Gamma \vdash B\ true}\supset\mathsf{E} & \Longrightarrow_R & \begin{array}{c}[\mathcal{E}/A\ true]\mathcal{D} \\ \Gamma \vdash B\ true\end{array} \\
\Updownarrow & & \Updownarrow \\
\dfrac{\dfrac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x{:}A.\, M : A \supset B}\supset\mathsf{I} \quad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x{:}A.\, M)\ N : B}\supset\mathsf{E} & \Longrightarrow_\beta & \Gamma \vdash [N/x]M : B
\end{array}$$

In this way, we obtain the following β-reductions for proof terms:

$$\begin{array}{rcl}
(\lambda x{:}A.\, M)\ N & \Longrightarrow_\beta & [N/x]M \\
\mathsf{fst}\ (M, N) & \Longrightarrow_\beta & M \\
\mathsf{snd}\ (M, N) & \Longrightarrow_\beta & N \\
\mathsf{case}\ \mathsf{inl}_B\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' & \Longrightarrow_\beta & [M/x]N \\
\mathsf{case}\ \mathsf{inr}_A\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' & \Longrightarrow_\beta & [M/y]N'
\end{array}$$

A β-reduction preserves the type of the proof term being reduced. This is called the *subject reduction* property because a typing judgment $M : C$ may be regarded as a sentence whose subject is M and whose predicate is C. The proof exploits the syntax-directedness of the type system: for any proof term M, there is a unique typing rule R for deducing $\Gamma \vdash M : C$; hence a proof of $\Gamma \vdash M : C$ by the rule R implies that the premise of the rule R holds as well.

Theorem 1.3 (Subject reduction). *If $\Gamma \vdash M : C$ and $M \Longrightarrow_\beta M'$, then $\Gamma \vdash M' : C$.*

Proof. By case analysis of $M \Longrightarrow_\beta M'$. We show three representative cases; the remaining two cases are similar. The proof reuses metavariable M.

Case $(\lambda x{:}A.\, M)\ N \Longrightarrow_\beta [N/x]M$

$\Gamma \vdash (\lambda x{:}A.\, M)\ N : C$	assumption
$\Gamma \vdash \lambda x{:}A.\, M : A' \supset C$ and $\Gamma \vdash N : A'$	by the rule $\supset\mathsf{E}$
$\Gamma, x : A \vdash M : C$ and $A = A'$	by the rule $\supset\mathsf{I}$ with $\Gamma \vdash \lambda x{:}A.\, M : A' \supset C$
$\Gamma \vdash [N/x]M : C$	by Theorem 1.2 with $\Gamma, x : A \vdash M : C$ and $\Gamma \vdash N : A'$ and $A = A'$

Case $\mathsf{fst}\ (M, N) \Longrightarrow_\beta M$

$\Gamma \vdash \mathsf{fst}\ (M, N) : C$	assumption
$\Gamma \vdash (M, N) : C \wedge A$	by the rule $\wedge\mathsf{E_L}$
$\Gamma \vdash M : C$	by the rule $\wedge\mathsf{I}$

Case $\mathsf{case}\ \mathsf{inl}_B\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' \Longrightarrow_\beta [M/x]N$

$\Gamma \vdash \mathsf{case}\ \mathsf{inl}_B\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' : C$	assumption
$\Gamma \vdash \mathsf{inl}_B\ M : A' \vee B'$ and $\Gamma, x : A' \vdash N : C$ and $\Gamma, y : B' \vdash N' : C$	by the rule $\vee\mathsf{E}$
$\Gamma \vdash M : A'$ and $B = B'$	by the rule $\vee\mathsf{I_L}$ with $\Gamma \vdash \mathsf{inl}_B\ M : A' \vee B'$
$\Gamma \vdash [M/x]N : C$	by Theorem 1.2 with $\Gamma, x : A' \vdash N : C$ and $\Gamma \vdash M : A'$

□

The β-reduction relation $\Longrightarrow_\beta$ can be generalized to a *structural congruence relation* $\Longrightarrow$ such that $M \Longrightarrow M'$ holds if a β-reduction is applied to a subterm of M to yield M'. (Such a subterm is commonly called a *redex*, or a *red*ucible *ex*pression.) For example, $((\lambda x{:}A.\, M)\ N, N') \Longrightarrow ([N/x]M, N')$ holds because a subterm $(\lambda x{:}A.\, M)\ N$ reduces to $[N/x]M$ by a β-reduction.

$$\frac{M \Longrightarrow_\beta M'}{M \Longrightarrow M'} \qquad \frac{M \Longrightarrow M'}{\lambda x\!:\!A.\, M \Longrightarrow \lambda x\!:\!A.\, M'} \qquad \frac{M \Longrightarrow M'}{M\ N \Longrightarrow M'\ N} \qquad \frac{N \Longrightarrow N'}{M\ N \Longrightarrow M\ N'}$$

$$\frac{M \Longrightarrow M'}{(M, N) \Longrightarrow (M', N)} \qquad \frac{N \Longrightarrow N'}{(M, N) \Longrightarrow (M, N')} \qquad \frac{M \Longrightarrow M'}{\mathsf{fst}\ M \Longrightarrow \mathsf{fst}\ M'} \qquad \frac{M \Longrightarrow M'}{\mathsf{snd}\ M \Longrightarrow \mathsf{snd}\ M'}$$

$$\frac{M \Longrightarrow M'}{\mathsf{inl}_B\ M \Longrightarrow \mathsf{inl}_B\ M'} \qquad \frac{M \Longrightarrow M'}{\mathsf{inr}_A\ M \Longrightarrow \mathsf{inr}_A\ M'}$$

$$\frac{M \Longrightarrow M'}{\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' \Longrightarrow \mathsf{case}\ M'\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N'}$$

$$\frac{N \Longrightarrow N''}{\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' \Longrightarrow \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N'' \mid \mathsf{inr}\ y \Rightarrow N'}$$

$$\frac{N' \Longrightarrow N''}{\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N' \Longrightarrow \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N''}$$

$$\frac{M \Longrightarrow M'}{\mathsf{abort}_C\ M \Longrightarrow \mathsf{abort}_C\ M'}$$

Figure 1.3: Rules for the structural congruence relation $\Longrightarrow$

Figure 1.3 shows the rules for the structural congruence relation $\Longrightarrow$. Note that there *is* a rule for reducing an abort term $\mathsf{abort}_C\ M$. Since a proof term M may contain multiple subterms to which β-reductions are applicable, the relation $\Longrightarrow$ is non-deterministic: given a proof term M, it does not always determine a unique proof term M' such that $M \Longrightarrow M'$. Theorem 1.3 now extends to the *subterm subject reduction* property:

Theorem 1.4 (Subterm subject reduction). *If* $\Gamma \vdash M : A$ *and* $M \Longrightarrow M'$*, then* $\Gamma \vdash M' : A$.

Proof. By induction on the structure of the proof of $M \Longrightarrow M'$. The proof uses Theorem 1.3. □

Like local reductions, local expansions are translated to expansions of proof terms under the Curry-Howard isomorphism. We refer to such expansions of proof terms as η-expansions; we write $M \Longrightarrow_\eta N$ for an η-expansion of M to N. Like a β-reduction, an η-expansion preserves the type of the proof term being expanded.

$$\begin{array}{rcll}
M : A \supset B & \Longrightarrow_\eta & \lambda x\!:\!A.\, M\ x & (x \text{ is not free in } M) \\
M : A \wedge B & \Longrightarrow_\eta & (\mathsf{fst}\ M, \mathsf{snd}\ M) & \\
M : A \vee B & \Longrightarrow_\eta & \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow \mathsf{inl}_B\ x \mid \mathsf{inr}\ y \Rightarrow \mathsf{inr}_A\ y & \\
M : \top & \Longrightarrow_\eta & \langle\rangle & \\
M : \bot & \Longrightarrow_\eta & \mathsf{abort}_\bot\ M &
\end{array}$$

For example, the η-expansions for $\wedge$ and $\supset$ are obtained as follows:

$$M : A \wedge B \qquad \Longrightarrow_\eta \qquad \frac{\dfrac{M : A \wedge B}{\mathsf{fst}\ M : A}\ \wedge\mathsf{E_L} \quad \dfrac{M : A \wedge B}{\mathsf{snd}\ M : B}\ \wedge\mathsf{E_R}}{(\mathsf{fst}\ M, \mathsf{snd}\ M) : A \wedge B}\ \wedge\mathsf{I}$$

$$M : A \supset B \qquad \Longrightarrow_\eta \qquad \frac{\dfrac{M : A \supset B \quad \overline{x : A}}{M\ x : B}\ \supset\mathsf{E}}{\lambda x\!:\!A.\, M\ x : A \supset B}\ \supset\mathsf{I}$$

1.4 Terms in normal form

Since it is a special case of a proof of $A\ true$, a proof of a neutral judgment $A\downarrow$ or a normal judgment $A\uparrow$ can be represented as a proof term of a special form under the Curry-Howard isomorphism. We use an

elim(ination) term E to represent a proof of $A\downarrow$ and an *intro(duction) term* I to represent a proof of $A\uparrow$:

$$\overset{\mathcal{D}}{A\downarrow} \quad \Longleftrightarrow \quad E : A \qquad\qquad \overset{\mathcal{E}}{A\uparrow} \quad \Longleftrightarrow \quad I : A$$

Then the inference rules for neutral and normal judgments (in Figure **??** or Figure **??**) are translated to the following definition of elim terms and intro terms:

$$\begin{array}{lrcl} \text{elim term} & E & ::= & x \mid E\ I \mid \mathsf{fst}\ E \mid \mathsf{snd}\ E \\ \text{intro term} & I & ::= & E \mid \lambda x{:}A.\, I \mid (I, I) \mid \mathsf{inl}_A\ I \mid \mathsf{inr}_A\ I \mid \mathsf{case}\ E\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow I \mid \mathsf{inr}\ x \Rightarrow I \mid \langle\rangle \mid \mathsf{abort}_A\ E \end{array}$$

For example, the rule $\mathsf{Hyp}_\downarrow$ specifies that variables be elim terms:

$$\frac{}{\Gamma_\downarrow, A\downarrow \vdash A\downarrow}\ \mathsf{Hyp}_\downarrow \quad \Longleftrightarrow \quad \frac{}{\Gamma, x : A \vdash x : A}\ \mathsf{Hyp}$$

The rule $\supset\mathsf{I}_\uparrow$ explains why $\lambda x{:}A.\, I$ is defined as an intro term; similarly the rule $\supset\mathsf{E}_\downarrow$ explains why $E\ I$ is defined as an elim term:

$$\frac{\Gamma_\downarrow, A\downarrow \vdash B\uparrow}{\Gamma_\downarrow \vdash A \supset B\uparrow}\ \supset\mathsf{I}_\uparrow \quad \Longleftrightarrow \quad \frac{\Gamma, x : A \vdash I : B}{\Gamma \vdash \lambda x{:}A.\, I : A \supset B}\ \supset\mathsf{I}$$

$$\frac{\Gamma_\downarrow \vdash A \supset B\downarrow \quad \Gamma_\downarrow \vdash A\uparrow}{\Gamma_\downarrow \vdash B\downarrow}\ \supset\mathsf{E}_\downarrow \quad \Longleftrightarrow \quad \frac{\Gamma \vdash E : A \supset B \quad \Gamma \vdash I : A}{\Gamma \vdash E\ I : B}\ \supset\mathsf{E}$$

Note also that the inclusion of elim terms as intro terms, not the other way around, is based on the rule $\downarrow\uparrow$.

With the definition of intro and elim terms, we can rewrite Theorem **??** as the following normalization theorem for proof terms. For the moment, we do not consider proof terms for disjunction $\vee$ and falsehood $\bot$. We write $\Longrightarrow^*$ for the reflexive and transitive closure of $\Longrightarrow$.

Theorem 1.5 (Normalization).
For every proof term M such that $\Gamma \vdash M : A$, there exists an intro term I such that $M \Longrightarrow^* I$.

Here is an example of a sequence of reductions from an ordinary proof term to an intro term, or simply a *normalization sequence*; the subterm being reduced at each step is underlined:

$$(\lambda x{:}A.\, \mathsf{fst}\ (x, z))\ \underline{\mathsf{fst}\ (y, z)} \Longrightarrow \underline{(\lambda x{:}A.\, \mathsf{fst}\ (x, z))\ y} \Longrightarrow \underline{\mathsf{fst}\ (y, z)} \Longrightarrow y$$

There are five alternative normalization sequences:

$$\begin{array}{l} (\lambda x{:}A.\, \underline{\mathsf{fst}\ (x, z)})\ \mathsf{fst}\ (y, z) \Longrightarrow \underline{(\lambda x{:}A.\, x)\ \mathsf{fst}\ (y, z)} \Longrightarrow \underline{\mathsf{fst}\ (y, z)} \Longrightarrow y \\ (\lambda x{:}A.\, \underline{\mathsf{fst}\ (x, z)})\ \mathsf{fst}\ (y, z) \Longrightarrow (\lambda x{:}A.\, x)\ \underline{\mathsf{fst}\ (y, z)} \Longrightarrow \underline{(\lambda x{:}A.\, x)\ y} \Longrightarrow y \\ (\lambda x{:}A.\, \mathsf{fst}\ (x, z))\ \underline{\mathsf{fst}\ (y, z)} \Longrightarrow (\lambda x{:}A.\, \underline{\mathsf{fst}\ (x, z)})\ y \Longrightarrow \underline{(\lambda x{:}A.\, x)\ y} \Longrightarrow y \\ \underline{(\lambda x{:}A.\, \mathsf{fst}\ (x, z))\ \mathsf{fst}\ (y, z)} \Longrightarrow \mathsf{fst}\ \underline{(\mathsf{fst}\ (y, z), z)} \Longrightarrow \underline{\mathsf{fst}\ (y, z)} \Longrightarrow y \\ \underline{(\lambda x{:}A.\, \mathsf{fst}\ (x, z))\ \mathsf{fst}\ (y, z)} \Longrightarrow \mathsf{fst}\ (\underline{\mathsf{fst}\ (y, z)}, z) \Longrightarrow \underline{\mathsf{fst}\ (y, z)} \Longrightarrow y \end{array}$$

Two other important properties of proof terms are *strong normalization* and *confluence*. Combined together, these two properties show that *every* normalization sequence produces a *unique* intro term.

Theorem 1.6 (Strong normalization, or termination).
For any proof term M such that $\Gamma \vdash M : A$, there is no infinite normalization sequence $M \Longrightarrow M_1 \Longrightarrow M_2 \Longrightarrow \cdots$.

Theorem 1.7 (Confluence, or Church-Rosser property).
Suppose $\Gamma \vdash M : A$. If $M \Longrightarrow^* N_1$ and $M \Longrightarrow^* N_2$, then there exists a proof term N such that $N_1 \Longrightarrow^* N$ and $N_2 \Longrightarrow^* N$.

In order for the normalization theorem to hold in the presence of proof terms for $\vee$ and $\bot$, the definition of $\Longrightarrow$ needs to be extended by incorporating commuting conversions for proof terms. A commuting conversion is allowed when a case term $\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N'$ appears in a position *where an elim term is expected.* To simplify the definition of a commuting conversion, we use a *commuting conversion context* κ, which is a proof term with a hole $\Box$ in a position where an elim term is expected. We write $\kappa[\![M]\!]$ for a proof term obtained by filling the hole in κ with M. (Note that κ is *not* defined inductively.)

$$\text{commuting conversion context} \qquad \kappa \quad ::= \quad \Box\ M \mid \mathsf{fst}\ \Box \mid \mathsf{snd}\ \Box \mid \mathsf{case}\ \Box\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow M \mid \mathsf{inr}\ x \Rightarrow M \mid \mathsf{abort}_A\ \Box$$

Then a commuting conversion of M to N, written as $M \Longrightarrow_c N$, is defined as follows:

$$\kappa[\![\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow N \mid \mathsf{inr}\ y \Rightarrow N']\!] \quad \Longrightarrow_c \quad \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x \Rightarrow \kappa[\![N]\!] \mid \mathsf{inr}\ y \Rightarrow \kappa[\![N']\!]$$

By extending the definition $\Longrightarrow$ with the following rule, we can show that the normalization theorem holds for all kinds of proof terms:

$$\frac{M \Longrightarrow_c M'}{M \Longrightarrow M'}$$

It can also be shown that strong normalization and confluence continue to hold.