CSE-490 Logic in Computer Science
Course Notes

Sungwoo Park

Fall 2006

Draft of January 27, 2007

This document is in draft form and is likely to contain errors.
Please do not distribute this document outside class.

Preface

This is a collection of course notes for CSE-490 *Logic in Computer Science* at POSTECH. The material is largely based on course notes for *15-399 Constructive Logic*, *15-815 Automated Theorem Proving*, and *15-816 Linear Logic*, all by Frank Pfenning at Carnegie Mellon University.

Any comments and suggestions will be greatly appreciated. I especially welcome feedback from students as to which part is difficult to follow and which part needs to be improved. The less background you have in logic and proof theory, the more useful your comments will be. So please do not hesitate if you are taking this course!

Contents

Chapter 1

Propositional Logic

This chapter develops *propositional logic*, *i.e.*, logic without universal or existential quantifiers. We formulate propositional logic in the judgmental style of Pfenning and Davies [**?**], which adopts Martin-Löf's methodology of distinguishing between *propositions* and *judgments* [**?**]. It differs from the traditional style of formulating logic which relies solely on propositions.

1.1 Propositions and judgments

In a judgmental formulation of logic, a proposition is an object of verification whose *truth* can be checked by inference rules, whereas a judgment is an object of knowledge which becomes evident by a *proof*. Examples of propositions are *'1 + 1 is equal to 0'* and *'1 + 1 is equal to 2'*, both under inference rules based on arithmetic. Examples of judgments are "*'1 + 1 is equal to 0' is true*", for which there is no proof, and "*'1 + 1 is equal to 2' is true,*" for which there is a proof.

To clarify the difference between propositions and judgments, consider a statement *'the moon is made of cheese.'* The statement is not yet an object of verification, or a proposition, since there is no way to check its truth — it becomes a proposition only when an inference rule is given. Here is an example of such an inference rule (written in a pedantic way):

$$\frac{\textit{'the moon is greenish white and has holes in it' is true}}{\textit{'the moon is made of cheese' is true}}\ \mathit{MoonCheese}$$

Now we can attempt to verify the proposition, for example, by taking a picture of the moon. That is, we still do not know whether the proposition is true or not, but by virtue of the inference rule, we know at least what counts as a verification of it. If the picture indeed shows that the moon is greenish white and has holes in it, the inference rule makes evident the judgment "*'the moon is made of cheese' is true.*" Now we know "*'the moon is made of cheese' is true*" by the proof consisting of the picture and the inference rule. Thus a proposition is an object of verification which may or may not be true, whereas a judgment is an object of knowledge which we either know or do not know, depending on the existence of a proof.

It is important that the notion of judgment takes priority over the notion of proposition. Simply put, the notion of judgment does not depend on the notion of proposition, and we must introduce new kinds of judgments without using particular propositions. On the other hand, propositions are always explained with existing judgments, which include at least truth judgments because propositions must be accompanied by inference rules for establishing their truth.

In developing a formal system of propositional logic, we use two judgments: $A\ \mathit{prop}$ and $A\ \mathit{true}$.

$$\begin{array}{lcl} A\ \mathit{prop} & \Leftrightarrow & \textit{A is a proposition} \\ A\ \mathit{true} & \Leftrightarrow & \textit{A is true} \end{array}$$

$A\ prop$ becomes evident by the presence of an inference rule deducing $A\ true$. We will inductively define the set of propositions using binary connectives (*e.g.*, implication $\supset$, conjunction $\wedge$, disjunction $\vee$) and unary connectives (*e.g.*, negation $\neg$). The inference rules will be designed in such a way that the definition of a connective does not involve another connective. We say that the resultant system is *orthogonal* in the sense that all connectives can be developed independently of each other.

Exercise 1.1. Suppose that $\neg A$ is a proposition standing for the logical negation of A and that $A\ false$ is a falsehood judgment denoting "*A cannot be true.*" What is wrong with the rule $\dfrac{\neg A\ true}{A\ false}\ \neg\mathsf{E}$ as a means of explaining the notion of falsehood judgments? What about $\dfrac{A\ false}{\neg A\ true}\ \neg\mathsf{I}$?

Exercise 1.2. Why is the rule $\dfrac{\neg A \vee B\ true}{A \supset B\ true}\ \supset\mathsf{I}$ bad, apart from its strange meaning?

1.2 Natural deduction system for propositional logic

Natural deduction [**?**] is a principle for building a system of logic whose main concepts are *introduction* and *elimination rules*. An introduction rule explains how to deduce a truth judgment involving a particular connective, exploiting those judgments in the premise. That is, it explains how to "introduce" the connective in a derivation (when read in the top-down way). For example, an introduction rule for the conjunction connective would look like:

$$\frac{\cdots}{A \wedge B\ true}\ \wedge\mathsf{I}$$

A dual concept is an elimination rule which explains how to exploit a truth judgment involving a particular connective to deduce another judgment in the conclusion. That is, it explains how to "eliminate" the connective in a derivation (when read in the top-down way). For example, an elimination rule for the conjunction connective would look like:

$$\frac{A \wedge B\ true}{\cdots}\ \wedge\mathsf{E}$$

An introduction rule usually conveys the intuition behind a connective and is thus relatively easy to design. In contrast, an elimination rule extracts the knowledge represented by a judgment and careful design is required to ensure that the resultant system is sound and complete in a sense to be explained in Section 1.5. For example, an ill-designed elimination rule may be so strong as to extract false knowledge that cannot be justified by its corresponding introduction rule. Or it may be too weak to deduce any interesting judgment. Note that an introduction rule takes precedence over its corresponding elimination rule because without an introduction rule, there is no use in designing an elimination rule. That is, an elimination rule cannot be considered separately from its corresponding introduction rule whereas the design of an introduction rule can be an isolated task.

Below we develop a natural deduction system for propositional logic, beginning with the conjunction connective $\wedge$ (which is the easiest case).

Conjunction

Before we investigate inference rules for $\wedge$, we need to know how to build valid propositions involving $\wedge$. Hence we need a *formation rule* to state that $A \wedge B$, read as "*A and B*" or "*A conjunction B,*" is a proposition if both A and B are propositions:

$$\frac{A\ prop \quad B\ prop}{A \wedge B\ prop}\ \wedge\mathsf{F}$$

In order to justify the rule $\wedge\mathsf{F}$, we need an inference rule for proving the truth of $A \wedge B$ on the assumption that there are inference rules for proving the truth of A and B. Since $A \wedge B$ is intended to be true whenever

both A and B are true, we use the following introduction rule to admit $A \wedge B$ as a proposition:

$$\frac{A\ true \quad B\ true}{A \wedge B\ true}\ \wedge\mathsf{I}$$

The rule $\wedge\mathsf{I}$ says that if both A and B are true, then $A \wedge B$ is true. It follows the usual interpretation of an inference rule: if the premise holds, then the conclusion holds. Now we may use the rule $\wedge\mathsf{I}$ to construct a proof of $A \wedge B\ true$ from a proof $\mathcal{D}_A$ of $A\ true$ and a proof $\mathcal{D}_B$ of $B\ true$; we write $\begin{array}{c}\mathcal{D}_A \\ A\ true\end{array}$ to mean that $\mathcal{D}_A$ is a proof of $A\ true$, including the last inference rule whose conclusion is $A\ true$:

$$\frac{\begin{array}{c}\mathcal{D}_A \\ A\ true\end{array} \quad \begin{array}{c}\mathcal{D}_B \\ B\ true\end{array}}{A \wedge B\ true}\ \wedge\mathsf{I}$$

The design of an elimination rule for $\wedge$ begins with $A \wedge B\ true$ as a premise. Since $A \wedge B\ true$ expresses that both A and B are true, we may conclude either $A\ true$ or $B\ true$ from $A \wedge B\ true$, as shown in the two elimination rules for $\wedge$:

$$\frac{A \wedge B\ true}{A\ true}\ \wedge\mathsf{E_L} \qquad \frac{A \wedge B\ true}{B\ true}\ \wedge\mathsf{E_R}$$

Implication

The implication connective $\supset$ requires the notion of a *hypothetical proof* which is a proof containing *hypotheses*. We read $A \supset B$ as "*A implies B*" or "*if A, then B*," and use the following formation rule:

$$\frac{A\ prop \quad B\ prop}{A \supset B\ prop}\ \supset\mathsf{F}$$

The intuition behind $\supset$ is that $A \supset B\ true$ holds whenever $A\ true$ implies $B\ true$, or a hypothesis of $A\ true$ leads to a proof of $B\ true$. We write a hypothesis of $A\ true$ as $\overline{A\ true}$, and obtain the following introduction rule for $\supset$:

$$\frac{\begin{array}{c}\overline{A\ true}^{\,x} \\ \vdots \\ B\ true\end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x$$

The premise of the rule $\supset\mathsf{I}^x$ is an example of a hypothetical proof because it contains a hypothesis, *i.e.*, a judgment that is assumed to hold. We say that the rule $\supset\mathsf{I}$ *internalizes* the hypothetical proof in its premise as a proposition $A \supset B$ in the sense that the truth of $A \supset B$ compactly represents the knowledge expressed by the hypothetical proof.

There are three observations to make about the rule $\supset\mathsf{I}^x$. First we annotate both the hypothesis $\overline{A\ true}$ and the rule name $\supset\mathsf{I}$ with the same label x. Thus a label in a hypothesis indicates from which inference rule the hypothesis originates. It is not necessary to annotate all hypotheses with different labels as long as no conflict occurs between two hypotheses with the same label. For example, the following derivation is okay even though both hypotheses are annotated with the same label x:

$$\frac{\dfrac{\begin{array}{c}\overline{A\ true}^{\,x} \\ \vdots \\ B\ true\end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x \quad \dfrac{\begin{array}{c}\overline{A'\ true}^{\,x} \\ \vdots \\ B'\ true\end{array}}{A' \supset B'\ true}\ \supset\mathsf{I}^x}{(A \supset B) \wedge (A' \supset B')\ true}$$

Second the hypothesis $\overline{A\ true}^{\,x}$ remains in effect only within the premise of the rule $\supset\mathsf{I}^x$. In other words, its scope is restricted to the premise of the rule $\supset\mathsf{I}^x$. After the rule $\supset\mathsf{I}^x$ is applied to deduce $A \supset B\ true$, $\overline{A\ true}^{\,x}$ may no longer be used as a valid hypothesis. We say that a hypothesis is *discharged* when its corresponding inference rule is applied and its scope is exited.

Note that while the premise of the rule $\supset\mathsf{I}^x$ is a hypothetical proof, the whole proof itself is *not* a hypothetical proof. Specifically the proof $\mathcal{D}$ below is a hypothetical proof, but the proof $\mathcal{E}$ is not:

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

The reason why $\mathcal{E}$ is not a hypothetical proof is that the hypothesis $\overline{A\ true}^{\,x}$ is discharged when the rule $\supset\mathsf{I}^x$ is applied, and thus is not visible to the outside. That is, we are free to use any hypothesis without turning the whole proof into a hypothetical proof as long as it is eventually discharged.

Third the hypothesis $\overline{A\ true}^{\,x}$ may be used not just once but as many times as necessary. In fact, we may even ignore it in the proof without using it at all. Here are examples of proofs that ignore $\overline{A\ true}^{\,x}$, use it once, and use it twice:

$$\dfrac{\dfrac{\overline{B\ true}^{\,y} \quad \begin{array}{c}\overline{A\ true}^{\,x}\\ \textit{(not used in the proof)}\end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x}{B \supset (A \supset B)\ true}\ \supset\mathsf{I}^y \qquad \dfrac{\overline{A\ true}^{\,x}}{A \supset A\ true}\ \supset\mathsf{I}^x \qquad \dfrac{\dfrac{\overline{A\ true}^{\,x} \quad \overline{A\ true}^{\,x}}{A \wedge A\ true}\ \wedge\mathsf{I}}{A \supset (A \wedge A)\ true}\ \supset\mathsf{I}^x$$

As with the elimination rules for $\wedge$, the design of the elimination rule for $\supset$ begins with a premise $A \supset B\ true$. Since $A \supset B\ true$ expresses that $A\ true$ implies $B\ true$, the only way to exploit it is by supplying a proof of $A\ true$ to conclude $B\ true$. Hence the elimination rule for $\supset$ uses both $A \supset B\ true$ and $A\ true$ as its premises:

$$\frac{A \supset B\ true \quad A\ true}{B\ true}\ \supset\mathsf{E}$$

The following example proves $(A \supset B) \supset (A \supset B)\ true$ using the rule $\supset\mathsf{E}$:

$$\dfrac{\dfrac{\dfrac{\overline{A \supset B\ true}^{\,x} \quad \overline{A\ true}^{\,y}}{B\ true}\ \supset\mathsf{E}}{A \supset B\ true}\ \supset\mathsf{I}^y}{(A \supset B) \supset (A \supset B)\ true}\ \supset\mathsf{I}^x$$

Disjunction

Like $\wedge$ and $\supset$, the disjunction connective $\vee$ is binary:

$$\frac{A\ prop \quad B\ prop}{A \vee B\ prop}\ \vee\mathsf{F}$$

$A \vee B$, read as "*A or B*" or "*A disjunction B*," is intended to be true when either A or B is true, but we do not necessarily know which alternative is true. In our formulation of propositional logic, an introduction rule for $\vee$ concludes $A \vee B\ true$ from a proof of either $A\ true$ or $B\ true$:

$$\frac{A\ true}{A \vee B\ true}\ \vee\mathsf{I}_\mathsf{L} \qquad \frac{B\ true}{A \vee B\ true}\ \vee\mathsf{I}_\mathsf{R}$$

The design of an elimination rule for $\vee$ is not obvious. A naive attempt would be to conclude one of $A\ true$ and $B\ true$ from $A \vee B\ true$:

$$\frac{A \vee B\ true}{A\ true}\ \vee\mathsf{E}_\mathsf{L}? \qquad \frac{A \vee B\ true}{B\ true}\ \vee\mathsf{E}_\mathsf{R}?$$

In a certain sense, both rules are too strong (or too powerful) because they conclude a judgment that cannot be justified by $A \vee B\ true$, which does not specify exactly which of $A\ true$ and $B\ true$ holds. In fact, each rule allows us to prove $A\ true$ for any proposition A:

$$\dfrac{\dfrac{\dfrac{\overline{B\ true}^{\,x}}{B \supset B\ true}\ \supset\mathsf{I}^x}{A \vee (B \supset B)\ true}\ \vee\mathsf{I}_\mathsf{R}}{A\ true}\ \vee\mathsf{E}_\mathsf{L}?$$

Since it is generally unknown which of $A\ true$ and $B\ true$ has been supplied in a proof of $A \vee B\ true$ (*e.g.*, when $A \vee B\ true$ is a hypothesis), the only logical way to exploit $A \vee B\ true$ is by considering both possibilities simultaneously. If we can prove $C\ true$ both from $A\ true$ and from $B\ true$ for a certain proposition C, then we may conclude $C\ true$ from $A \vee B\ true$, since $C\ true$ holds regardless of how the proof of $A \vee B\ true$ has been built. The elimination rule for $\vee$ expresses such a way of reasoning:

$$\dfrac{\begin{array}{ccc} & \overline{A\ true}^{\,x} & \overline{B\ true}^{\,y} \\ & \vdots & \vdots \\ A \vee B\ true & C\ true & C\ true \end{array}}{C\ true}\ \vee\mathsf{E}^{x,y}$$

Note that $A\ true$ and $B\ true$ are introduced as new hypotheses and are annotated with different labels x and y. As in the elimination rule for $\supset$, their scope is limited to their respective premises of the rule $\vee\mathsf{E}^{x,y}$ (*i.e.*, $\overline{A\ true}^{\,x}$ to the second premise and $\overline{B\ true}^{\,y}$ to the third premise), which means that both hypotheses are discharged when $C\ true$ is deduced in the conclusion.

Unlike the elimination rules for $\wedge$ and $\supset$, the elimination rule for $\vee$ exploits $A \vee B\ true$ in an indirect way in that its conclusion contains a proposition C that is not necessarily A, B, or their combination. That is, when applying the elimination rule to $A \vee B\ true$, we ourselves have to choose a proposition C (which can be completely unrelated to A and B) such that $C\ true$ is provable both from $A\ true$ and from $B\ true$. For this reason, the inclusion of $\vee$ in a system of logic makes it hard to investigate metalogical properties of the system, as we will see later.

As a trivial example, let us prove that $A\ true$ is stronger than $A \vee B\ true$:

$$\dfrac{\dfrac{\dfrac{\overline{A\ true}^{\,x}}{A \vee B\ true}\ \vee\mathsf{I}_\mathsf{L}}{A \supset (A \vee B)\ true}\ \supset\mathsf{I}^x$$

The converse does not hold, *i.e.*, $A \vee B\ true$ is strictly weaker than $A\ true$, because there is no way to prove $A\ true$ from $B\ true$ for arbitrary propositions A and B:

$$\dfrac{\dfrac{\begin{array}{ccc} & & \overline{B\ true}^{\,z} \\ & & \vdots \\ \overline{A \vee B\ true}^{\,x} & \overline{A\ true}^{\,y} & A\ true \quad \textit{(impossible)} \end{array}}{A\ true}}{(A \vee B) \supset A\ true}\ \begin{array}{l} \vee\mathsf{E}^{y,z} \\ \supset\mathsf{I}^x \end{array}$$

As another example, let us prove that the disjunction connective is commutative:

$$(A \vee B) \supset (B \vee A)\ true$$

We begin by applying the rule $\supset\mathsf{I}$ so that the problem reduces to proving $B \vee A\ \mathit{true}$ from $A \vee B\ \mathit{true}$:

$$\dfrac{\dfrac{\overline{A \vee B\ \mathit{true}}^{\,x}}{\vdots}}{(A \vee B) \supset (B \vee A)\ \mathit{true}}\ \supset\mathsf{I}^{x}$$

At this point, the proof may proceed either in a bottom-up way by applying an introduction rule $\vee\mathsf{I_L}$ or $\vee\mathsf{I_R}$ to $B \vee A\ \mathit{true}$, or in a top-down way by applying the elimination rule $\vee\mathsf{E}$ to $A \vee B\ \mathit{true}$. In the first case, we eventually get stuck because it is impossible to prove $A\ \mathit{true}$ or $B\ \mathit{true}$ from $A \vee B\ \mathit{true}$. For example, we cannot fill the gap in the proof shown below:

$$\dfrac{\dfrac{\dfrac{\overline{A \vee B\ \mathit{true}}^{\,x}}{\vdots}}{B \vee A\ \mathit{true}}\ \vee\mathsf{I_L}}{(A \vee B) \supset (B \vee A)\ \mathit{true}}\ \supset\mathsf{I}^{x}$$

In the second case, the problem reduces to separately proving $B \vee A\ \mathit{true}$ from $A\ \mathit{true}$ and from $B\ \mathit{true}$, which is accomplished by applying the introduction rules for $\vee$:

$$\dfrac{\dfrac{\overline{A \vee B\ \mathit{true}}^{\,x} \quad \dfrac{\overline{A\ \mathit{true}}^{\,y}}{B \vee A\ \mathit{true}}\ \vee\mathsf{I_R} \quad \dfrac{\overline{B\ \mathit{true}}^{\,z}}{B \vee A\ \mathit{true}}\ \vee\mathsf{I_L}}{B \vee A\ \mathit{true}}\ \vee\mathsf{E}^{y,z}}{(A \vee B) \supset (B \vee A)\ \mathit{true}}\ \supset\mathsf{I}^{x}$$

Exercise 1.3. We can rewrite the elimination rule for the disjunction connective by using the implication connective in place of hypothetical proofs:

$$\dfrac{A \vee B\ \mathit{true} \quad A \supset C\ \mathit{true} \quad B \supset C\ \mathit{true}}{C\ \mathit{true}}\ \vee\mathsf{E}$$

Why do we not use the new elimination rule which actually seems simpler than the previous one?

Truth and falsehood

Truth $\top$ is a proposition that is assumed to be always true. Hence a proof of $\top\ \mathit{true}$ requires no particular evidence and is always provable, as indicated by the empty premise in its introduction rule:

$$\dfrac{}{\top\ \mathit{prop}}\ \top\mathsf{F} \qquad \dfrac{}{\top\ \mathit{true}}\ \top\mathsf{I}$$

Then how do we exploit a proof of $\top\ \mathit{true}$ in an elimination rule? Since we have to provide no particular evidence in a proof of $\top\ \mathit{true}$, there is no logical content in it, which implies that there is no interesting way to exploit it. Therefore $\top$ has no elimination rule.

Falsehood $\bot$ is a proposition that is never true, or equivalently, whose truth is impossible to establish. The intuition is that it denotes a logical contradiction which must not be provable under any circumstance. Therefore there is no introduction rule for $\bot$. Interestingly, however, there *is* an elimination rule for $\bot$. Suppose that we have a proof of $\bot\ \mathit{true}$. If we think of $\bot\ \mathit{true}$ as something impossible to prove, or as something that is the most difficult to prove, the existence of its proof implies that we can prove everything (which is no more difficult to prove than $\bot\ \mathit{true}$)! Therefore the elimination rule for $\bot$ deduces $C\ \mathit{true}$ for an arbitrary proposition C:

$$\dfrac{}{\bot\ \mathit{prop}}\ \bot\mathsf{F} \qquad \dfrac{\bot\ \mathit{true}}{C\ \mathit{true}}\ \bot\mathsf{E}$$

Then why do we need an elimination rule for $\bot$ at all, if it is impossible to prove $\bot\ true$? While it is impossible to prove $\bot\ true$ out of nothing, it is possible to prove $\bot\ true$ using hypotheses. For example, $\bot\ true$ in the premise of the rule $\bot\mathsf{E}$ itself may be a hypothesis, as illustrated in the proof below:

$$\dfrac{\dfrac{\dfrac{}{\bot\ true}\,x}{C\ true}\,\bot\mathsf{E}}{\bot \supset C\ true}\,\supset\mathsf{I}^x$$

In essence, there is nothing wrong with reasoning from an assumption that something impossible to prove has been proven somehow.

We say that a system of logic is *inconsistent* if $\bot\ true$ is provable in it, and *consistent* if not. An inconsistent system is worthless because a judgment $A\ true$ is provable for an arbitrary proposition A. We will later present a proof that our system of propositional logic is consistent, whose discovery was in fact a major milestone in the history of logic.

Truth $\top$ and falsehood $\bot$ can also be viewed as the nullary cases of conjunction and disjunction, respectively. Consider a general n-ary case $\bigwedge_{i=1}^n A_i$ of conjunction with a single introduction rule and n elimination rules:

$$\dfrac{A_i\ true \quad \textit{for } i = 1, \cdots, n}{\bigwedge_{i=1}^n A_i\ true}\,\bigwedge\mathsf{I} \qquad \dfrac{\bigwedge_{i=1}^n A_i\ true}{A_i\ true}\,\bigwedge\mathsf{E}_i\ (1 \le i \le n)$$

If we let $\top = \bigwedge_{i=1}^n A_i$ with $n = 0$, the rule $\bigwedge\mathsf{I}$ turns into the rule $\top\mathsf{I}$ because it comes to have an empty premise, and each rule $\bigwedge\mathsf{E}_i$ disappears (*i.e.*, no elimination rule for $\top$). Similarly a general n-ary case $\bigvee_{i=1}^n A_i$ of disjunction has n introduction rules and a single elimination rule:

$$\dfrac{A_i\ true}{\bigvee_{i=1}^n A_i\ true}\,\bigvee\mathsf{I}_i(1 \le i \le n) \qquad \dfrac{\bigvee_{i=1}^n A_i\ true \qquad \begin{array}{c}\overline{A_i\ true}\,x_i \\ \vdots \\ C\ true\end{array} \quad \textit{for } i = 1, \cdots, n}{C\ true}\,\bigvee\mathsf{E}^x$$

If we let $\bot = \bigvee_{i=1}^n A_i$ with $n = 0$, each rule $\bigvee\mathsf{I}_i$ disappears (*i.e.*, no introduction rule for $\bot$), and the rule $\bigvee\mathsf{E}$ turns into the rule $\bot\mathsf{E}$ because all hypothetical proofs in its premise disappear.

Now it is clear that $\top$ and $\bot$ are identities for the binary connectives $\wedge$ and $\vee$, respectively. For example, we can identify $A \wedge \top$ with A: if $A\ true$ is provable, then $A \wedge \top\ true$ is also provable because $\top\ true$ automatically holds; the converse follows by the rule $\wedge\mathsf{E}_\mathsf{L}$. Similarly we can identify $A \vee \bot$ with A: if $A \vee \bot\ true$ is provable, $A\ true$ must also be provable because the second alternative $\bot\ true$ cannot be taken; the converse follows by the rule $\vee\mathsf{I}_\mathsf{L}$.

Negation

The only unary connective in propositional logic is negation $\neg$:

$$\dfrac{A\ prop}{\neg A\ prop}\,\neg\mathsf{F}$$

$\neg A$, read as *"not A"* or *"negation A,"* denotes the logical negation of A, and its truth means that A cannot be true. Below we consider three different approaches to designing inference rules for negation, all of which provide a means to express that A cannot be true.

The first approach is to define a falsehood judgment $A\ false$ denoting *"A cannot be true"* and then use the following rules to deduce and exploit $\neg A\ true$:

$$\dfrac{A\ false}{\neg A\ true}\,\neg\mathsf{I} \qquad \dfrac{\neg A\ true}{A\ false}\,\neg\mathsf{E}$$

(We do not discuss inference rules for deducing $A\ \mathit{false}$.) As in the rule $\supset\mathsf{I}$, we say that the rule $\neg\mathsf{I}$ internalizes $A\ \mathit{false}$ as a proposition $\neg A$ in the sense that the truth of $\neg A$ compactly represents the knowledge expressed by $A\ \mathit{false}$.

In the second approach, we deduce $\neg A\ \mathit{true}$ if an assumption of $A\ \mathit{true}$ leads to the provability of every truth judgment. The rationale is that if the system is known to be consistent (and thus not every truth judgment is provable), the provability of every truth judgment, *i.e.*, inconsistency of the system, as a consequence of an assumption of $A\ \mathit{true}$ implies that the assumption must be wrong, that is, A cannot be true.

In order to be able to express the provability of every truth judgment, we introduce a *propositional variable* p which stands for *any* proposition. We use a *parametric judgment* $p\ \mathit{true}$, or a judgment parametric in a propositional variable p, in the introduction rule for $\neg$:

$$\dfrac{\dfrac{}{A\ \mathit{true}}\,x \\ \vdots \\ p\ \mathit{true}}{\neg A\ \mathit{true}}\ \neg\mathsf{I}^{x,p}$$

Since the premise is a hypothetical judgment, we annotate the hypothesis $\overline{A\ \mathit{true}}$ and the rule name $\neg\mathsf{I}$ with the same label x. Moreover we annotate the rule name $\neg\mathsf{I}$ with the propositional variable p as well, since p is a fresh variable whose scope is restricted to the premise. The elimination rule for $\neg$ states that proofs of both $\neg A\ \mathit{true}$ and $A\ \mathit{true}$ license us to prove the truth of any proposition:

$$\dfrac{\neg A\ \mathit{true} \quad A\ \mathit{true}}{C\ \mathit{true}}\ \neg\mathsf{E}$$

Note that C in the conclusion can be any proposition, including propositional variables. As an example, we prove that A and $\neg A$ cannot be true simultaneously:

$$\dfrac{\dfrac{\dfrac{\overline{A \wedge \neg A\ \mathit{true}}\,x}{\neg A\ \mathit{true}}\ \wedge\mathsf{E_R} \quad \dfrac{\overline{A \wedge \neg A\ \mathit{true}}\,x}{A\ \mathit{true}}\ \wedge\mathsf{E_L}}{p\ \mathit{true}}\ \neg\mathsf{E}}{\neg(A \wedge \neg A)\ \mathit{true}}\ \neg\mathsf{I}^{x,p}$$

The third approach uses a *notational definition* by regarding $\neg A$ as a syntactic abbreviation of $A \supset \bot$. That is, $\neg$ plays no semantic role at all and $\neg A$ is simply expanded to $A \supset \bot$. The notational definition of $\neg$ justifies the following rules:

$$\dfrac{\dfrac{}{A\ \mathit{true}}\,x \\ \vdots \\ \bot\ \mathit{true}}{\neg A\ \mathit{true}}\ \neg\mathsf{I}^{x} \qquad \dfrac{\neg A\ \mathit{true} \quad A\ \mathit{true}}{\bot\ \mathit{true}}\ \neg\mathsf{E}$$

Note that if $\neg$ was defined as an independent connective rather than a notational convenience, these rules would destroy the orthogonality of the system because the meaning of $\neg$ would depend on the meaning of $\bot$. We use the third approach in our treatment of $\neg$ (which is the most popular definition in the literature).

As an example, we prove that if A is true, then $\neg A$ cannot be true:

$$\dfrac{\dfrac{\dfrac{\overline{\neg A\ \mathit{true}}\,y \quad \overline{A\ \mathit{true}}\,x}{\bot\ \mathit{true}}\ \neg\mathsf{E}}{\neg\neg A\ \mathit{true}}\ \neg\mathsf{I}^{y}}{A \supset \neg\neg A\ \mathit{true}}\ \supset\mathsf{I}^{x}$$

The converse $\neg\neg A \supset A\ \mathit{true}$ is *not* provable, however, which implies that $A\ \mathit{true}$ is strictly stronger than $\neg\neg A\ \mathit{true}$. That is, a proof that $\neg A$ cannot be true is not enough for concluding that A is true. A failed

$$\frac{A\ true \quad B\ true}{A \wedge B\ true}\ \wedge\mathsf{I} \qquad \frac{A \wedge B\ true}{A\ true}\ \wedge\mathsf{E_L} \qquad \frac{A \wedge B\ true}{B\ true}\ \wedge\mathsf{E_R}$$

$$\frac{\begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ B\ true\end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x \qquad \frac{A \supset B\ true \quad A\ true}{B\ true}\ \supset\mathsf{E}$$

$$\frac{A\ true}{A \vee B\ true}\ \vee\mathsf{I_L} \qquad \frac{B\ true}{A \vee B\ true}\ \vee\mathsf{I_R} \qquad \frac{A \vee B\ true \quad \begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ C\ true\end{array} \quad \begin{array}{c}\overline{B\ true}^{\,y}\\ \vdots\\ C\ true\end{array}}{C\ true}\ \vee\mathsf{E}^{x,y}$$

$$\frac{}{\top\ true}\ \top\mathsf{I} \qquad \frac{\bot\ true}{C\ true}\ \bot\mathsf{E} \qquad \frac{\begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ \bot\ true\end{array}}{\neg A\ true}\ \neg\mathsf{I}^x \qquad \frac{\neg A\ true \quad A\ true}{\bot\ true}\ \neg\mathsf{E}$$

Figure 1.1: Natural deduction system for propositional logic

attempt to prove $\neg\neg A \supset A\ true$ would look like:

$$\dfrac{\dfrac{\overline{\neg\neg A\ true}^{\,x} \qquad \begin{array}{c}\overline{\neg\neg A\ true}^{\,x}\\ \vdots\ ?\\ \neg A\ true\end{array}}{\dfrac{\bot\ true}{A\ true}\ \bot\mathsf{E}}\ \neg\mathsf{E}}{\neg\neg A \supset A\ true}\ \supset\mathsf{I}^x$$

The unprovability of $\neg\neg A \supset A\ true$ is a quintessential feature of the system of logic presented so far, or any system belonging to what is known as *constructive logic* or *intuitionistic logic*. In constructive logic, what $\neg A\ true$ proves is not exactly the direct opposite of what $A\ true$ proves. Rather it provides only indirect evidence that there is no proof of $A\ true$ by showing that the existence of such a proof leads to a logical contradiction. In contrast, *classical logic* assumes that every proposition is either true or false and has no intermediate state. Under classical logic, $\neg\neg A\ true$ is indistinguishable from $A\ true$ because A is either true or false and we have positive evidence that A cannot be false. The truth table method for proving the truth of a proposition is based on classical logic, which tries all possible combinations of truth and falsehood values for all atomic propositions. Until we come back to the topic of classical logic in Chapter **??**, we focus only on constructive logic.

Figure 1.1 shows all inference rules of propositional logic where the set of propositions is inductively defined as follows:

$$\text{proposition} \quad A \quad ::= \quad P \mid A \wedge A \mid A \supset A \mid A \vee A \mid \top \mid \bot \mid \neg A$$

P is called a *propositional constant* and denotes an atomic proposition (*e.g. '1 + 1 is equal to 0,' '1 + 1 is equal to 2' is true,' 'the moon is made of cheese,' etc*). The rules $\neg\mathsf{I}$ and $\neg\mathsf{E}$ are derived rules under the notational definition $\neg A = A \supset \bot$. From now on, we use the following operator precedence

$$\neg > \wedge > \vee > \supset$$

where $\wedge$, $\vee$, $\supset$ are all right-associative. Examples are:

$$\begin{array}{rcl} \neg A \wedge B & = & (\neg A) \wedge B \\ A \wedge B \vee C & = & (A \wedge B) \vee C \\ A \vee B \supset C & = & (A \vee B) \supset C \\ \neg A \wedge B \vee C \supset D & = & (((\neg A) \wedge B) \vee C) \supset D \end{array} \qquad \begin{array}{rcl} A \wedge B \wedge C & = & A \wedge (B \wedge C) \\ A \vee B \vee C & = & A \vee (B \vee C) \\ A \supset B \supset C & = & A \supset (B \supset C) \end{array}$$

1.3 Logical equivalence

We say that a proposition A is logically equivalent to another proposition B, written $A \equiv B$, if $A\ true$ implies $B\ true$ and vice versa. A notational definition of logical equivalence $A \equiv B$ is given as follows:

$$A \equiv B \quad = \quad (A \supset B) \wedge (B \supset A)\ true$$

If A and B are logically equivalent, an occurrence of A inside any proposition may be replaced by B (or an occurrence of B by A) without changing its meaning in that the resultant proposition remains logically equivalent to the original proposition. Thus logical equivalences enable us to simplify a proof involving a proposition that is logically equivalent to a less complex proposition. For example,

$$\neg\neg\neg A \supset (\neg\neg\neg B \supset \neg(A \vee B))\ true$$

becomes easy (or even obvious) to prove once we transform $\neg\neg\neg A \supset (\neg\neg\neg B \supset \neg(A \vee B))$ into $(\neg A \wedge \neg B) \supset \neg(A \vee B)$ by exploiting logical equivalences $\neg\neg\neg A \equiv \neg A$ and $A \supset (B \supset C) \equiv (A \wedge B) \supset C$.

Below we list logical equivalences of propositional logic which are divided into three groups.

Commutativity and idempotence. $\wedge$ and $\vee$ are commutative and idempotent. An implication $A \supset A$ is logically meaningless and reduces to $\top$.

(C1) $A \wedge B \equiv B \wedge A$
(C2) $A \vee B \equiv B \vee A$
(C3) $A \supset B \not\equiv B \supset A$
(I1) $A \wedge A \equiv A$
(I2) $A \vee A \equiv A$
(I3) $A \supset A \equiv \top$

Truth and falsehood. Each logical equivalence below deals with a proposition of the form $\top\ \phi\ A$, $\bot\ \phi\ A$, $A \supset \top$, or $A \supset \bot$ where ϕ is $\wedge$, $\vee$, or $\supset$.

(M1) $\top \wedge A \equiv A$
(M2) $\top \vee A \equiv \top$
(M3) $\top \supset A \equiv A$
(M4) $\bot \wedge A \equiv \bot$
(M5) $\bot \vee A \equiv A$
(M6) $\bot \supset A \equiv \top$
(M7) $A \supset \top \equiv \top$
(M8) $A \supset \bot = \neg A$

Interaction between connectives. Each logical equivalence below deals with a proposition of the form $A\ \phi\ (B\ \phi\ C)$ or $(A\ \phi\ B) \supset C$ where ϕ is $\wedge$, $\vee$, or $\supset$.

(L1)	$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$	*(associativity of $\wedge$)*
(L2)	$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$	*(distributivity of $\wedge$ over $\vee$)*
(L3)	$A \wedge (B \supset C) \equiv ?$	*(no interaction)*
(L4)	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$	*(distributivity of $\vee$ over $\wedge$)*
(L5)	$A \vee (B \vee C) \equiv (A \vee B) \vee C$	*(associativity of $\vee$)*
(L6)	$A \vee (B \supset C) \equiv ?$	*(no interaction)*
(L7)	$A \supset (B \wedge C) \equiv (A \supset B) \wedge (A \supset C)$	*(distributivity of $\supset$ over $\wedge$)*
(L8)	$A \supset (B \vee C) \equiv ?$	*(no interaction)*
(L9)	$A \supset (B \supset C) \equiv (A \wedge B) \supset C$	
(L10)	$(A \wedge B) \supset C \equiv A \supset (B \supset C)$	
(L11)	$(A \vee B) \supset C \equiv (A \supset C) \wedge (B \supset C)$	
(L12)	$(A \supset B) \supset C \equiv ?$	*(no interaction)*

1.4 Hypothetical judgments

While the rules in Figure 1.1 define a natural deduction system for propositional logic, they are unwieldy for writing hypothetical proofs. This is because no rule provides visual aid for keeping track of the scope of each hypothesis or an apparatus for preventing a hypothesis from escaping its scope. For example, the following hypothetical proof contains a wrong use of a hypothesis $\overline{A\ true}^{\,x}$ outside its scope:

$$\dfrac{\dfrac{\overline{A\ true}^{\,x}}{A \supset A\ true}\supset\mathsf{I}^x \quad \overline{A\ true}^{\,x}\textit{(wrong use)}}{(A \supset A) \wedge A\ true}\wedge\mathsf{I}$$

Whenever a new hypothesis is introduced, therefore, we could draw an imaginary contour to delineate its scope, as illustrated below:

$$\dfrac{\boxed{\begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ B\ true\end{array}}^{x}}{A \supset B\ true}\supset\mathsf{I}^x \qquad \dfrac{A \vee B\ true \qquad \boxed{\begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ C\ true\end{array}}^{x} \qquad \boxed{\begin{array}{c}\overline{B\ true}^{\,y}\\ \vdots\\ C\ true\end{array}}^{y}}{C\ true}\vee\mathsf{E}^{x,y}$$

Here the scope of a hypothesis $\overline{A\ true}^{\,x}$ is restricted to the contour labeled x. Note that a contour may be enclosed within another contour (as in a topography map), as shown in the following example:

$$\dfrac{\dfrac{\boxed{\begin{array}{l}\overline{A\ true}^{\,x}\\ \boxed{\dfrac{\overline{B\ true}^{\,y}}{A \wedge B\ true}\wedge\mathsf{I}}^{y}\end{array}}^{x}}{B \supset (A \wedge B)\ true}\supset\mathsf{I}^y}{A \supset (B \supset (A \wedge B))\ true}\supset\mathsf{I}^x$$

The hypothesis $\overline{A\ true}^{\,x}$ may be used inside the inner contour labeled y (which makes it possible to prove $A \wedge B\ true$ by the rule $\wedge\mathsf{I}$), but the hypothesis $\overline{B\ true}^{\,y}$ cannot be used outside the inner contour.

Hypothetical judgments provide a convenient way to keep track of the scope of each hypothesis in a hypothetical proof. A hypothetical judgment $J_1, \cdots, J_n \vdash J$ becomes evident by a hypothetical proof deducing a judgment J from a collection of hypotheses $\overline{J_1}, \cdots, \overline{J_n}$:

$$J_1, \cdots, J_n \vdash J \quad \Longleftrightarrow \quad \left\{\begin{array}{c}\overline{J_1} \quad \cdots \quad \overline{J_n}\\ \ddots\\ J\end{array}\right\} \text{ inference rules}$$

Thus we may read $J_1, \cdots, J_n \vdash J$ as "if judgments J_1, $\cdots$, J_n hold, then a judgment J hold." When $J_1, \cdots, J_n \vdash J$ holds, we say that J_1 through J_n *entail* J, or J is a consequence of J_1 through J_n. Hence $\vdash$ is called an *entailment relation* or a *consequence relation*. We refer to J_i, $1 \leq i \leq n$, as an *antecedent* and J as the *succedent*. We often abbreviate a collection of antecedents as Γ, as in a hypothetical judgment $\Gamma \vdash J$.

In developing a natural deduction system for propositional logic, we will use hypothetical judgments of the form $A_1\ true, \cdots, A_n\ true \vdash A\ true$ where antecedents and succedents are all truth judgments. Before presenting its inference rules, let us investigate properties of hypothetical judgments of the general form where antecedents and conclusions can be any judgments.

The definition of hypothetical judgments justifies two principles: *reflexivity* and *substitution principle*:

- (Reflexivity) $\Gamma, J, \Gamma' \vdash J$.
- (Substitution principle) If $\Gamma \vdash J$ and $\Gamma, J \vdash J'$, then $\Gamma \vdash J'$.

Reflexivity states that we may use a hypothesis $\overline{J}$ to conclude J. The substitution principle states that if we can prove a judgment J from a collection of hypotheses, we may use J as another hypothesis whenever the same collection of hypotheses is available. That is, we may use J as a lemma once we build a proof of $\Gamma \vdash J$.

To see how the substitution principle works, let us assume $\Gamma \vdash J$ and $\Gamma, J \vdash J'$ which imply that there are two hypothetical proofs $\mathcal{D}$ and $\mathcal{E}$ as shown below:

$$\Gamma \vdash J \iff \left.\begin{array}{c} \overline{\Gamma} \\ \vdots \\ J \end{array}\right\} \mathcal{D} \qquad\qquad \Gamma, J \vdash J' \iff \left.\begin{array}{c} \overline{\Gamma} \quad \cdots \quad \overline{J} \\ \ddots \\ J' \end{array}\right\} \mathcal{E}$$

Here $\overline{\Gamma}$ is a shorthand for $\{\overline{J} \,|\, J \in \Gamma\}$. Now we locate every occurrence of the hypothesis $\overline{J}$ in $\mathcal{E}$ and *substitute* $\mathcal{D}$ for it, which results in the following hypothetical proof:

$$\left.\begin{array}{c} \overline{\Gamma} \quad \cdots \quad J \\ \ddots \\ J' \end{array}\right\} \mathcal{D}$$

Since the same hypothesis (*e.g.*, one in $\overline{\Gamma}$) may be used as many times as necessary (see Page 4), the hypothetical judgment above makes evident the hypothetical judgment $\Gamma \vdash J'$, which is what the substitution principle concludes from $\Gamma \vdash J$ and $\Gamma, J \vdash J'$.

Our definition of hypothetical judgments makes two implicit assumptions: 1) the order of hypotheses is immaterial; 2) a hypothesis may be used zero or more times in a hypothetical proof. These assumptions are formally stated in the *structural properties* of hypothetical judgments:

- (Exchange) If $\Gamma, J_i, J_{i+1}, \Gamma' \vdash J$, then $\Gamma, J_{i+1}, J_i, \Gamma' \vdash J$.
- (Weakening) If $\Gamma, \Gamma' \vdash J$, then $\Gamma, J', \Gamma' \vdash J$ for any judgment J'.
- (Contraction) If $\Gamma, J_i, J_i, \Gamma' \vdash J$, then $\Gamma, J_i, \Gamma' \vdash J$.

Exchange states that we may ignore the order of antecedents in a hypothetical judgment. Weakening states that we may add a new antecedent without using it, thereby "weakening" what is being proven. ($\Gamma, J' \vdash J$ is weaker than $\Gamma \vdash J$ because it draws the same conclusion from more hypotheses.) By contraction, we may combine two copies of the same antecedent into one. Note that by weakening, a hypothesis may be used zero times and that by contraction, a hypothesis may be used more than once.

Here are a few further remarks on hypothetical judgments:

- Hypothetical judgments are just a *"convenient"* way, rather than a new way, to represent hypothetical proofs. That is, the entailment relation $\vdash$ is just a syntactic tool for displaying the hypotheses and the conclusion of a hypothetical judgment while hiding its internal structure, and thus does *not* introduce a new semantic notion. (In contrast, the relation $\models$ from model theory defines the notion of *semantic* consequence. Hence it has nothing to do with hypothetical judgments and is *not* a syntactic convenience.)

- There can be more than one hypothetical proof by which a given hypothetical judgment becomes evident, since hypothetical judgments is concerned only with hypotheses and conclusions.

- A hypothetical judgment itself is an example of a judgment and thus may be used as an antecedent or the conclusion in another hypothetical judgment, although we will not use hypothetical judgments in such a way in our discussion of logic. In fact, $J_1, \cdots, J_n \vdash J$ can be thought of as an abbreviation of a nested hypothetical judgment $J_1 \vdash (J_2 \vdash \cdots (J_n \vdash J) \cdots)$, where each antecedent J_i or the conclusion J may be another hypothetical judgment!

- While closely related to each other, a hypothetical judgment $J_1, \cdots, J_n \vdash J$ and a rule $\dfrac{J_1 \quad \cdots \quad J_n}{J}\ R$ are disparate concepts and thus impossible to compare for equivalence. The reason is simple: the former is a judgment whereas the latter is an inference rule. The existence of a proof of $J_1, \cdots, J_n \vdash J$ just implies that R is a derivable rule. Conversely, if the rule R is available, we can always prove $J_1, \cdots, J_n \vdash J$ with the following hypothetical proof:

  $$\dfrac{\overline{J_1} \quad \cdots \quad \overline{J_n}}{J}\ R$$

  Note, however, that the above hypothetical proof may not be the only way to prove $J_1, \cdots, J_n \vdash J$ if we can build another hypothetical proof $\begin{array}{c} \overline{J_1} \quad \cdots \quad \overline{J_n} \\ \vdots \\ J \end{array}$ without using the rule R at all.

- A hypothetical judgment $\cdot \vdash J$ with no antecedents is *not* equivalent to its succedent J. While the former states that J holds unconditionally (or categorically), the latter is unaware of whether there are hypotheses or not, and could be even a hypothesis in a hypothetical judgment. For example, from the assumption that J entails J' (*i.e.*, $J \vdash J'$), we can show that $\cdot \vdash J$ implies $\cdot \vdash J'$ by the substitution principle. The converse is not the case, however, because a proof of $\cdot \vdash J'$ does not necessarily extend a proof of $\cdot \vdash J$ so that J follows directly from J'. (If $\cdot \vdash J$ and J were equivalent, the converse would also be the case.)

- An important consequence of the structural properties is that the two hypothetical judgments in each rule, $\Gamma \vdash J$ from the *if* part and $\Gamma' \vdash J$ from the *then* part, represent hypothetical proofs not only of the same size (in terms of the number of applications of inference rules) but also of completely the same structure. As a result, when structural induction (or rule induction) is applicable to $\Gamma \vdash J$, *we may apply structural induction on* $\Gamma' \vdash J$ *instead.*

Figure 1.2 shows a natural deduction system for propositional logic using hypothetical judgments, which reuses the inference rule names from the previous natural deduction system. We use hypothetical judgments of the form $\Gamma \vdash A\ \mathit{true}$ where Γ is a collection of truth judgments and the exchange rule is built-in (*i.e.*, we may reorder antecedents as we like).

The rule Hyp expresses reflexivity of hypothetical judgments. All the other rules are justified by their counterparts in the previous natural deduction system. As an example, let us consider the rule $\supset$I. The premise $\Gamma, A\ \mathit{true} \vdash B\ \mathit{true}$ implies the existence of a hypothetical proof deducing $B\ \mathit{true}$ from hypotheses

$$\frac{A\ true \in \Gamma}{\Gamma \vdash A\ true}\ \mathsf{Hyp} \qquad \frac{\Gamma, A\ true \vdash B\ true}{\Gamma \vdash A \supset B\ true}\ \supset\mathsf{I} \qquad \frac{\Gamma \vdash A \supset B\ true \quad \Gamma \vdash A\ true}{\Gamma \vdash B\ true}\ \supset\mathsf{E}$$

$$\frac{\Gamma \vdash A\ true \quad \Gamma \vdash B\ true}{\Gamma \vdash A \wedge B\ true}\ \wedge\mathsf{I} \qquad \frac{\Gamma \vdash A \wedge B\ true}{\Gamma \vdash A\ true}\ \wedge\mathsf{E_L} \qquad \frac{\Gamma \vdash A \wedge B\ true}{\Gamma \vdash B\ true}\ \wedge\mathsf{E_R}$$

$$\frac{\Gamma \vdash A\ true}{\Gamma \vdash A \vee B\ true}\ \vee\mathsf{I_L} \qquad \frac{\Gamma \vdash B\ true}{\Gamma \vdash A \vee B\ true}\ \vee\mathsf{I_R} \qquad \frac{\Gamma \vdash A \vee B\ true \quad \Gamma, A\ true \vdash C\ true \quad \Gamma, B\ true \vdash C\ true}{\Gamma \vdash C\ true}\ \vee\mathsf{E}$$

$$\frac{}{\Gamma \vdash \top\ true}\ \top\mathsf{I} \qquad \frac{\Gamma \vdash \bot\ true}{\Gamma \vdash C\ true}\ \bot\mathsf{E} \qquad \frac{\Gamma, A\ true \vdash \bot}{\Gamma \vdash \neg A\ true}\ \neg\mathsf{I} \qquad \frac{\Gamma \vdash \neg A\ true \quad \Gamma \vdash A\ true}{\Gamma \vdash \bot\ true}\ \neg\mathsf{E}$$

Figure 1.2: Natural deduction system using hypothetical judgments

$\overline{\Gamma}$ and $\overline{A\ true}$ (where $\overline{\Gamma}$ is a shorthand for $\{\overline{J}\,|J \in \Gamma\}$):

$$\Gamma, A\ true \vdash B\ true \quad \Longleftrightarrow \quad \begin{array}{c} \overline{\Gamma} \quad \cdots \quad \overline{A\ true} \\ \ddots \\ B\ true \end{array}$$

Now we apply the rule $\supset$I (in Figure 1.1) to the conclusion $B\ true$ *with respect to the hypothesis* $\overline{A\ true}$. That is, the application of the rule $\supset$I designates $\overline{A\ true}$ as its corresponding hypothesis:

$$\frac{\begin{array}{c} \overline{\Gamma} \quad \cdots \quad \overline{A\ true}^{\,x} \\ \ddots \\ B\ true \end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x$$

Note that hypotheses in the proof include hypotheses $\overline{\Gamma}$ but *not* $\overline{A\ true}^{\,x}$, which is discharged when the rule $\supset$I is applied. That is, we have a hypothetical proof for the hypothetical judgment $\Gamma \vdash A \supset B$:

$$\frac{\begin{array}{c} \overline{\Gamma} \quad \cdots \quad \overline{A\ true}^{\,x} \\ \ddots \\ B\ true \end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x \qquad \Longleftrightarrow \quad \Gamma \vdash A \supset B\ true$$

Thus we can prove $\Gamma \vdash A \supset B\ true$ whenever we have a proof of $\Gamma, A\ true \vdash B\ true$, which justifies the rule $\supset$I in Figure 1.2.

The use of hypothetical judgments eliminates the need to annotate hypotheses with labels. For example, here is a proof of a hypothetical judgment $\cdot \vdash A \supset (B \supset (A \wedge B))\ true$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\ true, B\ true \vdash A\ true}\ \mathsf{Hyp} \quad \dfrac{}{A\ true, B\ true \vdash B\ true}\ \mathsf{Hyp}}{A\ true, B\ true \vdash A \wedge B\ true}\ \wedge\mathsf{I}}{A\ true \vdash B \supset (A \wedge B)\ true}\ \supset\mathsf{I}}{\cdot \vdash A \supset (B \supset (A \wedge B))\ true}\ \supset\mathsf{I}$$

There are two observations to make about the new natural deduction system. First each leaf in a derivation tree for $\Gamma \vdash C\ true$ (where $\Gamma \vdash C\ true$ is regarded as the root) is an application of either the rule Hyp or the rule $\top$I. In particular, if the rule $\top$I is not used (which is often the case), the derivation tree has the

following form:

$$\dfrac{}{\Gamma_1 \vdash A_1\ true}\ \mathsf{Hyp} \qquad \cdots \qquad \dfrac{}{\Gamma_n \vdash A_n\ true}\ \mathsf{Hyp}$$
$$\vdots$$
$$\Gamma \vdash C\ true$$

Second the set of antecedents always expands in an inference rule as we move from the conclusion to its premises (*i.e.*, in a bottom-up way). That is, an inference rule $\dfrac{\Gamma' \vdash A\ true \quad \cdots}{\Gamma \vdash C\ true}$ satisfies $\Gamma \subset \Gamma'$. Then the above derivation tree for $\Gamma \vdash C\ true$ satisfies $\Gamma \subset \Gamma_1, \cdots, \Gamma \subset \Gamma_n$.

Weakening and contraction are now stated as follows:

Proposition 1.4 (Structural properties).
(Weakening) *If* $\Gamma \vdash C\ true$, *then* $\Gamma, A\ true \vdash C\ true$.
(Contraction) *If* $\Gamma, A\ true, A\ true \vdash C\ true$, *then* $\Gamma, A\ true \vdash C\ true$.

Proof. By induction on the structure of the proof of $\Gamma \vdash C\ true$ and $\Gamma, A\ true, A\ true \vdash C\ true$. For weakening, the proof of $\Gamma, A\ true \vdash C\ true$ has exactly the same structure as the proof of $\Gamma \vdash C\ true$. When structural induction is applicable to $\Gamma \vdash C\ true$, therefore, we may apply structural induction on $\Gamma, A\ true \vdash C\ true$ instead. A similar observation holds for contraction. □

The provability of the substitution principle confirms that the system in Figure 1.2 adheres to the definition of hypothetical judgments. That is, if the substitution principle was unprovable, it would indicate that some rule in Figure 1.2 was not designed according to the relation between hypothetical judgments and hypothetical proofs.

Theorem 1.5 (Substitution). *If* $\Gamma \vdash A\ true$ *and* $\Gamma, A\ true \vdash C\ true$, *then* $\Gamma \vdash C\ true$.

Exercise 1.6. To which judgment do you think structural induction must be applied in the proof of Theorem 1.5? $\Gamma \vdash A\ true$ or $\Gamma, A\ true \vdash C\ true$? Why?

Before attempting to write a proof of Theorem 1.5, it is worthwhile to predict how the proof would proceed. It helps us, for example, to determine to which of $\Gamma \vdash A\ true$ and $\Gamma, A\ true \vdash C\ true$ structural induction must be applied. For the sake of simplicity, let us assume that the rule $\top\mathsf{I}$ is not used in the proof of $\Gamma, A\ true \vdash C\ true$. (The proof of $\Gamma \vdash A\ true$ may use the rule $\top\mathsf{I}$.) Then the derivation tree for $\Gamma, A\ true \vdash C\ true$ has the following form

$$\dfrac{}{\Gamma_1, A\ true \vdash C_1\ true}\ \mathsf{Hyp} \qquad \cdots \qquad \dfrac{}{\Gamma_n, A\ true \vdash C_n\ true}\ \mathsf{Hyp}$$
$$\vdots$$
$$\Gamma, A\ true \vdash C\ true$$

where each leaf $\dfrac{}{\Gamma_i, A\ true \vdash C_i\ true}\ \mathsf{Hyp}$ satisfies $\Gamma \subset \Gamma_i$ for $1 \leq i \leq n$. Now consider the i-the leaf. If $C_i\ true \in \Gamma_i$, the antecedent $A\ true$ in $\Gamma_i, A\ true \vdash C_i\ true$ plays no role in the proof and the leaf is safely replaced by $\dfrac{}{\Gamma_i \vdash C_i\ true}\ \mathsf{Hyp}$. If $C_i = A$, we weaken $\Gamma \vdash A\ true = \Gamma \vdash C_i\ true$ to obtain $\Gamma_i \vdash C_i\ true$, which is then *substituted* for $\Gamma_i, A\ true \vdash C_i\ true$. Now no leaf contains $A\ true$ as an antecedent, and by propagating these changes all the way down to the root, we transform the whole derivation tree into a new one for $\Gamma \vdash C\ true$. Thus we analyze the structure of the proof of $\Gamma, A\ true \vdash C\ true$ to located all leaves in it. That is, we apply structural induction on $\Gamma, A\ true \vdash C\ true$ rather than $\Gamma \vdash A\ true$.

Proof. By induction on the structure of the proof of $\Gamma, A\ true \vdash C\ true$.

We consider three cases Hyp, $\supset\mathsf{I}$, and $\supset\mathsf{E}$.

Case $\dfrac{C\ true \in \Gamma}{\Gamma, A\ true \vdash C\ true}\ \mathsf{Hyp}$
$\Gamma \vdash C\ true$ by the rule Hyp with $C\ true \in \Gamma$

Case $\dfrac{}{\Gamma, A\ true \vdash C\ true}\ \mathsf{Hyp}$ where $A = C$
$\Gamma \vdash C\ true$ from the assumption $\Gamma \vdash A\ true$

Case $\dfrac{\Gamma, A\ true, C_1\ true \vdash C_2\ true}{\Gamma, A\ true \vdash C_1 \supset C_2\ true}\ \supset\mathsf{I}$ where $C = C_1 \supset C_2$
$\Gamma, C_1\ true \vdash A\ true$ by weakening $\Gamma \vdash A\ true$
$\Gamma, C_1\ true \vdash C_2\ true$ by IH on $\Gamma, A\ true, C_1\ true \vdash C_2\ true$ with $\Gamma, C_1\ true \vdash A\ true$
$\Gamma \vdash C_1 \supset C_2\ true$ by the rule $\supset\mathsf{I}$ with $\Gamma, C_1\ true \vdash C_2\ true$

Case $\dfrac{\Gamma, A\ true \vdash C' \supset C\ true \quad \Gamma, A\ true \vdash C'\ true}{\Gamma, A\ true \vdash C\ true}\ \supset\mathsf{E}$
$\Gamma \vdash C' \supset C\ true$ by IH on $\Gamma, A\ true \vdash C' \supset C\ true$ with $\Gamma \vdash A\ true$
$\Gamma \vdash C'\ true$ by IH on $\Gamma, A\ true \vdash C'\ true$ with $\Gamma \vdash A\ true$
$\Gamma \vdash C\ true$ by the rule $\supset\mathsf{E}$ with $\Gamma \vdash C' \supset C\ true$ and $\Gamma \vdash C'\ true$

□

Thus, given a proof $\mathcal{D}$ of $\Gamma \vdash A\ true$ and a proof $\mathcal{E}$ of $\Gamma, A\ true \vdash C\ true$, we can always produce a proof, written as $[\mathcal{D}/A\ true]\mathcal{E}$, of $\Gamma \vdash C\ true$ by substituting $\mathcal{D}$ into $\mathcal{E}$.

1.5 Local soundness and completeness

All the inference rules presented so far seem to make sense intuitively, but their correctness is yet to be established in a formal way. For example, we would certainly expect an elimination rule for $\wedge$ by which $A\ true$ is deducible from $A \wedge B\ true$, but not an elimination rule that deduces $C\ true$ from $A \wedge B\ true$ if C is unrelated to A and B. Then, in designing a natural deduction system, what is the guiding principle to which we can appeal in order to decide whether to accept or reject an inference rule? The answer is that the system must satisfy two properties: *local soundness* and *local completeness*.

An introduction rule compresses the knowledge expressed in its premises into a truth judgment in the conclusion, whereas an elimination rule retrieves the knowledge compressed within a truth judgment in a premise to deduce another judgment in the conclusion. The local soundness property states that the knowledge retrieved from a judgment by an elimination rule is only part of the knowledge compressed within that judgment. Therefore, if local soundness fails, the elimination rule is too strong in the sense that it is capable of contriving some knowledge that cannot be justified by that judgment; thus local soundness ensures that the elimination rule is not too strong. The local completeness property states that the knowledge retrieved from a judgment by an elimination rule includes at least the knowledge compressed within that judgment. Therefore, if local completeness fails, the elimination rule is too weak in the sense that it is incapable of retrieving all the knowledge compressed within that judgment; thus local completeness ensures that the elimination rule is strong enough. If an elimination rule satisfies both properties, it retrieves exactly the same knowledge compressed within a judgment in a premise.

We verify the local soundness property by showing how to reduce a proof in which an introduction rule is immediately followed by a corresponding elimination rule. As an example, consider the following proof in which the introduction rule $\wedge\mathsf{I}$ is immediately followed by its corresponding elimination rule $\wedge\mathsf{E_L}$:

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ A\ true\end{matrix} \quad \begin{matrix}\mathcal{E}\\ B\ true\end{matrix}}{A \wedge B\ true}\ \wedge\mathsf{I}}{A\ true}\ \wedge\mathsf{E_L}$$

The rule $\wedge\mathsf{E_L}$ is not too strong because what it deduces in the conclusion, namely $A\ true$, is one of the two

judgments used to deduce $A \wedge B\ true$. Hence the whole proof reduces to a simpler proof $\mathcal{D}$:

$$\dfrac{\dfrac{\overset{\mathcal{D}}{A\ true} \quad \overset{\mathcal{E}}{B\ true}}{A \wedge B\ true}\ \wedge\mathsf{I}}{A\ true}\ \wedge\mathsf{E_L} \qquad \Longrightarrow_R \qquad \overset{\mathcal{D}}{A\ true}$$

If the rule $\wedge\mathsf{E_L}$ was too strong (*e.g.*, deducing $A \supset B\ true$ somehow), the proof would not be reducible.

As another example, consider the following proof in which the introduction rule $\supset\mathsf{I}$ is immediately followed by the elimination rule $\supset\mathsf{E}$:

$$\dfrac{\dfrac{\begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots \\ B\ true\end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x \quad \overset{\mathcal{D}}{A\ true}}{B\ true}\ \supset\mathsf{E}$$

The rule $\supset\mathsf{E}$ is not too strong because the whole proof reduces to a smaller proof of the same judgment $B\ true$ by substituting $\mathcal{D}$ for the hypothesis $\overline{A\ true}^{\,x}$ in the premise of the rule $\supset\mathsf{I}^x$:

$$\dfrac{\dfrac{\begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots \\ B\ true\end{array}}{A \supset B\ true}\ \supset\mathsf{I}^x \quad \overset{\mathcal{D}}{A\ true}}{B\ true}\ \supset\mathsf{E} \qquad \Longrightarrow_R \qquad \begin{array}{c}\overset{\mathcal{D}}{A\ true}\\ \vdots \\ B\ true\end{array}$$

For the natural deduction system based on hypothetical judgments, the substitution principle justifies $\Gamma \vdash B\ true$ when proofs of $\Gamma \vdash A\ true$ and $\Gamma, A\ true \vdash B\ true$ are given:

$$\dfrac{\dfrac{\overset{\mathcal{D}}{\Gamma, A\ true \vdash B\ true}}{\Gamma \vdash A \supset B\ true}\ \supset\mathsf{I} \quad \overset{\mathcal{E}}{\Gamma \vdash A\ true}}{\Gamma \vdash B\ true}\ \supset\mathsf{E} \qquad \Longrightarrow_R \qquad \begin{array}{c}[\mathcal{E}/A\ true]\mathcal{D}\\ \Gamma \vdash B\ true\end{array}$$

The case for $\vee$ is similar to the case for $\supset$ and uses the substitution principle:

$$\dfrac{\dfrac{\overset{\mathcal{D}}{A\ true}}{A \vee B\ true}\ \vee\mathsf{I_L} \quad \begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots \\ C\ true\end{array} \quad \begin{array}{c}\overline{B\ true}^{\,y}\\ \vdots \\ C\ true\end{array}}{C\ true}\ \vee\mathsf{E}^{x,y} \qquad \Longrightarrow_R \qquad \begin{array}{c}\overset{\mathcal{D}}{A\ true}\\ \vdots \\ C\ true\end{array}$$

$$\dfrac{\dfrac{\overset{\mathcal{D}}{\Gamma \vdash A\ true}}{\Gamma \vdash A \vee B\ true}\ \vee\mathsf{I_L} \quad \overset{\mathcal{E}_L}{\Gamma, A\ true \vdash C\ true} \quad \overset{\mathcal{E}_R}{\Gamma, B\ true \vdash C\ true}}{\Gamma \vdash C\ true}\ \vee\mathsf{E} \qquad \Longrightarrow_R \qquad \begin{array}{c}[\mathcal{D}/A\ true]\mathcal{E}_L\\ \Gamma \vdash C\ true\end{array}$$

We refer to these reductions $\Longrightarrow_R$ as *local reductions*. Note that there are no local reductions for $\top$ and $\bot$, since $\top$ has no elimination rule and $\bot$ has no introduction rule.

We verify the local completeness property by showing how to expand a proof of a judgment into another proof in which one or more elimination rules are followed by an introduction rule for the same judgment. As an example, consider a proof $\mathcal{D}$ of $A \wedge B\ true$. The elimination rules $\wedge\mathsf{E_L}$ and $\wedge\mathsf{E_R}$ are not too weak

because what they deduce in their conclusions, namely $A\ true$ and $B\ true$, are sufficient to reconstruct another proof of $A \wedge B\ true$:

$$\overset{\mathcal{D}}{A \wedge B\ true} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\overset{\mathcal{D}}{A \wedge B\ true}}{A\ true}\ \wedge\mathsf{E_L} \quad \dfrac{\overset{\mathcal{D}}{A \wedge B\ true}}{B\ true}\ \wedge\mathsf{E_R}}{A \wedge B\ true}\ \wedge\mathsf{I}$$

If the elimination rules were too weak (*e.g.*, being unable to deduce $A\ true$ somehow), the proof would not be expandable.

As another example, consider a proof $\mathcal{D}$ of $A \supset B\ true$. We can reconstruct another proof of the same judgment after applying the elimination rule $\supset\mathsf{E}$ to $\mathcal{D}$, which implies that the rule $\supset\mathsf{E}$ is not too weak:

$$\overset{\mathcal{D}}{A \supset B\ true} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\overset{\mathcal{D}}{A \supset B\ true} \quad \overline{A\ true}^{\,x}\ \supset\mathsf{E}}{B\ true}\ \supset\mathsf{I}^x}{A \supset B\ true}$$

In expanding the proof $\mathcal{D}$, we have to choose a fresh label x that is not already in use in $\mathcal{D}$, for any undischarged hypothesis $\overline{B\ true}^{\,x}$ with the same label x in $\mathcal{D}$ becomes associated with the rule $\supset\mathsf{I}^x$, resulting in an incorrect proof if $A \neq B$. For the natural deduction system based on hypothetical judgments, we weaken a proof $\mathcal{D}$ of $\Gamma \vdash A \supset B\ true$ to obtain a proof of $\Gamma, A\ true \vdash A \supset B\ true$ when reconstructing another proof of $\Gamma \vdash A \supset B\ true$:

$$\overset{\mathcal{D}}{\Gamma \vdash A \supset B\ true} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\Gamma, A\ true \vdash \overset{\mathcal{D}}{A \supset B\ true} \quad \overline{\Gamma, A\ true \vdash A\ true}\ \mathsf{Hyp}}{\Gamma, A\ true \vdash B\ true}\ \supset\mathsf{E}}{\Gamma \vdash A \supset B\ true}\ \supset\mathsf{I}$$

The case for $\vee$ is given as follows:

$$\overset{\mathcal{D}}{A \vee B\ true} \quad \Longrightarrow_E \quad \dfrac{\overset{\mathcal{D}}{A \vee B\ true} \quad \dfrac{\overline{A\ true}^{\,x}}{A \vee B\ true}\ \vee\mathsf{I_L} \quad \dfrac{\overline{B\ true}^{\,y}}{A \vee B\ true}\ \vee\mathsf{I_R}}{A \vee B\ true}\ \vee\mathsf{E}^{x,y}$$

$$\Gamma \vdash \overset{\mathcal{D}}{A \vee B\ true} \quad \Longrightarrow_E \quad \dfrac{\Gamma \vdash \overset{\mathcal{D}}{A \vee B\ true} \quad \dfrac{\overline{\Gamma, A\ true \vdash A\ true}\ \mathsf{Hyp}}{\Gamma, A\ true \vdash A \vee B\ true}\ \vee\mathsf{I_L} \quad \dfrac{\overline{\Gamma, B\ true \vdash B\ true}\ \mathsf{Hyp}}{\Gamma, B\ true \vdash A \vee B\ true}\ \vee\mathsf{I_R}}{\Gamma \vdash A \vee B\ true}\ \vee\mathsf{E}$$

We refer to these expansions $\Longrightarrow_E$ as *local expansions*.

Although there are no local reductions for $\top$ and $\bot$, there *are* local expansions for $\top$ and $\bot$. Recall that $\top$ and $\bot$ are the nullary cases of conjunction and disjunction, respectively. Hence a proof $\mathcal{D}$ of $\top\ true$ expands to another proof of $\top\ true$ that uses zero elimination rules and thus ignores $\mathcal{D}$:

$$\overset{\mathcal{D}}{\top\ true} \quad \Longrightarrow_E \quad \dfrac{}{\top\ true}\ \top\mathsf{I}$$

Similarly a proof of $\bot\ true$ expands to another proof of $\bot\ true$ that uses zero introduction rules:

$$\overset{\mathcal{D}}{\bot\ true} \quad \Longrightarrow_E \quad \dfrac{\overset{\mathcal{D}}{\bot\ true}}{\bot\ true}\ \bot\mathsf{E}$$

As every connective satisfies local soundness and completeness, the natural deduction system for propositional logic is said to be locally sound and complete. When the system is extended with a new connective, quantifier, or modality, we have to check that the system remains locally sound and complete by finding its local reduction and expansion, as we will see later.

1.6 Normal proofs

We have seen that a proof containing a *detour, i.e.*, an introduction rule immediately followed by a corresponding elimination rule, can be transformed to another proof by applying a local reduction. It turns out that for every proof of $A\ true$, there is a sequence of local reductions that lead to another proof of $A\ true$ containing no detour (the normalization theorem). We refer to the resultant proof as a *normal proof*. A normal proof is the most direct proof because a detour may be thought of as an example of indirect reasoning. Moreover it is minimal in size in a certain sense, irrespective of the size of its syntactic representation, because it does not reduce to another proof.

Since a normal proof contains no detour, it has the following form where top-down applications of elimination rules meet bottom-up applications of introduction rules in the middle:

$$\begin{array}{c}
\overline{A_1\ true}^{\,x_1} \quad \ldots \quad \overline{A_n\ true}^{\,x_n} \\
\downarrow \qquad \downarrow \qquad \downarrow \\
\text{elimination rules} \\
\downarrow \qquad \downarrow \\
\downarrow \\
\uparrow \\
\uparrow \qquad \uparrow \\
\text{introduction rules} \\
\uparrow \qquad \uparrow \qquad \uparrow \\
C\ true
\end{array}$$

Thus the structure of a normal proof conforms to our intuition in building a proof by repeatedly applying introduction rules in a bottom-up way, adding new hypotheses, and repeatedly applying elimination rules in a top-down way, starting from hypotheses. Here is an example of a normal proof of $(A \wedge B) \supset (B \wedge A)\ true$:

$$\dfrac{\dfrac{\dfrac{\overline{A \wedge B\ true}^{\,x}}{B\ true}\wedge\mathsf{E_R} \quad \dfrac{\overline{A \wedge B\ true}^{\,x}}{A\ true}\wedge\mathsf{E_L}}{B \wedge A\ true}\wedge\mathsf{I}}{(A \wedge B) \supset (B \wedge A)\ true}\supset\mathsf{I}^{x}$$

A proof of $(A \wedge B) \supset (B \wedge A)\ true$ that is not normal contains detours in it:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \wedge B\ true}^{\,x}}{A\ true}\wedge\mathsf{E_L} \quad \dfrac{\overline{A \wedge B\ true}^{\,x}}{B\ true}\wedge\mathsf{E_R}}{A \wedge B\ true}\wedge\mathsf{I}\ \textit{(detour)}}{B\ true}\wedge\mathsf{E_R} \quad \dfrac{\dfrac{\dfrac{\overline{A \wedge B\ true}^{\,x}}{A\ true}\wedge\mathsf{E_L} \quad \dfrac{\overline{A \wedge B\ true}^{\,x}}{B\ true}\wedge\mathsf{E_R}}{A \wedge B\ true}\wedge\mathsf{I}\ \textit{(detour)}}{A\ true}\wedge\mathsf{E_L}}{B \wedge A\ true}\wedge\mathsf{I}}{(A \wedge B) \supset (B \wedge A)\ true}\supset\mathsf{I}^{x}$$

Normal proofs are an indispensable tool in the study of logic because of their soundness and completeness properties: $A\ true$ holds if and only if there is a normal proof of $A\ true$. The soundness property holds trivially because a normal proof is just a proof of a special form. The completeness property (that every proof has a corresponding normal proof) has two important consequences. First, in order to prove $A\ true$, it suffices to find a normal proof of it. When proving $A\ true$, for example, it is safe to ignore proofs of the

$$\frac{\begin{array}{c}\overline{A\downarrow}^{\,x}\\ \vdots\\ B\uparrow\end{array}}{A\supset B\uparrow}\supset\mathsf{I}^x_\uparrow \qquad \frac{A\supset B\downarrow \quad A\uparrow}{B\downarrow}\supset\mathsf{E}_\downarrow \qquad \frac{A\uparrow \quad B\uparrow}{A\wedge B\uparrow}\wedge\mathsf{I}_\uparrow \qquad \frac{A\wedge B\downarrow}{A\downarrow}\wedge\mathsf{E}_{\mathsf{L}\downarrow} \qquad \frac{A\wedge B\downarrow}{B\downarrow}\wedge\mathsf{E}_{\mathsf{R}\downarrow}$$

$$\frac{A\uparrow}{A\vee B\uparrow}\vee\mathsf{I}_{\mathsf{L}\uparrow} \qquad \frac{B\uparrow}{A\vee B\uparrow}\vee\mathsf{I}_{\mathsf{R}\uparrow} \qquad \frac{A\vee B\downarrow \qquad \begin{array}{c}\overline{A\downarrow}^{\,x}\\ \vdots\\ C\uparrow\end{array} \qquad \begin{array}{c}\overline{B\downarrow}^{\,y}\\ \vdots\\ C\uparrow\end{array}}{C\uparrow}\vee\mathsf{E}^{x,y}_\uparrow$$

$$\frac{}{\top\uparrow}\top\mathsf{I}_\uparrow \qquad \frac{\bot\downarrow}{C\uparrow}\bot\mathsf{E}_\uparrow \qquad \frac{A\downarrow}{A\uparrow}\downuparrows$$

Figure 1.3: Inference rules for neutral and normal judgments

following form which cannot be normal proofs:

$$\frac{\begin{array}{c}\vdots\\ B\supset A\ \mathit{true}\end{array} \qquad \begin{array}{c}\vdots\\ B\ \mathit{true}\end{array}}{A\ \mathit{true}}\supset\mathsf{E}$$

That is, we need to concern ourselves only with the most direct proof rather an indirect proof, for example, by introducing an intermediate proposition B as shown above. Second, in order to refute $A\ \mathit{true}$, it suffices to try to build a normal proof of it (by alternating between bottom-up applications of introduction rules and top-down applications of elimination rules) and show that the process gets stuck or does not terminate.

Exercise 1.7. Give an informal argument why $\neg\neg A\supset A\ \mathit{true}$ is not provable.

To formalize all these ideas, we introduce two new judgments: *neutral judgments* $A\downarrow$ and *normal judgments* $A\uparrow$. A neutral judgment $A\downarrow$ becomes evident by a *neutral proof* of $A\ \mathit{true}$ which is either a hypothesis or an elimination rule applied to another neutral proof, whereas a normal judgment $A\uparrow$ becomes evident by a normal proof of $A\ \mathit{true}$ which is either a neutral judgment or an introduction rule applied to another normal proof. Thus the direction of the arrow in each judgment coincides with the direction in which the proof construction should proceed. Specifically we exploit an existing neutral judgment $A\downarrow$ in order to deduce another judgment by determining which elimination rule to be applied; hence the proof construction from a neutral judgment always proceeds downward ($\downarrow$). For a normal judgment $A\uparrow$ whose proof is incomplete yet, we determine which introduction rule must be applied in order to deduce it; hence the proof construction from a normal judgment always proceeds upward ($\uparrow$). When a neutral judgment $A\downarrow$ meets a normal judgment $A\uparrow$ in the middle, the proof construction is finished.

Figure 1.3 shows the inference rules for neutral and normal judgments. The rule $\downuparrows$, called the *coercion rule*, says that a neutral proof is a normal proof, and a typical construction of a normal proof is completed with an application of the rule $\downuparrows$. All the other rules are designed according to our intuition on building the most direct proofs, and thus are best read by following the direction of the arrow in each judgment. For example, suppose that we wish to build ($\uparrow$) a new proof of $A\supset B\ \mathit{true}$:

$$\begin{array}{c}\vdots\\ A\supset B\uparrow\end{array}$$

In order to build the most direct proof of $A\supset B\ \mathit{true}$, we first assume $A\ \mathit{true}$ as a hypothesis which is to be

exploited (↓) in deducing another judgment:

$$\begin{array}{c} \overline{A\downarrow} \\ \vdots \\ A \supset B\uparrow \end{array}$$

Then we try to build (↑) a proof of $B\ true$, which is precisely what the rule $\supset\mathsf{I}_\uparrow$ expresses:

$$\dfrac{\begin{array}{c} \overline{A\downarrow} \\ \vdots \\ B\uparrow \end{array}}{A \supset B\uparrow}$$

As another example, suppose that we wish to exploit (↓) an existing proof of $A \supset B\ true$:

$$\begin{array}{c} A \supset B\downarrow \\ \vdots \end{array}$$

In order to exploit it in the most direct manner, we need a proof of $A\ true$, which we do not have yet. Therefore we first build (↑) a new proof of $A\ true$:

$$\begin{array}{c} A \supset B\downarrow \quad A\uparrow \\ \vdots \end{array}$$

A proof of $A\ true$ then allows us to deduce $B\ true$. Since we now have a proof of $B\ true$ ready for use in deducing another judgment, we classify it as a neutral judgment, which is precisely what the rule $\supset\mathsf{E}_\downarrow$ expresses:

$$\dfrac{A \supset B\downarrow \quad A\uparrow}{B\downarrow}$$

Exercise 1.8. Analyze all the remaining rules in an analogous way. Note that the rule $\vee\mathsf{E}_\uparrow$ superficially deduces a normal judgment $C\uparrow$ by applying an elimination rule, thereby contradicting our intuition on a normal judgment which is supposed to be either a neutral judgment or an introduction rule applied to another normal judgment. The essence of the proof of $C\uparrow$, however, is found not in the application of the rule $\vee\mathsf{E}_\uparrow$ itself but in the two premises deducing $C\uparrow$. In this regard, the rule $\vee\mathsf{E}_\uparrow$ still adheres to our intuition on normal judgments. The rules $\top\mathsf{I}_\uparrow$ and $\bot\mathsf{E}_\uparrow$ are obtained as the nullary cases of $\wedge\mathsf{I}_\uparrow$ and $\vee\mathsf{E}_\uparrow$, respectively.

The rules in Figure 1.3 are all designed in such a way that a proof of a neutral or normal judgment contains no detour. First observe that no proof of a neutral judgment ends with an application of an introduction rule (see the rules $\supset\mathsf{E}_\downarrow$, $\wedge\mathsf{E}_{\mathsf{L}\downarrow}$, $\wedge\mathsf{E}_{\mathsf{R}\downarrow}$). Then observe that the principal premise in each elimination rule is a neutral judgment (*e.g.*, $A \supset B\downarrow$ in the rule $\supset\mathsf{E}_\downarrow$), which has been shown not to end with an introduction rule and thus does not give rise to a detour.

As an example, we show that the proof of $(A \wedge B) \supset (B \wedge A)\ true$ given earlier is indeed a normal proof by rewriting it in terms of neutral and normal judgments; we annotate each judgment in it with either ↓ or ↑ according to the rules in Figure 1.3 and check that no conflicting annotation arises:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{A \wedge B\downarrow}^{\,x}}{B\downarrow}\wedge\mathsf{E}_{\mathsf{R}\downarrow}}{B\uparrow}\downarrow\uparrow \quad \dfrac{\dfrac{\overline{A \wedge B\downarrow}^{\,x}}{A\downarrow}\wedge\mathsf{E}_{\mathsf{L}\downarrow}}{A\uparrow}\downarrow\uparrow}{B \wedge A\uparrow}\wedge\mathsf{I}_\uparrow}{(A \wedge B) \supset (B \wedge A)\uparrow}\supset\mathsf{I}^{x}_\uparrow$$

$$\frac{}{\Gamma_\downarrow, A\downarrow \vdash A\downarrow}\ \mathsf{Hyp}_\downarrow \qquad \frac{\Gamma_\downarrow, A\downarrow \vdash B\uparrow}{\Gamma_\downarrow \vdash A \supset B\uparrow}\ \supset\mathsf{I}_\uparrow \qquad \frac{\Gamma_\downarrow \vdash A \supset B\downarrow \quad \Gamma_\downarrow \vdash A\uparrow}{\Gamma_\downarrow \vdash B\downarrow}\ \supset\mathsf{E}_\downarrow$$

$$\frac{\Gamma_\downarrow \vdash A\uparrow \quad \Gamma_\downarrow \vdash B\uparrow}{\Gamma_\downarrow \vdash A \wedge B\uparrow}\ \wedge\mathsf{I}_\uparrow \qquad \frac{\Gamma_\downarrow \vdash A \wedge B\downarrow}{\Gamma_\downarrow \vdash A\downarrow}\ \wedge\mathsf{E}_{\mathsf{L}\downarrow} \qquad \frac{\Gamma_\downarrow \vdash A \wedge B\downarrow}{\Gamma_\downarrow \vdash B\downarrow}\ \wedge\mathsf{E}_{\mathsf{R}\downarrow}$$

$$\frac{\Gamma_\downarrow \vdash A\uparrow}{\Gamma_\downarrow \vdash A \vee B\uparrow}\ \vee\mathsf{I}_{\mathsf{L}\uparrow} \qquad \frac{\Gamma_\downarrow \vdash B\uparrow}{\Gamma_\downarrow \vdash A \vee B\uparrow}\ \vee\mathsf{I}_{\mathsf{R}\uparrow} \qquad \frac{\Gamma_\downarrow \vdash A \vee B\downarrow \quad \Gamma_\downarrow, A\downarrow \vdash C\uparrow \quad \Gamma_\downarrow, B\downarrow \vdash C\uparrow}{\Gamma_\downarrow \vdash C\uparrow}\ \vee\mathsf{E}_\uparrow$$

$$\frac{}{\Gamma_\downarrow \vdash \top\uparrow}\ \top\mathsf{I}_\uparrow \qquad \frac{\Gamma_\downarrow \vdash \bot\downarrow}{\Gamma_\downarrow \vdash C\uparrow}\ \bot\mathsf{E}_\uparrow \qquad \frac{\Gamma_\downarrow \vdash A\downarrow}{\Gamma_\downarrow \vdash A\uparrow}\ \downarrow\uparrow$$

Figure 1.4: Inference rules for neutral and normal judgments using hypothetical judgments

Note that a detour is impossible to annotate with arrows $\downarrow$ and $\uparrow$:

$$\frac{\dfrac{A\uparrow \quad B\uparrow}{A \wedge B\uparrow ?\downarrow ?}\ \wedge\mathsf{I}_\uparrow}{A\downarrow}\ \wedge\mathsf{E}_{\mathsf{L}\downarrow} \qquad \frac{\dfrac{\begin{array}{c}\overline{A\downarrow}^{\,x}\\ \vdots\\ B\uparrow\end{array}}{A \supset B\uparrow ?\downarrow ?}\ \supset\mathsf{I}^x_\uparrow \qquad A\uparrow}{B\downarrow}\ \supset\mathsf{E}_\downarrow \qquad \frac{A\uparrow}{A \vee B\uparrow ?\downarrow ?}\ \vee\mathsf{I}_{\mathsf{L}\uparrow} \quad \begin{array}{c}\overline{A\downarrow}^{\,x}\\ \vdots\\ C\uparrow\end{array} \quad \begin{array}{c}\overline{B\downarrow}^{\,y}\\ \vdots\\ C\uparrow\end{array}}{C\uparrow}\ \vee\mathsf{E}^{x,y}_\downarrow$$

As another example, we show that no proof of $A \vee \neg A\uparrow$ exists where A is an arbitrary proposition:

$$\frac{\textit{(stuck)}\quad A\uparrow}{A \vee \neg A\uparrow}\ \vee\mathsf{I}_{\mathsf{L}\uparrow} \qquad \frac{\begin{array}{c}\overline{A\downarrow}^{\,x}\\ \textit{(stuck)}\\ \vdots\\ \textit{(stuck)}\end{array}\quad \bot\uparrow}{\neg A\uparrow}\ \supset\mathsf{I}^x_\uparrow}{A \vee \neg A\uparrow}\ \vee\mathsf{I}_{\mathsf{R}\uparrow}$$

(Hence $A \vee \neg A$ *true* is not provable in constructive logic, although it is a tautology in classical logic.)

Figure 1.4 shows an equivalent system for neutral and normal judgments using hypothetical judgments $\Gamma_\downarrow \vdash A\uparrow$ and $\Gamma_\downarrow \vdash A\downarrow$, where $\Gamma_\downarrow = \{A\downarrow \mid A \in \Gamma\}$ is a collection of neutral judgments and the exchange rule is built-in; we reuse the inference rule names from the previous system for neutral and normal judgments. The two structural properties, weakening and contraction, are stated as expected. As it is based on hypothetical judgments, the system also satisfies the substitution principle.

Proposition 1.9 (Structural properties).

(Weakening) *If* $\Gamma_\downarrow \vdash C\downarrow$, *then* $\Gamma_\downarrow, A\downarrow \vdash C\downarrow$.
If $\Gamma_\downarrow \vdash C\uparrow$, *then* $\Gamma_\downarrow, A\downarrow \vdash C\uparrow$.

(Contraction) *If* $\Gamma_\downarrow, A\downarrow, A\downarrow \vdash C\downarrow$, *then* $\Gamma_\downarrow, A\downarrow \vdash C\downarrow$.
If $\Gamma_\downarrow, A\downarrow, A\downarrow \vdash C\uparrow$, *then* $\Gamma_\downarrow, A\downarrow \vdash C\uparrow$.

Theorem 1.10 (Substitution).

If $\Gamma_\downarrow \vdash A\downarrow$ *and* $\Gamma_\downarrow, A\downarrow \vdash C\downarrow$, *then* $\Gamma_\downarrow \vdash C\downarrow$.
If $\Gamma_\downarrow \vdash A\downarrow$ *and* $\Gamma_\downarrow, A\downarrow \vdash C\uparrow$, *then* $\Gamma_\downarrow \vdash C\uparrow$.

Proof. By induction on the structure of the proof of $\Gamma_\downarrow, A\downarrow \vdash C\downarrow$ and $\Gamma_\downarrow, A\downarrow \vdash C\uparrow$. □

1.7 Normalization

We state the soundness and completeness properties of normal proofs using normal judgments as follows:

Theorem 1.11.
(Soundness) *If $A\uparrow$ is provable, then $A\ true$ is provable.*
(Completeness) *If $A\ true$ is provable, then $A\uparrow$ is provable.*

The soundness property is easy to show because every rule in Figure 1.3 is transformed into a corresponding rule in Figure 1.1 by replacing all neutral and normal judgments by truth judgments (*i.e.*, $A\uparrow$ and $A\downarrow$ by $A\ true$). In conjunction with the fact that a proof of a neutral or normal judgment contains no detour, it implies that $A\uparrow$ expresses a particular strategy for proving $A\ true$, namely a strategy that disallows detours. For the completeness property, we prove the normalization theorem shown below. For the moment, we do not consider disjunction $\vee$ and falsehood $\bot$. (Chapter 3 gives another formal proof covering all connectives in propositional logic.)

Theorem 1.12 (Normalization). *For every proof of $A\ true$, there is a sequence of local reductions that lead to a proof of $A\uparrow$. That is, the proof of $A\ true$ resulting from the sequence of local reductions can be annotated with arrows $\downarrow$ and $\uparrow$ according to the rules in Figure 1.3.*

Another theorem, called the *strong normalization theorem*, states that every sequence of local reductions, *regardless of their order*, eventually terminates to yield a normal proof:

Theorem 1.13 (Strong normalization). *Every sequence of local reductions starting from a proof of $A\ true$ terminates with a proof of $A\uparrow$. That is, there is no infinite sequence of local reductions starting from any proof of $A\ true$.*

In the presence of disjunction $\vee$ and falsehood $\bot$ (*i.e.*, in the full system of propositional logic), however, the normalization theorem fails! That is, a proof to which no local reduction can be applied is *not* necessarily a normal proof. (The converse that a normal proof contains no detour still holds.) For example, the following proof of $(A \vee A) \supset A\ true$ is not normal, but contains no detour, either:

$$\dfrac{\dfrac{\dfrac{\overline{A\vee A\ true}^{\,z} \quad \dfrac{\overline{A\ true}^{\,x} \quad \overline{A\ true}^{\,x}}{A\wedge A\ true}\wedge\mathsf{I} \quad \dfrac{\overline{A\ true}^{\,y} \quad \overline{A\ true}^{\,y}}{A\wedge A\ true}\wedge\mathsf{I}}{A\wedge A\ true}\vee\mathsf{E}^{x,y}}{A\ true}\wedge\mathsf{E_L}}{(A\vee A)\supset A\ true}\supset\mathsf{I}^{z}$$

$$\dfrac{\dfrac{\dfrac{\overline{A{\downarrow}{\uparrow}}^{\,z} \quad \dfrac{\overline{A{\downarrow}{\uparrow}}^{\,x} \quad \overline{A{\downarrow}{\uparrow}}^{\,x}}{A\wedge A{\uparrow}}\wedge\mathsf{I}_{\uparrow} \quad \dfrac{\overline{A{\downarrow}{\uparrow}}^{\,y} \quad \overline{A{\downarrow}{\uparrow}}^{\,y}}{A\wedge A{\uparrow}}\wedge\mathsf{I}_{\uparrow}}{A\wedge A{\uparrow}?\,{\downarrow}?}\vee\mathsf{E}^{x,y}_{\downarrow}}{A{\downarrow}{\uparrow}}\wedge\mathsf{E}_{\mathsf{L}\downarrow}}{(A\vee A)\supset A{\uparrow}}\supset\mathsf{I}^{z}_{\uparrow}$$

Here and henceforth, we abbreviate the use of the rule $\dfrac{A\downarrow}{A\uparrow}\ {\downarrow}{\uparrow}$ as $A{\downarrow}{\uparrow}$.

The example suggests that in order to reduce an arbitrary proof to a normal proof, we need another strategy for transforming proofs involving disjunction $\vee$ and falsehood $\bot$. It turns out that we need a *commuting conversion*:

$$\dfrac{\overset{\mathcal{D}}{A\vee B\ true} \quad \begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ C\ true\end{array} \quad \begin{array}{c}\overline{B\ true}^{\,y}\\ \vdots\\ C\ true\end{array}}{\dfrac{C\ true}{C'\ true}\,R}\vee\mathsf{E}^{x,y} \qquad \Longrightarrow_C \qquad \dfrac{\overset{\mathcal{D}}{A\vee B\ true} \quad \begin{array}{c}\overline{A\ true}^{\,x}\\ \vdots\\ \dfrac{C\ true}{C'\ true}\,R\end{array} \quad \begin{array}{c}\overline{B\ true}^{\,y}\\ \vdots\\ \dfrac{C\ true}{C'\ true}\,R\end{array}}{C'\ true}\vee\mathsf{E}^{x,y}$$

Here the rule R is assumed to be an elimination rule, since there is no point in applying a commuting conversion when the rule R is an introduction rule. Note that a commuting conversion allows us to effectively ignore the elimination rule $\vee\mathsf{E}$ lying *between* the rule for proving $C\ true$ in the second or third premise and the rule R for proving $C'\ true$ from $C\ true$ in the conclusion. In a certain sense, the only role that the conclusion in the rule $\vee\mathsf{E}$ plays is to indicate that both hypotheses $\overline{A\ true}^{\,x}$ and $\overline{B\ true}^{\,y}$ lead to the same conclusion $C\ true$, instead of two different conclusions, say $C_1\ true$ and $C_2\ true$. In other words, $C\ true$ in the conclusion makes no contribution to the proof because it is the two premises that actually prove $C\ true$. Therefore the rule $\vee\mathsf{E}$ may be ignored as far as deducing another judgment from $C\ true$ in the conclusion is concerned. If we chose the following elimination rule for $\vee$ with a side condition that both hypotheses $\overline{A\ true}^{\,x}$ and $\overline{B\ true}^{\,y}$ lead to the same conclusion, no commuting conversion would be necessary:

$$\dfrac{\begin{array}{cc} A \vee B\ true & \\ \hline \overline{A\ true}^{\,x} & \overline{B\ true}^{\,y} \\ \vdots & \vdots \\ C\ true & C\ true \end{array}}{}\ \vee\mathsf{E}^{x,y}$$

Now applying a commuting conversion to the proof of $(A \vee A) \supset A\ true$ shown above yields another proof of the same judgment, to which a local reduction can be applied:

$$\cdots \qquad \Longrightarrow_C \qquad \dfrac{\dfrac{\overline{A \vee A\ true}^{\,z} \qquad \dfrac{\dfrac{\overline{A\ true}^{\,x} \quad \overline{A\ true}^{\,x}}{A \wedge A\ true}\ \wedge\mathsf{I} \qquad \dfrac{\dfrac{\overline{A\ true}^{\,y} \quad \overline{A\ true}^{\,y}}{A \wedge A\ true}\ \wedge\mathsf{I}}{A\ true}\ \wedge\mathsf{E_L}}{A\ true}\ \wedge\mathsf{E_L}}{A\ true}\ \vee\mathsf{E}^{x,y}}{(A \vee A) \supset A\ true}\ \supset\mathsf{I}^{z}$$

After removing the two detours in it, we obtain a normal proof annotated with arrows $\downarrow$ and $\uparrow$:

$$\dfrac{\dfrac{\overline{A \vee A\downarrow}^{\,z} \qquad \overline{A\downarrow\uparrow}^{\,x} \quad \overline{A\downarrow\uparrow}^{\,y}}{A\uparrow}\ \vee\mathsf{E}^{x,y}_{\downarrow}}{(A \vee A) \supset A\uparrow}\ \supset\mathsf{I}^{z}_{\uparrow}$$

1.8 Long normal proofs

While the normalization theorem guarantees the existence of a proof of $A\uparrow$ for every proof of $A\ true$, it does not address the uniqueness of proofs of $A\uparrow$. In fact, such a proof of $A\uparrow$ is not always unique! To see why, observe that the rule $\dfrac{A\downarrow}{A\uparrow}\ \downarrow\uparrow$ has no restriction on proposition A. Therefore, if a proof of $A\downarrow$ is given where A is not an atomic proposition (*e.g.*, $A = A_1 \supset A_2$), we may either appeal to the rule $\downarrow\uparrow$ to deduce $A\uparrow$ immediately, or apply an elimination rule to $A\downarrow$ to later build a proof of $A\uparrow$ by applying an introduction rule. For example, we may prove $(A \supset B) \supset (A \supset B)\uparrow$ by applying the rule $\downarrow\uparrow$ to $A \supset B\downarrow$:

$$\dfrac{\dfrac{\overline{A \supset B\downarrow}^{\,x}}{A \supset B\uparrow}\ \downarrow\uparrow}{(A \supset B) \supset (A \supset B)\uparrow}\ \supset\mathsf{I}^{x}_{\uparrow}$$

Alternatively we may prove the same judgment $(A \supset B) \supset (A \supset B)\uparrow$ by decomposing $A \supset B\downarrow$ until the rule $\downarrow\uparrow$ is applied to $B\downarrow$ for an atomic proposition B:

$$\dfrac{\dfrac{\dfrac{\overline{A \supset B\downarrow}^{\,x} \quad \dfrac{\overline{A\downarrow}^{\,y}}{A\uparrow}\downarrow\uparrow}{B\downarrow}\supset \mathsf{E}_{\downarrow}}{B\uparrow}\downarrow\uparrow}{A \supset B\uparrow}\supset \mathsf{I}^{y}_{\uparrow}}{(A \supset B) \supset (A \supset B)\uparrow}\supset \mathsf{I}^{x}_{\uparrow}$$

If we require that proposition A in the rule $\downarrow\uparrow$ be atomic, top-down applications of elimination rules meet bottom-up applications of introduction rules only through atomic propositions. Thus every normal proof applies elimination rules until only neutral judgments $A\downarrow$ for atomic propositions A remain, and starts to apply introduction rules only to normal judgments $A\uparrow$ for atomic propositions A. We call such normal proofs as *long normal proofs*. For example, the second proof of $(A \supset B) \supset (A \supset B)\uparrow$ shown above is a long normal proof while the first proof is not.

Now consider the system in Figure 1.3 in which proposition A in the rule $\downarrow\uparrow$ is required to be atomic:

$$\dfrac{A\downarrow}{A\uparrow}\downarrow\uparrow \textit{ (A atomic)}$$

If we can show that the original rule $\downarrow\uparrow$ (without the requirement on proposition A) is derivable, all the elimination rules in the system are strong enough in the sense that even if all propositions are decomposed into atomic propositions by elimination rules, no knowledge is essentially lost. (Here it helps to think of $A\uparrow$ and $A\downarrow$ as expressing a particular strategy for proving A *true*.) As a property of *all* elimination rules collectively, it is called the *global completeness* property. (Recall that the local completeness property states that a *specific* elimination rule is strong enough.)

The system in Figure 1.3 satisfies the global completeness property. We inductively show that the original rule $\downarrow\uparrow$ without the requirement on proposition A is derivable.

Proposition 1.14. *The rule* $\dfrac{A\downarrow}{A\uparrow}$ *is derivable.*

Proof. By induction on the structure of proposition A. If A is atomic, we apply the new rule $\downarrow\uparrow$ (with the requirement on proposition A). We show the case $A = A_1 \supset A_2$:

$$\dfrac{\dfrac{\dfrac{A_1 \supset A_2\downarrow \quad \dfrac{\overline{A_1\downarrow}^{\,x}}{A_1\uparrow}\textit{ IH on } A_1}{A_2\downarrow}\supset \mathsf{E}_{\downarrow}}{A_2\uparrow}\textit{ IH on } A_2}{A_1 \supset A_2\uparrow}\supset \mathsf{I}^{x}_{\uparrow}$$

□

Chapter 2

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.

2.1 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 2.1 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 $A \wedge B\ true$ is constructed from a proof $\mathcal{D}_A$ of $A\ true$ and a proof $\mathcal{D}_B$ of $B\ true$. 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):

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

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:

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

Implication

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

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

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

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

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

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

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 4 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 $\bot$.

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

$$\frac{\begin{array}{c}\overline{x : A} \\ \vdots \\ M : B\end{array}}{\lambda x{:}A.\, M : A \supset B}\ \supset\mathsf{I} \qquad \frac{M : A \supset B \quad N : A}{M\ N : B}\ \supset\mathsf{E}$$

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

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

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

Figure 2.1: 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 2.2: Typing rules using hypothetical judgments

2.2 Type system

Since hypothetical judgments are just a syntactic tool for displaying hypothetical proofs, it is straightforward to extend the translation in Section 2.1 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 2.2 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 2.1 (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 2.1 may proceed by induction on the structure of proof term M. This is because the type system in Figure 2.2 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 2.2 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 2.2 (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 $[M/x]N_1\ [M/x]N_2 = [M/x](N_1\ N_2)$

□

2.3 β-reductions and η-expansions

We have seen in Section 1.5 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:

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

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 2.3 (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 2.2 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 2.2 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 2.3: Rules for the structural congruence relation $\Longrightarrow$

Figure 2.3 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 2.3 now extends to the *subterm subject reduction* property:

Theorem 2.4 (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 2.3. □

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:

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

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

2.4 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 1.3 or Figure 1.4) 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 1.12 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 2.5 (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 2.6 (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 2.7 (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.

Chapter 3

Sequent Calculus

This chapter presents a *sequent calculus* for propositional logic. Although we set out to develop it as a device for proving the completeness property of normal proofs (Theorem 1.11), the sequent calculus also serves as a basis for proof search strategies implemented in theorem provers. Due to its important role in logic, a sequent calculus is not viewed as a secondary system derivable from a corresponding natural deduction system. Rather it is accepted as a valid formulation of a system of logic in itself, whether a corresponding natural deduction system has been formulated or not.

3.1 Sequent calculus for propositional logic

The sequent calculus for propositional logic consists of inference rules for *sequents* of the form $\Gamma \longrightarrow C$ where Γ is an unordered collection of propositions. Conceptually a sequent $A_1, \cdots, A_n \longrightarrow C$ becomes evident by a proof of $C\uparrow$ using neutral judgments $A_1\downarrow, \cdots, A_n\downarrow$:

$$A_1, \cdots, A_n \longrightarrow C \quad \Longleftrightarrow \quad \begin{array}{c} A_1\downarrow \quad \cdots \quad A_n\downarrow \\ \ddots \\ C\uparrow \end{array}$$

Note that the exchange rule is built into sequents because Γ in a sequent $\Gamma \longrightarrow C$ is an unordered collection of propositions.

It is important that unlike a hypothetical judgment $\Gamma \vdash J$ in which a judgment $J \in \Gamma$ is interpreted as a *hypothesis* $\overline{J}$, a proposition $A \in \Gamma$ in a sequent $\Gamma \longrightarrow C$ denotes just a *neutral judgment* $A\downarrow$, which may happen to originate from a hypothesis $\overline{A\downarrow}$, but not necessarily. For example, both proofs of $C\uparrow$ shown below make evident the same sequent $A \wedge B, A \longrightarrow C$:

$$\begin{array}{c} \dfrac{\overline{A \wedge B\downarrow}}{A\downarrow}\,\wedge\mathsf{E}_{\mathsf{L}\downarrow} \\ \vdots \\ C\uparrow \end{array} \qquad\qquad \begin{array}{c} \overline{A \wedge B\downarrow} \qquad\quad \overline{A\downarrow} \\ \ddots \\ C\uparrow \end{array}$$

In the left proof, $A \wedge B\downarrow$, as well as $A\downarrow$, is available as a neutral judgment because the same hypothesis may be used more than once. In the right proof, both neutral judgments $A \wedge B\downarrow$ and $A\downarrow$ happen to originate from hypotheses. Still, however, we may think of $A \in \Gamma$ in a sequent $\Gamma \longrightarrow C$ as denoting a hypothesis $\overline{A\downarrow}$ available in the proof of $C\uparrow$, since as far as the proof of $C\uparrow$ is concerned, using $A\downarrow$ as a neutral judgment or as a hypothesis makes no difference.

An advantage of the sequent calculus over the natural deduction system consists in the fact that a proof of $\Gamma \longrightarrow C$ always proceeds in a bottom-up way, which implies that every inference rule in the sequent calculus is best read in a bottom-up way. (This is not the case in the natural deduction system because every elimination rule is best read in a top-down way.) Consider a sequent $A_1, \cdots, A_i, \cdots, A_n \longrightarrow C$:

$$A_1, \cdots, A_i, \cdots, A_n \longrightarrow C \quad \Longleftrightarrow \quad \begin{array}{ccccc} A_1\downarrow & \cdots & A_i\downarrow & \cdots & A_n\downarrow \\ & \ddots & & \iddots & \\ & & C\uparrow & & \end{array}$$

For the sake of simplicity, let us assume that an introduction rule applied to $C\uparrow$ produces a new goal $C'\uparrow$ without producing a new hypothesis (as is the case for the rule $\vee\mathsf{I}_{\mathsf{L}\uparrow}$), and that an elimination rule applied to $A_i\downarrow$ produces a new neutral judgment $A_i'\downarrow$ without requiring a separate proof of a normal judgment (as is the case for the rule $\wedge\mathsf{E}_{\mathsf{L}\downarrow}$). If we choose to apply an introduction rule to $C\uparrow$, a new goal $C'\uparrow$ is produced. Thus we now have to prove $A_1, \cdots, A_i, \cdots, A_n \longrightarrow C'$:

$$\frac{A_1, \cdots, A_i, \cdots, A_n \longrightarrow C'}{A_1, \cdots, A_i, \cdots, A_n \longrightarrow C} \quad \Longleftrightarrow \quad \begin{array}{ccccc} A_1\downarrow & \cdots & A_i\downarrow & \cdots & A_n\downarrow \\ & \ddots & & \iddots & \\ & & \dfrac{C'\uparrow}{C\uparrow} & & \end{array}$$

Such an inference rule in the sequent calculus is called a *right rule* because it focuses on the right side C in a sequent $\Gamma \longrightarrow C$. A right rule then corresponds to an introduction rule in the natural deduction system. If we choose to apply an elimination rule to $A_i\downarrow$, a new neutral judgment $A_i'\downarrow$ is produced while the goal $C\uparrow$ remains the same. Thus we now have to prove $A_1, \cdots, A_i, A_i', \cdots, A_n \longrightarrow C$:

$$\frac{A_1, \cdots, A_i, A_i', \cdots, A_n \longrightarrow C}{A_1, \cdots, A_i, \cdots, A_n \longrightarrow C} \quad \Longleftrightarrow \quad \begin{array}{ccccc} A_1\downarrow & \cdots & \dfrac{A_i\downarrow}{A_i'\downarrow} & \cdots & A_n\downarrow \\ & \ddots & & \iddots & \\ & & C\uparrow & & \end{array}$$

Such an inference rule in the sequent calculus is called a *left rule* because it focuses on a proposition in the left side Γ in a sequent $\Gamma \longrightarrow C$. A left rule then corresponds to an elimination rule in the natural deduction system.

Keeping in mind the intuition behind sequents, let us consider each inference rule in the sequent calculus. The first rule is an axiom which deals with *initial sequents* of the form $\Gamma, A \longrightarrow A$:

$$\frac{}{\Gamma, A \longrightarrow A}\ \mathit{Init}$$

Note that the rule *Init* corresponds not to the rule Hyp but to the rule $\downarrow\uparrow$ in the natural deduction system: it is *not* a rule using a hypothesis; rather it is a rule deducing $A\uparrow$ from $A\downarrow$.

For conjunction $\wedge$, we need two left rules $\wedge L_L$ and $\wedge L_R$, corresponding to the elimination rules $\wedge\mathsf{E}_{\mathsf{L}\downarrow}$ and $\wedge\mathsf{E}_{\mathsf{R}\downarrow}$, and one right rule $\wedge R$, corresponding to the introduction rule $\wedge\mathsf{I}_{\uparrow}$:

$$\frac{\Gamma, A \wedge B, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\ \wedge L_L \qquad \frac{\Gamma, A \wedge B, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\ \wedge L_R \qquad \frac{\Gamma \longrightarrow A \quad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\ \wedge R$$

For implication $\supset$, we need one left rule, corresponding to the elimination rule $\supset\mathsf{E}_{\downarrow}$, and one right rule, corresponding to the introduction rule $\supset\mathsf{I}_{\uparrow}$. Suppose that we wish to prove $\Gamma, A \supset B \longrightarrow C$ by focusing on $A \supset B$ in the left side:

$$\Gamma, A \supset B \longrightarrow C \quad \Longleftrightarrow \quad \begin{array}{ccc} \Gamma\downarrow & \cdots & A \supset B\downarrow \\ & \ddots\ \iddots & \\ & C\uparrow & \end{array}$$

Here $\Gamma_\downarrow$ is a shorthand for $\{A\downarrow \mid A \in \Gamma\}$. In order to apply the rule $\supset\mathsf{E}_\downarrow$ to $A \supset B\downarrow$, we first have to build a proof of $A\uparrow$ using $\Gamma_\downarrow$ and $A \supset B\downarrow$, which means that we need a proof of $\Gamma, A \supset B \longrightarrow A$:

$$\frac{\Gamma, A \supset B \longrightarrow A \quad \cdots}{\Gamma, A \supset B \longrightarrow C} \qquad \Longleftrightarrow \qquad \begin{array}{c} \Gamma_\downarrow \quad \cdots \quad A \supset B\downarrow \\ \vdots \\ C\uparrow \end{array} \qquad \begin{array}{c} \Gamma_\downarrow \quad \cdots \quad A \supset B\downarrow \\ \vdots \\ A\uparrow \end{array}$$

Then a new neutral judgment $B\downarrow$ becomes available for the proof of $C\uparrow$, which means that it now suffices to prove $\Gamma, A \supset B, B \longrightarrow C$:

$$\frac{\Gamma, A \supset B \longrightarrow A \quad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C} \qquad \Longleftrightarrow \qquad \begin{array}{c} \Gamma_\downarrow \quad \cdots \\ \vdots \\ C\uparrow \end{array} \qquad \frac{\begin{array}{c} \Gamma_\downarrow \quad \cdots \quad A \supset B\downarrow \\ \vdots \\ A\uparrow \end{array} \qquad A \supset B\downarrow}{B\downarrow}\ \supset\mathsf{E}_\downarrow$$

Thus we obtain the following left rule $\supset L$; the right rule $\supset R$ is obtained by a similar analysis:

$$\frac{\Gamma, A \supset B \longrightarrow A \quad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\ \supset L \qquad \frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\ \supset R$$

For disjunction $\vee$, we need one left rule $\vee L$, corresponding to the elimination rule $\vee\mathsf{E}_\downarrow$, and two right rules $\vee R_L$ and $\vee R_R$, corresponding to the introduction rules $\vee\mathsf{I}_{\mathsf{L}\uparrow}$ and $\vee\mathsf{I}_{\mathsf{R}\uparrow}$; the rule $\vee L$ is obtained in a similar way to the rule $\supset L$:

$$\frac{\Gamma, A \vee B, A \longrightarrow C \quad \Gamma, A \vee B, B \longrightarrow C}{\Gamma, A \vee B \longrightarrow C}\ \vee L \qquad \frac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}\ \vee R_L \qquad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}\ \vee R_R$$

The rule $\vee L$ is designed in such a way that commuting conversion is built into the sequent calculus. To see why, observe that $\Gamma, A \vee B \longrightarrow C$ in the conclusion describes a proof of the goal $C\uparrow$ in which the rule $\vee\mathsf{E}$ is to be applied to $A \vee B\downarrow$. Then $\Gamma, A \vee B, A \longrightarrow C$ and $\Gamma, A \vee B, B \longrightarrow C$ in the premises indicate that the rule $\vee\mathsf{E}$ applied to $A \vee B\downarrow$ uses the same goal $C\uparrow$ in its conclusion. According to the rule $\vee L$, therefore, the conclusion in any instance of the rule $\vee\mathsf{E}$ in the natural deduction system is always the current goal, which means that commuting conversion is built into the sequent calculus.

For truth $\top$, we need a right rule $\top R$ corresponding to the introduction rule $\top\mathsf{I}$, but no left rule (because there is no elimination rule for $\top$); for falsehood $\bot$, we need a left rule $\bot L$ corresponding to the elimination rule $\bot\mathsf{E}$, but no right rule (because there is no introduction rule for $\bot$):

$$\frac{}{\Gamma \longrightarrow \top}\ \top R \qquad \frac{}{\Gamma, \bot \longrightarrow C}\ \bot L$$

Figure 3.1 shows all the inference rules in the sequent calculus for propositional logic. Note that each rule R focuses on a proposition in the sequent of the conclusion, which appears in the left side if R is a left rule, in the right side if R is a right rule, and in both sides if R is the rule *Init*. For example, the rule $\wedge L_L$ focuses on $A \wedge B$ in the left side, the rule $\wedge R$ on $A \wedge B$ in the right side, and the rule *Init* on A. We refer to such a proposition as the *principal formula* of the rule.

The rules $\neg L$ and $\neg R$ are obtained from the notational definition $\neg A = A \supset \bot$. In particular, the rule $\neg L$ implicitly uses the rule $\bot L$:

$$\frac{\Gamma, A \supset \bot \longrightarrow A \quad \dfrac{}{\Gamma, A \supset \bot, \bot \longrightarrow C}\ \bot L}{\Gamma, A \supset \bot \longrightarrow C}\ \supset L$$

$$\frac{}{\Gamma, A \longrightarrow A}\ \mathit{Init} \quad \frac{\Gamma, A \wedge B, A \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\ \wedge L_L \quad \frac{\Gamma, A \wedge B, B \longrightarrow C}{\Gamma, A \wedge B \longrightarrow C}\ \wedge L_R \quad \frac{\Gamma \longrightarrow A \quad \Gamma \longrightarrow B}{\Gamma \longrightarrow A \wedge B}\ \wedge R$$

$$\frac{\Gamma, A \supset B \longrightarrow A \quad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\ \supset L \quad \frac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\ \supset R$$

$$\frac{\Gamma, A \vee B, A \longrightarrow C \quad \Gamma, A \vee B, B \longrightarrow C}{\Gamma, A \vee B \longrightarrow C}\ \vee L \quad \frac{\Gamma \longrightarrow A}{\Gamma \longrightarrow A \vee B}\ \vee R_L \quad \frac{\Gamma \longrightarrow B}{\Gamma \longrightarrow A \vee B}\ \vee R_R$$

$$\frac{}{\Gamma \longrightarrow \top}\ \top R \quad \frac{}{\Gamma, \bot \longrightarrow C}\ \bot L \quad \frac{\Gamma, \neg A \longrightarrow A}{\Gamma, \neg A \longrightarrow C}\ \neg L \quad \frac{\Gamma, A \longrightarrow \bot}{\Gamma \longrightarrow \neg A}\ \neg R$$

Figure 3.1: Sequent calculus for propositional logic

As in the natural deduction system, the weakening and contraction properties allow us to use a proposition $A \in \Gamma$ zero times and more than once, respectively, in a proof of $\Gamma \longrightarrow C$. Note that the structural properties allow us to identify two sequents $\Gamma \longrightarrow C$ and $\Gamma' \longrightarrow C$ if Γ and Γ' are equivalent as sets (rather than as multisets), *i.e.*, $\{A \mid A \in \Gamma\} = \{A \mid A \in \Gamma'\}$.

Proposition 3.1 (Structural properties).
$\Gamma_\downarrow = \{A\downarrow \mid A \in \Gamma\}$ *is a collection of neutral judgments.*
(Weakening) *If* $\Gamma \longrightarrow C$, *then* $\Gamma, A \longrightarrow C$.
(Contraction) *If* $\Gamma, A, A \longrightarrow C$, *then* $\Gamma, A \longrightarrow C$.

Proof. By induction on the structure of the proof of $\Gamma \longrightarrow C$ and $\Gamma, A, A \longrightarrow C$. □

The sequent calculus in Figure 3.1 satisfies the *subformula property* that every proposition (or formula) in the premise of a rule is a subformula of a certain proposition (or formula) in the conclusion, where the subformula relation is defined as follows: (1) A is a subformula of A; (2) A and B are subformulae of $A \supset B$, $A \wedge B$, and $A \vee B$. For example, the premise of the rule $\wedge L_L$ introduces a new proposition A, which is a subformula of $A \wedge B$ in the conclusion; the premises of the rule $\wedge R$ introduce two new propositions A and B, both of which are subformulae of $A \wedge B$ in the conclusion.

Because of the subformula property, a proof of $\Gamma \longrightarrow C$ needs to consider only subformulae of those propositions in Γ or C. For example, a proof of $\cdot \longrightarrow A \supset (B \supset C)$ never involves an analysis of $A \supset B$ by applying the rule $\supset L$ or $\supset R$ because it is not a subformula of $A \supset (B \supset C)$. In conjunction with the structural properties, therefore, the subformula property implies that the sequent calculus in Figure 3.1 is decidable: there exists a procedure for deciding whether $\Gamma \longrightarrow C$ is provable or not. Intuitively a proof of $\Gamma \longrightarrow C$ generates a finite number of sequents because only a finite number of propositions need to be considered.

Proposition 3.2. *The sequent calculus in Figure 3.1 is decidable.*

Proof. Let us write $\Gamma^\star$ for a set $\{A \mid A \in \Gamma\}$ consisting of elements of a multiset Γ. By the structural properties, $\Gamma \longrightarrow C$ is provable if and only if $\Gamma^\star \longrightarrow C$ is provable. When proving a sequent, therefore, we implicitly use only those sequents of the form $\Gamma^\star \longrightarrow C$ in which no proposition appears more than once in $\Gamma^\star$.

Suppose that we wish to check the provability of the goal sequent $\Gamma \longrightarrow C$. First we generate the set S of all possible sequents using subformulae of propositions in Γ and C. S must be a finite set because of the subformula property. (If Γ and C produce n different subformulae, there are a total of $2^n \times n$ sequents in S.) Next we check each sequent in S and mark it as "proven" if it is provable by the rule Init, $\top R$, or $\bot L$, which are the rules with no premise. Then, for each rule except Init, $\top R$, or $\bot L$, we consider all possible combinations of those sequents marked as "proven" for its premise, and mark as "proven" the sequent corresponding to the conclusion if it is not marked as "proven" yet. For example, for the rule $\wedge L_L$, we mark every sequent of the form $\Gamma, A \wedge B \longrightarrow C$ as "proven" if $\Gamma, A \wedge B, A \longrightarrow C$ is already marked as "proven." Similarly, for the rule $\supset L$, we mark every sequent of the form $\Gamma, A \supset B \longrightarrow C$ as "proven" if $\Gamma, A \supset B \longrightarrow A$

and $\Gamma, A \supset B, B \longrightarrow C$ are already marked as "proven." We repeat the procedure until no more sequent in S can be marked as "proven." The procedure must eventually terminate because the number of combinations of the rules and the sequents in S is finite. If the goal sequent is marked as "proven," we decide that it is provable; otherwise it is not provable. (The procedure described above is the basis for a practical proof search technique called the *inverse method*.) □

The soundness and completeness properties of the sequent calculus show that it is equivalent to the natural deduction system for normal judgments.

Theorem 3.3 (Soundness of the sequent calculus). *If* $\Gamma \longrightarrow C$, *then* $\Gamma_\downarrow \vdash C\uparrow$.

Theorem 3.4 (Completeness of the sequent calculus). *If* $\Gamma_\downarrow \vdash C\uparrow$, *then* $\Gamma \longrightarrow C$.

Note that while $A\downarrow \in \Gamma_\downarrow$ in a hypothetical judgment $\Gamma_\downarrow \vdash C\uparrow$ denotes a hypothesis $\overline{A\downarrow}$, $A \in \Gamma$ in a sequent $\Gamma \longrightarrow C$ denotes just a neutral judgment $A\downarrow$, which is not necessarily a hypothesis. The discrepancy does not invalidate the two theorems, however, since as far as the proof of $C\uparrow$ is concerned, using $A\downarrow$ as a hypothesis when it is a neutral judgment, or vice versa, makes no difference.

The proof of the soundness property is straightforward:

Proof of Theorem 3.3. By induction on the structure of the proof of $\Gamma \longrightarrow C$. Below we show three representative cases. We reuse metavariables Γ and C.

Case $\dfrac{}{\Gamma, A \longrightarrow A}\ \mathit{Init}$

$\Gamma_\downarrow, A\downarrow \vdash A\downarrow$ by the rule $\mathsf{Hyp}_\downarrow$

$\Gamma_\downarrow, A\downarrow \vdash A\uparrow$ by the rule $\downarrow\uparrow$

Case $\dfrac{\Gamma, A \supset B \longrightarrow A \quad \Gamma, A \supset B, B \longrightarrow C}{\Gamma, A \supset B \longrightarrow C}\ \supset L$

$\Gamma_\downarrow, A \supset B\downarrow \vdash A \supset B\downarrow$ by the rule $\mathsf{Hyp}_\downarrow$

$\Gamma_\downarrow, A \supset B\downarrow \vdash A\uparrow$ by induction hypothesis on $\Gamma, A \supset B \longrightarrow A$

$\Gamma_\downarrow, A \supset B\downarrow \vdash B\downarrow$ by the rule $\supset\mathsf{E}_\downarrow$

$\Gamma_\downarrow, A \supset B\downarrow, B\downarrow \vdash C\uparrow$ by induction hypothesis on $\Gamma, A \supset B, B \longrightarrow C$

$\Gamma_\downarrow, A \supset B\downarrow \vdash C\uparrow$ by the substitution principle (Theorem 1.10)

Case $\dfrac{\Gamma, A \longrightarrow B}{\Gamma \longrightarrow A \supset B}\ \supset R$

$\Gamma_\downarrow, A\downarrow \vdash B\uparrow$ by induction hypothesis on $\Gamma, A \longrightarrow B$

$\Gamma_\downarrow \vdash A \supset B\uparrow$ by the rule $\supset\mathsf{I}_\uparrow$

□

On the other hand, the proof of the completeness property is not so straightforward. In fact, it is easy to see that a direct proof attempt fails because inference rules for normal judgments use neutral judgments as well, but the theorem does not mention neutral judgments at all. For example, if the proof of $\Gamma_\downarrow \vdash C\uparrow$ ends with an application of the rule $\downarrow\uparrow$, the premise is $\Gamma_\downarrow \vdash C\downarrow$, to which induction hypothesis cannot be applied. Therefore we need to generalize the theorem so that a hypothetical judgment $\Gamma_\downarrow \vdash A\downarrow$ is also related to sequents in a certain way.

Lemma 3.5 below generalizes Theorem 3.4. The proof itself is straightforward, but formulating the statement connecting $\Gamma_\downarrow \vdash A\downarrow$ to sequents is far from trivial. Theorem 3.4 follows as an immediate consequence of Lemma 3.5.

Lemma 3.5.
If $\Gamma_\downarrow \vdash A\downarrow$, *then* $\Gamma, A \longrightarrow C$ *implies* $\Gamma \longrightarrow C$.
If $\Gamma_\downarrow \vdash C\uparrow$, *then* $\Gamma \longrightarrow C$.

Proof. By simultaneous induction on the structure of the proof of $\Gamma_\downarrow \vdash A\downarrow$ and $\Gamma_\downarrow \vdash C\uparrow$. Below we reuse metavariables Γ, A, and C.

Case $\dfrac{}{\Gamma_\downarrow, A\downarrow \vdash A\downarrow}\ \mathsf{Hyp}_\downarrow$

$\Gamma, A, A \longrightarrow C$ assumption

$\Gamma, A \longrightarrow C$ by contraction

Case $\dfrac{\Gamma_\downarrow, A\downarrow \vdash B\uparrow}{\Gamma_\downarrow \vdash A \supset B\uparrow}\ \supset\mathsf{I}_\uparrow$

$\Gamma, A \longrightarrow B$ by induction hypothesis on $\Gamma_\downarrow, A\downarrow \vdash B\uparrow$

$\Gamma \longrightarrow A \supset B$ by the rule $\supset R$

Case $\dfrac{\Gamma_\downarrow \vdash A \supset B\downarrow \quad \Gamma_\downarrow \vdash A\uparrow}{\Gamma_\downarrow \vdash B\downarrow}\ \supset\mathsf{E}_\downarrow$

$\Gamma, B \longrightarrow C$ assumption

$\Gamma, A \supset B, B \longrightarrow C$ by weakening

$\Gamma \longrightarrow A$ by induction hypothesis on $\Gamma_\downarrow \vdash A\uparrow$

$\Gamma, A \supset B \longrightarrow A$ by weakening

$\Gamma, A \supset B \longrightarrow C$ by the rule $\supset L$ on $\Gamma, A \supset B \longrightarrow A$ and $\Gamma, A \supset B, B \longrightarrow C$

$\Gamma \longrightarrow C$ by induction hypothesis on $\Gamma_\downarrow \vdash A \supset B\downarrow$ with $\Gamma, A \supset B \longrightarrow C$

Case $\dfrac{\Gamma_\downarrow \vdash A\downarrow}{\Gamma_\downarrow \vdash A\uparrow}\ \downarrow\uparrow$

$\Gamma, A \longrightarrow A$ by the rule *Init*

$\Gamma \longrightarrow A$ by induction hypothesis on $\Gamma_\downarrow \vdash A\downarrow$ with $\Gamma, A \longrightarrow A$

□

The proof of Lemma 3.5 is *constructive*, as opposed to *declarative*, in the sense that it gives an algorithm for converting a proof of $\Gamma_\downarrow \vdash C\uparrow$ into a proof of $\Gamma \longrightarrow C$. The key to understanding its constructive nature is to observe that when converting $\Gamma_\downarrow \vdash A\downarrow$, a proof of $\Gamma, A \longrightarrow C$ for some proposition C is given as an assumption so that a new proof of $\Gamma \longrightarrow C$ is produced. For example, the case $\dfrac{\Gamma_\downarrow \vdash A\downarrow}{\Gamma_\downarrow \vdash A\uparrow}\ \downarrow\uparrow$ creates a proof of $\Gamma, A \longrightarrow A$ using the rule *Init*, which then serves as an assumption in converting the premise $\Gamma_\downarrow \vdash A\downarrow$. The case $\dfrac{}{\Gamma_\downarrow, A\downarrow \vdash A\downarrow}\ \mathsf{Hyp}_\downarrow$ uses the contraction property to deduce $\Gamma, A \longrightarrow C$ from such an assumption $\Gamma, A, A \longrightarrow C$.

3.2 Cut elimination

We have seen that in the natural deduction system based on hypothetical judgments, reflexivity and the substitution principle confirm that the system adheres to the definition of hypothetical judgments. That is, failure of reflexivity or the substitution principle indicates the existence of an inference rule that does not respect the definition of hypothetical judgments as concise representations of hypothetical proofs.

In the case of the sequent calculus, we may test its integrity by checking two similar principles, especially in view of the fact that $A \in \Gamma$ in a sequent $\Gamma \longrightarrow C$ can be thought of as denoting a hypothesis $\overline{A\downarrow}$. The first, corresponding to reflexivity in the natural deduction system, is the provability of every initial sequent $\Gamma, A \longrightarrow A$, which directly follows from the rule *Init*. The second, corresponding to the substitution principle, is the *admissibility of the cut rule* (where the cut rule is another rule to be explained later):

Theorem 3.6 (Admissiblity of the cut rule). *If* $\Gamma \longrightarrow A$ *and* $\Gamma, A \longrightarrow C$, *then* $\Gamma \longrightarrow C$.

Thus the admissibility of the cut rule is to the sequent calculus what the substitution principle is to the natural deduction system: if the substitution principle fails, it indicates that the natural deduction system is not sound (or even non-sense); similarly if the admissibility of the cut rule fails, it indicates that the sequent calculus is not sound (or even non-sense).

Theorem 3.6 implies that if a new rule $\dfrac{A\uparrow}{A\downarrow}\ \uparrow\downarrow$ is added to the natural deduction system for normal and neutral judgments, we can safely remove any occurrence of the rule $\uparrow\downarrow$ in a proof of $C\uparrow$. The intuition is that Theorem 3.6 may be rewritten in terms of normal and neutral judgments as follows:

$$\textit{If}\ \begin{array}{c}\Gamma_\downarrow\\ \vdots\\ \dfrac{A\uparrow}{A\downarrow}\uparrow\downarrow\\ \vdots\\ C\uparrow\end{array}\ \textit{, then}\ \begin{array}{c}\Gamma_\downarrow\\ \vdots\\ C\uparrow\end{array}.$$

Since an occurrence of the rule $\uparrow\downarrow$ in a proof of $C\uparrow$ corresponds to a detour in a proof of $C\ true$, Theorem 3.6 implies in turn that we can transform the *entire* proof of $C\ true$ so as to remove any detour in it. In this sense, Theorem 3.6 states that the natural deduction system for truth judgments is *globally sound*. (Recall that the local soundness property states that a detour specific to a connective can be locally eliminated, without transforming the entire proof.) We will later formalize the global soundness property as the normalization theorem (Theorem 3.8).

The proof of Theorem 3.6 proceeds by nested induction on the structure of: 1) proposition A which is called the *cut formula*; 2) proof of $\Gamma \longrightarrow A$; 3) proof of $\Gamma, A \longrightarrow C$. Here are a few examples of applying induction hypothesis in the proof of Theorem 3.6:

- We wish to prove that $\Gamma \longrightarrow A \supset B$ and $\Gamma, A \supset B \longrightarrow C$ imply $\Gamma \longrightarrow C$. Since A is a subformula of the cut formula $A \supset B$, the induction hypothesis on A proves that $\Gamma' \longrightarrow A$ and $\Gamma', A \longrightarrow C'$ imply $\Gamma' \longrightarrow C'$ for any Γ' and C', and also regardless of the structure of the proof of $\Gamma' \longrightarrow A$ and $\Gamma', A \longrightarrow C'$.

- We wish to prove that $\Gamma \longrightarrow A$ and $\Gamma, A \longrightarrow C$ imply $\Gamma \longrightarrow C$. Suppose that the proof of $\Gamma \longrightarrow A$ has the following structure:

  $$\dfrac{\cdots \quad \Gamma, B \overset{\mathcal{D}}{\longrightarrow} A}{\Gamma \longrightarrow A}\ R$$

  Then we weaken $\Gamma, A \longrightarrow C$ to obtain a proof $\mathcal{E}$ of $\Gamma, B, A \longrightarrow C$. Since $\mathcal{D}$ is strictly smaller than the proof of $\Gamma \longrightarrow A$, the induction hypothesis on proposition A, proof $\mathcal{D}$, and proof $\mathcal{E}$ yields $\Gamma, B \longrightarrow C$, irrespective of the structure (or size) of $\mathcal{E}$.

- We wish to prove that $\Gamma \longrightarrow A$ and $\Gamma, A \longrightarrow C$ imply $\Gamma \longrightarrow C$. Suppose that the proof of $\Gamma, A \longrightarrow C$ has the following structure:

  $$\dfrac{\cdots \quad \Gamma, B, A \overset{\mathcal{E}}{\longrightarrow} C'}{\Gamma, A \longrightarrow C}\ R$$

  Then we weaken $\Gamma \longrightarrow A$ to obtain a proof $\mathcal{D}$ of $\Gamma, B \longrightarrow A$, which has exactly the same structure (or size) as the proof of $\Gamma \longrightarrow A$. Since $\mathcal{E}$ is strictly smaller than the proof of $\Gamma, A \longrightarrow C$, the induction hypothesis on proposition A, proof $\mathcal{D}$, and proof $\mathcal{E}$ yields $\Gamma, B \longrightarrow C'$. (Then we typically apply the same rule R to deduce $\Gamma \longrightarrow C$.)

The proof of Theorem 3.6 considers all possible combinations of the last inference rule $R_\mathcal{D}$ in the proof $\mathcal{D}$ of $\Gamma \longrightarrow A$ and the last inference rule $R_\mathcal{E}$ in the proof $\mathcal{E}$ of $\Gamma, A \longrightarrow C$. The combinations of the rules $R_\mathcal{D}$ and $R_\mathcal{E}$ are divided as follows:

1. At least one of $R_\mathcal{D}$ and $R_\mathcal{E}$ is the rule Init.

   (a) $R_\mathcal{D}$ is the rule Init. In this case, we have $\Gamma = \Gamma', A$.

(b) $R_{\mathcal{E}}$ is the rule $Init$. In this case, we have either $\Gamma = \Gamma', C$ or $A = C$.

2. Neither of $R_{\mathcal{D}}$ and $R_{\mathcal{E}}$ is the rule $Init$.

   (a) A is the principal formula of both $R_{\mathcal{D}}$ and $R_{\mathcal{E}}$. In this case, $R_{\mathcal{D}}$ is a right rule and $R_{\mathcal{E}}$ is a left rule.

   (b) A is not the principal formula of $R_{\mathcal{D}}$. In this case, $R_{\mathcal{D}}$ is a left rule.

   (c) A is not the principal formula of $R_{\mathcal{E}}$. In this case, $R_{\mathcal{E}}$ can be both a left rule and a right rule.

Note that 1-(a) and 1-(b) overlap because both $R_{\mathcal{D}}$ and $R_{\mathcal{E}}$ can be the rule $Init$, and that 2-(b) and 2-(c) overlap because A may be the principal formula of neither $R_{\mathcal{D}}$ nor $R_{\mathcal{E}}$.

The proof of Theorem 3.6 is constructive because it gives an algorithm for building a proof of $\Gamma \longrightarrow C$ out of proofs of $\Gamma \longrightarrow A$ and $\Gamma, A \longrightarrow C$. The algorithm is non-deterministic because of the overlapping cases 1-(a) and 1-(b), and 2-(b) and 2-(c).

Proof of Theorem 3.6. By nested induction on the structure of: 1) cut formula A; 2) proof of $\Gamma \longrightarrow A$; 3) proof of $\Gamma, A \longrightarrow C$. Here we consider the fragment of the sequent calculus with the rules $Init$, $\supset L$, and $\supset R$ only.

We write $\mathcal{D} :: J$ or $\overset{\mathcal{D}}{J}$ to say that $\mathcal{D}$ is a proof of J. Let $R_{\mathcal{D}}$ be the last inference rule in the proof $\mathcal{D}$ of $\Gamma \longrightarrow A$ and $R_{\mathcal{E}}$ the last inference rule in the proof $\mathcal{E}$ of $\Gamma, A \longrightarrow C$.

Case 1-(a): $R_{\mathcal{D}}$ is the rule $Init$. We have $\Gamma = \Gamma', A$.

$$\mathcal{D} = \frac{}{\Gamma', A \longrightarrow A}\ Init$$

$\Gamma', A, A \longrightarrow C$	assumption
$\Gamma', A \longrightarrow C$	by contraction
$\Gamma \longrightarrow C$	from $\Gamma = \Gamma', A$

Case 1-(b): $R_{\mathcal{E}}$ is the rule $Init$.

Subcase: $\Gamma = \Gamma', C$

$$\mathcal{E} = \frac{}{\Gamma', C, A \longrightarrow C}\ Init$$

$\Gamma', C \longrightarrow C$	by the rule $Init$
$\Gamma \longrightarrow C$	from $\Gamma = \Gamma', C$

Subcase: $A = C$

$\Gamma \longrightarrow A$	assumption
$\Gamma \longrightarrow C$	from $A = C$

Case 2-(a): A is the principal formula of both $R_{\mathcal{D}}$ and $R_{\mathcal{E}}$. We have $A = A_1 \supset A_2$.

$$\mathcal{D} = \frac{\overset{\mathcal{D}_1}{\Gamma, A_1 \longrightarrow A_2}}{\Gamma \longrightarrow A_1 \supset A_2}\ \supset R \qquad \mathcal{E} = \frac{\overset{\mathcal{E}_1}{\Gamma, A_1 \supset A_2 \longrightarrow A_1} \quad \overset{\mathcal{E}_2}{\Gamma, A_1 \supset A_2, A_2 \longrightarrow C}}{\Gamma, A_1 \supset A_2 \longrightarrow C}\ \supset L$$

$\mathcal{E}'_1 :: \Gamma \longrightarrow A_1$	by IH on $A_1 \supset A_2$, $\mathcal{D}$, and $\mathcal{E}_1$
$\mathcal{D}' :: \Gamma, A_2 \longrightarrow A_1 \supset A_2$	by weakening $\mathcal{D} :: \Gamma \longrightarrow A_1 \supset A_2$
$\mathcal{E}'_2 :: \Gamma, A_2 \longrightarrow C$	by IH on $A_1 \supset A_2$, $\mathcal{D}'$, and $\mathcal{E}_2$
$\mathcal{D}'_1 :: \Gamma \longrightarrow A_2$	by IH on A_1, $\mathcal{E}'_1$, $\mathcal{D}_1$
$\Gamma \longrightarrow C$	by IH on A_2, $\mathcal{D}'_1$, and $\mathcal{E}'_2$

Case 2-(b): A is not the principal formula of $R_{\mathcal{D}}$. We have $\Gamma = \Gamma', B_1 \supset B_2$.

$$\mathcal{D} = \frac{\overset{\mathcal{D}_1}{\Gamma', B_1 \supset B_2 \longrightarrow B_1} \quad \overset{\mathcal{D}_2}{\Gamma', B_1 \supset B_2, B_2 \longrightarrow A}}{\Gamma', B_1 \supset B_2 \longrightarrow A}\ \supset L$$

$\mathcal{E}' :: \Gamma', B_1 \supset B_2, B_2, A \longrightarrow C$ by weakening $\mathcal{E} :: \Gamma', B_1 \supset B_2, A \longrightarrow C$
$\mathcal{E}'' :: \Gamma', B_1 \supset B_2, B_2 \longrightarrow C$ by IH on A, $\mathcal{D}_2$, and $\mathcal{E}'$
$\Gamma', B_1 \supset B_2 \longrightarrow C$ by the rule $\supset L$ with $\mathcal{D}_1$ and $\mathcal{E}''$
$\Gamma \longrightarrow C$ from $\Gamma = \Gamma', B_1 \supset B_2$

Case 2-(c): A is not the principal formula of $R_{\mathcal{E}}$.

Subcase: $\Gamma = \Gamma', B_1 \supset B_2$ where $B_1 \supset B_2$ is the principal formula of $R_{\mathcal{E}}$

$$\mathcal{E} :: \frac{\overset{\mathcal{E}_1}{\Gamma', B_1 \supset B_2, A \longrightarrow B_1} \quad \overset{\mathcal{E}_2}{\Gamma', B_1 \supset B_2, A, B_2 \longrightarrow C}}{\Gamma', B_1 \supset B_2, A \longrightarrow C} \supset L$$

$\mathcal{E}_1' :: \Gamma', B_1 \supset B_2 \longrightarrow B_1$ by IH on A, $\mathcal{D}$, and $\mathcal{E}_1$
$\mathcal{D}' :: \Gamma', B_1 \supset B_2, B_2 \longrightarrow A$ by weakening $\mathcal{D} :: \Gamma \longrightarrow A$ (with $\Gamma = \Gamma', B_1 \supset B_2$)
$\mathcal{E}_2' :: \Gamma', B_1 \supset B_2, B_2 \longrightarrow C$ by IH on A, $\mathcal{D}'$, and $\mathcal{E}_2$
$\Gamma', B_1 \supset B_2 \longrightarrow C$ by the rule $\supset L$ with $\mathcal{E}_1'$ and $\mathcal{E}_2'$
$\Gamma \longrightarrow C$ from $\Gamma = \Gamma', B_1 \supset B_2$

Subcase: $C = C_1 \supset C_2$ is the principal formula of $R_{\mathcal{E}}$

$$\mathcal{E} :: \frac{\overset{\mathcal{E}_1}{\Gamma, A, C_1 \longrightarrow C_2}}{\Gamma, A \longrightarrow C_1 \supset C_2} \supset R$$

$\mathcal{D}' :: \Gamma, C_1 \longrightarrow A$ by weakening $\mathcal{D} :: \Gamma \longrightarrow A$
$\mathcal{E}_1' :: \Gamma, C_1 \longrightarrow C_2$ by IH on A, $\mathcal{D}'$, and $\mathcal{E}_1$
$\Gamma \longrightarrow C_1 \supset C_2$ by the rule $\supset R$ with $\mathcal{E}_1'$
$\Gamma \longrightarrow C$ from $C = C_1 \supset C_2$
□

The admissibility of the cut rule has as a corollary one of the central theorems in the study of logic: *cut elimination* (also called *Hauptsatz* meaning "main theorem"). Consider an extension of the sequent calculus with the cut rule shown below, where we use sequents of the form $\Gamma \longrightarrow^{+} C$ to distinguish the extended system from the system in Figure 3.1:

$$\frac{\Gamma \longrightarrow^{+} A \quad \Gamma, A \longrightarrow^{+} C}{\Gamma \longrightarrow^{+} C} \; \mathit{Cut}$$

Cut elimination states that the rule Cut is redundant:

Theorem 3.7 (Cut elimination). *$\Gamma \longrightarrow C$ if and only if $\Gamma \longrightarrow^{+} C$.*

Proof. The *only if* part is trivial. For the *if* part, we proceed by induction on the structure of the proof of $\Gamma \longrightarrow^{+} C$. The only interesting case is the rule Cut:

Case $\dfrac{\Gamma \longrightarrow^{+} A \quad \Gamma, A \longrightarrow^{+} C}{\Gamma \longrightarrow^{+} C} \; \mathit{Cut}$

$\Gamma \longrightarrow A$ by induction hypothesis on $\Gamma \longrightarrow^{+} A$
$\Gamma, A \longrightarrow C$ by induction hypothesis on $\Gamma, A \longrightarrow^{+} C$
$\Gamma \longrightarrow C$ by Theorem 3.6
□

Note that the rule Cut destroys the subformula property: it does not analyze a proposition in the conclusion, so A can be an arbitrary proposition completely unrelated to Γ and C. Thus the presence of the rule Cut makes it difficult to prove a sequent because each application of the rule Cut must "guess" such a proposition A. Fortunately the cut elimination theorem says that the rule Cut can be discarded without sacrificing the expressive power of the sequent calculus.

3.3 Normalization for the natural deduction system

Theorem 3.8 states that for every proof of $A\ true$, there exists a proof of $A\uparrow$. We now appeal to the cut elimination theorem to prove the same result, but covering all connectives (including $\vee$ and $\bot$). Our goal is to prove the normalization theorem stated in terms of hypothetical judgments:

Theorem 3.8 (Normalization). $\Gamma \vdash A\ true$ *if and only if* $\Gamma_\downarrow \vdash A\uparrow$.

To this end, we introduce two *annotated judgments* $\Gamma_\downarrow \vdash^+ A\downarrow$ and $\Gamma_\downarrow \vdash^+ A\uparrow$, for which we use the following rule in addition to those rules for $\Gamma_\downarrow \vdash A\downarrow$ and $\Gamma_\downarrow \vdash A\uparrow$:

$$\frac{\Gamma_\downarrow \vdash^+ A\uparrow}{\Gamma_\downarrow \vdash^+ A\downarrow}\ \uparrow\downarrow$$

As it is based on hypothetical judgments, the new system satisfies the substitution principle (which extends Theorem 1.10); we assume that the exchange rule is built-in:

Theorem 3.9 (Substitution).
If $\Gamma_\downarrow \vdash^+ A\downarrow$ *and* $\Gamma_\downarrow, A\downarrow \vdash^+ C\downarrow$, *then* $\Gamma_\downarrow \vdash^+ C\downarrow$.
If $\Gamma_\downarrow \vdash^+ A\downarrow$ *and* $\Gamma_\downarrow, A\downarrow \vdash^+ C\uparrow$, *then* $\Gamma_\downarrow \vdash^+ C\uparrow$.

In conjunction with the rule $\downarrow\uparrow$, the rule $\uparrow\downarrow$ effectively collapses the distinction between $A\uparrow$ and $A\downarrow$: a proof of $A\uparrow$ leads to a proof of $A\downarrow$ and vice versa. Thus both $A\uparrow$ and $A\downarrow$ in the new system are essentially no different from $A\ true$, as stated in the two theorems below; $\Gamma_\downarrow = \{A\downarrow \mid A\ true \in \Gamma\}$ is a collection of neutral judgments derived from truth judgments in Γ:

Theorem 3.10 (Soundness of the annotated judgments).
If $\Gamma_\downarrow \vdash^+ A\downarrow$, *then* $\Gamma \vdash A\ true$.
If $\Gamma_\downarrow \vdash^+ A\uparrow$, *then* $\Gamma \vdash A\ true$.

Proof. By simultaneous induction on the structure of the proof of $\Gamma_\downarrow \vdash^+ A\downarrow$ and $\Gamma_\downarrow \vdash^+ A\uparrow$. □

Theorem 3.11 (Completeness of the annotated judgments).
If $\Gamma \vdash A\ true$, *then* $\Gamma_\downarrow \vdash^+ A\downarrow$.
If $\Gamma \vdash A\ true$, *then* $\Gamma_\downarrow \vdash^+ A\uparrow$.

Proof. By induction on the structure of the proof of $\Gamma \vdash A\ true$. We use the rules $\downarrow\uparrow$ and $\uparrow\downarrow$ to convert between $A\downarrow$ and $A\uparrow$ whenever necessary. We show two cases.

Case $\dfrac{\Gamma, A_1\ true \vdash A_2\ true}{\Gamma \vdash A_1 \supset A_2\ true}\ \supset\mathsf{I}$ where $A = A_1 \supset A_2$

$\Gamma_\downarrow, A_1\downarrow \vdash^+ A_2\uparrow$ by induction hypothesis on $\Gamma, A_1\ true \vdash A_2\ true$
$\Gamma_\downarrow \vdash^+ A_1 \supset A_2\uparrow$ by the rule $\supset\mathsf{I}_\uparrow$, proving the second clause
$\Gamma_\downarrow \vdash^+ A_1 \supset A_2\downarrow$ by the rule $\uparrow\downarrow$, proving the first clause

Case $\dfrac{\Gamma \vdash B \supset A\ true \quad \Gamma \vdash B\ true}{\Gamma \vdash A\ true}\ \supset\mathsf{E}$

$\Gamma_\downarrow \vdash^+ B \supset A\downarrow$ by induction hypothesis on $\Gamma \vdash B \supset A\ true$
$\Gamma_\downarrow \vdash^+ B\uparrow$ by induction hypothesis on $\Gamma \vdash B\ true$
$\Gamma_\downarrow \vdash^+ A\downarrow$ by the rule $\supset\mathsf{E}_\downarrow$, proving the first clause
$\Gamma_\downarrow \vdash^+ A\uparrow$ by the rule $\downarrow\uparrow$, proving the second clause

□

Now we can complete the proof of Theorem 3.8 by showing that $\Gamma \longrightarrow^+ A$ and $\Gamma_\downarrow \vdash^+ A\uparrow$ are equivalent (Theorems 3.12 and 3.13); we use an appropriate definition of $\Gamma_\downarrow$ depending on the definition of Γ:

$$
\begin{array}{lcll}
\Gamma_\downarrow \vdash A\uparrow & \Longleftrightarrow & \Gamma \longrightarrow A & \textit{by Theorems 3.3 and 3.4} \\
& \Longleftrightarrow & \Gamma \longrightarrow^+ A & \textit{by Theorem 3.7} \\
& \Longleftrightarrow & \Gamma_\downarrow \vdash^+ A\uparrow & \textit{by Theorems 3.12 and 3.13} \\
& \Longleftrightarrow & \Gamma \vdash A\ \mathit{true} & \textit{by Theorems 3.10 and 3.11}
\end{array}
$$

$\Gamma \vdash A\ \mathit{true}$ $\qquad$ $\Gamma_\downarrow \vdash A\uparrow$, $\Gamma_\downarrow \vdash A\downarrow$ $\qquad$ $\Gamma \longrightarrow A$ $\qquad$ $\Gamma \longrightarrow A$

$$\frac{\Gamma_\downarrow \vdash^+ A\uparrow}{\Gamma_\downarrow \vdash^+ A\downarrow}\ \Uparrow\Downarrow \qquad\qquad \frac{\Gamma \longrightarrow^+ A \quad \Gamma, A \longrightarrow^+ C}{\Gamma \longrightarrow^+ C}\ \mathit{Cut}$$

The proof of Theorems 3.12 and 3.13 is almost the same as the proof of Theorems 3.3 and 3.4, except for the additional case in which the rule $\Uparrow\Downarrow$ or Cut is involved. The proof of of Theorem 3.13 follows from a lemma similar to Lemma 3.5.

Theorem 3.12 (Soundness of the sequent calculus with the cut rule). *If* $\Gamma \longrightarrow^+ C$, *then* $\Gamma_\downarrow \vdash^+ C\uparrow$.

Proof. By induction on the structure of the proof of $\Gamma \longrightarrow^+ C$. We show the case for the rule Cut.

Case $\dfrac{\Gamma \longrightarrow^+ A \quad \Gamma, A \longrightarrow^+ C}{\Gamma \longrightarrow^+ C}\ \mathit{Cut}$

$\Gamma_\downarrow \vdash^+ A\uparrow$ by induction hypothesis on $\Gamma \longrightarrow^+ A$
$\Gamma_\downarrow \vdash^+ A\downarrow$ by the rule $\Uparrow\Downarrow$
$\Gamma_\downarrow, A\downarrow \vdash^+ C\uparrow$ by induction hypothesis on $\Gamma, A \longrightarrow^+ C$
$\Gamma_\downarrow \vdash^+ C\uparrow$ by Theorem 3.9 with $\Gamma_\downarrow \vdash^+ A\downarrow$ and $\Gamma_\downarrow, A\downarrow \vdash^+ C\uparrow$

□

Theorem 3.13 (Completeness of the sequent calculus with the cut rule). *If* $\Gamma_\downarrow \vdash^+ C\uparrow$, *then* $\Gamma \longrightarrow^+ C$.

Lemma 3.14.
If $\Gamma_\downarrow \vdash^+ A\downarrow$, *then* $\Gamma, A \longrightarrow^+ C$ *implies* $\Gamma \longrightarrow^+ C$.
If $\Gamma_\downarrow \vdash^+ C\uparrow$, *then* $\Gamma \longrightarrow^+ C$.

Proof. By simultaneous induction on the structure of the proof of $\Gamma_\downarrow \vdash^+ A\downarrow$ and $\Gamma_\downarrow \vdash^+ C\uparrow$. We show the case for the rule $\Uparrow\Downarrow$.

Case $\dfrac{\Gamma_\downarrow \vdash^+ A\uparrow}{\Gamma_\downarrow \vdash^+ A\downarrow}\ \Uparrow\Downarrow$

$\Gamma \longrightarrow^+ A$ by induction hypothesis on $\Gamma_\downarrow \vdash^+ A\uparrow$
$\Gamma, A \longrightarrow^+ C$ assumption
$\Gamma \longrightarrow^+ C$ by the rule Cut with $\Gamma \longrightarrow^+ A$ and $\Gamma, A \longrightarrow^+ C$

□

Implicit in the proof of Theorem 3.8 is that the proof is constructive: it gives an algorithm for converting a proof of $\Gamma \vdash A\ \mathit{true}$ into a proof of $\Gamma_\downarrow \vdash A\uparrow$. (Converting a proof of $\Gamma_\downarrow \vdash A\uparrow$ into a proof of $\Gamma \vdash A\ \mathit{true}$ is trivial.) It is a consequence of constructive proofs of all the theorems involved, in particular Theorem 3.13 (completeness of the sequent calculus with the cut rule) which is generalized to Lemma 3.14, and Theorem 3.6 (admissibility of the cut rule) which is used in the proof of Theorem 3.7 (cut elimination). To be specific, we convert a proof of $\Gamma_\downarrow \vdash A\uparrow$ into a proof of $\Gamma \vdash A\ \mathit{true}$ as follows:

1. $\Gamma \vdash A\ true$ to $\Gamma_\downarrow \vdash^+ A\uparrow$ by Theorem 3.11. We annotate the proof of $\Gamma \vdash A\ true$ by replacing $A\ true$ by $A\uparrow$ or $A\downarrow$, inserting the rule $\uparrow\downarrow$ whenever a detour is encountered.

2. $\Gamma_\downarrow \vdash^+ A\uparrow$ to $\Gamma \longrightarrow^+ A$ by Theorem 3.13. We insert the rule Cut whenever the rule $\uparrow\downarrow$ is encountered.

3. $\Gamma \longrightarrow^+ A$ to $\Gamma \longrightarrow A$ by Theorem 3.7. We use the proof of Theorem 3.6 to remove the rule Cut.

4. $\Gamma \longrightarrow A$ to $\Gamma_\downarrow \vdash A\uparrow$ by Theorem 3.3.

Thus in the heart of the proof of the normalization theorem lies the cut elimination theorem!

A corollary of the normalization theorem (or its proof) is consistency of propositional logic (or first-order logic if universal and existential quantifiers are added): $\bot\ true$ is not provable in propositional logic.

Corollary 3.15 (Consistency). *There is no proof of $\cdot \vdash \bot\ true$.*

Proof. It suffices to show that there is no proof of $\cdot \vdash \bot\uparrow$ (by the normalization theorem), or $\cdot \longrightarrow \bot$ (by Theorems 3.3 and 3.4). Since no rule is applicable to $\cdot \longrightarrow \bot$, there is no proof of $\cdot \longrightarrow \bot$. □

Another corollary is that $A \vee B\ true$ is provable only if either $A\ true$ or $B\ true$ is provable.

Corollary 3.16. *If $\cdot \vdash A \vee B\ true$, then either $\cdot \vdash A\ true$ or $\cdot \vdash B\ true$.*

Proof. $\cdot \vdash A \vee B\ true$ implies $\cdot \longrightarrow A \vee B$, as shown in the proof of the normalization theorem. Since the only way to prove $\cdot \longrightarrow A \vee B$ is by applying either $\vee R_L$ or $\vee R_R$, either $\cdot \longrightarrow A$ or $\cdot \longrightarrow B$ must hold. Therefore either $\cdot \vdash A\ true$ or $\cdot \vdash B\ true$ holds. □

Note, however, that $\Gamma \vdash A \vee B\ true$ does not necessarily imply either $\Gamma \vdash A\ true$ or $\Gamma \vdash B\ true$ if Γ is not empty. For example, $B \vee A\ true \vdash A \vee B\ true$ is provable, but neither $B \vee A\ true \vdash A\ true$ nor $B \vee A\ true \vdash B\ true$ is provable.

Finally constructive logic is shown to be different from classical logic: $A \vee \neg A\ true$, which is called *the law of excluded middle* and is an axiom in classical logic, is not provable in constructive logic.

Corollary 3.17. *There is no proof of $\cdot \vdash A \vee \neg A\ true$ for an arbitrary proposition A.*

Proof. If $\cdot \vdash A \vee \neg A\ true$ holds, then either $\cdot \longrightarrow A$ or $\cdot \longrightarrow \neg A$ holds, as shown in the proof of Corollary 3.16. The first sequent is not provable for an arbitrary proposition A. The second sequent is not provable because $A \longrightarrow \bot$ is not provable. □

Note that the law of excluded middle assumes an *arbitrary* proposition A; the use of a specific proposition A makes $A \vee \neg A\ true$ provable. For example, by letting $A = \top$, we obtain $\top \vee \neg\top\ true$, which is certainly provable.

3.4 Contraction and weakening

3.5 Proof terms for the sequent calculus

Chapter 4

First-Order Logic

4.1 Terms

$$\text{term} \quad t \quad ::= \quad x \mid y \mid \cdots \mid a \mid b \mid \cdots \mid f(t_1, \cdots, t_n) \mid c$$

- x is called a *term variable* which ranges over the set of terms.
- a is called a *parameter* and denotes an arbitrary/unspecified term about which we can make no assumption.
- f is called a *function symbol*. $f(t_1, \cdots, t_n)$ is a term where f is a function symbol of arity n and $t_1, \cdots, t_n$ are its arguments.
- f is not a function in that $f(t_1, \cdots, t_n)$ does not reduce to another term; $f(t_1, \cdots, t_n)$ is a term in itself.
- A *constant* c is a function symbol with zero arity. We abbreviate $c()$ as c.
- Terms are not to be confused with proof terms: a term can be any kind of object (*e.g.*, natural number, tree, boolean value, student name, *etc*) whereas a proof term is a particular kind of object representing a proof in logic.

Example

$$\text{natural number} \quad t \quad ::= \quad \mathbf{0} \mid \mathbf{s}(t)$$

4.2 Propositions in first-order logic

$$\text{proposition} \quad A \quad ::= \quad P(t_1, \cdots, t_n) \mid \cdots \mid \forall x.A \mid \exists x.A$$

$$\frac{}{P(t_1, \cdots, t_n)\ \mathit{prop}}\ P\mathsf{F} \qquad \frac{A\ \mathit{prop}}{\forall x.A\ \mathit{prop}}\ \forall\mathsf{F} \qquad \frac{A\ \mathit{prop}}{\exists x.A\ \mathit{prop}}\ \exists\mathsf{F}$$

- P is called a *predicate symbol*. A *predicate* $P(t_1, \cdots, t_n)$ is a proposition which expresses a certain relation between terms $t_1, \cdots, t_n$. Thus we may think of predicates as propositions about objects.
- A propositional constant P is a predicate symbol with zero arity. We abbreviate $P()$ as P.
- $\forall x.A$ uses a *universal quantifier* $\forall$ to introduce a term variable x. Roughly speaking, the truth of $\forall x.A$ means that A is true for "every" term x.

- $\exists x.A$ uses a *existential quantifier* $\exists$ to introduce a term variable x. Roughly speaking, the truth of $\exists x.A$ means that A is true for "some" term x.
- Quantifiers $\forall$ and $\exists$ have the lowest operator precedence. For example, $\forall x.A \supset B$ is understood as $\forall x.(A \supset B)$; similarly $\exists x.A \supset B$ is understood as $\exists x.(A \supset B)$.
- As quantifiers introduce term variables, there arises a need for substitutions for term variables in propositions or proofs. We write $[t/x]A$ for the result of substituting t for x in proposition A. Similarly we write $[t/x]\mathcal{D}$ for the result of substituting t for x throughout proof $\mathcal{D}$.
- we write $[t/a]A$ and $[t/a]\mathcal{D}$ for the result of substituting t for *parameter* a in A and $\mathcal{D}$, respectively.
- Variable captures never occur in first-order logic: in a substitution $[t/x]A$ or $[t/x]\mathcal{D}$, term t is always closed, *i.e.*, it does not contain free term variables.

Example

$$\text{proposition} \quad A \quad ::= \quad Nat(t) \mid Eq(t,t) \mid \cdots$$

4.3 Universal quantifier

$$\frac{[a/x]A\ true}{\forall x.A\ true}\ \forall \mathsf{I}^a \qquad \frac{\forall x.A\ true}{[t/x]A\ true}\ \forall \mathsf{E}$$

- We may think of $\forall x.A$ as an infinite conjunction

  $$[t_1/x]A \wedge [t_2/x]A \wedge \cdots \wedge [t_i/x]A \wedge \cdots$$

  where $t_1, t_2, \cdots, t_i, \cdots$ enumerate all terms available.
- Parameter a denotes an an arbitrary term about which we can make no assumption, and the proof of $[a/x]A\ true$ in the premise of the rule $\forall \mathsf{I}^a$ is parameteric in a.
- In the rule $\forall \mathsf{I}^a$, parameter a must be fresh in that it is not found in any undischarged hypothesis in the proof of the premise.
- We may think of parameter a in the rule $\forall \mathsf{I}^a$ as an "arbitrary" term that is specific to the rule $\forall \mathsf{I}^a$ or that becomes fixed when the premise $[a/x]A\ true$ is decided. Hence using parameter a again for another instance of the rule $\forall \mathsf{I}$ results in a wrong proof. In the proof shown below, parameter a is specific to the rule $\forall \mathsf{I}^a$ in the bottom, as indicated by the partial proof in the left. From the point of view of the rule $\forall \mathsf{I}^a$ in the top, therefore, the same parameter a cannot be read as an "arbitrary" term again because it can be read an "arbitrary" term only with respect to the rule $\forall \mathsf{I}^a$ in the bottom. Note that in the rule $\forall \mathsf{I}^a$ in the top, parameter a is already found in an undischarged hypothesis $\overline{Nat(a)\ true}^{\,w}$.

$$\frac{\begin{array}{c}\vdots\\ \forall y.Nat(a) \supset Nat(y)\ true\end{array}}{\forall x.\forall y.Nat(x) \supset Nat(y)\ true}\ \forall \mathsf{I}^a \quad \Longrightarrow \quad \frac{\dfrac{\dfrac{\dfrac{\overline{Nat(a)\ true}^{\,w}}{\forall x.Nat(x)\ true}\ \forall \mathsf{I}^a\ \textit{(wrong)}}{Nat(b)\ true}\ \forall \mathsf{E}}{Nat(a) \supset Nat(b)\ true}\ \supset \mathsf{I}^w}{\forall y.Nat(a) \supset Nat(y)\ true}\ \forall \mathsf{I}^b}{\forall x.\forall y.Nat(x) \supset Nat(y)\ true}\ \forall \mathsf{I}^a$$

- In the rule $\forall \mathsf{E}$, we may use any term for t — term variable, parameter, function symbol, constant, *etc*.

- Here is an example of a proof involving universal quantifiers which uses $[a/x](A \wedge B) = [a/x]A \wedge [a/x]B$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\forall x.A \wedge B}^{\,w}}{[a/x](A \wedge B)\ true}\,\forall\mathsf{E}}{[a/x]A\ true}\,\wedge\mathsf{E_L}}{\forall x.A\ true}\,\forall\mathsf{I}^a \qquad \dfrac{\dfrac{\dfrac{\overline{\forall x.A \wedge B}^{\,w}}{[a/x](A \wedge B)\ true}\,\forall\mathsf{E}}{[a/x]B\ true}\,\wedge\mathsf{E_R}}{\forall x.B\ true}\,\forall\mathsf{I}^a}{(\forall x.A) \wedge (\forall x.B)\ true}\,\wedge\mathsf{I}}{(\forall x.A \wedge B) \supset (\forall x.A) \wedge (\forall x.B)\ true}\,\supset\mathsf{I}^w$$

Hypothetical judgments

$$\dfrac{\Gamma \vdash [a/x]A\ true}{\Gamma \vdash \forall x.A\ true}\,\forall\mathsf{I}^a \qquad \dfrac{\Gamma \vdash \forall x.A\ true}{\Gamma \vdash [t/x]A\ true}\,\forall\mathsf{E}$$

Local reduction and local expansion

$$\dfrac{\dfrac{\overset{\mathcal{D}}{[a/x]A\ true}}{\forall x.A\ true}\,\forall\mathsf{I}^a}{[t/x]A\ true}\,\forall\mathsf{E} \quad \Longrightarrow_R \quad \overset{[t/a]\mathcal{D}}{[t/x]A\ true}$$

$$\overset{\mathcal{E}}{\forall x.A\ true} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\overset{\mathcal{E}}{\forall x.A\ true}}{[a/x]A\ true}\,\forall\mathsf{E}}{\forall x.A\ true}\,\forall\mathsf{I}^a$$

Normal and neutral judgments

$$\dfrac{[a/x]A\uparrow}{\forall x.A\uparrow}\,\forall\mathsf{I}^a \qquad \dfrac{\forall x.A\downarrow}{[t/x]A\downarrow}\,\forall\mathsf{E}$$

$$\dfrac{\Gamma_\downarrow \vdash [a/x]A\uparrow}{\Gamma_\downarrow \vdash \forall x.A\uparrow}\,\forall\mathsf{I}^a \qquad \dfrac{\Gamma \vdash \forall x.A\downarrow}{\Gamma \vdash [t/x]A\downarrow}\,\forall\mathsf{E}$$

4.4 Existential quantifier

$$\dfrac{[t/x]A\ true}{\exists x.A\ true}\,\exists\mathsf{I} \qquad \dfrac{\exists x.A\ true \qquad \begin{array}{c}\overline{[a/x]A\ true}^{\,w}\\ \vdots \\ C\ true\end{array}}{C\ true}\,\exists\mathsf{E}^{a,w}$$

- We may think of $\forall x.A$ as an infinite disjunction

$$[t_1/x]A \vee [t_2/x]A \vee \cdots \vee [t_i/x]A \vee \cdots$$

  where $t_1, t_2, \cdots, t_i, \cdots$ enumerate all terms available.

- The rule $\exists$I says that in order to prove $\exists x.A\ true$, we must present a concrete term, or a *witness*, t such that $[t/x]A\ true$ is provable. It is not enough to state that there exists such a term without actually knowing what it is.

- The necessity of a witness in the rule $\exists\mathsf{I}$ is a distinguishing feature of constructive logic. In contrast, a proof of $\exists x.A\ true$ in classical logic only needs to show that there exists a term t, *which may or may not be known*, such that $[t/x]A\ true$ is provable. In other words, a proof of $\exists x.A\ true$ essentially shows that it cannot happen that there exists no term t such that $[t/x]A\ true$ is provable. As a consequence, $\exists x.A$ is no different from $\neg\forall x.\neg A$ in classical logic.
- In the rule $\exists\mathsf{E}$, we do not know the witness for the proof of $\exists x.A\ true$ and thus cannot make any assumption about it. Therefore, if we are to make use of a proof of $\exists x.A\ true$, we have to introduce a fresh parameter a.
- In the rule $\exists\mathsf{E}$, parameter a must not be found in A or any undischarged hypothesis. In particular, it must not be found in C; otherwise the rule ends up with a conclusion that makes too strong an assumption about the witness, namely that it can be an arbitrary term! For example, the following proof draws a nonsensical conclusion that an arbitrary term is equal to a natural number $\mathbf{0}$, as it allows parameter a to appear in the conclusion:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{Nat(\mathbf{0})\ true} \quad \dfrac{\overline{\forall x.Eq(x,x)\ true}}{Eq(\mathbf{0},\mathbf{0})\ true}\,\forall\mathsf{E}}{Nat(\mathbf{0}) \wedge Eq(\mathbf{0},\mathbf{0})\ true}\,\wedge\mathsf{I}}{\exists x.Nat(x) \wedge Eq(x,\mathbf{0})\ true}\,\exists\mathsf{I} \quad \dfrac{\dfrac{}{Nat(a) \wedge Eq(a,\mathbf{0})\ true}\,w}{Eq(a,\mathbf{0})\ true}\,\wedge\mathsf{E_R}}{Eq(a,\mathbf{0})\ true}\,\exists\mathsf{E}^{a,w}$$

Example

- $\exists x.\neg A \supset \neg\forall x.A\ true$ is provable. Intuitively a proof of $\exists x.\neg A\ true$ gives us a witness t such that $[t/x]\neg A\ true$ is provable, and we can use t to refute $\forall x.A\ true$.

$$\dfrac{\dfrac{\dfrac{}{\exists x.\neg A\ true}\,w \quad \dfrac{\dfrac{}{[a/x]\neg A\ true}\,y \quad \dfrac{\overline{\forall x.A\ true}^{\,z}}{[a/x]A\ true}\,\forall\mathsf{E}}{\bot\ true}\,\neg\mathsf{E}}{\bot\ true}\,\exists\mathsf{E}^{a,y}}{\neg\forall x.A\ true}\,\neg\mathsf{I}^{z}}{\exists x.\neg A \supset \neg\forall x.A\ true}\,\supset\mathsf{I}^{w}$$

- $\neg\forall x.A \supset \exists x.\neg A\ true$ is not provable. Intuitively a proof of $\exists x.\neg A\ true$ requires a witness t such that $[t/x]\neg A\ true$ is provable, but no proof of $\neg\forall x.A\ true$ gives such a witness.

$$\dfrac{\dfrac{\dfrac{}{\neg\forall x.A\ true}\,w \quad \overset{?}{\forall x.A\ true}}{\bot\ true}\,\neg\mathsf{E}}{\exists x.\neg A\ true}\,\bot\mathsf{E}}{\neg\forall x.A \supset \exists x.\neg A\ true}\,\supset\mathsf{I}^{w}$$

- $(\forall x.A) \supset (\exists x.A)\ true$ is *not* provable. The reason is that although $\forall x.A\ true$ states that $[t/x]A\ true$ is provable for any term t, it does not decide a concrete term t such that $[t/x]A\ true$ is provable. In particular, if the set of terms is empty, $\forall x.A\ true$ holds trivially (because there is no term), but $\exists x.A\ true$ never holds because it is impossible to choose a term t for x, regardless of proposition A.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.A\ true}\,w}{[t/x]A\ true?}\,\forall\mathsf{E}}{\exists x.A\ true}\,\exists\mathsf{I}}{(\forall x.A) \supset (\exists x.A)\ true}\,\supset\mathsf{I}^{w}$$

- On the other hand, $\forall y.(\forall x.A) \supset (\exists x.A)\ true$ *is* provable even if y does not occur free in A. The difference from the previous example is that $\forall y$ allows us to make an assumption that the set of terms is not empty. In the proof shown below, paramter a denotes an arbitrary term in the set of terms, and its presence implies that the set of terms in not empty.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\forall x.A\ true}^{\,w}}{[a/x]A\ true}\,\forall\mathsf{E}}{\exists x.A\ true}\,\exists\mathsf{I}}{(\forall x.A) \supset (\exists x.A)\ true}\,\supset\mathsf{I}^{w}}{\forall y.(\forall x.A) \supset (\exists x.A)\ true}\,\forall\mathsf{I}^{a}$$

- The two examples above illustrate that in constructive logic, $\forall x.A$ is not equivalent to A even if x does not occur free in A at all: $\forall x.A$ asserts A on the assumption that the set of terms is not empty, whereas A without a universal quantifier cannot exploit such an assumption.

Hypothetical judgments

$$\dfrac{\Gamma \vdash [t/x]A\ true}{\Gamma \vdash \exists x.A\ true}\,\exists\mathsf{I} \qquad \dfrac{\Gamma \vdash \exists x.A\ true \quad \Gamma, [a/x]A\ true \vdash C\ true}{\Gamma \vdash C\ true}\,\exists\mathsf{E}^{a}$$

Local reduction and local expansion

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \dfrac{[t/x]A\ true}{\exists x.A\ true}\,\exists\mathsf{I}\end{array} \qquad \left.\begin{array}{c}\overline{[a/x]A\ true}^{\,w}\\ \vdots\\ C\ true\end{array}\right\}\mathcal{E}}{C\ true}\,\exists\mathsf{E}^{a,w} \quad \Longrightarrow_R \quad \left.\begin{array}{c}\mathcal{D}\\ [t/x]A\ true\\ \vdots\\ C\ true\end{array}\right\}[t/a]\mathcal{E}$$

$$\begin{array}{c}\mathcal{E}\\ \exists x.A\ true\end{array} \quad \Longrightarrow_E \quad \dfrac{\begin{array}{c}\mathcal{E}\\ \exists x.A\ true\end{array} \qquad \dfrac{\overline{[a/x]A\ true}^{\,w}}{\exists x.A\ true}\,\exists\mathsf{I}}{\exists x.A\ true}\,\exists\mathsf{E}^{a,w}$$

In the local reduction, $[t/a]\mathcal{E}$ does not affect the conclusion $C\ true$ because parameter a does not appear in C. $[t/x]A\ true$ is obtained from $[t/a][a/x]A\ true$.

Normal and neutral judgments

$$\dfrac{[t/x]A\uparrow}{\exists x.A\uparrow}\,\exists\mathsf{I} \qquad \dfrac{\exists x.A\downarrow \qquad \begin{array}{c}\overline{[a/x]A\downarrow}^{\,w}\\ \vdots\\ C\uparrow\end{array}}{C\uparrow}\,\exists\mathsf{E}^{a,w}$$

$$\dfrac{\Gamma_\downarrow \vdash [t/x]A\uparrow}{\Gamma_\downarrow \vdash \exists x.A\uparrow}\,\exists\mathsf{I} \qquad \dfrac{\Gamma_\downarrow \vdash \exists x.A\downarrow \quad \Gamma_\downarrow, [a/x]A\downarrow \vdash C\uparrow}{\Gamma_\downarrow \vdash C\uparrow}\,\exists\mathsf{E}^{a}$$

4.5 Examples

Axioms

$$\frac{}{Nat(\mathbf{0})\ true}\ Zero \qquad \frac{}{\forall x.Nat(x) \supset Nat(\mathbf{s}(x))\ true}\ Succ$$

$$\frac{}{\forall x.Eq(x,x)\ true}\ Eq_i \qquad \frac{}{\forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z)\ true}\ Eq_t$$

$$\frac{}{\forall x.Lt(x,\mathbf{s}(x))\ true}\ Lt_s \qquad \frac{}{\forall x.\forall y.Eq(x,y) \supset \neg Lt(x,y)\ true}\ Lt_\neg$$

Theorems

Proof of $\forall x.Nat(x) \supset (\exists y.Nat(y) \wedge Eq(x,y))\ true$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Nat(a)\ true}^{z} \quad \dfrac{\dfrac{}{\forall x.Eq(x,x)\ true}\ Eq_i}{Eq(a,a)\ true}\ \forall\mathsf{E}}{Nat(a) \wedge Eq(a,a)\ true}\ \wedge\mathsf{I}}{\exists y.Nat(y) \wedge Eq(a,y)\ true}\ \exists\mathsf{I}}{Nat(a) \supset (\exists y.Nat(y) \wedge Eq(a,y))\ true}\ \supset\mathsf{I}^{z}}{\forall x.Nat(x) \supset (\exists y.Nat(y) \wedge Eq(x,y))\ true}\ \forall\mathsf{I}^{a}$$

Proof of $\forall x.\forall y.Eq(x,y) \supset Eq(y,x)\ true$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z)\ true}\ Eq_t}{\forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z)\ true}\ \forall\mathsf{E}}{\forall z.(Eq(a,b) \wedge Eq(a,z)) \supset Eq(b,z)\ true}\ \forall\mathsf{E}}{(Eq(a,b) \wedge Eq(a,a)) \supset Eq(b,a)\ true}\ \forall\mathsf{E} \quad \dfrac{\dfrac{}{Eq(a,b)\ true}^{w} \quad \dfrac{\dfrac{}{\forall x.Eq(x,x)\ true}\ Eq_i}{Eq(a,a)\ true}\ \forall\mathsf{E}}{Eq(a,b) \wedge Eq(a,a)\ true}\ \wedge\mathsf{I}}{Eq(b,a)\ true}\ \supset\mathsf{E}}{Eq(a,b) \supset Eq(b,a)\ true}\ \supset\mathsf{I}^{w}}{\forall y.Eq(a,y) \supset Eq(y,a)\ true}\ \forall\mathsf{I}^{b}}{\forall x.\forall y.Eq(x,y) \supset Eq(y,x)\ true}\ \forall\mathsf{I}^{a}$$

Proof of $\neg\exists x.Eq(x,\mathbf{0}) \wedge Eq(x,\mathbf{s}(\mathbf{0}))\ true$:

$$\mathcal{D} = \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z)\ true}\ Eq_t}{\forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z)\ true}\ \forall\mathsf{E}}{\forall z.(Eq(a,\mathbf{0}) \wedge Eq(a,z)) \supset Eq(\mathbf{0},z)\ true}\ \forall\mathsf{E}}{(Eq(a,\mathbf{0}) \wedge Eq(a,\mathbf{s}(\mathbf{0}))) \supset Eq(\mathbf{0},\mathbf{s}(\mathbf{0}))\ true}\ \forall\mathsf{E} \quad \dfrac{}{Eq(a,\mathbf{0}) \wedge Eq(a,\mathbf{s}(\mathbf{0}))\ true}^{z}}{Eq(\mathbf{0},\mathbf{s}(\mathbf{0}))\ true}\ \supset\mathsf{E}$$

$$\dfrac{\dfrac{}{\exists x.Eq(x,\mathbf{0}) \wedge Eq(x,\mathbf{s}(\mathbf{0}))\ true}^{w} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x.\forall y.Eq(x,y) \supset \neg Lt(x,y)\ true}\ Lt_\neg}{\forall y.Eq(\mathbf{0},y) \supset \neg Lt(\mathbf{0},y)\ true}\ \forall\mathsf{E}}{Eq(\mathbf{0},\mathbf{s}(\mathbf{0})) \supset \neg Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))\ true}\ \forall\mathsf{E} \quad \overset{\mathcal{D}}{Eq(\mathbf{0},\mathbf{s}(\mathbf{0}))\ true}}{\neg Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))\ true}\ \supset\mathsf{E} \quad \dfrac{\dfrac{}{\forall x.Lt(x,\mathbf{s}(x))\ true}\ Lt_s}{Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))\ true}\ \forall\mathsf{E}}{\bot\ true}\ \neg\mathsf{E}}{\bot\ true}\ \exists\mathsf{E}^{a,z}$$

$$\frac{\bot\ true}{\neg\exists x.Eq(x,\mathbf{0}) \wedge Eq(x,\mathbf{s}(\mathbf{0}))\ true}\ \neg\mathsf{I}^{w}$$

4.6 Proof terms

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

A substitution $[t/x]M$ is defined as usual. Note that $[t/x]M$ may need a substitution $[t/x]A$ if x is found within a type A in M.

$$\frac{[a/x]M : [a/x]A}{\lambda x.\, M : \forall x.A}\ \forall\mathsf{I}^a \qquad \frac{M : \forall x.A}{M\ t : [t/x]A}\ \forall\mathsf{E}$$

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

Hypothetical judgments

$$\frac{\Gamma \vdash [a/x]M : [a/x]A}{\Gamma \vdash \lambda x.\, M : \forall x.A}\ \forall\mathsf{I}^a \qquad \frac{\Gamma \vdash M : \forall x.A}{\Gamma \vdash M\ t : [t/x]A}\ \forall\mathsf{E}$$

$$\frac{\Gamma \vdash M : [t/x]A}{\Gamma \vdash \langle t, M\rangle : \exists x.A}\ \exists\mathsf{I} \qquad \frac{\Gamma \vdash M : \exists x.A \quad \Gamma, w : [a/x]A \vdash [a/x]N : C}{\Gamma \vdash \mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ N : C}\ \exists\mathsf{E}^a$$

Local reduction and expansion

Univeral quantifier:

$$\frac{\dfrac{[a/x]M : [a/x]A}{\lambda x.\, M : \forall x.A}\ \forall\mathsf{I}^a}{(\lambda x.\, M)\ t : [t/x]A}\ \forall\mathsf{E} \qquad \Longrightarrow_R \qquad [t/a][a/x]M : [t/a][a/x]A$$

$$M : \forall x.A \qquad \Longrightarrow_E \qquad \frac{\dfrac{M : \forall x.A}{M\ a : [a/x]A}\ \forall\mathsf{E}}{\lambda x.\, M\ x : \forall x.A}\ \forall\mathsf{I}^a \ \textit{(where } M\ a = [a/x](M\ x)\textit{)}$$

$$\begin{array}{lcll}(\lambda x.\, M)\ t & \Longrightarrow_R & [t/x]M & \\ M : \forall x.A & \Longrightarrow_E & \lambda x.\, M\ x & \textit{(x is not free in M)}\end{array}$$

Existential quantifier:

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

$$M : \exists x.A \qquad \Longrightarrow_E \qquad \frac{M : \exists x.A \quad \dfrac{\overline{w : [a/x]A}}{\langle a, w\rangle : \exists x.A}\ \exists\mathsf{I}}{\mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ \langle x, w\rangle : \exists x.A}\ \exists\mathsf{E}^a \ \textit{(where } \langle a, w\rangle = [a/x]\langle x, w\rangle\textit{)}$$

$$\begin{array}{rcl}\mathsf{let}\ \langle x, w\rangle = \langle t, M\rangle\ \mathsf{in}\ N & \Longrightarrow_R & [M/w][t/x]N \\ M : \exists x.A & \Longrightarrow_E & \mathsf{let}\ \langle x, w\rangle = M\ \mathsf{in}\ \langle x, w\rangle\end{array}$$

Terms in normal form

$$\begin{array}{lrcl} \text{elim term} & E & ::= & \cdots \mid E\ t \\ \text{intro term} & I & ::= & \cdots \mid \lambda x.\, I \mid \langle t, I\rangle \mid \mathsf{let}\ \langle x, w\rangle = E\ \mathsf{in}\ I \end{array}$$

4.6.1 Examples

Axioms

$$\frac{}{\mathbf{Nat_0} : Nat(\mathbf{0})}\ Zero \qquad \frac{}{\mathbf{Nat_s} : \forall x.Nat(x) \supset Nat(\mathbf{s}(x))}\ Succ$$

$$\frac{}{\mathbf{Eq_i} : \forall x.Eq(x,x)}\ Eq_i \qquad \frac{}{\mathbf{Eq_t} : \forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z)}\ Eq_t$$

$$\frac{}{\mathbf{Lt_s} : \forall x.Lt(x, \mathbf{s}(x))}\ Lt_s \qquad \frac{}{\mathbf{Lt}_\neg : \forall x.\forall y.Eq(x,y) \supset \neg Lt(x,y)}\ Lt_\neg$$

Theorems

Proof term for $\forall x.Nat(x) \supset (\exists y.Nat(y) \wedge Eq(x,y))$ *true*:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{z : Nat(a)} \quad \dfrac{\dfrac{}{\mathbf{Eq_i} : \forall x.Eq(x,x)}\ Eq_i}{\mathbf{Eq_i}\ a : Eq(a,a)}\ \forall\mathsf{E}}{(z, \mathbf{Eq_i}\ a) : Nat(a) \wedge Eq(a,a)}\ \wedge\mathsf{I}}{\langle a, (z, \mathbf{Eq_i}\ a)\rangle : \exists y.Nat(y) \wedge Eq(a,y)}\ \exists\mathsf{I}}{\lambda z{:}Nat(a).\, \langle a, (z, \mathbf{Eq_i}\ a)\rangle : Nat(a) \supset (\exists y.Nat(y) \wedge Eq(a,y))}\ \supset\mathsf{I}^z}{\lambda x.\, \lambda z{:}Nat(x).\, \langle x, (z, \mathbf{Eq_i}\ x)\rangle : \forall x.Nat(x) \supset (\exists y.Nat(y) \wedge Eq(x,y))}\ \forall\mathsf{I}^a$$

Proof term for $\forall x.\forall y.Eq(x,y) \supset Eq(y,x)$ *true*:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{Eq_t} : \forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z)}\ Eq_t}{\mathbf{Eq_t}\ a : \forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z)}\ \forall\mathsf{E}}{\mathbf{Eq_t}\ a\ b : \forall z.(Eq(a,b) \wedge Eq(a,z)) \supset Eq(b,z)}\ \forall\mathsf{E}}{\mathbf{Eq_t}\ a\ b\ a : (Eq(a,b) \wedge Eq(a,a)) \supset Eq(b,a)}\ \forall\mathsf{E} \quad \dfrac{\dfrac{}{w : Eq(a,b)} \quad \dfrac{\dfrac{}{\mathbf{Eq_i} : \forall x.Eq(x,x)}\ Eq_i}{\mathbf{Eq_i}\ a : Eq(a,a)}\ \forall\mathsf{E}}{(w, \mathbf{Eq_i}\ a) : Eq(a,b) \wedge Eq(a,a)}\ \wedge\mathsf{I}}{\mathbf{Eq_t}\ a\ b\ a\ (w, \mathbf{Eq_i}\ a) : Eq(b,a)}\ \supset\mathsf{E}}{\lambda w{:}Eq(a,b).\, \mathbf{Eq_t}\ a\ b\ a\ (w, \mathbf{Eq_i}\ a) : Eq(a,b) \supset Eq(b,a)}\ \supset\mathsf{I}^w}{\lambda y.\, \lambda w{:}Eq(a,y).\, \mathbf{Eq_t}\ a\ y\ a\ (w, \mathbf{Eq_i}\ a) : \forall y.Eq(a,y) \supset Eq(y,a)}\ \forall\mathsf{I}^b}{\lambda x.\, \lambda y.\, \lambda w{:}Eq(x,y).\, \mathbf{Eq_t}\ x\ y\ x\ (w, \mathbf{Eq_i}\ x) : \forall x.\forall y.Eq(x,y) \supset Eq(y,x)}\ \forall\mathsf{I}^a$$

Proof term for $\neg\exists x.Eq(x, \mathbf{0}) \wedge Eq(x, \mathbf{s}(\mathbf{0}))$ *true*:

$$\mathcal{D} \quad = \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathbf{Eq_t} : \forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z)}\ Eq_t}{\mathbf{Eq_t}\ a : \forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z)}\ \forall\mathsf{E}}{\mathbf{Eq_t}\ a\ \mathbf{0} : \forall z.(Eq(a,\mathbf{0}) \wedge Eq(a,z)) \supset Eq(\mathbf{0},z)}\ \forall\mathsf{E}}{\mathbf{Eq_t}\ a\ \mathbf{0}\ \mathbf{s}(\mathbf{0}) : (Eq(a,\mathbf{0}) \wedge Eq(a,\mathbf{s}(\mathbf{0}))) \supset Eq(\mathbf{0},\mathbf{s}(\mathbf{0}))}\ \forall\mathsf{E} \quad \dfrac{}{z : Eq(a,\mathbf{0}) \wedge Eq(a,\mathbf{s}(\mathbf{0}))}}{\mathbf{Eq_t}\ a\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z : Eq(\mathbf{0},\mathbf{s}(\mathbf{0}))}\ \supset\mathsf{E}$$

$$\mathcal{E} \quad = \quad \dfrac{\dfrac{\dfrac{\dfrac{}{Lt_\neg : \forall x.\forall y.Eq(x,y) \supset \neg Lt(x,y)}\ Lt_\neg}{Lt_\neg\ \mathbf{0} : \forall y.Eq(\mathbf{0},y) \supset \neg Lt(\mathbf{0},y)}\ \forall\mathsf{E}}{Lt_\neg\ \mathbf{0}\ \mathbf{s}(\mathbf{0}) : Eq(\mathbf{0},\mathbf{s}(\mathbf{0})) \supset \neg Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))}\ \forall\mathsf{E} \quad \begin{array}{c}\mathcal{D}\\ \mathbf{Eq_t}\ a\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z : Eq(\mathbf{0},\mathbf{s}(\mathbf{0}))\end{array}}{(Lt_\neg\ \mathbf{0}\ \mathbf{s}(\mathbf{0}))\ (\mathbf{Eq_t}\ a\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z) : \neg Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))}\ \supset\mathsf{E}$$

$$\dfrac{\dfrac{}{w : \exists x.Eq(x,\mathbf{0}) \wedge Eq(x,\mathbf{s}(\mathbf{0}))} \quad \dfrac{\dfrac{\mathcal{E}}{(Lt_\neg\ \mathbf{0}\ \mathbf{s}(\mathbf{0}))\ (\mathbf{Eq_t}\ a\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z) : \neg Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))} \quad \dfrac{\dfrac{}{\mathbf{Lt_s} : \forall x.Lt(x,\mathbf{s}(x))}\ Lt_s}{\mathbf{Lt_s}\ \mathbf{0} : Lt(\mathbf{0},\mathbf{s}(\mathbf{0}))}\ \forall \mathsf{E}}{(Lt_\neg\ \mathbf{0}\ \mathbf{s}(\mathbf{0}))\ (\mathbf{Eq_t}\ a\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z)\ (\mathbf{Lt_s}\ \mathbf{0}) : \bot}\ \neg \mathsf{E}}{\mathsf{let}\ \langle x,z\rangle = w\ \mathsf{in}\ (Lt_\neg\ \mathbf{0}\ \mathbf{s}(\mathbf{0}))\ (\mathbf{Eq_t}\ x\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z)\ (\mathbf{Lt_s}\ \mathbf{0}) : \bot}\ \exists \mathsf{E}^a}{\lambda w{:}\exists x.Eq(x,\mathbf{0}) \wedge Eq(x,\mathbf{s}(\mathbf{0}))).\,\mathsf{let}\ \langle x,z\rangle = w\ \mathsf{in}\ (Lt_\neg\ \mathbf{0}\ \mathbf{s}(\mathbf{0}))\ (\mathbf{Eq_t}\ x\ \mathbf{0}\ \mathbf{s}(\mathbf{0})\ z)\ (\mathbf{Lt_s}\ \mathbf{0}) : \neg\exists x.Eq(x,\mathbf{0}) \wedge Eq(x,\mathbf{s}(\mathbf{0}))}\ \neg \mathsf{I}^w$$

4.7 Sequent calculus

$$\dfrac{\Gamma, \forall x.A, [t/x]A \longrightarrow C}{\Gamma, \forall x.A \longrightarrow C}\ \forall L \qquad \dfrac{\Gamma \longrightarrow [a/x]A}{\Gamma \longrightarrow \forall x.A}\ \forall R^a$$

$$\dfrac{\Gamma, \exists x.A, [a/x]A \longrightarrow C}{\Gamma, \exists x.A \longrightarrow C}\ \exists L^a \qquad \dfrac{\Gamma \longrightarrow [t/x]A}{\Gamma \longrightarrow \exists x.A}\ \exists R$$

Todo. Proof of cut elimination.
Todo. Normalization

Corollary 4.1. *If $\cdot \vdash \exists x.A$ true, then there exists a term t such that $\cdot \vdash [t/x]A$.*

Examples

Axioms

In order to build a proof of $C\uparrow$ using the sequent calculus, we prove $\Gamma_{\mathsf{axiom}} \longrightarrow C$ where Γ_{axiom} contains all the axioms:

$$\begin{array}{rcl}\Gamma_{\mathsf{axiom}} & = & Nat(\mathbf{0}), \forall x.Nat(x) \supset Nat(\mathbf{s}(x)),\\ & & \forall x.Eq(x,x), \forall x.\forall y.\forall z.(Eq(x,y) \wedge Eq(x,z)) \supset Eq(y,z),\\ & & \forall x.Lt(x,\mathbf{s}(x)), \forall x.\forall y.Eq(x,y) \supset \neg Lt(x,y)\end{array}$$

Theorems

Proof of $\Gamma_{\mathsf{axiom}} \longrightarrow \forall x.\forall y.Eq(x,y) \supset Eq(y,x)$:
We let

$$\begin{array}{rcl}\Gamma'_{\mathsf{axiom}} & = & \Gamma_{\mathsf{axiom}},\\ & & \forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z),\\ & & \forall z.(Eq(a,b) \wedge Eq(a,z)) \supset Eq(b,z),\\ & & (Eq(a,b) \wedge Eq(a,a)) \supset Eq(b,a)\end{array}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma'_{\mathsf{axiom}}, Eq(a,b) \longrightarrow Eq(a,b)}\ Init \quad \dfrac{\dfrac{}{\Gamma'_{\mathsf{axiom}}, Eq(a,b), Eq(a,a) \longrightarrow Eq(a,a)}\ Init}{\Gamma'_{\mathsf{axiom}}, Eq(a,b) \longrightarrow Eq(a,a)}\ \forall L}{\Gamma'_{\mathsf{axiom}}, Eq(a,b) \longrightarrow Eq(a,b) \wedge Eq(a,a)}\ \wedge R \quad \dfrac{}{\Gamma'_{\mathsf{axiom}}, Eq(a,b), Eq(b,a) \longrightarrow Eq(b,a)}\ Init}{\Gamma'_{\mathsf{axiom}}, Eq(a,b) \longrightarrow Eq(b,a)}\ \supset L}{\Gamma_{\mathsf{axiom}}, \forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z), \forall z.(Eq(a,b) \wedge Eq(a,z)) \supset Eq(b,z), Eq(a,b) \longrightarrow Eq(b,a)}\ \forall L}{\Gamma_{\mathsf{axiom}}, \forall y.\forall z.(Eq(a,y) \wedge Eq(a,z)) \supset Eq(y,z), Eq(a,b) \longrightarrow Eq(b,a)}\ \forall L}{\Gamma_{\mathsf{axiom}}, Eq(a,b) \longrightarrow Eq(b,a)}\ \forall L}{\Gamma_{\mathsf{axiom}} \longrightarrow Eq(a,b) \supset Eq(b,a)}\ \supset R}{\Gamma_{\mathsf{axiom}} \longrightarrow \forall y.Eq(a,y) \supset Eq(y,a)}\ \forall R^b}{\Gamma_{\mathsf{axiom}} \longrightarrow \forall x.\forall y.Eq(x,y) \supset Eq(y,x)}\ \forall R^a$$

Chapter 5

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.

5.1 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.

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

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:

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

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

Local reductions of terms:

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

5.2 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.*

5.3 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:

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

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

5.4 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:

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

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:

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

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

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

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

5.5 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:

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

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

5.6 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:

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

5.7 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 $t < \mathbf{s}(t)\ \mathit{true}$:

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

5.8 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:

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

5.9 Examples

5.9.1 $(\forall x \in \tau . A(x) \wedge B(x)) \supset \forall x \in \tau . A(x)$

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

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

5.9.2 $\exists x \in \tau . A(x) \vee B(x) \equiv (\exists x \in \tau . A(x)) \vee (\exists x \in \tau . B(x))$

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 $((\exists x \in \tau . A(x)) \vee (\exists x \in \tau . B(x))) \supset (\exists x \in \tau . A(x) \vee B(x))$:

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

5.9.3 $\forall x \in \mathsf{nat} . x =_{\mathsf{N}} x$

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:

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

5.9.4 $\forall x \in \mathsf{nat}.\forall y \in \mathsf{nat}.\forall z \in \mathsf{nat}.x =_{\mathsf{N}} y \supset y =_{\mathsf{N}} z \supset x =_{\mathsf{N}} z$

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*:

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

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

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

A correct definition of proof term *trans*:

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

5.9.5 $\forall x \in \mathsf{nat}.\neg(x =_{\mathsf{N}} \mathbf{0}) \supset \exists y \in \mathsf{nat}.\mathbf{s}(y) =_{\mathsf{N}} x\ \mathit{true}$

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

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

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

A wrong definition of pred:

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

A corrected definition of pred:

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

Chapter 6

Classical Logic

Unlike constructive logic which is usually explained operationally by associating each proposition A with a proof proving or refuting its truth, classical logic is a logic that is usually explained *denotationally* by associating each proposition A with a truth value, either true T or false F. As there are only two truth values in classical logic, it is both sound and complete to check the truth value of a proposition A with a truth table in which all possible combinations of truth values of atomic propositions in A are considered in turn. For example, the connectives in classical propositional logic can be explained as follows:

A	B	$A \wedge B$	$A \vee B$	$A \supset B$
T	T	T	T	T
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

In this chapter, we investigate proof-theoretic formulations of classical logic based on inference rules. We also investigate its operational interpretation, or its computational contents, to obtain useful constructs for programming languages.

6.1 A judgmental formulation of classical logic

We have already seen in Chapter 1 that the addition of the following rule to constructive logic yields classical logic:

$$\frac{}{A \vee \neg A \ \mathit{true}}\ \mathsf{EM}$$

The rule EM, called *the law of excluded middle*, asserts that for any proposition A, either $A\ \mathit{true}$ or $\neg A\ \mathit{true}$ must hold regardless of the existence of an actual proof. Another way to obtain classical logic is to add one of the following rules:

$$\frac{}{\neg\neg A \supset A\ \mathit{true}}\ \mathsf{DNE} \qquad \frac{}{((A \supset B) \supset A) \supset A\ \mathit{true}}\ \mathsf{Pierce}$$

The rule DNE, called *the law of double-negation elimination*, asserts that if A cannot be false, it must be true. The rule Pierce, called *Pierce's law*, says that a proof of $A\ \mathit{true}$ may freely assume $A \supset B\ \mathit{true}$ for an arbitrary proposition B. The three rules above are all equivalent to each other in that the addition of any of these rules renders the other two rules derivable.

Note that these rules destroy the orthogonality of the system. For example, in the presence of the rule EM which uses two connectives $\supset$ and $\neg$ in the conclusion, the meaning of $\supset$ depends on the meaning of $\neg$ (or vice versa). The rule Pierce is also bad because it tries to explain the meaning of $\supset$ presupposing the notion of $\supset$. (As a rule of thumb, an inference rule using multiple connectives, whether same or different,

is always bad.) Instead of using these rules, therefore, we develop a system of classical logic based purely on judgmental notions.

We recall that at the heart of classical logic lies the principle of *proof by contradiction* (which is commonly employed in mathematical proofs). Using a truth judgment $A\ true$ and a falsehood judgment $A\ false$ (which denotes "*A is false*"), we can explain the idea as follows: in order to prove $A\ true$, we first assume $A\ false$ and then show a contraction, which is established by a proof of $B\ true$ when an assumption $B\ false$ is available for some proposition B. Observe that we may have to make an assumption of a falsehood judgment in the course of a proof, but we never directly conclude a falsehood judgment. Thus we use a new form of hypothetical judgment $\Gamma; \Delta \vdash_{\mathsf{K}} C\ true$ (where the subscript K stands for *K*lassical), in which antecedents may include falsehood judgments, but the succedent is always a truth judgment:

$$\Gamma; \Delta \vdash_{\mathsf{K}} C\ true \qquad \text{where} \qquad \begin{array}{rcl} \Gamma & ::= & \cdot \mid \Gamma, A\ true \\ \Delta & ::= & \cdot \mid \Delta, A\ false \end{array}$$

All previous rules from constructive propositional logic (which do not need falsehood judgments) are turned into corresponding rules for classical propositional logic by rewriting $\Gamma \vdash C\ true$ as $\Gamma; \Delta \vdash_{\mathsf{K}} C\ true$. The principle of proof by contradiction is implemented by the following two rules:

$$\frac{\Gamma; \Delta, A\ false \vdash_{\mathsf{K}} A\ true}{\Gamma; \Delta \vdash_{\mathsf{K}} A\ true}\ \mathsf{Contra}\uparrow \qquad \frac{\Gamma; \Delta, A\ false \vdash_{\mathsf{K}} A\ true}{\Gamma; \Delta, A\ false \vdash_{\mathsf{K}} C\ true}\ \mathsf{Contra}\downarrow$$

The rule Contra $\uparrow$ states that in order to prove $A\ true$, we may assume $A\ false$ and then show a contradiction by proving $A\ true$; hence it is best read in a bottom-up way (as indicated by the upward arrow in Contra $\uparrow$). The rule Contra $\downarrow$ states that if antecedents Γ and $\Delta, A\ false$ are inconsistent (because $A\ true$ is provable from Γ and $\Delta, A\ false$), we may conclude $C\ true$ for any proposition C as long as the same antecedents Γ and $\Delta, A\ false$ are available; hence it is best read in a top-down way (as indicated by the downward arrow in Contra $\downarrow$). Note that the premise is the same in both rules.

Here are a couple of examples showing that the rules EM and DNE are now derivable.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A\ true; A \vee \neg A\ false \vdash_{\mathsf{K}} A\ true}\ \mathsf{Hyp}}{A\ true; A \vee \neg A\ false \vdash_{\mathsf{K}} A \vee \neg A\ true}\ \vee\mathsf{I}_{\mathsf{L}}}{A\ true; A \vee \neg A\ false \vdash_{\mathsf{K}} \bot\ true}\ \mathsf{Contra}\downarrow}{\cdot; A \vee \neg A\ false \vdash_{\mathsf{K}} \neg A\ true}\ \neg\mathsf{I}}{\cdot; A \vee \neg A\ false \vdash_{\mathsf{K}} A \vee \neg A\ true}\ \vee\mathsf{I}_{\mathsf{R}}}{\cdot; \cdot \vdash_{\mathsf{K}} A \vee \neg A\ true}\ \mathsf{Contra}\uparrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\neg A\ true; A\ false \vdash_{\mathsf{K}} \neg\neg A\ true}\ \mathsf{Hyp} \quad \dfrac{\dfrac{\dfrac{}{\neg\neg A\ true, A\ true; A\ false \vdash_{\mathsf{K}} A\ true}\ \mathsf{Hyp}}{\neg\neg A\ true, A\ true; A\ false \vdash_{\mathsf{K}} \bot\ true}\ \mathsf{Contra}\downarrow}{\neg\neg A\ true; A\ false \vdash_{\mathsf{K}} \neg A\ true}\ \neg\mathsf{I}}{\neg\neg A\ true; A\ false \vdash_{\mathsf{K}} \bot\ true}\ \neg\mathsf{E}}{\neg\neg A\ true; A\ false \vdash_{\mathsf{K}} A\ true}\ \bot\mathsf{E}}{\neg\neg A\ true; \cdot \vdash_{\mathsf{K}} A\ true}\ \mathsf{Contra}\uparrow}{\cdot; \cdot \vdash_{\mathsf{K}} \neg\neg A \supset A\ true}\ \supset\mathsf{I}$$

6.2 Proof terms

We apply the Curry-Howard isomorphism to the new form of hypothetical judgment $\Gamma; \Delta \vdash_{\mathsf{K}} C\ true$ by first associating variables with judgments in Γ and Δ and then assigning a proof term of type C. For $A\ true$ in Γ, we bind a variable x to type A to obtain $x : A$. For $A\ false$ in Δ, we need a new notation to indicate that a variable x is associated with a falsehood judgment $A\ false$. We choose a new notation $x : A\ false$, in

which $A\ \mathit{false}$ can be thought of as a type instead of a judgment; in the context of type theory, $A\ \mathit{false}$ would stand for the type of *continuations* of type A.

$$\begin{array}{rcl} \Gamma & ::= & \cdot \mid \Gamma, x : A \\ \Delta & ::= & \cdot \mid \Delta, x : A\ \mathit{false} \end{array}$$

As proof terms corresponding to the rules Contra $\uparrow$ and Contra $\downarrow$, we use $\mathsf{callcc}\ x : A\ \mathit{false}.\ M$ and $\mathsf{throw}\ M\ \mathsf{to}\ x$ which are constructs for capturing and throwing continuations in programming languages:

$$\frac{\Gamma; \Delta, x : A\ \mathit{false} \vdash_{\mathsf{K}} M : A}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{callcc}\ x : A\ \mathit{false}.\ M : A}\ \textsf{Callcc} \qquad \frac{\Gamma; \Delta, x : A\ \mathit{false} \vdash_{\mathsf{K}} M : A}{\Gamma; \Delta, x : A\ \mathit{false} \vdash_{\mathsf{K}} \mathsf{throw}\ M\ \mathsf{to}\ x : C}\ \textsf{Throw}$$

In the rule Throw, proof term M is allowed to contain variable x.

A simple case of reducing a proof term $\mathsf{throw}\ M\ \mathsf{to}\ x$ occurs when M does not contain x:

$$\mathsf{callcc}\ x : A\ \mathit{false}.\ \sigma[\![\mathsf{throw}\ M\ \mathsf{to}\ x]\!] \quad \Longrightarrow_R \quad M$$

Here $\sigma[\![\mathsf{throw}\ M\ \mathsf{to}\ x]\!]$ denotes a certain proof term containing $\mathsf{throw}\ M\ \mathsf{to}\ x$ as a subterm. By the rule Callcc, it has type A so that the whole proof term $\mathsf{callcc}\ x : A\ \mathit{false}.\ \sigma[\![\mathsf{throw}\ M\ \mathsf{to}\ x]\!]$ is assigned type A. By the rule Throw, proof term M also has type A, and thus the type of the proof term being reduced is preserved. For a full account of reductions of $\mathsf{callcc}\ x : A\ \mathit{false}.\ M$ and $\mathsf{throw}\ M\ \mathsf{to}\ x$, we need to formalize the definition of σ, which we do not pursue here.

A proof term LEM of type $A \vee \neg A$ is given as follows:

$$\mathit{LEM} \quad = \quad \mathsf{callcc}\ x : A \vee \neg A\ \mathit{false}.\ \mathsf{inr}_A\ \lambda y{:}A.\ \mathsf{throw}\ \mathsf{inl}_{\neg A}\ y\ \mathsf{to}\ x$$

A proof term DNE of type $\neg\neg A \supset A$ is given as follows:

$$\mathit{DNE} \quad = \quad \lambda x{:}\neg\neg A.\ \mathsf{callcc}\ y : A\ \mathit{false}.\ \mathsf{abort}_A\ (x\ (\lambda z{:}A.\ \mathsf{throw}\ z\ \mathsf{to}\ y))$$

6.3 Sequent calculus for classical logic

The sequent calculus for classical logic uses a new form of sequent whose definition is motivated by the principle of proof by contradiction, rather than the notion of normal proof as in constructive logic. We write $A_1, \cdots, A_n \Longrightarrow B_1, \cdots, B_m$ to mean that assumptions of $A_1\ \mathit{true}$, $\cdots$, $A_n\ \mathit{true}$ and $B_1\ \mathit{false}$, $\cdots$, $B_m\ \mathit{false}$ lead to a contradiction. Note that unlike a sequent $\Gamma \longrightarrow C$ for constructive logic, the new sequent allows multiple propositions in the right side. We use Γ for a collection of propositions in the left side and Δ for a collection of propositions in the right side (*e.g.*, $\Gamma \Longrightarrow \Delta$). We assume that Γ and Δ are unordered.

The definition of the new form of sequent justifies the following rule, which is similar to the rule Init in constructive logic but actually expresses the principle of proof by contradiction:

$$\frac{}{\Gamma, A \Longrightarrow A, \Delta}\ \mathit{Contra}$$

The rule Contra says that since assumptions of $A\ \mathit{true}$ and $A\ \mathit{false}$ lead to a contradiction, the proof of $\Gamma, A \Longrightarrow A, \Delta$ is completed immediately.

As in the sequent calculus for constructive logic, we give left and right rules for each connective. Although these rules appear to be mechanically derived from their corresponding rules in the sequent calculus for constructive logic, their interpretation is different because the definition of $\Gamma \Longrightarrow \Delta$ is motivated differently from the definition of $\Gamma \longrightarrow C$. As each rule focuses on a proposition in the left or right side in a given sequent, we choose to reuse the rule names from the sequent calculus for constructive logic.

Figure 6.1 shows all the rules in the sequent calculus for classical propositional logic. As in the sequent calculus for constructive logic, the proof of a sequent always proceeds in a bottom-up way. There are two important observations to make. First, unlike in the sequent calculus for constructive logic, the right side

$$\frac{}{\Gamma, A \Longrightarrow A, \Delta}\ \mathit{Contra}$$

$$\frac{\Gamma, A \wedge B, A \Longrightarrow \Delta}{\Gamma, A \wedge B \Longrightarrow \Delta}\ \wedge L_L \qquad \frac{\Gamma, A \wedge B, B \Longrightarrow \Delta}{\Gamma, A \wedge B \Longrightarrow \Delta}\ \wedge L_R \qquad \frac{\Gamma \Longrightarrow A, A \wedge B, \Delta \quad \Gamma \Longrightarrow B, A \wedge B, \Delta}{\Gamma \Longrightarrow A \wedge B, \Delta}\ \wedge R$$

$$\frac{\Gamma, A \vee B, A \Longrightarrow \Delta \quad \Gamma, A \vee B, B \Longrightarrow \Delta}{\Gamma, A \vee B \Longrightarrow \Delta}\ \vee L \qquad \frac{\Gamma \Longrightarrow A, A \vee B, \Delta}{\Gamma \Longrightarrow A \vee B, \Delta}\ \vee R_L \qquad \frac{\Gamma \Longrightarrow B, A \vee B, \Delta}{\Gamma \Longrightarrow A \vee B, \Delta}\ \vee R_R$$

$$\frac{}{\Gamma \Longrightarrow \top, \Delta}\ \top R \qquad \frac{}{\Gamma, \bot \Longrightarrow \Delta}\ \bot L \qquad \frac{\Gamma, \neg A \Longrightarrow A, \Delta}{\Gamma, \neg A \Longrightarrow \Delta}\ \neg L \qquad \frac{\Gamma, A \Longrightarrow \neg A, \Delta}{\Gamma \Longrightarrow \neg A, \Delta}\ \neg R$$

$$\frac{\Gamma, A \supset B \Longrightarrow A, \Delta \quad \Gamma, A \supset B, B \Longrightarrow \Delta}{\Gamma, A \supset B \Longrightarrow \Delta}\ \supset L \qquad \frac{\Gamma, A \Longrightarrow B, A \supset B, \Delta}{\Gamma \Longrightarrow A \supset B, \Delta}\ \supset R$$

Figure 6.1: Sequent calculus for classical propositional logic

of a sequent should *not* be read as a collection of conclusions to be drawn; rather it should be read as a collection of assumptions (consisting of falsehood judgments). Second the premise in a rule (especially in a right rule) always contains more assumptions than the conclusion: we never cancel an existing assumption because the goal is to elicit a contradiction from a collection of assumptions.

The left rules $\wedge L_L$, $\wedge L_R$, $\vee L$ can be read in the same manner as their counterparts in the sequent calculus for constructive logic. The right rules, however, cannot be read in the same manner because the right side of a sequent should be read not as a collection of conclusions but as a collection of assumptions. In the rule $\wedge R$, for example, we use an assumption $A \wedge B$ *false* to yield a contradiction, rather than deduce a conclusion $A \wedge B$ *false*. Since $A \wedge B$ *false* holds when either A *false* or B *false* holds, we have to show a contradiction in each case. (Using the right side as a collection of conclusions would make it difficult, or even impossible, to make sense of the rule $\wedge R$.) The right rules $\vee R_L$ and $\vee R_R$ show that we never cancel an existing assumption.

The rule $\top R$ makes sense because an assumption $\top$ *false* expresses a contradiction: $\top$ cannot be false. Likewise the rule $\bot L$ makes sense because $\bot$ cannot be true. The rule $\neg L$ states that assuming $\neg A$ *true* is equivalent to assuming A *false*; similarly the rule $\neg R$ states that assuming $\neg A$ *false* is equivalent to assuming A *true*. (Here we do not use the notational definition of $\neg A$ as $A \supset \bot$.)

Although Figure 6.1 includes the rules $\supset L$ and $\supset R$, these rules are in fact derived rules because implication is a derived notion: classical logic has no notion of transforming a proof into another because every proposition denotes just a truth value, and thus $A \supset B$ is *defined* as $\neg A \vee B$. Lack of the notion of implication in classical logic is also the reason why $A \supset B$ is assigned a truth value T whenever A is assigned a truth value F.

The sequent calculus in Figure 6.1 satisfies the weakening and contraction properties which allow us to use a proposition A in Γ or Δ as many times as necessary in a proof of $\Gamma \Longrightarrow \Delta$. It also satisfies the subformula property in the same sense as in the sequent calculus for constructive logic, which in turn implies that the sequent calculus is decidable.

Proposition 6.1 (Structural properties).

(Weakening) *If* $\Gamma \Longrightarrow \Delta$, *then* $\Gamma, A \Longrightarrow \Delta$.
If $\Gamma \Longrightarrow \Delta$, *then* $\Gamma \Longrightarrow A, \Delta$.

(Contraction) *If* $\Gamma, A, A \Longrightarrow \Delta$, *then* $\Gamma, A \Longrightarrow \Delta$.
If $\Gamma \Longrightarrow A, A, \Delta$, *then* $\Gamma \Longrightarrow A, \Delta$.

Proof. By induction on the structure of the proof of $\Gamma \Longrightarrow \Delta$ and $\Gamma, A, A \Longrightarrow \Delta$ and $\Gamma \Longrightarrow A, A, \Delta$. □

As in the sequent calculus for constructive logic, there is a cut rule whose admissibility implies that the sequent calculus is sound (*i.e.*, $\cdot \Longrightarrow \bot$ is not provable). Interestingly it expresses precisely the law of excluded middle:

$$\frac{\Gamma \Longrightarrow A, \Delta \quad \Gamma, A \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta}\ \mathit{Cut}$$

The two premises cover all possibilities because for every proposition A, either $A\ true$ or $A\ false$ must hold by the law of excluded middle. Hence the provability of the premises implies that a contradiction can always be reached whenever assumptions Γ and Δ are available. It turns out that the rule Cut is indeed admissible, and we can conclude that the law of excluded middle is built into the sequent calculus.

There is another way of interpreting a sequent $\Gamma \Longrightarrow \Delta$, which is called the *multi-conclusion view*. Under the multi-conclusion view, $A_1, \cdots, A_n \Longrightarrow B_1, \cdots, B_m$ means that if assumptions $A_1\ true$, $\cdots$, "and" $A_n\ true$ are available, a conclusion $B_1\ true$, $\cdots$, "or" $B_m\ true$ is provable. While all the rules in Figure 6.1 continue to make sense, they fail to express the essence of classical logic, namely the principle of proof by contradiction (or the law of excluded middle).

6.4 Double-negation translation and CPS transformation

We have seen that classical logic is obtained by augmenting constructive logic with two new rules $\mathsf{Contra}\uparrow$ and $\mathsf{Contra}\downarrow$. As it comes with more inference rules, classical logic allows us to prove more truth judgments than constructive logic. That is, a proof of $A\ true$ in constructive logic is a valid proof in classical logic as well, and therefore, if $A\ true$ is provable in constructive logic, it is also provable in classical logic. The converse is certainly untrue, as evidenced by such judgments as $A \vee \neg A\ true$, $\neg\neg A \supset A\ true$, and $((A \supset B) \supset A) \supset A\ true$.

It is important that more judgments being provable does not necessarily mean more expressive power. For example, a system in which $\bot\ true$ is provable is totally useless (and is said to be inconsistent), even though every truth judgment is provable. In the case of classical logic, it allows more judgments to be provable, but is *less* expressive than constructive logic, which is capable of simulating classical logic via the *double-negation translation* to be explained below. In essence, constructive logic can make a finer distinction between truth and falsehood than classical logic (as it allows not only truth and falsehood but also "excluded middle").

We write A° for the proposition in constructive logic corresponding to proposition A in classical logic under the double-negation translation. The translation is structural except for $A \supset B$, which is translated to $A^\circ \supset \neg\neg B^\circ$:

$$\begin{array}{rcl} (A \wedge B)^\circ & = & A^\circ \wedge B^\circ \\ (A \vee B)^\circ & = & A^\circ \vee B^\circ \\ \top^\circ & = & \top \\ \bot^\circ & = & \bot \\ (A \supset B)^\circ & = & A^\circ \supset \neg\neg B^\circ \\ P^\circ & = & P \end{array}$$

An analogy in programming language theory is that an invocation of a "classical" function of type $A \supset B$ is in effect an invocation of a "constructive" function of type $A \supset \neg\neg B = A \supset ((B \supset \bot) \supset \bot)$, which returns an answer (of type $\bot$) when given an argument of type A and a return address (of type $B \supset \bot$) expecting an argument of type B.

Theorem 6.2 shows that classical logic is embedded in constructive logic via the double-negation translation, where we use the subscript I in a hypothetical judgment to indicate that it is valid only in *I*ntuitionistic logic, which is another name for constructive logic. Antecedents in hypothetical judgments are translated as follows:

$$\Gamma^\circ = \{A^\circ\ true \mid A\ true \in \Gamma\}$$
$$\neg\Delta^\circ = \{\neg A^\circ\ true \mid A\ false \in \Delta\}$$

Theorem 6.2 (Embedding of classical logic in constructive logic). *If* $\Gamma; \Delta \vdash_{\mathsf{K}} C\ true$, *then* $\Gamma^\circ, \neg\Delta^\circ \vdash_{\mathsf{I}} \neg\neg C^\circ\ true$.

An immediate corollary of Theorem 6.2 is that classical logic is relatively consistent with constructive logic in the sense that classical logic is consistent (*i.e.*, $\bot\ true$ is not provable) if and only if constructive logic is consistent, since $\neg\neg\bot$ is logically equivalent to $\bot$. As we have shown that constructive logic is consistent (*i.e.*, $\bot\ true$ is unprovable), we conclude that classical logic is also consistent.

Instead of proving Theorem 6.2 directly, we give another translation that converts a proof term M of type A in classical logic into a proof term M° of type A° in constructive logic. The translation is usually called *the CPS (Continuation-Passing Style) translation*, which enables us to simulate $\mathsf{callcc}\ x : A\ \mathit{false}.\, M$ and $\mathsf{throw}\ M\ \mathsf{to}\ x$ with λ-abstractions. The main idea in the CPS translation is to interpret $\neg A = A \supset \bot$ as the type of a *continuation* for type A, which, when invoked with an argument of type A, initiates the rest of the evaluation. Note that an invocation of a continuation conceptually returns an "answer" but actually never returns, for if it did, it would return a value of type $\bot$, which is impossible.

The CPS translation is obtained by translating a proof of $\Gamma; \Delta \vdash_{\mathsf{K}} M : C$ to a proof of $\Gamma^\circ, \neg\Delta^\circ \vdash_{\mathsf{I}} M^\circ : \neg\neg C^\circ$ where Γ° and $\neg\Delta^\circ$ are defined as follows:

$$\begin{aligned} \Gamma^\circ &= \{x : A^\circ \mid x : A \in \Gamma\} \\ \neg\Delta^\circ &= \{x : \neg A^\circ \mid x : A\ \mathit{false} \in \Delta\} \end{aligned}$$

Theorem 6.3 (CPS translation). *If $\Gamma; \Delta \vdash_{\mathsf{K}} M : C$, there exists a proof term M° such that $\Gamma^\circ, \neg\Delta^\circ \vdash_{\mathsf{I}} M^\circ : \neg\neg C^\circ$.*

Proof. By induction on the structure of the proof of $\Gamma; \Delta \vdash_{\mathsf{K}} M : C$. The proof reuses metavariables M and C.

In each case, we only specify M°, which can be shown to satisfy $\Gamma^\circ, \neg\Delta^\circ \vdash_{\mathsf{I}} M^\circ : \neg\neg C^\circ$ by straightforward structural induction. M° is given as a λ-abstraction $\lambda k{:}C^\circ \supset \bot.\, \cdots$ (or equivalently $\lambda k{:}\neg C^\circ.\, \cdots$), where k can be thought of as a continuation expecting the result of evaluating M. A typical pattern in the CPS translation is that a proof term of type $\bot$ (returning an "answer") is built from a proof term N of type A by applying M° to a continuation $\lambda x{:}A^\circ.\, N'$ (as in $M^\circ\ (\lambda x{:}A^\circ.\, N')$) so that x is bound to the result of evaluating M and the evaluation of N' returns the final "answer."

Case $\dfrac{x : A \in \Gamma}{\Gamma; \Delta \vdash_{\mathsf{K}} x : A}\ \mathsf{Hyp}$

$x^\circ = \lambda k{:}A^\circ \supset \bot.\, k\ x$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A \quad \Gamma; \Delta \vdash_{\mathsf{K}} N : B}{\Gamma; \Delta \vdash_{\mathsf{K}} (M, N) : A \wedge B}\ \wedge\mathsf{I}$

$(M, N)^\circ = \lambda k{:}(A^\circ \wedge B^\circ) \supset \bot.\, M^\circ\ (\lambda x{:}A^\circ.\, N^\circ\ (\lambda y{:}B^\circ.\, k\ (x, y)))$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A \wedge B}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{fst}\ M : A}\ \wedge\mathsf{E}_{\mathsf{L}}$

$(\mathsf{fst}\ M)^\circ = \lambda k{:}A^\circ \supset \bot.\, M^\circ\ (\lambda x{:}A^\circ \wedge B^\circ.\, k\ (\mathsf{fst}\ x))$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A \wedge B}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{snd}\ M : B}\ \wedge\mathsf{E}_{\mathsf{R}}$

$(\mathsf{snd}\ M)^\circ = \lambda k{:}B^\circ \supset \bot.\, M^\circ\ (\lambda x{:}A^\circ \wedge B^\circ.\, k\ (\mathsf{snd}\ x))$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{inl}_B\ M : A \vee B}\ \vee\mathsf{I}_{\mathsf{L}}$

$(\mathsf{inl}_B\ M)^\circ = \lambda k{:}(A^\circ \vee B^\circ) \supset \bot.\, M^\circ\ (\lambda x{:}A^\circ.\, k\ (\mathsf{inl}_{B^\circ}\ x))$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : B}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{inr}_A\ M : A \vee B}\ \vee\mathsf{I}_{\mathsf{R}}$

$(\mathsf{inr}_A\ M)^\circ = \lambda k{:}(A^\circ \vee B^\circ) \supset \bot.\, M^\circ\ (\lambda x{:}B^\circ.\, k\ (\mathsf{inr}_{A^\circ}\ x))$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A \vee B \quad \Gamma, x_1 : A; \Delta \vdash_{\mathsf{K}} N_1 : C \quad \Gamma, x_2 : B; \Delta \vdash_{\mathsf{K}} N_2 : C}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x_1 \Rightarrow N_1 \mid \mathsf{inr}\ x_2 \Rightarrow N_2 : C}\ \vee\mathsf{I}_{\mathsf{R}}$

$(\mathsf{case}\ M\ \mathsf{of}\ \mathsf{inl}\ x_1 \Rightarrow N_1 \mid \mathsf{inr}\ x_2 \Rightarrow N_2)^\circ = \lambda k{:}C^\circ \supset \bot.\, M^\circ\ (\lambda x{:}A^\circ \vee B^\circ.\, \mathsf{case}\ x\ \mathsf{of}\ \mathsf{inl}\ x_1 \Rightarrow {N_1}^\circ\ k \mid \mathsf{inr}\ x_2 \Rightarrow {N_2}^\circ\ k)$

Case $\dfrac{\Gamma, x : A; \Delta \vdash_{\mathsf{K}} M : B}{\Gamma; \Delta \vdash_{\mathsf{K}} \lambda x{:}A.\, M : A \supset B}\ \supset\mathsf{I}$

$(\lambda x{:}A.\, M)^\circ = \lambda k{:}(A^\circ \supset \neg\neg B^\circ) \supset \bot.\, k\ (\lambda x{:}A^\circ.\, M^\circ)$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A \supset B \quad \Gamma; \Delta \vdash_{\mathsf{K}} N : A}{\Gamma; \Delta \vdash_{\mathsf{K}} M\ N : B}$

$(M\ N)^\circ = \lambda k{:}B^\circ \supset \bot.\, M^\circ\ (\lambda x{:}A^\circ \supset \neg\neg B^\circ.\, N^\circ\ (\lambda y{:}A^\circ.\, x\ y\ k))$

Case $\dfrac{}{\Gamma; \Delta \vdash_{\mathsf{K}} \langle\rangle : \top}$

$\langle\rangle^\circ = \lambda k{:}\top^\circ \supset \bot.\, k\ \langle\rangle$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : \bot}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{abort}_C\ M : C}$

$(\mathsf{abort}_C\ M)^\circ = \lambda k{:}C^\circ \supset \bot.\, M^\circ\ (\lambda x{:}\bot.\, x)$

Case $\dfrac{\Gamma; \Delta, x : A\ \mathit{false} \vdash_{\mathsf{K}} M : A}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{callcc}\ x : A\ \mathit{false}.\, M : A}$

$(\mathsf{callcc}\ x : A\ \mathit{false}.\, M)^\circ = \lambda k{:}A^\circ \supset \bot.\, [k/x]M^\circ\ k$

Case $\dfrac{\Gamma; \Delta \vdash_{\mathsf{K}} M : A \quad x : A\ \mathit{false} \in \Delta}{\Gamma; \Delta \vdash_{\mathsf{K}} \mathsf{throw}\ M\ \mathsf{to}\ x : C}$

$(\mathsf{throw}\ M\ \mathsf{to}\ x)^\circ = \lambda k{:}C^\circ \supset \bot.\, M^\circ\ x$ □