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 (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?

Two proofs are the same iff they have the same normal form.
- propositional logic: exponential blow-up
- predicate logic: elementary blow-up
- Identify a proof on an A4-page with a proof of the size of the universe corresponds to removing lemmas from a proof.
- But lemmas are important in mathematical proofs.
- Example: Normalizing Fürstenberg’s proof of the infinity of primes yields Euklid’s proof

\[
\begin{align*}
\vdash & \Gamma, A, B, C \quad \vdash & D, \Delta & \iff \quad \vdash & \Gamma, A, B, C \land D, \Delta \quad \vdash & \Gamma, A \lor B, C \land D, \Delta & (1) \\
\vdash & \Gamma, C \quad \vdash & D, \Delta & \iff \quad \vdash & \Gamma, C \land D, \Delta \quad \vdash & \Gamma, A \lor B, C \land D, \Delta & \text{weak} & (2) \\
\vdash & \Gamma, A, B, C \quad \vdash & \Gamma, A, B, D & \iff \quad \vdash & \Gamma, A, B, C \quad \vdash & \Gamma, A, B, D & \iff \quad \vdash & \Gamma, A \lor B, C \land D & (3)
\end{align*}
\]
This is a semantic tableau, a natural deduction proof, and a Coq script, all proving the same formula. Are these proofs “the same”?

- 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**: \( A ::= a | \overline{a} | A \land B | A \lor B \)

- **Negation**: \( \overline{a} = a \) \( A \land B = \overline{A} \lor \overline{B} \) \( A \lor B = \overline{A} \land \overline{B} \)

- **Implication**: \( A \supset B = \overline{A} \lor B \)

- **Sequents**: \( \Gamma ::= A_1, A_2, \ldots, A_k \)

- **Inference rules (sequent calculus)**:
  
  \[
  \begin{align*}
  \text{ax} & \quad \vdash a, \overline{a} \quad \lor \quad \vdash \Gamma, A, B \quad \land \quad \vdash \Gamma, A \vdash \Delta, B \\
  \text{cont} & \quad \vdash \Gamma, A, A \quad \text{weak} \quad \vdash \Gamma, A \\
  \end{align*}
  \]


First-Order Classical Logic (FOL) — Formulas

- **Terms**: \( t ::= x | f(t_1, \ldots, t_n) \)

- **Atoms**: \( a ::= p(t_1, \ldots, t_m) | \overline{p}(t_1, \ldots, t_m) \)

- **Formulas**: \( A ::= a | A \land A | A \lor A | \exists x A | \forall x A \)

- **Sequents**: \( \Gamma ::= A_1, A_2, \ldots, A_k \)

- **Negation**: \( \overline{a} = a \) \( p(t_1, \ldots, t_m) = \overline{p}(t_1, \ldots, t_m) \) \( \overline{p}(t_1, \ldots, t_m) = p(t_1, \ldots, t_m) \) \( A \land B = \overline{A} \lor \overline{B} \) \( A \lor B = \overline{A} \land \overline{B} \) \( A \land B = \overline{A} \lor \overline{B} \) \( A \lor B = \overline{A} \land \overline{B} \)


First-Order Classical Logic (FOL) — Sequent Calculus

- **Inference rules (sequent calculus)**:
  
  \[
  \begin{align*}
  \text{ax} & \quad \vdash a, \overline{a} \quad \lor \quad \vdash \Gamma, A, B \quad \land \quad \vdash \Gamma, A \vdash \Delta, B \\
  \text{cont} & \quad \vdash \Gamma, A, A \quad \text{weak} \quad \vdash \Gamma, A \\
  \end{align*}
  \]

  \( \exists \vdash \Gamma, A[x/t] \) \( \forall \vdash \Gamma, A \) \( \exists x A \vdash \Gamma, \exists x A \) \( x \) not free in \( \Gamma \)

- **Exercise 1.1**: Prove Pierce’s law \((a \supset b) \supset a\) in this sequent calculus.

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

\[ \begin{array}{c|c}
\text{multiplicative} & \text{additive} \\
\hline
\vdash \Gamma, A, B & \vdash \Gamma, A \lor B \\
\hline
\downarrow & \downarrow \\
\vdash \Gamma, A & \vdash \Gamma, A \lor B \\
\hline
\& \vdash \Gamma, A, B & \& \vdash \Gamma, A \otimes B \\
\hline
\wedge \vdash \Gamma, A & \vdash \Gamma, \Delta \wedge B \\
\hline
\otimes \vdash \Gamma, A & \vdash \Gamma, \Delta \otimes B \\
\end{array} \]

Multiplicative Linear Logic (MLL)

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

\[ \begin{array}{c|c|c}
\text{ax} & \vdash a, a^\perp & \vdash \Gamma, A \otimes B \\
\hline
\otimes & \vdash \Gamma, A, B & \vdash \Gamma, \Delta, A \otimes B \\
\wedge & \vdash \Gamma, A & \vdash \Gamma, \Delta \wedge B \\
\otimes & \vdash \Gamma, A & \vdash \Gamma, \Delta \otimes B \\
\end{array} \]

Multiplicative Additive Linear Logic (MALL)

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

\[ \begin{array}{c|c|c}
\text{ax} & \vdash a, a^\perp & \vdash \Gamma, A \otimes B \\
\hline
\otimes & \vdash \Gamma, A, B & \vdash \Gamma, \Delta, A \otimes B \\
\wedge & \vdash \Gamma, A & \vdash \Gamma, \Delta \wedge B \\
\oplus & \vdash \Gamma, A & \vdash \Gamma, \Delta \oplus B \\
\otimes & \vdash \Gamma, A & \vdash \Gamma, A \oplus B \\
\end{array} \]

- 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 \( A \& B \) imply each other, and similarly for \( A \oplus B \) and \( A \& B \). 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.

- **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**: \( A ::= a | a^\perp | A \& B | A \oplus B \)
- **Negation**: \( a^\perp \perp = a \) \( (A \& B)^\perp = A^\perp \& B^\perp \), \( (A \oplus B)^\perp = A^\perp \oplus B^\perp \)
- **Sequents**: \( \Gamma ::= A_1, A_2, \ldots, A_k \)
- **Inference rules (sequent calculus)**:
  
  \[
  \begin{array}{c}
  \text{ax} \\
  \hline
  \Gamma, a \perp \vdash a, a^\perp \\
  \Gamma, a \perp \vdash \Gamma, a^\perp \\
  \Gamma, a^\perp \vdash \Gamma, a \perp
  \end{array}
  \]

Intuitionistic Propositional Logic (IPL)

- **Formulas**: \( A ::= a | A \land B | A \implies B \)
- **Sequents**: \( A_1, A_2, \ldots, A_k \vdash B \)
- **Inference rules (sequent calculus)**:
  
  \[
  \begin{array}{c}
  \text{ax} \\
  \hline
  a \vdash a \\
  \Gamma, A \land B \vdash \Gamma, C \\
  \Gamma, A, B \vdash \Gamma, C \\
  \Gamma, A \implies B \vdash \Gamma, B \\
  \Gamma, A \land B \vdash \Gamma, C \\
  \end{array}
  \]

Intuitionistic Multiplicative Linear Logic (IMLL)

- **Formulas**: \( A ::= a | A \otimes B | A \multimap B \)
- **Sequents**: \( A_1, A_2, \ldots, A_k \vdash B \)
- **Inference rules (sequent calculus)**:
  
  \[
  \begin{array}{c}
  \text{ax} \\
  \hline
  a \vdash a \\
  \Gamma, A, B \vdash \Gamma, C \\
  \Gamma, A, B \vdash \Gamma, C \\
  \end{array}
  \]

- 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?

- In this course we only consider the disjunction-free fragment of intuitionistic logic (\( \lor \) and \( \bot \) are absent).
- Note that implication is now a primitive.
- Negation can be recovered by fixing a fresh propositional variable \( a_0 \) and define \( \neg A = A \multimap a_0 \).
- We need two-sided sequents, and there is exactly one formula on the right.
- Each logical connective has a left-rule and a right-rule.

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

- Recall that in classical logic we can encode \( A \implies B \) as \( \overline{A} \lor B \). So, every formula in IPL can be seen as a formula in CPL.
- In the same way we have in linear logic that \( A \multimap B \) is the same as \( A^\perp \otimes B \). Hence, every formula of IMLL is also a formula of MLL.
- **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 \( \lambda \)-terms are (isomorphic to) intuitionistic natural-deduction proofs

\[
\begin{align*}
\text{[x : A]}^x & \quad \vdash \lambda x. M : A \supset B \\
\vdash \lambda x. M : A \supset B & \quad \vdash \lambda x. M \vdash \lambda x. N : B \\
\end{align*}
\]

- This includes pairs/products/conjunction

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

The Curry–Howard isomorphism (or correspondence where it is less strong) connects proof theory and computation. It makes proof-theoretical constructions relevant to typed functional programming, where they may convey guarantees of termination, safety, or other good properties.

- Exercise 2.1: One case of the definition of (typed) \( \lambda \)-terms is missing: the variable case. What should it look like?

- Exercise 2.2: The isomorphism also includes sums (or coproduct, or disjunction). What are the term constructors and typing rules for it?

Beta-reduction

- For implication:

\[
\begin{align*}
\text{[x : A]}^x & \quad \vdash \lambda x. M : A \supset B \\
\vdash \lambda x. M : A \supset B & \quad \vdash \lambda x. M \vdash \lambda x. N : B \\
\end{align*}
\]

- For conjunction:

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

- (\( \lor / \land \))–Intuitionistic natural deduction is ideal:
  - Reduction is confluent and strongly normalizing
  - Normal forms represent the meaning of a term

The correspondence relates formulae to types, proofs to programs, and here, normalization to computation (\( \beta \)-reduction).

- Exercise 2.3: Give also the reductions for disjunction.

- Exercise 2.4: The rules for disjunction give permutations, and do not give unique normal forms. Construct an example of this, or find one in the literature.
Sequent calculus

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 (\( \Rightarrow L \)). 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 (\( \text{Ax}, \Rightarrow R, \land R \)) terms and proofs are constructed correspondingly, as they are in natural deduction. But for the left-rules (\( \Rightarrow L, \land L \)), 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 \( A \Rightarrow B \) 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 \( \Rightarrow R \) on the previous slide. Left-rules build up a proof at its assumptions, as \( \Rightarrow L \) 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.

- Not an isomorphism between terms and proofs
- Builds terms in normal form (except the cut-rule)

\[ \vdash \]
Permutations

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

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

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 $\land$ and $\lor$)

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:
$$A \拗 B = \overline{A} \lor B$$

But this breaks computational interpretations.

Enter linear logic:
$$A \拗 B = ?A \bot B$$

- $A \bot B$ 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.)
Linear logic

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

Multiplicatives:

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

Additives:

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

Exponentials:

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

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

\[ \vdash \Delta \quad \vdash \Delta \quad \text{Cut} \]

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

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

Is the ? on the context ?Γ necessary here?

The dream of the 90s

New grounds in typed functional programming:

- Linearity gives destructive updates (i.e. mutable state)
- Multiplicatives \(\otimes\) and \(\times\) 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.

\[ \vdash A, A \quad \vdash \Gamma, A \quad \vdash A, B, \Delta \quad \vdash \Gamma, A \times B, \Delta \quad \vdash \Gamma, A \otimes B, \Delta \quad \vdash \Gamma, A \leftrightarrow B, \Delta \quad \vdash \Gamma, A, \Delta \quad \vdash \Gamma, \Delta \quad \text{Cut} \]

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

\[ \vdash a, \overline{a} \quad \vdash b, \overline{b} \quad \vdash c, \overline{c} \]

\[ \vdash a, \overline{a} \otimes b, \overline{b} \otimes c, \overline{c} \]

\[ \vdash a, \overline{a} \otimes b \quad \vdash b, \overline{b} \otimes c \]

\[ \vdash a, \overline{a} \otimes b, \overline{b} \otimes c, \overline{c} \]

Two perspectives

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

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

Correctness

Which proof structures come from proofs?

correct: \[ \begin{array}{c} A \\ \overline{A} \end{array} \] incorrect: \[ \begin{array}{c} A \\ \overline{A} \end{array} \]

All rules except \( \& \) create connected acyclic graphs

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

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

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.

**Exercise 2.12:** Give also the graph–of–rules definition.

**Exercise 2.13:** Prove that the two definitions are isomorphic.

---

**Example**

```
\[ a \otimes b \quad \& \quad c \otimes c \quad a \& d \otimes d \]
```

---

**Contractibility**

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

```
   \[
   \begin{array}{c}
   \vdots \\
   \downarrow \\
   \vdots \\
   \end{array}
   \]
```

3. Correct $\iff$ contracts to a single point

Implemented in linear time via union–find.

---
**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.

---

The proofs of these theorems are a little tricky, but not infeasible. You can attempt them yourself.


3. Lecture
Cographs and Handsome Proof Nets

Willem Heijltjes and Lutz Straßburger

Graphs (directed and undirected)

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

  Example:

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

- A directed graph is a pair $G = (V_G, E_G)$ 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:

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

Examples

undirected graph

\[
\begin{align*}
  V &= \{u, v, w, z\} \\
  E &= \{uw, uv, vw, zw\}
\end{align*}
\]

directed graph

\[
\begin{align*}
  V &= \{u, v, w, z\} \\
  E &= \{u \rightarrow u, u \rightarrow v, v \rightarrow u, v \rightarrow w, v \rightarrow z, z \rightarrow z\}
\end{align*}
\]
Graph Homomorphisms and Isomorphisms

- A graph homomorphism \( h : \mathcal{G} \to \mathcal{H} \) is a map \( h : V_\mathcal{G} \to V_\mathcal{H} \) such that for all \( x, y \in V_\mathcal{G} \) we have that \( xy \in E_\mathcal{G} \) implies \( h(x)h(y) \in E_\mathcal{H} \).
- A graph isomorphism \( h : \mathcal{G} \to \mathcal{H} \) is a bijection \( h : V_\mathcal{G} \to V_\mathcal{H} \) such that \( h \) and \( h^{-1} \) are both homomorphisms.
- A directed graph homomorphism \( h : \mathcal{G} \to \mathcal{H} \) is a map \( h : V_\mathcal{G} \to V_\mathcal{H} \) such that for all \( x, y \in V_\mathcal{G} \) we have that \( x \to_y \mathcal{G} \) implies \( h(x) \to_y \mathcal{H} \).
- A directed graph isomorphism \( h : \mathcal{G} \to \mathcal{H} \) is a bijection \( h : V_\mathcal{G} \to V_\mathcal{H} \) such that \( h \) and \( h^{-1} \) are both homomorphisms.

Paths and Cycles

- A path \( p \) in a graph \( \mathcal{G} \) is a sequence of vertices \( x_0, x_1, \ldots, x_n \in V_\mathcal{G} \) such that \( x_0 x_1, x_1 x_2, \ldots, x_{n-1} x_n \in E_\mathcal{G} \). In this the length of \( p \) is \( n \). The path is elementary if all \( x_0, \ldots, x_n \) are pairwise distinct. A path is a cycle if \( x_0 = x_n \). The cycle is elementary if all \( x_1, \ldots, x_n \) are pairwise distinct (i.e., all vertices, except for \( x_0 = x_n \)).

- Similarly for directed graphs.

Subgraphs and Induced Subgraphs

- Let \( \mathcal{H} = \langle V_\mathcal{H}, E_\mathcal{H} \rangle \) and \( \mathcal{G} = \langle V_\mathcal{G}, E_\mathcal{G} \rangle \) be graphs. We say that \( \mathcal{H} \) is a subgraph of \( \mathcal{G} \) if \( V_\mathcal{H} \subseteq V_\mathcal{G} \) and \( E_\mathcal{H} \subseteq E_\mathcal{G} \).
- We say that \( \mathcal{H} \) is an induced subgraph of \( \mathcal{G} \) if additionally \( \forall u, v \in V_\mathcal{H} : uv \in E_\mathcal{G} \iff uv \in E_\mathcal{H} \).
- We say that \( \mathcal{G} \) contains \( \mathcal{H} \) as induced subgraph if there is an injective homomorphism \( f : \mathcal{H} \to \mathcal{G} \) such that \( f(\mathcal{H}) \) is an induced subgraph of \( \mathcal{G} \). If this is not the case, we say that \( \mathcal{G} \) is \( \mathcal{H} \)-free.

- Example: \( \begin{array}{cccc} \bullet & \bullet & & \\
\end{array} \) is a subgraph of \( \begin{array}{cccc} \bullet & & & \\
\end{array} \) (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 \( \begin{array}{cccc} \bullet & & & \\
\end{array} \). 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:**
  \[ G + H = \langle V_G \cup V_H, E_G \cup E_H \rangle \]

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

**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: cograph | cotree
\[ \begin{array}{c}
   c \quad d \\
   a \quad b
\end{array} \quad \begin{array}{c}
   + \\
   \times
\end{array} \]

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

( where \( P_4 \) is the graph \( \bullet -- \bullet -- \bullet -- \bullet \) )

Graphs of Formulas

Mapping \( [] \) from MLL formulas to (labelled) graphs:
\[
[a] = \bullet \quad (\text{single vertex labelled } a) \\
[a^+] = \bullet^+ \quad (\text{single vertex labelled } a^+) \\
[A \otimes B] = [A] + [B] \\
[A \otimes B] = [A] \times [B]
\]

Example: \( \llbracket c \otimes ((a \otimes b) \otimes d) \rrbracket = \begin{array}{c}
   c \\
   a \\
   b \\
   d
\end{array} \)

**Equivalence of formulas:**
\[
A \otimes B \equiv B \otimes A \\
A \otimes B \equiv B \otimes A
\]

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

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

- \( \bullet \quad \bullet \quad \bullet \quad \bullet \)
- \( \bullet \quad \bullet \quad \bullet \quad \bullet \)
- \( \bullet \quad \bullet \quad \bullet \quad \bullet \)

**Exercise 3.6:** Draw the graphs of the following formulas:
\[
\bullet \quad (a \otimes a^+) \otimes (b \otimes b^+) \\
\bullet \quad (a \otimes b^+) \otimes (a^+ \otimes b) \\
\bullet \quad (a \otimes a^+) \otimes (b \otimes b^+) \\
\bullet \quad (a \otimes a^+) \otimes (b \otimes b^+) \\
\bullet \quad (a \otimes a^+) \otimes (b \otimes b^+)
\]

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

- A matching $M$ in a graph $G = \langle V_G, E_G \rangle$ is 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:

```
\[ \bullet \quad \_ \quad \_ \quad \_ \quad \_ \quad \bullet \]
```

```
\[ \bullet \quad \_ \quad \_ \quad \_ \quad \_ \quad \bullet \]
```

**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 $G = \langle V_G, R_G, B_G \rangle$, 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 $G = \langle V_G, R_G, B_G \rangle$, where $\langle V_G, R_G \rangle$ is a cograph.

Examples:

```
\[ \times \quad \_ \quad \_ \quad \_ \quad \_ \quad \_ \]
```

```
\[ \_ \quad \_ \quad \_ \quad \_ \quad \_ \quad \times \]
```

Æ-Paths and Æ-Cycles

- An alternating elementary path (or Æ-path) in an RB-graph $G = \langle V_G, R_G, B_G \rangle$ 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 $G = \langle V_G, R_G, B_G \rangle$ 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 $v_i v_j \in R_G$ (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:** 
- 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:
- **prenet** = graph constructed from the sequent forest $\Gamma$ and axiom edges $\ell$; denoted as $\pi(\Gamma, \ell)$
- **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 $T_{RB}(A)$:

   $T_{RB}(\varnothing)$
   $T_{RB}(A \otimes B)$
   $T_{RB}(A)$
   $T_{RB}(B)$

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

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

Example

Theorem: $\pi(\Gamma, \ell)$ is correct iff $\tau(\Gamma, \ell)$ is $\ae$-acyclic and $\ae$-connected.

In that case $\tau(\Gamma, \ell)$ is called a **(tree-like) RB-proof net**.
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:
   \[
   [a] = \bullet a \quad \text{(single vertex labelled } a) \\
   [a^\perp] = \bullet a^\perp \quad \text{(single vertex labelled } a^\perp) \\
   [A \otimes B] = [A] + [B] \\
   [A \otimes B] = [A] \times [B]
   \]

   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 \( \ell \).

Example

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

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

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

Folding and Unfolding

- 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 \( \ell \).

- Exercise 3.13: Convince yourself of the need of the “chordless” condition. For this, consider the following two sequents:
  \[
  \vdash a \otimes b, (a^\perp \otimes b^\perp) \otimes (c^\perp \otimes d^\perp), c \otimes d \\
  \vdash a \otimes b, (a^\perp \otimes c^\perp) \otimes (b^\perp \otimes d^\perp), c \otimes d
  \]
  Note that for both sequents there is a unique axiom linking. Try to prove both in the sequent calculus for MLL, and draw the handsome RB-prenet for both. Both contain an \( \& \)-cycle, but in only one it is chordless.

- Exercise 3.14: (Difficult) Show the following:
  1. \( \rho(\Gamma, \ell) \) is \( \& \)-acyclic iff \( \tau(\Gamma, \ell) \) is \( \& \)-acyclic.
  2. \( \rho(\Gamma, \ell) \) is \( \& \)-connected iff \( \tau(\Gamma, \ell) \) is \( \& \)-connected.

- Observe that in handsome RB-prenets, \( \& \) -paths and \( \& \) -cycles do have chords.

- Retoré also uses the term critically chorded for a RB-cograph that is \( \& \)-acyclic and \( \& \)-connected. In other words, a handsome proof net is a critically chorded RB-cograph. And every RB-cograph is a handsome prenet.

- The folding and unfolding are performed by small steps that preserve chordless \( \& \) -paths and chordless \( \& \) -cycles. Details can be found here:
  - Christian Retoré: "Handsome proof-nets: R&B-graphs, perfect matchings and series-parallel graphs". Rapport de Recherche RR-3652, INRIA
From Proof Nets to Combinatorial Proofs
–
A New Approach to Hilbert’s 24th Problem

4. Lecture
Deep Inference

Willem Heijltjes and Lutz Straßburger

Deep inference

A method for structuring proofs

\[
\frac{A}{C} ::= a \mid \frac{A_1}{C_1} \times \frac{A_2}{C_2} \mid \frac{A}{B_1} \implies \frac{B_2}{C}
\]

A derivation from \(A\) to \(C\):

- Atom \(a\)
- Horizontal construction with connective \(\ast\)
- 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

\[
\frac{A}{C} ::= a \mid \frac{A_1}{C_1} \times \frac{A_2}{C_2} \mid \frac{A}{B_1} \implies \frac{B_2}{C}
\]

- Connectives: \(\land, \lor, \top, \bot\)
- Invertible rules:

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

- Non-invertible rules:

\[
\begin{align*}
\vdash & \quad (A \lor B) \land C \\
\vdash & \quad A \land A \\
\vdash & \quad A \lor A \\
\vdash & \quad A \land A
\end{align*}
\]
Example: Peirce's law

\[
\top \Rightarrow (\top \vee a) \Rightarrow a
\]

Intuitionistic logic

\[
\begin{array}{c|c|c}
A \land (B \land C) & A \land B & A \\
\hline
(B \land A) & B & A \land A \\
\end{array}
\]

- Connectives: $\Rightarrow, \land, \top$
- Invertible rules:
  \[
  \begin{align*}
  A \land (B \land C) & \Rightarrow (A \land B) \land C \\
  (A \land B) \land C & \Rightarrow A \land B \\
  B & \Rightarrow \top \land A
  \end{align*}
  \]
- Non-invertible rules:
  \[
  \begin{align*}
  B & \Rightarrow A \Rightarrow (B \land A) \\
  (A \Rightarrow B) \land A & \Rightarrow A \land A \\
  A \land A & \Rightarrow \top \\
  \end{align*}
  \]

Example: switch

\[
\begin{array}{c|c|c}
(A \Rightarrow B) \land C & (A \Rightarrow (B \land C)) \land A \\
\hline
((A \Rightarrow (B \land C)) \land A & (A \Rightarrow B) \land C \\
\hline
A \Rightarrow (B \land C) & A \Rightarrow (B \land C)
\end{array}
\]
Multiplicative Linear Logic

\[
A \otimes (B \otimes C) \quad A \otimes B \quad A \quad A \otimes (B \otimes C) \quad A \otimes B \quad A
\]

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

\[
\frac{A}{A \otimes A} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)}
\]

- Non-invertible rules:

\[
\frac{A}{A \otimes A} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)} \quad \frac{A \otimes B}{A \otimes (B \otimes C)}
\]

Example

\[
((a \otimes b) \otimes c) \quad (d \otimes (c \otimes d))
\]

From deep inference to proof nets

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

\[
\frac{A}{B} \quad \frac{B}{A}
\]

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

Inductive translation

\[
\begin{align*}
A &::= a | A_1 \times A_2 | A_1 B_1 \parallel B_2 \parallel C \\
A &| B_1 | r | B_2 | C
\end{align*}
\]
Composition

- Composition is path composition across the common formula

\[
\begin{align*}
\sigma \land \overline{b} & = \sigma \land \overline{b} \\
((\sigma \land \overline{b}) \land b) & = ((\sigma \land \overline{b}) \land b) \\
((\overline{b} \land (\sigma \land c)) & = ((\overline{b} \land (\sigma \land c))
\end{align*}
\]

Rules

\[
\begin{align*}
\frac{I}{A \land A} & \quad \frac{(A \land B) \land C}{A \land (B \land C)} & \quad \frac{A \land \overline{A}}{\bot} \\
A \land A & \quad (A \land B) \land C & \quad A \land (B \land C) \\
A \land (B \land C) & \quad (A \land B) \land C & \quad A \land (B \land C) \\
A \land (B \land C) & \quad A \land B & \quad A \land (B \land C) \\
A \land (B \land C) & \quad A \land B & \quad A \land (B \land C) \\
A \land (B \land C) & \quad \bot & \quad A \land (B \land C) \\
A \land (B \land C) & \quad \bot & \quad A \land (B \land C) \\
A \land (B \land C) & \quad \bot & \quad A \land (B \land C) \\
A \land (B \land C) & \quad \bot & \quad A \land (B \land C) \\
A \land (B \land C) & \quad \bot & \quad A \land (B \land C) \\
A \land (B \land C) & \quad \bot & \quad A \land (B \land C) \\
\end{align*}
\]

Switch

\[
\begin{align*}
(B \land C) \land D & = (B \land (C \land D)) \\
B \land C & = B \land (C \land D) \\
\end{align*}
\]
MLL is the logic of undirected, connected, acyclic networks.

- We can add the mix rule to allow disconnected networks

\[
\frac{A \otimes B}{A \rhd B}^{\text{mix}}
\]

- 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)

\[
\frac{A_1 \otimes \ldots \otimes A_m, C_1 \rhd \ldots \rhd C_m}{f}
\]

From sequent calculus

\[
\frac{\vdash A, A^{Ax}}{} \quad \Rightarrow \quad \frac{}{\vdash A \rhd A}
\]

\[
\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}^{\otimes} \quad \Rightarrow \quad \frac{\vdash (\Gamma A) \otimes (B \rhd \Delta)}{(A \otimes B) \rhd \Delta}^{s}
\]

\[
\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \rhd B}^{\rhd} \quad \Rightarrow \quad \frac{\vdash \Gamma \rhd (A \rhd B)}{\Gamma \rhd (A \rhd B)}^{s}
\]

\[
\frac{\vdash \Gamma, A \quad \vdash \Delta, \Delta}{\vdash \Gamma, \Delta}^{\text{Cut}} \quad \Rightarrow \quad \frac{\vdash (\Gamma A) \otimes (\Delta \rhd \Delta)}{A \otimes (\Delta \rhd \Delta)}^{(s)}
\]

\[
\frac{\vdash \Gamma, A \quad \vdash \Delta, \Delta}{\vdash \Gamma, \Delta}^{\text{Cut}} \quad \Rightarrow \quad \frac{\vdash A \otimes (\Delta \rhd \Delta)}{A \otimes (\Delta \rhd \Delta)}^{(s)}
\]
### Eta-expansion

\[ \vdash A \otimes B, A \Rightarrow B, \text{Ax} \]

\[ \vdash \bar{A} \otimes \bar{B}, (A \Rightarrow B), \text{Ax} \]

\[ \vdash \bar{A} \otimes \bar{B}, A, \text{Ax} \]

\[ \vdash \bar{A} \otimes \bar{B}, B, \text{Ax} \]

\[ \vdash \bar{A} \otimes \bar{B}, A \Rightarrow B \]

### Cut-elimination

\[ \vdash \Gamma, \bar{A} \]

\[ \vdash \Delta, \bar{B}, \text{Cut} \]

\[ \vdash \Theta, A, B, \text{Cut} \]

\[ \vdash \Gamma, \Delta, \bar{A} \]

\[ \vdash \Delta, \Theta, \text{Cut} \]

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

\[ \vdash \bar{A} \otimes \bar{B}, A \Rightarrow B \]

\[ \vdash A, B \]

### Big picture

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


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

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

Sequent calculus:
\[
\begin{align*}
A \vdash A & \quad \text{Ax} & A \vdash C, B \vdash C & \quad \oplus L & A \vdash C_1, C_2 & \quad \oplus R \\
A \vdash B, B \vdash C & \quad \text{Cut} & A \vdash C & \quad \& L & A \vdash A_1, A_2 \vdash C & \quad \& R \\
A \vdash B, A \vdash C & \quad \text{Cut} & \vdash A, B \vdash C & \quad \& R
\end{align*}
\]

Sum and product

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

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

\[
\vdash A, A \quad \vdash A, A + B \\
\vdash A + A + B \quad \vdash A, A + B \\
\vdash A + B \quad \vdash A, A + B \\
\vdash A + B, A + B + R, 1 \\
\vdash A, A \times A \times A \quad \vdash A, A \times A + B \\
\vdash A + A + B + R, 1 \\
\vdash A \times A \times A \quad \vdash A, A + B \times R \\
\vdash A + A \times A + B + R, 1 \\
\vdash A + A \times A + B + R, 1
\]

4/23

Proof nets: (two-sided) sequent + axioms

\[
\vdash A \quad \vdash A \times A \\
\vdash A + B \quad \vdash A + B \\
\vdash A + A + B \quad \vdash A + A + B \\
\vdash A + B \quad \vdash A + B \\
\vdash A + B \quad \vdash A + B \\
\vdash A + B \times R \\
\vdash A + B \times R \\
\vdash A + B \times R \\
\vdash A + B \times R
\]

5/23

\[
\vdash A \quad \vdash A \\
\vdash B \quad \vdash B \\
\vdash C \quad \vdash C \\
\vdash D \quad \vdash D \\
\vdash A \quad \vdash A \\
\vdash B \quad \vdash B \\
\vdash C \quad \vdash C \\
\vdash D \quad \vdash D \\
\vdash A \quad \vdash A \\
\vdash B \quad \vdash B \\
\vdash C \quad \vdash C \\
\vdash D \quad \vdash D
\]

6/23
**Eta-expansion**

\[ A + B \vdash A \vdash A + B \]

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

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

\[ A \vdash A + B \]

\[ A \vdash A + B \]

\[ A \vdash A + B \]

\[ A \vdash A + B \]

\[ A \vdash A + B \]

- Links may be restricted to atoms

**Composition**

- Composition is *relational composition*
- Requires *eta-expansion*

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

\[ b \times a \]

\[ b \times a \]

\[ b \times b \]

\[ b \times b \]

- Give a simple example to show that relational composition requires eta-expanded proof nets.

**Correctness**

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

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

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

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

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

Example

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

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 choose 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 \( B_i \), then \( s \) chooses \( B_i \). 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

\[ \frac{B \vdash D}{B \times C \vdash D} \quad \frac{C \vdash D}{B \times C \vdash D} \quad \frac{B \vdash D}{B + C \vdash D} \quad \frac{B \vdash D \quad C \vdash D}{B \times C \vdash D} \quad \frac{B \vdash D \quad C \vdash D}{B + C \vdash D} \]

\[ \frac{B \times C \vdash D}{B \times C \vdash D} \quad \frac{B \times C \vdash D}{B \times C \vdash D} \quad \frac{B + C \vdash D}{B + C \vdash D} \quad \frac{B + C \vdash D}{B + C \vdash D} \]

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

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

Example again

- Apply coalescence steps to show correctness of this example.

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 \( O(|A| \times |B|) \).
Deep inference: classical logic

\[
\begin{align*}
A \& C &::= a | \quad A_1 \& A_2 | \quad \frac{A}{B_1} &\quad \frac{B_2}{C} \\
\end{align*}
\]

- Connectives: \(\land, \lor, \top, \bot\)
- Invertible rules:

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

- Non-invertible rules:

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

Deep inference: additive linear logic

\[
\begin{align*}
A \& C &::= a | \quad A_1 \& A_2 | \quad \frac{A}{B_1} &\quad \frac{B_2}{C} \\
\end{align*}
\]

- Connectives: \(\times, +, 0, 1\)
- Invertible rules:

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

- Non-invertible rules:

\[
\begin{align*}
0 &\quad A + A &\quad A &\quad A \times A &\quad A \\
}\top &\quad A &\quad A &\quad A \times A &\quad A \\
\end{align*}
\]

Inductive translation

\[
\begin{align*}
A \& C &::= a | \quad A_1 \& A_2 | \quad \frac{A}{B_1} &\quad \frac{B_2}{C} \\
\end{align*}
\]

- \(a\) and \(A_1 \& A_2\) are convertible
- \(a\) and \(C_1 \& C_2\) are convertible
- \(A\) is convertible with \(A_1\) and \(A_2\)
- \(a\) is convertible with \(C_1\) and \(C_2\)
- \(r\) is convertible with \(B_2\)
- \(C\) is convertible with \(B_2\)
Example again again

$$A + (B + C) \vdash A + B \quad \frac{A + B \vdash A}{A + B \vdash \lambda} \quad \frac{A \times (B + C) \vdash A \times B \quad A \times B \vdash A \times A \quad A \times A \vdash 1}{A \times (B + C) \vdash (A \times B) \times C} \quad \frac{A \times (B + C) \vdash A \times B \quad A \times B \vdash A \times A \quad A \times A \vdash 1}{A \times (B + C) \vdash (B \times C) \times A}$$

$$A \vdash A^* \quad B \vdash D \quad C \vdash D \quad B \vdash C \vdash D$$

$$A \vdash B \quad B \vdash C \quad A \vdash C \quad A \vdash B \quad A \vdash C$$

$$A \vdash A^* \quad B \vdash D \quad C \vdash D$$

$$A \vdash B \quad B \vdash C \quad A \vdash C$$

$$A \vdash A^* \quad B \vdash D \quad C \vdash D$$

$$A \vdash B \quad B \vdash C \quad A \vdash C$$
Big picture

Sequent calculus $\rightarrow$ Deep inference

$\Downarrow$

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

H. Hu, Contractible Coherence Spaces and Maximal Maps, ENTCS 20, pp. 309–319, 1999


What are Fibrations?

- originate in topology as a generalization of fiber bundles:

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

- fibrations in topology:
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 \( w' \to_H f(v) \) then there is a unique \( w \in V_G \) with \( w \to_G v \) and \( f(w) = w' \).

![Diagram of directed graph fibrations](image)

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 \( f(w) = w' \).

![Diagram of undirected graph fibrations](image)

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 \( \{w', f(w)\} \notin E_H \).

![Diagram of skew undirected graph fibrations](image)

- 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:
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 \( \{ w', f(w) \} \not\in E_H \).

Examples:

\[
\begin{array}{c}
\text{a} \quad \text{b} \\
\Rightarrow \quad \Rightarrow \\
\text{c} \quad \text{a} \quad \text{b} \\
\text{c} \quad \text{a} \quad \text{b}
\end{array}
\]

\[
\begin{array}{c}
\text{a} \quad \text{b} \\
\Rightarrow \quad \Rightarrow \\
\text{c} \quad \text{a} \quad \text{b} \\
\text{c} \quad \text{a} \quad \text{b}
\end{array}
\]

\[
\begin{array}{c}
\text{a} \quad \text{b} \\
\Rightarrow \quad \Rightarrow \\
\text{c} \quad \text{a} \quad \text{b} \\
\text{c} \quad \text{a} \quad \text{b}
\end{array}
\]

\[
\begin{array}{c}
\text{a} \quad \text{b} \\
\Rightarrow \quad \Rightarrow \\
\text{c} \quad \text{a} \quad \text{b} \\
\text{c} \quad \text{a} \quad \text{b}
\end{array}
\]

\[
\begin{array}{c}
\text{a} \quad \text{b} \\
\Rightarrow \quad \Rightarrow \\
\text{c} \quad \text{a} \quad \text{b} \\
\text{c} \quad \text{a} \quad \text{b}
\end{array}
\]

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

\[
\begin{align*}
\text{SKSg}_\downarrow: \\
& i \downarrow A \quad \vdash T \\
& w \downarrow A \\
& c \downarrow A \\
& s \downarrow C \vee (A \wedge B) \\
& j \downarrow (C \vee A) \wedge B \\
& \end{align*}
\]

Theorem: (Decomposition)

\[
\begin{align*}
\text{SKSg}_\uparrow: \\
& i \uparrow A \\
& w \uparrow A \\
& c \uparrow A \\
& s \uparrow C \vee (A \wedge B) \\
& j \uparrow (C \vee A) \wedge B \\
& \end{align*}
\]

Decomposition for the Down-Fragment (without units)

\[
\begin{align*}
\text{SKSg'}_\downarrow: \\
& i \downarrow A \quad \vdash B \wedge (A \vee A) \\
& w \downarrow B \quad \vdash B \vee A \\
& c \downarrow A \vee A \\
& s \downarrow C \vee (A \wedge B) \\
& j \downarrow (C \vee A) \wedge B \\
& \end{align*}
\]

Theorem: (Decomposition)

\[
\begin{align*}
\text{SKSg'}_\uparrow: \\
& i \uparrow A \\
& w \uparrow A \\
& c \uparrow A \\
& s \uparrow C \vee (A \wedge B) \\
& j \uparrow (C \vee A) \wedge B \\
& \end{align*}
\]

First Skew Fibration Theorem

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

Then there is a skew fibration \( f: \llbracket A \rrbracket \rightarrow \llbracket B \rrbracket \) iff \( \llbracket (c \downarrow w \downarrow) \rrbracket \rightarrow \llbracket A \rrbracket \rightarrow \llbracket B \rrbracket \)

Examples:

\[
\begin{align*}
& a \rightarrow b \quad a \rightarrow b \\
& a \rightarrow a \quad b \rightarrow b \\
& a \rightarrow a \quad a \rightarrow b \\
& a \rightarrow b \\
& c \rightarrow b \\
& (a \wedge b) \vee (a \wedge b) \vee ((a \vee b) \wedge (a \vee b)) \\
& (a \wedge b) \vee (a \wedge b) \vee ((a \vee b) \wedge (a \vee b)) \vee c \\
& (a \wedge b) \vee ((a \vee b) \wedge (a \vee b)) \vee c \\
& \end{align*}
\]
Exercise 6.1: Show that every skew fibration defines an ALL proof net.

Exercise 6.2: (Difficult) Prove the 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 $f: \llbracket C \rrbracket \rightarrow \llbracket B \rrbracket$ and $g: \llbracket \overline{C} \rrbracket \rightarrow \llbracket \overline{A} \rrbracket$. 
Summary (Multiplicative Part)

sequent calculus:
\[
\frac{}{\Gamma, A, B} \quad \frac{}{\Gamma, A \otimes B} \\
\frac{}{\Gamma, A} \quad \frac{}{\Gamma, B} \\
\frac{}{\Gamma, A, B} \\
\frac{}{\Gamma, A \otimes B}
\]

deep inference:
\[
\frac{}{\Gamma, A, B} \quad \frac{}{\Gamma, A \otimes B} \\
\frac{}{\Gamma, A} \quad \frac{}{\Gamma, B} \\
\frac{}{\Gamma, A, B} \\
\frac{}{\Gamma, A \otimes B}
\]

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

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

Summary (Additive Part)

sequent calculus:
\[
\frac{}{\Gamma, A, B} \quad \frac{}{\Gamma, A \otimes B} \\
\frac{}{\Gamma, A} \quad \frac{}{\Gamma, B} \\
\frac{}{\Gamma, A, B} \\
\frac{}{\Gamma, A \otimes B}
\]

deep inference:
\[
\frac{}{\Gamma, A, B} \quad \frac{}{\Gamma, A \otimes B} \\
\frac{}{\Gamma, A} \quad \frac{}{\Gamma, B} \\
\frac{}{\Gamma, A, B} \\
\frac{}{\Gamma, A \otimes B}
\]

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

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

Handsome proof net
Cograph
Skew fibration
Conclusion

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 $A \not\approx \overline{A}$
- Distributivity gives canonical normal forms $\implies$ collapse

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

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

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

Lafont’s examples: non-confluence

\[ \frac{\vdash \Gamma, \Delta^w}{\vdash \Gamma, \Delta} \leftrightarrow \frac{\vdash \Gamma, A^w \quad \vdash \overline{A}, \Delta}{\vdash \Gamma, \Delta} \quad \text{Cut} \]

Lafont’s examples: non-termination

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

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

- 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 $A \vdash B = A \lor B$

Intuitionistic natural deduction proves $A \vdash (A \vdash \bot) \vdash \bot$, and extends to classical natural deduction by including a rule $\bot E$:

$$
\begin{array}{c}
A \\
\bot E
\end{array}
$$

which proves $(A \vdash \bot) \vdash \bot \vdash A$ (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 $\top$ 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.

**Example: distributivity**

\[
a \land (b \lor c) \implies (a \land b) \lor (a \land c)
\]
The translation from sequent proofs is non-deterministic (unless we allow mix in our handsome proof nets)
The case conjunction–contraction:

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

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


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

Negation:
$\top = 0$
$\bot = 1$
$\overline{\rho(t_1, \ldots, t_n)} = \rho(t_1, \ldots, t_n)$
$\overline{\rho(t_1, \ldots, t_n)} = \rho(t_1, \ldots, t_n)$
$\exists x.A = \forall x.\overline{A}$
$A \land B = A \lor B$
$\forall x.A = \exists x.\overline{A}$
$A \lor B = A \land B$

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 \( \cdot \) from formulas to (labelled) graphs:

\[ a \rightarrow \bullet \]

\[ J \cdot (A \lor B) \rightarrow J \cdot A \lor J \cdot B \]

\[ J \cdot (\exists x. A) \rightarrow \exists x \cdot [A] \]

Example:

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

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

Graphs of Formulas

Congruence Relation \( \equiv \) on Formulas:

\[ A \lor B \equiv B \lor A \]

\[ A \land B \equiv B \land A \]

\[ \forall x. \forall y. A \equiv \forall y. \forall x. A \]

\[ \exists x. \exists y. A \equiv \exists y. \exists x. A \]

\[ \forall x. (A \lor B) \equiv (\forall x. A) \lor B \]

\[ \exists x. (A \land B) \equiv (\exists x. A) \land B \]

(x not free in \(B\))

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

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

Example:

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

First-Order Combinatorial Proofs

Definition:

A \textit{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:** 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 the 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:**

\[
\begin{array}{c}
x \quad px \\
y \quad \neg y \\
\end{array} \quad z
\]

\[
\begin{array}{c}
x \quad px \\
y \quad \neg y \\
\end{array} \quad z
\]

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 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:**

\[
\begin{array}{c}
x \quad px \\
y \quad \neg y \\
\end{array} \quad z
\]

\[
\begin{array}{c}
x \quad px \\
y \quad \neg y \\
\end{array} \quad z
\]

linked fograph dualizer leap graph

**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.
Skew Bifibrations

Definition: Let $G$ and $H$ be fographs. A skew bifibration $\phi : G \rightarrow H$ 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 $\phi : C \rightarrow G$ where $C$ is a fonet and $G$ is the graph of the formula $A$.

Examples:

More Compact Notation:

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

Exercise 8.2:
Show that the combinatorial proofs for classical propositional logic are indeed a special case of the first-order combinatorial proofs.

Exercise 8.3:
Are there sequent calculus derivations corresponding to the example of combinatorial proofs on the previous slide? If yes, give them.
First-Order Deep Inference System

First-Order Linear Proofs

Exercise 8.4: Give the deep inference derivations for the four combinatorial proofs above.
Resource Management in First-Order Proofs

**Theorem:**

\[
\begin{align*}
A & \vdash A' & [A] \\
\{m, m_2, m_3, =\} & \models B' & [B] \\
A & \models \{\alpha, \alpha, \gamma\} & \leftarrow \text{skew bifibration}
\end{align*}
\]

\[
\begin{align*}
\frac{A \land C \lor (B \land D) \quad (A \lor B) \land (C \lor D)}{A \lor V (\forall x. A) \lor (\forall x. B)}
\end{align*}
\]

\[
\begin{align*}
\frac{\forall x. A \lor (\exists x. B)}{\exists x. (A \lor B)}
\end{align*}
\]

\[
\begin{align*}
\forall \exists \land \lor (\forall x. A) \lor (\forall x. B)
\end{align*}
\]

**Example**

\[
\begin{align*}
\frac{\alpha}{x} \quad \frac{\beta}{y} & \quad \frac{\gamma}{z} \\
\alpha & \lor \beta & \lor \gamma
\end{align*}
\]

\[
\begin{align*}
\frac{\forall \exists \land \lor (\forall x. A) \lor (\forall x. B)}{\exists x. (A \lor B)}
\end{align*}
\]

\[
\begin{align*}
\frac{\forall \exists \land \lor (\forall x. A) \lor (\forall x. B)}{\exists x. (A \lor B)}
\end{align*}
\]

**Example**

\[
\begin{align*}
\frac{\alpha}{qab} \quad \frac{\beta}{qba} \quad \frac{\gamma}{xy}
\end{align*}
\]

\[
\begin{align*}
\frac{\forall \exists \land \lor (\forall x. A) \lor (\forall x. B)}{\exists x. (A \lor B)}
\end{align*}
\]

\[
\begin{align*}
\frac{\forall \exists \land \lor (\forall x. A) \lor (\forall x. B)}{\exists x. (A \lor B)}
\end{align*}
\]
Example (Drinker’s Formula)

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 drinking”: $\exists x (px \supset \forall y py)$.

That formula is interesting for several reasons:

- 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.

\[
\exists x (px \supset \forall y py)
\]
A Last Word on Classical Combinatorial Proofs

**Definition:**
A *combinatorial proof* of a formula $A$ is a skew fibration $f : C \rightarrow [A]$ 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:**

```
\[ \begin{array}{ccc}
  & a & \\
  b & \rightarrow & a \\
  & a & \\
\end{array} \quad \begin{array}{ccc}
  \langle (a \lor b) \land \lnot a \rangle & \lor & a \\
  (a \lor b) \land \lnot a & \lor & a \\
  \lnot (a \lor b) \land \lnot a & \lor & a \\
\end{array} \quad \begin{array}{ccc}
  \langle (a \lor b) \land \lnot a \rangle & \lor & a \\
  (a \lor b) \land \lnot a & \lor & a \\
  \lnot (a \lor b) \land \lnot a & \lor & a \\
\end{array} \]
```

Another Last Word on Classical Combinatorial Proofs

```
\[ \begin{array}{ccc}
  & c \lor a \\
  \lnot a \land c, c \land a, b \lor c, (b \lor b) \lor a \\
  (c \lor a) \lor b \lor c, (b \lor b) \lor a \\
\end{array} \quad \begin{array}{ccc}
  & c \lor a \\
  \lnot a \land c, c \land a, b \lor c, (b \lor b) \lor a \\
  (c \lor a) \lor b \lor c, (b \lor b) \lor a \\
\end{array} \]
```

In different publications and also in different lectures in this course, 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.

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)

\[
\begin{align*}
2 \cdot a & \quad \iff \quad ((a \circ (a \circ (y \circ (y \circ (y \circ y)))))
\end{align*}
\]

From Classical to Intuitionistic Combinatorial Proofs

\[
((a \circ a) \circ b) \circ (b \circ b)
\]

cograph = classical logic formula
arena = intuitionistic logic formula

Arenas

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

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

- **Σ-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:**
  \[
  G \quad H \quad G \quad H \\
  G + H \quad G \uplus H
  \]

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

- **Translating formulas to dags:**
  \[
  [P] = \bullet \quad [A \land B] = [A] + [B] \quad [A \supset B] = [A] \uplus [B]
  \]

**Example:**
\[
((P \supset P) \supset Q) \land Q \supset R \land ((S \supset S) \supset S)
\]

From Formulas to Arenas

**Example:**
\[
((P \supset P) \supset Q) \land Q \supset R \land ((S \supset S) \supset S)
\]

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

**Theorem:**

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

where \(\equiv\) is generated from

\[
(A \land B) \supset C \equiv A \supset (B \supset C) \quad \quad \quad A \land (B \land C) \equiv (A \land B) \land C \quad \quad \quad A \land B \equiv B \land A
\]

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:**
\[
P^* \rightarrow P^* \rightarrow Q^* \rightarrow Q^*
\]

**link graph (of a linked arena):**
remove all edges into even vertices
add a link edge for each link
(from odd to even)

**Example:**
\[
P^* \rightarrow Q^* \rightarrow Q^*
\]
Arena Nets

**Example:**

- **linked arena**
  - $P \bullet P \rightarrow Q \bullet Q \rightarrow R \bullet R \rightarrow S \bullet S$
- **link graph**
  - $P \rightarrow P \rightarrow Q \rightarrow Q \rightarrow R \rightarrow R \rightarrow S \rightarrow S$

**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:**

- $((P \multimap P) \multimap Q) \multimap R \multimap Q \multimap ((R \multimap S) \multimap S)$

Recall: IMLL stands for intuitionistic multiplicative linear logic.

This theorem is due to
- Willem Heijltjes, Dominic Hughes and Lutz Straßburger: "Intuitionistic Proofs without Syntax". LICS 2019
- Its proof shows the equivalence between the arena correctness criterion and the essential net criterion due to François Lamarche:
Skew fibrations

\[ \forall v \land w \quad \text{iff} \quad v \land w \quad \text{are distinct and meet at odd depth or do not meet at all} \]

\[ \forall v \land w \quad \text{otherwise} \]

Example:

\[ \begin{align*}
  u^v & \rightarrow w^v \rightarrow y^v \\
  u \land v & \quad w \land x \\
  v^x \rightarrow x^z & \rightarrow z^z \\
  w \land w & \quad x \land y
\end{align*} \]

Definition: A skew fibration \( f : G \rightarrow H \), of an arena \( G \) over an arena \( H \), is a function \( f : V_G \rightarrow V_H \) that preserves

- edges: \( v \rightarrow_G w \) implies \( f(v) \rightarrow_H f(w) \)
- equivalence: \( v \sim_G w \) implies \( f(v) \sim_H f(w) \)
- roots: \( r \rightarrow_G \) implies \( f(r) \rightarrow_H \)
- conjuncts: \( v \land_G w \) implies \( f(v) \land_H f(w) \)

and satisfies the following skew lifting condition:

- if \( f(v) \land_H w \) then there exists \( u \) with \( v \leftarrow_G u \) and \( f(u) \leftarrow_H w \).

Intuitionistic combinatorial proofs

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

Examples:

From Sequent Calculus to ICP

\[ \begin{align*}
  \vdash P & \quad \vdash P \\
  \vdash (P \Rightarrow P) & \quad \vdash Q \\
  \vdash (P \Rightarrow P) \Rightarrow Q & \quad \vdash Q \land Q
\end{align*} \]

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

Theorem: This translation is surjective.

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

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

\[
\begin{align*}
\Gamma, \Delta \vdash x : P & \quad \vdash \lambda x : P \\
\Gamma, \Delta \vdash (P \vdash P) & \quad \vdash \lambda y : Q \vdash P \\
\Gamma, \Delta \vdash (P \vdash Q) : Q & \quad \vdash \lambda y : (Q \vdash Q) \\
\end{align*}
\]

\[
\begin{align*}
\Gamma, \Delta \vdash z : P & \quad \vdash \lambda z : P \\
\Gamma, \Delta \vdash (P \vdash P) & \quad \vdash \lambda y : Q \vdash P \\
\Gamma, \Delta \vdash (P \vdash Q) : Q & \quad \vdash \lambda y : (Q \vdash Q) \\
\end{align*}
\]

Translating sequent proofs to \(\lambda\)-terms can cause exponential blow-up!

ICPs form a semantics of proofs that respects complexity.

Sequent Calculus Rule Permutations

\[
\begin{align*}
\text{B, } \Delta, \Gamma, C & \vdash D \\
\Gamma, A & \vdash B, \Delta \vdash C \vdash D & \Rightarrow & \Gamma, A & \vdash B, \Delta, C \vdash D & \checkmark \\
\Gamma, A & \vdash B, \Delta \vdash C \vdash D & \Rightarrow & \Gamma, A & \vdash B, \Delta, C \vdash D & \checkmark \\
\end{align*}
\]

\[
\begin{align*}
\text{\(\Delta \vdash C\)} \quad \text{weak} \\
\Gamma, A & \vdash B, \Delta \vdash C & \Rightarrow & \Delta & \vdash C & \checkmark \\
\Gamma, A & \vdash B, \Delta \vdash C & \Rightarrow & \Gamma, A & \vdash B, \Delta \vdash C & \checkmark \\
\end{align*}
\]

\[
\begin{align*}
\text{\(\Gamma, A \vdash B, \Delta \vdash C\)} \quad \text{cont} \\
\Gamma, A & \vdash B, \Delta \vdash C & \Rightarrow & \Gamma, A & \vdash B, \Delta \vdash C & \checkmark \\
\Gamma, A & \vdash B, \Delta \vdash C & \Rightarrow & \Gamma, A & \vdash B, \Delta \vdash C & \checkmark \\
\end{align*}
\]

\[
\begin{align*}
\text{Exercise 9.2: How many possible rule permutations are there?}
\end{align*}
\]

Comonoid Transformations

\[
\begin{align*}
\text{\(\Gamma, A, B, C \vdash C\)} \quad \text{cont} \\
\Gamma, A & \vdash B, C & \Rightarrow & \Gamma, A & \vdash B, A, B & \vdash C & \checkmark \\
\Gamma, A & \vdash B, C & \Rightarrow & \Gamma, A & \vdash B, A, B & \vdash C & \checkmark \\
\end{align*}
\]

\[
\begin{align*}
\text{\(\Gamma, A \vdash C\)} \quad \text{weak} \\
\Gamma, A & \vdash B, C & \Rightarrow & \Gamma, A & \vdash B, A, B & \vdash C & \checkmark \\
\Gamma, A & \vdash B, C & \Rightarrow & \Gamma, A & \vdash B, A, B & \vdash C & \checkmark \\
\end{align*}
\]

\[
\begin{align*}
\text{\(\Gamma, A \vdash B\)} \quad \text{weak} \\
\Gamma, A & \vdash B & \Rightarrow & \Gamma, A & \vdash B & \checkmark \\
\Gamma, A & \vdash B & \Rightarrow & \Gamma, A & \vdash B & \checkmark \\
\end{align*}
\]

\[\text{16/20} \quad \text{17/20} \quad \text{18/20} \quad \text{19/20} \]
<table>
<thead>
<tr>
<th>Transformation</th>
<th>Equivalence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Γ, A,B ⊢ C</td>
<td>Γ, A,B ⊢ C</td>
</tr>
<tr>
<td>Γ, A ∧ B ⊢ C</td>
<td>Γ, A,B ⊢ C</td>
</tr>
<tr>
<td>Γ ⊢ A</td>
<td>Δ ⊢ B</td>
</tr>
<tr>
<td>Γ, Δ ⊢ A ∧ B</td>
<td>Γ, Δ ⊢ B ∧ A</td>
</tr>
<tr>
<td>Γ, A,B,C ⊢ D</td>
<td>Γ, A,B,C ⊢ D</td>
</tr>
<tr>
<td>Γ, A ∧ B,C ⊢ D</td>
<td>Γ, A,B,C ⊢ D</td>
</tr>
<tr>
<td>Γ ⊢ A</td>
<td>Δ ⊢ B</td>
</tr>
<tr>
<td>Γ, Δ ⊢ A ∧ B ∧ C</td>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
</tr>
<tr>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
</tr>
<tr>
<td>Γ ⊢ A</td>
<td>Δ ⊢ B</td>
</tr>
<tr>
<td>Γ, Δ ⊢ A ∧ B ∧ C</td>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
</tr>
<tr>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
</tr>
<tr>
<td>Γ ⊢ A</td>
<td>Δ ⊢ B</td>
</tr>
<tr>
<td>Γ, Δ ⊢ A ∧ B ∧ C</td>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
</tr>
<tr>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
<td>Γ, Δ,Δ ⊢ A ∧ B ∧ C</td>
</tr>
</tbody>
</table>

Formula-isomorphism transformations
Intuitionistic composition

\[ \Gamma \vdash A, \Delta \vdash B \to \Gamma, \Delta \vdash B \]

For intuitionistic combinatorial proofs

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

\[ \Gamma \vdash A, \Delta \vdash B \to \Gamma, A \triangleright A, \Delta \vdash B \]

- 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*)

**Option 2:** compute the result directly

\[ \Gamma \vdash A, \Delta \vdash B \to \Gamma, \Delta \vdash B \]

- 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.

*Exercise 10.1:* Add an explicit cut-connective to MLL proof nets in this way, and give the cut-elimination steps.
Option 3 (new!): build a tree

\[ \Gamma \vdash A \quad + \quad A, \Delta \vdash B \quad \Rightarrow \quad \Gamma, \Delta \vdash B \]

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

Building trees

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

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