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?
   \( \Rightarrow \) define proofs independently from the proof systems

2. When are two proofs the same?
   \( \Rightarrow \) define a notion of proof identity
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

When are two proofs the same?

- Normalization?
- Curry-Howard-Correspondence
  - formulas = types
  - proofs = programs
  - normalization = computation

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

- Rule permutation?

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?

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

Are these proofs “the same”?

Combining proof identity

Two proofs are the same if they correspond to the same combinatorial proof.

When are two proofs (in normal form) 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
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:
\[
\begin{align*}
\overline{a} &= a \\
\overline{p(t_1, \ldots , t_m)} &= \overline{p}(t_1, \ldots , t_m) \\
\overline{A \land B} &= \overline{A} \lor \overline{B} \\
\overline{A \lor B} &= \overline{A} \land \overline{B} \\
\overline{\exists x.A} &= \forall x.\overline{A} \\
\overline{\forall x.A} &= \exists x.\overline{A}
\end{align*}
\]

We are using formulas in \textit{negation normal form} (nnf) because we need only half as many inference rules.

As in the propositional case, we consider formulas in nfv.

We do not discuss semantics in this course.

First-Order Classical Logic (FOL) — Sequent Calculus

In this course, we do not discuss the cut rule to the propositional system.

We also do not discuss the cut rule.

As in the propositional case, we consider formulas in NNF.

We do not discuss semantics in this course.

We don’t have the constants \textit{truth} and \textit{falsum} in the language to avoid unnecessary complications in later parts of the course. Note that in classical logic we can pick a fresh propositional variable \( \varphi_0 \) and define \textit{truth} as \( \varphi_0 \lor \overline{\varphi_0} \) and \textit{falsum} as \( \varphi_0 \land \overline{\varphi_0} \).

We don’t have the constants \textit{truth} and \textit{falsum} in the language to avoid unnecessary complications in later parts of the course. Note that in classical logic we can pick a fresh propositional variable \( \varphi_0 \) and define \textit{truth} as \( \varphi_0 \lor \overline{\varphi_0} \) and \textit{falsum} as \( \varphi_0 \land \overline{\varphi_0} \).

\( \bullet \) Sequents are multisets of formulas. The meaning is the disjunction of the formulas.

\( \bullet \) In this course, we always consider the comma in the sequent notation to be associative and commutative.

\( \bullet \) Sequent derivations are formally trees whose nodes are the instances of the inference rules, and whose edges are the sequents. \textit{(Sequent) proofs} are derivations where all leaves are axioms.

\textbf{Exercise 1.1:} Prove Pierce’s law \( ((a \supset b) \supset a) \supset a \) in this sequent calculus.

\( \bullet \) As in the propositional case, we consider formulas in nfv.

\( \bullet \) We do not discuss semantics in this course.

\textbf{Exercise 1.2:} Prove the drinker’s formula \( \exists z (p x \supset \forall y. p y) \) in the sequent calculus.
Additive vs. Multiplicative

\[
\begin{align*}
\text{multiplicative} & \quad \text{additive} \\
\lor & \quad \lor \\
\vdash & \quad \vdash \\
\Gamma, A, B & \quad \Gamma, A \lor B \\
\Delta, A \land B & \quad \Delta, A \land B \\
\otimes & \quad \oplus \\
\Gamma, A & \quad \Gamma, A \otimes B \\
\Delta, A & \quad \Delta, A \otimes B \\
\end{align*}
\]

Multiplicative Linear Logic (MLL)

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

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

Multiplicative Additive Linear Logic (MALL)

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

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

- 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 \( A \oplus B \) imply each other, and similarly for \( A \otimes B \) and \( A \oplus B \). 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:** \[ A ::= a \mid a^\perp \mid A \otimes B \mid A \oplus B \]
- **Negation:** \[ a^\perp \perp = a \] \[ (A \otimes B)^\perp = A^\perp \otimes B^\perp, \ (A \oplus B)^\perp = A^\perp \oplus B^\perp \]
- **Sequents:** \[ \Gamma ::= A_1, A_2, \ldots, A_k \]
- **Inference rules (sequent calculus):**

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

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:** \[ A ::= a \mid A \land B \mid A \rightarrow B \]
- **Sequents:** \[ A_1, A_2, \ldots, A_k \vdash B \]
- **Inference rules (sequent calculus):**

\[
\begin{align*}
\text{ax} & : a \\
\Gamma, A & \vdash \Gamma, B \\
\Gamma & \vdash \Gamma, A \land B \\
\Gamma & \vdash \Gamma, A \rightarrow B
\end{align*}
\]

In this course we only consider the disjunction-free fragment of intuitionistic logic (\( \lor \) and \( \perp \) are absent).

- Note that implication is now a primitive.
- Negation can be recovered by fixing a fresh propositional variable \( a_0 \) and define \( \neg A = A \rightarrow a_0 \).
- 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 \((a \rightarrow b) \rightarrow b \rightarrow a\) in intuitionistic logic?

Recall that in classical logic we can encode \( A \rightarrow B \) 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 \( A \otimes B \) 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).

Intuitionistic Multiplicative Linear Logic (IMLL)

- **Formulas:** \[ A ::= a \mid A \otimes B \mid A \multimap B \]
- **Sequents:** \[ A_1, A_2, \ldots, A_k \vdash B \]
- **Inference rules (sequent calculus):**

\[
\begin{align*}
\text{ax} & : a \\
\Gamma, A, B & \vdash \Gamma, C \\
\Gamma & \vdash \Gamma, A \otimes B \\
\Gamma & \vdash \Gamma, A \multimap B
\end{align*}
\]

- Recall that in classical logic we can encode \( A \multimap B \) as \( A \rightarrow B \). So, every formula in IMLL 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 IMLL (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

\[
\frac{[x : A]^x \vdots \ldots \quad M : B \quad \lambda x. M : A \supset B}{\supset_i, x} \quad \frac{M : A \supset B \quad N : A}{MN : B \supset_E}
\]

- This includes pairs/products/conjunction

\[
\frac{M : A \quad M : B}{\langle M, N \rangle : A \land B \land I} \quad \frac{M : A \land B}{\pi_1(M) : A \land E, 1} \quad \frac{M : A \land B}{\pi_2(M) : B \land E, 2}
\]

Beta-reduction

- For implication:

\[
\frac{[x : A]^x \vdots \ldots \quad M : B \quad \lambda x. M : A \supset B}{\supset_i, x} \quad \frac{M : A \supset B \quad N : A}{MN : B \supset_E}
\]

- For conjunction:

\[
\frac{M : A \quad N : B \quad \langle M, N \rangle : A \land B \land !}{\pi_1(M,N) : A \land E, 1} \quad \frac{M : A \land B \quad \langle M, N \rangle : A \land B \land !}{\pi_2(M,N) : B \land E, 2} \quad \frac{M : A \quad N : B}{\langle M, N \rangle : A \land B \land !}
\]

- ($\supset / \land$)–Intuitionistic natural deduction is ideal:
  - Reduction is confluent and strongly normalizing
  - Normal forms represent the meaning of a term
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 ($\supset L$). 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 ($\text{Ax}$, $\supset R$, $\land R$) 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.
Permutations

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

\[
\begin{align*}
\Gamma, f: A \supset B, \Delta &\vdash x: B, \Delta, y: C \vdash N[f M/x]: D \\
\Gamma, f: A \supset B, \Delta &\vdash \lambda y. N[f M/x]: C \supset D
\end{align*}
\]

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 \( \supset \))

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:

\[
A \supset B = ?A \Rightarrow B
\]

- \( ?A \Rightarrow B \) 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

\[ \vdash A, A \text{ Ax} \]

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

Multiplicatives:

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

\[ \vdash \Gamma, A \otimes B \]

Additives:

\[ \vdash \Gamma, A \oplus B \]

Exponentials:

\[ \vdash ?\Gamma, A \]

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

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

The connectives are called: \textit{par} \( \& \), \textit{tensor} \( \otimes \), \textit{plus} \( \oplus \), with \( k_1 \) and why not \( ? \). The \textit{modality} \( ? \) governs whether formulas can be weakened and contracted or not. The rule \( ! \) for the \textit{bang} requires that the context \( ?\Gamma = ?A_1, \ldots, ?A_n \) contains only formulates that can be contracted. This is necessary for cut-elimination to work.

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

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

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

The dream of the 90s

New grounds in typed functional programming:

\begin{itemize}
  \item Linearity gives destructive updates (i.e. mutable state)
  \item Multiplicatives \( \otimes \) and \( \Rightarrow \) give deadlock-free concurrency
\end{itemize}

But:

\begin{itemize}
  \item The proof system is a sequent calculus!
  \item What about natural deduction?
\end{itemize}

Proof nets

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

\[ \vdash A, A \text{ Ax} \quad \implies \quad A, A \]

\[ \vdash \Gamma, A \quad \vdash B, \Delta \quad \implies \quad A, B \]

\[ \vdash \Gamma, A \otimes B, \Delta \quad \implies \quad A \otimes B \]

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

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

\begin{itemize}
  \item inferences become \textit{nodes}, called \textit{links},
  \item in a \textit{graph} called a \textit{proof structure}
\end{itemize}

Linear logic led to a great diversification of research in the areas around proof theory and computation in the 1990s, going into the 2000s and 2010s. Early new entrants were game semantics, interaction nets, and geometry of interaction.

From the start, the hope was that linear logic could give a proof-theoretic (and thus typed, and thus safe) account of various computational phenomena, such as destructive updates and concurrency. The programming language Clean, developed independently and simultaneously with linear logic, implements the former idea. Work towards the latter continues in the area of session types.
Example

\[
\begin{align*}
\vdash a, Ax & \quad \vdash b, Ax \quad \vdash c, Ax \\
\vdash a, Ax & \quad \vdash b, Ax \quad \vdash c, Ax \\
\vdash a, Ax & \quad \vdash b, Ax \\
\vdash a, Ax & \quad \vdash b, Ax \quad \vdash c, Ax
\end{align*}
\]

\[
\begin{align*}
\vdash a, Ax & \quad \vdash b, Ax \\
\vdash a, Ax & \quad \vdash b, Ax \\
\vdash a, Ax & \quad \vdash b, Ax
\end{align*}
\]

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

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

\[
\begin{align*}
\vdash a, Ax & \quad \vdash a, Ax & \quad \vdash a, Ax \\
\vdash a, Ax & \quad \vdash a, Ax & \quad \vdash a, Ax \\
\vdash a, Ax & \quad \vdash a, Ax & \quad \vdash a, Ax
\end{align*}
\]

\[
\begin{align*}
\vdash a, Ax & \quad \vdash a, Ax & \quad \vdash a, Ax \\
\vdash a, Ax & \quad \vdash a, Ax & \quad \vdash a, Ax
\end{align*}
\]

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 $\otimes$
- 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

```
a
\otimes
b
\&
c
\otimes
d
\&
a
\otimes
b
\&
c
\otimes
da
\&
a
\otimes
b
\&
c
\otimes
da
\&
a
\otimes
b
\&
c
\otimes
da
\&
a
\otimes
b
\&
c
\otimes
da
\&
a
\otimes
b
\&
c
\otimes
da
\&
```

Contractibility

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

```
\text{\begin{figure*}[h]
\begin{center}
\includegraphics[width=\textwidth]{contractibility_graph.png}
\end{center}
\end{figure*}}
```

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

Implemented in linear time via union-find

This is the sequent + axioms definition.

- **Exercise 2.12:** Give also the graph–of–rules definition.
- **Exercise 2.13:** Prove that the two definitions are isomorphic.
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.
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:

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

  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:

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

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

Examples

- undirected graph

  \[
  \begin{array}{ccc}
  w & & z \\
  u & \rightarrow & v \\
  \end{array}
  \]

  \[ V = \{u, v, w, z\} \]

  \[ E = \{uw, uv, vw, vz\} \]

- directed graph

  \[
  \begin{array}{ccc}
  w & & z \\
  u & \rightarrow & v \\
  \end{array}
  \]

  \[ V = \{u, v, w, z\} \]

  \[ w \rightarrow u, u \rightarrow v, v \rightarrow u, v \rightarrow w, v \rightarrow z, z \rightarrow z \]
Graph Homomorphisms and Isomorphisms

- A **graph homomorphism** \( h : G \to H \) is a map \( h : V_G \to V_H \) 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** \( h : G \to H \) is a bijection \( h : V_G \to V_H \) such that \( h \) and \( h^{-1} \) are both homomorphisms.

- A **directed graph homomorphism** \( h : G \to H \) is a map \( h : V_G \to V_H \) such that for all \( x, y \in V_G \) we have that \( x \rightarrow_G y \) implies \( h(x) \rightarrow_H h(y) \).

- A **directed graph isomorphism** \( h : G \to H \) is a bijection \( h : V_G \to V_H \) such that \( h \) and \( h^{-1} \) are both homomorphisms.

---

Paths and Cycles

- A **path** \( p \) in a graph \( G \) is a sequence of vertices \( x_0, x_1, \ldots, x_n \in V_G \) such that \( x_0x_1, x_1x_2, \ldots, x_{n-1}x_n \in E_G \). 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 \( H = \langle V_H, E_H \rangle \) and \( G = \langle V_G, E_G \rangle \) 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 \( \forall u, v \in V_H. uv \in E_H \implies uv \in E_G \).

- We say that \( G \) contains \( H \) as induced subgraph if there is an injective homomorphism \( f : H \to G \) such that \( f(H) \) is an induced subgraph of \( G \). If this is not the case, we say that \( G \) is \( H \)-free.

- Example: is a subgraph of (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”.

---

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 . 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:**

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

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

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

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:**

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

\[
\begin{array}{c}
c \\
\downarrow \\
a \\
\end{array} \quad \begin{array}{c}
b \\
\downarrow \\
d
\end{array}
\]

**Equivalence of formulas:**

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

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

Exercise 3.6: 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:

```
      •—•
     /   |
    /    |
   /     |
```

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:

```
  X X
  •—•
```

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: 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

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 $\alpha$-connected and $\alpha$-acyclic.

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 $T_{RB}(A)$:

\[
\begin{align*}
T_{RB}(\varnothing) & \\
T_{RB}(\alpha) & \\
T_{RB}(A \otimes B) & \\
T_{RB}(A) \otimes T_{RB}(B) & \\
T_{RB}(A \otimes B) & \\
T_{RB}(A) \otimes T_{RB}(B) & \\
T_{RB}(A) & \\
T_{RB}(B) &
\end{align*}
\]

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 $\alpha$-acyclic and $\alpha$-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 \([\cdot] : \text{MLL formulas to (labelled) graphs:}\)
   \[
   \begin{align*}
   [a] & = \bullet a \quad \text{(single vertex labelled } a) \\
   [a^\perp] & = \bullet a^\perp \quad \text{(single vertex labelled } a^\perp) \\
   [A \otimes B] & = [A] + [B] \\
   [A \otimes B] & = [A] \times [B]
   \end{align*}
   \]
   Let \( J_\Gamma K = J_{A_1} K + J_{A_2} K + \cdots + J_{A_n} K \)

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{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

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

\[
\begin{array}{c}
  b^\perp \quad c \quad c^\perp \quad d^\perp
\end{array}
\]

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 \textit{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:

Deep inference

A method for structuring proofs

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

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

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

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

\[
\begin{align*}
A \lor (B \lor C) & \alpha & A \lor B & \beta & A & \land (B \land C) & \alpha & A \land B & \beta & A & \top & \beta \\
(\neg A \lor \neg B) \lor \neg C & \alpha & \neg A \lor \neg \neg \neg A & \beta & (\neg A \land \neg B) \land \neg C & \alpha & \neg B \land \neg \neg \neg A & \beta & \neg A \land \neg \neg \neg A & \beta & \neg \top & \beta \\
\end{align*}
\]
- Non-invertible rules:

\[
\begin{align*}
\top & \alpha & (A \lor B) \land C & \alpha & A \land \neg A & \beta & A \lor \neg A & \beta & A & \bot & \beta \\
\end{align*}
\]
Example: Peirce's law

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

Intuitionistic logic

\[ A \quad C \quad \vdash A \lor B \quad A \land B \quad A \rightarrow B \]

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

Example: switch

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

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

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

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

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

\[ A \lor (B \land C) \]
Multiplicative Linear Logic

\[
\begin{array}{c}
\begin{array}{c|c|c}
\hline
A & A_1 & A_2 \\
\hline
C & C_1 & C_2 \\
\hline
A & B_1 \\
C & B_2 \\
\hline
\end{array}
\end{array}
\]

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

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

- Non-invertible rules:

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

Example

\[
\begin{align*}
((\alpha \otimes \beta) \otimes b) & \otimes c \\
\alpha & \quad \beta \\
\beta & \quad b
\end{align*}
\]

From deep inference to proof nets

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

\[
\begin{array}{c}
\begin{array}{c|c|c}
\hline
A & \rightarrow & B \\
\hline
\end{array}
\end{array}
\]

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

Inductive translation

\[
A \downarrow C ::= a \mid A_1 \times A_2 \mid C_1 \times C_2
\]
Composition

- Composition is path composition across the common formula

\[
\sigma \mathrel{\&} \bar{\sigma} \\
((\sigma \mathrel{\&} \bar{\sigma}) \mathrel{\&} b) \mathrel{\&} \bar{b} \\
((\sigma \mathrel{\&} \bar{\sigma}) \mathrel{\&} b) \mathrel{\&} \bar{b} \\
((\bar{\sigma} \mathrel{\&} \sigma) \mathrel{\&} c) \mathrel{\&} \tau
\]

= \[
((\sigma \mathrel{\&} \bar{\sigma}) \mathrel{\&} b) \mathrel{\&} \bar{b} \\
((\bar{\sigma} \mathrel{\&} \sigma) \mathrel{\&} c) \mathrel{\&} \tau
\]

Rules

\[
\frac{I}{A \mathrel{\&} \bar{A}} \\
\frac{(A \mathrel{\&} B) \mathrel{\&} C}{A \mathrel{\&} (B \mathrel{\&} C)} \\
\frac{A \mathrel{\&} \bar{A}}{\bot}
\]

\[
\frac{(A \mathrel{\&} B) \mathrel{\&} C}{A \mathrel{\&} (B \mathrel{\&} C)} \\
\frac{A \mathrel{\&} \bar{A}}{\bot}
\]

\[
A \mathrel{\&} (B \mathrel{\&} C) \\
(\bar{A} \mathrel{\&} B) \mathrel{\&} C^\alpha \\
B \mathrel{\&} \bar{A}^\sigma
\]

\[
A \mathrel{\&} B \\
A \mathrel{\&} (B \mathrel{\&} C) \\
A \mathrel{\&} B
\]

\[
A \mathrel{\&} (B \mathrel{\&} C) \\
B \mathrel{\&} A \\
\bot
\]

\[
A \mathrel{\&} (B \mathrel{\&} C) \\
A \mathrel{\&} B \\
A \mathrel{\&} (B \mathrel{\&} C) \\
A \mathrel{\&} B
\]

\[
A \mathrel{\&} B
\]

\[
A \mathrel{\&} (B \mathrel{\&} C) \\
(\sigma \mathrel{\&} B) \mathrel{\&} C
\]

Switch

\[
(B \mathrel{\&} C) \mathrel{\&} D \\
B \mathrel{\&} (C \mathrel{\&} D)
\]

\[
A \\
B \mathrel{\&} C \\
C \mathrel{\&} D
\]

\[
\frac{A \mathrel{\&} D}{B \mathrel{\&} C} \\
\frac{A \mathrel{\&} D}{B \mathrel{\&} C}
\]

\[
A \\
D
\]

\[
B \\
E
\]

\[
B \mathrel{\&} C \\
B \mathrel{\&} E
\]

\[
A \\
D
\]

\[
B \\
C \\
D
\]

\[
B \\
E
\]
MLL is the logic of undirected, connected, acyclic networks.

- We can add the mix rule to allow disconnected networks
  \[ A \otimes B \vdash A \ux B \]

- 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)

\[
\begin{array}{c}
A_1 \otimes \cdots \otimes A_m \\
C_1 \ux \cdots \ux C_m \\
\end{array}
\]

\[
\begin{array}{c}
A_1 \\
\cdots \\
A_n \\
\end{array}
\]

\[
\begin{array}{c}
f \\
C_1 \\
\cdots \\
C_m \\
\end{array}
\]

From sequent calculus

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

\[
\frac{\Gamma, A \otimes B, \Delta}{\Gamma, A \ux B, \Delta}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]

\[
\frac{\Gamma, A, B}{\Gamma, A \ux B}
\]
**Eta-expansion**

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

**Cut-elimination**

\[
\frac{\vdash \Gamma, \Theta, A, B}{\vdash \Gamma, \Delta, \Theta} \quad \to \\
\frac{\vdash \Gamma, \Delta, B}{\vdash \Gamma, \Delta, \Theta}
\]

**Big picture**

Sequent calculus $\xrightarrow{1}$ Deep inference $\xrightarrow{1}$

1. Proof nets (graph-of-rules) $\xrightarrow{2}$ Proof nets (sequent + axioms)

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

5. Lecture

Proof Nets for Additive Linear Logic

Additive linear logic

Formulæ:

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

Sequents:

\[ A \vdash C \] or \[ \vdash \overline{A}, C \]

Sequent calculus:

\[
\frac{A \vdash A}{A \vdash A} \quad \frac{A \vdash C \quad B \vdash C}{A \oplus B \vdash C} \quad A \vdash C_1 \quad C \vdash C_2 \quad \frac{\vdash A_{C_1} \& C_2}{A \vdash C_1 \& C_2}
\]

\[
A \vdash B \quad B \vdash C \quad \frac{A \vdash C}{A \vdash B \& C} \quad \frac{A \vdash B \quad A \vdash C}{A \vdash B \oplus C}
\]

Sum and product

\[
\frac{B \vdash D \quad C \vdash D}{B \& C \vdash D} \quad \frac{B \vdash C \quad C \vdash D}{B \oplus C \vdash D} \quad \frac{A \vdash B \quad A \vdash C}{A \vdash B \& C} \quad \frac{A \vdash B \quad A \vdash C}{A \vdash B \oplus C}
\]
Proof nets: graph–of–rules

\[
\begin{align*}
\Gamma, A, A &\vdash A, A, A + R, 1 \\
\Gamma, A + B, A, A &\vdash A, A, A + R, 1 \\
\Gamma &\not\vdash A, A, A \times R, 1
\end{align*}
\]

\[
\begin{align*}
\Gamma &\not\vdash A, A, A \times R, 1 \\
\Gamma &\not\vdash A, A, A \times R, 1 \\
\Gamma, A, A, A &\vdash A, A, A + R, 1 \\
\Gamma, A + B, A, A &\vdash A, A, A + R, 1 \\
\Gamma &\not\vdash A, A, A \times R, 1
\end{align*}
\]

Proof nets: (two-sided) sequent + axioms

\[
\begin{align*}
A &\vdash A, A \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1
\end{align*}
\]

\[
\begin{align*}
A &\vdash A, A \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1
\end{align*}
\]

\[
\begin{align*}
A &\vdash A, A \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1
\end{align*}
\]

\[
\begin{align*}
A &\vdash A, A \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1 \\
A &\vdash A + B, A, A &\vdash A, A, A + R, 1
\end{align*}
\]
**Eta-expansion**

\[
\begin{align*}
A + B \vdash & A + B \quad \text{\textit{Ax}} \\
\iff & \quad \frac{A \vdash A}{A + A \vdash A} +R,1 \quad \frac{B \vdash B}{B + A \vdash A} +R,2 \quad +L
\end{align*}
\]

- Links may be restricted to atoms

**Composition**

- Composition is relational composition
- Requires eta-expansion

**Correctness**

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

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

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

Example

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 \( B_i \) 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

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

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

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

\[
\begin{align*}
\Gamma & \vdash \Delta \\
A & \vdash B_1 \\
C & \vdash B_2
\end{align*}
\]

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

Deep inference: additive linear logic

\[
\begin{align*}
\Gamma & \vdash \Delta \\
A & \vdash B_1 \\
C & \vdash B_2
\end{align*}
\]

- Connectives: \(\times, +, 1, 0\)
- Invertible rules:
  \[
  \begin{align*}
  A + (B + C) & \sigma A \times (B \times C) \\
  (A + B) \times C & \alpha A \times (B \times C) \\
  \end{align*}
  \]
- Non-invertible rules:
  \[
  \begin{align*}
  0 & \lor A  \\
  A & \top \\
  \end{align*}
  \]

Inductive translation

\[
\begin{align*}
\Gamma & \vdash \Delta \\
A & \vdash B_1 \\
C & \vdash B_2
\end{align*}
\]

- Connectives: \(\land, \lor, \top, \bot\)
- Invertible rules:
  \[
  \begin{align*}
  A \lor (B \lor C) & \alpha A \lor B \lor C \\
  
  \end{align*}
  \]
- Non-invertible rules:
  \[
  \begin{align*}
  A \lor A & \Delta A \\
  A & \top \\
  \end{align*}
  \]
Example again again

\[
\begin{align*}
A + (B + C) & \quad A + B & \quad A + (B + C) \\
(A + B) + C & \quad B + A & \quad (A + B) + C \\
A & \quad A & \quad A \\
A × (B × C) & \quad A × B & \quad A × (B × C) \\
A & \quad (A × B) × C & \quad B × A \\
A × (B × C) & \quad A × B & \quad A × (B × C) \\
A × B & \quad B × A & \quad A × B \\
A & \quad 1 & \quad A \\
\end{align*}
\]
Big picture

Sequent calculus \rightarrow 1 \rightarrow \text{Deep inference}

\rightarrow 2 \rightarrow \text{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


6. Lecture
Fibrations and Skew Fibrations

Willem Heijltjes and Lutz Straßburger

What are Fibrations?

- originate in topology as a generalization of fiber bundles:

What are Fibrations?

- 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 \( w' \overset{f(v)}{\rightarrow}_H \) then there is a unique \( w \in V_G \) with \( w \overset{f}{\rightarrow}_G v \) and \( f(w) = w' \).

![Diagram of fibrations in directed graphs]

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 fibrations in undirected graphs]

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 fibrations in undirected graphs]

- 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 $f: G \rightarrow H$ 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:

Examples:

\[
\begin{align*}
\text{skew fibration:} & \quad & \quad & \\
\text{not skew fibration:} & \quad & \\
\end{align*}
\]

SKSg: A Deep Inference System for Classical Logic

\[
\begin{align*}
& i_\downarrow \vdash \frac{T}{A \lor A} \\
& s \vdash \frac{(C \lor A) \land B}{C \lor (A \land B)} \\
& c \vdash \frac{A \lor A}{A} \\
& w \vdash \frac{\bot}{A} \\
\end{align*}
\]

down-fragment $\text{SKS}_\downarrow$  

up-fragment $\text{SKS}_\uparrow$

rewriting modulo:

$A \land B = B \land A$ \quad $A \land (B \land C) = (A \land B) \land C$ \quad $A \land \top = A$

$A \lor B = B \lor A$ \quad $A \lor (B \lor C) = (A \lor B) \lor C$ \quad $A \lor \bot = A$

SKSg: A Deep Inference System for Classical Logic

\[
\begin{align*}
& i_\downarrow \vdash \frac{T}{A \lor A} \\
& s \vdash \frac{(C \lor A) \land B}{C \lor (A \land B)} \\
& c \vdash \frac{A \lor A}{A} \\
& w \vdash \frac{\bot}{A} \\
\end{align*}
\]

Theorem: (Decomposition and Interpolation)
Decomposition for the Down-Fragment

\[
\begin{align*}
\text{SKS}_g : & \quad \downarrow_i \quad \frac{T}{A \lor A} \\
& \quad \downarrow_w \quad \frac{B}{B \lor A} \\
& \quad \downarrow_c \quad \frac{A \lor A}{A}
\end{align*}
\]

Theorem: (Decomposition)

\[
\begin{align*}
\text{SKS}_g : & \quad \downarrow_i \quad \frac{T}{A} \\
& \quad \downarrow_{(i,s)} \quad \frac{A'}{A} \\
& \quad \downarrow_{(c,w)} \quad \frac{A'}{A}
\end{align*}
\]

\[\text{this is MLL}\]

Decomposition for the Down-Fragment (without units)

\[
\begin{align*}
\text{SKS}_g' : & \quad \downarrow_i \quad \frac{B}{B \land (A \lor A)} \\
& \quad \downarrow_w \quad \frac{A \lor A}{A}
\end{align*}
\]

Theorem: (Decomposition)

\[
\begin{align*}
\text{SKS}_g' : & \quad \downarrow_{(i,s)} \quad \frac{B}{A'} \\
& \quad \downarrow_{(c,w)} \quad \frac{B}{A}
\end{align*}
\]

\[\text{this is unit-free MLL}\]

First Skew Fibration Theorem

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

Then there is a skew fibration \( f: \downarrow[A] \rightarrow \downarrow[B] \) iff \( \downarrow{(c,w)} \quad \downarrow[B] \)

\( A \)

Examples:

- \( a \land b \lor (a \land b) \lor ((a \lor b) \land (a \lor b)) \)

- \( c \lor \frac{(a \land b) \lor (a \land b) \lor ((a \lor b) \land (a \lor b)) \lor c}{(a \land b) \lor ((a \lor b) \land (a \lor b)) \lor c} \)

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

This theorem has first been observed by Hughes:  

An alternative proof using deep inference can be found here:
Exercise 6.1: Show that every skew fibration defines an ALL proof net.

Exercise 6.2: (Difficult) Prove the theorem.
Summary (Multiplicative Part)

sequent calculus:

\[
\frac{\Gamma, A, B}{\Gamma, A \land B}
\]

\[
\frac{\Gamma, A}{\Gamma, A \lor B}
\]

\[
\frac{\Delta, B}{\Gamma, A \oplus B}
\]

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

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

deep inference:

\[
\frac{1}{\Gamma, A \otimes 1}
\]

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

handsome proof nets
critically chorded RB-cographs (polynomial)

Summary (Additive Part)

sequent calculus:

\[
\frac{\Gamma, A, B}{\Gamma, A \land B}
\]

\[
\frac{\Gamma, A}{\Gamma, A \lor B}
\]

\[
\frac{\Delta, B}{\Gamma, A \oplus B}
\]

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

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

deep inference:

\[
\frac{w}{A}
\]

\[
\frac{A \lor A}{A}
\]

\[
\left(\frac{w}{A}\right)
\]

\[
\frac{A}{A \land A}
\]

skew fibrations
homomorphisms on cographs with skew lifting property (polynomial)
7. Lecture

Classical Propositional Combinatorial Proofs

Willem Heijltjes and Lutz Straßburger

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

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

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

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

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

Lafont’s examples: non-confluence

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

Lafont’s examples: non-termination

• 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 $\overline{\overline{\alpha}} \cong \alpha$
- conjunction and disjunction are products and coproducts
- disjunction acts as implication $A \supset B = \overline{A} \lor B$

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

\[
\frac{}{\bot} \text{ (ex falso sequitur quodlibet)}
\]

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**

```
((P \supset Q) \supset P) \supset P
```

- Handsome proof net
- Cograph
- Skew fibration
- Conclusion

**Example: distributivity**

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

\[
\overline{a} \lor (\overline{b} \land \overline{c}) \lor (a \land b) \lor (a \land c)
\]
Special case:

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

\[
\begin{array}{c}
F \\
G \\
H \\
K
\end{array} \quad + \quad \begin{array}{c}
F \\
G \\
H \\
K
\end{array} \quad \mapsto \quad \begin{array}{c}
F \\
G \\
H \\
K
\end{array}
\]

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

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

\[
\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A \conjunction, \Delta \conjunction}
\]

\[
\frac{\vdash \Gamma, A \conjunction, \Delta \conjunction}{\vdash \Gamma, A \conjunction, \Delta \conjunction}
\]

Combinatorial proofs are complexity-sensitive

Example: distributivity, the other direction (1)

\[
a \conjunction (b \disjunction c) \iff (a \conjunction b) \disjunction (a \conjunction c)
\]

Example: distributivity, the other direction (2)

\[
a \conjunction (b \disjunction c) \iff (a \conjunction b) \disjunction (a \conjunction c)
\]
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


8. Lecture
First-Order Combinatorial Proofs

Willem Heijltjes and Lutz Straßburger

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: $t ::= x | f(t_1, \ldots, t_n)$
Atoms: $a ::= 1 | 0 | p(t_1, \ldots, t_n) | \bar{p}(t_1, \ldots, t_n)$
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:
$\top = 0$
$\bot = 1$
$\exists x. A = \forall x. \bar{A}$
$\forall x. A = \exists x. \bar{A}$
$A \land B = \bar{A} \lor \bar{B}$
$A \lor B = \bar{A} \land \bar{B}$

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:

\[ [a] = \bullet a \quad \text{(for any atom } a) \]

\[ [A \lor B] = [A] + [B] \]

\[ [\exists x.A] = \bullet x \times [A] \]

Example:

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

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

Graphs of Formulas

Congruence Relation \(\equiv\) on Formulas:

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

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

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

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

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

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

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

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

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

Example:

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

First-Order Combinatorial Proofs

Definition:

A \textit{first-order combinatorial proof} of a formula \(A\) is a skew bifibration \(\phi: C \to G\) 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 \( \{x, \exists y\} \) 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:

![Diagram of linked fograph, dualizer, and leap graph](image)

- 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:

![Diagram of linked fograph, dualizer, and leap graph](image)

- 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

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.

First-Order Sequent Calculus

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 $\rightarrow$ LK1 proof ✗
Exercise 8.4: Give the deep inference derivations for the four combinatorial proofs above.
Resource Management in First-Order Proofs

Example

1. \(\forall y (py \land pfy)\)

Example

\(qab \lor qba \lor \exists x \exists y qxy\)
Example (Drinker’s Formula)

\exists x (px \supset \forall y py)

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

This 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.
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:**

These are all the same combinatorial proofs, just drawn in different ways with some formulas in the conclusion flipped upside-down.

---

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:

- The last one on the first row 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.
A Final Last Word on Classical Combinatorial Proofs
(and Deep Inference)

\[\iff\]

From Classical to Intuitionistic Combinatorial Proofs

\[
\text{arena net} \\
\text{skew fibration} \\
\text{arena}
\]

\[((a \supset a) \supset b) \supset (b \wedge b)\]

cograph = classical logic formula
to associativity and commutativity

arena = intuitionistic logic formula
to associativity and commutativity

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:

The details for the material in this lecture can be found in

- Willem Heijltjes and Dominic Hughes and Lutz Straßburger: "Intuitionistic Proofs without Syntax". LICS 2019
From Formulas to Arenas

- **Operations on dags:**
  - $G \cdot H$
  - $G + H$
  - $G \triangleright H$

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

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

**Example:**
- $((P \supset P) \supset Q) \land Q \supset R \land ((S \supset S) \supset S)$

\[ P \rightarrow P \rightarrow Q \rightarrow R \]
\[ S \rightarrow S \rightarrow S \]

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

**Theorem:** $[A] = [B]$ iff $A \equiv B$.

where $\equiv$ is generated from

- $(A \land B) \supset C \equiv A \supset (B \supset C)$
- $A \land (B \land C) \equiv (A \land B) \land C$
- $A \land B \equiv B \land A$

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

From Formulas to Arenas

**Example:**
- $((P \supset P) \supset Q) \land Q \supset R \land ((S \supset S) \supset S)$

\[ P \rightarrow P \rightarrow Q \rightarrow R \]
\[ S \rightarrow S \rightarrow S \]

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

**Theorem:** $[A] = [B]$ iff $A \equiv B$.

where $\equiv$ is generated from

- $(A \land B) \supset C \equiv A \supset (B \supset C)$
- $A \land (B \land C) \equiv (A \land B) \land C$
- $A \land B \equiv B \land A$

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

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**

- **linked arena:** every partition (link) consists of exactly two vertices: one even and one odd

  **Example:**
  - $P^* \rightarrow P^* \rightarrow Q^* \rightarrow Q^*\overset{\text{odd vertex (⋆) = odd depth}}{\rightarrow} R^* \rightarrow S^* \rightarrow S^*\overset{\text{even vertex (○) = even depth}}{\rightarrow}$

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

  **Example:**
  - $P^* \rightarrow P^* \rightarrow Q^* \rightarrow Q^*\overset{\text{add a link edge for each link}}{\rightarrow} R^* \rightarrow S^* \rightarrow S^*$
An arena net is a linked arena that obeys a correctness criterion, i.e., that is a linked arena that comes from a proof.

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
Skew fibrations

\[(\forall v, w) \iff v \land w \text{ and } w \not\land v\]

\[\forall v, w : v \not\land w \iff w \not\land v\]

Example:

\[u \Rightarrow w \Rightarrow y \quad u \land v \quad w \not\land x\]

\[v \Rightarrow x \Rightarrow z \quad w \not\land w \quad x \land y\]

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

- edges: \(v \Rightarrow w\) implies \(f(v) \Rightarrow f(w)\)
- equivalence: \(v \equiv w\) implies \(f(v) \equiv f(w)\)
- roots: \(r \Rightarrow w\) implies \(f(r) \Rightarrow f(w)\)
- conjuncts: \(v \land w\) implies \(f(v) \land f(w)\)

and satisfies the following skew lifting condition:

- if \(f(v) \land f(w)\) then there exists \(u\) with \(v \Rightarrow u\) and \(f(u) \not\land f(w)\).

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).

Intuitionistic combinatorial proofs

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

Examples:

- [Diagram of ICP examples]

From Sequent Calculus to ICP

- [Diagram of sequent calculus to ICP examples]

Theorem: The statement of the third theorem is also called polynomial full completeness (and the term has been invented for ICP, following the full completeness known from game semantics).
Comparison to $\lambda$-Calculus

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

\[
\begin{align*}
\frac{f : (P \Rightarrow P) \vdash Q \vdash \Gamma}{f : (P \Rightarrow P) \vdash} & \quad \frac{f : (P \Rightarrow P) \vdash Q \vdash}{f : (P \Rightarrow P) \vdash}
\end{align*}
\]

\[
\begin{align*}
\frac{\Gamma \vdash Q \vdash \Gamma}{\Gamma \vdash Q \vdash} & \quad \frac{\Gamma \vdash Q \vdash}{\Gamma \vdash Q \vdash}
\end{align*}
\]

Sequent Calculus Rule Permutations

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

Comonoid Transformations

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

Exercise 9.2: How many possible rule permutations are there?
<table>
<thead>
<tr>
<th>Transformation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma, A, B \vdash C \quad \Delta \vdash \Delta \vdash B \quad \Gamma, B \vdash C$</td>
<td>$\Gamma, A, B \vdash C \quad \Delta \vdash B \quad \Gamma, A \vdash C$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash B$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash (B \land C)$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash (B \land C)$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash (B \land C)$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash (B \land C)$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash (B \land C)$</td>
</tr>
<tr>
<td>$\Gamma, \Delta \vdash A \land B \quad \Gamma, \Delta \vdash A \land B$</td>
<td>$\Gamma, \Delta \vdash \Gamma, \Delta \vdash (B \land C)$</td>
</tr>
</tbody>
</table>
For intuitionistic combinational proofs

Option 1: use a special left-implication connective ▷

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

- The conclusion is still \( \Gamma, \Delta \vdash B \) and \( A \vdash A \) is hidden
- ▷ 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

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

- 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 \( B \).
\( \Gamma \vdash M : A \quad \Delta \vdash N : C \)

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

\[
\begin{array}{c}
\Gamma \\
F \\
G \\
A \\
\hline
B \\
\hline
\Delta \\
\end{array}
\quad + 
\begin{array}{c}
\Gamma \\
H \\
K \\
C \\
\hline
A \\
\hline
\Delta \\
\end{array}
\quad \rightarrow 
\begin{array}{c}
\Gamma \\
F \\
G \\
H \\
K \\
\Delta \\
\end{array}
\]

\[ \begin{array}{c}
\Gamma \\
A \\
\hline
\supset \\
B \\
\hline
\end{array} \quad \Delta \quad \rightarrow 
\begin{array}{c}
\Gamma \\
A \\
\hline
\supset \\
B \\
\hline
\end{array}
\]

\[ \begin{array}{c}
\Gamma, x : A \vdash M : B \quad \Delta \\
\Gamma, x : A \vdash M[x/y] : B \\
\Lambda, x, A \vdash N : A \quad \Delta \\
\Delta \vdash N[\lambda M : B] : C
\end{array} \]

Building trees

\[ \begin{array}{c}
\Gamma \vdash M : A \quad \Delta \vdash N : B \\
\Gamma, \Delta \vdash N[M/x] : B
\end{array} \]

\[ \begin{array}{c}
\Gamma, \Delta \\
F, K \\
\hline
H \\
\hline
B \\
\hline
\end{array} \quad ::= \quad \begin{array}{c}
\Gamma, \Delta \\
F, K \\
\hline
H \\
\hline
B \\
\hline
\end{array} \quad \vdash 
\begin{array}{c}
\Gamma \\
G \\
\Delta \\
\hline
A \\
\hline
\hline
L \\
\hline
K \\
\hline
B \\
\hline
\end{array} \]
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