Normalization by Evaluation for Intuitionistic Propositional Logic

Andreas Abel

July 2018

1 Intuitionistic Propositional Logic (IPL)

Formulas and hypotheses lists (contexts).

$$
\begin{array}{llllll}
\mathsf{Atom} & \ni & P, Q & & & \text{atomic propositions} \\
\mathsf{Form} & \ni & A, B, C, \dots & ::= & \top \mid A \wedge B \mid A \Rightarrow B & \text{negative formulas} \\
 & & & \mid & \bot \mid A \vee B \mid P & \text{positive formulas} \\
 \\
\mathsf{Cxt} & \ni & \Gamma, \Delta & ::= & \emptyset & \text{empty context} \\
 & & & \mid & \Gamma.A & \text{context extension}
\end{array}
$$

We write just A for the singleton context $\emptyset.A$.

Indices x into the context Γ of hypothesis, locating hypothesis A, are written $x : \mathsf{Hyp}(A)_\Gamma$ and defined inductively by the following constructors:

$$
\frac{}{\mathsf{top} : \mathsf{Hyp}(A)_{\Gamma.A}} \qquad \frac{x : \mathsf{Hyp}(A)_\Gamma}{\mathsf{pop}_B\, x : \mathsf{Hyp}(A)_{\Gamma.B}}
$$

Derivations t of the truth of a formula A under assumptions Γ (judgement $\Gamma \vdash A$) are written $t : \mathsf{Tm}(A)_\Gamma$ and defined inductively as follows:

Implication and hypotheses.

$$
\frac{x : \mathsf{Hyp}(A)_\Gamma}{\mathsf{hyp}\, x : \mathsf{Tm}(A)_\Gamma} \qquad \frac{t : \mathsf{Tm}(B)_{\Gamma.A}}{\mathsf{impI}\, t : \mathsf{Tm}(A \Rightarrow B)_\Gamma} \qquad \frac{t : \mathsf{Tm}(A \Rightarrow B)_\Gamma \qquad u : \mathsf{Tm}(A)_\Gamma}{\mathsf{impE}\, t\, u : \mathsf{Tm}(B)_\Gamma}
$$

Truth and conjunction.

$$
\frac{}{\mathsf{trueI} : \mathsf{Tm}(\top)_\Gamma} \qquad \frac{t : \mathsf{Tm}(A)_\Gamma \qquad u : \mathsf{Tm}(B)_\Gamma}{\mathsf{andI}\, t\, u : \mathsf{Tm}(A \wedge B)_\Gamma}
$$

$$
\frac{t : \mathsf{Tm}(A \wedge B)_\Gamma}{\mathsf{andE}_1\, t : \mathsf{Tm}(A)_\Gamma} \qquad \frac{t : \mathsf{Tm}(A \wedge B)_\Gamma}{\mathsf{andE}_2\, t : \mathsf{Tm}(B)_\Gamma}
$$

Absurdity and disjunction.

$$\frac{t : \mathsf{Tm}(A)_\Gamma}{\mathsf{orI}_1\, t : \mathsf{Tm}(A \vee B)_\Gamma} \qquad \frac{t : \mathsf{Tm}(B)_\Gamma}{\mathsf{orI}_2\, t : \mathsf{Tm}(A \vee B)_\Gamma}$$

$$\frac{t : \mathsf{Tm}(\bot)_\Gamma}{\mathsf{falseE}\ t : \mathsf{Tm}(C)_\Gamma} \qquad \frac{t : \mathsf{Tm}(A \vee B)_\Gamma \qquad u : \mathsf{Tm}(C)_{\Gamma.A} \qquad v : \mathsf{Tm}(C)_{\Gamma.B}}{\mathsf{orE}\ t\ u\ v : \mathsf{Tm}(C)_\Gamma}$$

2 Subformula property and normal derivations

For every derivable judgement $\Gamma \vdash A$ there are infinitely many derivations $t : \mathsf{Tm}(A)_\Gamma$, since arbitrary detours are allowed. For instance to prove $\top$ we can also proceed very indirectly by introducing a hypothesis $\top$ and then eliminating it by proof trueI:

$$\mathsf{impE}\ (\mathsf{impI}\ (\mathsf{hyp}\ \mathsf{top}))\ \mathsf{trueI} : \mathsf{Tm}(\top)_\emptyset.$$

A sensible restriction is that the proof rules guarantee that only *subformulas* of A and the hypotheses in Γ are mentioned when deriving $\Gamma \vdash A$.

The following indexed grammar for *normal* derivations $\mathsf{Nf}(A)_\Gamma$ of judgement $\Gamma \vdash A$ ensures the subformula property. It is defined mutually with a grammar $\mathsf{Ne}(A)_\Gamma$ of *neutral* derivations. Neutral derivations are "straight-line" consequences of hypotheses (without case distinction or absurdity elimination).

Implication and hypotheses.

$$\frac{x : \mathsf{Hyp}(A)_\Gamma}{\mathsf{hyp}\ x : \mathsf{Ne}(A)_\Gamma} \qquad \frac{t : \mathsf{Nf}(B)_{\Gamma.A}}{\mathsf{impI}\ t : \mathsf{Nf}(A \Rightarrow B)_\Gamma} \qquad \frac{t : \mathsf{Ne}(A \Rightarrow B)_\Gamma \qquad u : \mathsf{Nf}(A)_\Gamma}{\mathsf{impE}\ t\ u : \mathsf{Ne}(B)_\Gamma}$$

Truth and conjunction.

$$\frac{}{\mathsf{trueI} : \mathsf{Nf}(\top)_\Gamma} \qquad \frac{t : \mathsf{Nf}(A)_\Gamma \qquad u : \mathsf{Nf}(B)_\Gamma}{\mathsf{andI}\ t\ u : \mathsf{Nf}(A \wedge B)_\Gamma}$$

$$\frac{t : \mathsf{Ne}(A \wedge B)_\Gamma}{\mathsf{andE}_1\, t : \mathsf{Ne}(A)_\Gamma} \qquad \frac{t : \mathsf{Ne}(A \wedge B)_\Gamma}{\mathsf{andE}_2\, t : \mathsf{Ne}(B)_\Gamma}$$

Absurdity and disjunction.

$$\frac{t : \mathsf{Nf}(A)_\Gamma}{\mathsf{orI}_1\, t : \mathsf{Nf}(A \vee B)_\Gamma} \qquad \frac{t : \mathsf{Nf}(B)_\Gamma}{\mathsf{orI}_2\, t : \mathsf{Nf}(A \vee B)_\Gamma}$$

$$\frac{t : \mathsf{Ne}(\bot)_\Gamma}{\mathsf{falseE}\ t : \mathsf{Nf}(C)_\Gamma} \qquad \frac{t : \mathsf{Ne}(A \vee B)_\Gamma \qquad u : \mathsf{Nf}(C)_{\Gamma.A} \qquad v : \mathsf{Nf}(C)_{\Gamma.B}}{\mathsf{orE}\ t\ u\ v : \mathsf{Nf}(C)_\Gamma}$$

Embedding.

$$\frac{t : \mathsf{Ne}(P)_\Gamma}{\mathsf{ne}\ t : \mathsf{Nf}(P)_\Gamma}$$

The introduction rules generate normal forms (Nfs) from Nfs. The elimination rules for negative formulas generate neutrals (Nes) from Nes, where impE requires a side argument in normal form.

Some care is needed with the elimination rules for positive formulas which are falseE and orE. These prove an arbitrary formula C, which, for the sake of the subformula property, should not be subject to further elimination but directly prove the goal. Thus, in addition to restriction the elimination to neutrals (perform case distinction only on neutrals), we require them to produce a normal form in case of orE from normal branches.

Any neutral derivation at atomic proposition P is considered normal. The restriction on atoms is not needed for the subformula property, but forces derivations to be η-long. We could drop the restriction, but even with this restriction, the calculus is complete, meaning that for every derivation (Tm) there exists a normal derivation (Nf). This statement is called *normalization* and we will prove it constructively in the following.

3 Categories and Presheaves

We define order-preserving embeddings (OPEs) $\tau : (\Delta \leq \Gamma)$ of a context Γ into a larger context Δ inductively by the following constructors.

$$\frac{}{\mathsf{id}_\Gamma : \Gamma \leq \Gamma} \qquad \frac{\tau : \Delta \leq \Gamma}{\mathsf{weak}_A\, \tau : \Delta.A \leq \Gamma} \qquad \frac{\tau : \Delta \leq \Gamma}{\mathsf{lift}_A\, \tau : \Delta.A \leq \Gamma.A}$$

For example $\mathsf{weak}_D\,(\mathsf{lift}_C\,(\mathsf{weak}_B\,\mathsf{id}_A)) : A.B.C.D \leq A.C$. OPEs let us introduce extra, unused hypotheses into the context. Derivations of $\Gamma \vdash A$ can be weakened to derivations $\Delta \vdash A$ for $\Delta \leq \Gamma$.

OPEs are closed under composition: If $\tau : \Delta \leq \Gamma$ and $\tau' : \Phi \leq \Delta$ then $\tau \circ \tau' : \Phi \leq \Delta$. Let OPE be the category of contexts (as objects) and OPEs (as morphisms).

Exercise 1 *Define composition of OPEs by induction and show that* OPE *is indeed a category.*

The context-indexed sets $\mathsf{Hyp}(A)$, $\mathsf{Tm}(A)$, $\mathsf{Nf}(A)$, and $\mathsf{Ne}(A)$ are closed under weakening. E. g. Hyp: for each $\tau : \Delta \leq \Gamma$ we have a function

$$\mathsf{Hyp}(A)_\tau : \mathsf{Hyp}(A)_\Gamma \to \mathsf{Hyp}(A)_\Delta,$$

i. e., a morphism in the category SET of sets and functions. The collection $\mathsf{Hyp}(A) : (\tau : \Delta \leq \Gamma) \to (\mathsf{Hyp}(A)_\Gamma \to \mathsf{Hyp}(A)_\Delta)$ of these function constitutes a contravariant functor $\mathsf{Hyp}(A) : \mathsf{OPE} \to \mathsf{SET}$ mapping OPEs to functions. In other words, $\mathsf{Hyp}(A)$ is a *presheaf* over OPE for each A, and so are $\mathsf{Tm}(A)$, $\mathsf{Nf}(A)$, and $\mathsf{Ne}(A)$.

Exercise 2 *Prove that* $\mathsf{Hyp}(A)$*,* $\mathsf{Tm}(A)$*,* $\mathsf{Nf}(A)$*, and* $\mathsf{Ne}(A)$ *are presheaves over* OPE *by defining the map functions and proving the functor laws!*

A function f between presheaves $\mathcal{A}$ and $\mathcal{B}$ is defined pointwise; we write $f : \mathcal{A} \dot{\to} \mathcal{B}$ for the Cxt-indexed collection

$$f_\Gamma : \mathcal{A}_\Gamma \to \mathcal{B}_\Gamma$$

of functions (morphisms in SET). Presheaves and functions between them make a category by virtue of pointwise identity and composition. However, typically the functions we consider are *natural* in the context index, i. e., they are *natural transformations* between presheaves, which means that they commute with OPEs. Given $\tau : \Delta \leq \Gamma$ and $f : \mathcal{A} \dot{\to} \mathcal{B}$, naturality means

$$f_\Delta \circ \mathcal{A}_\tau = \mathcal{B}_\tau \circ f_\Gamma.$$

In words, first weakening with τ and then applying f has the same effect as applying f first and perform the weakening later. This guarantees some form of *parametricity* of f in the context, in particular, f cannot make decisions based on the length of the context or its precise contents.

Presheaves over OPE and natural transformations form a category PSh, traditionally called $\widehat{\mathsf{OPE}}$.

Exercise 3 *Prove that* PSh *is indeed a category.*

Exercise 4 *Prove that the following functions are morphisms in* PSh*:*

$$\begin{array}{lcllcl} \mathsf{hyp} & : & \mathsf{Hyp}(A) & \dot{\to} & \mathsf{Ne}(A) \\ \mathsf{andE}_1 & : & \mathsf{Ne}(A \wedge B) & \dot{\to} & \mathsf{Ne}(A) \\ \mathsf{orI}_1 & : & \mathsf{Nf}(A) & \dot{\to} & \mathsf{Nf}(A \vee B) \\ \mathsf{falseE} & : & \mathsf{Ne}(\bot) & \dot{\to} & \mathsf{Nf}(C) \end{array}$$

Find other examples of presheaf morphisms.

4 NbE for the negative fragment

We wish to define a normalization function for derivations of any formula A

$$\mathsf{norm}^A : \mathsf{Tm}(A) \dot{\to} \mathsf{Nf}(A).$$

Normalization by evaluation (NbE) achieves normalization by first evaluating terms, obtaining values and functions that compute by means of the computation in the meta language, i.e., in SET. As a second step, these values are reified back into syntax, yielding a normal form. Technically, we define a suitable presheaf $[\![A]\!]$ for each formula A and two functions

$$\begin{array}{lcllcl} \mathsf{eval}^A & : & \mathsf{Tm}(A) & \dot{\to} & [\![A]\!] \\ \mathsf{reify}^A & : & [\![A]\!] & \dot{\to} & \mathsf{Nf}(A) \end{array}$$

which compose to norm.

4.1 Semantics, reflection and reification

For now, let us focus on the negative formulas $\top$, $A \wedge B$, $A \Rightarrow B$ and atoms P.

$$\begin{array}{lcllcl} [\![P]\!] & = & \mathsf{Nf}(P) & \mathsf{reify}^P_\Gamma\, t & = & t \\ [\![\top]\!] & = & \hat{1} & \mathsf{reify}^\top_\Gamma\, \_ & = & \mathsf{trueI} \\ [\![A \wedge B]\!] & = & [\![A]\!] \mathbin{\hat{\times}} [\![B]\!] & \mathsf{reify}^{A \wedge B}_\Gamma\, (a, b) & = & \mathsf{andI}\ (\mathsf{reify}^A_\Gamma\, a)\ (\mathsf{reify}^B_\Gamma\, b) \end{array}$$

Herein, we use the unit presheaf $\hat{1}_\Gamma = 1$ where 1 is the unit set, and the pointwise product of presheaves $(\mathcal{A} \mathbin{\hat{\times}} \mathcal{B})_\Gamma = \mathcal{A}_\Gamma \times \mathcal{B}_\Gamma$. We note that by these definitions, the category PSh has products, with the pointwise terminal morphisms:

$$\begin{array}{lcllcl} ! & : & \mathcal{A} \mathrel{\dot{\to}} \hat{1} & [\_,\_] & : & (f : \mathcal{C} \mathrel{\dot{\to}} \mathcal{A})(g : \mathcal{C} \mathrel{\dot{\to}} \mathcal{B}) \to (\mathcal{C} \mathrel{\dot{\to}} \mathcal{A} \mathbin{\hat{\times}} \mathcal{B}) \\ !_\Gamma(a) & = & () & [f,g]_\Gamma(c) & = & (f(c), g(c)) \end{array}$$

The interpretation $[\![A \Rightarrow B]\!]$ of implication will follow the Brouwer-Heyting-Kolmogorov (BHK) interpretation of intuitionistic logic:

A proof of $A \Rightarrow B$ is a method turning any proof of A into a proof of B.

However, the direct lifting of the function space $[\![A \Rightarrow B]\!]_\Gamma = [\![A]\!]_\Gamma \to [\![B]\!]_\Gamma$ does not work. It does not give a presheaf because of the contravariant occurrence of presheaf $[\![A]\!]$. Instead, we force monotonicity by quantifying over all OPEs of Γ:

$$[\![A \Rightarrow B]\!]_\Gamma = \prod_{\Delta:\mathsf{Cxt}} \prod_{\tau:\Delta\leq\Gamma} ([\![A]\!]_\Delta \to [\![B]\!]_\Delta)$$

Reification is also challenging. Let $f : [\![A \Rightarrow B]\!]_\Gamma$ and attempt

$$\mathsf{reify}^{A\Rightarrow B}_\Gamma\, f = \mathsf{impI}\ (\mathsf{reify}^B_{\Gamma.A}\,(f_{\Gamma.A}\ (\mathsf{weak}_A\ \mathsf{id}_\Gamma)\ ?))$$

where hole "?" should be filled with an element of $[\![B]\!]_{\Gamma.A}$ which intuitively should represent the new hypothesis $\top : \mathsf{Hyp}(A)_{\Gamma.A}$ introduced by impI. Thus we need a natural transformation $\mathsf{Hyp}(A) \mathrel{\dot{\to}} [\![A]\!]$ which *reflects* variables into the semantics. In order to define it by induction on A, we need to generalize it to

$$\mathsf{reflect}^A : \mathsf{Ne}(A) \mathrel{\dot{\to}} [\![A]\!].$$

Let us make sure we can define it at atoms and conjunctions:

$$\begin{array}{lcllcl} \mathsf{reflect}^P & = & \mathsf{ne} & : & \mathsf{Ne}(P) \mathrel{\dot{\to}} \mathsf{Nf}(P) \\ \mathsf{reflect}^\top & = & ! & : & \mathsf{Ne}(\top) \mathrel{\dot{\to}} \hat{1} \\ \mathsf{reflect}^{A\wedge B} & = & \langle\, \mathsf{reflect}^A \circ \mathsf{andE}_1\,,\ \mathsf{reflect}^B \circ \mathsf{andE}_2\,\rangle & : & \mathsf{Ne}(A \wedge B) \mathrel{\dot{\to}} [\![A]\!] \mathbin{\hat{\times}} [\![B]\!] \end{array}$$

Observe the eliminations andE_i introduced to reflect conjunctive hypotheses!

Now, we can complete the story for implication. Let $\tau : \Delta \leq \Gamma$ and $a : [\![A]\!]_\Delta$ and set:

$$\begin{array}{lcl} \mathsf{reflect}^{A\Rightarrow B} & : & \mathsf{Ne}(A \Rightarrow B) \mathrel{\dot{\to}} [\![A \Rightarrow B]\!] \\ \mathsf{reflect}^{A\Rightarrow B}_\Gamma\ t\ \Delta\ \tau\ a & = & \mathsf{reflect}^B_\Delta\ (\mathsf{impE}\ (\mathsf{Ne}(B)_\tau t)\ (\mathsf{reify}^A_\Delta\ a)) \\ \mathsf{reify}^{A\Rightarrow B} & : & [\![A \Rightarrow B]\!] \mathrel{\dot{\to}} \mathsf{Nf}(A \Rightarrow B) \\ \mathsf{reify}^{A\Rightarrow B}_\Gamma\ f & = & \mathsf{impI}\ (\mathsf{reify}^B_{\Gamma.A}\ (f_{\Gamma.A}\ (\mathsf{weak}_A\ \mathsf{id}_\Gamma)\ (\mathsf{reflect}^A_{\Gamma.A}\ (\mathsf{hyp}\ \mathsf{top})))) \end{array}$$

Exercise 5 *Let $A = ((P \Rightarrow Q) \wedge (Q \Rightarrow \top)) \Rightarrow Q$ for some atoms P, Q. Compute* $\mathsf{reflect}^A_A\,(\mathsf{hyp\ top})$.

Exercise 6 *Let $A = (P \Rightarrow Q) \Rightarrow (P \Rightarrow Q)$ for some atoms P, Q. Compute* $\mathsf{reify}^A \circ \mathsf{reflect}^A$.

Exercise 7 *Using the constructions in this section, show that the category of presheaves and natural transformations is Cartesian closed.*

4.2 Evaluation

Evaluation is turning an expression $t : \mathsf{Tm}(A)_\Gamma$ into a value in $[\![A]\!]$. However, a direct definition $\mathsf{eval}^A : \mathsf{Tm}(A) \dot{\to} [\![A]\!]$ by induction on the expression fails in case impI:

$$\begin{array}{lcl} \mathsf{eval}^{A\Rightarrow B}_\Gamma & : & \mathsf{Tm}(A)_\Gamma \to [\![A \Rightarrow B]\!]_\Gamma \\ \mathsf{eval}^{A\Rightarrow B}\,(\mathsf{impI}\ t)\,_\Delta\, a & = & ? \end{array}$$

To be able to pass the argument a to the function $\mathsf{eval}\,(\mathsf{impI}\ t)$, we have to generalize evaluation to take an valuation of the context into account.

$$\mathsf{eval}^{A\Rightarrow B}_\Gamma(t) \ : \ [\![\Gamma]\!] \dot{\to} [\![A \Rightarrow B]\!]$$

Here, $[\![\Gamma]\!]$ is the presheaf obtained as the product of the hypotheses contained in Γ:

$$\begin{array}{lcl} [\![\emptyset]\!] & = & \hat{1} \\ [\![\Gamma.A]\!] & = & [\![\Gamma]\!] \mathbin{\hat{\times}} [\![A]\!] \end{array}$$

Looking up values in the environment is the following family of natural transformations:

$$\begin{array}{lcl} \mathsf{lookup}^A_\Gamma & : & \mathsf{Hyp}(A)_\Gamma \to ([\![\Gamma]\!] \dot{\to} [\![A]\!]) \\ \mathsf{lookup}^A_{\Gamma.A}(\mathsf{top}) \quad {}_\Delta\,(\gamma, a) & = & a \\ \mathsf{lookup}^A_{\Gamma.B}(\mathsf{pop}\ x)\,{}_\Delta\,(\gamma, a) & = & \mathsf{lookup}^A_\Gamma(x)\,{}_\Delta\,\gamma \end{array}$$

Exercise 8 *Express* lookup *via the projections $\pi_i : \mathcal{A}_1 \mathbin{\hat{\times}} \mathcal{A}_2 \dot{\to} \mathcal{A}_i$ out of the presheaf product.*

We are ready to define evaluation:

$$\begin{array}{lcl} \mathsf{eval}^A_\Gamma(\mathsf{hyp}\ x) & = & \mathsf{lookup}^A_\Gamma(x) \\ \mathsf{eval}^{A\Rightarrow B}_\Gamma(\mathsf{impI}\ t)\,{}_\Delta\,\gamma\,{}_{\Delta'}\,(\tau : \Delta' \le \Delta)\, a & = & \mathsf{eval}^B_{\Gamma.A}(t)\,{}_{\Delta'}\,([\![\Gamma]\!]_\tau \gamma, a) \\ \mathsf{eval}^B_\Gamma(\mathsf{impE}\ t\ u)\,{}_\Delta\,\gamma & = & \mathsf{eval}^{A\Rightarrow B}_\Gamma(t)\,{}_\Delta\,\gamma\,{}_\Delta\,\mathsf{id}_\Delta\,(\mathsf{eval}^A_\Gamma(u)\,{}_\Delta\,\gamma) \end{array}$$

Exercise 9 *Complete the definition of* eval *for conjunctions.*

Exercise 10 *Express* eval *in terms of the operations of the CCC of presheaves!*

4.3 Normalization

Normalization is reification after evaluation in the identity environment:

$$\begin{array}{lcl} \mathsf{norm}^A & : & \mathsf{Tm}(A) \dot{\to} \mathsf{Nf}(A) \\ \mathsf{norm}^A_\Gamma\, t & = & \mathsf{reify}^A_\Gamma\,(\mathsf{eval}^A_\Gamma(t)\,{}_\Gamma\,\mathsf{env}_\Gamma) \end{array}$$

The identity environment env_Γ is defined by induction on Γ as follows:

$$\begin{array}{lcl} \mathsf{env}_\Gamma & : & [\![\Gamma]\!]_\Gamma \\ \mathsf{env}_\emptyset & = & () \\ \mathsf{env}_{\Gamma.A} & = & ([\![\Gamma]\!]_{\mathsf{weak}_A\,\mathsf{id}_\Gamma}(\mathsf{env}_\Gamma),\ \mathsf{reflect}^A_{\Gamma.A}\,(\mathsf{hyp}\ \mathsf{top})) \end{array}$$

4.4 Digression: The CCC of reifiable presheaves

Instead of defining the semantics $[\![A]\!]$ of formula A by induction on A, we can consider a category whose objects are interpretations of formulas, and show that this category is Cartesian closed.

An object in this category would be a quadruple consisting of:

$$\begin{array}{lcll} \mathcal{A} & : & \mathsf{PSh} & \text{a presheaf} \\ A & : & \mathsf{Form} & \text{a formula} \\ \mathsf{reflect}^A & : & \mathsf{Ne}(A) \dot{\to} \mathcal{A} & \text{a method to reflect neutrals into } \mathcal{A} \\ \mathsf{reify}^A & : & \mathcal{A} \dot{\to} \mathsf{Nf}(A) & \text{a method to reify values of } \mathcal{A} \end{array}$$

Exercise 11 *Complete this definition by a suitable notion of morphism.*

Exercise 12 *Show that this category is Cartesian closed.*

5 Disjunction and absurdity

For sets $S_1, S_2 : \mathsf{SET}$ let $S_1 + S_2$ denote the disjoint union with injections $\mathsf{inj}_i : S_i \to S_1 + S_2$ and case distinction distinction $[f_1, f_2] : S_1 + S_2 \to T$ for $f_i : S_i \to T$ such that $[f_1, f_2] \circ \mathsf{inj}_i = f_i$.

Presheaf coproduct is given pointwise by $(\mathcal{A} \hat{+} \mathcal{B})_\Gamma = \mathcal{A}_\Gamma + \mathcal{B}_\Gamma$.

Exercise 13 *Show that* $\hat{+}$ *is indeed a coproduct in* PSh.

5.1 Semantics, reflection and reification

The first idea how to model disjunction, $[\![A \vee B]\!]_\Gamma = [\![A]\!]_\Gamma + [\![B]\!]_\Gamma$, fails because we cannot reflect disjunctive hypotheses. E. g., an element of $[\![A \vee B]\!]_{A \vee B}$ should be of the form $\mathsf{inj}_1\, a$ or $\mathsf{inj}_2\, b$, but the decision on first or second injection cannot be made here; it depends on the hypothesis $A \vee B$. The solution is to extend the semantics to allow case distinction on variables (and neutrals) before making the decisions needed for disjunctions.

Given a presheaf $\mathcal{A}$, we define a new presheaf $\mathsf{Cover}\,(\mathcal{A})$ inductively by the following constructors:

$$\frac{a : \mathcal{A}_\Gamma}{\mathsf{returnC}\ a : \mathsf{Cover}\,(\mathcal{A})_\Gamma} \qquad \frac{t : \mathsf{Ne}(\bot)_\Gamma}{\mathsf{falseC}\ t : \mathsf{Cover}\,(\mathcal{A})_\Gamma}$$

$$\frac{t : \mathsf{Ne}(C \vee D)_\Gamma \qquad c : \mathsf{Cover}\,(\mathcal{A})_{\Gamma.C} \qquad d : \mathsf{Cover}\,(\mathcal{A})_{\Gamma.D}}{\mathsf{orC}\ t\ c\ d : \mathsf{Cover}\,(\mathcal{A})_\Gamma}$$

Exercise 14 *Show that* $\mathsf{Cover}\,(\mathcal{A})$ *is indeed a presheaf.*

We can think of $c : \mathsf{Cover}\,(\mathcal{A})_\Gamma$ as a decision tree with leaves in $\mathcal{A}$ and nodes labeled by terms we case on. Nodes of type falseC have no subtrees since *ex falsum quod libet*, and orC nodes casing on $t : \mathsf{Ne}(C \vee D)_\Gamma$ have two subtrees, one having a new hypothesis C, and one having a new hypothesis D.

We can replace all the leaves according to a function $f : \mathcal{A} \dot{\to} \mathcal{B}$, this makes Cover a *functor* in the category of presheaves. We can also replace all the leaves by new case trees via a natural transformation

$$\mathsf{joinC} \ : \ \mathsf{Cover}\,(\mathsf{Cover}\,(\mathcal{A})) \dot{\to} \mathsf{Cover}\,(\mathcal{A})$$

which makes Cover a *monad* in PSh.

Exercise 15 *Define the functorial action* $\mathsf{Cover}\,(f)$ *and* joinC *and prove the monad laws.*

Further, if the leaves of a case tree are normal forms, we can turn it into one big normal form via:

$$\mathsf{pasteNf} \;:\; \mathsf{Cover}\,(\mathsf{Nf}(A)) \,\dot{\to}\, \mathsf{Nf}(A)$$

Exercise 16 *Define* $\mathsf{pasteNf}$.

We have everything now to model disjunction:

$$\begin{array}{lcl}
[\![A \vee B]\!] & = & \mathsf{Cover}\,([\![A]\!] \,\hat{+}\, [\![B]\!]) \\[1ex]
\mathsf{reflect}^{A\vee B} & : & \mathsf{Ne}(A \vee B) \,\dot{\to}\, [\![A \vee B]\!] \\
\mathsf{reflect}^{A\vee B}_{\Gamma}\; t & = & \mathsf{orC}\; t\; (\mathsf{returnC}\; (\mathsf{inj}_1(\mathsf{reflect}^{A}_{\Gamma.A}\; (\mathsf{hyp\ top})))) \\
 & & \qquad (\mathsf{returnC}\; (\mathsf{inj}_2(\mathsf{reflect}^{B}_{\Gamma.B}\; (\mathsf{hyp\ top})))) \\[1ex]
\mathsf{reifyOr}^{A\vee B} & : & [\![A]\!] \,\hat{+}\, [\![B]\!] \,\dot{\to}\, \mathsf{Nf}(A \vee B) \\
\mathsf{reifyOr}^{A\vee B}_{\Gamma}\; (\mathsf{inj}_1\, a) & = & \mathsf{orI}_1\; (\mathsf{reify}^{A}_{\Gamma}\; a) \\
\mathsf{reifyOr}^{A\vee B}_{\Gamma}\; (\mathsf{inj}_2\, b) & = & \mathsf{orI}_2\; (\mathsf{reify}^{B}_{\Gamma}\; b) \\[1ex]
\mathsf{reify}^{A\vee B} & : & [\![A \vee B]\!] \,\dot{\to}\, \mathsf{Nf}(A \vee B) \\
\mathsf{reify}^{A\vee B} & = & \mathsf{pasteNf} \circ \mathsf{Cover}\,(\mathsf{reifyOr}^{A\vee B})
\end{array}$$

Reification works by first reifying the leaves of the case tree using $\mathsf{reifyOr}$, and then turning the whole tree into a normal form using $\mathsf{pasteNf}$.

Exercise 17 *Compute* $\mathsf{reify}^{P\vee Q}_{P\vee Q}\,(\mathsf{reflect}^{P\vee Q}_{P\vee Q}\,(\mathsf{hyp\ top}))$.

For absurdity, we use the presheaf $\hat{0}_\Gamma = \{\}$ with initial morphism $\bot\mathsf{-elim} : \hat{0} \,\dot{\to}\, \mathcal{C}$.

$$\begin{array}{lcl}
[\![\bot]\!] & = & \mathsf{Cover}\,\hat{0} \\[1ex]
\mathsf{reflect}^{\bot} & : & \mathsf{Ne}(\bot) \,\dot{\to}\, [\![\bot]\!] \\
\mathsf{reflect}^{\bot}_{\Gamma}\; t & = & \mathsf{falseC}\; t \\[1ex]
\mathsf{reify}^{\bot} & : & [\![\bot]\!] \,\dot{\to}\, \mathsf{Nf}(\bot) \\
\mathsf{reify}^{\bot} & = & \mathsf{pasteNf} \circ \mathsf{Cover}\,(\bot\mathsf{-elim})
\end{array}$$

To reify a value $c : [\![\bot]\!]$, observe that because $\hat{0}$ is empty, c can only be a case tree without leaves, i. e., all branches end in falseC nodes. Thus, the mapping $\mathsf{Cover}\,(\bot\mathsf{-elim}) : \mathsf{Cover}\,(\hat{0}) \,\dot{\to}\, \mathsf{Cover}\,(\mathsf{Nf}(\bot))$ is merely a type cast. The subsequent $\mathsf{pasteNf}$ will turn the leafless tree c into a normal form consisting entirely of splits orE and falseE.

Exercise 18 *Prove that* $\mathsf{reify}^A \circ \mathsf{reflect}^A$ *produces the long η-normal form if applied to variables (i. e., hypotheses).*

5.2 Evaluation

Evaluation poses one more challenge. Consider the term:

$$\frac{x : \mathsf{Hyp}(C \vee D)_\Gamma \qquad f : \mathsf{Tm}(A \Rightarrow B)_{\Gamma.C} \qquad g : \mathsf{Tm}(B)_{\Gamma.D.A} \qquad a : \mathsf{Tm}(A)_\Gamma}{\mathsf{impE}\ (\mathsf{orE}\ (\mathsf{hyp}\ x)\ f\ (\mathsf{impI}\ g))\ a : \mathsf{Tm}(B)_\Gamma}$$

There is a redex $\mathsf{impE}\ (\mathsf{impI}\ g)\ a$ separated by an orE. The function term $\mathsf{orE}\ (\mathsf{hyp}\ x)\ f\ (\mathsf{impI}\ g)$ would naturally evaluate to a case tree with functions in the leaves. To apply it to the evaluation of a, we have to apply each leaf to that argument. We facilitate this by implementing

$$\mathsf{paste}^{A \Rightarrow B} : \mathsf{Cover}\,(\llbracket A \Rightarrow B \rrbracket) \dot{\to} \llbracket A \Rightarrow B \rrbracket$$

that turns a case tree of functions into a function which expects an argument that will then be passed to all the functions in the case tree. In fact, we need pasting at all formulas A:

$$\begin{array}{lcl}
\mathsf{paste}^A & : & \mathsf{Cover}\,\llbracket A \rrbracket \dot{\to} \llbracket A \rrbracket \\
\mathsf{paste}^P & = & \mathsf{pasteNf} \\
\mathsf{paste}^\perp & = & \mathsf{joinC} \\
\mathsf{paste}^{A \vee B} & = & \mathsf{joinC} \\
\mathsf{paste}^{A \wedge B} & = & \langle\, \mathsf{paste}^A \circ \mathsf{Cover}\,(\pi_1)\,,\ \mathsf{paste}^B \circ \mathsf{Cover}\,(\pi_2)\,\rangle \\
\end{array}$$
$$\mathsf{paste}^{A \Rightarrow B}_\Gamma\ c\ {}_\Delta\ (\tau : \Delta {\le} \Gamma)\ (a : \llbracket A \rrbracket_\Delta)\ =\ (\mathsf{paste}^B_\Delta\ \circ \mathsf{mapC}_\Delta(\varphi) \circ \mathsf{Cover}\,(\llbracket A \Rightarrow B \rrbracket_\tau))\,c$$

In the case of implication $A \Rightarrow B$, we need a stronger version of functoriality of Cover which only requires a

$$\varphi : \prod_\Phi \prod_{\delta : \Phi \le \Delta} \mathcal{A}_\Phi \to \mathcal{B}_\Phi$$

to produce a $\mathsf{mapC}_\Delta(\varphi) : \mathsf{Cover}\,(\mathcal{A})_\Delta \to \mathsf{Cover}\,(\mathcal{B})_\Delta$. In our case, $\mathcal{A} = \llbracket A \Rightarrow B \rrbracket$ and $\mathcal{B} = \llbracket B \rrbracket$ and

$$\varphi\ {}_\Phi\ \delta\ f = f\ \mathsf{id}_\Phi\ (\llbracket A \rrbracket_\delta\, a).$$

Pasting allows us to complete the definition of evaluation.

$$\begin{array}{rcl}
\mathsf{eval}^A_\Gamma & : & \mathsf{Tm}(A)_\Gamma \to \llbracket \Gamma \rrbracket \dot{\to} \llbracket A \rrbracket \\
\mathsf{eval}^C_\Gamma(\mathsf{orE}\ t\ u\ v)_\Delta\ \gamma & = & \mathsf{paste}^C_\Delta\ (\mathsf{mapC}_\Delta(\lambda\,{}_\Phi\,(\tau : \Phi \le \Delta) \to [f, g])\ c) \\
\text{where } c & = & \mathsf{eval}^{A \vee B}_\Gamma(t)\ {}_\Delta\ \gamma \\
f\ a & = & \mathsf{eval}^C_{\Gamma.A}(u)\ {}_\Phi\ (\llbracket \Gamma \rrbracket_\tau\, \gamma, a) \\
g\ b & = & \mathsf{eval}^C_{\Gamma.B}(v)\ {}_\Phi\ (\llbracket \Gamma \rrbracket_\tau\, \gamma, b)
\end{array}$$

Exercise 19 *Really complete the definition of* eval *(all missing cases).*

That's it!

6 Literature

Further reading on normalization by evaluation in general:

1. U. Berger and H. Schwichtenberg. An inverse to the evaluation functional for typed λ-calculus. In *LICS'91*, 1991

2. A. Jung and J. Tiuryn. A new characterization of lambda definability. In *TLCA'93*, volume 664 of *LNCS*, 1993

3. T. Altenkirch, M. Hofmann, and T. Streicher. Categorical reconstruction of a reduction free normalization proof. In *CTCS'95*, volume 953 of *LNCS*, 1995

4. A. Abel. *Normalization by Evaluation: Dependent Types and Impredicativity.* Habilitation thesis, Ludwig-Maximilians-University Munich, 2013

Literature dealing in particular with the case of disjunction / disjoint sums:

1. O. Danvy. Type-directed partial evaluation. In *POPL'96*, 1996

2. M. P. Fiore and A. K. Simpson. Lambda definability with sums via Grothendieck logical relations. In *TLCA'99*, volume 1581 of *LNCS*, 1999

3. T. Altenkirch, P. Dybjer, M. Hofmann, and P. J. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In *LICS'01*, 2001

4. V. Balat, R. D. Cosmo, and M. P. Fiore. Extensional normalisation and type-directed partial evaluation for typed lambda calculus with sums. In *POPL'04*, 2004

5. T. Altenkirch and T. Uustalu. Normalization by evaluation for $\lambda^{\to 2}$. In *FLOPS'04*, volume 2998 of *LNCS*, 2004

6. F. Barral. *Decidability for non-standard conversions in lambda-calculus.* PhD thesis, Ludwig-Maximilians-University Munich, 2008

7. G. Scherer. Deciding equivalence with sums and the empty type. In *POPL'17*, 2017

7 Conclusion

Acknowledgments. The presentation of this NbE proof and its Agda formalization benefited greatly from discussions with Christian Sattler. Thanks to Jesper Cockx for listening to a prerun of the talk.