The subformula property in Intuitionistic sequents proof-search

Dominique Larchey-Wendling

LORIA – CNRS

Nancy, France

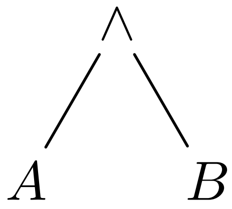
Overview of the talk

- We discuss proof/counter-model search IPL
- We deal with sequent calculi, old and new
- Presentation on the sub-formula property (SFP)
  - strict SFP, local rules (context untouched)
- Impact of the SFP (termination, complexity, indexation)
- Implementation issues
  - data structures for sequents and strategies
  - constant time rule application
- Transform the rules in the new system LSJ into local rules

Proof-search in the sequent calculus

- Left introduction rule for conjunction in IL (or CL)

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, \boxed{A \wedge B} \vdash \Delta} \; [\wedge_L]$$

$A \wedge B =$ 

- A, B (direct) subformulas of the principal formula $\boxed{A \wedge B}$
- Consequences of the SFP:
  - decreasing complexity: $\mathsf{size}(A) + \mathsf{size}(B) < \mathsf{size}(A \wedge B)$
  - bounded set of formulae occuring in (backward) proof-search
  - guaranteed termination of proof-search (for CL, not IL)

The sub-formula property (SFP)

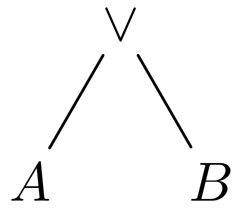
- Every formula introduced in backward proof-search is a sub-formula of the principal formula
- The SFP does not ensure termination (Gentzen LJ):

$$\frac{\Gamma, A \supset B \vdash A \qquad \Gamma, B \vdash C}{\Gamma, \boxed{A \supset B} \vdash C}\ [\supset_L] \qquad \cfrac{\cfrac{\cfrac{\text{loop} \quad \cdots}{A \supset B \vdash A \quad \cdots}}{A \supset B \vdash A \qquad \cdots}}{A \supset B \vdash A}$$

The strict sub-formula property (SSFP)

- The principal formula is *removed* and *replaced* by some of its (direct or strict) subformulae, with no duplications, e.g.

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, \boxed{A \vee B} \vdash \Delta}\ [\vee_L] \qquad A \vee B =$$



- In this case (e.g. CL), SSFP ensures termination:
  - size of sequents decreases from conclusions to premisses
  - proof-search depth linearly bounded by size of initial sequent
  - $\mathcal{O}(n \log n)$ space proof-search algorithm

SFP/SSFP not necessary for termination

- LJT, contraction free sequent calculus for IL (Dyckhoff 92)

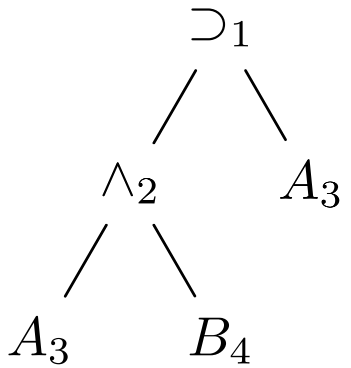
$$\frac{\Gamma, B \supset C \vdash A \supset B \qquad \Gamma, C \vdash D}{\Gamma, \boxed{(A \supset B) \supset C} \vdash D} \; [\supset_L^4]$$

- $B \supset C$ is not a subformula of $(A \supset B) \supset C$
- size$(B \supset C)$ + size$(A \supset B)$ is not lower than size$((A \supset B) \supset C)$
- but both $B \supset C$ and $A \supset B$ are strictly smaller than $(A \supset B) \supset C$
- the well-founded multiset ordering ensures termination

Application of SFP: indexation

- Associate a number to each subformula
- Structurally different subformulas should have different indexes
- Structurally identical subformulas can have the same index
- Identical variables should have the same index
- Proof-search on indexes

$$\dfrac{\dfrac{\dfrac{}{A, B \vdash A}\ \mathrm{Id}}{A \wedge B \vdash A}\ \wedge_L}{\vdash A \wedge B \supset A}\ \supset_R$$



$$\dfrac{\dfrac{\dfrac{}{3, 4 \vdash 3}\ \mathrm{Id}}{2 \vdash 3}\ \wedge_L}{\vdash 1}\ \supset_R$$

Recognizing axioms (the naive way)

- Axioms are usually of the form

$$\frac{}{\Gamma, A \vdash \Delta, A} \qquad \text{or} \qquad \frac{}{\Gamma \vdash \Delta}\ [\Gamma \cap \Delta \neq \emptyset]$$

- Complexity of naive implementation (e.g. lists):

$$\text{size}(\Gamma) \times \text{size}(\Delta) \times \text{size}(\text{average formula})$$

- Axioms should be tested at each step of proof-search
  - indeed, they might close/end the proof-search branch
- An efficient implementation of axioms recognition is thus crucial

Recognizing axioms (the indexed way)

- $\Gamma \vdash \Delta$ is indexed, e.g. $\vdash A_2 \wedge_3 B_4 \supset_1 A_3$
- Γ (resp. Δ) associated to a set of indexes (e.g. array of booleans)
- Each time Γ (or Δ) is modified, check for axiom (and mark)
- If $\Gamma \vdash \Delta$ not an axiom then $\Gamma - \{A\} \vdash \Delta$ not an axiom
  - $\Gamma, A \vdash \Delta$ axiom iff $A \in \Delta$ (e.g. $\Delta(A) = \text{true}$)
- To recognize axioms, check for the mark in constant time

$$\dfrac{\dfrac{\dfrac{}{A_3, B_4 \vdash A_3}\ \text{Id because 3}}{A_3 \wedge_2 B_4 \vdash A_3}\ \wedge_L \text{ and mark(3)}}{\vdash A_2 \wedge_3 B_4 \supset_1 A_3}\ \supset_R$$

How to select the rule to apply ?

- In the calculi we consider: select the principal formula

$$A_1, \ldots, \boxed{A_i}, \ldots, A_n \vdash B_1, \ldots, B_k$$

- Criteria for proof-search strategies:
  - lh/rh side, position in the list $A_1, \ldots, A_n$
  - outmost logical connective, complexity of the formula
- A "bad" choice may lead to failure:

$$\dfrac{\dfrac{\text{fails}}{A \vee B \vdash A}}{A \vee B \vdash \boxed{A \vee B}}\ \vee_R^1 \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A}\ \text{Id}}{A \vdash \boxed{A \vee B}}\ \vee_R^1 \quad \dfrac{\dfrac{}{B \vdash B}\ \text{Id}}{B \vdash \boxed{A \vee B}}\ \vee_R^2}{\boxed{A \vee B} \vdash A \vee B}\ \vee_L$$

Representation and update of sequents

- Γ and Δ, both as lists and sets of indexes;
- Update in constant time:

$$\frac{\Gamma \vdash \Delta_l, A_i, \Delta_r \qquad \cdots}{\Gamma \vdash \Delta_l, \boxed{A_i \wedge_k B_j}, \Delta_r} [\wedge_R]$$

- Remove the principal formula, insert one or two subformulae
- Beware *non-local rules* in STRIP (Larchey-W. et al. 2001)
  - all formulae $(\cdot) \supset C$ removed when decomposing $(A \supset B) \supset C$

$$\frac{\cdots \qquad\qquad \Gamma, C \vdash G}{\Gamma, \boxed{(A \supset B) \supset C}, D_1 \supset C, \ldots, D_k \supset C \vdash G} [\supset_L^4]$$

Constant time proof-search step

$\mathrm{PS}(\Gamma, \boxed{A \vee B} \vdash \Delta) =$

1. replace $A \vee B$ by A, push $(A \leadsto A \vee B)$
2. result $=$ $\mathrm{PS}(\Gamma, A \vdash \Delta)$ (recursion)
3. pop $(A \leadsto A \vee B)$, replace A by $A \vee B$
4. if result $=$ fail then return fail
5. replace $A \vee B$ by B, push $(B \leadsto A \vee B)$
6. result $=$ $\mathrm{PS}(\Gamma, B \vdash \Delta)$ (recursion)
7. pop $(B \leadsto A \vee B)$, replace B by $A \vee B$
8. return result

$$\frac{\Gamma, A \vdash \Delta \qquad \Gamma, B \vdash \Delta}{\Gamma, \boxed{A \vee B} \vdash \Delta}$$

Terminating proof-search for IPL

- From Gentzen (LJ) to Dyckhoff 92 (LJT) and Hudelmaier 93
- Dyckhoff & Pinto 96 (LJT/CRIP), Dyckhoff & Negri 2000
- Formalization: Weich 98 (Coq, extraction)
- Larchey-Wendling et al. 2001 (STRIP)
- Fiorino et al. 2000+ (tableaux variants of LJT)

- One of our longstanding problem: certified STRIP
- A new lead: the new system LSJ with SSFP
- Our contribution: optimize LSJ for indexed proof search

A sequent system for IPL with SSFP

"Contraction-free Linear Depth Sequent Calculi for IPL with the Subformula Property and Minimal Depth Counter-Models"

(Ferrari, Fiorentini and Fiorino, to appear in JAR)

- A new system LSJ with sequents of the form $\Theta \mid \Gamma \vdash \Delta$
- A finite refutation semantics: $\mathcal{T}$ refutes $\Theta \mid \Gamma \vdash \Delta$ if

$$\mathcal{T} \Vdash_s \Theta \quad \text{and} \quad \mathcal{T} \Vdash \Gamma \quad \text{and} \quad \mathcal{T} \nVdash \Delta$$

- Recover semantics for formulae: $\mathcal{T} \nVdash A$ iff $\mathcal{T}$ refutes $\emptyset \mid \emptyset \vdash A$
- A valid sequent has no refutation tree: $\emptyset \mid \emptyset \vdash A$ valid iff A valid

Finite Kripke semantics for IPL

- Var = set of propositional variables
- A tree: $\mathcal{T} = (S_{\mathcal{T}}, [\mathcal{T}_1, \ldots, \mathcal{T}_k])$, with $S_{\mathcal{T}} \subseteq_f$ Var
- A Kripke tree = monotonicity for all subtrees : $S_{\mathcal{T}} \subseteq S_{\mathcal{T}_i}$
- Subtree ($\leqslant$): $\mathcal{T} \leqslant \mathcal{T}$ and $\mathcal{T}' \leqslant \mathcal{T}_i$ implies $\mathcal{T}' \leqslant \mathcal{T}$
- Strict subtree ($<$): $\mathcal{T}' \leqslant \mathcal{T}_i$ implies $\mathcal{T}' < \mathcal{T}$
- Monotonic Kripke semantics: $\forall \mathcal{T}' \leqslant \mathcal{T},\ \mathcal{T} \Vdash A \Rightarrow \mathcal{T}' \Vdash A$

$$\begin{aligned} \mathcal{T} \Vdash A \supset B \quad &\Leftrightarrow \quad \forall \mathcal{T}' \leqslant \mathcal{T},\ \mathcal{T}' \Vdash A \Rightarrow \mathcal{T}' \Vdash B \\ \mathcal{T} \Vdash_s A \quad &\Leftrightarrow \quad \forall \mathcal{T}' < \mathcal{T},\ \mathcal{T}' \Vdash A \end{aligned}$$

- This is a sound and complete semantics for IPL

The rules of LSJ (implicational fragment)

- Formulae in Θ are not active
- But they are activated by rightmost premisse of $[\supset_L]$ and $[\supset_R]$
- Strict sub-formula property (SSFP), but some rules are not local

$$\frac{}{\Theta \mid \Gamma, A \vdash A, \Delta}\ [\mathrm{Id}] \qquad \frac{\Theta \mid A, \Gamma \vdash B, \Delta \qquad \emptyset \mid A, \Theta, \Gamma \vdash B}{\Theta \mid \Gamma \vdash \boxed{A \supset B}, \Delta}\ [\supset_R]$$

$$\frac{\Theta \mid B, \Gamma \vdash \Delta \qquad B, \Theta \mid \Gamma \vdash A, \Delta \qquad B \mid \Theta, \Gamma \vdash A}{\Theta \mid \boxed{A \supset B}, \Gamma \vdash \Delta}\ [\supset_L]$$

Sound and completeness for LSJ

- Soundness for LSJ
  - if $\mathcal{T}$ refutes the conclusion of some rule then there exists $\mathcal{T}' \leqslant \mathcal{T}$ that refutes one premisse of the rule
  - axioms have no refutation trees
  - hence no tree refutes a provable sequent
  - also impacts the depth of counter-models
- Completeness for LSJ
  - a dual refutation calculus RJ
  - extract a refutation tree from any (dual) proof in RJ
  - algorithm that builds either a LSJ-proof or (dual) RJ-proof
  - in the spirit of LJT/CRIP (Pinto & Dyckhoff 95)

LSJ rules are not local rules

- Formulas of Θ are moved in Γ
- Formulas of Δ are removed all together
- Hence rules touch the context

$$\frac{\cdots \qquad \cdots \qquad B \mid \boxed{\Theta}, \Gamma \vdash A}{\Theta \mid A \supset B, \Gamma \vdash \boxed{\Delta}} \; [\supset_L]$$

$$\frac{\cdots \qquad \emptyset \mid A, \boxed{\Theta}, \Gamma \vdash B}{\Theta \mid \Gamma \vdash A \supset B, \boxed{\Delta}} \; [\supset_R]$$

- Our solution: refinement of LSJ into an indexed version

How to cope with Θ and Δ: indexed sequents

- Let $n_1, \ldots, n_r, p_1, \ldots, p_k$ be non-negative integers
- Let $\Sigma = n_1 : A_1, \ldots, n_r : A_r$ and $\Omega = p_1 : B_1, \ldots, p_s : B_s$
- $\Sigma \vdash_n^p \Omega$ is an indexed sequent if
  - n and p are non-negative integers and
  - $n_i \leqslant n + 1$ and $p_j \leqslant p$ for any i, j
- Associated LSJ sequent $\Theta \mid \Gamma \vdash \Delta$ with

$$\Theta = \{A_i \mid n_i = n + 1\} \quad \Gamma = \{A_i \mid n_i \leqslant n\} \quad \Delta = \{B_j \mid p = p_j\}$$

- We propose an indexed sequent calculus associated to LSJ

Indexed LSJ (part one)

$$\frac{}{\Theta \mid \Gamma, A \vdash A, \Delta}\ [\mathrm{Id}]$$

$$\frac{}{i : A, \Sigma \vdash_n^p \Omega, p : A \quad \text{with } i \leqslant n}$$

Indexed LSJ (part two)

$$\frac{\Theta \mid A, \Gamma \vdash B, \Delta \qquad \emptyset \mid A, \Theta, \Gamma \vdash B}{\Theta \mid \Gamma \vdash \Delta, A \supset B}\ [\supset_R]$$

$$\frac{n : A, \Sigma \vdash_n^p \Omega, p : B \qquad n+1 : A, \Sigma \vdash_{n+1}^{p+1} \Omega, p+1 : B}{\Sigma \vdash_n^p \Omega, p : A \supset B}$$

Indexed LSJ (part three)

$$\frac{\Theta \mid B, \Gamma \vdash \Delta \qquad B, \Theta \mid \Gamma \vdash A, \Delta \qquad B \mid \Theta, \Gamma \vdash A}{\Theta \mid A \supset B, \Gamma \vdash \Delta}\ [\supset_L]$$

$$\frac{i : B, \Sigma \vdash_n^p \Omega \qquad n+1 : B, \Sigma \vdash_n^p \Omega, p : A \qquad n+2 : B, \Sigma \vdash_{n+1}^{p+1} \Omega, p+1 : A}{i : A \supset B, \Sigma \vdash_n^p \Omega \quad \text{with } i \leqslant n}$$

Properties of the indexed LSJ sequent calculus

- Has the SSFP, and thus terminates
- Sound and complete for IPL (as LSJ) (also counter-models)
- Local rules: context is preserved by rule application
- Each rule application implies a bounded number of operations
  - one removal, and one or two introductions
  - rules can be applied in constant time
- As with LSJ (unlike STRIP), manageable formalization (Coq)

Conclusion

- A new indexed sequent calculus for IPL based on LSJ
- Well suited for the implementation of proof-search (local SSFP)
- Soundness & completeness proved formally

Perspectives

- A certified indexed proof-search engine for IPL (Coq, extraction)
- Certified compilation of proof-search in IPL, potentially as efficient as STRIP