A conciliatory short proof of strong normalization for permutative conversions in natural deduction

Favio E. Miranda-Perea
Facultad de Ciencias, Departamento de Matemáticas, UNAM.
favio@ciencias.unam.mx

Eduardo Ugalde-Reyes
Posgrado en Filosofía de la Ciencia, UNAM.
edugare@gmail.com

This Research Project has been funded by a grant DGAPA-UNAM PAPIIT 101723 and also from a grant from CONAHCYT.

XVI LANMR, Oaxaca, México **07/09/2024**

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.
- While this result is a well-known aspect of the Curry-Howard correspondence, our approach revisits and reconstructs the proof techniques presented by Joachimski & Matthes (2003).

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.
- While this result is a well-known aspect of the Curry-Howard correspondence, our approach revisits and reconstructs the proof techniques presented by Joachimski & Matthes (2003).
- Building on this foundation, we will focus on incorporating permutation cases in Nm and introducing the π-rule to handle the conversion cases.

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.
- While this result is a well-known aspect of the Curry-Howard correspondence, our approach revisits and reconstructs the proof techniques presented by Joachimski & Matthes (2003).
- Building on this foundation, we will focus on incorporating permutation cases in Nm and introducing the π-rule to handle the conversion cases.
- Through this reconstruction, we have two primary goals:

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.
- While this result is a well-known aspect of the Curry-Howard correspondence, our approach revisits and reconstructs the proof techniques presented by Joachimski & Matthes (2003).
- Building on this foundation, we will focus on incorporating permutation cases in Nm and introducing the π-rule to handle the conversion cases.
- Through this reconstruction, we have two primary goals:
  1. In our effort to achieve a "computer-checkable formalization and proof-theoretic analysis of normalization proofs" (Joachimski & Matthes, 2003, p. 59), we address specific gaps that were overlooked in the original proof by Joachimski & Matthes (2003)

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.
- While this result is a well-known aspect of the Curry-Howard correspondence, our approach revisits and reconstructs the proof techniques presented by Joachimski & Matthes (2003).
- Building on this foundation, we will focus on incorporating permutation cases in Nm and introducing the π-rule to handle the conversion cases.
- Through this reconstruction, we have two primary goals:
    1. In our effort to achieve a "computer-checkable formalization and proof-theoretic analysis of normalization proofs" (Joachimski & Matthes, 2003, p. 59), we address specific gaps that were overlooked in the original proof by Joachimski & Matthes (2003)
    2. To refine our understanding of the Curry-Howard correspondence by integrating insights from the practice of normalizing proofs in natural deduction.

Outline

- In this talk, which extends our previous research presented for our last meeting in the LANMR 2023, we will show how to handle permutative conversions in natural deduction through an embedding into a typed lambda calculus with sums for strong normalization results.
- While this result is a well-known aspect of the Curry-Howard correspondence, our approach revisits and reconstructs the proof techniques presented by Joachimski & Matthes (2003).
- Building on this foundation, we will focus on incorporating permutation cases in Nm and introducing the π-rule to handle the conversion cases.
- Through this reconstruction, we have two primary goals:
  1. In our effort to achieve a "computer-checkable formalization and proof-theoretic analysis of normalization proofs" (Joachimski & Matthes, 2003, p. 59), we address specific gaps that were overlooked in the original proof by Joachimski & Matthes (2003)
  2. To refine our understanding of the Curry-Howard correspondence by integrating insights from the practice of normalizing proofs in natural deduction.
- Both goals reflect two contrasting perspectives on mathematical practice, described as the romantic attitude versus the cool attitude (Barendregt & Wiedijk, 2005).

Outline

- The romantic attitude concerns the traditional approach of practicing mathematics where the acts of defining, proving, and calculating are kept on a "humane scale". **Romantic proofs are informal yet rigorous.**

Outline

- The romantic attitude concerns the traditional approach of practicing mathematics where the acts of defining, proving, and calculating are kept on a "humane scale". **Romantic proofs are informal yet rigorous.**
- The cool attitude, in contrast, advocates for active support from proof assistants in those same activities, recognizing their need in daily practice to handle the growing complexity of mathematical results. **Cool proofs are mechanizable proofs.**

Outline

- The romantic attitude concerns the traditional approach of practicing mathematics where the acts of defining, proving, and calculating are kept on a "humane scale". **Romantic proofs are informal yet rigorous.**
- The cool attitude, in contrast, advocates for active support from proof assistants in those same activities, recognizing their need in daily practice to handle the growing complexity of mathematical results. **Cool proofs are mechanizable proofs.**
- All in all, we aim to provide a proof that reconciles the "algebraic reasoning" embodied by the lambda calculus in normalization results intended to be verified by a computer with the "diagrammatic reasoning" of natural deduction, where the analysis of normal proofs is conducted by manipulating proof-trees.

Outline

- The romantic attitude concerns the traditional approach of practicing mathematics where the acts of defining, proving, and calculating are kept on a "humane scale". **Romantic proofs are informal yet rigorous.**
- The cool attitude, in contrast, advocates for active support from proof assistants in those same activities, recognizing their need in daily practice to handle the growing complexity of mathematical results. **Cool proofs are mechanizable proofs.**
- All in all, we aim to provide a proof that reconciles the "algebraic reasoning" embodied by the lambda calculus in normalization results intended to be verified by a computer with the "diagrammatic reasoning" of natural deduction, where the analysis of normal proofs is conducted by manipulating proof-trees.
- We achieve this by giving a "diagrammatic" interpretation to the "algebraic" reasoning provided by Joachimski & Matthes (2003).

Curry-Howard at work: ND-$\lambda^{\supset,\vee}$

- The most straightforward way to grasp Curry-Howard at work is by introducing the well-known system *ND*-$\lambda^{\supset,\vee}$.

Curry-Howard at work: ND-$\lambda^{\supset,\vee}$

- The most straightforward way to grasp Curry-Howard at work is by introducing the well-known system $\mathit{ND}\text{-}\lambda^{\supset,\vee}$.
- This comprises minimal logic with implication and disjunction, but into which we have embedded proof-objects from the simply-typed lambda calculus augmented with **injections and case analysis**: $\mathit{ND}\text{-}\lambda^{\supset,\vee} = \mathit{Nm} + \lambda^{\supset,\vee} - \mathit{terms}$.

Curry-Howard at work: ND-$\lambda^{\supset,\vee}$

- The most straightforward way to grasp Curry-Howard at work is by introducing the well-known system *ND-*$\lambda^{\supset,\vee}$.
- This comprises minimal logic with implication and disjunction, but into which we have embedded proof-objects from the simply-typed lambda calculus augmented with **injections and case analysis**: *ND-*$\lambda^{\supset,\vee} = Nm + \lambda^{\supset,\vee} - terms$.

$$x : A$$

$$\dfrac{\begin{array}{c}\Gamma, [x : A] \\ \vdots \\ r : B\end{array}}{\lambda x.r : A \supset B} \supset I, x \qquad \dfrac{\begin{array}{c}\Gamma \\ \vdots \\ r : A \supset B\end{array} \qquad \begin{array}{c}\Delta \\ \vdots \\ s : A\end{array}}{rs : B} \supset E$$

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ r : A \vee B\end{array} \qquad \begin{array}{c}\Delta, [x : A] \\ \vdots \\ s : C\end{array} \qquad \begin{array}{c}\Theta, [y : B] \\ \vdots \\ t : C\end{array}}{\mathsf{case}(r, x.s, y.t) : C} \vee E, x, y$$

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ r : A\end{array}}{\mathsf{inl}\,r : A \vee B} \vee I_1 \qquad \dfrac{\begin{array}{c}\Gamma \\ \vdots \\ s : B\end{array}}{\mathsf{inr}\,s : A \vee B} \vee I_2$$

Curry-Howard at work: ND-$\lambda^{\supset,\vee}$

- The most straightforward way to grasp Curry-Howard at work is by introducing the well-known system *ND-*$\lambda^{\supset,\vee}$.
- This comprises minimal logic with implication and disjunction, but into which we have embedded proof-objects from the simply-typed lambda calculus augmented with **injections and case analysis**: *ND-*$\lambda^{\supset,\vee} = Nm + \lambda^{\supset,\vee} - terms$.

$$x : A$$

$$\dfrac{\begin{matrix}\Gamma, [x : A] \\ \vdots \\ r : B\end{matrix}}{\lambda x.r : A \supset B} \supset I, x \qquad \dfrac{\begin{matrix}\Gamma \\ \vdots \\ r : A \supset B\end{matrix} \qquad \begin{matrix}\Delta \\ \vdots \\ s : A\end{matrix}}{rs : B} \supset E$$

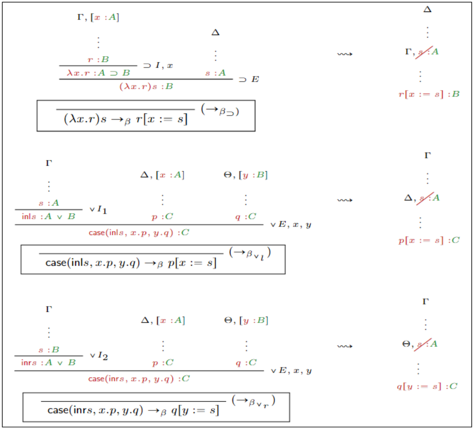
$$\dfrac{\begin{matrix}\Gamma \\ \vdots \\ r : A \vee B\end{matrix} \qquad \begin{matrix}\Delta, [x : A] \\ \vdots \\ s : C\end{matrix} \qquad \begin{matrix}\Theta, [y : B] \\ \vdots \\ t : C\end{matrix}}{\mathrm{case}(r, x.s, y.t) : C} \vee E, x, y$$

$$\dfrac{\begin{matrix}\Gamma \\ \vdots \\ r : A\end{matrix}}{\mathrm{inl}\, r : A \vee B} \vee I_1 \qquad \dfrac{\begin{matrix}\Gamma \\ \vdots \\ s : B\end{matrix}}{\mathrm{inr}\, s : A \vee B} \vee I_2$$

$$\boxed{\text{r,s,t} ::= \text{x} \mid \lambda x.r \mid \mathrm{inl}\, r \mid \mathrm{inr}\, s \mid rs \mid \mathrm{case}(r, x.s, y.t)}$$

Normalization in *ND*

- Normalization in *ND* involves simplifying proofs so that they avoid making détours.



$$\dfrac{\dfrac{\begin{matrix}\Gamma, [x:A]\\ \vdots\\ r:B\end{matrix}}{\lambda x.r : A \supset B}\supset I, x \qquad \begin{matrix}\Delta\\ \vdots \\ s:A\end{matrix}}{(\lambda x.r)s : B}\supset E \qquad \rightsquigarrow \qquad \begin{matrix}\Delta\\ \vdots\\ \Gamma, \cancel{s}:A\\ \vdots\\ r[x:=s]:B\end{matrix}$$

$$\boxed{\dfrac{}{(\lambda x.r)s \to_\beta r[x:=s]}\,(\to_{\beta_\supset})}$$

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ s:A\end{matrix}}{\mathsf{inl}s : A \vee B}\vee I_1 \qquad \begin{matrix}\Delta, [x:A]\\ \vdots\\ p:C\end{matrix} \qquad \begin{matrix}\Theta, [y:B]\\ \vdots\\ q:C\end{matrix}}{\mathsf{case}(\mathsf{inl}s, x.p, y.q) : C}\vee E, x, y \qquad \rightsquigarrow \qquad \begin{matrix}\Gamma\\ \vdots\\ \Delta, \cancel{s}:A\\ \vdots\\ p[x:=s]:C\end{matrix}$$

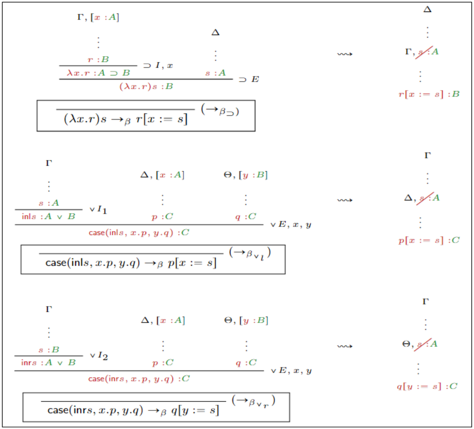
$$\boxed{\dfrac{}{\mathsf{case}(\mathsf{inl}s, x.p, y.q) \to_\beta p[x:=s]}\,(\to_{\beta_{\vee_l}})}$$

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ s:B\end{matrix}}{\mathsf{inr}s : A \vee B}\vee I_2 \qquad \begin{matrix}\Delta, [x:A]\\ \vdots\\ p:C\end{matrix} \qquad \begin{matrix}\Theta, [y:B]\\ \vdots\\ q:C\end{matrix}}{\mathsf{case}(\mathsf{inr}s, x.p, y.q) : C}\vee E, x, y \qquad \rightsquigarrow \qquad \begin{matrix}\Gamma\\ \vdots\\ \Theta, \cancel{s}:A\\ \vdots\\ q[y:=s]:C\end{matrix}$$

$$\boxed{\dfrac{}{\mathsf{case}(\mathsf{inr}s, x.p, y.q) \to_\beta q[y:=s]}\,(\to_{\beta_{\vee_r}})}$$

Normalization in *ND*

- Normalization in *ND* involves simplifying proofs so that they avoid making détours.
- We recall that a **détour** in a derivation consists of applying an *I − rule* followed by the *E − rule* of the same operator in succession.



$$\dfrac{\dfrac{\begin{matrix}\Gamma, [x:A]\\ \vdots\\ r:B\end{matrix}}{\lambda x.r : A \supset B}\supset I, x \qquad \begin{matrix}\Delta\\ \vdots\\ s:A\end{matrix}}{(\lambda x.r)s : B}\supset E \quad \rightsquigarrow \quad \begin{matrix}\Delta\\ \vdots\\ \Gamma, \cancel{s}:A\\ \vdots\\ r[x:=s]:B\end{matrix}$$

$$\boxed{\dfrac{}{(\lambda x.r)s \to_\beta r[x:=s]}(\to_{\beta_\supset})}$$

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ s:A\end{matrix}}{\mathsf{inl}s : A \vee B}\vee I_1 \qquad \begin{matrix}\Delta, [x:A]\\ \vdots\\ p:C\end{matrix} \qquad \begin{matrix}\Theta, [y:B]\\ \vdots\\ q:C\end{matrix}}{\mathsf{case}(\mathsf{inl}s, x.p, y.q) : C}\vee E, x, y \quad \rightsquigarrow \quad \begin{matrix}\Gamma\\ \vdots\\ \Delta, \cancel{s}:A\\ \vdots\\ p[x:=s]:C\end{matrix}$$

$$\boxed{\dfrac{}{\mathsf{case}(\mathsf{inl}s, x.p, y.q) \to_\beta p[x:=s]}(\to_{\beta_{\vee_l}})}$$

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ s:B\end{matrix}}{\mathsf{inr}s : A \vee B}\vee I_2 \qquad \begin{matrix}\Delta, [x:A]\\ \vdots\\ p:C\end{matrix} \qquad \begin{matrix}\Theta, [y:B]\\ \vdots\\ q:C\end{matrix}}{\mathsf{case}(\mathsf{inr}s, x.p, y.q) : C}\vee E, x, y \quad \rightsquigarrow \quad \begin{matrix}\Gamma\\ \vdots\\ \Theta, \cancel{s}:A\\ \vdots\\ q[y:=s]:C\end{matrix}$$

$$\boxed{\dfrac{}{\mathsf{case}(\mathsf{inr}s, x.p, y.q) \to_\beta q[y:=s]}(\to_{\beta_{\vee_r}})}$$

Normalization in *ND*

- Unlike **modus ponens**, the conclusion of **case analysis** introduces an arbitrary formula that is neither part of the major premise nor part of the hypotheses. Consequently, the sub-formula property for derivations is no longer guaranteed.

Normalization in *ND*

- Unlike **modus ponens**, the conclusion of **case analysis** introduces an arbitrary formula that is neither part of the major premise nor part of the hypotheses. Consequently, the sub-formula property for derivations is no longer guaranteed.
- To address this issue, permutation convertibilities are added to *ND*, which help uncover hidden détours that could be separated by the application of the case analysis rule.

Normalization in *ND*

- Unlike **modus ponens**, the conclusion of **case analysis** introduces an arbitrary formula that is neither part of the major premise nor part of the hypotheses. Consequently, the sub-formula property for derivations is no longer guaranteed.
- To address this issue, permutation convertibilities are added to *ND*, which help uncover hidden détours that could be separated by the application of the case analysis rule.

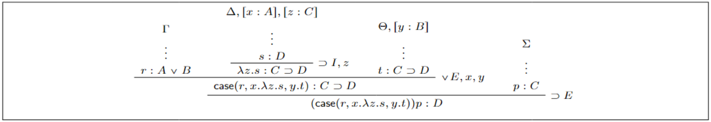
Normalization in *ND*

- Unlike **modus ponens**, the conclusion of **case analysis** introduces an arbitrary formula that is neither part of the major premise nor part of the hypotheses. Consequently, the sub-formula property for derivations is no longer guaranteed.
- To address this issue, permutation convertibilities are added to *ND*, which help uncover hidden détours that could be separated by the application of the case analysis rule.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ A\vee B\end{array}\quad \begin{array}{c}[A]\\ \vdots\\ C\end{array}\quad \begin{array}{c}[B]\\ \vdots\\ C\end{array}}{C}\,\vee\mathcal{E}\qquad \vdots}{D}\,\mathsf{r} \quad \text{converts to} \quad \dfrac{\begin{array}{c}\vdots\\ A\vee B\end{array}\quad \dfrac{\begin{array}{c}[A]\\ \vdots\\ C\end{array}\quad \vdots}{D}\,\mathsf{r}\quad \dfrac{\begin{array}{c}[B]\\ \vdots\\ C\end{array}\quad \vdots}{D}\,\mathsf{r}}{D}\,\vee\mathcal{E}$$

Normalization in *ND*: The Permutations

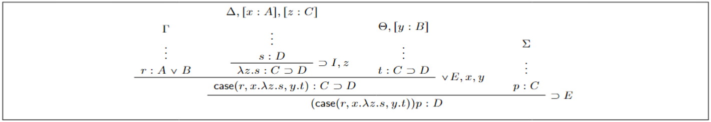
- Adding disjunction to *Nm* creates a new kind of cut, known as the permutation.



$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix} \qquad \dfrac{\begin{matrix}\Delta, [x:A], [z:C]\\ \vdots \\ s : D\end{matrix}}{\lambda z.s : C \supset D}\supset I, z \qquad \begin{matrix}\Theta,[y:B]\\ \vdots\\ t : C\supset D\end{matrix}}{\mathsf{case}(r, x.\lambda z.s, y.t) : C \supset D}\vee E, x, y \qquad \begin{matrix}\Sigma\\ \vdots\\ p : C\end{matrix}}{(\mathsf{case}(r, x.\lambda z.s, y.t))p : D}\supset E$$

Normalization in *ND*: The Permutations

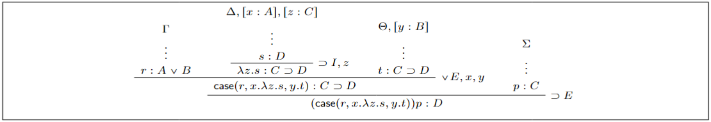
- Adding disjunction to *Nm* creates a new kind of cut, known as the permutation.
- Permutative convertions are needed to recover the sub-formula property of *Nm* when dealing with case analysis as in the following proof-situation:



$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A\vee B\end{matrix} \quad \dfrac{\begin{matrix}\Delta,[x:A],[z:C]\\ \vdots\\ s:D\end{matrix}}{\lambda z.s : C\supset D}\supset I, z \quad \begin{matrix}\Theta,[y:B]\\ \vdots\\ t: C\supset D\end{matrix}}{\mathsf{case}(r, x.\lambda z.s, y.t) : C\supset D}\vee E, x, y \quad \begin{matrix}\Sigma\\ \vdots\\ p:C\end{matrix}}{(\mathsf{case}(r,x.\lambda z.s,y.t))p : D}\supset E$$

Normalization in *ND*: The Permutations

- Adding disjunction to *Nm* creates a new kind of cut, known as the permutation.
- Permutative convertions are needed to recover the sub-formula property of *Nm* when dealing with case analysis as in the following proof-situation:

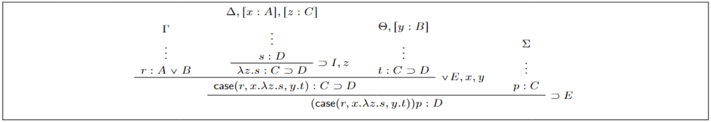


$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r:A\vee B\end{matrix} \quad \dfrac{\begin{matrix}\Delta,[x:A],[z:C]\\ \vdots\\ s:D\end{matrix}}{\lambda z.s:C\supset D}\supset I,z \quad \begin{matrix}\Theta,[y:B]\\ \vdots\\ t:C\supset D\end{matrix}}{\mathsf{case}(r,x.\lambda z.s,y.t):C\supset D}\vee E,x,y \quad \begin{matrix}\Sigma\\ \vdots\\ p:C\end{matrix}}{(\mathsf{case}(r,x.\lambda z.s,y.t))p:D}\supset E$$

- Which we would like to transform into:

Normalization in *ND*: The Permutations

- Adding disjunction to *Nm* creates a new kind of cut, known as the permutation.
- Permutative convertions are needed to recover the sub-formula property of *Nm* when dealing with case analysis as in the following proof-situation:



$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix} \qquad \dfrac{\begin{matrix}\Delta, [x : A], [z : C]\\ \vdots\\ s : D\end{matrix}}{\lambda z.s : C \supset D}\supset I, z \qquad \begin{matrix}\Theta, [y : B]\\ \vdots\\ t : C \supset D\end{matrix}}{\mathsf{case}(r, x.\lambda z.s, y.t) : C \supset D}\vee E, x, y \qquad \begin{matrix}\Sigma\\ \vdots\\ p : C\end{matrix}}{(\mathsf{case}(r, x.\lambda z.s, y.t))p : D}\supset E$$

- Which we would like to transform into:

Normalization in *ND*: The Permutations

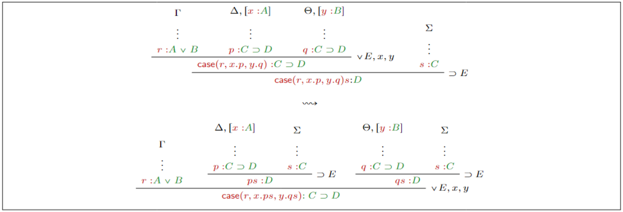
- Adding disjunction to *Nm* creates a new kind of cut, known as the permutation.
- Permutative convertions are needed to recover the sub-formula property of *Nm* when dealing with case analysis as in the following proof-situation:

$$\dfrac{\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A \vee B\end{array} \quad \dfrac{\begin{array}{c}\Delta, [x : A], [z : C]\\ \vdots\\ s : D\end{array}}{\lambda z.s : C \supset D}\supset I, z \quad \begin{array}{c}\Theta, [y : B]\\ \vdots\\ t : C \supset D\end{array}}{\mathsf{case}(r, x.\lambda z.s, y.t) : C \supset D}\vee E, x, y \quad \begin{array}{c}\Sigma\\ \vdots\\ p : C\end{array}}{(\mathsf{case}(r, x.\lambda z.s, y.t))p : D}\supset E$$

- Which we would like to transform into:

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A \vee B\end{array} \quad \dfrac{\dfrac{\begin{array}{c}\Delta, [x : A], [z : C]\\ \vdots\\ s : D\end{array}}{\lambda w.s : C \supset D}\supset I, z \quad \begin{array}{c}\Sigma\\ \vdots\\ p : C\end{array}}{(\lambda z.s)p : D}\supset E \quad \dfrac{\begin{array}{c}\Theta, [y : B]\\ \vdots\\ t : C \supset D\end{array} \quad \begin{array}{c}\Sigma\\ \vdots\\ p : C\end{array}}{tp : D}\supset E}{\mathsf{case}(r, x.(\lambda z.s)p, y.tp) : D}\vee E, x, y$$

Normalization in *ND*:π – *reductions* & β – *reductions*



$$\dfrac{\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r:A\vee B\end{array}\qquad \begin{array}{c}\Delta,[x:A]\\ \vdots\\ p:C\supset D\end{array}\qquad \begin{array}{c}\Theta,[y:B]\\ \vdots\\ q:C\supset D\end{array}}{\mathsf{case}(r,x.p,y.q):C\supset D}\vee E,x,y \qquad \begin{array}{c}\Sigma\\ \vdots\\ s:C\end{array}}{\mathsf{case}(r,x.p,y.q)s:D}\supset E$$

$$\rightsquigarrow$$

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r:A\vee B\end{array}\qquad \dfrac{\begin{array}{c}\Delta,[x:A]\\ \vdots\\ p:C\supset D\end{array}\quad \begin{array}{c}\Sigma\\ \vdots\\ s:C\end{array}}{ps:D}\supset E \qquad \dfrac{\begin{array}{c}\Theta,[y:B]\\ \vdots\\ q:C\supset D\end{array}\quad \begin{array}{c}\Sigma\\ \vdots\\ s:C\end{array}}{qs:D}\supset E}{\mathsf{case}(r,x.ps,y.qs):C\supset D}\vee E,x,y$$

Normalization in *ND*:π *− reductions* & β *− reductions*

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix}\qquad \begin{matrix}\Delta, [x : A]\\ \vdots\\ p : C \supset D\end{matrix}\qquad \begin{matrix}\Theta, [y : B]\\ \vdots\\ q : C \supset D\end{matrix}}{\mathsf{case}(r, x.p, y.q) : C \supset D}\vee E, x, y \qquad \begin{matrix}\Sigma\\ \vdots\\ s : C\end{matrix}}{\mathsf{case}(r, x.p, y.q)s : D}\supset E$$

$$\rightsquigarrow$$

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix}\qquad \dfrac{\begin{matrix}\Delta, [x : A]\\ \vdots\\ p : C \supset D\end{matrix}\qquad \begin{matrix}\Sigma\\ \vdots\\ s : C\end{matrix}}{ps : D}\supset E \qquad \dfrac{\begin{matrix}\Theta, [y : B]\\ \vdots\\ q : C \supset D\end{matrix}\qquad \begin{matrix}\Sigma\\ \vdots\\ s : C\end{matrix}}{qs : D}\supset E}{\mathsf{case}(r, x.ps, y.qs) : C \supset D}\vee E, x, y$$

$$\boxed{\dfrac{}{\mathsf{case}(r, x.p, y.q)s \to_{\Pi} \mathsf{case}(r, x.ps, y.qs)}(\to_{\Pi})}$$

Normalization in *ND*:π − *reductions* & β − *reductions*

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r:A\vee B\end{matrix}\quad \begin{matrix}\Delta,[x:A]\\ \vdots\\ p:C\supset D\end{matrix}\quad \begin{matrix}\Theta,[y:B]\\ \vdots\\ q:C\supset D\end{matrix}}{\mathsf{case}(r,x.p,y.q):C\supset D}\vee E,x,y \quad \begin{matrix}\Sigma\\ \vdots\\ s:C\end{matrix}}{\mathsf{case}(r,x.p,y.q)s{:}D}\supset E$$

$$\leadsto$$

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r:A\vee B\end{matrix}\quad \dfrac{\begin{matrix}\Delta,[x:A]\\ \vdots\\ p:C\supset D\end{matrix}\quad \begin{matrix}\Sigma\\ \vdots\\ s:C\end{matrix}}{ps:D}\supset E \quad \dfrac{\begin{matrix}\Theta,[y:B]\\ \vdots\\ q:C\supset D\end{matrix}\quad \begin{matrix}\Sigma\\ \vdots\\ s:C\end{matrix}}{qs:D}\supset E}{\mathsf{case}(r,x.ps,y.qs):\ C\supset D}\vee E,x,y$$

$$\boxed{\dfrac{}{\mathsf{case}(r,x.p,y.q)s\to_{\Pi}\mathsf{case}(r,x.ps,y.qs)}\ (\to_{\Pi})}$$

$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r:A\vee B\end{matrix}\quad \begin{matrix}\Delta,[x:A]\\ \vdots\\ p:C\vee D\end{matrix}\quad \begin{matrix}\Theta,[y:B]\\ \vdots\\ q:C\vee D\end{matrix}}{\mathsf{case}(r,x.p,y.q):C\vee D}\vee E,x,y \quad \begin{matrix}\Sigma,[w:C]\\ \vdots\\ s:E\end{matrix}\quad \begin{matrix}\Phi,[z:D]\\ \vdots\\ t:E\end{matrix}}{\mathsf{case}(\mathsf{case}(r,x.p,y.q),w.s,z.t):E}\vee E,w,z$$

$$\leadsto$$

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r:A\vee B\end{matrix}\quad \dfrac{\begin{matrix}\Delta,[x:A]\\ \vdots\\ p:C\vee D\end{matrix}\quad \begin{matrix}\Sigma,[w:C]\\ \vdots\\ s:E\end{matrix}\quad \begin{matrix}\Phi,[z:D]\\ \vdots\\ t:E\end{matrix}}{\mathsf{case}(p,w.s,z.t):E}\vee E,w,z \quad \dfrac{\begin{matrix}\Theta,[y:B]\\ \vdots\\ q:C\vee D\end{matrix}\quad \begin{matrix}\Sigma,[w:C]\\ \vdots\\ s:E\end{matrix}\quad \begin{matrix}\Phi,[z:D]\\ \vdots\\ t:E\end{matrix}}{\mathsf{case}(q,w.s,z.t):E}\vee E,w,z}{\mathsf{case}(r,x.\mathsf{case}(p,w.s,z.t),y.\mathsf{case}(q,w.s,z.t)):E}\vee E,x,y$$

Normalization in *ND*:π – *reductions* & β – *reductions*

$$\dfrac{\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A \vee B\end{array}\qquad \begin{array}{c}\Delta, [x : A]\\ \vdots\\ p : C \supset D\end{array}\qquad \begin{array}{c}\Theta, [y : B]\\ \vdots\\ q : C \supset D\end{array}}{\mathsf{case}(r, x.p, y.q) : C \supset D}\vee E, x, y \qquad \begin{array}{c}\Sigma\\ \vdots\\ s : C\end{array}}{\mathsf{case}(r, x.p, y.q)s : D}\supset E$$

$$\rightsquigarrow$$

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A \vee B\end{array}\qquad \dfrac{\begin{array}{c}\Delta, [x : A]\\ \vdots\\ p : C \supset D\end{array}\quad \begin{array}{c}\Sigma\\ \vdots\\ s : C\end{array}}{ps : D}\supset E \qquad \dfrac{\begin{array}{c}\Theta, [y : B]\\ \vdots\\ q : C \supset D\end{array}\quad \begin{array}{c}\Sigma\\ \vdots\\ s : C\end{array}}{qs : D}\supset E}{\mathsf{case}(r, x.ps, y.qs) : C \supset D}\vee E, x, y$$

$$\boxed{\dfrac{}{\mathsf{case}(r, x.p, y.q)s \to_{\Pi} \mathsf{case}(r, x.ps, y.qs)}(\to_{\Pi})}$$

$$\dfrac{\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A \vee B\end{array}\qquad \begin{array}{c}\Delta, [x : A]\\ \vdots\\ p : C \vee D\end{array}\qquad \begin{array}{c}\Theta, [y : B]\\ \vdots\\ q : C \vee D\end{array}}{\mathsf{case}(r, x.p, y.q) : C \vee D}\vee E, x, y \qquad \begin{array}{c}\Sigma, [w : C]\\ \vdots\\ s : E\end{array}\qquad \begin{array}{c}\Phi, [z : D]\\ \vdots\\ t : E\end{array}}{\mathsf{case}(\mathsf{case}(r, x.p, y.q), w.s, z.t) : E}\vee E, w, z$$

$$\rightsquigarrow$$

$$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A \vee B\end{array}\quad \dfrac{\begin{array}{c}\Delta, [x : A]\\ \vdots\\ p : C \vee D\end{array}\quad \begin{array}{c}\Sigma, [w : C]\\ \vdots\\ s : E\end{array}\quad \begin{array}{c}\Phi, [z : D]\\ \vdots\\ t : E\end{array}}{\mathsf{case}(p, w.s, z.t) : E}\vee E, w, z \quad \dfrac{\begin{array}{c}\Theta, [y : B]\\ \vdots\\ q : C \vee D\end{array}\quad \begin{array}{c}\Sigma, [w : C]\\ \vdots\\ s : E\end{array}\quad \begin{array}{c}\Phi, [z : D]\\ \vdots\\ t : E\end{array}}{\mathsf{case}(q, w.s, z.t) : E}\vee E, w, z}{\mathsf{case}(r, x.\mathsf{case}(p, w.s, z.t), y.\mathsf{case}(q, w.s, z.t)) : E}\vee E, x, y$$

$$\boxed{\dfrac{}{\mathsf{case}(\mathsf{case}(r, x.p, y.q), w.s, z.t) \to_{\Pi} \mathsf{case}(r, x.\mathsf{case}(p, w.s, z.t), y.\mathsf{case}(q, w.s, z.t))}(\to_{\Pi})}$$

Normalization in *ND*:$\pi -$ *reductions* & $\beta -$ *reductions*

- To have a single diagram for permutations as in J.Y.Girard, and encode in type theory the behavior of rules of *ND* for more complex situations as happens when normalizing proofs, we introduce informally the notion of elimination in which we distinguish two cases:

Normalization in *ND*:π − *reductions* & β − *reductions*

- To have a single diagram for permutations as in J.Y.Girard, and encode in type theory the behavior of rules of *ND* for more complex situations as happens when normalizing proofs, we introduce informally the notion of elimination in which we distinguish two cases:

Normalization in *ND*:π − *reductions* & β − *reductions*

- To have a single diagram for permutations as in J.Y.Girard, and encode in type theory the behavior of rules of *ND* for more complex situations as happens when normalizing proofs, we introduce informally the notion of elimination in which we distinguish two cases:

$\boxed{\mathsf{e} ::= \star s \quad or \quad \mathsf{case}(\star, x.r, y.t)}$

$\boxed{\mathsf{e}\langle r\rangle = \text{“}\mathsf{e}[\star := r]\text{”}}$

Normalization in *ND*:π − *reductions* & β − *reductions*

- To have a single diagram for permutations as in J.Y.Girard, and encode in type theory the behavior of rules of *ND* for more complex situations as happens when normalizing proofs, we introduce informally the notion of elimination in which we distinguish two cases:

$$\boxed{\mathsf{e} ::= \star s \quad or \quad \mathsf{case}(\star, x.r, y.t)} \qquad\qquad \boxed{\mathsf{e}\langle r\rangle = \text{“}\mathsf{e}[\star := r]\text{”}}$$

- Eliminations in its diagrammatic form, we have:

$$\boxed{\dfrac{\star : A \supset B \qquad \begin{array}{c}\vdots\\ s : A\end{array}}{\star s : B}\supset E \qquad\qquad \dfrac{\star : A \vee B \qquad \begin{array}{c}[x : A]\\ \vdots\\ s : C\end{array} \qquad \begin{array}{c}[y : B]\\ \vdots\\ t : C\end{array}}{\mathsf{case}(\star, x.s, y.t) : C}\vee E, x, y}$$

Normalization in *ND*:π − *reductions* & β − *reductions*

- To have a single diagram for permutations as in J.Y.Girard, and encode in type theory the behavior of rules of *ND* for more complex situations as happens when normalizing proofs, we introduce informally the notion of elimination in which we distinguish two cases:

$$\boxed{\mathsf{e} ::= \star s \quad or \quad \mathsf{case}(\star, x.r, y.t)} \qquad\qquad \boxed{\mathsf{e}\langle r\rangle = \text{“}\mathsf{e}[\star := r]\text{”}}$$

- Eliminations in its diagrammatic form, we have:

$$\boxed{\dfrac{\star : A \supset B \qquad \begin{array}{c}\vdots \\ s : A\end{array}}{\star s : B}\supset E \qquad\qquad \dfrac{\star : A \vee B \qquad \begin{array}{c}[x : A] \\ \vdots \\ s : C\end{array} \qquad \begin{array}{c}[y : B] \\ \vdots \\ t : C\end{array}}{\mathsf{case}(\star, x.s, y.t) : C}\vee E, x, y}$$

Normalization in *ND*:π − *reductions* & β − *reductions*

- To have a single diagram for permutations as in J.Y.Girard, and encode in type theory the behavior of rules of *ND* for more complex situations as happens when normalizing proofs, we introduce informally the notion of elimination in which we distinguish two cases:

$$\boxed{\mathsf{e} ::= \star s \quad or \quad \mathsf{case}(\star, x.r, y.t)} \qquad\qquad \boxed{\mathsf{e}\langle r\rangle = \text{“}\mathsf{e}[\star := r]\text{”}}$$

- Eliminations in its diagrammatic form, we have:

$$\dfrac{\star : A \supset B \qquad \begin{array}{c}\vdots\\ s : A\end{array}}{\star s : B}\supset E \qquad\qquad \dfrac{\star : A \vee B \qquad \begin{array}{c}[x : A]\\ \vdots\\ s : C\end{array} \qquad \begin{array}{c}[y : B]\\ \vdots\\ t : C\end{array}}{\mathsf{case}(\star, x.s, y.t) : C}\vee E, x, y$$

Normalization in *ND*:$\pi - reductions$ & $\beta - reductions$

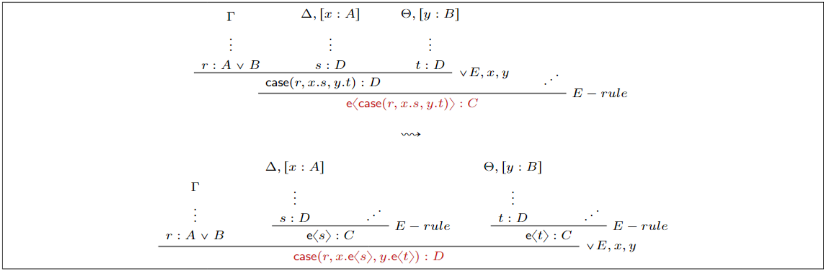
- So, finally, in a single diagram we can accommodate both instances of permutations as in J.Y. Girard (1994):

Normalization in *ND*:π − *reductions* & β − *reductions*

- So, finally, in a single diagram we can accommodate both instances of permutations as in J.Y. Girard (1994):

Normalization in *ND*:$\pi -$ *reductions* & $\beta -$ *reductions*

- So, finally, in a single diagram we can accommodate both instances of permutations as in J.Y. Girard (1994):



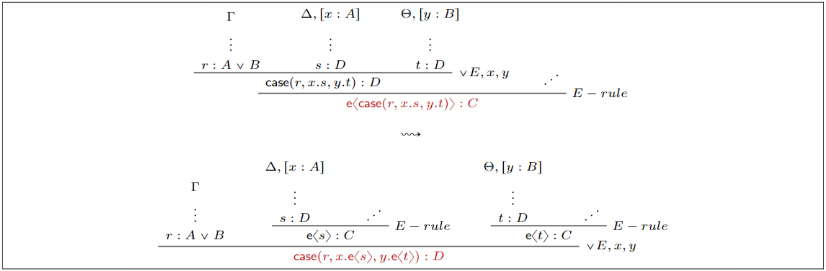
$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix} \qquad \begin{matrix}\Delta, [x : A]\\ \vdots\\ s : D\end{matrix} \qquad \begin{matrix}\Theta, [y : B]\\ \vdots\\ t : D\end{matrix}}{\mathsf{case}(r, x.s, y.t) : D}\ \vee E, x, y \qquad \iddots}{\mathsf{e}\langle \mathsf{case}(r, x.s, y.t)\rangle : C}\ E - rule$$

$$\rightsquigarrow$$

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix} \qquad \begin{matrix}\Delta, [x : A]\\ \vdots\\ \dfrac{s : D \qquad \iddots}{\mathsf{e}\langle s\rangle : C}\ E - rule\end{matrix} \qquad \begin{matrix}\Theta, [y : B]\\ \vdots\\ \dfrac{t : D \qquad \iddots}{\mathsf{e}\langle t\rangle : C}\ E - rule\end{matrix}}{\mathsf{case}(r, x.\mathsf{e}\langle s\rangle, y.\mathsf{e}\langle t\rangle) : D}\ \vee E, x, y$$

Normalization in *ND*:π $-$ *reductions* & β $-$ *reductions*

- So, finally, in a single diagram we can accommodate both instances of permutations as in J.Y. Girard (1994):



$$\dfrac{\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix} \qquad \begin{matrix}\Delta, [x : A]\\ \vdots\\ s : D\end{matrix} \qquad \begin{matrix}\Theta, [y : B]\\ \vdots\\ t : D\end{matrix}}{\mathsf{case}(r, x.s, y.t) : D}\,\vee E, x, y \qquad \cdot^{\cdot^{\cdot}}}{\mathsf{e}\langle \mathsf{case}(r, x.s, y.t)\rangle : C}\, E - rule$$

$$\rightsquigarrow$$

$$\dfrac{\begin{matrix}\Gamma\\ \vdots\\ r : A \vee B\end{matrix} \qquad \dfrac{\begin{matrix}\Delta, [x : A]\\ \vdots\\ s : D\end{matrix} \qquad \cdot^{\cdot^{\cdot}}}{\mathsf{e}\langle s\rangle : C}\, E - rule \qquad \dfrac{\begin{matrix}\Theta, [y : B]\\ \vdots\\ t : D\end{matrix} \qquad \cdot^{\cdot^{\cdot}}}{\mathsf{e}\langle t\rangle : C}\, E - rule}{\mathsf{case}(r, x.\mathsf{e}\langle s\rangle, y.\mathsf{e}\langle t\rangle) : D}\,\vee E, x, y$$

- Despite the apparent similarity between the two types of diagrams, our approach is distinct. Rather than relying on metanotation to interpret the labels that decorate proof-trees, we have internalized the meaning of these labels by assigning formal rules from the typed lambda calculus. These rules, in turn, guide the construction of the proof-trees.

Normalization in *ND*:$\pi - reductions$ & $\beta - reductions$

- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta - redex$ or a $\pi - redex$, we introduce the notion of elimination chains.

Normalization in *ND*:$\pi - reductions$ & $\beta - reductions$

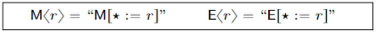
- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta - redex$ or a $\pi - redex$, we introduce the notion of elimination chains.
- Elimination chains are multiple applications of modus ponens, or arbitrary applications modus pones or case analysis:

Normalization in *ND*:$\pi - reductions$ & $\beta - reductions$

- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta - redex$ or a $\pi - redex$, we introduce the notion of elimination chains.
- Elimination chains are multiple applications of modus ponens, or arbitrary applications modus pones or case analysis:

Normalization in *ND*:$\pi -$ *reductions* & $\beta -$ *reductions*

- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta -$ *redex* or a $\pi -$ *redex*, we introduce the notion of elimination chains.
- Elimination chains are multiple applications of modus ponens, or arbitrary applications modus pones or case analysis:



$\mathsf{M}\langle r\rangle = \text{“}\mathsf{M}[\star := r]\text{”} \qquad \mathsf{E}\langle r\rangle = \text{“}\mathsf{E}[\star := r]\text{”}$

Normalization in *ND*:$\pi - \mathit{reductions}$ & $\beta - \mathit{reductions}$

- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta - \mathit{redex}$ or a $\pi - \mathit{redex}$, we introduce the notion of elimination chains.
- Elimination chains are multiple applications of modus ponens, or arbitrary applications modus pones or case analysis:

$$\boxed{\mathsf{M}\langle r\rangle = \text{“}\mathsf{M}[\star := r]\text{”} \qquad \mathsf{E}\langle r\rangle = \text{“}\mathsf{E}[\star := r]\text{”}}$$

- This helps us construct, perhaps, more intricate proof schemata which are better suited to handle more complex situations during normalization.

$$
\dfrac{\dfrac{\begin{array}{c}\Delta, [x : C_1 \supset \ldots C_n \supset (A \vee B)] \\ \ddots \\ \mathsf{M}\langle x\rangle : A \vee B\end{array} \qquad \begin{array}{c}\Gamma, [y : A] \\ \vdots \\ s : D\end{array} \qquad \begin{array}{c}\Theta, [z : B] \\ \vdots \\ t : D\end{array}}{\mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t) : D} \vee E, y, z \qquad \iddots}{\mathsf{e}\langle \mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t)\rangle : F} E - rule
$$

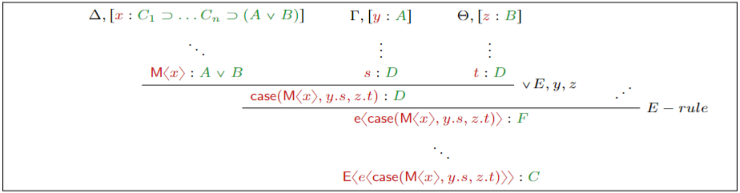
$$
\begin{array}{c} \ddots \\ \mathsf{E}\langle e\langle \mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t)\rangle\rangle : C \end{array}
$$

Normalization in *ND*:π − *reductions* & β − *reductions*

- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta -$ *redex* or a $\pi -$ *redex*, we introduce the notion of elimination chains.
- Elimination chains are multiple applications of modus ponens, or arbitrary applications modus pones or case analysis:

$$\boxed{\mathsf{M}\langle r\rangle = \text{“}\mathsf{M}[\star := r]\text{”} \qquad \mathsf{E}\langle r\rangle = \text{“}\mathsf{E}[\star := r]\text{”}}$$

- This helps us construct, perhaps, more intricate proof schemata which are better suited to handle more complex situations during normalization.



$$\Delta, [x : C_1 \supset \ldots C_n \supset (A \vee B)] \qquad \Gamma, [y : A] \qquad \Theta, [z : B]$$

$$\ddots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\dfrac{\dfrac{\mathsf{M}\langle x\rangle : A \vee B \qquad s : D \qquad t : D}{\mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t) : D}\ \vee E, y, z \qquad \iddots}{\mathsf{e}\langle \mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t)\rangle : F}\ E - rule$$

$$\ddots$$

$$\mathsf{E}\langle e\langle \mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t)\rangle\rangle : C$$

Normalization in *ND*:π − *reductions* & β − *reductions*

- As we are interested in proof schemata that can be able to show the process of proof-searching of either a $\beta - redex$ or a $\pi - redex$, we introduce the notion of elimination chains.
- Elimination chains are multiple applications of modus ponens, or arbitrary applications modus pones or case analysis:

$$\boxed{\mathsf{M}\langle r\rangle = \text{“}\mathsf{M}[\star := r]\text{”} \qquad \mathsf{E}\langle r\rangle = \text{“}\mathsf{E}[\star := r]\text{”}}$$

- This helps us construct, perhaps, more intricate proof schemata which are better suited to handle more complex situations during normalization.

$$\boxed{\begin{array}{c}
\dfrac{\dfrac{\begin{matrix}\Delta, [x : C_1 \supset \ldots C_n \supset (A \vee B)] \\ \ddots \\ \mathsf{M}\langle x\rangle : A \vee B\end{matrix} \qquad \begin{matrix}\Gamma, [y : A] \\ \vdots \\ s : D\end{matrix} \qquad \begin{matrix}\Theta, [z : B] \\ \vdots \\ t : D\end{matrix}}{\mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t) : D}\vee E, y, z \qquad \iddots}{\mathsf{e}\langle \mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t)\rangle : F}\, E - rule \\
\ddots \\
\mathsf{E}\langle e\langle \mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t)\rangle\rangle : C
\end{array}}$$

- Where we have a first, top-left most $\pi - redex$ in an $E - chain$.

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.
- The diagrammatic schemata for handling the conversions in *ND* can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.
- The diagrammatic schemata for handling the conversions in *ND* can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?
- From the structure of *ND*, we can abstract what we call a typology of proofs.

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.
- The diagrammatic schemata for handling the conversions in *ND* can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?
- From the structure of *ND*, we can abstract what we call a typology of proofs.
- This typology arises from observing that the three rules of *ND* give rise to three forms of proof, which we use to establish an inductive argument about proofs that normalize:

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.
- The diagrammatic schemata for handling the conversions in *ND* can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?
- From the structure of *ND*, we can abstract what we call a typology of proofs.
- This typology arises from observing that the three rules of *ND* give rise to three forms of proof, which we use to establish an inductive argument about proofs that normalize:
    1. H-proofs, that is proofs that end with the rule of hypothesis: x.

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.
- The diagrammatic schemata for handling the conversions in *ND* can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?
- From the structure of *ND*, we can abstract what we call a typology of proofs.
- This typology arises from observing that the three rules of *ND* give rise to three forms of proof, which we use to establish an inductive argument about proofs that normalize:
    1. H-proofs, that is proofs that end with the rule of hypothesis: x.
    2. I-proofs, proofs that end with an introduction rule: $\lambda x.r$, inl$r$ and inr$s$.

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to *Nm*, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in *ND*.
- The diagrammatic schemata for handling the conversions in *ND* can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?
- From the structure of *ND*, we can abstract what we call a typology of proofs.
- This typology arises from observing that the three rules of *ND* give rise to three forms of proof, which we use to establish an inductive argument about proofs that normalize:
  1. H-proofs, that is proofs that end with the rule of hypothesis: x.
  2. I-proofs, proofs that end with an introduction rule: $\lambda x.r$, $\mathsf{inl}r$ and $\mathsf{inr}s$.
  3. E-proofs, proofs that end with an elimination rule: rs and $\mathsf{case}(r, x.s, y.t)$.

Détour & Permutation Proof-Search: The typology of proofs

- As seen, with the addition of case analysis to Nm, we need to incorporate permutations into the conversions used to transform a non-normal proof into a normal one. However, this requires us to revise our criterion for what constitutes a normal proof in ND.
- The diagrammatic schemata for handling the conversions in ND can guide us on what to do when handling a redundancy, but they do not reveal how to search for them within a proof. So, how do we perform a proof search for détours and permutations?
- From the structure of ND, we can abstract what we call a typology of proofs.
- This typology arises from observing that the three rules of ND give rise to three forms of proof, which we use to establish an inductive argument about proofs that normalize:
  1. H-proofs, that is proofs that end with the rule of hypothesis: x.
  2. I-proofs, proofs that end with an introduction rule: $\lambda x.r$, inl$r$ and inr$s$.
  3. E-proofs, proofs that end with an elimination rule: rs and case$(r, x.s, y.t)$.
- The typology of proofs will allow us to recursively organize détour and permutation proof-search by analyzing how the major premise of an elimination rule was constructed.

Détour & Permutation Proof-Search: The typology of proofs

- Applying this insight into how convertibilities are formed in *ND* and the $\lambda - calculus$, we observe two things:

Détour & Permutation Proof-Search: The typology of proofs

- Applying this insight into how convertibilities are formed in *ND* and the $\lambda - calculus$, we observe two things:
  1. There is nothing in the rules of *ND* or the standard grammar of the $\lambda - calculus$ to prevent us from combining their rules and creating détours or permutations arbitrarily, except the conversions schemata for $\beta - reduction$ or $\pi - reduction$.

Détour & Permutation Proof-Search: The typology of proofs

- Applying this insight into how convertibilities are formed in *ND* and the $\lambda - calculus$, we observe two things:

  1. There is nothing in the rules of *ND* or the standard grammar of the $\lambda - calculus$ to prevent us from combining their rules and creating détours or permutations arbitrarily, except the conversions schemata for $\beta - reduction$ or $\pi - reduction$.

  2. Convertibilities arise from a sanctioned application of an $E - rule$. Détours, for example, result from a pathological use of the rules modus ponens or case analysis, where the major premise of an $E - rule$ is constructed using an $I - rule$. Permutations occur when the major premise of an $E - rule$ is constructed using case analysis.

Détour & Permutation Proof-Search: The typology of proofs

- For instance, having pq or $\mathrm{case}(r, x.p, y.q)$ does not allow us to distinguish between legitimate and illegitimate instances of modus ponens or case analysis. To address this issue, we rely on our typology for the analysis of $E - proofs$ to specify occurrences of the rule applied according to the form of the term p in pq or r in $\mathrm{case}(r, x.s, y.t)$:

Détour & Permutation Proof-Search: The typology of proofs

- For instance, having pq or case$(r, x.p, y.q)$ does not allow us to distinguish between legitimate and illegitimate instances of modus ponens or case analysis. To address this issue, we rely on our typology for the analysis of $E - proofs$ to specify occurrences of the rule applied according to the form of the term p in pq or r in case$(r, x.s, y.t)$:
    1. If p or r are an H-proof, we have an assumptive modus ponens or an assumptive case analysis

Détour & Permutation Proof-Search: The typology of proofs

- For instance, having pq or case$(r, x.p, y.q)$ does not allow us to distinguish between legitimate and illegitimate instances of modus ponens or case analysis. To address this issue, we rely on our typology for the analysis of $E - proofs$ to specify occurrences of the rule applied according to the form of the term p in pq or r in case$(r, x.s, y.t)$:
  1. If p or r are an H-proof, we have an assumptive modus ponens or an assumptive case analysis
  2. If p or r are an I-proof, we have a redundant modus ponens or a redundant case analysis

Détour & Permutation Proof-Search: The typology of proofs

- For instance, having pq or case$(r, x.p, y.q)$ does not allow us to distinguish between legitimate and illegitimate instances of modus ponens or case analysis. To address this issue, we rely on our typology for the analysis of $E - proofs$ to specify occurrences of the rule applied according to the form of the term p in pq or r in case$(r, x.s, y.t)$:
  1. If p or r are an H-proof, we have an assumptive modus ponens or an assumptive case analysis
  2. If p or r are an I-proof, we have a redundant modus ponens or a redundant case analysis
  3. If p or r are an E-proof, then we reiterate the process by analyzing how the major premise of that $E - proof$ was obtained. This construction gives us what we call elimination chains, which are nested applications of $E - rules$, possibly empty, of modus ponens or case analysis. $E - chains$ cannot grow indefinitely, and the analysis continues until we eventually hit a détour or a permutation.

     Détour & permutation proof-search terminates.

Détour & Permutation Proof-Search: The typology of proofs

- For instance, having pq or $\text{case}(r, x.p, y.q)$ does not allow us to distinguish between legitimate and illegitimate instances of modus ponens or case analysis. To address this issue, we rely on our typology for the analysis of $E - proofs$ to specify occurrences of the rule applied according to the form of the term p in pq or r in $\text{case}(r, x.s, y.t)$:
    1. If p or r are an H-proof, we have an assumptive modus ponens or an assumptive case analysis
    2. If p or r are an I-proof, we have a redundant modus ponens or a redundant case analysis
    3. If p or r are an E-proof, then we reiterate the process by analyzing how the major premise of that $E - proof$ was obtained. This construction gives us what we call elimination chains, which are nested applications of $E - rules$, possibly empty, of modus ponens or case analysis. $E - chains$ cannot grow indefinitely, and the analysis continues until we eventually hit a détour or a permutation.

       Détour & permutation proof-search terminates.
- The intuition behind this maneuver is that by analyzing all possible applications and reductions within the $\lambda - calculus$, we have all the necessary elements to fully describe the set of well-formed and well-typed expressions of $\lambda^{\supset,\vee}$ that strongly normalize.

Détour & Permutation Proof-Search: GPR

- We refine our approach to analyzing E-proofs using the typology of proofs by introducing a formal grammar that captures E-chains' construction.

Détour & Permutation Proof-Search: GPR

- We refine our approach to analyzing E-proofs using the typology of proofs by introducing a formal grammar that captures E-chains' construction.

E-Chains Grammar

$\mathsf{E} ::= \star \mid e\langle E\rangle \qquad M ::= \star \mid (\star s)\langle M\rangle \qquad e ::= \; \star s \mid \mathsf{case}(\star, y.s, y.t)$

Détour & Permutation Proof-Search: GPR

- We refine our approach to analyzing E-proofs using the typology of proofs by introducing a formal grammar that captures E-chains' construction.

E-Chains Grammar

$\mathsf{E} ::= \star \mid e\langle E\rangle \qquad M ::= \star \mid (\star s)\langle M\rangle \qquad e ::= \ \star s \mid \mathsf{case}(\star, y.s, y.t)$

GPR Grammar

$r, s, t ::= x \mid \lambda x.r \mid \mathsf{inl}r \mid \mathsf{inr}s \mid E\langle(\lambda x.r)s \mid M\langle x\rangle s \mid \mathsf{case}(M\langle x\rangle, y.s, z.t) \mid \mathsf{E}\langle\mathsf{case}(\mathsf{inl}r, x.p, y.q)\rangle$

$\mid \mathsf{E}\langle\mathsf{case}(\mathsf{inrs}, x.p, y.q)\rangle \mid \mathsf{E}\langle\mathsf{e}\langle\mathsf{case}(M\langle x\rangle, y.r, z.\mathsf{t})\rangle\rangle$

Détour & Permutation Proof-Search: GPR

- We refine our approach to analyzing E-proofs using the typology of proofs by introducing a formal grammar that captures E-chains' construction.

E-Chains Grammar

$\mathrm{E} ::= \star \mid e\langle E\rangle \qquad M ::= \star \mid (\star s)\langle M\rangle \qquad e ::= \ \star s \mid \mathsf{case}(\star, y.s, y.t)$

GPR Grammar

$r, s, t ::= x \mid \lambda x.r \mid \mathsf{inl}r \mid \mathsf{inr}s \mid E\langle(\lambda x.r)s \mid M\langle x\rangle s \mid \mathsf{case}(M\langle x\rangle, y.s, z.t) \mid \mathrm{E}\langle \mathsf{case}(\mathsf{inl}r, x.p, y.q)\rangle$

$\mid \mathrm{E}\langle \mathsf{case}(\mathsf{inr}s, x.p, y.q)\rangle \mid \mathrm{E}\langle \mathrm{e}\langle \mathsf{case}(M\langle x\rangle, y.r, z.t)\rangle\rangle$

- This allows us to incorporate our distinction between assumptive and redundant occurrences of modus ponens or case analysis within the application of respective rules, which features in a new grammar for $\lambda^{\supset,\vee}$.

Détour & Permutation Proof-Search: GPR

- We refine our approach to analyzing E-proofs using the typology of proofs by introducing a formal grammar that captures E-chains' construction.

E-Chains Grammar

$\mathsf{E} ::= \star \mid e\langle E\rangle \qquad M ::= \star \mid (\star s)\langle M\rangle \qquad e ::= \ \star s \mid \mathsf{case}(\star, y.s, y.t)$

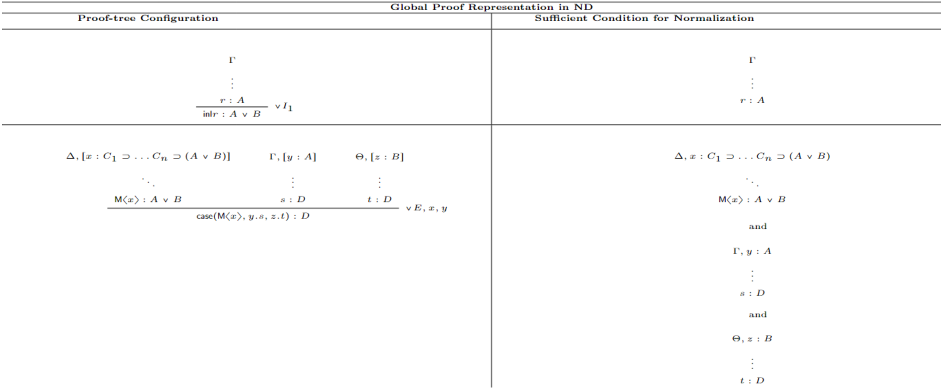
GPR Grammar

$r, s, t ::= x \mid \lambda x.r \mid \mathsf{inl} r \mid \mathsf{inr} s \mid E\langle(\lambda x.r)s \mid M\langle x\rangle s \mid \mathsf{case}(M\langle x\rangle, y.s, z.t) \mid \mathsf{E}\langle\mathsf{case}(\mathsf{inl} r, x.p, y.q)\rangle$

$\mid \mathsf{E}\langle\mathsf{case}(\mathsf{inr} s, x.p, y.q)\rangle \mid \mathsf{E}\langle\mathsf{e}\langle\mathsf{case}(M\langle x\rangle, y.r, z.t)\rangle\rangle$

- This allows us to incorporate our distinction between assumptive and redundant occurrences of modus ponens or case analysis within the application of respective rules, which features in a new grammar for $\lambda^{\supset,\vee}$.
- GPR stands for Global Proof Representation because the terms obtained through this grammar are exhaustive with respect to the usual classification and help us achieve a finer granularity in normalization results by introducing E-chains.

Détour & Permutation: GPR sufficient conditions



Global Proof Representation in ND	
Proof-tree Configuration	Sufficient Condition for Normalization
$\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ r : A\end{array}}{\mathsf{inl}r : A \vee B}\,\vee I_1$	$\begin{array}{c}\Gamma\\ \vdots\\ r : A\end{array}$
$\dfrac{\begin{array}{c}\Delta, [x : C_1 \supset \ldots C_n \supset (A \vee B)]\\ \ddots\\ \mathsf{M}\langle x\rangle : A \vee B\end{array} \quad \begin{array}{c}\Gamma, [y : A]\\ \vdots\\ s : D\end{array} \quad \begin{array}{c}\Theta, [z : B]\\ \vdots\\ t : D\end{array}}{\mathsf{case}(\mathsf{M}\langle x\rangle, y.s, z.t) : D}\,\vee E, x, y$	$\begin{array}{c}\Delta, x : C_1 \supset \ldots C_n \supset (A \vee B)\\ \ddots\\ \mathsf{M}\langle x\rangle : A \vee B\end{array}$ and $\begin{array}{c}\Gamma, y : A\\ \vdots\\ s : D\end{array}$ and $\begin{array}{c}\Theta, z : B\\ \vdots\\ t : D\end{array}$

Détour & Permutation Proof-Search: GPR sufficient conditions

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\Delta\\ \vdots\\ s : A\end{array}}{\mathsf{inr}\,s : A \vee B}\vee I_2 \qquad \begin{array}{c}\Gamma, [x : A]\\ \vdots\\ p : D\end{array} \qquad \begin{array}{c}\Theta, [y : B]\\ \vdots\\ q : D\end{array}}{\mathsf{case}(\mathsf{inr}\,s, x.p, y.q) : D}\vee E, x, y\\ \ddots\\ \mathsf{E}\langle\mathsf{case}(\mathsf{inr}\,s, x.p, y.q)\rangle : C\end{array}$$

$$\begin{array}{c}\Delta\\ \vdots\\ \Theta, \cancel{s : B}\\ \vdots\\ q[y := s] : D\\ \ddots\\ \mathsf{E}\langle q[y := s]\rangle : C\end{array}$$

and

$$\begin{array}{c}\Gamma, x : A\\ \vdots\\ p : D\end{array}$$

and

$$\begin{array}{c}\Theta, y : B\\ \vdots\\ q : D\end{array}$$

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\Delta, [x : C_1 \supset \dots C_n \supset (A \vee B)]\\ \ddots\\ \mathsf{M}\langle x\rangle : A \vee B\end{array} \qquad \begin{array}{c}\Gamma, [y : A]\\ \vdots\\ r : D\end{array} \qquad \begin{array}{c}\Theta, [z : B]\\ \vdots\\ s : D\end{array}}{\mathsf{case}(\mathsf{M}\langle x\rangle, y.r, z.t) : D}\vee E, y, z \qquad \cdot^{\cdot^{\cdot}}}{\mathsf{e}\langle\mathsf{case}(\mathsf{M}\langle x\rangle, y.r, z.t)\rangle : F}E - rule\\ \ddots\\ \mathsf{E}\langle e\langle\mathsf{case}(\mathsf{M}\langle x\rangle, y.r, z.t)\rangle\rangle : C\end{array}$$

$$\begin{array}{c}\Delta, [x : C_1 \supset \dots C_n \supset (A \vee B)]\\ \ddots\\ \mathsf{M}\langle x\rangle : A \vee B\\ \vdots\\ \mathsf{e}\langle\mathsf{case}(\mathsf{M}\langle x\rangle, y.r, z.t)\rangle : F\\ \ddots\\ \mathsf{E}\langle\mathsf{e}\langle\mathsf{case}(\mathsf{M}\langle x\rangle, y.r, z.t)\rangle\rangle : C\end{array}$$

and

Main Lemma

- Previous GPR conditions, captured diagrammatically, translate algebraically into the rules that describe the set of strongly normalizing terms: *SN*.

$$\frac{r \in SN}{\mathsf{inl}\, r \in SN}\ (SN_{\mathsf{inl}})$$

$$\frac{s \in SN}{\mathsf{inr}\, s \in SN}\ (SN_{\mathsf{inr}})$$

$$\frac{M\langle x\rangle \in SN \qquad s \in SN \qquad t \in SN}{\mathsf{case}(M\langle x\rangle, y.s, z.t) \in SN}\ (SN_{M_{\mathsf{case}}})$$

$$\frac{r \in SN}{\lambda x.r \in SN}\ (SN_{\lambda})$$

$$\frac{}{x \in SN}\ (SN_{Var})$$

$$\frac{\mathsf{E}\langle \mathsf{case}(M\langle x\rangle, y.\mathsf{e}\langle r\rangle, z.\mathsf{e}\langle s\rangle)\rangle \in SN}{\mathsf{E}\langle \mathsf{e}\langle \mathsf{case}(M\langle x\rangle, y.r, z.s)\rangle\rangle \in SN}\ (SN_{\pi})$$

$$\frac{M\langle x\rangle \in SN \qquad s \in SN}{M\langle x\rangle s \in SN}\ (SN_{Mmp})$$

$$\frac{\mathsf{E}\langle p[x := r]\rangle \in SN \qquad \mathsf{E}\langle p\rangle \in SN \qquad r \in SN}{\mathsf{E}\langle \mathsf{case}(\mathsf{inl}\, r, x.p, y.q)\rangle \in SN}\ (SN_{\beta_{\vee l}})$$

$$\frac{\mathsf{E}\langle q[y := s]\rangle \in SN \qquad \mathsf{E}\langle q\rangle \in SN \qquad s \in SN}{\mathsf{E}\langle \mathsf{case}(\mathsf{inr}\, s, x.p, y.q)\rangle \in SN}\ (SN_{\beta_{\vee r}})$$

$$\frac{\mathsf{E}\langle r[x := s]\rangle \in SN \qquad s \in SN}{\mathsf{E}\langle (\lambda x.r)s\rangle \in SN}\ (SN_{\beta_{\supset}})$$

Main Lemma

SN is closed under modus ponens, case analysis and substitution.

Main Lemma: Proof

Main Lemma in its Original Version (Romantic):

If $r, s, t \in SN$ then $rs \in SN$, $\mathrm{case}(r, x.s, y.t) \in SN$, and $r[x := s] \in SN$. *NB: All terms are assumed to be typable.*

Main Lemma in its Cool Version:

$$\forall A \forall r\ (r \in SN \Rightarrow (\forall \Delta \forall B\ (\Delta \vdash r : A \supset B \Rightarrow \forall s\ (\Gamma \vdash s : A \Rightarrow s \in SN \Rightarrow rs \in SN))))$$

$$\wedge \forall B \forall C \forall \Gamma\ (A = B \vee C \Rightarrow \Gamma \vdash r : A \Rightarrow \forall D \forall \Delta \forall x \forall s\ (\Delta, x : B \vdash s : D \Rightarrow s \in SN \Rightarrow$$

$$\forall \Sigma \forall y \forall t\ (\Sigma, y : C \vdash t : D \Rightarrow t \in SN \Rightarrow \mathrm{case}(r, x.s, y.t) \in SN))$$

$$\wedge\ \forall s\ (\Gamma \vdash s : A \Rightarrow s \in SN\ \ \Rightarrow \forall \Delta \forall c \forall x\ (\Delta, x : A \vdash r : C \Rightarrow r[x := s] \in SN))$$

- **Proof:** By simultaneous induction on the formula A and $r \in SN$.■

  Moral: Mechanizing a mathematical proof is a challenging task.

$ND\text{−}\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t \ \in \mathsf{SN}$

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$

$ND\!-\!\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t \ \in$ SN

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$
    - Basis: $t = x$, immediate since $x \in SN$ by rule (SN_{Var}).

$ND\!-\!\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t\ \in$ SN

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$
  - Basis: $t = x$, immediate since $x \in SN$ by rule (SN_{Var}).
  - Inductive step : $t = \lambda x.r$, immediate from the I.H $(r \in SN)$ and rule (SN_{λ}).

$ND{-}\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t \ \in \mathsf{SN}$

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$
  - Basis: $t = x$, immediate since $x \in SN$ by rule (SN_{Var}).
  - Inductive step : $t = \lambda x.r$, immediate from the I.H ($r \in SN$) and rule (SN_{λ}).
  - Inductive step : $t = \mathsf{inl} r$, immediate from the I.H ($r \in SN$) and rule (SN_{inl}).

$ND\text{−}\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t \ \in \mathsf{SN}$

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$
  - Basis: $t = x$, immediate since $x \in SN$ by rule (SN_{Var}).
  - Inductive step : $t = \lambda x.r$, immediate from the I.H $(r \in SN)$ and rule (SN_λ).
  - Inductive step : $t = \mathsf{inl} r$, immediate from the I.H $(r \in SN)$ and rule (SN_{inl}).
  - Inductive step : $t = \mathsf{inr} s$, immediate from the I.H $(s \in SN)$ and rule (SN_{inr}).

$ND-\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t \ \in \mathsf{SN}$

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$
  - Basis: $t = x$, immediate since $x \in SN$ by rule (SN_{Var}).
  - Inductive step : $t = \lambda x.r$, immediate from the I.H $(r \in SN)$ and rule (SN_{λ}).
  - Inductive step : $t = \mathsf{inl} r$, immediate from the I.H $(r \in SN)$ and rule (SN_{inl}).
  - Inductive step : $t = \mathsf{inr} s$, immediate from the I.H $(s \in SN)$ and rule (SN_{inr}).
  - Inductive step: $t = rs$. Immediate from the I.H. $(r, s \in SN)$ and the main lemma.

$ND{-}\lambda^{\supset,\vee}$ strongly normalizes

Theorem

If $\Gamma \vdash t : A$ then $t \ \in \mathrm{SN}$

- Proof: By induction on the derivability relation $\Gamma \vdash t : A$
  - Basis: $t = x$, immediate since $x \in SN$ by rule (SN_{Var}).
  - Inductive step : $t = \lambda x.r$, immediate from the I.H $(r \in SN)$ and rule (SN_{λ}).
  - Inductive step : $t = \mathrm{inl}r$, immediate from the I.H $(r \in SN)$ and rule (SN_{inl}).
  - Inductive step : $t = \mathrm{inr}s$, immediate from the I.H $(s \in SN)$ and rule (SN_{inr}).
  - Inductive step: $t = rs$. Immediate from the I.H. $(r, s \in SN)$ and the main lemma.
  - Inductive step: $t = \mathrm{case}(r, x.p, y.q)$. Immediate from the I.H. $(r, p, q \in SN)$, and the main lemma, which guarantees that SN is closed under modus ponens, case analysis and substitution.■

Final remarks

- Our interest in this research has been to lay the foundation for a conciliatory proof.

Final remarks

- Our interest in this research has been to lay the foundation for a conciliatory proof.
- This reconciliation involves adapting the formal rules of the lambda calculus, which is amenable to mechanization, to the act of proving in natural deduction by supplementing with diagrammatic constructions that guide our intuition in areas that are too abstract for our imagination.

Final remarks

- Our interest in this research has been to lay the foundation for a conciliatory proof.
- This reconciliation involves adapting the formal rules of the lambda calculus, which is amenable to mechanization, to the act of proving in natural deduction by supplementing with diagrammatic constructions that guide our intuition in areas that are too abstract for our imagination.
- Our proposal addresses a gap between proof theory and the lambda calculus concerning how normalization results have been approached through the Curry-Howard correspondence without tipping the balance towards either side of the algebra/geometry dichotomy.

Final remarks

- Our interest in this research has been to lay the foundation for a conciliatory proof.
- This reconciliation involves adapting the formal rules of the lambda calculus, which is amenable to mechanization, to the act of proving in natural deduction by supplementing with diagrammatic constructions that guide our intuition in areas that are too abstract for our imagination.
- Our proposal addresses a gap between proof theory and the lambda calculus concerning how normalization results have been approached through the Curry-Howard correspondence without tipping the balance towards either side of the algebra/geometry dichotomy.
- Furthermore, it opens a new avenue in the formalization and mechanization of proofs by contrasting and reconciling two formally equivalent systems but epistemologically very diverse.

References

- Girard, J.Y.,Taylor, P.& Lafont, Y. (trans.) (1990). Proofs and Types. UK: Cambridge University Press.

- Joachimski, F., & Matthes, R. (2002, December 12). Short proofs of normalization for the simply-typed lambda calculus, permutative conversions and Gödel's T. Arch. Math. Logic, 42, 59-87.

- Mancosu, P., Galvan, S., & Zach, R. (2021). An Introduction to Proof Theory: Normalization, Cut-Elimination, and Consistency Proofs. UK: Oxford University Press.

- Miranda-Perea, F. E., and González-Huesca, L. d. C. (2022). On the Conciliation of Traditional and Computer-Assisted Proofs. In Philosophy of Computing: Themes from the IACAP 2019 (pp. 73-112). Springer.

¡Muchas gracias por su atención!