Deep Inference for Graphical Theorem Proving

Pablo Donato

2024-05-29

Institut Polytechnique de Paris, LIX, PARTOUT

PhD defense, Palaiseau

Supervised by Benjamin Werner & Pierre-Yves Strub

Introduction

Context

Logic

- Study of *sound* reasoning

Logic

- Study of *sound* reasoning
- Example of everyday life deduction:

$$\frac{\text{it rains 🌧️} \qquad \text{you don't have ☂️}}{\text{you are wet 💧}}$$

- Study of *sound* reasoning
- Example of everyday life deduction:

premisses it rains 🌧️ you don't have ☂️

conclusion you are wet 💧

Logic

- Study of *sound* reasoning
- Example of everyday life deduction:

premisses

conclusion

it rains 🌧️ you don't have ☂️ you are outside

you are not under a bus shelter …

---

you are wet 💧

Logic

- Study of *sound* reasoning
- Example of everyday life deduction:

premisses	it rains 🌧️ you don't have ☂️ you are outside you are not under a bus shelter …
conclusion	you are wet 💧

- Hidden assumptions ⇒ lack of **certainty**

Formal logic

Let's try another one (Aristotle – 4th century BC):

$$\frac{\textit{Socrates is human} \qquad \textit{All humans are mortal}}{\textit{Socrates is mortal}}$$

Formal logic

Let's try another one (Aristotle – 4th century BC):

$$\frac{\textit{Socrates is human} \qquad \textit{All humans are mortal}}{\textit{Socrates is mortal}}$$

- Better! But *why* does it hold?

Formal logic

Let’s try another one (Aristotle – 4th century BC):

$$\frac{\textit{Socrates is human} \qquad \textit{All humans are mortal}}{\textit{Socrates is mortal}}$$

- Better! But *why* does it hold?
- **Forget** everything about *reality*:

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q}$$

Formal logic

Let's try another one (Aristotle – 4th century BC):

$$\frac{\textit{Socrates is human} \qquad \textit{All humans are mortal}}{\textit{Socrates is mortal}}$$

- Better! But *why* does it hold?
- **Forget** everything about *reality*:

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q}$$

↪ Formal essence of logical reasoning

Mathematical logic

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q}$$

Mathematical logic

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q} \quad \rightsquigarrow \quad \frac{P(x) \qquad \forall y.P(y) \Rightarrow Q(y)}{Q(x)}$$

Mathematical logic

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q} \quad \rightsquigarrow \quad \frac{P(x) \qquad \forall y. P(y) \Rightarrow Q(y)}{Q(x)}$$

- *Generic patterns* of deduction as **rules**

Mathematical logic

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q} \quad \rightsquigarrow \quad \frac{P(x) \qquad \forall y. P(y) \Rightarrow Q(y)}{Q(x)}$$

- *Generic patterns* of deduction as **rules**
- Formalist school (Hilbert – 20th century):

Maths as a huge **game**

Goal: to *prove theorems* by following *inference rules*

Mathematical logic

$$\frac{x \text{ is } P \qquad \text{All } P \text{ are } Q}{x \text{ is } Q} \quad \rightsquigarrow \quad \frac{P(x) \qquad \forall y. P(y) \Rightarrow Q(y)}{Q(x)}$$

- *Generic patterns* of deduction as **rules**
- Formalist school (Hilbert – 20th century):

Maths as a huge **game**

Goal: to *prove theorems* by following *inference rules*

- **Proof theory:** design & study of *rule systems* capturing maths

Proof assistants

- Inference rules represented with **symbols**

Proof assistants

- Inference rules represented with **symbols**
- **Computers** very good at *manipulating symbols* and *following rules*

Proof assistants

- Inference rules represented with **symbols**
- **Computers** very good at *manipulating symbols* and *following rules*

↳ Teach computers how to do maths with proof theory!

Proof assistants

- Inference rules represented with **symbols**
- **Computers** very good at *manipulating symbols* and *following rules*

↳ Teach computers how to do maths with proof theory!

- Problem: maths is *hard* ⇒ need for a **human** in the loop

↳ Interactive Theorem Provers (ITPs)

Contributions

Textual vs. Graphical

State-of-the art: build proofs by writing **textual** commands

Textual vs. Graphical

State-of-the art: build proofs by writing **textual** commands

🙇 : “Please apply *this* rule”

Textual vs. Graphical

State-of-the art: build proofs by writing **textual** commands

🙇 : “Please apply *this* rule”

💻 : “✅ OK here is the result!”

Textual vs. Graphical

State-of-the art: build proofs by writing **textual** commands

🙇 : “Please apply *this* rule”

💻 : “❌ ERROR: dkfsljfjdklsfjdkfjsldjfkdlsfj”

Textual vs. Graphical

State-of-the art: build proofs by writing **textual** commands

🙇 : “Please apply *this* rule”

💻 : “❌ ERROR: dkfsljfjdklsfjdkfjsldjfkdlsfj”

1st contribution: build proofs by direct manipulation of *formulas*

↳ No need to *memorize* the rules

↳ More *straightforward* interaction

Symbolic vs. Iconic

- **Symbols** are hard to:
  - *learn* ⇒ purely conventional meaning
  - *manipulate* ⇒ need for very precise gestures

Symbolic vs. Iconic

- **Symbols** are hard to:
  - *learn* $\Rightarrow$ purely conventional meaning
  - *manipulate* $\Rightarrow$ need for very precise gestures
- Formulas can **interact** by being *moved* in the same space

Symbolic vs. Iconic

- **Symbols** are hard to:
  - *learn* $\Rightarrow$ purely conventional meaning
  - *manipulate* $\Rightarrow$ need for very precise gestures
- Formulas can **interact** by being *moved* in the same space

2nd contribution: replace logical symbols by geometrical diagrams

Symbolic Manipulations

Proof-by-Action

*A **demo** is worth a thousand words!*

Paradigm

- Fully graphical: no textual proof language

Paradigm

- Fully graphical: no textual proof language
- Both spatial and temporal:

$$\text{proof} \quad = \quad \text{gesture sequence}$$

Paradigm

- Fully graphical: no textual proof language
- Both spatial and temporal:

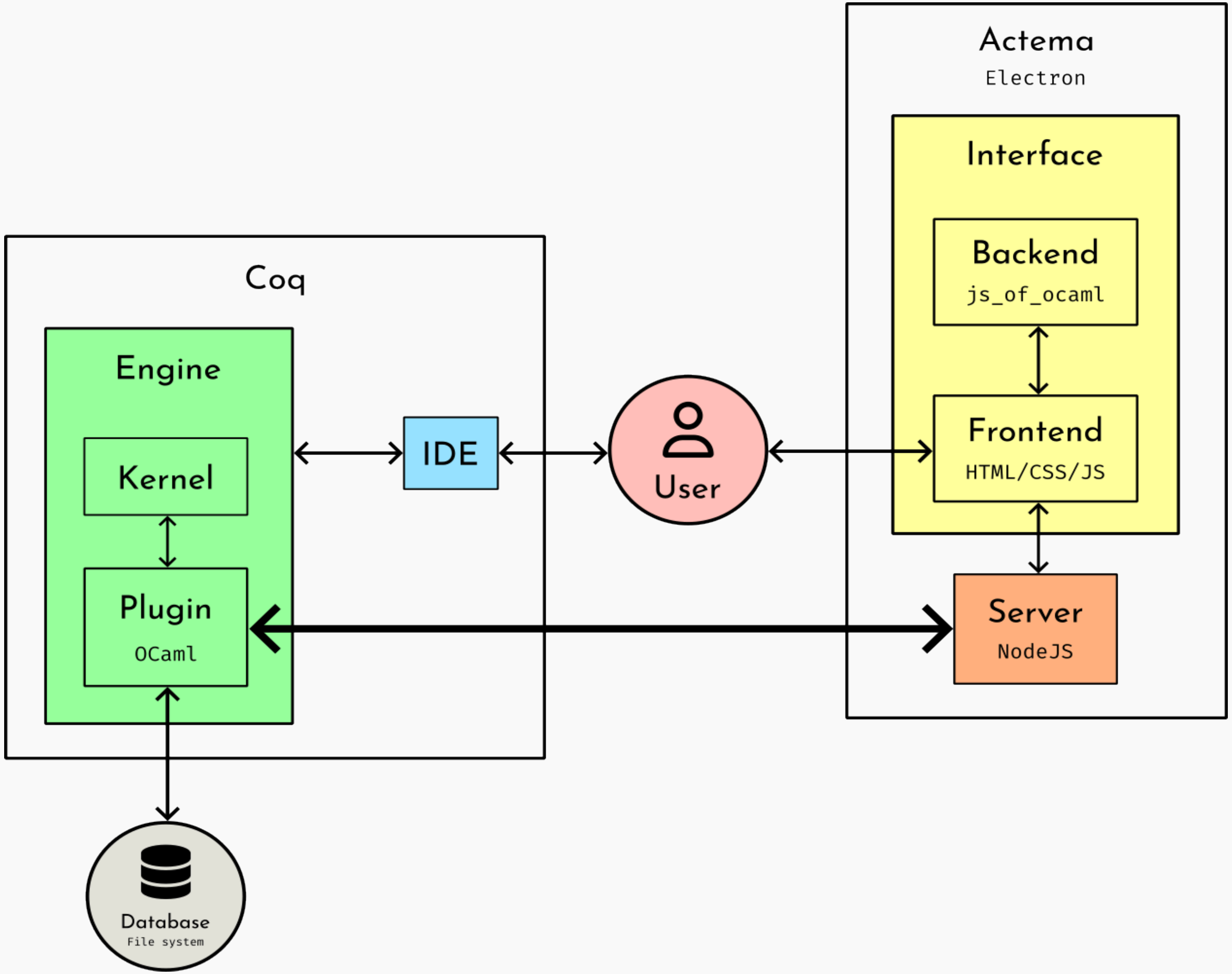
$$\text{proof} \quad = \quad \text{gesture sequence}$$

- **Different modes** of reasoning with a **single "syntax"**:

$$\begin{array}{rcl} \text{Click} & \Longleftrightarrow & \text{introduction/elimination} \\ \text{Drag-and-Drop} & \Longleftrightarrow & \text{backward/forward} \end{array}$$

coq-actema

(Bouverot, Donato, Najjar, Strub, Werner)



Semantics of Drag-and-Drop

Subformula Linking (Chaudhuri 2013)

Idea: bring matching subformulas through switch rules

$$\underline{A} \wedge B \circledast B \wedge (\underline{A} \vee C) \wedge D$$

Subformula Linking (Chaudhuri 2013)

Idea: bring matching subformulas through switch rules

$$\text{switch}\left\{\begin{array}{ll} & \underline{A} \wedge B \circledS B \wedge (\underline{A} \vee C) \wedge D \\ \rightarrow & B \wedge (\underline{A} \wedge B \circledS (\underline{A} \vee C) \wedge D) \\ \rightarrow & B \wedge (\underline{A} \wedge B \circledS \underline{A} \vee C) \wedge D \\ \rightarrow & B \wedge ((\underline{A} \wedge B \circledS \underline{A}) \vee C) \wedge D \\ \rightarrow & B \wedge ((B \Rightarrow \underline{A} \circledS A) \vee C) \wedge D \end{array}\right.$$

Subformula Linking

(Chaudhuri 2013)

Idea: bring matching subformulas through switch rules

$$\text{switch}\begin{cases} \underline{A} \wedge B \circledcirc B \wedge (\underline{A} \vee C) \wedge D \\ \to B \wedge (\underline{A} \wedge B \circledcirc (\underline{A} \vee C) \wedge D) \\ \to B \wedge (\underline{A} \wedge B \circledcirc \underline{A} \vee C) \wedge D \\ \to B \wedge ((\underline{A} \wedge B \circledcirc \underline{A}) \vee C) \wedge D \\ \to B \wedge ((B \Rightarrow \underline{A} \circledcirc \underline{A}) \vee C) \wedge D \end{cases}$$

$$\text{identity}\begin{cases} \to B \wedge ((B \Rightarrow \top) \vee C) \wedge D \end{cases}$$

Subformula Linking (Chaudhuri 2013)

Idea: bring matching subformulas through switch rules

$$
\begin{aligned}
&\text{switch} \begin{cases}
& \underline{A} \wedge B \circledcirc B \wedge (\underline{A} \vee C) \wedge D \\
\to & B \wedge (\underline{A} \wedge B \circledcirc (\underline{A} \vee C) \wedge D) \\
\to & B \wedge (\underline{A} \wedge B \circledcirc \underline{A} \vee C) \wedge D \\
\to & B \wedge ((\underline{A} \wedge B \circledcirc \underline{A}) \vee C) \wedge D \\
\to & B \wedge ((B \Rightarrow \underline{A} \circledcirc \underline{A}) \vee C) \wedge D
\end{cases} \\
&\text{identity} \begin{cases} \to B \wedge ((B \Rightarrow \top) \vee C) \wedge D \end{cases} \\
&\text{unit elimination} \begin{cases}
\to & B \wedge (\top \vee C) \wedge D \\
\to & B \wedge \top \wedge D \\
\to & B \wedge D
\end{cases}
\end{aligned}
$$

Subformula Linking (Chaudhuri 2013)

Idea: bring matching subformulas through switch rules

$$
\begin{aligned}
\text{switch} &\left\{\begin{array}{l}
\quad \underline{A} \land B \circledcirc B \land (\underline{A} \lor C) \land D \\
\to B \land (\underline{A} \land B \circledcirc (\underline{A} \lor C) \land D) \\
\to B \land (\underline{A} \land B \circledcirc \underline{A} \lor C) \land D \\
\to B \land ((\underline{A} \land B \circledcirc \underline{A}) \lor C) \land D \\
\to B \land ((B \Rightarrow \underline{A} \circledcirc \underline{A}) \lor C) \land D
\end{array}\right. \\
\text{identity} &\left\{\ \to B \land ((B \Rightarrow \top) \lor C) \land D\right. \\
\text{unit elimination} &\left\{\begin{array}{l}
\to B \land (\top \lor C) \land D \\
\to B \land \top \land D \\
\to B \land D
\end{array}\right.
\end{aligned}
$$

Variant of the **Calculus of Structures** (Guglielmi 1999)

$$\exists y.\ \forall x.\underline{R(x, y)} \circledcirc \forall a.\exists b.\underline{R(a, b)}$$

Linking under quantifiers (Donato, Strub, and Werner 2022)

- Unify linked subformulas

$$\exists y.\, \forall x.\underline{R(x,y)} \otimes \forall a.\exists b.\underline{R(a,b)}$$

$$x \longmapsto a$$

$$y \longleftarrow\!\!\mid\; b$$

Linking under quantifiers (Donato, Strub, and Werner 2022)

- Unify linked subformulas
- **Check** for ∀∃ dependency cycles

$$\exists y.\, \forall x.\underline{R(x,y)} \otimes \forall a.\exists b.\underline{R(a,b)}$$

$$x \longmapsto a$$

$$y \longleftarrow\!\dashv b$$

✓

Linking under quantifiers (Donato, Strub, and Werner 2022)

- Unify linked subformulas
- **Check** for ∀∃ dependency cycles
- **Switch** uninstantiated quantifiers

$$\exists y.\, \forall x.\underline{R(x,y)} \otimes \forall a.\exists b.\underline{R(a,b)}$$
$$\to \forall y.\left(\forall x.\underline{R(x,y)} \otimes \forall a.\, \exists b.\underline{R(a,b)}\right)$$
$$\to \forall y.\forall a.\left(\forall x.\underline{R(x,y)} \otimes \exists b.\, \underline{R(a,b)}\right)$$

$$x \longmapsto a$$

$$y \longleftarrow\!\!\dashv b$$

✓

Linking under quantifiers (Donato, Strub, and Werner 2022)

- Unify linked subformulas
- **Check** for ∀∃ dependency cycles
- **Switch** uninstantiated quantifiers
- **Instantiate** unified variables

$$\exists y. \forall x. R(x, y) \otimes \forall a. \exists b. R(a, b)$$
$$\to \forall y. \big(\forall x. R(x, y) \otimes \forall a. \exists b. R(a, b)\big)$$
$$\to \forall y. \forall a. \big(\forall x. R(x, y) \otimes \exists b. R(a, b)\big)$$
$$\to \forall y. \forall a. \big(\forall x. R(x, y) \otimes R(a, y)\big)$$
$$\to \forall y. \forall a. \big(R(a, y) \otimes R(a, y)\big)$$
$$\to^{*} \top$$

$$x \longmapsto a$$

$$y \longleftarrow\!\shortmid\; b$$

✓

Linking under quantifiers (Donato, Strub, and Werner 2022)

- Unify linked subformulas

$$\forall a.\exists b.\underline{R(a,b)} \otimes \exists y.\forall x.\underline{R(x,y)}$$

$$x \longleftarrow\!\dashv a$$

$$y \longmapsto b$$

Linking under quantifiers (Donato, Strub, and Werner 2022)

- Unify linked subformulas
- **Check** for ∀∃ dependency cycles

$$\forall a.\exists b.\underline{R(a,b)} \ogreaterthan \exists y.\forall x.\underline{R(x,y)}$$

$$\begin{tikzcd} x \arrow[d, bend right] & a \arrow[l, mapsto] \\ y \arrow[r, mapsto] & b \arrow[u, bend right] \end{tikzcd}$$

✗

Rewriting equalities (Donato, Strub, and Werner 2022)

Add 4 rules $\Longrightarrow$ `rewrite` tactic for free!

$$\underline{t} = u \otimes A \quad \rightarrow \quad A[u/t] \qquad\qquad t = \underline{u} \otimes A \quad \rightarrow \quad A[t/u]$$

$$\underline{t} = u \circledast A \quad \rightarrow \quad A[u/t] \qquad\qquad t = \underline{u} \circledast A \quad \rightarrow \quad A[t/u]$$

Rewriting equalities (Donato, Strub, and Werner 2022)

Add 4 rules $\Longrightarrow$ `rewrite` tactic for free!

$$\underline{t} = u \circledcirc A \quad \to \quad A[u/t] \qquad\qquad t = \underline{u} \circledcirc A \quad \to \quad A[t/u]$$

$$\underline{t} = u \circledast A \quad \to \quad A[u/t] \qquad\qquad t = \underline{u} \circledast A \quad \to \quad A[t/u]$$

Compositional with semantics of **connectives**:

- **Quantifiers:** rewrite modulo *unification*
- **Implication:** *conditional* rewrite
- **Arbitrary** combinations are possible:

$$\forall x. x \neq 0 \Rightarrow \underline{f(x)} = g(x) \circledcirc \exists y. A\left(\underline{f(y)}\right) \vee B(y)$$

$$\to^* \ \exists y. (y \neq 0 \wedge A(g(y))) \vee B(y)$$

Completeness

Add the following rules:

- Init $C^{+}\boxed{A \Rightarrow B} \to C^{+}\boxed{A \circledcirc B}$ $C^{-}\boxed{A \wedge B} \to C^{-}\boxed{A \circledast B}$
- Release $C^{+}\boxed{A \circledcirc B} \to C^{+}\boxed{A \Rightarrow B}$ $C^{-}\boxed{A \circledast B} \to C^{-}\boxed{A \wedge B}$
- Contraction $C^{-}\boxed{A} \to C^{-}\boxed{A \wedge A}$

Theorem (Completeness): If $\Gamma \vdash A$ is provable in sequent calculus, then

$$\bigwedge \Gamma \Rightarrow A \to^{*} \top$$

Conclusion

- `coq-actema` still in development, but *already usable*
  ↳ follow install instructions on GitHub!

- Based on the solid proof theory of **subformula linking**

- Next step: exposure to *real users*
  - **Beginners/students:** introductory logic/proof assistants course
  - **Experts:** real maths codebases

Iconic Manipulations

Bubble Calculi

The chemical metaphor

Item	$\Longleftrightarrow$	Ion
Color	$\Longleftrightarrow$	Polarity
Logical connective	$\Longleftrightarrow$	Chemical bond
Click	$\Longleftrightarrow$	Heating
Drag-and-Drop	$\Longleftrightarrow$	Bimolecular reaction

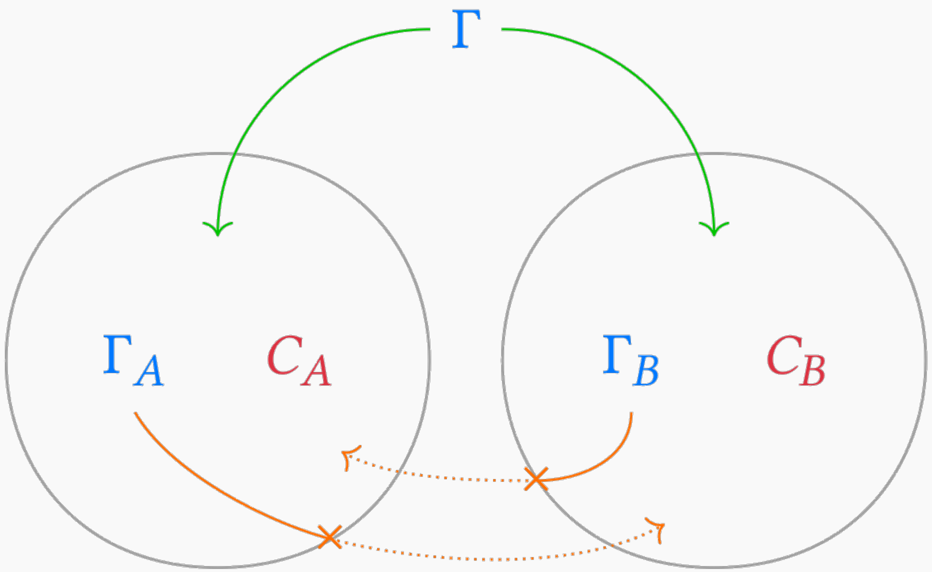
The chemical metaphor

Item	$\Longleftrightarrow$	Ion
Color	$\Longleftrightarrow$	Polarity
Logical connective	$\Longleftrightarrow$	Chemical bond
Click	$\Longleftrightarrow$	Heating
Drag-and-Drop	$\Longleftrightarrow$	Bimolecular reaction

Breaks on rules that create subgoals (e.g. click on $\wedge$)

Bubbles

Natural way to depict context scoping



Two main inspirations:

- The **chemical abstract machine** (Berry and Boudol 1989)
- **Nested sequents** (Brünnler 2009)

Example proof

$$(A \vee B \Rightarrow C) \Rightarrow (A \Rightarrow C) \wedge (B \Rightarrow C)$$

Example proof

$A \vee B \Rightarrow C$ $(A \Rightarrow C) \wedge (B \Rightarrow C)$

Example proof

$A \vee B \Rightarrow C$

$A \Rightarrow C$

$B \Rightarrow C$

Example proof

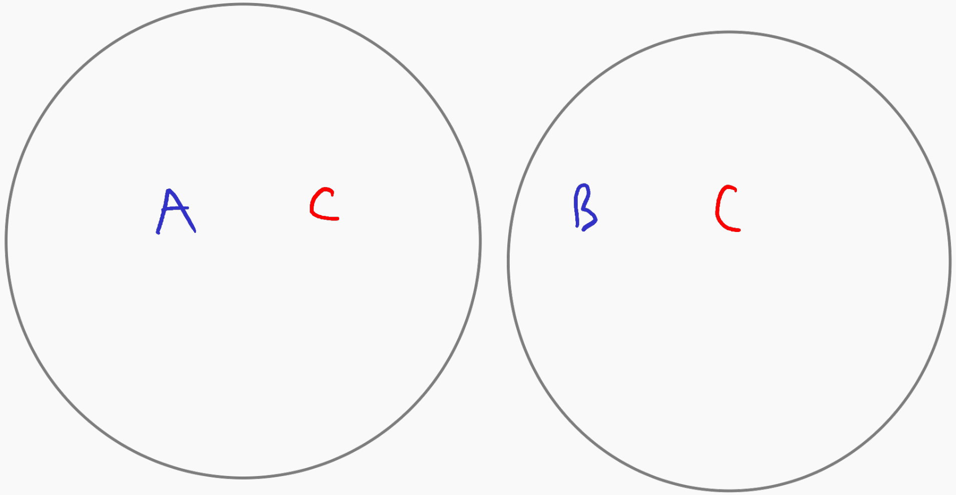
$A \lor B \Rightarrow C$

$A \quad C$

$B \Rightarrow C$

Example proof

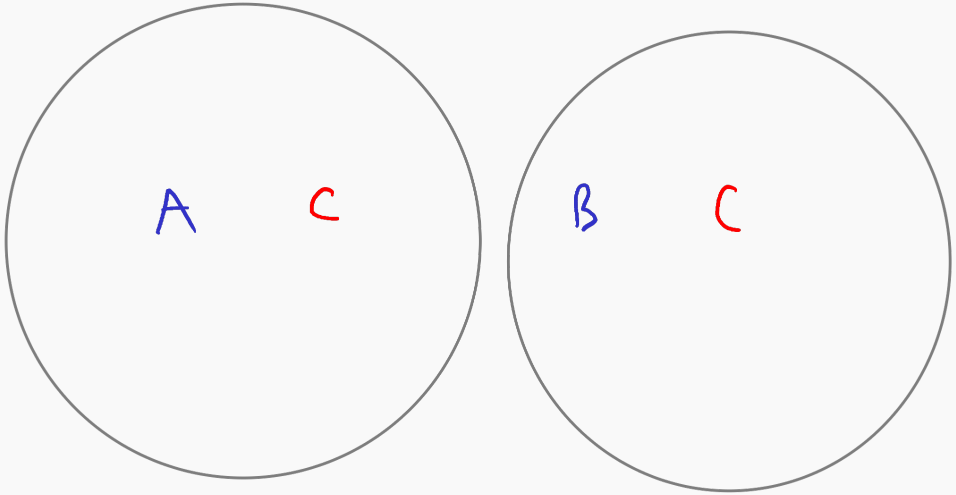
$A \vee B \Rightarrow C$



Example proof

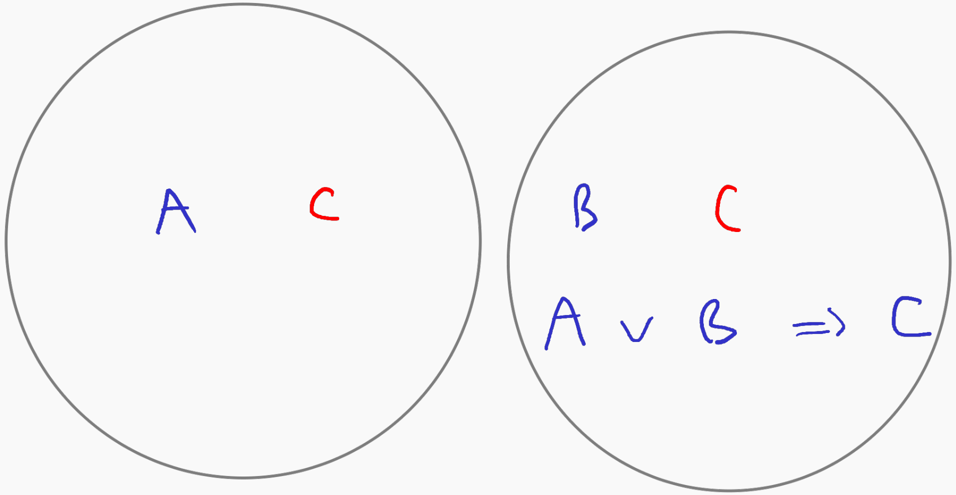
$A \vee B \Rightarrow C$

$A \vee B \Rightarrow C$

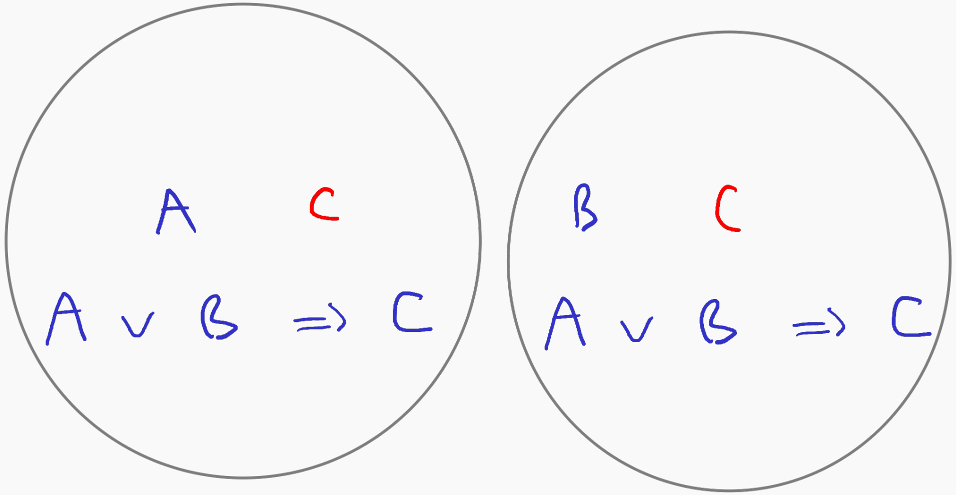


Example proof

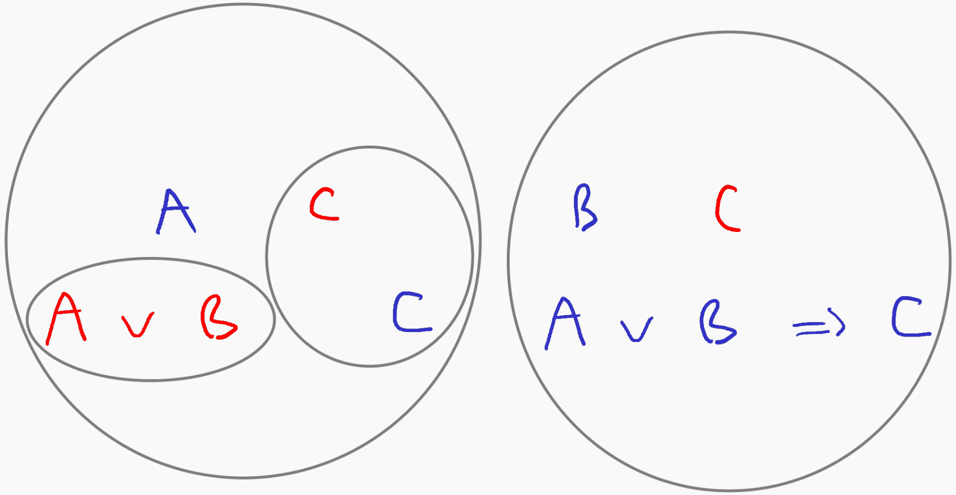
$A \vee B \Rightarrow C$



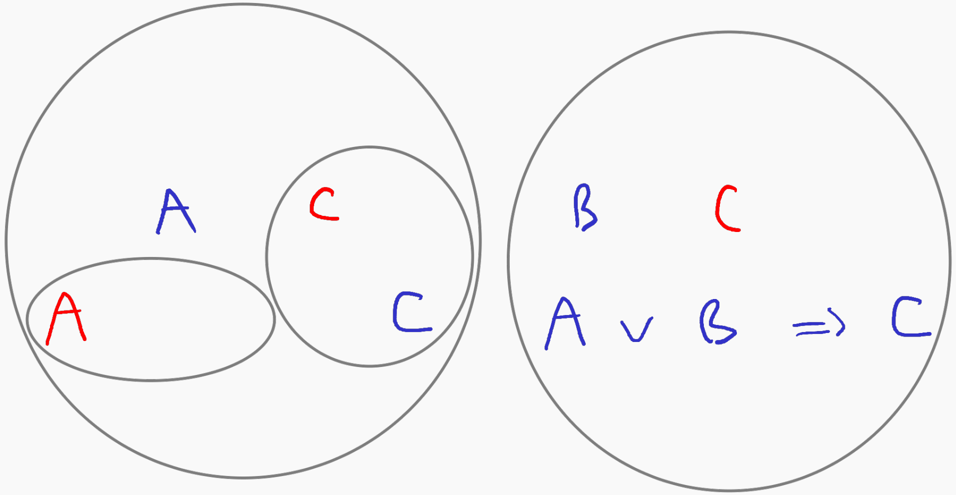
Example proof



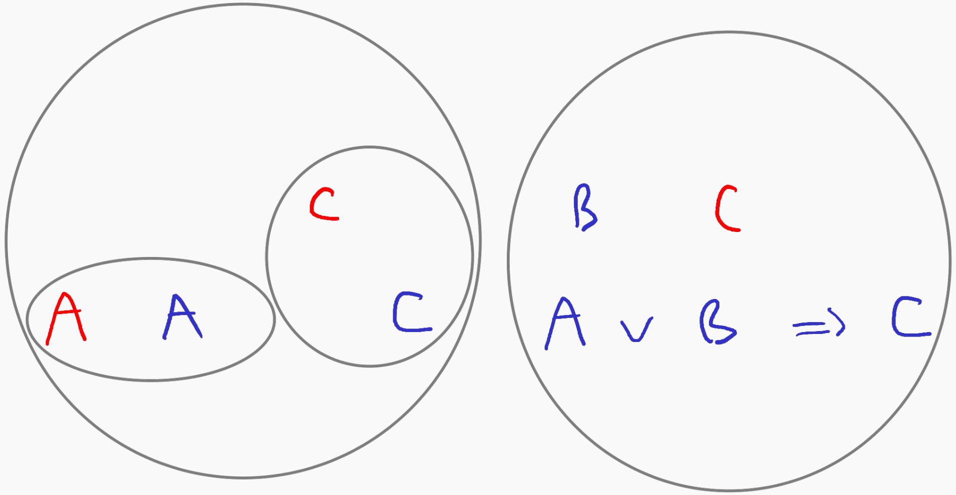
Example proof



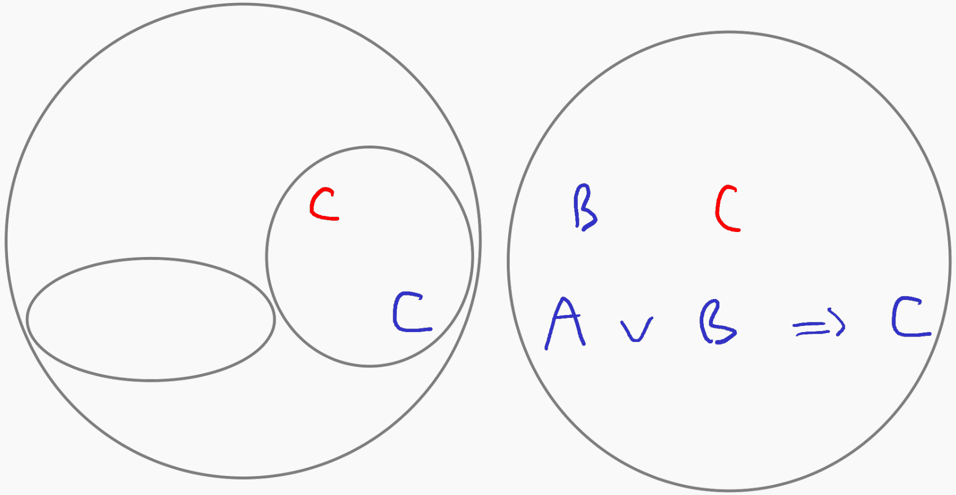
Example proof



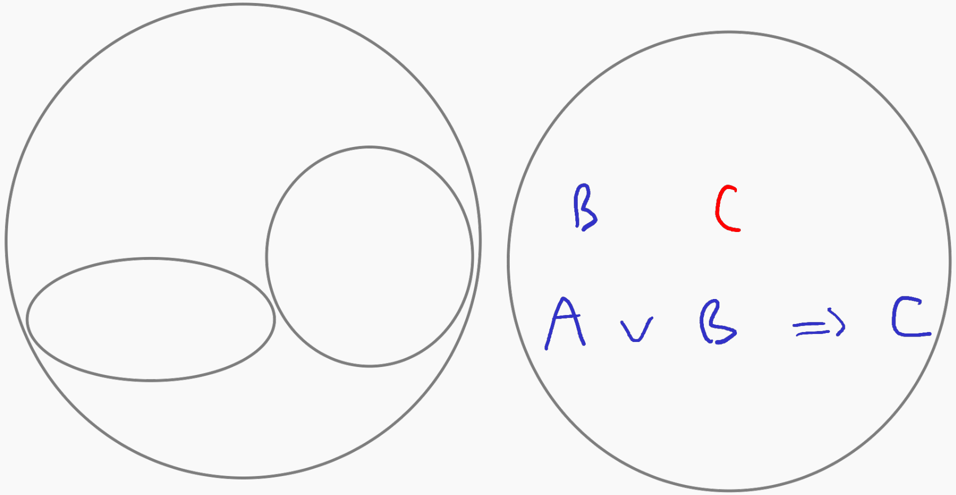
Example proof



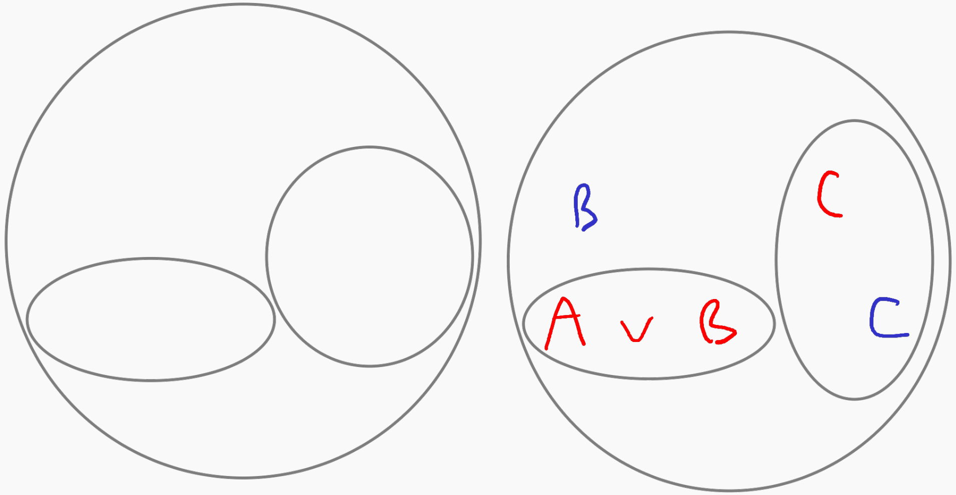
Example proof



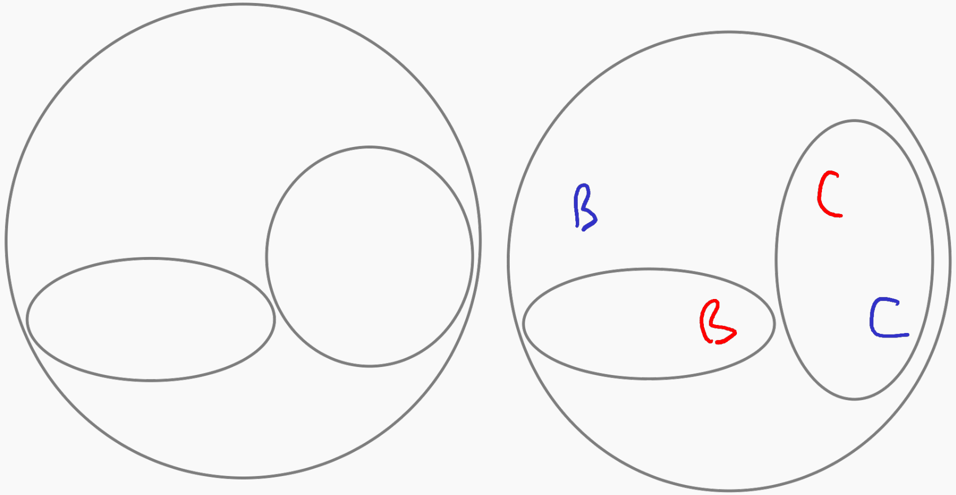
Example proof



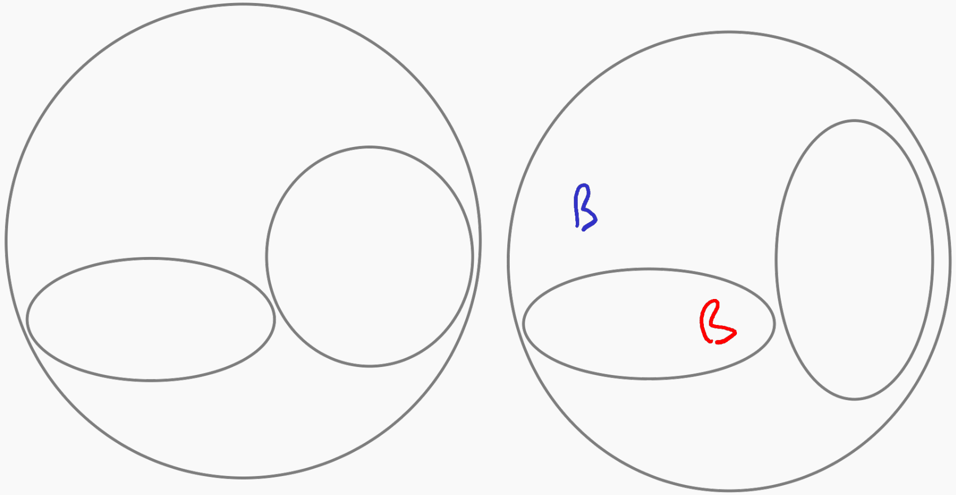
Example proof



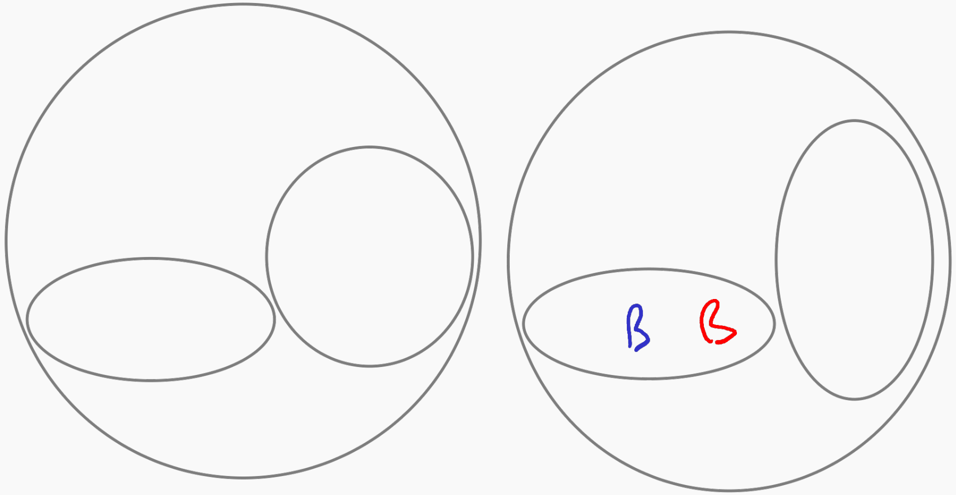
Example proof



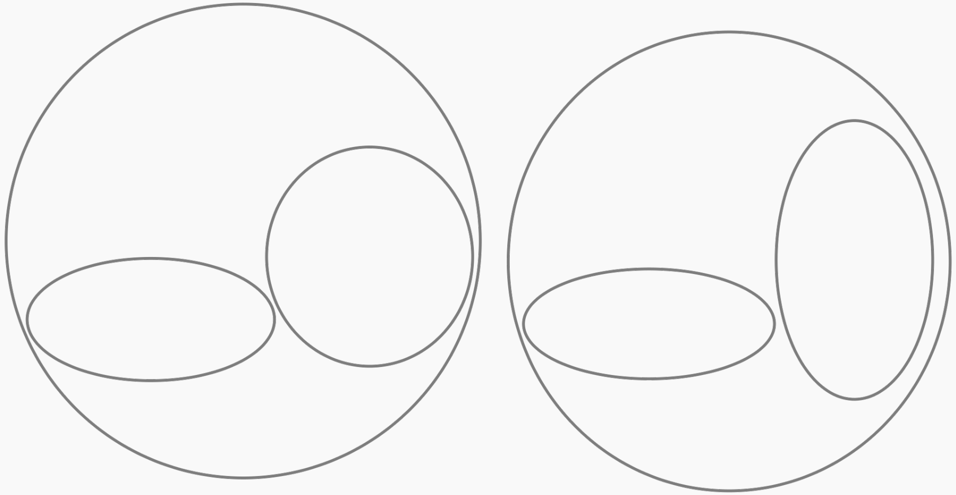
Example proof



Example proof

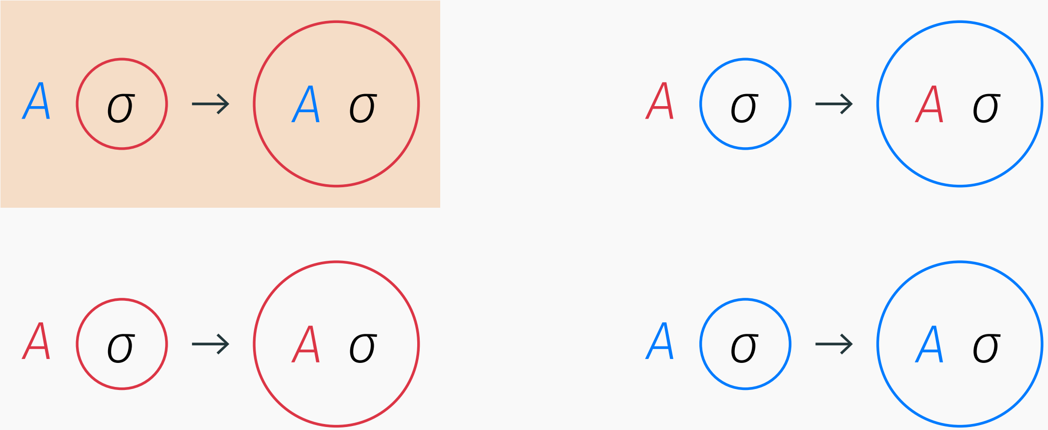


Example proof



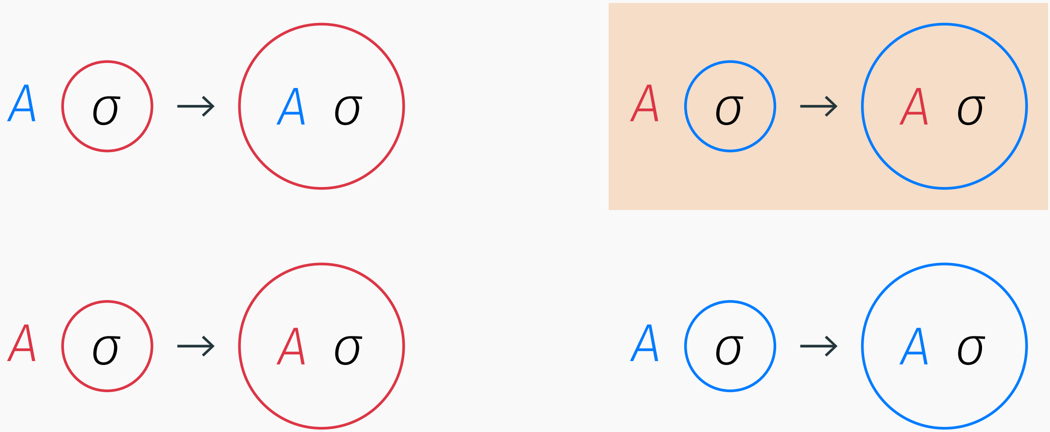
Example proof

Polarized bubbles



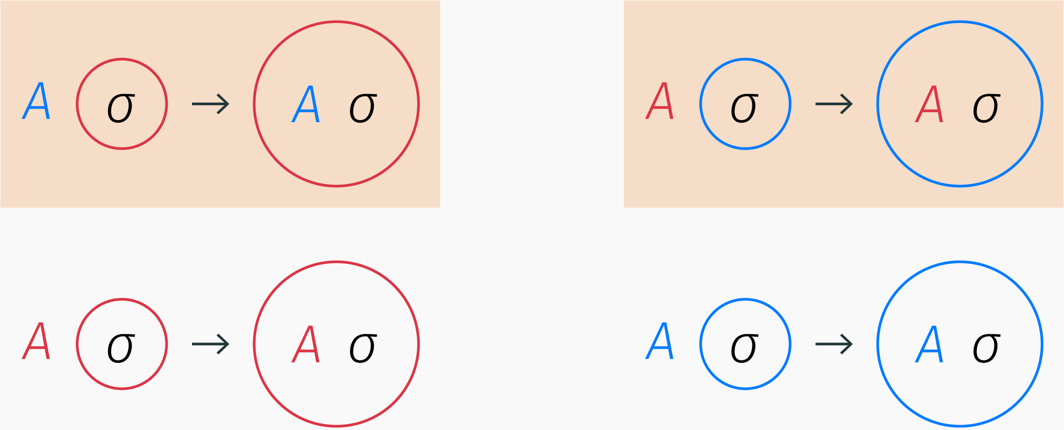
Intuitionistic logic

Polarized bubbles



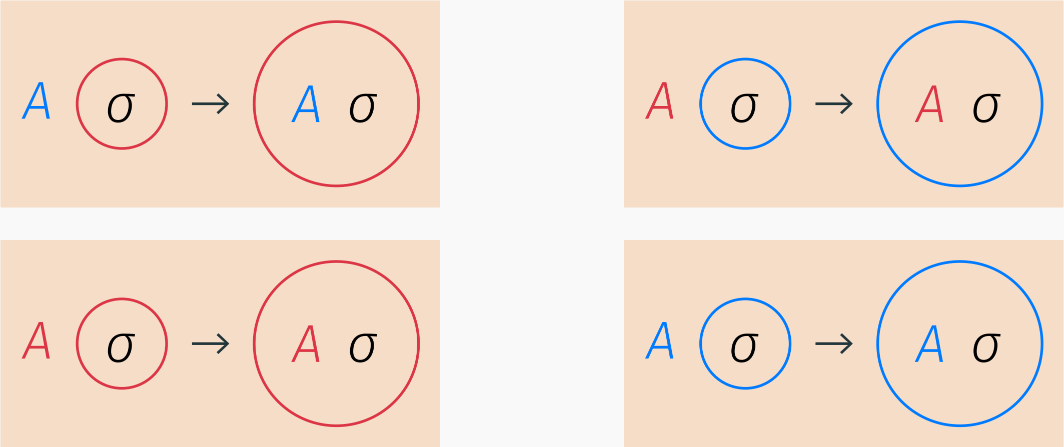
Dual-intuitionistic logic

Polarized bubbles



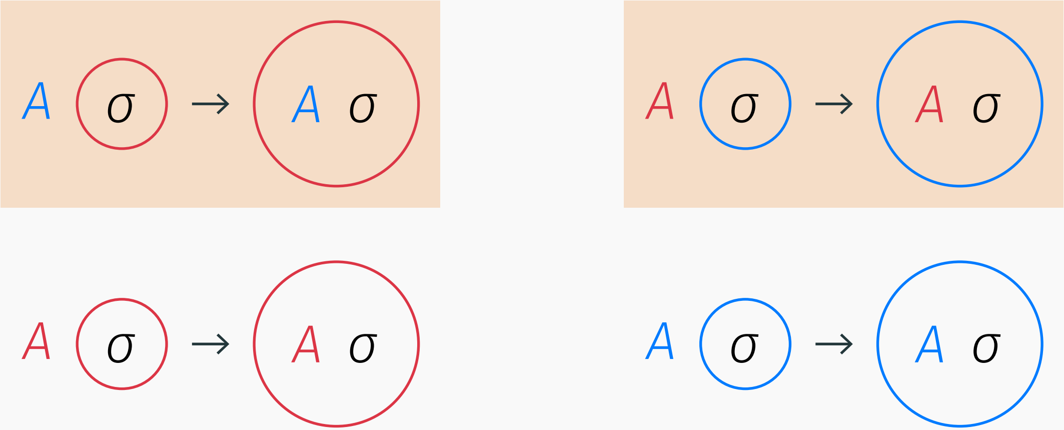
Bi-intuitionistic logic

Polarized bubbles



Classical logic

Polarized bubbles



Intuitionism = same polarities repel eachother

Flower Calculus

Polarity meets Space

Bubble calculi are not **fully iconic** (need for *symbolic* connectives)

Polarity meets Space

Bubble calculi are not **fully iconic** (need for *symbolic* connectives)

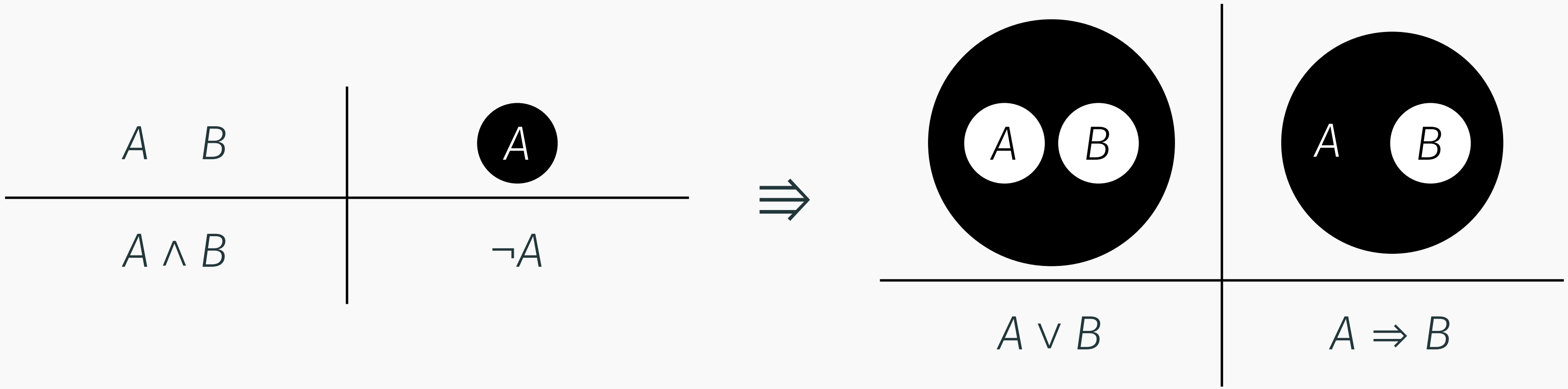
Key idea: space is *polarized*, not **objects**

- **Diagrammatic** proof system invented by C. S. Peirce around 1890
- Topological representation of **negation** as nested “*cuts*” (Jordan curves):

$A \quad B$	A
$A \wedge B$	$\neg A$

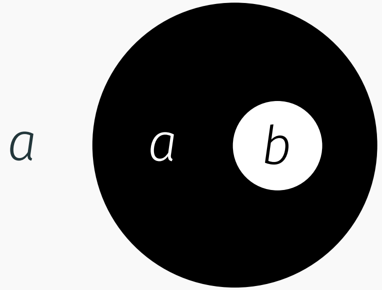
Existential Graphs (Peirce 1906)

- **Diagrammatic** proof system invented by C. S. Peirce around 1890
- Topological representation of **negation** as nested “*cuts*” (Jordan curves):



Illative transformations

Inference rules on locations

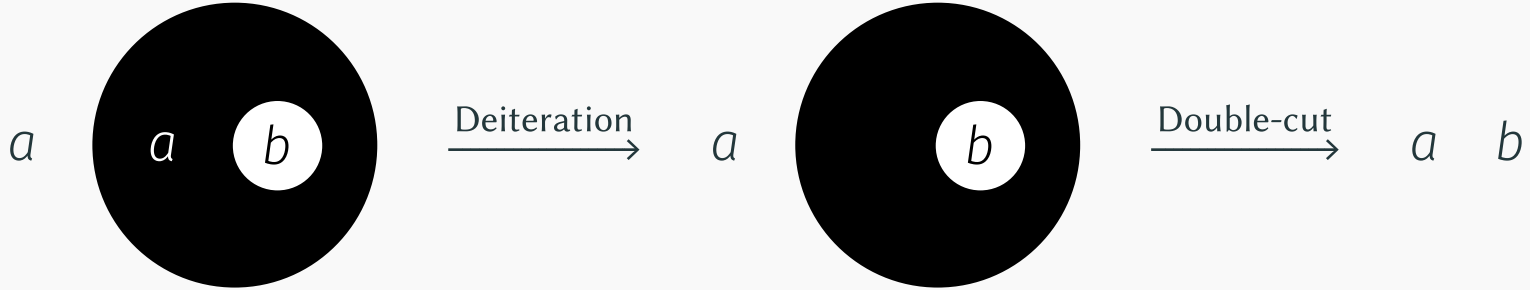


Illative transformations

Inference rules on locations

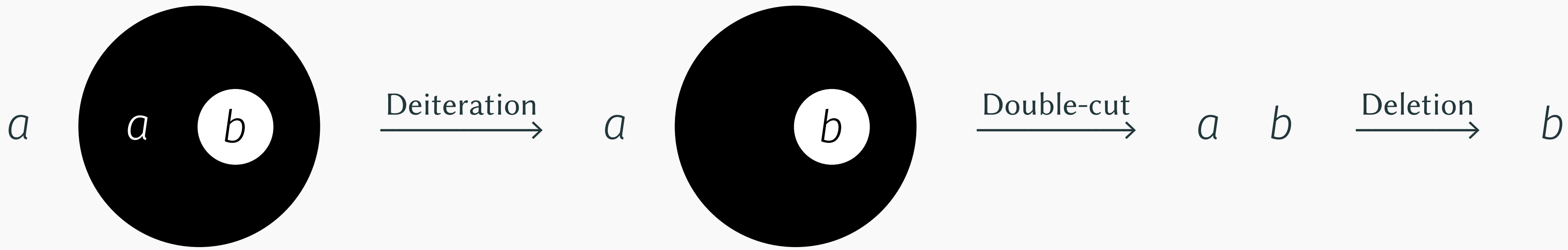
Illative transformations

Inference rules on locations



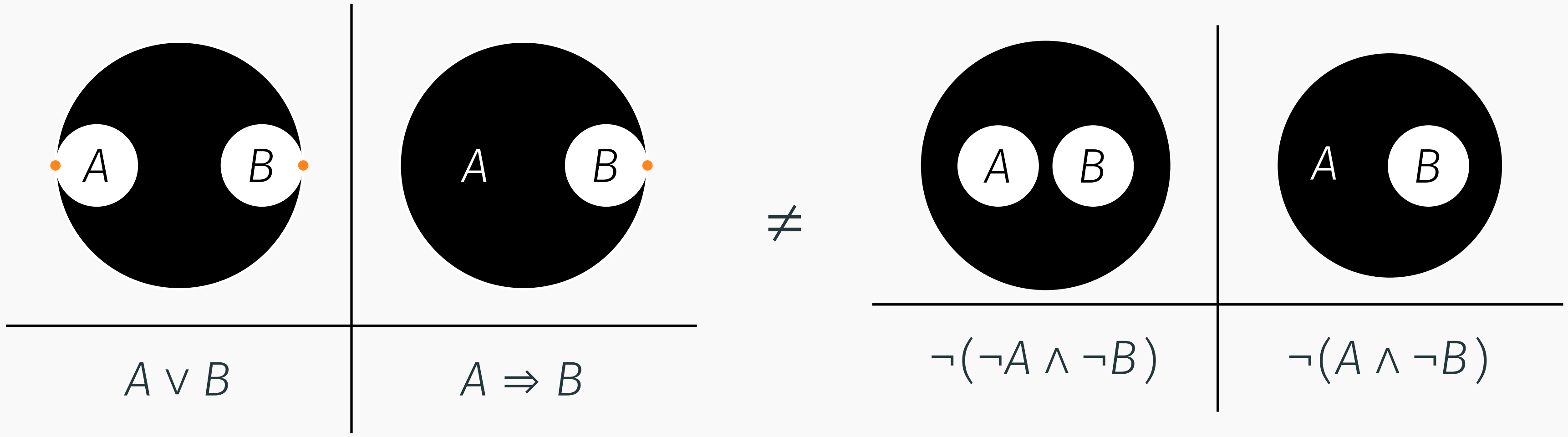
Illative transformations

Inference rules on locations

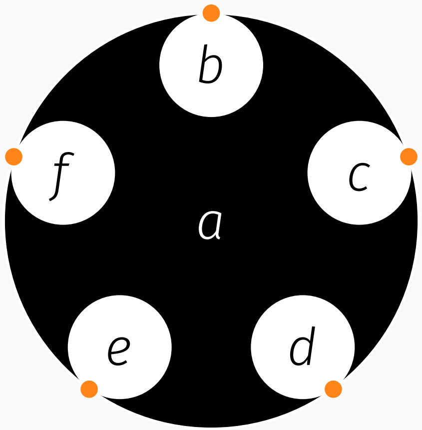


Intuitionistic Existential Graphs (Oostra 2011)

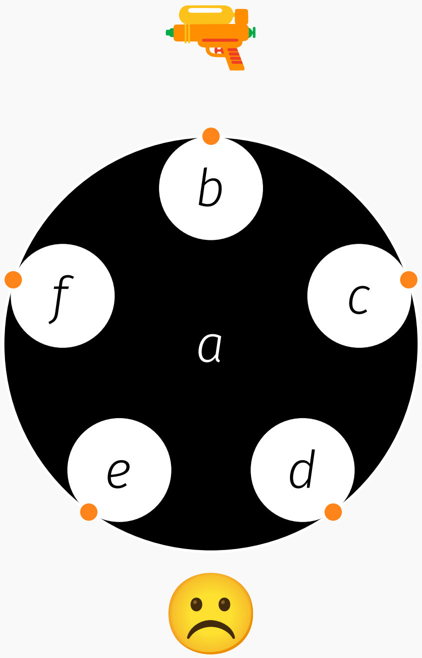
- Topological representation of **implication** with Peirce's "*scroll*"
- Scroll = continuously joined nested cuts:



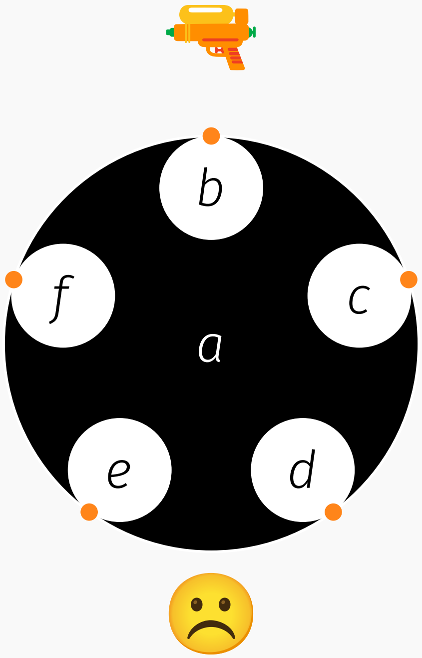
Blooming

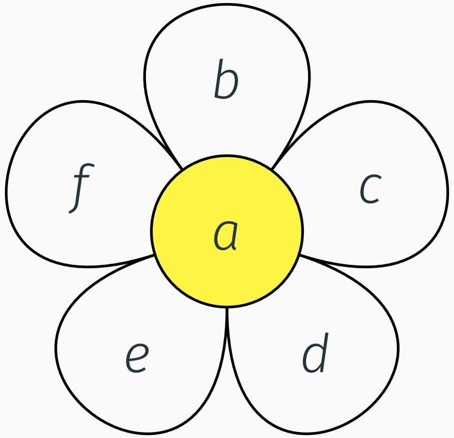


Blooming



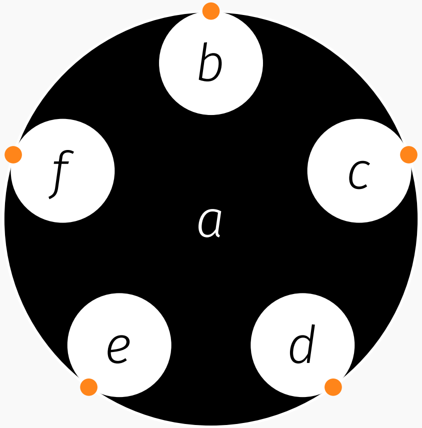
Blooming

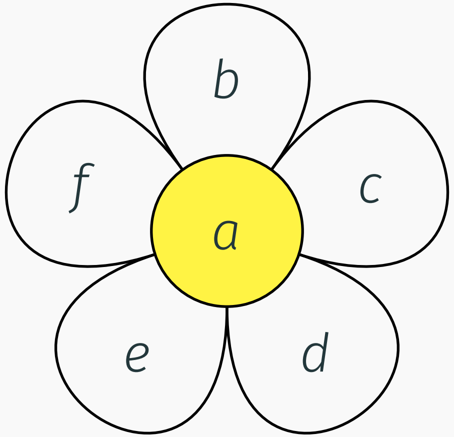




Turn **inloops** into petals.

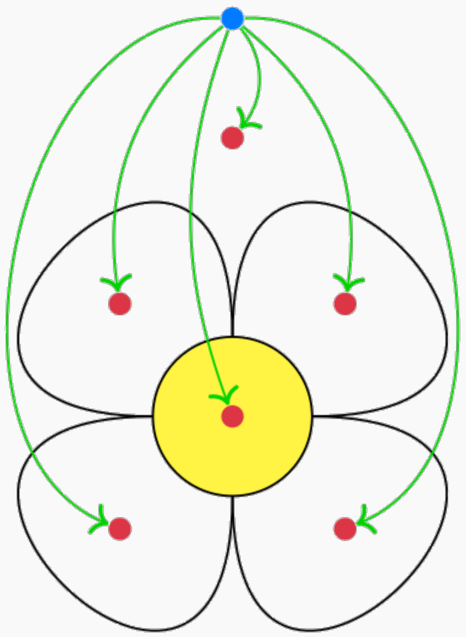
Blooming



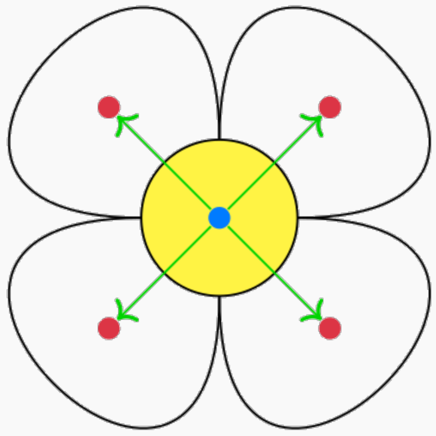


"Make love, not war"

Pollination



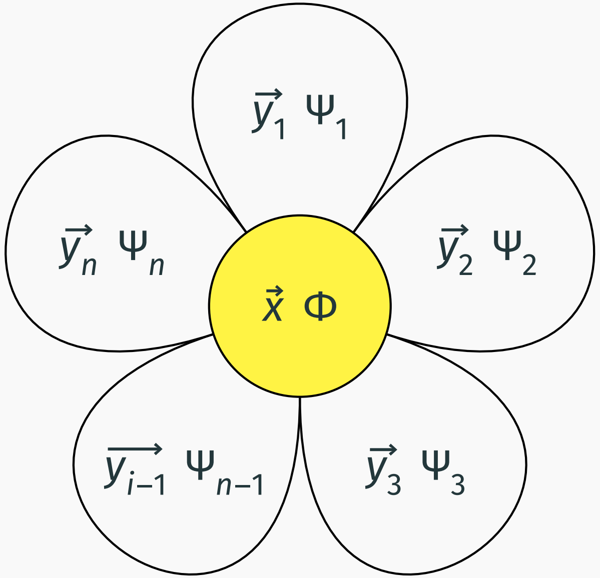
Cross-pollination



Self-pollination

- Support for **quantification** with *binders* $\vec{x}$
- Interpretation as geometric formulas from *topos theory*
- Inference rules divided in two fragments:
  - **Nature** ❀ = analytic and invertible
  - **Culture** ✂ = non-invertible

Theorem (Analytic completeness): If a flower is *valid* (i.e. true in every Kripke model), then it is ❀-provable.



$$\forall \vec{x}. \left(\bigwedge \Phi \Rightarrow \bigvee_i \exists \vec{y}_i . \Psi_i \right)$$

Flower Prover

(mascarpone cream)

GUI in the Proof-by-Action paradigm based on the flower calculus

- Represent flowers as nested **boxes**
- **Modal interface** to interpret gestural actions:

Proof mode $\Longleftrightarrow$ Natural (invertible and analytic) rules

Edit mode $\Longleftrightarrow$ Cultural (non-invertible) rules

Navigation mode $\Longleftrightarrow$ Contextual closure (functoriality)

Thank you!

Bibliography

Berry, Gerard, and Gerard Boudol. 1989. "The Chemical Abstract Machine". In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 81–94. POPL '90. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/96709.96717

Brünnler, Kai. 2009. "Deep Sequent Systems for Modal Logic". *Archive for Mathematical Logic* 48 (6): 551–77. https://doi.org/10.1007/s00153-009-0137-3

Chaudhuri, Kaustuv. 2013. "Subformula Linking as an Interaction Method". Edited by Sandrine Blazy, Christine Paulin-Mohring, David Pichardie, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, et al.. *Interactive Theorem Proving*. Berlin,

Bibliography

Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-39634-2_28

Donato, Pablo. 2024. "The Flower Calculus". https://hal.science/hal-04472717/

Donato, Pablo, Pierre-Yves Strub, and Benjamin Werner. 2022. "A Drag-and-Drop Proof Tactic". In *Proceedings of the 11th ACM SIGPLAN International Conference on Certified Programs and Proofs*, 197–209. CPP 2022. Philadelphia, PA, USA: Association for Computing Machinery. https://doi.org/10.1145/3497775.3503692

Guglielmi, Alessio. 1999. "A Calculus of Order and Interaction". https://www.researchgate.net/publication/2807151_A_Calculus_of_Order_and_Interaction

Bibliography

Oostra, Arnold. 2011. *Gráficos Existenciales Beta Intuicionistas*. Cuadernos De Sistemática Peirceana. Centro de Sistemática Peirceana

Peirce, Charles Sanders. 1906. "Prolegomena to an Apology for Pragmaticism". *The Monist* 16 (4): 492–546. https://www.jstor.org/stable/27899680