Chapter udf

Propositions as Types

This is a *very experimental* draft of a chapter on the Curry-Howard correspondence. It needs more explanation and motivation, and there are probably errors and omissions. The proof of normalization should be reviewed and expanded. There are no examples for the product type. Permuation and simplification conversions are not covered. It will make a lot more sense once there is also material on the (typed) lambda calculus which is basically presupposed here. Use with extreme caution.

pty.1 Introduction

int:pty:int:sec

Historically the lambda calculus and intuitionistic logic were developed separately. Haskell Curry and William Howard independently discovered a close similarity: types in a typed lambda calculus correspond to formulas in intuitionistic logic in such a way that a derivation of a formula corresponds directly to a typed lambda term with that formula as its type. Moreover, beta reduction in the typed lambda calculus corresponds to certain transformations of derivations.

For instance, a derivation of $\varphi \to \psi$ corresponds to a term $\lambda x^{\varphi}.\, N^{\psi}$, which has the function type $\varphi \to \psi$. The inference rules of natural deduction correspond to typing rules in the typed lambda calculus, e.g.,

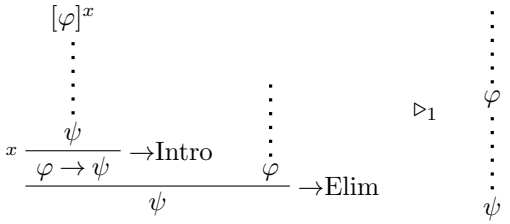
$$\begin{array}{c} [\varphi]^x \\ \vdots \\ \psi \\ \hline \varphi \to \psi \end{array} x \to\text{Intro} \qquad \text{corresponds to} \qquad \frac{x : \varphi \;\Rightarrow\; N : \psi}{\Rightarrow\; \lambda x^{\varphi}.\, N^{\psi} : \varphi \to \psi}\,\lambda$$

where the rule on the right means that if x is of type φ and N is of type ψ, then $\lambda x^{\varphi}.\, N$ is of type $\varphi \to \psi$.

The $\to$Elim rule corresponds to the typing rule for composition terms, i.e.,

$$\dfrac{\dfrac{\varphi \to \psi \qquad \varphi}{\psi}\,{\to}\text{Elim}}{\Rightarrow P : \varphi \to \psi \qquad \Rightarrow Q : \varphi \;\; \text{app}} \quad \text{corresponds to}$$
$$\Rightarrow P^{\varphi\to\psi} Q^{\varphi} : \psi$$

If a $\to$Intro rule is followed immediately by a $\to$Elim rule, the derivation can be simplified:



which corresponds to the beta reduction of lambda terms

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

Similar correspondences hold between the rules for $\wedge$ and "product" types, and between the rules for $\vee$ and "sum" types.

This correspondence between terms in the simply typed lambda calculus and natural deduction derivations is called the "Curry-Howard", or "propositions as types" correspondence. In addition to formulas (propositions) corresponding to types, and proofs to terms, we can summarize the correspondences as follows:

logic	program
proposition	type
proof	term
assumption	variable
discharged assumption	bind variable
not discharged assumption	free variable
implication	function type
conjunction	product type
disjunction	sum type
absurdity	bottom type

The Curry-Howard correspondence is one of the cornerstones of automated proof assistants and type checkers for programs, since checking a proof witnessing a proposition (as we did above) amounts to checking if a program (term) has the declared type.

pty.2 Sequent Natural Deduction

Let us write $\Gamma \Rightarrow \varphi$ if there is a natural deduction derivation with Γ as undischarged assumptions and φ as conclusion; or $\Rightarrow \varphi$ if Γ is empty. int:pty:snd:sec

We write $\Gamma, \varphi_1, \ldots, \varphi_n$ for $\Gamma \cup \{\varphi_1, \ldots, \varphi_n\}$, and Γ, Δ for $\Gamma \cup \Delta$.

Observe that when we have $\Gamma \Rightarrow \varphi \wedge \varphi$, meaning we have a derivation with Γ as undischarged assumptions and $\varphi \wedge \varphi$ as end-formula, then by applying $\wedge$Elim at the bottom, we can get a derivation with the same undischarged assumptions and φ as conclusion. In other words, if $\Gamma \Rightarrow \varphi \wedge \psi$, then $\Gamma \Rightarrow \varphi$.

$$\frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \varphi} \wedge\text{Elim} \qquad \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \psi} \wedge\text{Elim}$$

The label $\wedge$Elim hints at the relation with the rule of the same name in natural deduction.

Likewise, suppose we have $\Gamma, \varphi \Rightarrow \psi$, meaning we have a derivation with undischarged assumptions Γ, φ and end-formula ψ. If we apply the $\rightarrow$Intro rule, we have a derivation with Γ as undischarged assumptions and $\varphi \rightarrow \psi$ as the end-formula, i.e., $\Gamma \Rightarrow \varphi \rightarrow \psi$. Note how this has made the discharge of assumptions more explicit.

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \rightarrow \psi} \rightarrow\text{Intro}$$

We can draw conclusions from other rules in the same fashion, which is spelled out as follows:

$$\frac{\Gamma \Rightarrow \varphi \qquad \Delta \Rightarrow \psi}{\Gamma, \Delta \Rightarrow \varphi \wedge \psi} \wedge\text{Intro}$$

$$\frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \varphi} \wedge\text{Elim}_1 \qquad \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \psi} \wedge\text{Elim}_2$$

$$\frac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \varphi \vee \psi} \vee\text{Intro}_1 \qquad \frac{\Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \vee \psi} \vee\text{Intro}_2$$

$$\frac{\Gamma \Rightarrow \varphi \vee \psi \qquad \Delta, \varphi \Rightarrow \chi \qquad \Delta', \psi \Rightarrow \chi}{\Gamma, \Delta, \Delta' \Rightarrow \chi} \vee\text{Elim}$$

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \rightarrow \psi} \rightarrow\text{Intro} \qquad \frac{\Delta \Rightarrow \varphi \rightarrow \psi \qquad \Gamma \Rightarrow \varphi}{\Gamma, \Delta \Rightarrow \psi} \rightarrow\text{Elim}$$

$$\frac{\Gamma \Rightarrow \bot}{\Gamma \Rightarrow \varphi} \bot_I$$

Any assumption by itself is a derivation of φ from φ, i.e., we always have $\varphi \Rightarrow \varphi$.

$$\frac{}{\varphi \Rightarrow \varphi}$$

Together, these rules can be taken as a calculus about what natural deduction derivations exist. They can also be taken as a notational variant of natural deduction, in which each step records not only the formula derived but also the undischarged assumptions from which it was derived.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi \Rightarrow \varphi}}{\varphi \Rightarrow \varphi \vee (\varphi \to \bot)} \quad \psi \Rightarrow \psi}}{\varphi, \psi \to \Rightarrow \bot}}{(\psi \Rightarrow \varphi \to \bot}}{(\psi \Rightarrow \varphi \vee (\varphi \to \bot)} \quad (\psi \Rightarrow \psi}}{(\psi \Rightarrow \bot}}{\Rightarrow \psi \to \bot}$$

where ψ is short for $(\varphi \vee (\varphi \to \bot)) \to \bot$.

pty.3 Proof Terms

We give the definition of proof terms, and then establish its relation with natural deduction derivations. int:pty:ter:sec

Definition pty.1 (Proof terms)**.** Proof terms are inductively generated by the following rules:

1. A single variable x is a proof term.
2. If P and Q are proof terms, then PQ is also a proof term.
3. If x is a variable, φ is a formula, and N is a proof term, then $\lambda x^{\varphi}. N$ is also a proof term.
4. If P and Q are proof terms, then $\langle P, Q \rangle$ is a proof term.
5. If M is a proof term, then $\mathrm{p}_i(M)$ is also a proof term, where i is 1 or 2.
6. If M is a proof term, and φ is a formula, then $\mathrm{in}_i^{\varphi}(M)$ is a proof term, where i is 1 or 2.
7. If M, N_1, N_2 is proof terms, and x_1, x_2 are variables, then $\mathrm{case}(M, x_1.N_1, x_2.N_2)$ is a proof term.
8. If M is a proof term and φ is a formula, then $\mathrm{contr}_{\varphi}(M)$ is proof term.

Each of the above rules corresponds to an inference rule in natural deduction. Thus we can inductively assign proof terms to the formulas in a derivation. To make this assignment unique, we must distinguish between the two versions of $\wedge$Elim and of $\vee$Intro. For instance, the proof terms assigned to the conclusion of $\vee$Intro must carry the information whether $\varphi \vee \psi$ is inferred from φ or from ψ. Suppose M is the term assigned to $\varphi$from which $\varphi \vee \psi$ is inferred. Then the proof term assigned to $\varphi \vee \psi$ is $\mathrm{in}_1^{\varphi}(M)$. If we instead infer $\psi \vee \varphi$ then the proof term assigned is $\mathrm{in}_2^{\varphi}(M)$.

The term $\lambda x^{\varphi}. N$ is assigned to the conclusion of $\to$Intro. The φ represents the assumption being discharged; only have we included it can we infer the formula of $\lambda x^{\varphi}. N$ based on the formula of N.

Definition pty.2 (Typing context)**.** A *typing context* is a mapping from variables to formulas. We will call it simply the "context" if there is no confusion. We write a context Γ as a set of pairs $\langle x, \varphi \rangle$.

A pair $\Gamma \Rightarrow M$ where M is a proof term represents a derivation of a formula with context Γ.

Definition pty.3 (Typing pair)**.** A *typing pair* is a pair $\langle \Gamma, M \rangle$, where Γ is a typing context and M is a proof term.

Since in general terms only make sense with specific contexts, we will speak simply of "terms" from now on instead of "typing pair"; and it will be apparent when we are talking about the literal term M.

pty.4 Converting Derivations to Proof Terms

int:pty:pt:sec

We will describe the process of converting natural deduction derivations to pairs. We will write a proof term to the left of each formula in the derivation, resulting in expressions of the form $M : \varphi$. We'll then say that, M *witnesses* φ. Let's call such an expression a *judgment*.

First let us assign to each assumption a variable, with the following constraints:

1. Assumptions discharged in the same step (that is, with the same number on the square bracket) must be assigned the same variable.
2. For assumptions not discharged, assumptions of different formulas should be assigned different variables.

Such an assignment translates all assumptions of the form

$$\varphi \qquad \text{into} \qquad x : \varphi.$$

With assumptions all associated with variables (which are terms), we can now inductively translate the rest of the deduction tree. The modified natural deduction rules taking into account context and proof terms are given below. Given the proof terms for the premise(s), we obtain the corresponding proof term for conclusion.

$$\frac{M_1 : \varphi_1 \qquad M_2 : \varphi_2}{\langle M_1, M_2 \rangle : \varphi_1 \wedge \varphi_2} \wedge\text{Intro}$$

$$\frac{M : \varphi_1 \wedge \varphi_2}{\mathrm{p}_i(M) : \varphi_1} \wedge\text{Elim}_1 \qquad \frac{M : \varphi_1 \wedge \varphi_2}{\mathrm{p}_i(M) : \varphi_2} \wedge\text{Elim}_2$$

In $\wedge$Intro we assume we have φ_1 witnessed by term M_1 and φ_2 witnessed by term M_2. We pack up the two terms into a pair $\langle M_1, M_2 \rangle$ which witnesses $\varphi_1 \wedge \varphi_2$.

In $\wedge\text{Elim}_i$ we assume that M witnesses $\varphi_1 \wedge \varphi_2$. The term witnessing φ_i is $\mathrm{p}_i(M)$. Note that M is not necessary of the form $\langle M_1, M_2\rangle$, so we cannot simply assign M_1 to the conclusion φ_i.

Note how this coincides with the BHK interpretation. What the BHK interpretation does not specify is how the function used as proof for $\varphi \to \psi$ is supposed to be obtained. If we think of proof terms as proofs or functions of proofs, we can be more explicit.

$$\frac{\begin{array}{c}[x:\varphi]\\ \vdots \\ N:\psi\end{array}}{\lambda x^{\varphi}.\, N : \varphi \to \psi}\ {\to}\text{Intro} \qquad \frac{P : \varphi \to \psi \qquad Q : \varphi}{PQ : \psi}\ {\to}\text{Elim}$$

The λ notation should be understood as the same as in the lambda calculus, and PQ means applying P to Q.

$$\frac{M_1 : \varphi_1}{\mathrm{in}_1^{\varphi_1}(M_1) : \varphi_1 \vee \varphi_2}\ {\vee}\text{Intro}_1 \qquad \frac{M_2 : \varphi_2}{\mathrm{in}_2^{\varphi_2}(M_2) : \varphi_1 \vee \varphi_2}\ {\vee}\text{Intro}_2$$

$$\frac{M : A_1 \vee \varphi_2 \qquad \begin{array}{c}[x_1:\varphi_1]\\ \vdots \\ N_1:\chi\end{array} \qquad \begin{array}{c}[x_2:\varphi_2]\\ \vdots \\ N_2:\chi\end{array}}{\mathrm{case}(M, x_1.N_1, x_2.N_2) : \chi}\ {\vee}\text{Elim}$$

The proof term $\mathrm{in}_1^{\varphi_1}(M_1)$ is a term witnessing $\varphi_1 \vee \varphi_2$, where M_1 witnesses φ_1.

The term $\mathrm{case}(M, x_1.N_1, x_2.N_2)$ mimics the case clause in programming languages: we already have the derivation of $\varphi \vee \psi$, a derivation of χ assuming φ, and a derivation of χ assuming ψ. The *case* operator thus select the appropriate proof depending on M; either way it's a proof of χ.

$$\frac{N : \bot}{\mathrm{contr}_{\varphi}(N) : \varphi}\ \bot_I$$

$\mathrm{contr}_{\varphi}(N)$ is a term witnessing φ, whenever N is a term witnessing $\bot$.

Now we have a natural deduction derivation with all formulas associated with a term. At each step, the relevant typing context Γ is given by the list of assumptions remaining undischarged at that step. Note that Γ is well defined: since we have forbidden assumptions of different undischarged assumptions to be assigned the same variable, there won't be any disagreement about the formulas mapped to which a variable is mapped.

We now give some examples of such translations:

Consider the derivation of $\neg\neg(\varphi \vee \neg\varphi)$, i.e., $((\varphi \vee (\varphi \to \bot)) \to \bot) \to \bot$. Its translation is:

$$\dfrac{\dfrac{[y : (\varphi \vee (\varphi \to \bot)) \to \bot]^2 \qquad \dfrac{[x : \varphi]^1}{\mathrm{in}_1^{\varphi \to \bot}(x) : \varphi \vee (\varphi \to \bot)}}{\dfrac{y(\mathrm{in}_1^{\varphi \to \bot}(x)) : \bot}{\lambda x^{\varphi}.\, y(\mathrm{in}_1^{\varphi \to \bot}(x)) : \varphi \to \bot}\,1}}{\dfrac{[y : (\varphi \vee (\varphi \to \bot)) \to \bot]^2 \qquad \mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y(\mathrm{in}_1^{\varphi \to \bot}(x))) : \varphi \vee (\varphi \to \bot)}{y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi \to \bot}(x))) : \bot}}$$

$$2\ \dfrac{y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi \to \bot}(x))) : \bot}{\lambda y^{(\varphi \vee (\varphi \to \bot)) \to \bot}.\, y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi \to \bot}(x))) : ((\varphi \vee (\varphi \to \bot)) \to \bot) \to \bot}$$

The tree has no assumptions, so the context is empty; we get:

$$\vdash \lambda y^{(\varphi \vee (\varphi \to \bot)) \to \bot}.\, y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi \to \bot}(x))) : ((\varphi \vee (\varphi \to \bot)) \to \bot) \to \bot$$

If we leave out the last $\to$Intro, the assumptions denoted by y would be in the context and we would get:

$$y : ((\varphi \vee (\varphi \to \bot)) \to \bot) \vdash y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi \to \bot}(x))) : \bot$$

Another example: $\vdash \varphi \to (\varphi \to \bot) \to \bot$

$$2\ \dfrac{1\ \dfrac{[x : \varphi]^2 \qquad [y : \varphi \to \bot]^1}{yx : \bot}}{\lambda y^{\varphi \to \bot}.\, yx : (\varphi \to \bot) \to \bot}}{\lambda x^{\varphi}.\, \lambda y^{\varphi \to \bot}.\, yx : \varphi \to (\varphi \to \bot) \to \bot}$$

Again all assumptions are discharged and thus the context is empty, the resulting term is

$$\vdash \lambda x^{\varphi}.\, \lambda y^{\varphi \to \bot}.\, yx : \varphi \to (\varphi \to \bot) \to \bot$$

If we leave out the last two $\to$Intro inferences, the assumptions denoted by both x and y would be in context and we would get

$$x : \varphi, y : \varphi \to \bot \vdash yx : \bot$$

pty.5 Recovering Derivations from Proof Terms

int:pty:tp:sec

Now let us consider the other direction: translating terms back to natural deduction trees. We will use still use the double refutation of the excluded middle as example, and let S denote this term, i.e.,

$$\lambda y^{(\varphi \vee (\varphi \to \bot)) \to \bot}.\, y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi \to \bot}(x))) : ((\varphi \vee (\varphi \to \bot)) \to \bot) \to \bot$$

For each natural deduction rule, the term in the conclusion is always formed by wrapping some operator around the terms assigned to the premise(s). Rules

correspond uniquely to such operators. For example, from the structure of the S we infer that the last rule applied must be $\to$Intro, since it is of the form $\lambda y^{\cdots}. \ldots$, and the λ operator corresponds to $\to$Intro. In general we can recover the skeleton of the derivation solely by the structure of the term, e.g.,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[x]^1}{\mathrm{in}_1^{\varphi\to\bot}(x):}\vee\mathrm{Intro}_1 \quad [y:]^2}{y(\mathrm{in}_1^{\varphi\to\bot}(x)):}\to\mathrm{Elim}}{\lambda x^{\varphi}.\, y(\mathrm{in}_1^{\varphi\to\bot}(x)):}\to\mathrm{Intro}\,1}{\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi\to\bot}(x)):}\vee\mathrm{Intro}_2 \quad [y:]^2}{y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y\mathrm{in}_1^{\varphi\to\bot}(x))):}\to\mathrm{Elim}}{\lambda y^{(\varphi\vee(\varphi\to\bot))\to\bot}.\, y(\mathrm{in}_2^{\varphi}(\lambda x^{\varphi}.\, y(\mathrm{in}_1^{\varphi\to\bot}(x)))):}\to\mathrm{Intro}\,2$$

Our next step is to recover the formulas these terms witness. We define a function $F(\Gamma, M)$ which denotes the formula witnessed by M in context Γ, by induction on M as follows:

$$\begin{aligned}
F(\Gamma, x) &= \Gamma(x) \\
F(\Gamma, \langle N_1, N_2\rangle &= F(\Gamma, N_1) \wedge F(\Gamma, N_2) \\
F(\Gamma, \mathrm{p}_i(N) &= \varphi_i \text{ if } F(\Gamma, N) = \varphi_1 \wedge \varphi_2 \\
F(\Gamma, \mathrm{in}_i^{\varphi}(N) &= \begin{cases} F(N) \vee \varphi & \text{if } i = 1 \\ \varphi \vee F(N) & \text{if } i = 2 \end{cases} \\
F(\Gamma, \mathrm{case}(M, x_1.N_1, x_2.N_2)) &= F(\Gamma \cup \{x_i : F(\Gamma, M)\}, N_i) \\
F(\Gamma, \lambda x^{\varphi}.\, N) &= \varphi \to F(\Gamma \cup \{x : \varphi\}, N) \\
F(\Gamma, NM) &= \psi \text{ if } F(\Gamma, N) = \varphi \to \psi
\end{aligned}$$

where $\Gamma(x)$ means the formula mapped to by x in Γ and $\Gamma \cup \{x : \varphi\}$ is a context exactly as Γ except mapping x to φ, whether or not x is already in Γ.

Note there are cases where $F(\Gamma, M)$ is not defined, for example:

1. In the first line, it is possible that x is not in Γ.
2. In recursive cases, the inner invocation may be undefined, making the outer one undefined too.
3. In the third line, its only defined when $F(\Gamma, M)$ is of the form $\varphi_1 \vee \varphi_2$, and the right hand is independent on i.

As we recursively compute $F(\Gamma, M)$, we work our way up the natural deduction derivation. The every step in the computation of $F(\Gamma, M)$ corresponds to a term in the derivation to which the derivation-to-term translation assigns M, and the formula computed is the end-formula of the derivation. However, the result may not be defined for some choices of Γ. We say that such pairs $\langle \Gamma, M\rangle$ are *ill-typed*, and otherwise *well-typed*. However, if the term M results from

translating a derivation, and the formulas in Γ correspond to the undischarged assumptions of the derivation, the pair $\langle \Gamma, M \rangle$ will be well-typed.

Proposition pty.4. *If D is a derivation with undischarged assumptions φ_1, …, φ_n, M is the proof term associated with D and $\Gamma = \{x_1 : \varphi_1, \ldots, x_n : \varphi_n\}$, then the result of recovering derivation from M in context Γ is D.*

In the other direction, if we first translate a typing pair to natural deduction and then translate it back, we won't get the same pair back since the choice of variables for the undischarged assumptions is underdetermined. For example, consider the pair $\langle \{x : \varphi, y : \varphi \to \psi\}, yx \rangle$. The corresponding derivation is

$$\frac{\varphi \to \psi \qquad \varphi}{\psi} \to\text{Elim}$$

By assigning different variables to the undischarged assumptions, say, u to $\varphi \to \psi$ and v to φ, we would get the term uv rather than yx. There is a connection, though: the terms will be the same up to renaming of variables.

Now we have established the correspondence between typing pairs and natural deduction, we can prove theorems for typing pairs and transfer the result to natural deduction derivations.

Similar to what we did in the natural deduction section, we can make some observations here too. Let $\Gamma \vdash M : \varphi$ denote that there is a pair (Γ, M) witnessing the formula φ. Then always $\Gamma \vdash x : \varphi$ if $x : \varphi \in \Gamma$, and the following rules are valid:

$$\frac{\Gamma \vdash M_1 : \varphi_1 \qquad \Delta \vdash M_2 : \varphi_2}{\Gamma, \Delta \vdash \langle M_1, M_2 \rangle : \varphi_1 \wedge \varphi_2} \wedge\text{Intro} \qquad \frac{\Gamma \vdash M : \varphi_1 \wedge \varphi_2}{\Gamma \vdash \mathrm{p}_i(M) : \varphi_i} \wedge\text{Elim}_i$$

$$\frac{\Gamma \vdash M_1 : \varphi_1}{\Gamma \vdash \mathrm{in}_1^{\varphi_2}(M) : \varphi_1 \vee \varphi_2} \vee\text{Intro}_1 \qquad \frac{\Gamma \vdash M_2 : \varphi_2}{\Gamma \vdash \mathrm{in}_2^{\varphi_1}(M) : \varphi_1 \vee \varphi_2} \vee\text{Intro}_2$$

$$\frac{\Gamma \vdash M : \varphi \vee \psi \qquad \Delta_1, x_1 : \varphi_1 \vdash N_1 : \chi \qquad \Delta_2, x_2 : \varphi_2 \vdash N_2 : \chi}{\Gamma, \Delta, \Delta' \vdash \mathrm{case}(M, x_1.N_1, x_2.N_2) : \chi} \vee\text{Elim}$$

$$\frac{\Gamma, x : \varphi \vdash N : \psi}{\Gamma \vdash \lambda x^{\varphi}.\, N : \varphi \to \psi} \to\text{Intro} \qquad \frac{\Gamma \vdash Q : \varphi \qquad \Delta \vdash P : \varphi \to \psi}{\Gamma, \Delta \vdash PQ : \psi} \to\text{Elim}$$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \mathrm{contr}_{\varphi}(M) : \varphi} \bot\text{Elim}$$

These are the typing rules of the simply typed lambda calculus extended with product, sum and bottom.

In addition, the $F(\Gamma, M)$ is actually a type checking algorithm; it returns the type of the term with respect to the context, or is undefined if the term is ill-typed with respect to the context.

pty.6 Reduction

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

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

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

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

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:

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

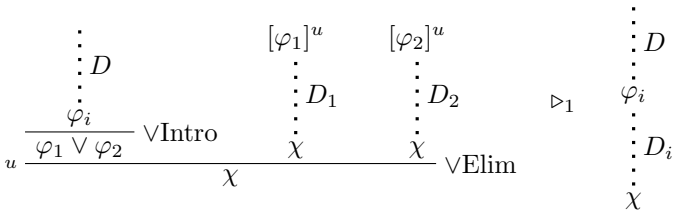
In the typed lambda calculus, this is the beta reduction rule for the product type.

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:



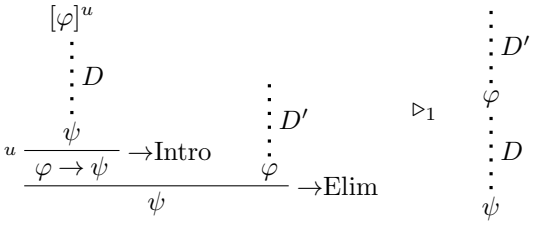
This corresponds to a reduction on proof terms:

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

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.5. 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.

pty.7 Normalization

int:pty:nor:sec

In this section we prove that, via some reduction order, any deduction can be reduced to a normal deduction, which is called the *normalization property*. We will make use of the propositions-as-types correspondence: we show that every proof term can be reduced to a normal form; normalization for natural deduction derivations then follows.

Firstly we define some functions that measure the complexity of terms. The *length* $\mathrm{len}(\varphi)$ of a formulas is defined by

$$\begin{aligned}\mathrm{len}(p) &= 0\\ \mathrm{len}(\varphi \wedge \psi) &= \mathrm{len}(\varphi) + \mathrm{len}(\psi) + 1\\ \mathrm{len}(\varphi \vee \psi) &= \mathrm{len}(\varphi) + \mathrm{len}(\psi) + 1\\ \mathrm{len}(\varphi \to \psi) &= \mathrm{len}(\varphi) + \mathrm{len}(\psi) + 1.\end{aligned}$$

The complexity of a redex M is measured by its *cut rank* $\mathrm{cr}(M)$:

$$\begin{aligned}\mathrm{cr}((\lambda x^{\varphi}.\, N^{\psi})Q) &= \mathrm{len}(\varphi) + \mathrm{len}(\psi) + 1\\ \mathrm{cr}(\mathrm{p}_i(\langle M^{\varphi}, N^{\psi}\rangle)) &= \mathrm{len}(\varphi) + \mathrm{len}(\psi) + 1\\ \mathrm{cr}(\mathrm{case}(\mathrm{in}_i^{\varphi_i}(M^{\varphi_i}), x_1^{\varphi_1}.N_1^{X}, x_2^{\varphi_2}.N_2^{X})) &= \mathrm{len}(\varphi) + \mathrm{len}(\psi) + 1\end{aligned}$$

The complexity of a proof term is measured by the most complex redex in it, and 0 if it is normal:

$$\mathrm{mr}(M) = \max\{\mathrm{cr}(N) | N \text{ is a sub term of } M \text{ and is redex}\}$$

Lemma pty.6. *If $M[N^{\varphi}/x^{\varphi}]$ is a redex and $M \not\equiv x$, then one of the following cases holds:* int:pty:nor:lem:subst

1. *M is itself a redex, or*
2. *M is of the form $\mathrm{p}_i(x)$, and N is of the form $\langle P_1, P_2\rangle$*

3. *M is of the form* $\mathrm{case}(i, x_1.P_1, x_2.P_2)$*, and N is of the form* $\mathrm{in}_i(Q)$

4. *M is of the form xQ, and N is of the form $\lambda x.\, P$*

In the first case, $\mathrm{cr}(M[N/x]) = \mathrm{cr}(M)$*; in the other cases,* $\mathrm{cr}(M[N/x]) = \mathrm{len}(\varphi)$).

Proof. Proof by induction on M.

1. If M is a single variable y and $y \not\equiv x$, then $y[N/x]$ is y, hence not a redex.

2. If M is of the form $\langle N_1, N_2\rangle$, or $\lambda x.\, N$, or $\mathrm{in}_i^{\varphi}(N)$, then $M[N^{\varphi}/x^{\varphi}]$ is also of that form, and so is not a redex.

3. If M is of the form $\mathrm{p}_i(P)$, we consider two cases.

   a) If P is of the form $\langle P_1, P_2\rangle$, then $M \equiv \mathrm{p}_i(\langle P_1, P_2\rangle)$ is a redex, and clearly
   $$M[N/x] \equiv \mathrm{p}_i(\langle P_1[N/x], P_2[N/x]\rangle)$$
   is also a redex. The cut ranks are equal.

   b) If P is a single variable, it must be x to make the substitution a redex, and N must be of the form $\langle P_1, P_2\rangle$. Now consider
   $$M[N/x] \equiv \mathrm{p}_i(x)[\langle P_1, P_2\rangle/x],$$
   which is $\mathrm{p}_i(\langle P_1, P_2\rangle)$. Its cut rank is equal to $\mathrm{cr}(x)$, which is $\mathrm{len}(\varphi)$.

The cases of $\mathrm{case}(N, x_1.N_1, x_2.N_2)$ and PQ are similar. □

Lemma pty.7. *If M contracts to M', and* $\mathrm{cr}(M) > \mathrm{cr}(N)$ *for all proper redex sub-terms N of M, then* $\mathrm{cr}(M) > \mathrm{mr}(M')$.

Proof. Proof by cases.

1. If M is of the form $\mathrm{p}_i(\langle M_1, M_2\rangle)$, then M' is M_i; since any sub-term of M_i is also proper sub-term of M, the claim holds.

2. If M is of the form $(\lambda x^{\varphi}.\, N)Q^{\varphi}$, then M' is $N[Q^{\varphi}/x^{\varphi}]$. Consider a redex in M'. Either there is corresponding redex in N with equal cut rank, which is less than $\mathrm{cr}(M)$ by assumption, or the cut rank equals $\mathrm{len}(\varphi)$, which by definition is less than $\mathrm{cr}((\lambda x^{\varphi}.\, N)Q)$.

3. If M is of the form
   $$\mathrm{case}(\mathrm{in}_i(N^{\varphi_i}), x_1^{\varphi_1}.N_1^{\chi}, x_2^{\varphi_2}.N_2^{\chi}),$$
   then $M' \equiv N_i[N/x_i^{\varphi_i}]$. Consider a redex in M'. Either there is corresponding redex in N_i with equal cut rank, which is less than $\mathrm{cr}(M)$ by assumption; or the cut rank equals $\mathrm{len}(\varphi_i)$, which by definition is less than $\mathrm{cr}(\mathrm{case}(\mathrm{in}_i(N^{\varphi_i}), x_1^{\varphi_1}.N_1^{\chi}, x_2^{\varphi_2}.N_2^{\chi}))$.

□

Theorem pty.8. *All proof terms reduce to normal form; all derivations reduce to normal derivations.*

Proof. The second follows from the first. We prove the first by complete induction on $m = \mathrm{mr}(M)$, where M is a proof term.

1. If $m = 0$, M is already normal.
2. Otherwise, we proceed by induction on n, the number of redexes in M with cut rank equal to m.
   a) If $n = 1$, select any redex N such that $m = \mathrm{cr}(N) > \mathrm{cr}(P)$ for any proper sub-term P which is also a redex of course. Such a redex must exist, since any term only has finitely many subterms.
   Let N' denote the reductum of N. Now by the lemma $\mathrm{mr}(N') < \mathrm{mr}(N)$, thus we can see that n, the number of redexes with $\mathrm{cr}(=)m$ is decreased. So m is decreased (by 1 or more), and we can apply the inductive hypothesis for m.
   b) For the induction step, assume $n > 1$. the process is similar, except that n is only decreased to a positive number and thus m does not change. We simply apply the induction hypothesis for n.

□

The normalization of terms is actually not specific to the reduction order we chose. In fact, one can prove that regardless of the order in which redexes are reduced, the term always reduces to a normal form. This property is called *strong normalization*.

Photo Credits

Bibliography