Complétude en logiques
Habilitation à diriger des recherches

Olivier Hermant
MINES ParisTech, PSL Research University

2017, April 20th
Completeness in Logics

Major role of the Completeness Theorem:
- Gödel sense
- exhaustive proof-search succeeds (eventually)
- fundamental property: cut elimination

Logics:
- automated theorem proving [P. Halmagrand]
- proof checking [R. Saillard]
- application domain: formal methods
  - large (mathematical) proofs
  - safe, bug-free, system conception
- theory of programming languages (type systems, semantics, static analysis) [T. Giang-Le, V. Maisonneuve]
- model checking, realizability, ...
Key Properties of Logical Systems

Proofs (Syntax)

Truth (Semantics)

Soundness

Completeness

\[ A \vdash B \]

\[ A \vDash B \]

Theorem (Soundness)
If a statement is provable, it is (universally) true.

Corollary (Consistency)
Not all statements have proofs.

Theorem (Completeness)
If a statement is (universally) true, it is provable.
Key Properties of Logical Systems

- Syntax
- Semantics
- Soundness
- Completeness
- Cut
- Admissibility

Theorem (Cut Admissibility)

If a statement is provable, then it is provable without cut.

- consistency
- automated proof-search
- focus on computation (CS point of view):
  - the site for interaction
  - proof terms
  - normalization (termination of proof-term reduction)
1. Introduction
2. Extension to Other Logics
3. Getting Rid of Tableaux
4. Opening the Box
5. Conclusion
Propositional Logic

- atomic formulas, connectives $\land$, $\lor$, $\Rightarrow$, $\neg$, $\bot$, $\top$
- semantics, truth tables

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>$A \land B$</th>
<th>$A \lor B$</th>
<th>$A \Rightarrow B$</th>
<th>$\neg A$</th>
<th>$\bot$</th>
<th>$\top$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- syntax, a proof-search method called the tableaux method.
- refutation-based method: to show $F$, derive a contradiction from $\neg F$.
- one rule per connective, another for its negation

$$
\frac{\bot}{\bigcirc \bot} \quad \frac{\neg \top}{\bigcirc \neg \top} \quad \frac{F, \neg F}{\bigcirc \text{cl}}
$$

$$
\frac{A \land B}{A, B} \quad \frac{\neg (A \lor B)}{\neg A, \neg B} \quad \frac{\neg (A \Rightarrow B)}{A, \neg B} \\
\frac{\neg (A \land B)}{\neg A \land \neg B} \quad \frac{A \lor B}{A} \quad \frac{A \Rightarrow B}{A \lor \neg B}
$$
Example

- prove \((B \lor A) \Rightarrow (A \lor B)\)

\[\neg((B \lor A) \Rightarrow (A \lor B))\]

- tableau as a tree
- choice for rule application
- proof iff each branch is closed
- notation \(F_1, \ldots, F_n \leftarrow \circ\)
Example

- prove \((B \lor A) \Rightarrow (A \lor B)\)

\[
\neg((B \lor A) \Rightarrow (A \lor B)) \\
\frac{\neg}{B \lor A, \neg(A \lor B)}
\]

- tableau as a tree
- choice for rule application
- proof iff each branch is closed
- notation \(F_1, \ldots, F_n \leftarrow \odot\)
Example

- prove \((B \lor A) \Rightarrow (A \lor B)\)

\[
\neg \neg((B \lor A) \Rightarrow (A \lor B)) \quad \Rightarrow
\]

\[
\begin{array}{c}
\neg ((B \lor A) \Rightarrow (A \lor B)) \\
B \lor A, \neg (A \lor B) \\
\hline
B \\
A
\end{array}
\]

- tableau as a tree
- choice for rule application
- proof iff each branch is closed
- notation \(F_1, \cdots, F_n \hookrightarrow \odot\)
Example

▶ prove \((B \lor A) \Rightarrow (A \lor B)\)

\[
\neg((B \lor A) \Rightarrow (A \lor B)) \\
\frac{B \lor A, \neg(A \lor B)}{B} \lor \frac{A}{\neg A, \neg B}
\]

▶ tableau as a tree
▶ choice for rule application
▶ proof iff each branch is closed
▶ notation \(F_1, \cdots, F_n \hookrightarrow \odot\)
Example

- prove \((B \lor A) \Rightarrow (A \lor B)\)

\[
\begin{align*}
\neg((B \lor A) \Rightarrow (A \lor B)) & \quad \neg\Rightarrow \\
B \lor A, \neg(A \lor B) & \quad \lor \\
B & \quad \neg\lor \\
\neg A, \neg B & \quad \circ
\end{align*}
\]

- tableau as a tree
- choice for rule application
- proof iff each branch is closed
- notation \(F_1, \cdots, F_n \hookrightarrow \circ\)
Example

- prove \((B \lor A) \Rightarrow (A \lor B)\)

\[
\neg\neg((B \lor A) \Rightarrow (A \lor B))
\]

\[
\Rightarrow
\]

\[
B \lor A, \neg(A \lor B)
\]

\[
\lor
\]

\[
B
\]

\[
\lor
\]

\[
\neg A, \neg B
\]

\[
\lor
\]

\[
A
\]

\[
\lor
\]

\[
\neg A, \neg B
\]

\[
\lor
\]

- tableau as a tree
- choice for rule application
- proof iff each branch is closed
- notation \(F_1, \cdots, F_n \mapsto \circ\)
Example

prove \((B \lor A) \Rightarrow (A \lor B)\)

\[
\frac{\neg((B \lor A) \Rightarrow (A \lor B))}{B \lor A, \neg(A \lor B)} \quad \Rightarrow
\]

\[
\begin{array}{c}
B \\
\hline
\neg A, \neg B
\end{array}
\]

\[
\begin{array}{c}
\neg
\
\hlf
\end{array}
\]

\[
\begin{array}{c}
A \\
\hline
\neg A, \neg B
\end{array}
\]

\[
\begin{array}{c}
\neg
\
\hlf
\end{array}
\]

- tableau as a tree
- choice for rule application
- proof iff each branch is closed
- notation \(F_1, \cdots, F_n \leftarrow \circ\)
Soundness and Completeness for Tableaux

**Soundness**

If \( F_1, \cdots, F_n \nleftrightarrow \emptyset \), then \( F_1 \land \cdots \land F_n \) is unsatisfiable.

- no model of \( F_1, \cdots, F_n \)
- induction on the tableau proof and case analysis.
- refutation: \( \neg F \) unsatisfiable \( \sim \) for all interpretations, \( \llbracket F \rrbracket = 1 \)

**Completeness**

If a tableau \( F_1, \cdots, F_n \) cannot be closed, then \( F_1 \land \cdots \land F_n \) is satisfiable.

- another view of tableaux rules:
  - exhaustively searching for a countermodel
  - if for all interpretations, \( \llbracket F \rrbracket = 1 \), then search finds no consistent countermodel on input \( \neg F \sim \) closable tableau.
Countermodel from Exhaustion

- try to prove $A \Rightarrow (A \land B)$

\[
\begin{align*}
\neg(A \Rightarrow (A \land B)) \\
A, \neg(A \land B) \\
\neg A & \quad \neg B \\
\circ
\end{align*}
\]

- right branch *open and complete*

Complete Branch

A branch of a tableau is complete if all applicable rules have been applied.

- need to construct an **exhaustive proof-search algorithm**
  - collect literals (plain and negated atoms), $A$ and $\neg B$,
  - assign the truth values accordingly, $\llbracket A \rrbracket = 1$ and $\llbracket B \rrbracket = 0$,
  - yields $\llbracket \neg(A \Rightarrow (A \land B)) \rrbracket = 1$ (**falsifies** $A \Rightarrow (A \land B)$).
Completeness and Cut Admissibility

A \vdash B

A \models B

Soundness

A \vdash^{*} B

A, \neg B \hookrightarrow \odot

Tableaux

Completeness

???
Sequent Calculus

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

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

\[
\frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash A \lor B, \Delta} \quad ^\lor_R
\]

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

\[
\frac{\Gamma \vdash A \rightarrow B, \Delta}{\Gamma \vdash A \vdash B, \Delta} \quad ^\Rightarrow_R
\]

\[
\frac{\Gamma \vdash A \vdash B, \Delta}{\Gamma \vdash A \vdash B, \Delta} \quad ^\Rightarrow_L
\]

- upside down tableaux (two-sided and context duplication)
  - ¬□ rule
  - □ rule
  - no cut in the translation

\[
\neg((B \lor A) \Rightarrow (A \lor B)) \quad \Rightarrow \\
\neg((B \lor A) \Rightarrow (A \lor B)) \quad \Rightarrow \\
\neg B \quad \lor \\
\neg A, \neg B \quad \lor \\
\neg A, \neg B \quad \lor
\]

Tableaux

\[
\frac{B \vdash A, B}{B \vdash A \lor B} \quad ^\lor_L
\]

\[
\frac{A \vdash A, B}{A \vdash A \lor B} \quad ^\lor_R
\]

\[
\frac{B \lor A \vdash A \lor B}{(B \lor A) \Rightarrow (A \lor B)} \quad \Rightarrow_R
\]

Sequent Calculus
2. Extensions to Other Logics
Switching to First-Order

- variables, terms and quantifiers

\[ \forall x(P(x) \Rightarrow P(s(0))) \]

- first-order tableaux, first-order sequent calculus

- cut admissibility by the previous method

- but the complete exhaustive proof-search is **highly inefficient**
  - enumerates all the terms of the language \( t_0, t_1, \cdots \)
  - complete branch with \( \forall xF \) must have \( F[t_0/x], F[t_1/x], \cdots \)
  - some sweat to keep proof-search **fair**
Efficiency in First-Order Tableaux

- unefficient naive enumeration, what if $F[t_{2017}/x]$ right choice?
- do not know: *wait* to instantiate!
- free variable tableaux

\[
\begin{align*}
\neg (\exists x (D(x) \Rightarrow \forall y D(y))) \\
\therefore \neg (D(X) \Rightarrow \forall y D(y)) \\
\therefore D(X), \neg \forall y D(y) \\
\neg \forall y D(y) \\
\neg D(c) \\
\therefore \{ X \approx c \}
\end{align*}
\]

- Exponential speedups, *connection lost* with sequent calculus
  - freshness condition *globally* ensured, not *locally*
  - re-expand, double inverted induction, duplication

Exponential speedups, connection lost with sequent calculus

- freshness condition *globally* ensured, not *locally*
- re-expand, double inverted induction, duplication
Switching to Deduction Modulo Theory

Rewrite Rule

A term (resp. proposition) rewrite rule is a pair of terms (resp. formulæ) \( l \rightarrow r \), where \( \text{FV}(l) \subseteq \text{FV}(r) \) and, in the propositiona case, \( l \) is atomic.

Examples:

- term rewrite rule:
  \[
  A \cup \emptyset \rightarrow A
  \]

- proposition rewrite rule:
  \[
  A \subseteq B \rightarrow \forall x \ x \in A \Rightarrow x \in B
  \]

Conversion modulo a Rewrite System

We consider the congruence \( \equiv \) generated by a set of proposition rewrite rules \( \mathcal{R} \) and a set of term rewrite rules \( \mathcal{E} \) (often implicit)

Example:

\[
A \cup \emptyset \subseteq A \quad \equiv \quad \forall x \ x \in A \Rightarrow x \in A
\]
(Classical) Sequent Calculus modulo

We add two conversion rules:

\[
\begin{align*}
\frac{}{\Gamma, A \vdash \Delta} \quad \text{conv}_R, [A \equiv B] \\
\frac{}{\Gamma \vdash A, \Delta} \\
\frac{}{\Gamma \vdash B, \Delta}
\end{align*}
\]

\[
\begin{align*}
\frac{}{\Gamma \vdash B, \Delta} \\
\frac{}{\Gamma, B \vdash \Delta} \quad \text{conv}_L, [A \equiv B]
\end{align*}
\]

Or embed conversions modulo \(\mathcal{RE}\) directly inside the rules (next slide).
(Classical) Sequent Calculus

\[\vdash A \quad \text{ax}\]

\[
\Gamma, A, B \vdash \Delta \\
\Gamma, A \land B \vdash \Delta \\
\text{^L}
\]

\[
\Gamma, A \vdash \Delta \\
\Gamma, B \vdash \Delta \\
\Gamma, A \lor B \vdash \Delta \\
\text{^V_L}
\]

\[
\Gamma, B \vdash \Delta \\
\Gamma \vdash A, \Delta \\
\Gamma \vdash B, \Delta \\
\Gamma \vdash A \Rightarrow B, \Delta \\
\text{^L_R}
\]

\[
\Gamma \vdash A, \Delta \\
\Gamma \vdash B, \Delta \\
\Gamma \vdash A \land B, \Delta \\
\text{^R_L}
\]

\[
\Gamma \vdash A \land B, \Delta \\
\Gamma \vdash A \lor B, \Delta \\
\text{^L_R}
\]

\[
\Gamma \vdash A \lor B, \Delta \\
\Gamma \vdash A \Rightarrow B, \Delta \\
\text{^R_R}
\]
(Classical) Sequent Calculus Modulo

\[ \frac{A \vdash B}{\text{ax, } [A \equiv B]} \]
\[ \frac{\Gamma, A, B \vdash \Delta}{\Gamma, C \vdash \Delta} \quad \land_L, [C \equiv A \land B] \]
\[ \frac{\Gamma, A \vdash \Delta}{\Gamma, C \vdash \Delta} \quad \lor_L, [C \equiv A \lor B] \]
\[ \frac{\Gamma, B \vdash \Delta}{\Gamma, C \vdash \Delta} \quad \Rightarrow_L, [C \equiv A \Rightarrow B] \]

\[ \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash \Delta} \quad \text{cut, } [A \equiv B] \]
\[ \frac{\Gamma \vdash A, \Delta}{\Gamma \vdash C} \quad \text{\land_R, } [C \equiv A \land B] \]
\[ \frac{\Gamma \vdash B, \Delta}{\Gamma \vdash C} \quad \lor_R, [C \equiv A \lor B] \]
\[ \frac{\Gamma \vdash A, B, \Delta}{\Gamma \vdash C} \quad \Rightarrow_R, [C \equiv A \Rightarrow B] \]
Proof of $A \subseteq A$ with and without DM

▷ without:

\[
A \subseteq A \Rightarrow [\cdots], x \in A \vdash x \in A, A \subseteq A
\]
\[
A \subseteq A \Rightarrow [\cdots] \vdash x \in A \Rightarrow x \in A, A \subseteq A
\]
\[
A \subseteq A \Rightarrow [\cdots] \vdash \forall x(x \in A \Rightarrow x \in A), A \subseteq A
\]
\[
A \subseteq A \Rightarrow \forall x(x \in A \Rightarrow x \in A), \forall x(x \in A \Rightarrow x \in A) \Rightarrow A \subseteq A \vdash A \subseteq A
\]
\[
A \subseteq A \iff \forall x(x \in A \Rightarrow x \in A) \vdash A \subseteq A
\]
\[
\forall Y(A \subseteq Y \iff \forall x(x \in A \Rightarrow x \in Y)) \vdash A \subseteq A
\]
\[
\forall X \forall Y(X \subseteq Y \iff \forall x(x \in X \Rightarrow x \in Y)) \vdash A \subseteq A
\]

▷ with:

\[
x \in A \vdash x \in A
\]
\[
\vdash x \in A \Rightarrow x \in A
\]
\[
\vdash \forall x(x \in A \Rightarrow x \in A)
\]
\[
\Rightarrow_R
\]
\[
\forall_R
\]
\[
\text{conv}_R [A \subseteq A \equiv \forall x(x \in A \Rightarrow x \in A)]
\]
Proof of $A \subseteq A$ with and without DM

- without:

\[
\begin{align*}
A \subseteq A & \Rightarrow [\cdots], x \in A \vdash x \in A, A \subseteq A \\
A \subseteq A & \Rightarrow [\cdots] \vdash x \in A \Rightarrow x \in A, A \subseteq A \\
A \subseteq A & \Rightarrow [\cdots] \vdash \forall x(x \in A \Rightarrow x \in A), A \subseteq A \\
A \subseteq A & \Rightarrow [\cdots], A \subseteq A \vdash A \subseteq A
\end{align*}
\]

- with:

\[
\begin{align*}
x \in A & \vdash x \in A \\
\vdash x \in A \Rightarrow x \in A \Rightarrow_R \forall x(x \in A \Rightarrow x \in A) \\
\vdash A \subseteq A \Rightarrow_R \forall x(x \in A \Rightarrow x \in A) \\
\forall x(x \in A \Rightarrow x \in A) & \vdash A \subseteq A
\end{align*}
\]
Tableaux and Cuts in Deduction Modulo Theory

- beyond first order (*axiomless* higher-order logic, arithmetic, ...)
- everything depends on $\mathcal{RE}$.
  - consistency ($A \rightarrow \neg A$)
  - cut elimination ($A \rightarrow (A \Rightarrow A)$)
  - cut admissibility
  - undecidable, even if $\mathcal{RE}$ confluent terminating.

![Diagram showing relationships between Completeness, Tableaux Completeness, Consistency, Cut elimination, and Normalization]
Generic Approach for Tableaux

Nevertheless, genericity:

- as far as possible
  - needs only confluence
  - everything except countermodel construction

- difficulties (besides models)
  - fair and exhausting proof-search design (STEP)
  - interleave quantifier instantiation and rewriting
  - add free-variables

- optimized proof-search, holes on the complete branch
  - fill the gaps to get a (semi-)valuation
  - not forgetting rewriting
Semantics for Deduction Modulo Theory

- your favorite semantics
- add one constraint

Model of $\mathcal{RE}$

An interpretation $⟦[]⟧$ is a model of $\mathcal{RE}$ if for any $F, F'$, such that $F \equiv F'$, we have $⟦F⟧ = ⟦F'⟧$.

- straightforward Soundness Theorem
Specific Countermodel Constructions

Completeness of tableaux, hence cut admissibility for

- positive rewrite systems

\[
\begin{align*}
even(0) & \rightarrow \top \\
even(S(x)) & \rightarrow \neg \text{odd}(x) \\
odd(S(x)) & \rightarrow \neg \text{even}(x)
\end{align*}
\]

- ordered rewrite systems

- higher-order logic as a rewrite system
3. Getting Rid of Tableaux
Direct Completeness

- most difficulties in Tableaux Completeness
Direct Completeness

- most difficulties in Tableaux Completeness
- most difficulties in **Strong Completeness**
  - more flexibility in the semantics
  - 0/1 Boolean algebra (or Kripke structures) imposed by tableaux.
More Flexible Semantics: Algebraic Structures

- propositional intuitionistic logic here (first-order, higher-order possible)
- Heyting algebras
- a universe $\Omega$, operators $\land$, $\lor$, $\rightarrow$
- an order $\leq$: $\Omega$ is a lattice.
- lowest upper bound (join: $\land$), greatest lower bound (meet: $\lor$)

$$
a \land b \leq a \quad a \land b \leq b \quad c \leq a \text{ and } c \leq b \text{ implies } c \leq a \land b
$$

$$
a \leq a \lor b \quad b \leq a \lor b \quad a \leq c \text{ and } b \leq c \text{ implies } a \lor b \leq c
$$

- like Boolean algebras (classical case), but
- weak complement (aka implication property):

$$
a \land b \leq c \text{ iff } a \leq b \Rightarrow c
$$

- example: $\mathbb{R}$ and open sets:

$$
b \Rightarrow c := \text{ the interior of } b \cup \overline{a}
$$
Cut Admissibility: Algebraic Way

Base Elements of the Lindenbaum Algebra
\[
[A] = \{B \mid A \vdash B \text{ and } B \vdash A\}
\]

Lindenbaum algebra:
- interpretation of formulas
  - \([A] = [A]\) on atoms, then induction
  - \([A] \leq [B]\) iff \(A \vdash B\)

Fundamental Lemma
For any formula \(A\), \(\llbracket A \rrbracket = [A]\)

- what do we have?

Completeness
If \(\llbracket A \rrbracket \leq \llbracket B \rrbracket\) in all models, then \(A \vdash B\).

- this is the definition of \(\leq\) in the Lindenbaum algebra.
- need the cut rule
Base Elements of the Lindenbaum Algebra

\[ [A] = \{ B \mid A \vdash B \text{ and } B \vdash A \} \]
Cut Admissibility: Algebraic Way

Base Elements of the Context Algebra

\[[A] = \{\Gamma \mid \Gamma \vdash A\}\]

- \(\leq\) is \(\subseteq\) and g.l.b. (\(\wedge\)) and l.u.b. (\(\vee\)) are “intersection” and “union”
- close \(\Omega\) by arbitrary intersection:

The Algebra \(\Omega\)

\[\Omega = \left\{ \bigcap_{C \in C} [C] \mid \text{for } C \text{ set of formulas} \right\}\]

\(\Omega\) is composed of arbitrary intersections of base elements

- \(\Omega\) not closed by union
  - ★ there are other ways to compute a least upper bound ...

O. Hermant (MINES ParisTech)
Cut Admissibility: Algebraic Way

- set the interpretation of the atoms to be: $[[A]] = [A]$

**Key Theorem**

For *any* formula $A$, $[[A]] = [A]$. 
Cut Admissibility: Algebraic Way

- set the interpretation of the atoms to be: $[[A]] = [A]$

**Key Theorem**

For any formula $A$, $[[A]] = [A]$.

- what do we have?

**Completeness**

if $[[A]] \leq [[B]]$ in all models, then $A \vdash B$.

- (trivial) $A \in [A]$
- $[[A]] = [A]$ (Key Theorem)
- $[[A]] \subseteq [[B]]$ (Hypothesis)
- $[[B]] = [B]$ (Key Theorem)
- means $A \vdash B$
Cut Admissibility: Algebraic Way

- set the interpretation of the atoms to be: $[A] = \lceil A \rceil$

**Key Theorem**

For any formula $A$, $[A] = \lceil A \rceil$.

- what do we really need?

**Completeness**

If $[A] \leq [B]$ in all models, then $A \vdash B$.

- $A \in [A]$ (Key Theorem)
- $[A] \subseteq [B]$ (Hypothesis)
- $[B] \subseteq [B]$ (Key Theorem)
- means $A \vdash B$
**Cut Admissibility: Algebraic Way**

- $\Omega$ contains arbitrary intersections of base elements.

**Base Elements**

$\lceil A \rceil = \{ \Gamma \mid \Gamma \vdash A \}$

- $\leq$ is $\subseteq$. Gives a lattice. Also a Heyting algebra.
- set the interpretation of the atoms to be: $\llbracket A \rrbracket = \lceil A \rceil$

**Key Theorem**

For any formula $A$, $\llbracket A \rrbracket = \lceil A \rceil$.

- what do we have?

**Completeness**

if $\llbracket A \rrbracket \leq \llbracket B \rrbracket$ in all models, then $A \vdash B$.

Proof: $A \in \lceil A \rceil = \llbracket A \rrbracket \subseteq \llbracket B \rrbracket = \lceil B \rceil$. 
Cut Admissibility: Algebraic Way

- $\Omega$ contains arbitrary intersections of base elements.

**Base Elements**

$$[A] = \{ \Gamma \mid \Gamma \vdash^* A \}$$

- $\leq$ is $\subseteq$. Gives a lattice. Also a Heyting algebra ($\Rightarrow$ property difficult)
- set the interpretation of the atoms to be: $[[A]] = [A]$

**Key Theorem**

For any formula $A$, $A \in [[A]] \subseteq [A]$

- Similarities with Reducibility Candidate-models (Logical Relations)
  $$NE \subseteq R_A \subseteq SN \text{ (simplified)}$$

**Strong Completeness**

if $[[A]] \leq [[B]]$ in all models, then $A \vdash^* B$.

Cut Admissibility, Second Order: Algebraic Way

- \( \Omega \) contains arbitrary intersections of base elements.

**Base Elements**

\[
[A] = \{ \Gamma \mid \Gamma \vdash^* A \}
\]

- \( \leq \) is \( \subseteq \). Gives a lattice. Also a Heyting algebra (\( \Rightarrow \) property difficult)
- set the interpretation of the atoms to be: \([A] = [A]\)

**Key Theorem**

For any formula \( A \), \( A \sigma \in [A] \phi \subseteq [A \sigma] \),
for any \( \phi, \sigma \) such that \( \sigma(X_i) \in \phi(X_i) \subseteq [\sigma(X_i)] \)

- Similarities with Reducibility Candidate-models (Logical Relations)

\[
NE \subseteq R_A \subseteq SN \text{ (simplified)}
\]

\([A] \phi \in R_{A \sigma}, \text{ for any } \phi, \sigma \text{ s.t. } \phi(X_i) \in R_{\sigma(X_i)} \)

**Strong Completeness**

if \([A] \leq [B] \) in all models, then \( A \vdash^* B \).
Application to Higher-Order Logics

- does not apply directly to higher-order logic
- intensional logic
  \[ P(\top) \iff P(\top \land \top) \]
- \( \llbracket \top \rrbracket \neq \top \)
- V-complexes [Takahashi], [Prawitz], [Andrews]
- adapted to
  - intuitionnistic case,
  - linear case (phase semantics),
  - the Deduction modulo theory expression of HOL (classical and intuitionnistic).
4. Opening the Box
Inside Constructive Proofs

- Cut admissibility through tableaux, almost constructive
  - rebuild proof from scratch

- Henkin completeness proofs
Computational Content of Algebraic Proofs

- switch to Natural Deduction
- more work existing
  - Normalization by Evaluation
  - all Kripke (-like)
- easier to compare
  - and understand (at least, so we thought)
  - no problem with disjunction in Heyting algebra
What Had to be Done

- from Sequent Calculus to Natural Deduction
What Had to be Done

- from Sequent Calculus to Natural Deduction
- notion of cut-free proof

Cut-Free Proofs

A proof is **neutral** it is an elimination with cut-free premises and neutral principal premiss. A proof is **cut-free** it is an introduction with cut-free premises.

\[
\frac{\Gamma \vdash_{ne} A}{\Gamma \vdash^* A} \text{ coerce}
\]

\[
\frac{\Gamma \vdash^* A \quad \Gamma \vdash^* B}{\Gamma \vdash^* A \land B} \land_I
\]

\[
\frac{\Gamma \vdash^* A}{\Gamma \vdash^* A \lor B} \lor_I
\]

\[
\frac{\Gamma \vdash^* B}{\Gamma \vdash^* A \lor B} \lor_I
\]

\[
\frac{\Gamma \vdash_{ne} A \land B}{\Gamma \vdash_{ne} A} \land_{E_I}
\]

\[
\frac{A \in \Gamma}{\Gamma \vdash_{ne} A} \text{ ax}
\]

\[
\frac{\Gamma \vdash_{ne} A \land B}{\Gamma \vdash_{ne} B} \land_{E_r}
\]

\[
\frac{\Gamma \vdash_{ne} A \lor B}{\Gamma \vdash_{ne} C} \lor_E
\]

\[
\frac{A, \Gamma \vdash^* C}{B, \Gamma \vdash^* C} \lor_E
\]

\[
\frac{\Gamma \vdash_{ne} A \Rightarrow B}{\Gamma \vdash^* A \Rightarrow B} \Rightarrow_I
\]

\[
\frac{\Gamma \vdash_{ne} A}{\Gamma \vdash_{ne} B} \Rightarrow_E
\]
What Had to be Done

- from Sequent Calculus to Natural Deduction
- notion of cut-free proof

Cut-Free Proofs

A proof is **neutral** it is an elimination with cut-free premises and neutral principal premiss. A proof is **cut-free** it is an introduction with cut-free premises.

\[
\frac{\Gamma \vdash_{ne} A}{\Gamma \vdash A} \text{coerce}
\]
\[
\frac{\Gamma \vdash A \land B}{\Gamma \vdash_{ne} A \land B} \land_i
\]
\[
\frac{\Gamma \vdash A}{\Gamma \vdash A \lor B} \lor_l
\]
\[
\frac{\Gamma \vdash B}{\Gamma \vdash A \lor B} \lor_r
\]
\[
\frac{A \in \Gamma}{\Gamma \vdash_{ne} A} \text{ax}
\]
\[
\frac{\Gamma \vdash A \land B}{\Gamma \vdash_{ne} A \land B} \land_i
\]
\[
\frac{\Gamma \vdash A \land B}{\Gamma \vdash_{ne} A} \land_E
\]
\[
\frac{A, \Gamma \vdash C}{\Gamma \vdash_{ne} C} \lor_E
\]
\[
\frac{\Gamma \vdash A \Rightarrow B}{\Gamma \vdash_{ne} A} \Rightarrow_i
\]
\[
\frac{\Gamma \vdash A \Rightarrow B}{\Gamma \vdash_{ne} B} \Rightarrow_E
\]

- show that constructions are still valid
What had to be Done - 2

- works for first-order logic (probably more)
What had to be Done - 2

- works for first-order logic (probably more)
- formalize in Coq (propositional logic)
What had to be Done - 2

- works for first-order logic (probably more)
- **formalize** in Coq (propositional logic)
- **extract** the algorithm:
  - limitations of Coq
  - either we face proof-irrelevance
  - or universe inconsistency
What had to be Done - 2

- works for first-order logic (probably more)
- **formalize** in Coq (propositional logic)
- **extract** the algorithm:
  - limitations of Coq
  - either we face proof-irrelevance
  - or universe inconsistency
- we can at least observe inside Coq
- or have a potentially unsound algorithm
On Examples

- how a ⇒-cut is reduced

\[
\begin{align*}
A, A & \vdash A \\
\therefore A & \vdash A \quad \text{⇒I} \\
A & \vdash A \\
\end{align*}
\]

\[
\begin{align*}
A & \vdash A \quad \text{⇒E} \\
\end{align*}
\]
On Examples

- how a $\lor$-cut is reduced

\[
\frac{A \vdash A}{A \vdash A \lor A} \quad \text{ax) } \quad \frac{A, A \vdash A}{A, A \vdash A \lor A} \quad \text{\lor_l} \\
\frac{A \vdash A}{A \vdash A \lor A} \quad \text{\lor_r} \\
\frac{A, A \vdash A}{A, A \vdash A \lor A} \quad \text{\lor_l} \\
\frac{A \vdash A}{A \vdash A \lor A} \quad \text{\lor_r} \\
\frac{A \vdash A \lor A}{A \vdash A \lor A} \quad \text{\lor_e} \\
\frac{A \vdash A \lor A}{A \vdash A \lor A} \quad \text{\lor_l} \\
\frac{A \vdash A \lor A}{A \vdash A \lor A} \quad \text{\lor_r}
\]

O. Hermant (MINES ParisTech)  HDR – Complétude en logiques  2017, April 20th
On Examples

- $\eta$-expansion

$\vdash A \lor B \dashv A \lor B$
On Examples

- $\eta$-expansion

\[
\begin{align*}
A \lor B & \vdash A \lor B \\
A \lor B, A & \vdash A \\
A \lor B & \vdash A \lor B \\
\frac{A \lor B, A \vdash A}{A \lor B, A \lor B} & \text{ax} \\
\frac{A \lor B, B \vdash B}{A \lor B, A \lor B} & \text{ax}
\end{align*}
\]
On Examples

- $\eta$-expansion, one more step
Conclusion

**Computational content** of algebraic methods:
- still to explore
- commutative cuts

A lot of **domains**:
- logics with constraints (higher order)
- **polarized** Deduction Modulo Theory
  - model theory
  - theoretical results
  - tools
  - better Skolem symbols (rewriting)

This is first order, no **dependent types**
- $\lambda\Pi$-calculus Modulo Theory
- Dedukti