pty.1 Reduction

int:pty:red: sec

In natural deduction derivations, an introduction rule that is followed by an elimination rule is a detour that can be "reduced". For instance, an ∧Intro followed by ∧Elim "cancel out," since the conclusion of ∧Elim is always one of the conjuncts of the conjunction that ∧Intro produces, and the premises of ∧Intro are both conjuncts. Thus if we need a derivation of either conjunct, we can simply use that derivation without introducing the conjunction followed by eliminating it. We have

$$\dfrac{\dfrac{\begin{array}{cc}\vdots\,\delta_1 & \vdots\,\delta_2\\ \varphi_1 & \varphi_2\end{array}}{\varphi_1 \wedge \varphi_2}\,\wedge\text{Intro}}{\varphi_i}\,\wedge\text{Elim}_i \quad\rightarrow\quad \begin{array}{c}\vdots\,\delta_i\\ \varphi_i\end{array}$$

We can consider a similar relation of reduction on the associated proof terms suggested by this simplification. If N_1 is the proof term for δ_1 and N_2 the one for δ_2, then we can say the proof term for the proof on the left reduces (in one step) to the proof term for the proof on the right:

$$\pi_i(\langle N_1, N_2, \rangle) \to N_i$$

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

In natural deduction, a pair of inferences such as a ∧Intro folloed by ∧Elim, i.e., a pair that is subject to reduction, is called a *cut*. In the typed lambda calculus the term on the left of → 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 involving $\langle N, M\rangle$, $\pi_i(N)$, $\iota_i(\varphi)N$, $\delta\, N\, x_1.M_1\, x_2.M_2$, and $\varepsilon^{\varphi}(N)$, with corresponding redexes.

Similarly we have reduction for disjunction:

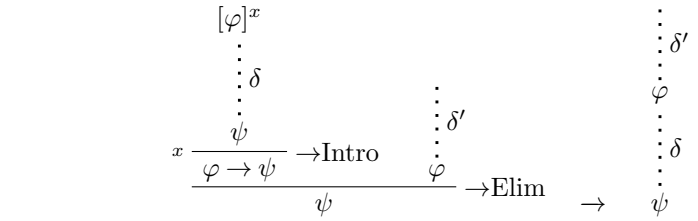
$$\dfrac{\dfrac{\begin{array}{ccc}\vdots\,\delta & [\varphi_1]^{x_1} & [\varphi_2]^{x_2}\\ \varphi_i & \vdots\,\delta_1 & \vdots\,\delta_2\end{array}}{\varphi_1 \vee \varphi_2}\,\vee\text{Intro} \qquad \chi \qquad \chi}{\chi}\,{\scriptstyle x_1,\,x_2}\,\vee\text{Elim} \quad\rightarrow\quad \begin{array}{c}\vdots\,\delta\\ \varphi_i\\ \vdots\,\delta_i\\ \chi\end{array}$$

This corresponds to a reduction on proof terms:

$$\delta\, \iota_i^{\varphi_i}(M)\, x_1.N_1\, x_2.N_2 \to N_i[M/x_i]$$

where M is the proof term of δ, and N_i is the proof term of δ_i. This is the beta reduction rule of for sum types. Here, $M[N_i/x]$ means replacing all free occurrences of the variable x in M by N_i.

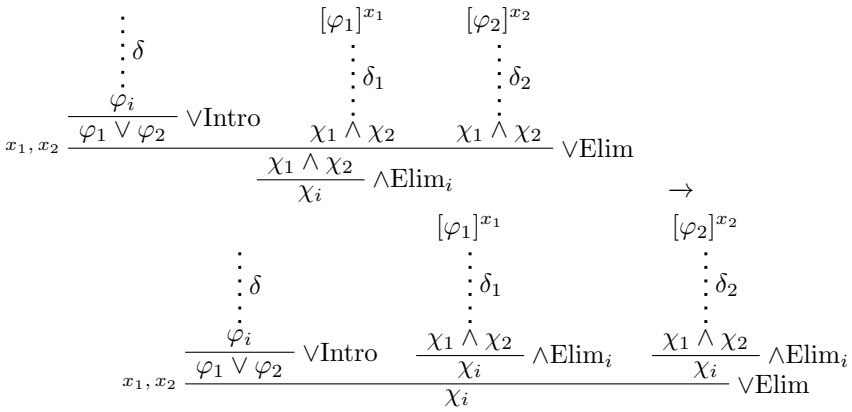
The reduction of the function type corresponds to removal of a detour of →Intro followed by a →Elim.



For proof terms, this amounts to ordinary beta reduction:

$$(\lambda x^{\varphi}.\, NM) \to N[M/x]$$

In addition to reduction conversions on derivations, we can consider permutation conversions. These apply to (sub)derivations that end in a ∨Elim followed by another elimination rule, e.g.:



If the proof terms of δ, δ_1, and δ_2 are M, N_1, and N_2, respectively, the corresponding reduction rule for proof terms, i.e., λ terms, is:

$$\pi_i(\delta\, M\, x_1.N_1\, x_2.N_2) \to \delta\, M\, x_1.\pi_i(N_1)\, x_2.\pi_i(N_2)$$

Of course we can eliminate detours not just at the ends of derivations, but also within derivations, by applying the reduction conversion to a sub-derivation ending in a detour. Similarly, β-reduction applies to subterms of λ-terms. If N is a subterm of M, $N \to N'$, we can replace the subterm N in M by N' to obtain M'. In that case we also write $M \to M'$. If there is a sequence $M = M_1 \to M_2 \to M_2 \to \ldots \to M_k = M'$ (including where $k = 1$, i.e., $M = M'$) we write $M \twoheadrightarrow M'$.

A derivation where no conversion can be applied to any sub-derivation is called *normal*. Similarly, a λ-term where no conversion can be applied (to any sub-term) is called *normal*. A normal λ-term is also called a *normal form*.

Photo Credits

Bibliography