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

$$\dfrac{\dfrac{\dfrac{\varphi \qquad \varphi \to \psi}{\psi}\to\text{Elim} \qquad [\chi]}{\psi \wedge \chi}\wedge\text{Intro}}{\dfrac{\psi}{\chi \to \psi}\wedge\text{Elim}}\to\text{Intro}$$

can be replaced with the simpler derivation:

$$\dfrac{\dfrac{\varphi \qquad \varphi \to \psi}{\psi}\to\text{Elim}}{\chi \to \psi}\to\text{Intro}$$

As we see, an $\wedge$Intro followed by $\wedge$Elim "cancel out." In general, we see that the conclusion of $\wedge$Elim is always the formula on one side of the conjunction, and the premises of $\wedge$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

$$\dfrac{\dfrac{\begin{array}{cc}\vdots & \vdots \\ D_1 & D_2 \\ \varphi_1 & \varphi_2\end{array}}{\varphi_1 \wedge \varphi_2}\wedge\text{Intro}}{\varphi_i}\wedge\text{Elim}_i \quad \triangleright_1 \quad \begin{array}{c}\vdots \\ D_i \\ \varphi_i\end{array}$$

The $\triangleright_1$ 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:

$$(\Gamma, p_i\langle M_1^{\varphi_1}, M_2^{\varphi_2}\rangle) \triangleright_1 (\Gamma, M_i)$$

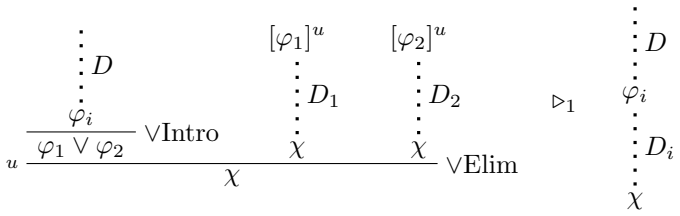
In the typed lambda calculus, this is the beta reduction rule for the product type.

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 $\triangleright_1$ 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 $\langle N, M\rangle$, $\mathrm{p}_i(N)$, $\mathrm{in}_i^{\varphi}(N)$, $\mathrm{case}(N, x_1.M_1, x_2.M_2)$, and $\mathrm{contr}_N()$, with corresponding redexes.

Similarly we have reduction for disjunction:



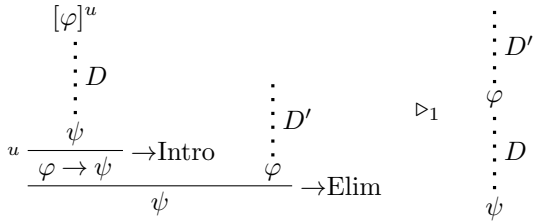
This corresponds to a reduction on proof terms:

$$(\Gamma, \mathrm{case}(\mathrm{in}_i^{\varphi_i}(M^{\varphi_i}), x_1^{\varphi_1}.N_1^{\chi}, x_2^{\varphi_2}.N_2^{\chi})) \triangleright_1 (\Gamma, N_i^{\chi}[M^{\varphi_i}/x_i^{\varphi_i}])$$

This is the beta reduction rule of for sum types. Here, $M[N/x]$ means replacing all assumptions denoted by variable x in M with N,

It would be nice if we pass the context Γ to the substitution function so that it can check if the substitution makes sense. For example, $xy[ab/y]$ does not make sense under the context $\{x : \varphi \to \theta, y : \varphi, a : \psi \to \chi, b : \psi\}$ since then we would be substituting a derivation of χ where a derivation of φ is expected. However, as long as our usage of substitution is careful enough to avoid such errors, we won't have to worry about such conflicts. Thus we can define it recursively as we did for untyped lambda calculus as if we are dealing with untyped terms.

Finally, the reduction of the function type corresponds to removal of a detour of a $\to$Intro followd by a $\to$Elim.



For proof terms, this amounts to ordinary beta reduction:

$$(\Gamma, (\lambda x^{\varphi}.\, N^{\psi})Q^{\varphi}) \triangleright_1 (\Gamma, N^{\psi}[Q^{\varphi}/x^{\varphi}])$$

Absurdity has only an elimination rule and no introduction rule, thus there is no such reduction for it.

Note that the above notion of reduction concerns only deductions with a cut at the end of a derivation. We would of course like to extend it to reduction of cuts anywhere in a derivation, or reductions of subterms of proof terms which constitute redexes. Note that, however, the conclusion of the reduction does not change after reduction, thus we are free to continue applying rules to

both sides of $\triangleright_1$. 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