Lecture Notes on
Types as Propositions

15-814: Types and Programming Languages
Frank Pfenning

Lecture 17
Thursday, November 4, 2021

1 Introduction

> *These lecture notes are pieced together from several lectures in an undergraduate course on* Constructive Logic*, so they are a bit more extensive than what we discussed in the lecture.*

2 Natural Deduction

The goal of this section is to develop the two principal notions of logic, namely *propositions* and *proofs*. There is no universal agreement about the proper foundations for these notions. One approach, which has been particularly successful for applications in computer science, is to understand the meaning of a proposition by understanding its proofs. In the words of Martin-Löf [ML96, Page 27]:

> *The meaning of a proposition is determined by [. . .] what counts as a verification of it.*

A *verification* may be understood as a certain kind of proof that only examines the constituents of a proposition. This is analyzed in greater detail by Dummett [Dum91] although with less direct connection to computer science. The system of inference rules that arises from this point of view is *natural deduction*, first proposed by Gentzen [Gen35] and studied in depth by Prawitz [Pra65].

In this chapter we apply Martin-Löf's approach, which follows a rich philosophical tradition, to explain the basic propositional connectives.

We will define the meaning of the usual connectives of propositional logic (conjunction, implication, disjunction) by rules that allow us to infer when they should be true, so-called *introduction rules*. From these, we derive rules for the use of propositions, so-called *elimination rules*. The resulting system of *natural deduction* is the foundation of intuitionistic logic which has direct connections to functional programming and logic programming.

3 Judgments and Propositions

The cornerstone of Martin-Löf's foundation of logic is a clear separation of the notions of judgment and proposition. A *judgment* is something we may know, that is, an object of knowledge. A judgment is *evident* if we in fact know it.

We make a judgment such as "*it is raining*", because we have evidence for it. In everyday life, such evidence is often immediate: we may look out the window and see that it is raining. In logic, we are concerned with situation where the evidence is indirect: we deduce the judgment by making correct inferences from other evident judgments. In other words: a judgment is evident if we have a proof for it.

The most important judgment form in logic is "A *is true*", where A is a proposition. There are many others that have been studied extensively. For example, "A *is false*", "A *is true at time* t" (from temporal logic), "A *is necessarily true*" (from modal logic), "*program* M *has type* τ" (from programming languages), etc.

Returning to the first judgment, let us try to explain the meaning of conjunction. We write A *true* for the judgment "A *is true*" (presupposing that A is a proposition. Given propositions A and B, we can form the compound proposition "A *and* B", written more formally as $A \wedge B$. But we have not yet specified what conjunction *means*, that is, what counts as a verification of $A \wedge B$. This is accomplished by the following inference rule:

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

Here the name $\wedge I$ stands for "conjunction introduction", since the conjunction is introduced in the conclusion.

This rule allows us to conclude that $A \wedge B$ *true* if we already know that A *true* and B *true*. In this inference rule, A and B are *schematic variables*, and $\wedge I$ is the name of the rule. Intuitively, the $\wedge I$ rule says that a proof of $A \wedge B$ *true* consists of a proof of A *true* together with a proof of B *true*.

The general form of an inference rule is

$$\frac{J_1 \ \ldots \ J_n}{J} \ \textit{name}$$

where the judgments $J_1, \ldots, J_n$ are called the *premises*, the judgment J is called the *conclusion*. In general, we will use letters J to stand for judgments, while A, B, and C are reserved for propositions.

We take conjunction introduction as specifying the meaning of $A \wedge B$ completely. So what can be deduced if we know that $A \wedge B$ is true? By the above rule, to have a verification for $A \wedge B$ means to have verifications for A and B. Hence the following two rules are justified:

$$\frac{A \wedge B \ \textit{true}}{A \ \textit{true}} \ \wedge E_1 \qquad\qquad \frac{A \wedge B \ \textit{true}}{B \ \textit{true}} \ \wedge E_2$$

The name $\wedge E_1$ stands for "first/left conjunction elimination", since the conjunction in the premise has been eliminated in the conclusion. Similarly $\wedge E_2$ stands for "second/right conjunction elimination". Intuitively, the $\wedge E_1$ rule says that A *true* follows if we have a proof of $A \wedge B$ *true*, because "we must have had a proof of A *true* to justify $A \wedge B$ *true*".

We will later see what precisely is required in order to guarantee that the formation, introduction, and elimination rules for a connective fit together correctly. For now, we will informally argue the correctness of the elimination rules, as we did for the conjunction elimination rules.

As a second example we consider the proposition "*truth*" written as $\top$. Truth should always be true, which means its introduction rule has no premises.

$$\frac{}{\top \ \textit{true}} \ \top I$$

Consequently, we have no information if we know $\top$ *true*, so there is no elimination rule.

A conjunction of two propositions is characterized by one introduction rule with two premises, and two corresponding elimination rules. We may think of truth as a conjunction of zero propositions. By analogy it should then have one introduction rule with zero premises, and zero corresponding elimination rules. This is precisely what we wrote out above.

4 Hypothetical Judgments

Consider the following derivation, for arbitrary propositions A, B, and C:

$$\dfrac{\dfrac{A \wedge (B \wedge C)\ \mathit{true}}{B \wedge C\ \mathit{true}}\ \wedge E_2}{B\ \mathit{true}}\ \wedge E_1$$

Have we actually proved anything here? At first glance it seems that cannot be the case: B is an arbitrary proposition; clearly we should not be able to prove that it is true. Upon closer inspection we see that all inferences are correct, but the first judgment $A \wedge (B \wedge C)$ *true* has not been justified. We can extract the following knowledge:

> *From the assumption that $A \wedge (B \wedge C)$ is true, we deduce that B must be true.*

This is an example of a *hypothetical judgment*, and the figure above is an *hypothetical deduction*. In general, we may have more than one assumption, so a hypothetical deduction has the form

$$\begin{array}{ccc} J_1 & \cdots & J_n \\ & \vdots & \\ & J & \end{array}$$

where the judgments $J_1, \ldots, J_n$ are unproven assumptions, and the judgment J is the conclusion. All instances of the inference rules are hypothetical judgments as well (albeit possibly with 0 assumptions if the inference rule has no premises).

Many mistakes in reasoning arise because dependencies on some hidden assumptions are ignored. When we need to be explicit, we will write $J_1, \ldots, J_n \vdash J$ for the hypothetical judgment which is established by the hypothetical deduction above. We may refer to $J_1, \ldots, J_n$ as the antecedents and J as the succedent of the hypothetical judgment. For example, the hypothetical judgment $A \wedge (B \wedge C)$ *true* $\vdash B$ *true* is proved by the above hypothetical deduction that B *true* indeed follows from the hypothesis $A \wedge (B \wedge C)$ *true* using inference rules.

Substitution Principle for Hypotheses: We can always substitute a proof for any hypothesis J_i to eliminate the assumption. Into the above hypothetical deduction, a proof of its hypothesis J_i

$$\begin{array}{ccc} K_1 & \cdots & K_m \\ & \vdots & \\ & J_i & \end{array}$$

can be substituted in for J_i to obtain the hypothetical deduction

$$\begin{array}{ccccc} & & K_1 \;\cdots\; K_m & & \\ & & \vdots & & \\ J_1 & \cdots & J_i & \cdots & J_n \\ & & \vdots & & \\ & & J & & \end{array}$$

This hypothetical deduction concludes J from the unproven assumptions $J_1, \ldots, J_{i-1}, K_1, \ldots, K_m, J_{i+1}, \ldots, J_n$ and justifies the hypothetical judgment

$$J_1, \ldots, J_{i-1}, K_1, \ldots, K_m, J_{i+1}, \ldots, J_n \vdash J$$

That is, into the hypothetical judgment $J_1, \ldots, J_n \vdash J$, we can always substitute a derivation of the judgment J_i that was used as a hypothesis to obtain a derivation which no longer depends on the assumption J_i. A hypothetical deduction with 0 assumptions is a *proof* of its conclusion J.

One has to keep in mind that hypotheses may be used more than once, or not at all. For example, for arbitrary propositions A and B,

$$\dfrac{\dfrac{A \wedge B\ \mathit{true}}{B\ \mathit{true}}\ \wedge E_2 \qquad \dfrac{A \wedge B\ \mathit{true}}{A\ \mathit{true}}\ \wedge E_1}{B \wedge A\ \mathit{true}}\ \wedge I$$

can be seen a hypothetical derivation of $A \wedge B$ *true* $\vdash B \wedge A$ *true*. Similarly, a minor variation of the first proof in this section is a hypothetical derivation for the hypothetical judgment $A \wedge (B \wedge C)$ *true* $\vdash B \wedge A$ *true* that uses the hypothesis twice.

With hypothetical judgments, we can now explain the meaning of implication "A *implies* B" or "*if* A *then* B" (more formally: $A \supset B$). The introduction rule reads: $A \supset B$ is true, if B is true under the assumption that A is true.

$$\dfrac{\begin{array}{c}\overline{A\ \mathit{true}}\ u \\ \vdots \\ B\ \mathit{true}\end{array}}{A \supset B\ \mathit{true}}\ \supset I^u$$

The tricky part of this rule is the label u and its bar. If we omit this annotation, the rule would read

$$\dfrac{\begin{array}{c}A\ \mathit{true} \\ \vdots \\ B\ \mathit{true}\end{array}}{A \supset B\ \mathit{true}}\ \supset I$$

which would be incorrect: it looks like a derivation of $A \supset B$ *true* from the hypothesis A *true*. But the assumption A *true* is introduced in the process of proving $A \supset B$ *true*; the conclusion should not depend on it! Certainly, whether the implication $A \supset B$ is true is independent of the question whether A itself is actually true. Therefore we label uses of the assumption with a new name u, and the corresponding inference which introduced this assumption into the derivation with the same label u.

The rule makes intuitive sense, a proof justifying $A \supset B$ *true* assumes, hypothetically, the left-hand side of the implication so that A *true*, and uses this to show the right-hand side of the implication by proving B *true*. The proof of $A \supset B$ *true* constructs a proof of B *true* from the additional assumption that A *true*.

As a concrete example, consider the following proof of $A \supset (B \supset (A \wedge B))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A\ \mathit{true}}\,u \quad \dfrac{}{B\ \mathit{true}}\,w}{A \wedge B\ \mathit{true}}\,\wedge I}{B \supset (A \wedge B)\ \mathit{true}}\,{\supset} I^w}{A \supset (B \supset (A \wedge B))\ \mathit{true}}\,{\supset} I^u$$

Note that this derivation is not hypothetical (it does not depend on any assumptions). The assumption A *true* labeled u is discharged in the last inference, and the assumption B *true* labeled w is discharged in the second-to-last inference. It is critical that a discharged hypothesis is no longer available for reasoning, and that all labels introduced in a derivation are distinct.

Finally, we consider what the elimination rule for implication should say. By the only introduction rule, having a proof of $A \supset B$ *true* means that we have a hypothetical proof of B *true* from A *true*. By the substitution principle, if we also have a proof of A *true* then we get a proof of B *true*.

$$\dfrac{A \supset B\ \mathit{true} \quad A\ \mathit{true}}{B\ \mathit{true}}\,{\supset} E$$

This completes the rules concerning implication.

With the rules so far, we can write out proofs of simple properties concerning conjunction and implication. The first expresses that conjunction is commutative—intuitively, an obvious property.

$$\dfrac{\dfrac{\dfrac{}{A \wedge B\ \mathit{true}}\,u}{B\ \mathit{true}}\,{\wedge}E_2 \quad \dfrac{\dfrac{}{A \wedge B\ \mathit{true}}\,u}{A\ \mathit{true}}\,{\wedge}E_1}{\dfrac{B \wedge A\ \mathit{true}}{(A \wedge B) \supset (B \wedge A)\ \mathit{true}}\,{\supset}I^u}\,{\wedge}I$$

When we construct such a derivation, we generally proceed by a combination of bottom-up and top-down reasoning. The next example is a distributivity law, allowing us to move implications over conjunctions. This time, we show the partial proofs in each step. Of course, other sequences of steps in proof constructions are also possible.

$$\begin{array}{c} \vdots \\ (A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true} \end{array}$$

First, we use the implication introduction rule bottom-up.

$$\dfrac{\begin{array}{c} \dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \\ \vdots \\ (A \supset B) \wedge (A \supset C)\ \mathit{true} \end{array}}{(A \supset (B \wedge C) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true}}\,{\supset}I^u$$

Next, we use the conjunction introduction rule bottom-up, copying the available assumptions to both branches in the scope.

$$\dfrac{\dfrac{\begin{array}{c} \dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \\ \vdots \\ A \supset B\ \mathit{true} \end{array} \quad \begin{array}{c} \dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \\ \vdots \\ A \supset C\ \mathit{true} \end{array}}{(A \supset B) \wedge (A \supset C)\ \mathit{true}}\,{\wedge}I}{(A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true}}\,{\supset}I^u$$

We now pursue the left branch, again using implication introduction bottom-up.

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \quad \dfrac{}{A\ \mathit{true}}\,w \\ \vdots \\ B\ \mathit{true}\end{array}}{A \supset B\ \mathit{true}}\,{\supset}I^w \qquad \begin{array}{c}\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \\ \vdots \\ A \supset C\ \mathit{true}\end{array}}{(A \supset B) \wedge (A \supset C)\ \mathit{true}}\,{\wedge}I}{(A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true}}\,{\supset}I^u$$

Note that the hypothesis A *true* is available only in the left branch and not in the right one: it is discharged at the inference $\supset I^w$. We now switch to top-down reasoning, taking advantage of implication elimination.

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \quad \dfrac{}{A\ \mathit{true}}\,w}{B \wedge C\ \mathit{true}}\,{\supset}E \\ \vdots \\ B\ \mathit{true}\end{array}}{A \supset B\ \mathit{true}}\,{\supset}I^w \qquad \begin{array}{c}\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \\ \vdots \\ A \supset C\ \mathit{true}\end{array}}{(A \supset B) \wedge (A \supset C)\ \mathit{true}}\,{\wedge}I}{(A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true}}\,{\supset}I^u$$

Now we can close the gap in the left-hand side by conjunction elimination.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \quad \dfrac{}{A\ \mathit{true}}\,w}{B \wedge C\ \mathit{true}}\,{\supset}E}{B\ \mathit{true}}\,{\wedge}E_1}{A \supset B\ \mathit{true}}\,{\supset}I^w \qquad \begin{array}{c}\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \\ \vdots \\ A \supset C\ \mathit{true}\end{array}}{(A \supset B) \wedge (A \supset C)\ \mathit{true}}\,{\wedge}I}{(A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true}}\,{\supset}I^u$$

The right premise of the conjunction introduction can be filled in analogously. We skip the intermediate steps and only show the final derivation.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \quad \dfrac{}{A\ \mathit{true}}\,w}{B \wedge C\ \mathit{true}}\,{\supset}E}{B\ \mathit{true}}\,{\wedge}E_1}{A \supset B\ \mathit{true}}\,{\supset}I^w \quad \dfrac{\dfrac{\dfrac{\dfrac{}{A \supset (B \wedge C)\ \mathit{true}}\,u \quad \dfrac{}{A\ \mathit{true}}\,v}{B \wedge C\ \mathit{true}}\,{\supset}E}{C\ \mathit{true}}\,{\wedge}E_2}{A \supset C\ \mathit{true}}\,{\supset}I^v}{(A \supset B) \wedge (A \supset C)\ \mathit{true}}\,{\wedge}I}{(A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))\ \mathit{true}}\,{\supset}I^u
$$

5 Disjunction and Falsehood

So far we have explained the meaning of conjunction, truth, and implication. The disjunction "A *or* B" (written as $A \vee B$) is more difficult, but does not require any new judgment forms. Disjunction is characterized by two introduction rules: $A \vee B$ is true, if either A or B is true.

$$
\dfrac{A\ \mathit{true}}{A \vee B\ \mathit{true}}\,{\vee}I_1 \qquad \dfrac{B\ \mathit{true}}{A \vee B\ \mathit{true}}\,{\vee}I_2
$$

Now it would be incorrect to have an elimination rule such as

$$
\dfrac{A \vee B\ \mathit{true}}{A\ \mathit{true}}\,{\vee}E_1?
$$

because even if we know that $A \vee B$ is true, we do not know whether the disjunct A or the disjunct B is true. Concretely, with such a rule we could derive the truth of *every* proposition A as follows:

$$
\dfrac{\dfrac{\dfrac{}{\top\ \mathit{true}}\,{\top}I}{A \vee \top\ \mathit{true}}\,{\vee}I_2}{A\ \mathit{true}}\,{\vee}E_1?
$$

Thus we take a different approach. If we know that $A \vee B$ is true, we must consider two cases: A *true* and B *true*. If we can prove a conclusion C *true* in both cases, then C must be true! Written as an inference rule:

$$
\dfrac{A \vee B\ \mathit{true} \quad \begin{array}{c} \dfrac{}{A\ \mathit{true}}\,u \\ \vdots \\ C\ \mathit{true} \end{array} \quad \begin{array}{c} \dfrac{}{B\ \mathit{true}}\,w \\ \vdots \\ C\ \mathit{true} \end{array}}{C\ \mathit{true}}\,{\vee}E^{u,w}
$$

If we know that $A \vee B$ *true* then we also know C *true*, if that follows both in the case where $A \vee B$ *true* because A is true and in the case where $A \vee B$ *true* because B is true. Note that we use once again the mechanism of hypothetical judgments. In the proof of the second premise we may use the assumption A *true* labeled u, in the proof of the third premise we may use the assumption B *true* labeled w. Both are discharged at the disjunction elimination rule.

Let us justify the conclusion of this rule more explicitly. By the first premise we know $A \vee B$ *true*. The premises of the two possible introduction rules are A *true* and B *true*. In case A *true* we conclude C *true* by the substitution principle and the second premise: we substitute the proof of $A\ true$ for any use of the assumption labeled u in the hypothetical derivation. The case for B *true* is symmetric, using the hypothetical derivation in the third premise.

Because of the complex nature of the elimination rule, reasoning with disjunction is more difficult than with implication and conjunction. As a simple example, we prove the commutativity of disjunction.

$$\begin{array}{c} \vdots \\ (A \vee B) \supset (B \vee A)\ \textit{true} \end{array}$$

We begin with an implication introduction.

$$\dfrac{\begin{array}{c} \overline{A \vee B\ \textit{true}}^{\,u} \\ \vdots \\ B \vee A\ \textit{true} \end{array}}{(A \vee B) \supset (B \vee A)\ \textit{true}} \supset\! I^u$$

At this point we cannot use either of the two disjunction introduction rules. The problem is that neither B nor A follow from our assumption $A \vee B$! So first we need to distinguish the two cases via the rule of disjunction elimination.

$$\dfrac{\dfrac{\overline{A \vee B\ \textit{true}}^{\,u} \quad \begin{array}{c} \overline{A\ \textit{true}}^{\,v} \\ \vdots \\ B \vee A\ \textit{true} \end{array} \quad \begin{array}{c} \overline{B\ \textit{true}}^{\,w} \\ \vdots \\ B \vee A\ \textit{true} \end{array}}{B \vee A\ \textit{true}} \vee\! E^{v,w}}{(A \vee B) \supset (B \vee A)\ \textit{true}} \supset\! I^u$$

The assumption labeled u is still available for each of the two proof obligations, but we have omitted it, since it is no longer needed.

Now each gap can be filled in directly by the two disjunction introduction rules.

$$\dfrac{\dfrac{}{A \vee B\ \mathit{true}}\,u \quad \dfrac{\dfrac{}{A\ \mathit{true}}\,v}{B \vee A\ \mathit{true}}\,{\vee}I_2 \quad \dfrac{\dfrac{}{B\ \mathit{true}}\,w}{B \vee A\ \mathit{true}}\,{\vee}I_1}{\dfrac{B \vee A\ \mathit{true}}{(A \vee B) \supset (B \vee A)\ \mathit{true}}\,{\supset}I^u}\,{\vee}E^{v,w}$$

This concludes the discussion of disjunction. Falsehood (written as $\bot$, sometimes called absurdity) is a proposition that should have no proof! Therefore there are no introduction rules.

Since there cannot be a proof of $\bot$ *true*, it is sound to conclude the truth of any arbitrary proposition if we know $\bot$ *true*. This justifies the elimination rule

$$\dfrac{\bot\ \mathit{true}}{C\ \mathit{true}}\,\bot E$$

We can also think of falsehood as a disjunction between zero alternatives. By analogy with the binary disjunction, we therefore have zero introduction rules, and an elimination rule in which we have to consider zero cases. This is precisely the $\bot E$ rule above.

From this is might seem that falsehood it useless: we can never prove it. This is correct, except that we might reason from contradictory hypotheses! We will see some examples when we discuss negation, since we may think of the proposition "*not* A" (written $\neg A$) as $A \supset \bot$. In other words, $\neg A$ is true precisely if the assumption A *true* is contradictory because we could derive $\bot$ *true*.

6 Summary of Natural Deduction

The judgments, propositions, and inference rules we have defined so far collectively form a system of *natural deduction*. It is a minor variant of a system introduced by Gentzen [Gen35] and studied in depth by Prawitz [Pra65]. One of Gentzen's main motivations was to devise rules that model mathematical reasoning as directly as possible, although clearly in much more detail than in a typical mathematical argument.

The specific interpretation of the truth judgment underlying these rules is *intuitionistic* or *constructive*. This differs from the *classical* or *Boolean* interpretation of truth. For example, classical logic accepts the proposition $A \vee (A \supset B)$ as true for arbitrary A and B, although in the system we have presented so far this would have no proof. Classical logic is based on the

Introduction Rules	**Elimination Rules**
$\dfrac{A\ \mathit{true} \quad B\ \mathit{true}}{A \wedge B\ \mathit{true}}\ \wedge I$	$\dfrac{A \wedge B\ \mathit{true}}{A\ \mathit{true}}\ \wedge E_1 \quad \dfrac{A \wedge B\ \mathit{true}}{B\ \mathit{true}}\ \wedge E_2$
$\dfrac{}{\top\ \mathit{true}}\ \top I$	*no* $\top E$ *rule*
$\begin{array}{c} \overline{A\ \mathit{true}}\ u \\ \vdots \\ \dfrac{B\ \mathit{true}}{A \supset B\ \mathit{true}}\ \supset I^u \end{array}$	$\dfrac{A \supset B\ \mathit{true} \quad A\ \mathit{true}}{B\ \mathit{true}}\ \supset E$
$\dfrac{A\ \mathit{true}}{A \vee B\ \mathit{true}}\ \vee I_1 \quad \dfrac{B\ \mathit{true}}{A \vee B\ \mathit{true}}\ \vee I_2$	$\dfrac{A \vee B\ \mathit{true} \quad \begin{array}{c} \overline{A\ \mathit{true}}\ u \\ \vdots \\ C\ \mathit{true} \end{array} \quad \begin{array}{c} \overline{B\ \mathit{true}}\ w \\ \vdots \\ C\ \mathit{true} \end{array}}{C\ \mathit{true}}\ \vee E^{u,w}$
no $\bot I$ *rule*	$\dfrac{\bot\ \mathit{true}}{C\ \mathit{true}}\ \bot E$

Figure 1: Rules for intuitionistic natural deduction

principle that every proposition must be true or false. If we distinguish these cases we see that $A \vee (A \supset B)$ should be accepted, because in case that A is true, the left disjunct holds; in case A is false, the right disjunct holds. In contrast, intuitionistic logic is based on explicit evidence, and evidence for a disjunction requires evidence for one of the disjuncts. We will return to classical logic and its relationship to intuitionistic logic later; for now our reasoning remains intuitionistic since, as we will see, it has a direct connection to functional computation, which classical logic lacks.

We summarize the rules of inference for the truth judgment introduced so far in Figure 1.

7 Propositions as Types

We now investigate a computational interpretation of constructive proofs and relate it to functional programming. On the propositional fragment of logic this is called the Curry-Howard isomorphism [How80]. From the very outset of the development of constructive logic and mathematics, a central idea has been that *proofs ought to represent constructions*. The Curry-Howard isomorphism is only a particularly poignant and beautiful realization of this idea. In a highly influential subsequent paper, Per Martin-Löf [ML80] developed it further into a more expressive calculus called *type theory*.

In order to illustrate the relationship between proofs and programs we introduce a new judgment:

$M : A$ $\quad$ M is a proof term for proposition A

We presuppose that A is a proposition when we write this judgment. We will also interpret $M : A$ as "M *is a program of type* A". These dual interpretations of the same judgment is the core of the Curry-Howard isomorphism. We either think of M as a syntactic term that represents the proof of A *true*, or we think of A as the type of the program M. As we discuss each connective, we give both readings of the rules to emphasize the analogy.

We intend that if $M : A$ then A *true*. Conversely, if A *true* then $M : A$ for some appropriate proof term M. But we want something more: every deduction of $M : A$ should correspond to a deduction of A *true* with an identical structure and vice versa. In other words we annotate the inference rules of natural deduction with proof terms. The property above should then be obvious. In that way, proof term M of $M : A$ will correspond directly to the corresponding proof of A *true*.

Conjunction. Constructively, we think of a proof of $A \wedge B$ *true* as a pair of proofs: one for A *true* and one for B *true*. So if M is a proof of A and N is a proof of B, then the pair $\langle\!| M, N |\!\rangle$ is a proof of $A \wedge B$.

$$\frac{M : A \quad N : B}{\langle\!| M, N |\!\rangle : A \wedge B} \wedge I$$

The elimination rules correspond to the projections from a pair to its first and second elements to get the individual proofs back out from a pair M.

$$\frac{M : A \wedge B}{\mathsf{fst}\, M : A} \wedge E_1 \qquad \frac{M : A \wedge B}{\mathsf{snd}\, M : B} \wedge E_2$$

Hence the conjunction $A \wedge B$ proposition corresponds to the (lazy) product type $A \mathbin{\&} B$. And, indeed, product types in functional programming languages have the same property that conjunction propositions $A \wedge B$ have. Constructing a pair $\langle\!| M, N |\!\rangle$ of type $A \mathbin{\&} B$ requires a program M of type A and a program N of type B (as in $\wedge I$). Given a pair M of type $A \mathbin{\&} B$, its first component of type A can be retrieved by the projection $\mathsf{fst}\ M$ (as in $\wedge E_1$), its second component of type B by the projection $\mathsf{snd}\ M$ (as in $\wedge E_2$).

Truth. Constructively, we think of a proof of $\top$ *true* as a unit element that carries no information.

$$\frac{}{\langle\!|\,|\!\rangle : \top}\ \top I$$

Hence $\top$ corresponds to the (lazy) unit type with one element that we haven't encountered yet explicity, but is the nullary version of the lazy product, also written as $\top$. There is no elimination rule and hence no further proof term constructs for truth. Indeed, we have not put any information into $\langle\!|\,|\!\rangle$ when constructing it via $\top I$, so cannot expect to get any information back out when trying to eliminate it.

Implication. Constructively, we think of a proof of $A \supset B$ *true* as a function which transforms a proof of A *true* into a proof of B *true*.

We now use the notation of λ-abstraction to annotate the rule of implication introduction with proof terms.

$$\frac{\begin{array}{c}\overline{u : A}\ u \\ \vdots \\ M : B\end{array}}{\lambda u.\, M : A \supset B}\ \supset I^u$$

The hypothesis label u acts as a variable, and any use of the hypothesis labeled u in the proof of B corresponds to an occurrence of u in M. Notice how a constructive proof of B *true* from the additional assumption A *true* to establish $A \supset B$ *true* also describes the transformation of a proof of A *true* to a proof of B *true*. But the proof term $\lambda u.\, M$ explicitly represents this transformation syntactically as a function, instead of leaving this construction implicit by inspection of whatever the proof does.

As a concrete example, consider the (trivial) proof of $A \supset A$ *true*:

$$\dfrac{\dfrac{}{A\ \mathit{true}}\ u}{A \supset A\ \mathit{true}}\ {\supset}I^u$$

If we annotate the deduction with proof terms, we obtain

$$\dfrac{\dfrac{}{u : A}\ u}{(\lambda u.\, u) : A \supset A}\ {\supset}I^u$$

So our proof corresponds to the identity function *id* at type A which simply returns its argument. It can be defined with the identity function $\mathit{id}(u) = u$ or $\mathrm{id} = (\lambda u.\, u)$.

Constructively, a proof of $A \supset B$ *true* is a function transforming a proof of A *true* to a proof of B *true*. Using $A \supset B$ *true* by its elimination rule ${\supset}E$, thus, corresponds to providing the proof of A *true* that $A \supset B$ *true* is waiting for to obtain a proof of B *true*. The rule for implication elimination corresponds to function application.

$$\dfrac{M : A \supset B \quad N : A}{M\, N : B}\ {\supset}E$$

What is the meaning of $A \supset B$ as a type? From the discussion above it should be clear that it can be interpreted as a function type $A \to B$. The introduction and elimination rules for implication can also be viewed as formation rules for functional abstraction $\lambda u.\, M$ and application $M\, N$. Forming a functional abstraction $\lambda u.\, M$ corresponds to a function that accepts input parameter u of type A and produces M of type B (as in ${\supset}I$). Using a function $M : A \to B$ corresponds to applying it to a concrete input argument N of type A to obtain an output $M\, N$ of type B.

Note that we obtain the usual introduction and elimination rules for implication if we erase the proof terms. This will continue to be true for all rules in the remainder of this section and is immediate evidence for the soundness of the proof term calculus, that is, if $M : A$ then A *true*.

As a second example we consider a proof of $(A \wedge B) \supset (B \wedge A)$ *true*.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \wedge B\ \mathit{true}}\ u}{B\ \mathit{true}}\ {\wedge}E_2 \quad \dfrac{\dfrac{}{A \wedge B\ \mathit{true}}\ u}{A\ \mathit{true}}\ {\wedge}E_1}{B \wedge A\ \mathit{true}}\ {\wedge}I}{(A \wedge B) \supset (B \wedge A)\ \mathit{true}}\ {\supset}I^u$$

When we annotate this derivation with proof terms, we obtain the swap function which takes a pair $\langle M, N\rangle$ and returns the reverse pair $\langle N, M\rangle$.

$$\dfrac{\dfrac{\dfrac{}{u : A \wedge B}\,u}{\mathsf{snd}\ u : B}\,\wedge E_2 \quad \dfrac{\dfrac{}{u : A \wedge B}\,u}{\mathsf{fst}\ u : A}\,\wedge E_1}{\dfrac{\langle\!| \mathsf{snd}\ u, \mathsf{fst}\ u |\!\rangle : B \wedge A}{(\lambda u.\ \langle\!| \mathsf{snd}\ u, \mathsf{fst}\ u |\!\rangle) : (A \wedge B) \supset (B \wedge A)}\,\supset I^u}\,\wedge I$$

Disjunction. Constructively, we think of a proof of $A \vee B$ *true* as either a proof of A *true* or B *true*. Disjunction therefore corresponds to a disjoint sum type $A + B$ that either store something of type A or something of type B. The two introduction rules correspond to the left and right injection into a sum type.

$$\frac{M : A}{\mathbf{l} \cdot M : A \vee B}\,\vee I_1 \qquad \frac{N : B}{\mathbf{r} \cdot N : A \vee B}\,\vee I_2$$

When using a disjunction $A \vee B$ *true* in a proof, we need to be prepared to handle A *true* as well as B *true*, because we don't know whether $\vee I_1$ or $\vee I_2$ was used to prove it. The elimination rule corresponds to a case construct which discriminates between a left and right injection into a sum types.

$$\dfrac{M : A \vee B \qquad \begin{array}{c}\overline{u : A}\,u \\ \vdots \\ N : C\end{array} \qquad \begin{array}{c}\overline{w : B}\,w \\ \vdots \\ P : C\end{array}}{\mathsf{case}\ M\ (\mathbf{l} \cdot u \Rightarrow N \mid \mathbf{r} \cdot w \Rightarrow P) : C}\,\vee E^{u,w}$$

Recall that the hypothesis labeled u is available only in the proof of the second premise and the hypothesis labeled w only in the proof of the third premise. This means that the scope of the variable u is N, while the scope of the variable w is P.

Falsehood. There is no introduction rule for falsehood ($\perp$). We can therefore view it as the empty type 0. The corresponding elimination rule allows a term of $\perp$ to stand for an expression of any type when wrapped in a case with no alternatives. There can be no valid reduction rule for falsehood, which means during computation of a valid program we will never try to evaluate a term of the form case M ( ).

$$\frac{M : \perp}{\mathsf{case}\ M\ (\,) : C}\,\perp E$$

Interaction Laws. This completes our assignment of proof terms to the logical inference rules. Now we can interpret the interaction laws we introduced early as programming exercises. Consider the following distributivity law:

(L11a) $(A \supset (B \wedge C)) \supset (A \supset B) \wedge (A \supset C)$ *true*

Interpreted constructively, this assignment can be read as:

> Write a function which, when given a function from A to pairs of type $B \wedge C$, returns two functions: one which maps A to B and one which maps A to C.

This is satisfied by the following function:

$$\lambda u.\, \langle\!|(\lambda w.\, \mathsf{fst}\ (u\,w)), (\lambda v.\, \mathsf{snd}\ (u\,v))|\!\rangle$$

The following deduction provides the evidence:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{u : A \supset (B \wedge C)}\,u \qquad \dfrac{}{w : A}\,w}{u\,w : B \wedge C}\,{\supset}E}{\mathsf{fst}\ (u\,w) : B}\,{\wedge}E_1}{\lambda w.\, \mathsf{fst}\ (u\,w) : A \supset B}\,{\supset}I^w \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{u : A \supset (B \wedge C)}\,u \qquad \dfrac{}{v : A}\,v}{u\,v : B \wedge C}\,{\supset}E}{\mathsf{snd}\ (u\,v) : C}\,{\wedge}E_2}{\lambda v.\, \mathsf{snd}\ (u\,v) : A \supset C}\,{\supset}I^v}{\langle\!|(\lambda w.\, \mathsf{fst}\ (u\,w)), (\lambda v.\, \mathsf{snd}\ (u\,v))|\!\rangle : (A \supset B) \wedge (A \supset C)}\,{\wedge}I}{\lambda u.\, \langle\!|(\lambda w.\, \mathsf{fst}\ (u\,w)), (\lambda v.\, \mathsf{snd}\ (u\,v))|\!\rangle : (A \supset (B \wedge C)) \supset ((A \supset B) \wedge (A \supset C))}\,{\supset}I^u$$

Programs in constructive propositional logic are somewhat uninteresting in that they do not manipulate basic data types such as natural numbers, integers, lists, trees, etc. We introduce such data types later in this course, following the same method we have used in the development of logic.

Summary. To close this section we recall the *guiding principles behind the assignment of proof terms to deductions*.

1. For every deduction of A *true* there is a proof term M and deduction of $M : A$.

2. For every deduction of $M : A$ there is a deduction of A *true*

3. The correspondence between proof terms M and deductions of A *true* is a bijection.

8 Reduction

In the preceding section, we have introduced the assignment of proof terms to natural deductions. If proofs are programs then we need to explain how proofs are to be executed, and which results may be returned by a computation.

We explain the operational interpretation of proofs in two steps. In the first step we introduce a judgment of *reduction* written $M \longrightarrow M'$ and read "M *reduces to* M'". In the second step, a computation then proceeds by a sequence of reductions $M \longrightarrow M_1 \longrightarrow M_2 \ldots$, according to a fixed strategy, until we reach a value which is the result of the computation.

As in the development of propositional logic, we discuss each of the connectives separately, taking care to make sure the explanations are independent. This means we can consider various sublanguages and we can later extend our logic or programming language without invalidating the results from this section. Furthermore, it greatly simplifies the analysis of properties of the reduction rules.

In general, we think of the proof terms corresponding to the introduction rules as the *constructors* and the proof terms corresponding to the elimination rules as the *destructors*.

Conjunction. The constructor forms a pair, while the destructors are the left and right projections. The reduction rules prescribe the actions of the projections.

$$\begin{array}{rcl} \mathsf{fst}\,\langle\!|M, N|\!\rangle & \longrightarrow & M \\ \mathsf{snd}\,\langle\!|M, N|\!\rangle & \longrightarrow & N \end{array}$$

These (computational) reduction rules directly corresponds to the proof term analogue of the logical reductions for the local soundness detailed in Section 11. For example:

$$\dfrac{\dfrac{M : A \quad N : B}{\langle\!|M, N|\!\rangle : A \wedge B}\,\wedge I}{\mathsf{fst}\,\langle\!|M, N|\!\rangle : A}\,\wedge E_1 \quad \longrightarrow \quad M : A$$

Truth. The constructor just forms the unit element, $\langle\!|\,|\!\rangle$. Since there is no destructor, there is no reduction rule.

Implication. The constructor forms a function by λ-abstraction, while the destructor applies the function to an argument. The notation for the substitution of N for occurrences of u in M is $[N/u]M$. We therefore write the reduction rule as

$$(\lambda u.\, M)\, N \quad \longrightarrow \quad [N/u]M$$

We have to be somewhat careful so that substitution behaves correctly. In particular, no variable in N should be bound in M in order to avoid conflict. We can always achieve this by renaming bound variables—an operation which clearly does not change the meaning of a proof term. Again, this computational reduction directly relates to the logical reduction from the local soundness using the substitution notation for the right-hand side:

$$\dfrac{\dfrac{\dfrac{}{u : A}\,u \quad \vdots \quad M : B}{\lambda u.\, M : A \supset B}\supset I^u \qquad N : A}{(\lambda u.\, M)\, N : B}\supset E \quad \longrightarrow \quad [N/u]M$$

Disjunction. The constructors inject into a sum types; the destructor distinguishes cases. We need to use substitution again.

$$\begin{array}{lcl} \mathsf{case}\ \mathbf{l}\cdot M\ (\mathbf{l}\cdot u \Rightarrow N \mid \mathbf{r}\cdot w \Rightarrow P) & \longrightarrow & [M/u]N \\ \mathsf{case}\ \mathbf{r}\cdot M\ (\mathbf{l}\cdot u \Rightarrow N \mid \mathbf{r}\cdot w \Rightarrow P) & \longrightarrow & [M/w]P \end{array}$$

The analogy with the logical reduction again works, for example:

$$\dfrac{\dfrac{M : A}{\mathbf{l}\cdot M : A \vee B}\vee I_1 \qquad \dfrac{}{u : A}\,u \ \vdots\ N : C \qquad \dfrac{}{w : B}\,w \ \vdots\ P : C}{\mathsf{case}\ \mathbf{l}\cdot M\ (\mathbf{l} cdot u \Rightarrow N \mid \mathbf{r}\cdot w \Rightarrow P) : C}\vee E^{u,w} \quad \longrightarrow \quad [M/u]N$$

Falsehood. Since there is no constructor for the empty type there is no reduction rule for falsehood. There is no computation rule and we will not try to evaluate case M ( ).

This concludes the definition of the reduction judgment. Observe that the construction principle for the (computational) reductions is to investigate what happens when a destructor is applied to a corresponding constructor.

This is in correspondence with how (logical) reductions for local soundness consider what happens when an elimination rule is used in succession on the output of an introduction rule (when reading proofs top to bottom).

9 Summary of Proof Terms

Judgments.

$M : A$	M is a proof term for proposition A, see Figure 2
$M \longrightarrow M'$	M reduces to M', see Figure 3

10 Summary of the Curry-Howard Correspondence

The Curry-Howard correspondence we have elaborated in this lecture has three central components:

- Propositions are interpreted as types
- Proofs are interpreted as programs
- Proof reductions are interpreted as computation

This correspondence goes in both directions, but it does not capture everything we have been using so far.

Proposition	**Type**
$A \wedge B$	$\tau \mathbin{\&} \sigma$
$A \supset B$	$\tau \to \sigma$
$A \vee B$	$\tau + \sigma$
$\top$	$\top$
$\bot$	0
?	$A \times B$
?	1
??	$\mu\alpha.\,\tau$

For $A \times B$ and 1 we obtain other forms of logical conjunction and truth that have the same introduction rules as $A \wedge B$ and $\top$, respectively, but different elimination rules:

$$\dfrac{A \times B \qquad \begin{array}{c} \overline{A}^{\,u} \quad \overline{B}^{\,w} \\ \vdots \\ C \end{array}}{C} \times E^{u,w} \qquad\qquad \dfrac{1 \quad C}{C}\, 1E$$

Constructors

Destructors

$$\frac{M : A \quad N : B}{\langle\!| M, N |\!\rangle : A \wedge B}\ \wedge I$$

$$\frac{M : A \wedge B}{\mathsf{fst}\ M : A}\ \wedge E_1$$

$$\frac{M : A \wedge B}{\mathsf{snd}\ M : B}\ \wedge E_2$$

$$\frac{}{\langle\!|\ |\!\rangle : \top}\ \top I$$

no destructor for $\top$

$$\frac{\begin{array}{c}\overline{u : A}\ u\\ \vdots \\ M : B\end{array}}{\lambda u.\, M : A \supset B}\ \supset I^u$$

$$\frac{M : A \supset B \quad N : A}{M\, N : B}\ \supset E$$

$$\frac{M : A}{\mathbf{l} \cdot M : A \vee B}\ \vee I_1$$

$$\frac{M : A \vee B \quad \begin{array}{c}\overline{u : A}\ u\\ \vdots \\ N : C\end{array} \quad \begin{array}{c}\overline{w : B}\ w\\ \vdots \\ P : C\end{array}}{\mathsf{case}\ M\ (\mathbf{l} \cdot u \Rightarrow N \mid \mathbf{r} \cdot w \Rightarrow P) : C}\ \vee E^{u,w}$$

$$\frac{N : B}{\mathbf{r} \cdot N : A \vee B}\ \vee I_2$$

no constructor for $\bot$

$$\frac{M : \bot}{\mathsf{case}\ M\ (\,) : C}\ \bot E$$

Figure 2: Proof term assignment for natural deduction

$$
\begin{array}{rcl}
\mathsf{fst}\,\langle\!|M, N|\!\rangle & \longrightarrow & M \\
\mathsf{snd}\,\langle\!|M, N|\!\rangle & \longrightarrow & N \\
\text{no reduction for } \langle\!|\,|\!\rangle & & \\
(\lambda u.\, M)\, N & \longrightarrow & [N/u]M \\
\mathsf{case}\ (\mathbf{l} \cdot M)\ (\mathbf{l} \cdot u \Rightarrow N \mid \mathbf{r} \cdot w \Rightarrow P) & \longrightarrow & [M/u]N \\
\mathsf{case}\ (\mathbf{r} \cdot M)\ (\mathbf{l} \cdot u \Rightarrow N \mid \mathbf{r} \cdot w \Rightarrow P) & \longrightarrow & [M/w]P \\
\text{no reduction for } \mathsf{case}\ M\ (\,) & &
\end{array}
$$

Figure 3: Proof term reductions

These are logically equivalent to existing connectives ($A \times B \equiv A \wedge B$ and $1 \equiv \top$), so they are not usually used in a treatment of intuitionistic logic, but their operational interpretations are different (eager vs. lazy).

As for general recursive types $\rho\alpha.\, \tau$, there aren't any good propositional analogues on the logical side in general. The overarching study of type theory (encompassing both logic and its computational interpretation) treats the so-called inductive and coinductive types as special cases. Similarly, the fixed point construction $\mathbf{fix}\, x.\, e$ does not have a good logical analogue, only special cases of it do.

11 Harmony

This is bonus material only touched upon in lecture. It elaborates on how proof reduction arises in the study of logic.

In the verificationist definition of the logical connectives via their introduction rules we have briefly justified the elimination rules. We now study the balance between introduction and elimination rules more closely.

We elaborate on the verificationist point of view that logical connectives are defined by their introduction rules. We show that for intuitionistic logic as presented so far, the elimination rules are in harmony with the introduction rules in the sense that they are neither too strong nor too weak. We demonstrate this via local reductions and expansions, respectively.

In order to show that introduction and elimination rules are in harmony we establish two properties: *local soundness* and *local completeness*.

Local soundness shows that the elimination rules are not too strong: no matter how we apply elimination rules to the result of an introduction we cannot gain any new information. We demonstrate this by showing that we can find a more direct proof of the conclusion of an elimination than one that first introduces and then eliminates the connective in question. This is witnessed by a *local reduction* of the given introduction and the subsequent elimination.

Local completeness shows that the elimination rules are not too weak: there is always a way to apply elimination rules so that we can reconstitute a proof of the original proposition from the results by applying introduction rules. This is witnessed by a *local expansion* of an arbitrary given derivation into one that introduces the primary connective.

Connectives whose introduction and elimination rules are in harmony in the sense that they are locally sound and complete are properly defined from the verificationist perspective. If not, the proposed connective should be viewed with suspicion. Another criterion we would like to apply uniformly is that both introduction and elimination rules do not refer to other propositional constants or connectives (besides the one we are trying to define), which could create a dangerous dependency of the various connectives on each other. As we present correct definitions we will occasionally also give some counterexamples to illustrate the consequences of violating the principles behind the patterns of valid inference.

In the discussion of each individual connective below we use the notation

$$\begin{array}{c}\mathcal{D}\\ A\ \mathit{true}\end{array} \Longrightarrow_R \begin{array}{c}\mathcal{D}'\\ A\ \mathit{true}\end{array}$$

for the local reduction of a deduction $\mathcal{D}$ to another deduction $\mathcal{D}'$ of the same judgment A *true*. In fact, $\Longrightarrow_R$ can itself be a higher level judgment relating two proofs, $\mathcal{D}$ and $\mathcal{D}'$, although we will not directly exploit this point of view. Similarly,

$$\begin{array}{c}\mathcal{D}\\ A\ \mathit{true}\end{array} \Longrightarrow_E \begin{array}{c}\mathcal{D}'\\ A\ \mathit{true}\end{array}$$

is the notation of the local expansion of $\mathcal{D}$ to $\mathcal{D}'$.

Conjunction. We start with local soundness, i.e., locally reducing an elimination of a conjunction that was just introduced. Since there are two elimination rules and one introduction, we have two cases to consider, because there are two different elimination rules $\wedge E_1$ and $\wedge E_2$ that could follow the

$\wedge I$ introduction rule. In either case, we can easily reduce.

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ A\ \mathit{true}\end{matrix} \quad \begin{matrix}\mathcal{E}\\ B\ \mathit{true}\end{matrix}}{A \wedge B\ \mathit{true}}\wedge I}{A\ \mathit{true}}\wedge E_1 \quad \Longrightarrow_R \quad \begin{matrix}\mathcal{D}\\ A\ \mathit{true}\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ A\ \mathit{true}\end{matrix} \quad \begin{matrix}\mathcal{E}\\ B\ \mathit{true}\end{matrix}}{A \wedge B\ \mathit{true}}\wedge I}{B\ \mathit{true}}\wedge E_2 \quad \Longrightarrow_R \quad \begin{matrix}\mathcal{E}\\ B\ \mathit{true}\end{matrix}$$

These two reductions justify that, after we just proved a conjunction $A \wedge B$ to be true by the introduction rule $\wedge I$ from a proof $\mathcal{D}$ of A *true* and a proof $\mathcal{E}$ of B *true*, the only thing we can get back out by the elimination rules is something that we have put into the proof of $A \wedge B$ *true*. This makes $\wedge E_1$ and $\wedge E_2$ locally sound, because the only thing we get out is A *true* which already has the direct proof $\mathcal{D}$ as well as B *true* which has the direct proof $\mathcal{E}$. The above two reductions make $\wedge E_1$ and $\wedge E_2$ locally sound.

Local completeness establishes that we are not losing information from the elimination rules. Local completeness requires us to apply eliminations to an arbitrary proof of $A \wedge B$ *true* in such a way that we can reconstitute a proof of $A \wedge B$ from the results.

$$\begin{matrix}\mathcal{D}\\ A \wedge B\ \mathit{true}\end{matrix} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\begin{matrix}\mathcal{D}\\ A \wedge B\ \mathit{true}\end{matrix}}{A\ \mathit{true}}\wedge E_1 \quad \dfrac{\begin{matrix}\mathcal{D}\\ A \wedge B\ \mathit{true}\end{matrix}}{B\ \mathit{true}}\wedge E_2}{A \wedge B\ \mathit{true}}\wedge I$$

This local expansion shows that, collectively, the elimination rules $\wedge E_1$ and $\wedge E_2$ extract all information from the judgment $A \wedge B$ *true* that is needed to reprove $A \wedge B$ *true* with the introduction rule $\wedge I$. Remember that the hypothesis $A \wedge B$ *true*, once available, can be used multiple times, which is very apparent in the local expansion, because the proof $\mathcal{D}$ of $A \wedge B$ *true* can simply be repeated on the left and on the right premise.

As an example where local completeness fails, consider the case where we "forget" the second/right elimination rule $\wedge E_2$ for conjunction. The remaining rule is still locally sound, because it proves something that was put into the proof of $A \wedge B$ *true*, but not locally complete because we cannot extract a proof of B from the assumption $A \wedge B$. Now, for example, we cannot prove $(A \wedge B) \supset (B \wedge A)$ even though this should clearly be true.

Substitution Principle. We need the defining property for hypothetical judgments before we can discuss implication. Intuitively, we can always substitute a deduction of A *true* for any use of a hypothesis A *true*. In order to avoid ambiguity, we make sure assumptions are labelled and we substitute for all uses of an assumption with a given label. Note that we can only substitute for assumptions that are not discharged in the subproof we are considering. The substitution principle then reads as follows:

> If
>
> $$\begin{array}{c} \overline{A\ \mathit{true}}\ u \\ \mathcal{E} \\ B\ \mathit{true} \end{array}$$
>
> is a hypothetical proof of B *true* under the undischarged hypothesis A *true* labelled u, and
>
> $$\begin{array}{c} \mathcal{D} \\ A\ \mathit{true} \end{array}$$
>
> is a proof of A *true* then
>
> $$\begin{array}{c} \dfrac{\mathcal{D}}{A\ \mathit{true}}\ u \\ \mathcal{E} \\ B\ \mathit{true} \end{array}$$
>
> is our notation for substituting $\mathcal{D}$ for all uses of the hypothesis labelled u in $\mathcal{E}$. This deduction, also sometime written as $[\mathcal{D}/u]\mathcal{E}$ no longer depends on u.

Implication. To witness local soundness, we reduce an implication introduction followed by an elimination using the substitution operation.

$$\dfrac{\dfrac{\begin{array}{c} \overline{A\ \mathit{true}}\ u \\ \mathcal{E} \\ B\ \mathit{true} \end{array}}{A \supset B\ \mathit{true}} \supset\! I^u \qquad \begin{array}{c} \mathcal{D} \\ A\ \mathit{true} \end{array}}{B\ \mathit{true}} \supset\! E \quad \Longrightarrow_R \quad \begin{array}{c} \dfrac{\mathcal{D}}{A\ \mathit{true}}\ u \\ \mathcal{E} \\ B\ \mathit{true} \end{array}$$

The conditions on the substitution operation is satisfied, because u is introduced at the $\supset\! I^u$ inference and therefore not discharged in $\mathcal{E}$.

Local completeness is witnessed by the following expansion.

$$\begin{array}{c} \mathcal{D} \\ A \supset B \; \textit{true} \end{array} \quad \Longrightarrow_E \quad \dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ A \supset B \; \textit{true}\end{array} \quad \dfrac{}{A \; \textit{true}}\,u}{B \; \textit{true}}\,{\supset}E}{A \supset B \; \textit{true}}\,{\supset}I^u$$

Here u must be chosen fresh: it only labels the new hypothesis A *true* which is used only once.

Disjunction. For disjunction we also employ the substitution principle because the two cases we consider in the elimination rule introduce hypotheses. Also, in order to show local soundness we have two possibilities for the introduction rule, in both situations followed by the only elimination rule.

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ A \; \textit{true}\end{array}}{A \vee B \; \textit{true}}\,{\vee}I_L \quad \begin{array}{c}\dfrac{}{A \; \textit{true}}\,u \\ \mathcal{E} \\ C \; \textit{true}\end{array} \quad \begin{array}{c}\dfrac{}{B \; \textit{true}}\,w \\ \mathcal{F} \\ C \; \textit{true}\end{array}}{C \; \textit{true}}\,{\vee}E^{u,w} \quad \Longrightarrow_R \quad \begin{array}{c}\dfrac{\mathcal{D}}{A \; \textit{true}}\,u \\ \mathcal{E} \\ C \; \textit{true}\end{array}$$

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ B \; \textit{true}\end{array}}{A \vee B \; \textit{true}}\,{\vee}I_R \quad \begin{array}{c}\dfrac{}{A \; \textit{true}}\,u \\ \mathcal{E} \\ C \; \textit{true}\end{array} \quad \begin{array}{c}\dfrac{}{B \; \textit{true}}\,w \\ \mathcal{F} \\ C \; \textit{true}\end{array}}{C \; \textit{true}}\,{\vee}E^{u,w} \quad \Longrightarrow_R \quad \begin{array}{c}\dfrac{\mathcal{D}}{B \; \textit{true}}\,w \\ \mathcal{F} \\ C \; \textit{true}\end{array}$$

An example of a rule that would not be locally sound is

$$\dfrac{A \vee B \; \textit{true}}{A \; \textit{true}}\,{\vee}E_1?$$

and, indeed, we would not be able to reduce

$$\dfrac{\dfrac{B \; \textit{true}}{A \vee B \; \textit{true}}\,{\vee}I_R}{A \; \textit{true}}\,{\vee}E_1?$$

In fact we can now derive a contradiction from no assumption, which means the whole system is incorrect.

$$\dfrac{\dfrac{\dfrac{}{\top \; \textit{true}}\,{\top}I}{\bot \vee \top \; \textit{true}}\,{\vee}I_R}{\bot \; \textit{true}}\,{\vee}E_1?$$

Local completeness of disjunction distinguishes cases on the known $A \vee B$ *true*, using $A \vee B$ *true* as the conclusion.

$$\begin{array}{c}\mathcal{D}\\ A \vee B\ \mathit{true}\end{array} \Longrightarrow_E \dfrac{\begin{array}{c}\mathcal{D}\\ A \vee B\ \mathit{true}\end{array} \quad \dfrac{\dfrac{}{A\ \mathit{true}}\,u}{A \vee B\ \mathit{true}}\,\vee I_L \quad \dfrac{\dfrac{}{B\ \mathit{true}}\,w}{A \vee B\ \mathit{true}}\,\vee I_R}{A \vee B\ \mathit{true}}\,\vee E^{u,w}$$

Visually, this looks somewhat different from the local expansions for conjunction or implication. It looks like the elimination rule is applied last, rather than first. Mostly, this is due to the notation of natural deduction: the above represents the step from using the knowledge of $A \vee B$ *true* and eliminating it to obtain the hypotheses A *true* and B *true* in the two cases.

Truth. The local constant $\top$ has only an introduction rule, but no elimination rule. Consequently, there are no cases to check for local soundness: any introduction followed by any elimination can be reduced, because $\top$ has no elimination rules.

However, local completeness still yields a local expansion: Any proof of $\top$ *true* can be trivially converted to one by $\top I$.

$$\begin{array}{c}\mathcal{D}\\ \top\ \mathit{true}\end{array} \Longrightarrow_E \dfrac{}{\top\ \mathit{true}}\,\top I$$

Falsehood. As for truth, there is no local reduction because local soundness is trivially satisfied since we have no introduction rule.

Local completeness is slightly tricky. Literally, we have to show that there is a way to apply an elimination rule to any proof of $\bot$ *true* so that we can reintroduce a proof of $\bot$ *true* from the result. However, there will be zero cases to consider, so we apply no introductions. Nevertheless, the following is the right local expansion.

$$\begin{array}{c}\mathcal{D}\\ \bot\ \mathit{true}\end{array} \Longrightarrow_E \dfrac{\begin{array}{c}\mathcal{D}\\ \bot\ \mathit{true}\end{array}}{\bot\ \mathit{true}}\,\bot E$$

Reasoning about situation when falsehood is true may seem vacuous, but is common in practice because it corresponds to reaching a contradiction. In intuitionistic reasoning, this occurs when we prove $A \supset \bot$ which is often abbreviated as $\neg A$. In classical reasoning it is even more frequent, due to the rule of proof by contradiction.

Exercises

Exercise 1 One proposition is *more general* than another if we can instantiate the propositional variables in the first to obtain the second. For example, $A \supset (B \supset A)$ is more general than $A \supset (\bot \supset A)$ (with $[\bot/B]$), $(C \wedge D) \supset (B \supset (C \wedge D))$ (with $[C \wedge D/A]$, but not more general than $C \supset (D \supset E)$.

For each of the following proof terms, give the most general proposition proved by it. (We are justified in saying "*the most general*" because the most general proposition is unique up to the names of the propositional variables.)

1. $\lambda u.\, \lambda w.\, \lambda k.\, w\,(u\,k)$
2. $\lambda w.\, \langle(\lambda u.\, w\,(\mathbf{l} \cdot u)), (\lambda k.\, w\,(\mathbf{r} \cdot k))\rangle$
3. $\lambda x.\, (\mathsf{fst}\, x)\, (\mathsf{snd}\, x)\, (\mathsf{snd}\, x)$
4. $\lambda x.\, \lambda y.\, \lambda z.\, (x\,z)\,(y\,z)$

Exercise 2 Write out a proof term for each of the following propositions. As you know from this lecture, this is the same as writing a program of the translated type in our program language without the use of fixed points.

1. $(A \wedge (A \supset \bot)) \supset B$
2. $(A \vee (A \supset \bot)) \supset (((A \supset \bot) \supset \bot) \supset A)$

References

[Dum91] Michael Dummett. *The Logical Basis of Metaphysics*. Harvard University Press, Cambridge, Massachusetts, 1991. The William James Lectures, 1976.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.

[How80] W. A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, 1980. Hitherto unpublished note of 1969, rearranged, corrected, and annotated by Howard.

[ML80] Per Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science VI*, pages 153–175. North-Holland, 1980.

[ML96] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996. Notes for three lectures given in Siena, April 1983.

[Pra65] Dag Prawitz. *Natural Deduction*. Almquist & Wiksell, Stockholm, 1965.