From Proof Nets to Combinatorial Proofs - A New Approach to Hilbert’s 24th Problem
Willem Heijltjes, Lutz Strassburger

To cite this version:
Willem Heijltjes, Lutz Strassburger. From Proof Nets to Combinatorial Proofs - A New Approach to Hilbert’s 24th Problem. École thématique. From Proof Nets to Combinatorial Proofs - A New Approach to Hilbert’s 24th Problem, Utrecht / Virtual, Netherlands. 2021. hal-03316571

HAL Id: hal-03316571
https://inria.hal.science/hal-03316571
Submitted on 6 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Overview

1. The problem of proof identity
2. MLL: sequent proofs and proof nets
3. Cographs and handsome proof nets
4. Deep inference
5. ALL: sequent proofs, proof nets, and deep inference
6. Fibrations and skew fibrations
7. Classical propositional combinatorial proofs
8. First-order combinatorial proofs
9. Intuitionistic combinatorial proofs
10. Normalization

5 x 90 min \(\Rightarrow\) 10 x 45 min

MLL = multiplicative linear logic
ALL = additive linear logic
What is Proof Theory?

- Group theory = theory of groups
  - well-established definition of group
  - two groups are the same if they are isomorphic

- Graph theory = theory of graphs
  - well-established definition of graph
  - two graphs are the same if they are isomorphic

- Proof theory = theory of (formal) proofs ???
  - no well-established definition of formal proof
  - no idea when two proofs are the same

At the current state of the art, Proof theory is not the theory of proofs but the theory of proof systems.

All important results in proof theory are about proof systems:
- soundness
- completeness
- cut elimination
- focusing
- p-equivalence

Can we make proof theory a theory of proofs?

1. What is a proof?
   ⇒ define proofs independently from the proof systems

2. When are two proofs the same?
   ⇒ define a notion of proof identity
Hilbert's 24th problem

As 24th problem in my Paris lecture, I wanted to ask the question: Find criteria of simplicity or rather prove the greatest simplicity of given proofs. More generally develop a theory of proof methods in mathematics. Under given conditions there can be only one simplest proof. And if one has 2 proofs for a given theorem, then one must not rest before one has reduced one to the other or discovered which different premises (and auxiliary means) have been used in the proofs: When one has two routes then one must not just go these routes or find new routes, but the whole area lying between these two routes must be investigated . . .

Hilbert was thinking about adding the problem of proof identity as 24th problem to his famous lecture with the famous 23 problems that was held in 1900. But proof theory as a field was only established in 1928 with the appearance of the Book "Grundzüge der theoretischen Logik" by Hilbert and Ackermann. So, the problem of proof identity is older than proof theory itself.

Sources:
- Notebook of Hilbert: David Hilbert, Mathematische Notizbücher, Niedersächsische Staats- und Universitätbibliothek, Cod. Ms. D. Hilbert 600:3, S.25 (Scan from a hardcopy made by Rüdiger Thiele)
- Translation: Lutz Straßburger

See also:

When are two proofs the same?

Curry-Howard-Correspondence
- formulas = types
- proofs = programs
- normalization = computation
fondations of functional programming languages

When are two proofs (in normal form) the same?

Two proofs are the same iff they can be transformed into each other via a sequence of rule permutation steps.
- works only for sequent calculus like formalisms
- PSPACE-hard if (2) is present
- exponential blow-up of the size of the proof if (3) is present

Hilbert was thinking about adding the problem of proof identity as 24th problem to his famous lecture with the famous 23 problems that was held in 1900. But proof theory as a field was only established in 1928 with the appearance of the Book "Grundzüge der theoretischen Logik" by Hilbert and Ackermann. So, the problem of proof identity is older than proof theory itself.

Sources:
- Notebook of Hilbert: David Hilbert, Mathematische Notizbücher, Niedersächsische Staats- und Universitätbibliothek, Cod. Ms. D. Hilbert 600:3, S.25 (Scan from a hardcopy made by Rüdiger Thiele)
- Translation: Lutz Straßburger

See also:

When are two proofs the same?

Curry-Howard-Correspondence
- formulas = types
- proofs = programs
- normalization = computation
fondations of functional programming languages

When are two proofs (in normal form) the same?

Two proofs are the same iff they can be transformed into each other via a sequence of rule permutation steps.
- works only for sequent calculus like formalisms
- PSPACE-hard if (2) is present
- exponential blow-up of the size of the proof if (3) is present
When are two proofs (in normal form) the same?

A ∧ (A → B) ∧ (¬A ∨ C) ∧ (C ∧ B → D) → D
¬(A ∧ (A → B) ∧ (¬A ∨ C) ∧ (C ∧ B → D))
D
¬A
¬(A → B)
¬(¬A ∨ C)
¬(C ∧ B → D)
A
closed
¬B
A
closed
¬C
C ∧ B
B
closed
C
closed
¬D
closed

This is a semantic tableau, a natural deduction proof, and a Coq script, all proving the same formula.

Are these proofs “the same”?

Combinatorial Proof Identity

The term “combinatorial proof identity” does not yet occur in the literature. We invented it for this course.

This course is about the stuff that goes into the yellow blob in the middle.

The technical details about “the yellow blob in the middle” depend on the logic. In every logic behaves differently when it comes to the structure of its proofs. This means that the answer to the question of when two proofs are the same might be different for every logic. In this course we look into the following five logics:

- classical propositional logic
- classical first-order logic
- intuitionistic propositional logic
- multiplicative linear logic
- additive linear logic
Classical Propositional Logic (CPL)

- **Formulas:** $A ::= a \mid \overline{a} \mid A \land B \mid A \lor B$
- **Negation:** $\overline{a} = a \quad A\land B = \overline{A} \lor \overline{B} \quad \overline{A} \lor B = \overline{A} \land \overline{B}$
- **Implication:** $A \Rightarrow B = \overline{A} \lor B$
- **Sequents:** $\Gamma ::= A_1, A_2, \ldots, A_k$
- **Inference rules (sequent calculus):**
  
  - $\text{ax} \quad \frac{\vdash a, \overline{a}}{}$
  - $\text{cont} \quad \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}$
  - $\text{weak} \quad \frac{\vdash \Gamma}{\vdash \Gamma, A}$
  - $\forall \quad \frac{\vdash \Gamma, A[x/t]}{\vdash \Gamma, \exists x.A}$
  - $\forall \quad \frac{\forall \vdash \Gamma, A}{\vdash \Gamma, \forall x.A}$
  - $\exists \quad \frac{\exists \vdash \Gamma, A[x/t]}{\vdash \Gamma, \exists x.A}$
  - $\overline{a} = a \quad \overline{p(t_1, \ldots, t_m)} = p(t_1, \ldots, t_m)$
  - $\overline{A} \land B = \overline{A} \lor \overline{B}$
  - $\overline{A} \lor B = \overline{A} \land \overline{B}$
  - $\overline{A} 

First-Order Classical Logic (FOL) — Formulas

- **Terms:** $t ::= x \mid f(t_1, \ldots, t_n)$
- **Atoms:** $a ::= p(t_1, \ldots, t_m) \mid \overline{p(t_1, \ldots, t_m)}$
- **Formulas:** $A ::= a \mid A \land A \mid A \lor A \mid \exists x.A \mid \forall x.A$
- **Sequents:** $\Gamma ::= A_1, A_2, \ldots, A_k$
- **Negation:** $\overline{a} = a \quad \overline{p(t_1, \ldots, t_m)} = p(t_1, \ldots, t_m)$
  - $\overline{A} \land B = \overline{A} \lor \overline{B}$
  - $\overline{A} \lor B = \overline{A} \land \overline{B}$
  - $\overline{A} \land B = \overline{A} \lor \overline{B}$
  - $\overline{A} \lor B = \overline{A} \land \overline{B}$

First-Order Classical Logic (FOL) — Sequent Calculus

- **Inference rules (sequent calculus):**
  - $\text{ax} \quad \frac{\vdash a, \overline{a}}{}$
  - $\forall \quad \frac{\forall \vdash \Gamma, A}{\vdash \Gamma, \forall x.A}$
  - $\exists \quad \frac{\exists \vdash \Gamma, A[x/t]}{\vdash \Gamma, \exists x.A}$
  - $\overline{a} = a \quad \overline{p(t_1, \ldots, t_m)} = p(t_1, \ldots, t_m)$
  - $\overline{A} \land B = \overline{A} \lor \overline{B}$
  - $\overline{A} \lor B = \overline{A} \land \overline{B}$
  - $\overline{A} \land B = \overline{A} \lor \overline{B}$
  - $\overline{A} \lor B = \overline{A} \land \overline{B}$

- **Exercise 1.1:** Prove Pierce’s law $((0 \Rightarrow 0) \Rightarrow 0) \Rightarrow 0$ in this sequent calculus.

- **Exercise 1.2:** Prove the drinker’s formula $\exists x (px \Rightarrow \forall y py)$ in the sequent calculus.
Additive vs. Multiplicative

multiplicative
\[
\vdash \Gamma, A, B \\
\vdash \Gamma, A \lor B \\
\downarrow \\
\otimes \vdash \Gamma, A \otimes B \\
\land \vdash \Gamma, A \land B \\
\downarrow \\
\otimes \vdash \Gamma, A \otimes B \\
\& \vdash \Gamma, A \& B
\]

additive
\[
\vdash \Gamma, A, B \\
\vdash \Gamma, A \lor B \\
\downarrow \\
\oplus \vdash \Gamma, A \oplus B \\
\land \vdash \Gamma, A \land B \\
\downarrow \\
\oplus \vdash \Gamma, A \oplus B \\
\& \vdash \Gamma, A \& B
\]

• When contraction and weakening are in the system, additive and multiplicative formulations of the and-rule (resp. or-rule) are equivalent. But when contraction and weakening are absent, they define different connectives.

**Exercise 1.3:** Show that with contraction and weakening \( A \otimes B \) and \( A \mathbin{\#} B \) imply each other, and similarly for \( A \otimes B \) and \( A \mathbin{\#} B \). Which implications hold when only weakening is present, and which when only contraction is present?

The terminology “additive/multiplicative” is due to Girard. Same for the choice of symbols.


Multiplicative Linear Logic (MLL)

- **Formulas:** \( A ::= a \mid \bot \mid A \otimes B \mid A \mathbin{\#} B \)
- **Negation:** \( a\perp = a, (A \otimes B)\perp = A\perp \mathbin{\#} B\perp, (A \mathbin{\#} B)\perp = A\perp \otimes B\perp \)
- **Implication:** \( A \to B = A\perp \mathbin{\#} B \)
- **Sequents:** \( \Gamma ::= A_1, A_2, \ldots, A_k \)
- **Inference rules (sequent calculus):**

\[
\begin{align*}
\text{ax} & \vdash a, a\perp & \vdash \Gamma, A, B & \vdash \Gamma, A \otimes B & \vdash \Gamma, \Delta, A \otimes B \\
\end{align*}
\]

Multiplicative Additive Linear Logic (MALL)

- **Formulas:** \( A ::= a \mid \bot \mid A \otimes B \mid A \mathbin{\#} B \mid A \& B \mid A \mathbin{\#} B \)
- **Negation:** \( a\perp = a, (A \otimes B)\perp = A\perp \mathbin{\#} B\perp, (A \mathbin{\#} B)\perp = A\perp \otimes B\perp, (A \& B)\perp = A\perp \mathbin{\#} B\perp \\
\)
- **Implication:** \( A \to B = A\perp \mathbin{\#} B \)
- **Sequents:** \( \Gamma ::= A_1, A_2, \ldots, A_k \)
- **Inference rules (sequent calculus):**

\[
\begin{align*}
\text{ax} & \vdash a, a\perp & \vdash \Gamma, A, B & \vdash \Gamma, A \otimes B & \vdash \Gamma, \Delta, A \otimes B \\
\end{align*}
\]

**Exercise 1.4:** Find a formula that is provable in classical logic but not in MLL. What about the other way around?
Additive Linear Logic (ALL)

- **Formulas**: \( A := a \mid a^\perp \mid A \& B \mid A \oplus B \)
- **Negation**: \( a^\perp = a \quad (A \& B)^\perp = A^\perp \oplus B^\perp, \quad (A \oplus B)^\perp = A^\perp \& B^\perp \)
- **Sequents**: \( \Gamma ::= A_1, A_2, \ldots, A_k \)
- **Inference rules (sequent calculus)**:

\[
\begin{align*}
&\text{ax} \quad \Gamma, a \vdash a \\
&\quad \vdash \Gamma, A \quad \vdash \Gamma, B \\
&\quad \vdash \Gamma, A \& B \\
&\quad \vdash \Gamma, A \oplus B \\
&\quad \vdash \Gamma, A \& B
\end{align*}
\]

Observe that every provable sequent in ALL has exactly two formulas.

**Exercise 1.5**: Find a two-formula sequent that is provable in classical logic but not in ALL. What about the other way around?

**Exercise 1.6**: Find a two-formula sequent that is provable in MLL but not in ALL. What about the other way around?

Intuitionistic Propositional Logic (IPL)

- **Formulas**: \( A ::= a \mid A \& B \mid A \supset B \)
- **Sequents**: \( A_1, A_2, \ldots, A_k \vdash B \)
- **Inference rules (sequent calculus)**:

\[
\begin{align*}
&\text{ax} \quad a \vdash a \\
&\quad \vdash \Gamma, A, B \vdash C \\
&\quad \vdash \Gamma, A \vdash B \\
&\quad \vdash \Gamma, B \\
&\quad \vdash \Gamma, A \& B \\
&\quad \vdash \Gamma, A \supset B
\end{align*}
\]

Recall that in classical logic we can encode \( A \supset B \) as \( \neg A \lor B \). So, every formula in IPL can be seen as a formula in CPL. In the same way we have in linear logic that \( A \supset B \) is the same as \( A^\perp \otimes B \). Hence, every formula of IMLL is also a formula of MLL.

**Exercise 1.7**: Can you prove Pierce’s law \( ((a \supset b) \supset a) \supset a \) in intuitionistic logic?

**Exercise 1.8**: Find an IPL formula that is provable in CPL but not in IPL (or prove that such a formula does not exist).

**Exercise 1.9**: Find an IMLL formula that is provable in MLL but not in IMLL (or prove that such a formula does not exist).
The Curry–Howard Isomorphism

- Typed λ-terms are (isomorphic to) intuitionistic natural-deduction proofs

\[
\begin{align*}
\Gamma &:\ A \\
\vdots \\
M &:\ B \\
\lambda x. M &:\ A \supset B \\
\vdots \\
N &:\ A \\
\ldots & \\
M N &:\ B
\end{align*}
\]

- This includes pairs/products/conjunction

\[
\begin{align*}
M &:\ A \\
M &:\ B \\
\langle M, N \rangle &:\ A \land B \\
\pi_1(M) &:\ A \\
\pi_2(M) &:\ B
\end{align*}
\]

Beta-reduction

- For implication:

\[
\begin{align*}
\Gamma &:\ A \\
\vdots \\
M &:\ B \\
\lambda x. M &:\ A \supset B \\
\vdots \\
N &:\ A \\
\vdots & \\
M[N/x] &:\ B
\end{align*}
\]

- For conjunction:

\[
\begin{align*}
\langle M, N \rangle &:\ A \land B \\
\pi_1(M,N) &:\ A \\
\pi_2(M,N) &:\ B \\
\vdots \\
\langle M, N \rangle &:\ A \land B \\
\pi_1(M,N) &:\ A \\
\pi_2(M,N) &:\ B \\
\vdots \\
N &:\ B
\end{align*}
\]

- (⊔/∧) – Intuitionistic natural deduction is ideal:
  - Reduction is confluent and strongly normalizing
  - Normal forms represent the meaning of a term
Sequent calculus

\[
\begin{align*}
\Gamma, x: A &\vdash M: B \\
\Gamma, x: A &\vdash M: B^w \\
\Gamma, x: A &\vdash M[x/y]: B^C
\end{align*}
\]

This is a standard annotation of sequent proofs with \(\lambda\)-terms. The implication and conjunctions are formulated with multiplicative contexts, and weakening and contraction are explicit.

- **Exercise 2.5:** Try to give an additive (context-sharing) formulation of intuitionistic sequent calculus. The annoying rule is the implication-left one (\(\supset L\)). Test it by typing the Church numeral three, \(\lambda f . \lambda x . f(f(f x))\). Conclude that the multiplicative formulation is nicer.

For the axiom and right-rules (Ax, \(\supset R\), \(\land R\)) terms and proofs are constructed correspondingly, as they are in natural deduction. But for the left-rules (\(\supset L\), \(\land L\)), the structural rules \((w, c)\), and the cut-rule, term construction and proof construction are dissimilar.

- **Exercise 2.6:** Prove that cut-free proofs construct \(\lambda\)-terms in normal form.

The sequent rule above expresses that a natural deduction proof of \(B\) with open assumptions \(A\), as below left, can be transformed into one of \(A \supset B\) where the assumptions \(A\) are closed.

All sequent rules can be interpreted as constructing natural deduction proofs, in this way, and in fact Gentzen explicitly introduced sequent calculus from this perspective in his 1934/1935 papers.

Sequent-calculus right-rules build up a natural deduction proof at its conclusion, as \(\supset R\) on the previous slide. Left-rules build up a proof at its assumptions, as \(\supset L\) here.

- **Exercise 2.7:** Complete the demonstration of how sequent rules construct natural deduction proofs for the other rules. Include also disjunction if you have given these rules previously. This makes the translation from sequent calculus to natural deduction explicit (it was implicit already in the annotation with \(\lambda\)-terms).

- **Exercise 2.8:** Give also a reverse translation, from natural deduction to sequent calculus.

- **Note:** this may duplicate or delete \(M\).

- **Possible exponential growth from sequent proof to \(\lambda\)-term**
**Permutations**

- The same $\lambda$-term can be constructed in multiple ways.
- This gives rise to **permutations** in sequent proofs.

$$\Gamma \vdash M : A, \Delta, x : B, \Delta, y : C \vdash N : D$$

$$\Gamma, f : A \supset B, \Delta, y : C \vdash \lambda y. N[f M/x] : C \supset D$$

**Exercise 2.9:** Show three more permutations.

**Exercise 2.10:** Find a way to count, list, or organize all permutations without having to actually write them out fully.

---

**Intuitionistic logic summary**

**Natural deduction:**
- Object-level calculus
- Isomorphic with simply-typed $\lambda$-calculus
- Does not need permutations (for $\wedge$ and $\supset$)

**Sequent calculus:**
- Meta-level calculus
- Describes construction of natural-deduction proofs
- Needs permutations
- Direct characterization of normal forms as cut-free

*Natural deduction factors out the permutations of sequent calculus*

---

**Linear logic: the idea**

In classical logic:

$$A \supset B = \overline{A} \lor B$$

But this breaks computational interpretations.

Enter linear logic:

$$A \supset B = ?A \& B$$

- $A \& B$ is a **linear disjunction**
- $?A$ **non-linearizes** $A$
- $\overline{A}$ is an **involutive** (i.e. classical) negation

This is computational! (Or at least, not un-computational.)

---

The origins of linear logic are in its semantics of **coherence spaces**, which are an alternative by Girard to Scott’s domains as a semantics of (untyped) $\lambda$-calculus. However, what makes linear logic convincing is its simple and natural sequent calculus, the intuitive idea of linearly-used resources, and its intriguing decomposition of intuitionistic implication.

With the **linear implication** $\vdash$, the encoding is

$$A \supset B = !A \rightarrow B$$

where $!$ is the dual to $?$. 

---

Page Dimensions: 595.3x841.9

Page Number: 7/24
Linear logic

\[ \vdash A, A \quad \vdash \Gamma, A \quad \vdash \overline{A}, \Delta \quad \text{Cut} \]

**Multiplicatives:**

\[ \vdash \Gamma, A, B \quad \vdash \Gamma, A \otimes B \quad \vdash \Gamma, A \otimes B, \Delta \quad \text{⊗} \]

**Additives:**

\[ \vdash \Gamma, A \oplus B \quad \vdash \Gamma, A \oplus B \quad \vdash \Gamma, A \oplus B \quad \Delta \quad \text{⊕} \]

**Exponentials:**

\[ \vdash ?\Gamma, A \quad \vdash \Gamma, ?A \quad \vdash \Gamma, ?A, ?A \quad ? \]

The connectives are called: *par* \(\&\), *tensor* \(\otimes\), *plus* \(\oplus\), with \(\&\), *bang* \(!\), and why not \(?\).

The *modality* \(?\) governs whether formulas can be weakened and contracted or not. The rule \(!\) for the *bang* requires that the context \(?\Gamma = ?A_1, \ldots, ?A_n\) contains only formulae that can be contracted. This is necessary for cut-elimination to work.

**Exercise 2.11:** Investigate this by giving a cut-reduction step for:

\[ \vdash ?\Gamma, ?A, ?A \quad \vdash ?A, ?A \quad \text{Cut} \]

Is the \(?\) on the context \(?\Gamma\) necessary here?

The dream of the 90s

New grounds in typed functional programming:

- Linearity gives *destructive updates* (i.e. mutable state)
- Multiplicatives \(\otimes\) and \(\\overline{\otimes}\) give *deadlock-free concurrency*

But:

- The proof system is a *sequent calculus*!
- What about *natural deduction*?

Proof nets

Idea: remove the contexts \(\Gamma, \Delta\) from the rules.

\[ \vdash A, A \quad \vdash \Gamma, A \quad \vdash A, \overline{A}, \Delta \quad \text{Ax} \]

\[ \vdash \Gamma, A, B \quad \vdash \Gamma, A \otimes B \quad \vdash \Gamma, A \otimes B \quad \Delta \quad \text{⊗} \]

\[ \vdash \Gamma, A \oplus B \quad \vdash \Gamma, A \oplus B \quad \vdash \Gamma, A \oplus B \quad \Delta \quad \text{⊕} \]

\[ \vdash \Gamma, A \quad \vdash \Gamma, \Delta \quad \text{Cut} \]

- inferences become *nodes*, called *links*,
- in a *graph* called a *proof structure*
Example

\[ \frac{\vdash b, b^{Ax}}{\vdash a, a^{Ax} \quad \vdash c, c^{Ax}} \]\n
\[ \frac{\vdash b, b \otimes c, c^{Ax}}{\vdash a, a \otimes b, b \otimes c, c^{Ax}} \]\n
Two perspectives

Graph–of–rules:
- Nodes/links represent inferences
- Natural deduction–like
- Focus: computation

\[ \begin{array}{ccc} a & \rightarrow & b \\
\rightarrow & a & b \rightarrow \\
\rightarrow & \rightarrow & c \rightarrow \end{array} \]

Sequent + axioms:
- Nodes represent connectives
- String diagram–like
- Focus: canonicity

\[ \begin{array}{ccc} a & \rightarrow & b \\
\rightarrow & \rightarrow & c \rightarrow \end{array} \]

Correctness

Which proof structures come from proofs?

Correct: \[ \begin{array}{ccc} a & \rightarrow & A \\
\rightarrow & \rightarrow & \rightarrow \end{array} \]

Incorrect: \[ \begin{array}{ccc} a & \rightarrow & A \\
\rightarrow & \rightarrow & \rightarrow \end{array} \]

All rules except \& create connected acyclic graphs

\[ \begin{array}{ccc} \vdash A, A^{Ax} & \vdash A & \vdash A, A^{Ax} \\
\vdash \Gamma, A, \vdash A, \Delta & \vdash \Gamma, A, \vdash A, \Delta \end{array} \]

Cut

The premises of \& must already be connected
**MLL proof nets**

**Definition**

A *proof net* is a sequent \( \Gamma \) with a correct linking, where:

- A *linking* on a sequent \( \Gamma \) is a partitioning of its atoms into dual, unordered pairs.
- A *switching* of \( \Gamma \) is a choice of left/right for each \( \& \)
- A *switching graph* for a linking and a switching on \( \Gamma \) is the undirected graph where nodes are connectives and atoms of \( \Gamma \) and edges connect:
  - each \( \otimes \) to both children
  - each \( \& \) to the child indicated by the switching
  - each pair of atoms in the linking
- A linking for \( \Gamma \) is **correct** if each switching graph is acyclic and connected.

---

**Example**

![Example Diagram](image)

---

**Contractibility**

1. Start from an unlabelled graph with paired \( \& \)-edges
2. Contract by:

![Contract Diagram](image)

3. Correct \( \iff \) contracts to a single point

[Implemented in linear time via union-find](image)
Sequentialization: proof nets $\mapsto$ sequent proofs

De-sequentialization: sequent proofs $\mapsto$ proof nets

---

**Theorem**

Sequentialization and de-sequentialization are inverses (up to permutations).

---

**Theorem**

A linking is correct if and only if it contracts, if and only if it sequentializes.

---

**Theorem**

Two sequent proofs are equivalent under permutations if and only if they translate to the same proof net.
Bibliography


3. Lecture
Cographs and Handsome Proof Nets

Willem Heijltjes and Lutz Straßburger

Graphs (directed and undirected)

- An (undirected) graph is a pair $G = \langle V_G, E_G \rangle$ of a (finite) set $V_G$ of vertices and a set $E_G$ of edges which are two-element subsets of $V_G$.

  Example: \[ \bullet \quad \bullet \]

  For $x, y \in V_G$, we write $xy \in E_G$ for $\{x, y\} \in E_G$.

- A directed graph is a pair $G = \langle V_G, E_G \rangle$ of a (finite) set $V_G$ of vertices and a set $E_G \subseteq V_G \times V_G$ of edges which are two-element subsets of $V_G$.

  Example: \[ \bullet \quad \bullet \quad \bullet \]

  For $x, y \in V_G$, we write $x \rightarrow_G y$ for $(x, y) \in E_G$.

Examples

<table>
<thead>
<tr>
<th>undirected graph</th>
<th>directed graph</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="wzuv" alt="Graph" /></td>
<td><img src="wzuv" alt="Graph" /></td>
</tr>
<tr>
<td>$V = {u, v, w, z}$</td>
<td>$V = {u, v, w}$</td>
</tr>
<tr>
<td>$E = {uw, uv, wv, zv}$</td>
<td>$w \rightarrow u, u \rightarrow v, v \rightarrow u, v \rightarrow w, v \rightarrow z, z \rightarrow z$</td>
</tr>
</tbody>
</table>
Graph Homomorphisms and Isomorphisms

- A **graph homomorphism** \( h : G \rightarrow H \) is a map \( h : V_G \rightarrow V_H \) such that for all \( x, y \in V_G \) we have that \( xy \in E_G \) implies \( h(x)h(y) \in E_H \).
- A **graph isomorphism** \( h : G \rightarrow H \) is a bijection \( h : V_G \rightarrow V_H \) such that \( h \) and \( h^{-1} \) are both homomorphisms.
- A **directed graph homomorphism** \( h : G \rightarrow H \) is a map \( h : V_G \rightarrow V_H \) such that for all \( x, y \in V_G \) we have that \( x \to_G y \) implies \( h(x) \to_H h(y) \).
- A **directed graph isomorphism** \( h : G \rightarrow H \) is a bijection \( h : V_G \rightarrow V_H \) such that \( h \) and \( h^{-1} \) are both homomorphisms.

Paths and Cycles

- A **path** \( p \) in a graph \( G \) is a sequence of vertices \( x_0, x_1, ..., x_n \in V_G \) such that \( x_0x_1, x_1x_2, ..., x_{n-1}x_n \in E_G \). In this the **length** of \( p \) is \( n \). The path is **elementary** if all \( x_0, ..., x_n \) are pairwise distinct. A path is a **cycle** if \( x_0 = x_n \). The cycle is **elementary** if all \( x_1, ..., x_n \) are pairwise distinct (i.e., all vertices, except for \( x_0 = x_n \)).

- Similarly for directed graphs.

Subgraphs and Induced Subgraphs

- Let \( H = (V_H, E_H) \) and \( G = (V_G, E_G) \) be graphs. We say that \( H \) is a **subgraph** of \( G \) if \( V_H \subseteq V_G \) and \( E_H \subseteq E_G \).
- We say that \( H \) is an **induced subgraph** of \( G \) if additionally \( \forall u, v \in V_H. \ uv \in E_H \iff uv \in E_G \).
- We say that \( G \) contains \( H \) as induced subgraph if there is an injective homomorphism \( f : H \rightarrow G \) such that \( f(H) \) is an induced subgraph of \( G \). If this is not the case, we say that \( G \) is \( H \)-**free**.
- Example: \( \begin{array}{c} \vdots \end{array} \) is a subgraph of \( \begin{array}{c} \vdots \end{array} \) (but not an induced subgraph).

- **Homomorphisms** are structure preserving maps. In the case of graphs, they are maps that preserve the graph structure.
- An **isomorphism** makes the two structures “indistinguishable”.

- The length of the path is its number of edges.
- The shortest possible cycle is an undirected graph has length 3. In a directed graph, the shortest possible cycle has length 1.
- In the rest of the course, all paths and cycles are elementary, but we won’t say it explicitly all the time.
- **Exercise 3.1:** Give the definition of (elementary) path and cycle in directed graphs.

- **Exercise 3.2:** Give the definition of subgraph and induced subgraph for directed graphs.
- For the moment, we only consider undirected graphs. Directed graphs return in Lectures 8, 9, and 10.
- **Exercise 3.3:** We call \( P_4 \) the graph \( \begin{array}{c} \vdots \end{array} \) : Which of the following 9 graphs is \( P_4 \)-free?
Operations on Graphs

- **Complement:**
  \[ \overline{G} = \langle V_G, \{xy \mid x \neq y \text{ and } xy \notin E_G\} \rangle \]

- **Disjoint Union:**
  \[ G + H = \langle V_G \uplus V_H, E_G \uplus E_H \rangle \]

- **Join:**
  \[ G \times H = \langle V_G \uplus V_H, E_G \uplus E_H \uplus \{xy \mid x \in V_G \text{ and } y \in V_H\} \rangle \]

**Exercise 3.4:** Show that \( G \times H = \overline{G} + \overline{H} \) and \( \overline{\overline{G}} = G \)

Cographs

- A **cograph** is a graph that can be constructed from single vertex graphs using the operations + and ×.
- A **cotree** is the term tree constructing the cograph.

**Example:**

\[
\text{cograph: } \begin{array}{c}
  c \\
  \downarrow \\
  d \\
\end{array} \quad \times \quad \begin{array}{c}
  c \\
  \downarrow \\
  d \\
\end{array} \quad + \quad \begin{array}{c}
  a \\
  \downarrow \\
  b \\
\end{array}
\]

**Theorem:** A graph is a cograph if and only if it is \( P_4 \)-free.

(Where \( P_4 \) is the graph: \( \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \))

Graphs of Formulas

**Mapping \( [\cdot] \) from MLL formulas to (labelled) graphs:**

- \([a] = \bullet a\) (single vertex labelled \( a \))
- \([a^+] = \bullet a^+\) (single vertex labelled \( a^+ \))
- \([A \otimes B] = [A] + [B]\)
- \([A \otimes B] = [A] \times [B]\)

**Example:**

\[
[ c \otimes ((a \otimes b) \otimes d)] = \begin{array}{c}
  c \\
  \bullet \\
  a \\
  \bullet \\
  b \\
  \bullet \\
  d \\
\end{array}
\]

**Equivalence of Formulas:**

- \( A \otimes B \equiv B \otimes A \)
- \( (A \otimes B) \otimes C \equiv A \otimes (B \otimes C) \)
- \( A \otimes B \equiv B \otimes A \)
- \( (A \otimes B) \otimes C \equiv A \otimes (B \otimes C) \)

**Theorem:** \([A] = [B]\) iff \( A \equiv B \)

- **Exercise 3.5:** Which of these graphs are cographs? Give the corresponding cotrees.

- **Exercise 3.6:** Draw the graphs of the following formulas:
  - \((a \otimes a^+) \otimes (b \otimes b^+)\)
  - \((a \otimes b) \otimes (a^+ \otimes b^+)\)
  - \((a \otimes (a^+ \otimes b^+)\))
  - \((a \otimes a^+) \otimes ((a \otimes a^+) \otimes (a \otimes a^+))\)

- **Exercise 3.7:** Prove the theorem.
Perfect Matchings

- A matching $M$ in a graph $G = \langle V_G, E_G \rangle$ is a subset $M \subset E_G$ of pairwise disjoint edges, i.e., no two edges in $M$ are adjacent (share a common vertex).
- A matching $M$ is perfect if for all $v \in V_G$ there is a $w \in V_G$ such that $vw \in E_G$ (i.e., every vertex is incident to a matching edge).

Example:

\[ \begin{array}{c}
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\end{array} \]

Attention: In the textbook definition of matching (the one we give here), the matching is part of the graph, i.e., every matching edge is also an edge in the graph. But in this course, we often consider cases where the matching not part of a certain graph, i.e., formally, we have two graphs with the same vertex set: one is a graph with a certain property, very often a cograph, and the other is a perfect matching, i.e., a graph in which every edge participates in the matching.

RB-graphs and RB-cographs

- An RB-graph is a triple $\mathcal{G} = \langle V_G, R_G, B_G \rangle$, where $\langle V_G, R_G \rangle$ is a graph and $B_G$ is a perfect matching in the graph $\langle V_G, B_G \rangle$.
- An RB-cograph is an RB-graph $\mathcal{G} = \langle V_G, R_G, B_G \rangle$, where $\langle V_G, R_G \rangle$ is a cograph.

Examples:

\[ \begin{array}{c}
\bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet \\
\end{array} \]

Æ-Paths and Æ-Cycles

- An alternating elementary path (or Æ-path) in an RB-graph $\mathcal{G} = \langle V_G, R_G, B_G \rangle$ is an elementary path in $\langle V_G, R_G \cup B_G \rangle$, such that the edges of the path are alternating in $R_G$ and $B_G$.
- An alternating elementary cycle (or Æ-cycle) in an RB-graph $\mathcal{G} = \langle V_G, R_G, B_G \rangle$ is an elementary cycle in $\langle V_G, R_G \cup B_G \rangle$, such that the edges of the cycle are alternating in $R_G$ and $B_G$.
- A chord in an Æ-path $v_0v_1, \ldots, v_n$ (or in an Æ-cycle $v_0v_1, \ldots, v_n$ with $v_0 = v_n$) is an edge $v_iv_j \in R_G$ (with $0 \leq i \neq j \leq n$) that is not part of the Æ-path (or Æ-cycle). An Æ-path (Æ-cycle) in an RB-graph is chordless if it has no chord.
- An RB-graph is Æ-connected if any two vertices are connected by a chordless Æ-path.
- An RB-graph is Æ-acyclic if it has no chordless Æ-cycle.

Attention: In the textbook definition of matching (the one we give here), the matching is part of the graph, i.e., every matching edge is also an edge in the graph. But in this course, we often consider cases where the matching not part of a certain graph, i.e., formally, we have two graphs with the same vertex set: one is a graph with a certain property, very often a cograph, and the other is a perfect matching, i.e., a graph in which every edge participates in the matching.

RB-graphs and RB-cographs

- An RB-graph can also be seen as a multigraph $\mathcal{G} = \langle V_G, R_G \cup B_G \rangle$ (i.e., a graph which can have more than one edge between two vertices), in which $B_G$ is a perfect matching.
- Following the work of Christian Retoré, we draw the $R$-edges of RB-graphs in red/regular and the $B$-edges in blue/bold.


Æ-Paths and Æ-Cycles

- These definitions come from Christian Retoré’s work.
- Observe that an Æ-cycle always has even length.
What does all this have to do with proof nets?

Terminology:
- \textit{prenet} = graph constructed from the sequent forest $\Gamma$ and axiom edges $\ell$; denoted as $\pi(\Gamma, \ell)$
- \textit{proof net} = a prenet that is correct, i.e., every switching is acyclic and connected

Translating MLL-Prenets into RB-graphs (Method 1)

Let a sequent $\Gamma$ and an axiom linking $\ell$ be given.

1. Translate every formula $A$ in $\Gamma$ into an RB-tree $T_{RB}(A)$:

   - $T_{RB}(A)$
   - $T_{RB}(A \otimes B)$
   - $T_{RB}(A \parallel B)$

2. For each linking edge in $\ell$ add a matching edge between the corresponding atoms

The result is an RB-cograph and called the \textit{(tree-like) RB-prenet} $\tau(\Gamma, \ell)$

Example

\textbf{Theorem:} $\pi(\Gamma, \ell)$ is correct iff $\tau(\Gamma, \ell)$ is $\omega$-acyclic and $\omega$-connected.
In that case $\tau(\Gamma, \ell)$ is called a \textit{(tree-like) RB-proof net}. 

- Recall: MLL-proof net is a formula tree with a linking on the atoms, such that a correctness criterion is obeyed
- Today we see two ways to translate an MLL-proof net into an RB-graph
- In both cases we obtain an RB-cograph
- In both cases we get the same correctness criterion: An RB-cograph $G$ is the translation of a proof net if and only if it is $\omega$-connected and $\omega$-acyclic.

- Graph theoretically, $T_{RB}(A)$ is not a tree, but it carries the structure of the formula tree.
- \textbf{Exercise 3.8:} Show that if $\pi(\Gamma, \ell)$ is a MLL-prenet, then $\tau(\Gamma, \ell)$ is indeed an RB-cograph.
- \textbf{Exercise 3.9:} Show that in $\tau(\Gamma, \ell)$, no $\omega$-path or $\omega$-cycle has a chord.

- Observe that in a tree-like RB-prenet, every $\omega$-path and every $\omega$-cycle is chordless. Therefore, checking for $\omega$-acyclicity amounts to checking that there is no $\omega$-cycle.
- \textbf{Exercise 3.10:} Prove the theorem. More precisely, show the following:
  1. $\tau(\Gamma, \ell)$ is $\omega$-acyclic iff every switching of $\pi(\Gamma, \ell)$ is acyclic.
  2. $\tau(\Gamma, \ell)$ is $\omega$-connected iff every switching of $\pi(\Gamma, \ell)$ is connected.
Translating MLL-Prenets into RB-graphs (Method 2)

Let a sequent \( \Gamma = A_1, A_2, \ldots, A_n \) and an axiom linking \( \ell \) be given.

1. Mapping \([\cdot]\) from MLL formulas to (labelled) graphs:
   \[
   \begin{align*}
   [a] &= •a \quad \text{(single vertex labelled } a) \\
   [a^\perp] &= •a^\perp \quad \text{(single vertex labelled } a^\perp) \\
   [A \otimes B] &= [A] + [B] \\
   [A \otimes B] &= [A] \times [B]
   \end{align*}
   \]

Let \([\Gamma] = [A_1] + [A_2] + \cdots + [A_n] \)

2. For each linking edge in \( \ell \) add a matching edge between the corresponding atoms

The result is an RB-cograph and called the handsome RB-prenet \( \rho(\Gamma, \ell) \)

Every RB-cograph can be obtained in this way from a sequent \( \Gamma \) and an axiom linking \( \ell \).

Exercise 3.11: Show that if \( \pi(\Gamma, \ell) \) is a MLL-prenet, then \( \rho(\Gamma, \ell) \) is indeed an RB-cograph.

Exercise 3.12: Prove the converse, i.e., prove that every RB-cograph can be labelled such that it is a handsome prenet \( \rho(\Gamma, \ell) \) for some sequent \( \Gamma \) and some linking \( \ell \).

Example

\[
\begin{align*}
\begin{array}{cccccc}
a & b & b^\perp & c & c^\perp & d \\
\gamma & & & & & \gamma \\
\gamma & & & & & \gamma \\
\end{array}
\end{align*}
\]

The result is an RB-cograph.

Theorem: \( \pi(\Gamma, \ell) \) is correct iff \( \rho(\Gamma, \ell) \) is \( \omega \)-acyclic and \( \omega \)-connected.

In that case, \( \rho(\Gamma, \ell) \) is called a handsome proof net.

Folding and Unfolding

The folding and unfolding are performed by small steps that preserve chordless \( \omega \)-paths and chordless \( \omega \)-cycles. Details can be found here:

- Christian Retoé: "Handsome proof-nets: R&B-graphs, perfect matchings and series-parallel graphs". Rapport de Recherche RR-3652, INRIA
4. Lecture

Deep Inference

Willem Heijltjes and Lutz Straßburger

Deep inference

A method for structuring proofs

A derivation from $A$ to $C$:
- Atom $a$
- Horizontal construction with connective $\star$
- Vertical construction with rule $r$ from $B_1$ to $B_2$

A deep inference proof system is given by:
- A set of connectives
- A set of rules

Classical logic

Connectives: $\land$, $\lor$, $\top$, $\bot$

Invertible rules:

Non-invertible rules:
Example: Peirce's law

\[
\vdash \top \\
\frac{\vdash \bot}{\vdash \top}
\]

\[
\frac{\vdash \bot \land \bot}{\vdash \bot} \\
\frac{\vdash \bot \lor \bot}{\vdash \bot}
\]

\[
((\top \lor b) \land \top) \lor a
\]

4/22

Intuitionistic logic

\[
A \land C : = \begin{array}{c}
a \mid A_1 \parallel A_2 \parallel C_1 \parallel C_2 \parallel B_2 \parallel C \\
A \land C \land A
\end{array}
\]

- Connectives: $\lor$, $\land$, $\top$
- Invertible rules:
  \[
  \frac{A \land (B \land C)}{(A \land B) \land C} \quad \frac{A \land B}{B \land A} \quad \frac{A \land A}{\top \land A}
  \]
- Non-invertible rules:
  \[
  \frac{B}{A \lor (B \land A)} \quad \frac{(A \lor B) \land A}{A \land A} \quad \frac{A}{A \land A} \quad \frac{A}{\top}
  \]

5/22

Example: switch

\[
(A \lor B) \land C
\]

\[
\frac{((A \lor B) \land C) \land A}{(A \lor B) \land C} \\
\frac{(A \lor B) \land C}{B}
\]

\[
(A \lor B) \land C
\]

6/22
Multiplicative Linear Logic

\[
\begin{array}{c}
A \vdash C \quad := \quad a \\
\hline
A_1 \quad \ast \quad A_2 \\
\hline
B_1 \quad \ast \quad B_2
\end{array}
\]

- Connectives: \(\otimes, \& , I, \bot\)
- Invertible rules:

\[
\begin{align*}
A \otimes (B \otimes C) & \quad \overset{\alpha}{\Rightarrow} \quad (A \otimes B) \otimes C \\
A \otimes B & \quad \overset{\lambda}{\Rightarrow} \quad B \otimes A \\
A & \quad \overset{\lambda}{\Rightarrow} \quad I \otimes A
\end{align*}
\]

- Non-invertible rules:

\[
\begin{align*}
I & \quad \Rightarrow \quad (A \otimes B) \otimes C \\
A \otimes (B \otimes C) & \quad \Rightarrow \quad A \otimes \bot
\end{align*}
\]

Example

\[
\begin{array}{c}
((a \otimes b) \otimes b) \otimes c \\
\hline
\hline
(a \otimes b) \otimes b \\
\hline
\hline
b \otimes b \\
\hline
\hline
\hline
\hline
\end{array}
\]

From deep inference to proof nets

A derivation from \(A\) to \(B\) becomes a two-sided proof net:

\[
\begin{array}{c}
A \quad \Rightarrow \quad B \\
\hline
\hline
A \\
\hline
B
\end{array}
\]

- Informally: by tracing atoms through a derivation
- Formally: by defining horizontal and vertical construction on proof nets
Tracing atoms

Inductive translation

\[
\begin{align*}
A &::= a \mid A_1 \times A_2 \mid A \\
A_1 &::= B_1 \mid C_1 \\
A_2 &::= B_2 \mid C_2
\end{align*}
\]
Composition

- **Composition is path composition across the common formula**

\[
\sigma \triangleright b \quad (\sigma \triangleright b) \triangleright b \quad (\sigma \triangleright b) \triangleright c \\
((\sigma \triangleright b) \triangleright b) \triangleright b \quad (\sigma \triangleright b) \triangleright b \quad (\sigma \triangleright b) \triangleright c \\
((\sigma \triangleright b) \triangleright b) \triangleright c.
\]

Rules

\[
\frac{I}{A \triangleright \overline{A}} \quad (A \triangleright B) \odot (B \odot C) \quad \frac{A \odot \overline{A}}{\bot} \\

\quad (A \triangleright B) \odot (B \odot C) \quad \frac{A}{A \triangleright \overline{A}} \quad \frac{A}{A \triangleright \overline{A}} \\

\quad \frac{A \triangleright (B \odot C)}{\overline{A}} \quad \frac{(A \triangleright B) \odot \overline{A}^\sigma}{\overline{A}} \quad \frac{\overline{A} \triangleright \overline{A}^\lambda}{\overline{A}} \quad \frac{A \odot (B \odot C)}{\overline{A} \triangleright \overline{A}^\lambda} \quad \frac{A \odot B}{\overline{A} \triangleright \overline{A}^\lambda} \\

\quad (A \triangleright B) \odot C \quad \frac{(A \triangleright B) \odot C}{\overline{A}} \quad \frac{A \odot (B \odot C)}{\overline{A}} \quad \frac{A \odot B}{\overline{A}} \quad \frac{A}{A \triangleright \overline{A}} \\

\quad (A \triangleright B) \odot C \quad \frac{(A \triangleright B) \odot C}{\overline{A}} \quad \frac{A \odot (B \odot C)}{\overline{A}} \quad \frac{A \odot B}{\overline{A}} \quad \frac{A}{A \triangleright \overline{A}}
\]

Switch

\[
\frac{(B \triangleright C) \odot D}{B \triangleright (C \odot D)}
\]
MLL is the logic of undirected, connected, acyclic networks.

- We can add the mix rule to allow disconnected networks
  \[
  A \otimes B \\
  A \vdash B^{\text{mix}}
  \]

- We can drop the axiom and cut to get directed networks (this is called weakly or linearly distributive logic)

- We can add primitives (with inputs and outputs)

\[
\begin{array}{c}
A_1 \otimes \ldots \otimes A_m \\
C_1 \& \ldots \& C_m
\end{array}
\]

\[
f
\]

\[
A_1, A_n \\
\ldots
\]

\[
C_1, C_m
\]

---

From sequent calculus

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash A) \otimes (B \vdash \Delta) \quad s
\]

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash (A \otimes (B \vdash \Delta))) \quad s
\]

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash (A \not\vdash B)) \quad s
\]

---

From sequent calculus

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash A) \otimes (B \vdash \Delta) \quad s
\]

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash (A \not\vdash B)) \quad s
\]

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash (A \not\vdash B)) \quad s
\]

---

From sequent calculus

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash A) \otimes (B \vdash \Delta) \quad s
\]

\[
\Gamma, A \not\vdash B, \Delta \quad \Rightarrow \quad (\Gamma \vdash (A \not\vdash B)) \quad s
\]
### Eta-expansion

\[
\frac{\vdash \overline{A} \otimes B, A \rightarrow B}{\vdash \overline{A} \otimes B, A \rightarrow B} \quad \text{Ax}
\]

\[
\frac{\vdash \overline{A} \otimes B, A, B}{\vdash \overline{A} \otimes B, A \rightarrow B} \quad \otimes
\]

\[
\frac{\vdash \overline{A} \otimes B, A \rightarrow B}{\vdash \overline{A} \otimes B, A \rightarrow B} \quad \Rightarrow
\]

### Cut-elimination

\[
\frac{\vdash \Gamma, A \vdash \Delta, B}{\vdash \Gamma, \Delta, \overline{A} \otimes B} \quad \otimes
\]

\[
\frac{\vdash \Gamma, \Delta, \overline{A} \otimes B}{\vdash \Gamma, \Delta, A \rightarrow B} \quad \text{Cut}
\]

\[
\frac{\vdash \Gamma, A \vdash \Delta, B}{\vdash \Gamma, A \vdash \Delta, A} \quad \text{Cut}
\]

\[
\frac{\vdash \Gamma, \Delta, A}{\vdash \Gamma, \Delta, A} \quad \text{Cut}
\]

\[
\frac{\vdash \Gamma, \Delta, \Theta}{\vdash \Gamma, \Delta, A} \quad \text{Cut}
\]

\[
\frac{\vdash \Gamma, \Delta, \Theta}{\vdash \Gamma, \Delta, A} \quad \text{Cut}
\]

\[
\frac{\vdash \Gamma, \Delta, A}{\vdash \Gamma, \Delta, A} \quad \text{Cut}
\]

\[
\frac{\vdash \Gamma, \Delta, A}{\vdash \Gamma, \Delta, A} \quad \text{Cut}
\]

### Big picture

1. Translations
2. Cut-elimination and eta-expansion


Brünnler & McKinley, An Algorithmic Interpretation of a Deep Inference System, LPAR 2008

Guglielmi, Gundersen & Parigot, A proof calculus which reduces syntactic bureaucracy, RTA 2010

Additive linear logic

Formulae:
\[ A, B, C ::= a | A \oplus B | A \otimes B \]

Sequents:
\[ A \vdash C \quad \text{or} \quad \vdash \overline{A}, C \]

Sequent calculus:

- **Ax**: \[ \frac{}{A \vdash A} \]
- **L**: \[ \frac{A \vdash C \quad B \vdash C}{A \oplus B \vdash C} \]
- **R**: \[ \frac{A, i \vdash C}{A \vdash C_i \otimes C_2} \]
- **Cut**: \[ \frac{A \vdash B \quad B \vdash C}{A \vdash C} \]
- **&L**: \[ \frac{A_1 \vdash C \quad A_2 \vdash C}{A_1 \otimes A_2 \vdash C} \]
- **&R**: \[ \frac{A \vdash B \quad A \vdash C}{A \vdash B \otimes C} \]

Sum and product

\[ \begin{align*}
A & \quad \vdash \quad B \quad \vdash \quad C \\
B & \quad \vdash \quad B \otimes C \\
C & \quad \vdash \quad C \\
B \vdash D & \quad B \otimes C \vdash D \\
B \otimes C \vdash D & \quad A \vdash C \\
A \vdash B & \quad A \vdash B \otimes C \\
A \vdash B & \quad A \vdash B \otimes C \\
A \vdash C & \quad A \vdash B \otimes C
\end{align*} \]
Proof nets: graph–of–rules

\[
\begin{align*}
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta}^{Ax} & \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta + \Gamma}^{+L} \\
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Gamma \times \Delta}^{\times L} & \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta \times \Gamma}^{\times R}
\end{align*}
\]

\[
\begin{align*}
\frac{\Gamma \vdash \Delta, \Delta}{\Gamma \vdash \Delta}^{Ax} & \quad \frac{\Gamma \vdash \Delta, \Delta}{\Gamma \vdash \Delta, \Delta + \Gamma}^{+R,1} \\
\frac{\Gamma \vdash \Delta, \Delta}{\Gamma \vdash \Gamma \times \Delta, \Delta}^{\times R} & \quad \frac{\Gamma \vdash \Delta, \Delta}{\Gamma \vdash \Delta, \Delta + \Gamma}^{+R,1}
\end{align*}
\]

\[
\begin{align*}
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta}^{Ax} & \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta + \Gamma}^{+L} \\
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Gamma \times \Delta}^{\times L} & \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta \times \Gamma}^{\times R}
\end{align*}
\]

Proof nets: (two-sided) sequent + axioms

\[
\begin{align*}
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta}^{Ax} & \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta + \Gamma}^{+L} \\
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Gamma \times \Delta}^{\times L} & \quad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta \times \Gamma}^{\times R}
\end{align*}
\]
**Eta-expansion**

\[
\frac{A + B \vdash A + B}{Ax} \quad \rightsquigarrow \quad \frac{A \vdash A}{Ax} + \frac{B \vdash B}{Ax}
\]

\[\frac{A + A \vdash A + B}{+R,1}\]

\[\frac{B + B \vdash A + B}{+R,2}\]

\[\frac{A + B \vdash A + B}{+L}\]

- Links may be restricted to atoms

---

**Composition**

- Composition is relational composition
- Requires eta-expansion

\[\frac{b \times a}{a \times (b + b)} \quad \rightsquigarrow \quad \frac{b \times a}{b \times b}\]

---

**Correctness**

- A product \(\times\) (and a sum \(\vdash\) on the left) is switched
- A switching deletes one entire subtree (not only the edge)

\[\frac{A \vdash B \quad A \vdash C}{A \vdash B \times C}\]

\[\frac{A \vdash B \times C}{A \vdash A \times C}\]

\[\frac{A \times C}{A \times C}\]
De-sequentialization and switching

\[
\begin{align*}
B \vdash D &\quad \quad C \vdash D &\quad \quad B \vdash D, C \vdash D \\
B \times C \vdash D &\quad \quad C \times D \vdash L &\quad \quad B \times C \vdash D \\
B \vdash D &\quad \quad C \vdash D &\quad \quad B \vdash D, C \vdash D \\
B \times C \vdash D &\quad \quad C \times D \vdash L &\quad \quad B \times C \vdash D \\
B \vdash D &\quad \quad C \vdash D &\quad \quad B \vdash D, C \vdash D \\
B \times C \vdash D &\quad \quad C \times D \vdash L &\quad \quad B \times C \vdash D
\end{align*}
\]

Example

\[
\begin{align*}
A \vdash C &\quad \quad A \vdash C &\quad \quad A \vdash C \\
A \vdash B + C &\quad \quad A \vdash B + C &\quad \quad A \vdash B \times C \\
\end{align*}
\]

Definition

- A linking for an additive sequent \( A \vdash B \) is a relation \( L \subseteq \text{sub}(A) \times \text{sub}(B) \) where \( \text{sub}(A) \) are the subformula occurrences of \( A \). An axiom linking is one where every link relates two occurrences of the same formula.
- A switching \( s \) of \( A \) is a choice of \( B \) or \( C \) for every product \( B \times C \) in \( \text{sub}(A) \). A co-switching chooses on sums \( B + C \).
- The resolution \( s(A) \subseteq \text{sub}(A) \) given by a switching \( s \) consists of those subformula occurrences \( C \) where for every \( B_1 \times B_2 \) in \( A \), if \( C \) is in \( B_1 \) then \( s \) chooses \( B_1 \). A co-resolution is defined analogously for a co-switching.
- A resolution of a linking \( L \) for \( A \vdash B \) is the restriction of \( L \) to \( r(A) \times s(B) \) for some co-switching \( r \) of \( A \) and switching \( s \) of \( B \).
- An additive proof net is a sequent \( A \vdash B \) with an axiom linking \( L \) such that every resolution is a singleton.
Coalescence: correctness by rewriting

\[
\begin{align*}
\frac{B \vdash D}{B \times C \vdash D} & \quad \times L & \quad \frac{C \vdash D}{B \times C \vdash D} & \quad \times L & \quad \frac{B \vdash D \quad C \vdash D}{B + C \vdash D} & \quad + L \\
B \times C & \quad B \times C & \quad B \times C & \quad B \times C & \quad B \times C & \quad B \times C
\end{align*}
\]

Example again

Theorem
A linking is a proof net if and only if it coalesces, if and only if it sequentializes.

Theorem
Coalescence for a linking over \( A \vdash B \) is decidable in \( O(|A| \times |B|) \).

- Apply coalescence steps to show correctness of this example.
Deep inference: classical logic

\[
\begin{array}{c}
A \\
C
\end{array}
::= 
\begin{array}{c}
A_1 \\
C_1
\end{array} \times 
\begin{array}{c}
A_2 \\
C_2
\end{array} \\
\begin{array}{c}
A \\
B_1 \\
C
\end{array} \quad \begin{array}{c}
A \\
B_2 \\
C
\end{array}
\]

- Connectives: $\land$, $\lor$, $\top$, $\bot$
- Invertible rules:

\[
\begin{align*}
A \lor (B \land C) & \quad (A \lor B) \land C \\
\neg \lor A & \quad \neg \land A \\
(A \land B) \lor C & \quad (A \land B) \land C
\end{align*}
\]

- Non-invertible rules:

\[
\begin{align*}
\top & \quad \bot \\
A \lor \bot & \quad \bot \lor A \\
A & \quad A \land \bot
\end{align*}
\]

Deep inference: additive linear logic

\[
\begin{array}{c}
A \\
C
\end{array}
::= 
\begin{array}{c}
A_1 \\
C_1
\end{array} \times 
\begin{array}{c}
A_2 \\
C_2
\end{array} \\
\begin{array}{c}
A \\
B_1 \\
C
\end{array} \quad \begin{array}{c}
A \\
B_2 \\
C
\end{array}
\]

- Connectives: $\times$, $+$, $1$, $0$
- Invertible rules:

\[
\begin{align*}
A + (B + C) & \quad (A + B) + C \\
\neg + A & \quad \neg + A \\
(A + B) \times C & \quad (A \times B) \times C
\end{align*}
\]

- Non-invertible rules:

\[
\begin{align*}
0 & \quad A + A \lor A \\
\bot & \quad \bot \lor \bot
\end{align*}
\]

Inductive translation

\[
\begin{array}{c}
A \\
C
\end{array}
::= 
\begin{array}{c}
A_1 \\
C_1
\end{array} \times 
\begin{array}{c}
A_2 \\
C_2
\end{array} \\
\begin{array}{c}
A \\
B_1 \\
C
\end{array} \quad \begin{array}{c}
A \\
B_2 \\
C
\end{array} \\
\begin{array}{c}
a \\
A_1 \land A_2 \\
C_1 \land C_2
\end{array} \quad \begin{array}{c}
a \\
C_1 \land C_2
\end{array} \\
\begin{array}{c}
\bot \\
B_1
\end{array} \quad \begin{array}{c}
\bot \\
B_2
\end{array} \quad \begin{array}{c}
A \\
r
\end{array} \quad \begin{array}{c}
\top \\
C
\end{array}
\]

\[
\begin{array}{c}
\top \\
\top
\end{array}
\]

\[
\begin{array}{c}
\bot \\
\bot
\end{array}
\]

\[
\begin{array}{c}
\bot \\
\bot
\end{array}
\]
Example again again

\[
\begin{align*}
\Delta & \quad a + (b \cdot c) \\
\Delta & \quad a + (b \cdot c) \\
\Delta & \quad a + (b \cdot c)
\end{align*}
\]

20/23
Big picture

Sequent calculus \[ \rightarrow \] Deep inference
\[ \rightarrow \]

Proof nets (sequent + axioms)

1. Translation
2. Translation with eta-expansion and cut-elimination

Bibliography


A. Joyal, *Free Lattices, Communication and Money Games*, Proc. 10th Int. Cong. of Logic, Methodology and Philosophy of Science, 1995


What are Fibrations?

- originate in topology as a generalization of fiber bundles:

\[ f^{-1}(Y) \]

- fibrations in topology:

\[ f^{-1}(Y) \]
Fibrations in Directed Graphs

**Definition:**
A **fibration** is a graph homomorphism \( f: G \to H \) such that for all \( v \in V_G \) and \( w' \in V_H \), if \( w' \nRightarrow_H f(v) \) then there is a unique \( w \in V_G \) with \( w \Rightarrow_G v \) and \( f(w) = w' \).

Fibrations in Undirected Graphs

**Definition:**
A **fibration** is a graph homomorphism \( f: G \to H \) such that for all \( v \in V_G \) and \( w' \in V_H \), if \( \{w', f(v)\} \in E_H \) then there is a unique \( w \in V_G \) with \( \{w, v\} \in E_G \) and \( f(w) = w' \).

Skew Fibrations in Undirected Graphs

**Definition:**
A **skew fibration** is a graph homomorphism \( f: G \to H \) such that for all \( v \in V_G \) and \( w' \in V_H \), if \( \{w', f(v)\} \in E_H \) then there is a \( w \) with \( \{w, v\} \in E_G \) and \( \{w', f(w)\} \notin E_H \).

- In skew fibrations, two conditions that hold in fibrations are dropped:
  1. uniqueness of \( w \) is not demanded anymore, only existence
  2. we no longer demand that \( f(w) = w' \) but only that there is no edge between the two.
- Note that in an undirected graph there are no reflexive edges. That means that \( f(w) = w' \) is allowed, and a fibrations is indeed a special case of a skew fibration.
- The term **skew fibration** is due to Hughes:
  - Dominic Hughes: "Proofs Without Syntax".
Skew Fibrations in Undirected Graphs

A skew fibration is a graph homomorphism \( f: G \to H \) such that for all \( v \in V_G \) and \( w' \in V_H \), if \( \{w', f(v)\} \in E_H \) then there is a \( w \) with \( \{w, v\} \in E_G \) and \( \{w', f(w)\} \notin E_H \).

Examples:

```
  a —— b   a —— f(b) —— c
  ↓    ↓   ↓    ↓
  a —— b
```

```
  a —— b   a —— f(b) —— c
  ↓        ↓
  a —— f(b) —— c
```

```
  a —— b
  ↓    ↓
  a —— b
```

```
  a —— b   a —— f(b) —— c
  ↓        ↓
  a —— f(b) —— c
```

For details on SKS consult:
Decomposition for the Down-Fragment

\[
\begin{align*}
\text{SKSg} & : \not\! \vdash A \rightarrow A \\
& \quad \downarrow_i \quad \not\! \vdash A \rightarrow (C \lor A) \land B \\
& \quad \downarrow_s \quad \not\! \vdash C \lor (A \land B) \\
\end{align*}
\]

Theorem: (Decomposition)

\[
\begin{align*}
\text{SKSg} & \not\! \vdash A \\
\iff & \not\! \vdash (i_\downarrow, s_\downarrow) \\
& \not\! \vdash (c_\downarrow, w_\downarrow) \\
\end{align*}
\]

Recall from earlier this week, that proofs in unit-free MLL can be described by proof nets and critically chorded RB-cographs.

Decomposition for the Down-Fragment (without units)

\[
\begin{align*}
\text{SKSg'} \not\! \vdash A \\
& \quad \downarrow_i \quad \not\! \vdash A \rightarrow A \\
& \quad \downarrow_j \quad \not\! \vdash B \rightarrow (A \land A) \\
& \quad \downarrow_s \quad \not\! \vdash (C \lor A) \land B \\
& \quad \downarrow_c \quad \not\! \vdash A \lor A \\
\end{align*}
\]

Theorem: (Decomposition)

\[
\begin{align*}
\text{SKSg'} & \not\! \vdash A \\
\iff & \not\! \vdash (i_\downarrow, s_\downarrow) \\
& \not\! \vdash (c_\downarrow, w_\downarrow) \\
\end{align*}
\]

First Skew Fibration Theorem

Theorem: Let \( A \) and \( B \) be (classical logic) formulas.

Then there is a skew fibration \( f : \lbrack A \rbrack \rightarrow \lbrack B \rbrack \) iff \( (c_\downarrow, w_\downarrow) \not\! \vdash B \)

Examples:

\[
\begin{align*}
& \quad \not\! \vdash (a \land b) \lor (a \land b) \lor ((a \lor b) \land (a \lor b)) \\
& \quad \downarrow_w \quad \not\! \vdash (a \land b) \lor (a \land b) \lor ((a \lor b) \land (a \lor b)) \lor c \\
& \quad \not\! \vdash (a \land b) \lor ((a \lor b) \land (a \lor b)) \lor c \\
\end{align*}
\]

This theorem has first been observed by Hughes:


An alternative proof using deep inference can be found here:

ALL Proof Nets and Skew Fibrations

\[(a \times b) + (a \times c)\]
\[a \times (b + c)\]

\[(a + b) \times (a + c)\]

This is a skew fibration!

Exercise 6.1: Show that every skew fibration defines an ALL proof net.

Another ALL Proof Net

\[a + (b \times c)\]

\[(a + b) \times (a + c)\]

Not even a map!

Not a skew fibration!

Exercise 6.2: (Difficult) Prove the theorem.

Second Skew Fibration Theorem

**Theorem:** Let \( A \) and \( B \) be formulas. Then the following are equivalent:

1. There is an ALL proof net for \( A \vdash B \).
2. There is a formula \( C \) such that there are skew fibrations \( f: \llbracket C \rrbracket \to \llbracket B \rrbracket \) and \( g: \llbracket \overline{C} \rrbracket \to \llbracket \overline{A} \rrbracket \).
Summary (Multiplicative Part)

sequent calculus:
\[
\begin{align*}
\text{ax} & : x, x \vdash \Gamma, \Delta, \neg x \\
\text{↓} & : \Gamma, \Delta, \neg x \vdash \Gamma, \Delta \\
\end{align*}
\]

deep inference:
\[
\begin{align*}
\text{ax} & : 1 \vdash \Gamma, \Delta, \neg x \\
\text{↓} & : \Gamma, \Delta, \neg x \vdash \Gamma, \Delta \\
\end{align*}
\]

MLL proof nets
- switching criterion (exponential)
- contraction criterion (polynomial)

handsome proof nets
- critically chorded RB-cographs (polynomial)

Summary (Additive Part)

sequent calculus:
\[
\begin{align*}
\text{ax} & : x, x \vdash \Gamma, \Delta, x \lor (x \land \neg x) \\
\text{↓} & : \Gamma, \Delta, x \lor (x \land \neg x) \vdash \Gamma, \Delta \\
\end{align*}
\]

deep inference:
\[
\begin{align*}
\text{w} & : A \vdash A \\
\text{↓} & : A \vdash A, A \\
\end{align*}
\]

ALL proof nets
- switching criterion (exponential)
- coalescence criterion (polynomial)

skew fibrations
- homomorphisms on cographs with skew lifting property (polynomial)
Combinatorial proofs

Aim

What is a good semantics of classical logic?

- Reduction is non-confluent $\implies$ no canonical normal forms
- Cartesian closed categories with duality $\implies$ collapse
- Embeddings in intuitionistic logic $\implies$ break duality $A \not\sim \overline{A}$
- Distributivity gives canonical normal forms $\implies$ collapse

$$A \land (B \lor C) \sim (A \land B) \lor (A \land C)$$
Naive cut-reduction

\[ \frac{\vdash \Gamma, A \quad \vdash \neg A, \Delta}{\vdash \Gamma, \Delta} \quad \text{Cut} \]

\[ \frac{\vdash \Gamma, A, A \quad \vdash \neg A, \Delta}{\vdash \Gamma, \Delta} \quad \text{Cut} \]

Lafont’s examples: non-confluence

\[ \frac{\vdash \Gamma}{\vdash \Gamma, \Delta} \quad \text{w} \]

\[ \frac{\vdash \Gamma, A, \Delta}{\vdash \neg A, \Delta} \quad \text{Cut} \]

Lafont’s examples: non-termination

\[ \frac{\vdash \Gamma, A, A}{\vdash \Gamma, \Delta} \quad \text{c} \]

\[ \frac{\vdash \Gamma, A, \Delta}{\vdash \neg A, \Delta} \quad \text{Cut} \]

- This is a cut on two contracted formulae.
- First, the cut interacts with the blue (right) contraction. This duplicates the left branch (the red cut) and creates a new series of contractions below the two new cuts.
- Second, the top cut interacts with its red (left) contraction. This duplicates the right branch and creates a new series of contractions below the two new cuts.
- The cut at the bottom is now in the original situation, with a cut on two contracted formulae.
- It follows that naive cut-reduction is not strongly normalizing.
We cannot have a non-trivial semantics of classical proof and:

- involutive negation \( \overline{A} \cong A \)
- conjunction and disjunction are products and coproducts
- disjunction acts as implication \( A \supset B = \overline{A} \lor B \)

Intuitionistic natural deduction proves \( A \vdash (A \supset \top) \supset \bot \), and extends to classical natural deduction by including a rule \( \bot \mathbf{E} \),

\[
\frac{}{\bot}
\]

which proves \( (A \supset \bot) \supset \bot \vdash A \) (double-negation elimination). However, the two proofs do not form an isomorphism: composing them does not necessarily give an identity. Classical natural deduction is computational, and double-negation elimination relates to the call/cc construct (call-with-current-continuation); see Parigot’s \( \lambda\mu \)-calculus and subsequent work.

- Omitting some of the equations for products and coproducts (while keeping the necessary contraction and weakening rules) can be sufficient to prevent the semantics from becoming trivial. However, equating proofs under cut-elimination automatically gives products and coproducts, so this is no longer possible.
- Additive linear logic has involutive negation, products, and coproducts, but its disjunction does not have an axiom (only the separate meta-connective \( \vdash \) does), and so cannot take the role of an implication. While it is possible to build a classical model this way, again equating proofs under cut-elimination is ruled out.

### Combinatorial proofs

Handsome proof net

Cograph

Skew fibration

Conclusion

Example: distributivity

\[ a \land (b \lor c) \implies (a \land b) \lor (a \land c) \]

\[ a \lor (b \land c), (a \land b) \lor (a \land c) \]
Special case:

\[
\Gamma, A \vdash B, \Delta \\
\Gamma, A \land B \vdash \Delta \quad \sim \\
\Gamma, A \quad \vdash \quad \Gamma, A \land B, \Delta
\]

Special special case:

\[
\Gamma, A \land B, \Delta \\
\Gamma, A \quad \vdash B, \Delta, \Delta \quad \leftrightarrow \\
\Gamma, A \land B \quad \vdash \Delta, \Delta \
\]

The translation from sequent proofs is non-deterministic (unless we allow mix in our handsome proof nets)
The case conjunction–contraction:

\[
\begin{align*}
\frac{\Gamma, A, A}{\Gamma, A} & \frac{\Gamma, A}{} \frac{B, \Delta}{\Gamma, A \land B, \Delta} \\
\quad \frac{\Gamma, A, A}{\Gamma, A \land B, \Delta} & \frac{?}{\Gamma, A, A} \frac{B, \Delta}{\Gamma, A \land B, \Delta} \\
\end{align*}
\]

\[
\begin{align*}
\frac{\Gamma, A, A}{\Gamma, A \land B, \Delta} & \frac{B, \Delta}{\Gamma, A \land B, \Delta} \\
\quad \frac{\Gamma, A \land B, A \land B, \Delta}{\Gamma, A \land B, \Delta} & \frac{\Delta}{\Gamma, A \land B, \Delta} \\
\end{align*}
\]

Combinatorial proofs are *complexity-sensitive*

---

**Example: distributivity, the other direction (1)**

\[ a \land (b \lor c) \iff (a \land b) \lor (a \land c) \]

---

**Example: distributivity, the other direction (2)**

\[ a \land (b \lor c) \iff (a \land b) \lor (a \land c) \]
Example: distributivity?

\[ a \land (b \lor c) \iff (a \land b) \lor (a \land c) \]

Not a combinatorial proof: the lower part is an additive proof net but not a skew fibration

Big question

Are combinatorial proofs a good semantics of classical logic?

Bibliography


L. Straßburger, Towards a Theory of Proofs of Classical Logic, Habilitation à diriger des recherches, École Polytechnique, Palaiseau, France
First-Order Combinatorial Proofs

Definition:
A first-order combinatorial proof of a formula $A$ is a skew bifibration $\phi : C \to G$ where $C$ is a fonet and $G$ is the graph of the formula $A$.

First-Order Formulas

Terms: $t ::= x | \text{f}(t_1, \ldots, t_n)$
Atoms: $a ::= 1 | 0 | \rho(t_1, \ldots, t_n) | \overline{\rho(t_1, \ldots, t_n)}$
Formulas: $A ::= a | A \land A | A \lor A | \exists x.A | \forall x.A$
Sequents: $\Gamma ::= A_1, A_2, \ldots, A_m$

Negation:
$\top = 0$
$\bot = 1$
$\exists x.A = \forall x.A$
$A \land B = \overline{\exists x.\overline{A}}$
$A \lor B = \overline{\forall x.\overline{A}}$

Rectified Formulas: all bound variables are distinct from one another and from all free variables
Graphs of Formulas

Operations on (undirected) graphs:

\[ G + H \]
\[ G \times H \]

Mapping \([\bullet] \) from formulas to (labelled) graphs:

\[ [a] = \bullet a \] (for any atom \( a \))

\[ [A \lor B] = [A] + [B] \]
\[ [A \land B] = [A] \times [B] \]
\[ [\exists x. A] = x \times [A] \]

Example:

\[ [\exists x. (\bar{p}x \lor (\forall y. py))] = \]

Definition: A fograph (first-order graph) is the graph \([A]\) of a formula \( A \).

Graphs of Formulas

Congruence Relation \( \equiv \) on Formulas:

\[ A \land B \equiv B \land A \]
\[ A \lor B \equiv B \lor A \]
\[ \forall x. \forall y. A \equiv \forall y. \forall x. A \]
\[ \exists x. \exists y. A \equiv \exists y. \exists x. A \]
\[ (A \land B) \land C \equiv A \land (B \land C) \]
\[ (A \lor B) \lor C \equiv A \lor (B \lor C) \]
\[ \forall x. (A \lor B) \equiv (\forall x. A) \lor B \]
\[ \exists x. (A \land B) \equiv (\exists x. A) \land B \]

\( x \) not free in \( B \)

Theorem: For rectified formulas \( A \) and \( B \) we have

\[ A \equiv B \iff [A] = [B] \]

Example:

\[ \exists x. (\bar{p}x \lor (\forall y. py)) \equiv \exists y. (py \lor \bar{p}x) \]

First-Order Combinatorial Proofs

Definition:

A first-order combinatorial proof of a formula \( A \) is a skew bifibration \( \phi : C \rightarrow G \) where \( C \) is a fonet and \( G \) is the graph of the formula \( A \).
**Fonets**

**Linked Fograph:** a partially colored fograph, where
- each color consists of two literals with dual predicate symbols
- every literal is either 1-labelled or in a link

**Dualizer:** substitution unifying all the links

**Dependency:** pair \( \{x, y\} \) such that most general dualizer assigns to the existential \( x \) a term containing the universal \( y \)

**Leap Graph:** vertices: as in the fograph
edges: links and dependencies

**Example:**

![Example Diagram]

- **linked fograph**
- **dualizer**
- **leap graph**

**Definition:** A set of vertices induces a bimatching if it induces a matching in the fograph and a matching in the leap graph.

**Definition:** A fonet (or first-order net) is a linked fograph which has a dualizer but no induced bimatching.

**Example:**

![Example Diagram]

- **linked fograph**
- **dualizer**
- **leap graph**

- Observe that “no bimatching” is the same condition as “critically chorded” for RB-cograph. The difference is that here the “blue” edges (the edges of the leap graph) do not form a perfect matching for the vertex set.

- **Exercise 8.1:** Show that Retoré’s “handsome proof nets” are a special case of fonets.

---

**First-Order Combinatorial Proofs**

**Definition:**

A first-order combinatorial proof of a formula \( A \) is a skew bifibration \( \phi: C \to G \) where \( C \) is a fonet and \( G \) is the graph of the formula \( A \).
Skew Bifibrations

\[ \exists \omega \in \mathcal{W} \]  
\[ \phi(v) \]

Definition: Let \( \mathcal{G} \) and \( \mathcal{H} \) be fographs. A skew bifibration \( \phi: \mathcal{G} \rightarrow \mathcal{H} \) is a label- and existential-preserving graph homomorphism that is a skew fibration on the underlying cographs and a fibration on the binding graphs.

Example:

First-Order Combinatorial Proofs

Definition:

A first-order combinatorial proof (focp) of \( A \) is a skew bifibration \( \phi: \mathcal{C} \rightarrow \mathcal{G} \) where \( \mathcal{C} \) is a fonet and \( \mathcal{G} \) is the graph of the formula \( A \).

Examples:

More Compact Notation:

Theorem:

First-order combinatorial proofs are sound and complete for first-order logic.

First-Order Sequent Calculus

\[ \frac{\text{ax}}{\Gamma, A, B} \quad \frac{\Gamma, A \lor B}{\Gamma, A} \quad \frac{\Gamma, B}{\Gamma, A \land B} \]

1 \[ \frac{\text{mix}}{\Gamma, \Delta, \Delta} \]

\[ \frac{\exists x \Gamma, A[x/t]}{\Gamma, \exists x A} \quad \frac{\Gamma, A \text{ (x not free in } \Gamma)}{\Gamma, \forall x A} \]

cont \[ \frac{\Gamma, A}{\Gamma, A} \quad \text{weak} \[ \frac{\Gamma, A}{\Gamma, A} \]

LK1: sequent calculus for first-order logic (all rules)  
MLL1: linear fragment (dashed box)

Relation between LK1 and focp:  
MLL1 proof \rightarrow fonet √  
LK1 proof \rightarrow focp √  
fonet \rightarrow MLL1 proof √  
focp × LK1 proof ×
Exercise 8.4: Give the deep inference derivations for the four combinatorial proofs above.
Resource Management in First-Order Proofs

Theorem:

\[ \begin{align*}
A \\
\{m,m,v,m_3,=\} & \vdash A' \\
\{w,v,=\} & \vdash B \\
\implies \quad \{w,v,=\} & \equiv \{w,v,=\} \\
\vdash A' & \iff \{w,v,=\} \\
\{w,v,=\} & \equiv \{w,v,=\} \\
\implies \quad \text{skew bifibration} \\
\vdash [B]
\end{align*} \]

Example

\[ A \lor A \]

\[ A \lor A \]

\[ A \lor A \]

\[ A \lor A \]

\[ A \lor A \]

\[ A \lor A \]

\[ (\forall x px) \lor (\forall y py) \]

Example

\[ qab \lor qba \]

\[ qab \lor qba \]

\[ qab \lor qba \]

\[ qab \lor qba \]

\[ qab \lor qba \]

\[ qab \lor qba \]

\[ qab \lor qba \lor (\exists x y qxy) \]

\[ qab \lor qba \lor (\exists x y qxy) \]

\[ qab \lor qba \lor (\exists x y qxy) \]

\[ qab \lor qba \lor (\exists x y qxy) \]

\[ qab \lor qba \lor (\exists x y qxy) \]

\[ qab \lor qba \lor (\exists x y qxy) \]

\[ qab \lor qba \lor (\exists x y qxy) \]
That formula is interesting for several reasons:

- This formula is called the **drinker’s formula** because it can be read as “In every bar there is a person \(x\), such that whenever that person is drinking then everyone else is also dinking”: \(\exists x (px \supset \forall y py)\).
- It is not valid intuitionistically.
- In the sequent calculus the whole formula needs to be duplicated first. This can be simulated in deep inference. However, in deep inference and in combinatorial proofs, this duplication is not necessary.
A Last Word on Classical Combinatorial Proofs

**Definition:**
A **combinatorial proof** of a formula $A$ is a skew fibration $f : C \to [A]$ from a critically chorded RB-cograph (aka nicely colored cograph) $C$ to the cograph of $A$, such that two vertices that paired in $C$ are mapped to dual atoms in $[A]$.

**Example:**

In different publications and also in different lectures in this curse, different notations are used. But it should clear that these are indeed only different notations for the same thing.

The last one on the first second row has the advantage of using the least amount of ink. It has been introduced in:

It has the advantage that one can easily see the correspondence to the flow graph of the atoms in a derivation, as indicated on the following two slides.

Another Last Word on Classical Combinatorial Proofs

These are all the same combinatorial proofs, just drawn in different ways with some formulas in the conclusion flipped upside-down.
A Final Last Word on Classical Combinatorial Proofs (and Deep Inference)

From Classical to Intuitionistic Combinatorial Proofs

Arenas

Definition: An arena is a directed acyclic graph (dag) that is

- L-free: if \( v \leftarrow u \rightarrow x \rightarrow w \) then \( v \rightarrow w \)

- \( \Sigma \)-free: if \( u \leftarrow x \rightarrow v \leftarrow y \rightarrow w \) then \( x \rightarrow w \) or \( y \rightarrow u \)

- and has a partition (equivalence relation \( \sim \) on vertices)

Example:
From Formulas to Arenas

- **Operations on dags:**

\[
\begin{align*}
G & \cap H \\
G + H \\
\begin{array}{c}
G \\
H
\end{array}
\end{align*}
\]

- **Formulas:**

\[
A, B ::= P \mid A \land B \mid A \supset B
\]

- **Translating formulas to dags:**

\[
\begin{align*}
[A \land B] & = [A] + [B] \\
[A \supset B] & = [A] \triangleright [B]
\end{align*}
\]

**Example:**

\[
((P \supset P) \supset Q) \land Q \supset R \land ((S \supset S) \supset S)
\]

\[
\begin{array}{c}
P \\
\text{\rightarrow}
\end{array}
\begin{array}{c}
Q \\
\text{\rightarrow}
\end{array}
\begin{array}{c}
R \\
\text{\rightarrow}
\end{array}
\begin{array}{c}
S \\
\text{\rightarrow}
\end{array}
\]

**Theorem:** A dag is an arena if and only if it is the translation of a formula.

**Theorem:**

\[
[A] = [B] \iff A \equiv B.
\]

where \(\equiv\) is generated from:

\[
\begin{align*}
(A \land B) \supset C & \equiv A \supset (B \supset C) \\
A \land (B \land C) & \equiv (A \land B) \land C \\
A \land B & \equiv B \land A
\end{align*}
\]

and \(A \equiv B\) if \(A\) and \(B\) are the same up to a renaming of atoms.

---

**Arena Nets**

- **linked arena:**

  every partition (link) consists of exactly two vertices:

  - one even and one odd

  **Example:**

  \[
  \begin{array}{c}
P^* \rightarrow P^* \\
\end{array}
  \begin{array}{c}
Q^* \rightarrow Q^* \\
\end{array}
  \begin{array}{c}
R^* \\
\end{array}
  \begin{array}{c}
S^* \\
\end{array}
  \]

- **link graph (of a linked arena):**

  remove all edges into even vertices
  add a link edge for each link
  (from odd to even)

  **Example:**

  \[
  \begin{array}{c}
P^* \\
\end{array}
  \begin{array}{c}
Q^* \\
\end{array}
  \begin{array}{c}
R^* \\
\end{array}
  \begin{array}{c}
S^* \\
\end{array}
  \]

- The term arena comes from games semantics, e.g.:


  - However, the definition we present here originates from:

    - Willem Heijltjes and Dominic Hughes and Lutz Straßburger: *Intuitionistic Proofs without Syntax*. LICS 2019

  - The purpose of our definition is to cover exactly the notion of IPL formula modulo \(\equiv\).
**Arena Nets**

**Example:**

linked arena

```
\begin{align*}
P \to P & \to Q \to Q \\
R \to S & \to S
\end{align*}
```

link graph

```
\begin{align*}
P & \to P \\
Q & \to Q \\
R & \to R \\
S & \to S
\end{align*}
```

**Definition:** An arena net is a linked arena satisfying:

- **Acyclicity:** The link graph is acyclic.
- **Functionality:** In the link graph, every path from an odd vertex \( v \) to a root visits a consequent of \( v \) (in the arena).

**Theorem:** Let \( [A] \) be a linked arena. TFAE:

1. \( [A] \) is an arena net;
2. the corresponding linking on \( A \) forms an IMLL proof net for \( A \);
3. \( [A] \) can be constructed via IMLL sequent calculus.

**Example:**

\(( (P \supset P) \supset Q) \land R \supset Q \land ((R \supset S) \supset S)\)

**Intuitionistic Combinatorial Proofs**

\(( (a \supset a) \supset b) \supset (b \land b)\)
Skew fibrations

\( \forall x, y \) iff \( x \) and \( y \) are distinct and meet at odd depth or do not meet at all
\( \forall x, y \) otherwise

Example: 
\[ u^0 \rightarrow w^0 \rightarrow y^0 \quad u \not\equiv v \quad w \not\equiv x \]
\[ v^0 \rightarrow x^0 \rightarrow z^0 \quad w \not\equiv x \quad x \not\equiv y \]

Definition: A skew fibration \( f: \mathcal{G} \rightarrow \mathcal{H} \), of an arena \( \mathcal{G} \) over an arena \( \mathcal{H} \), is a function \( f: V_\mathcal{G} \rightarrow V_\mathcal{H} \) that preserves

edges: \( v \rightarrow_\mathcal{G} w \) implies \( f(v) \rightarrow_\mathcal{H} f(w) \)

equivalence: \( v \sim_\mathcal{G} w \) implies \( f(v) \sim_\mathcal{H} f(w) \)

roots: \( r \rightarrow_\mathcal{G} r \) implies \( f(r) \rightarrow_\mathcal{H} \)

conjuncts: \( v \land_\mathcal{G} w \) implies \( f(v) \land_\mathcal{H} f(w) \)

and satisfies the following skew lifting condition:

- if \( f(v) \land_\mathcal{H} w \) then there exists \( u \) with \( v \rightarrow_\mathcal{G} u \) and \( f(u) \land_\mathcal{H} w \).

Intuitionistic combinatorial proofs

Definition: An intuitionistic combinatorial proof or ICP of a formula \( A \) is a skew fibration \( f: \mathcal{G} \rightarrow [A] \) of an arena net \( \mathcal{G} \) over the arena \([A]\).

Examples:

From Sequent Calculus to ICP

Theorem: The translation \([\pi]\) of a sequent proof \( \pi \) is an ICP.

Theorem: This translation is surjective.

Theorem: This translation is polynomial (in both directions).

Theorem: \([\pi_1] = [\pi_2] \) iff \( \pi_1 \equiv \pi_2 \) (modulo non-duplicating rule permutations).
Comparison to $\lambda$-Calculus

Sequent Calculus Rule Permutations

<table>
<thead>
<tr>
<th>$\Gamma, A, B, C \vdash D$</th>
<th>$\Gamma \vdash A, B, \Delta, C \vdash D$</th>
<th>$\Gamma, A \triangleright B, \Delta \triangleright C \triangleright D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma, A \triangleright B, \Delta \triangleright C \triangleright D$</td>
<td>$\Gamma, A \triangleright B, \Delta, C \vdash D$</td>
<td>$\Gamma, A \triangleright B, \Delta \triangleright C \triangleright D$</td>
</tr>
<tr>
<td>$\Delta \vdash C$</td>
<td>$\Delta \vdash C$</td>
<td>$\Delta \vdash C$</td>
</tr>
<tr>
<td>$\Gamma \vdash A, B, \Delta \vdash C$</td>
<td>$\Gamma \vdash A, B, \Delta \vdash C$</td>
<td>$\Gamma \vdash A, B, \Delta \vdash C$</td>
</tr>
<tr>
<td>$\Gamma, A \triangleright B, \Delta \vdash C$</td>
<td>$\Gamma, A \triangleright B, \Delta \vdash C$</td>
<td>$\Gamma, A \triangleright B, \Delta \vdash C$</td>
</tr>
<tr>
<td>$\Gamma \vdash A, B, \Delta \vdash C$</td>
<td>$\Gamma \vdash A, B, \Delta \vdash C$</td>
<td>$\Gamma \vdash A, B, \Delta \vdash C$</td>
</tr>
</tbody>
</table>

Comonoid Transformations

<table>
<thead>
<tr>
<th>$\Gamma, A, B, B \vdash C$</th>
<th>$\Gamma, A, A, B, B \vdash C$</th>
<th>$\Gamma, A, A, B, B \vdash C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma, A, A, B, B \vdash C$</td>
<td>$\Gamma, A, A, B, B \vdash C$</td>
<td>$\Gamma, A, A, B, B \vdash C$</td>
</tr>
<tr>
<td>$\Gamma, A, B \vdash C$</td>
<td>$\Gamma, A, B \vdash C$</td>
<td>$\Gamma, A, B \vdash C$</td>
</tr>
<tr>
<td>$\Gamma, A, A \vdash B$</td>
<td>$\Gamma, A, A \vdash B$</td>
<td>$\Gamma, A, A \vdash B$</td>
</tr>
</tbody>
</table>

Exercise 9.2: How many possible rule permutations are there?
Formula-isomorphism transformations

\[
\begin{align*}
\Gamma, A, B &\vdash C \\ \Gamma, A \land B &\vdash C & \equiv & \Gamma, A, B &\vdash C \\
\Gamma, A &\vdash \Delta \vdash B & \Delta &\vdash B & \Delta &\vdash A & \equiv & \Delta &\vdash B & \Delta &\vdash A \\
\Gamma, A, B, C &\vdash D & \Gamma, A, B &\vdash C &\vdash D & \Lambda &\equiv & \Gamma, A, B, C &\vdash D & \Gamma, A, B &\vdash C &\vdash D \\
\Gamma, A, B, C &\vdash D & \Gamma, A &\vdash \Delta &\vdash (A \land B) &\vdash C &\equiv & \Gamma, A &\vdash \Delta &\vdash (A \land B) &\vdash C \\
\Gamma, A &\vdash \Delta \vdash B & \Delta &\vdash B & \Lambda &\vdash C &\equiv & \Gamma, A &\vdash \Delta &\vdash B & \Lambda &\vdash C \\
\Gamma, A, B &\vdash C & \Gamma, A &\vdash B &\vdash C &\equiv & \Gamma, A &\vdash B &\vdash C \\
\Gamma, A &\vdash \Delta \vdash B & \Delta &\vdash B &\vdash C &\equiv & \Gamma, A &\vdash \Delta &\vdash B &\vdash C \\
\Gamma, A &\vdash \Delta \vdash B & \Delta &\vdash B &\vdash C &\equiv & \Gamma, A &\vdash \Delta &\vdash B &\vdash C \\
\Gamma, A, B &\vdash C, \Lambda &\vdash D & \Gamma, A &\vdash \Delta &\vdash B &\vdash C, \Lambda &\vdash D & \equiv & \Gamma, A &\vdash \Delta &\vdash B &\vdash C, \Lambda &\vdash D \\
\Gamma, A, B &\vdash C, \Lambda &\vdash D & \Gamma, A &\vdash \Delta &\vdash B &\vdash C, \Lambda &\vdash D & \equiv & \Gamma, A &\vdash \Delta &\vdash B &\vdash C, \Lambda &\vdash D
\end{align*}
\]
Intuitionistic composition

\[ \Gamma \vdash A + A, \Delta \vdash B \iff \Gamma, A \triangleright B \]

For intuitionistic combinatorial proofs

Option 1: use a special left-implication connective \( \triangleright \)

\[ \Gamma \vdash A + A, \Delta \vdash B \iff \Gamma, A \triangleright A, \Delta \vdash B \]

- The conclusion is still \( \Gamma, \Delta \vdash B \) and \( A \triangleright A \) is hidden
- \( \triangleright \) is a meta-connective and may only occur at top level
- Cut-elimination is standard (local rewriting with kingdoms)

Exercise 10.1: Add an explicit cut-connective to MLL proof nets in this way, and give the cut-elimination steps.

Option 2: compute the result directly

\[ \Gamma \vdash A + A, \Delta \vdash B \iff \Gamma, \Delta \vdash B \]

- Fixes the reduction strategy
- Game semantics and Geometry of Interaction do this

Option 1 is a standard way to add explicit cuts to proof nets in the sequent + axioms paradigm.
Option 3 (new!): build a tree

\[
\begin{align*}
\Gamma & \vdash A & \quad \Delta & \vdash B \\
\implies & & \quad \Gamma, \Delta & \vdash B
\end{align*}
\]

ICPs as nodes

An ICP over a sequent \(A_1, \ldots, A_n \vdash B\) becomes a node with premises/inputs \(A_1\) through \(A_n\) and conclusion/output \(B\).
We can apply the following rules to build trees.

\[
\Gamma \vdash M : A \quad x : B, \Delta \vdash N : C
\]
\[
\Gamma, f : A \supset B, \Delta \vdash N[f M/x] : C
\]

Building trees:

\[
\Gamma \vdash M : B
\]

\[
\Gamma, x : A \vdash M : B
\]

\[
\Gamma, x : A, y : A \vdash M : B
\]

\[
\Gamma, x : A \vdash M[x/y] : B
\]

\[
\Gamma \vdash M : A \quad x : A, \Delta \vdash N : B
\]

\[
\Gamma, \Delta \vdash N[M/x] : B
\]
Theorem

ICP reduction is confluent and strongly normalizing.
Observations

- ICP reduction is sequent calculus cut-elimination, but
  - without permutations
  - with all cuts at top level
- Contraction on ICP trees requires graphs
- Abstraction on ICP trees is lambda-lifting
- ICP reduction is closed reduction (in $\lambda$-calculus: a redex $(\lambda x. M)N$ may only be reduced if it has no free variables)
- Reduction uses only sub-ICPs of those in the original tree