

pty.1 Reduction

int:pty:red:
sec In natural deduction **derivations**, an introduction rule that is followed by an elimination rule is redundant. For instance, the **derivation**

$$\frac{\frac{\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \rightarrow\text{Elim} \quad [\chi]}{\psi \wedge \chi} \wedge\text{Intro}}{\frac{\psi}{\psi} \wedge\text{Elim}} \rightarrow\text{Intro}$$

can be replaced with the simpler **derivation**:

$$\frac{\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \rightarrow\text{Elim}}{\chi \rightarrow \psi} \rightarrow\text{Intro}$$

As we see, an $\wedge\text{Intro}$ followed by $\wedge\text{Elim}$ “cancel out.” In general, we see that the conclusion of $\wedge\text{Elim}$ is always the formula on one side of the conjunction, and the premises of $\wedge\text{Intro}$ requires both sides of the conjunction, thus if we need a derivation of either side, we can simply use that derivation without introducing the conjunction followed by eliminating it.

Thus in general we have

$$\frac{\frac{\begin{array}{c} \vdots \\ D_1 \\ \vdots \\ \varphi_1 \end{array} \quad \begin{array}{c} \vdots \\ D_2 \\ \vdots \\ \varphi_2 \end{array}}{\varphi_1 \wedge \varphi_2} \wedge\text{Intro} \quad \wedge\text{Elim}_i \quad \rightarrow \quad \begin{array}{c} \vdots \\ D_i \\ \vdots \\ \varphi_i \end{array}$$

The \rightarrow symbol has a similar meaning as in the lambda calculus, i.e., a single step of a reduction. In the proof term syntax for **derivations**, the above reduction rule thus becomes:

$$(\Gamma, p_i \langle M_1^{\varphi_1}, M_2^{\varphi_2} \rangle) \rightarrow (\Gamma, M_i)$$

In the typed lambda calculus, this is the beta reduction rule for the product type.

Note the type annotation on M_1 and M_2 : while in the standard term syntax only $\lambda x^\varphi. N$ has such notion, we reuse the notation here to remind us of the formula the term is associated with in the corresponding natural deduction **derivation**, to reveal the correspondence between the two kinds of syntax.

In natural deduction, a pair of inferences such as those on the left, i.e., a pair that is subject to cancelling is called a *cut*. In the typed lambda calculus the term on the left of \rightarrow is called a *redex*, and the term to the right is called the *reductum*. Unlike untyped lambda calculus, where only $(\lambda x. N)Q$ is considered to be redex, in the typed lambda calculus the syntax is extended to terms

both sides of \rightarrow . The resulting pairs of trees constitutes an extended notion of reduction; it is analogous to compatibility in the untyped lambda calculus.

It's easy to see that the context Γ does not change during the reduction (both the original and the extended version), thus it's unnecessary to mention the context when we are discussing reductions. In what follows we will assume that every term is accompanied by a context which does no change during reduction. We then say "proof term" when we mean a proof term accompanied by a context which makes it well-typed.

As in lambda calculus, the notion of normal-form term and normal deduction is given:

Definition pty.1. A proof term with no redex is said to be in *normal form*; likewise, a *derivation* without cuts is a *normal derivation*. A proof term is in normal form if and only if its counterpart *derivation* is normal.

Photo Credits

Bibliography