

Cut-elimination and Proof-search for Bi-Intuitionistic Logic Using Nested Sequents

Rajeev Goré Linda Postniece Alwen Tiu

Computer Sciences Laboratory
The Australian National University

Estonian CS Theory Days
Jõulumäe, 3-5 October 2008

Introduction: Sequent Calculus and Proof Search

- **Sequent calculus**: proof system
- **Sequent**: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \rightarrow \bigvee \Delta$

- **Derivation**: tree of sequents

$$\text{e.g. } \frac{\frac{A, B \vdash A \quad A, B \vdash B}{A, B \vdash A \wedge B} \wedge_R}{A \vdash B \rightarrow (A \wedge B)} \rightarrow_R$$

- Leaf nodes: axioms
- Root: logical consequence to be proven
- Rules link nodes
- **Backward proof search**: build derivation from root towards leaves

- **Cut rule**: encodes transitivity

$$\frac{\Gamma_1 \vdash \Delta_1, A \quad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$

- Non-deterministic when applied backwards
- **Cut-elimination**: proof that uses cut \rightsquigarrow proof that does not
 - If $\Gamma \vdash \Delta$ derivable using cut then $\Gamma \vdash \Delta$ derivable without using cut

Introduction: Bi-Intuitionistic logic

- Bi-Intuitionistic logic: extension of intuitionistic logic
- Exclusion connective \multimap dual to implication \rightarrow :

$$\bullet \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_R \qquad \frac{A \vdash B, \Delta}{A \multimap B \vdash \Delta} \multimap_L$$

- Hilbert calculus, algebraic and Kripke semantics (Rauszer 1974)
- Type theoretic interpretation of co-routines (Crolard 2004)
- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- More recent calculi either have:
 - cut-elimination (Goré 1998), or
 - proof search (Uustalu, Pinto draft; Buisman/Postniece, Goré 2007)
 - but not both
- Goal: a sequent calculus for BiInt that has both cut-elimination and proof search

- 1 Bi-Intuitionistic Logic
 - Syntax and Semantics
 - Bilnt Challenges

- 2 LBilnt
 - Nested Sequents
 - Sequent Rules
 - Soundness

- 3 Proof Search
 - Strategy
 - Termination
 - Completeness

- 4 Conclusion

Syntax and Kripke Models

- Connectives: $\wedge \vee \rightarrow \multimap$
- Constants: $\top \perp$
- Kripke semantics: model $\langle W, \leq, V \rangle$
 - \leq is a reflexive and transitive binary relation over W
 - V maps atoms to 2^W
 - $V(\top) = W$ and $V(\perp) = \emptyset$
 - V satisfies: if $w \in V(p)$ and $w \leq u$ then $u \in V(p)$
- $w \Vdash p$ iff $w \in V(p)$
- $w \Vdash A \wedge B$ iff $w \Vdash A$ & $w \Vdash B$
- $w \Vdash A \vee B$ iff $w \Vdash A$ or $w \Vdash B$
- $w \Vdash A \rightarrow B$ iff $\forall u \geq w. u \not\Vdash A$ or $u \Vdash B$
- $w \Vdash A \multimap B$ iff $\exists u \leq w. u \Vdash A$ & $u \not\Vdash B$
- BIInt is a conservative extension of Int
- Persistence: $w \Vdash A \Rightarrow \forall u \geq w. u \Vdash A$
- Reverse Persistence: $w \not\Vdash A \Rightarrow \forall u \leq w. u \not\Vdash A$

Validity and Falsifiability

- A **valid** iff **for all** models $\langle W, \leq, V \rangle$ and all worlds $w \in W$, $w \Vdash A$
- A **falsifiable** iff **exists** a model $\langle W, \leq, V \rangle$, a world $w \in W$, $w \not\Vdash A$
- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \not\Vdash \Delta$ iff $w \not\Vdash A$ for all $A \in \Delta$
- $\Gamma \vdash \Delta$ **valid** iff $\bigwedge \Gamma \rightarrow \bigvee \Delta$ valid
 - i.e. **for all** models $\langle W, \leq, V \rangle$ and all worlds $w \in W$, $w \Vdash \Gamma$ implies $w \Vdash B$ for some $B \in \Delta$
- $\Gamma \vdash \Delta$ **falsifiable** iff $\bigwedge \Gamma \rightarrow \bigvee \Delta$ falsifiable
 - i.e. **exists** a model $\langle W, \leq, V \rangle$, a world $w \in W$, $w \Vdash \Gamma$ and $w \not\Vdash \Delta$

Uustalu's Example: Using Cut

- Rauszer's \rightarrow_R and \leftarrow_L require singleton succedent/antecedent:

$$\bullet \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_R \quad \frac{A \vdash B, \Delta}{A \leftarrow B \vdash \Delta} \leftarrow_L$$

- $p \vdash q, r \rightarrow ((p \leftarrow q) \wedge r)$ is not cut-free derivable in Rauszer's G1
- Failed derivation attempt:

$$\frac{\frac{\frac{}{p \vdash p} \text{Id}}{p, r \vdash p \leftarrow q} \leftarrow_R \quad \frac{}{p, r \vdash r} \text{Id}}{p, r \vdash (p \leftarrow q) \wedge r} \wedge_R}{\frac{p \vdash r \rightarrow ((p \leftarrow q) \wedge r)}{p \vdash q, r \rightarrow ((p \leftarrow q) \wedge r)} \rightarrow_R} \text{WR}$$

- Derivation using cut:

$$\frac{\frac{\frac{}{p \vdash q, p} \text{Id}}{p \vdash q, p \leftarrow q} \leftarrow_R \quad \frac{}{q \vdash q} \text{Id}}{p \vdash q, r \rightarrow ((p \leftarrow q) \wedge r)} \text{cut} \quad \frac{\frac{\frac{}{p \leftarrow q, r \vdash p \leftarrow q} \text{Id}}{p \leftarrow q, r \vdash (p \leftarrow q) \wedge r} \wedge_R}{p \leftarrow q \vdash r \rightarrow ((p \leftarrow q) \wedge r)} \rightarrow_R}}{p \vdash q, r \rightarrow ((p \leftarrow q) \wedge r)} \text{cut}$$

Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$

Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$

(w)

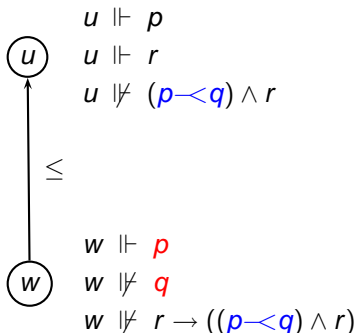
$w \Vdash p$

$w \not\Vdash q$

$w \not\Vdash r \rightarrow ((p \multimap q) \wedge r)$

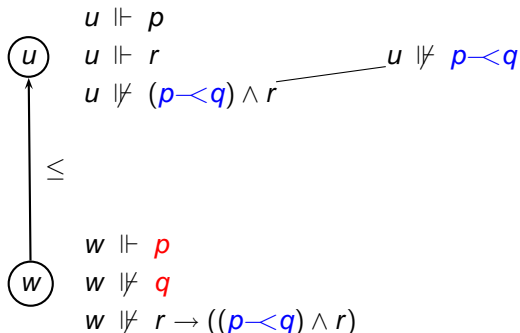
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$



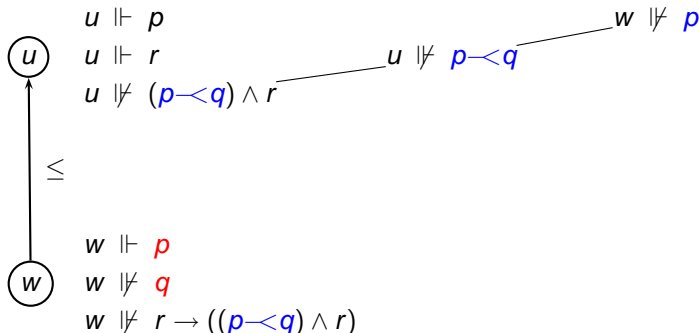
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$



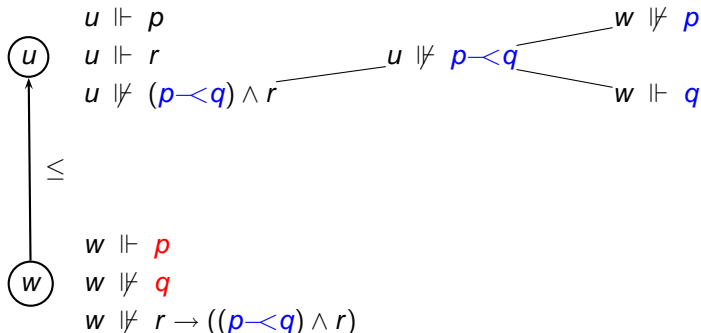
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$



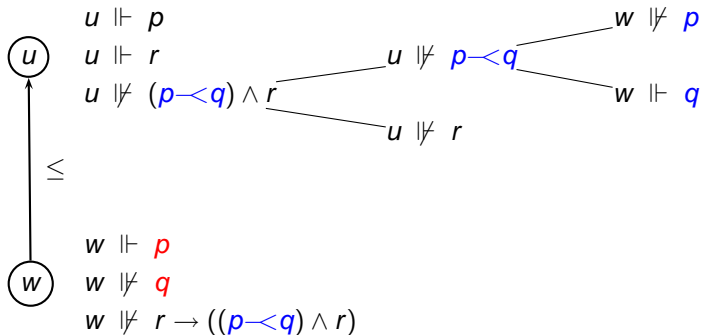
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$



Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \multimap q) \wedge r)$ is valid
- Counter-model construction
 - $w \Vdash p$
 - $w \not\Vdash q, r \rightarrow ((p \multimap q) \wedge r)$



Nested Sequents

- Negative structures: $N := \emptyset \mid A \mid (N, N) \mid N < P$
- Positive structures: $P := \emptyset \mid A \mid (P, P) \mid N > P$
- Sequents: $X \vdash Y$ where
 - X is a negative structure
 - Y is a positive structure
- $\{X\}$ denote top-level formulae of X
- Examples:
 - $r \vdash p < q$
 - $(p < q), r \vdash (q > r), ((p < q) > w)$
 - $\{(p < q), r, s\} = \{r, s\}$

LBilnt Rules

$$\frac{}{X, A \vdash A, Y} \textit{id}$$

LBilnt Rules

$$\overline{X, A \vdash A, Y} \text{ id}$$

$$\frac{X, A \rightarrow B \vdash A, Y \quad X, A \rightarrow B, B \vdash Y}{X, A \rightarrow B \vdash Y} \rightarrow_L$$

LBilnt Rules

$$\overline{X, A \vdash A, Y} \text{ id}$$

$$\frac{X, A \rightarrow B \vdash A, Y \quad X, A \rightarrow B, B \vdash Y}{X, A \rightarrow B \vdash Y} \rightarrow_L$$

$$\frac{X \vdash A, A \multimap B, Y \quad X, B \vdash A \multimap B, Y}{X \vdash A \multimap B, Y} \multimap_R$$

LBilnt Rules

$$\overline{X, A \vdash A, Y} \text{ id}$$

$$\frac{X, A \rightarrow B \vdash A, Y \quad X, A \rightarrow B, B \vdash Y}{X, A \rightarrow B \vdash Y} \rightarrow_L$$

$$\frac{X \vdash A, A \prec B, Y \quad X, B \vdash A \prec B, Y}{X \vdash A \prec B, Y} \prec_R$$

$$\frac{X_2 \vdash Y_2, \{Y_1\}}{X_1, (X_2 < Y_2) \vdash Y_1} < \{Y_1\} \not\subseteq \{Y_2\} \quad \frac{\{X_1\}, X_2 \vdash Y_2}{X_1 \vdash Y_1, (X_2 > Y_2)} > \{X_1\} \not\subseteq \{X_2\}$$

LBilnt Rules

$$\overline{X, A \vdash A, Y} \text{ id}$$

$$\frac{X, A \rightarrow B \vdash A, Y \quad X, A \rightarrow B, B \vdash Y}{X, A \rightarrow B \vdash Y} \rightarrow_L$$

$$\frac{X \vdash A, A \prec B, Y \quad X, B \vdash A \prec B, Y}{X \vdash A \prec B, Y} \prec_R$$

$$\frac{X_2 \vdash Y_2, \{Y_1\}}{X_1, (X_2 < Y_2) \vdash Y_1} < \{Y_1\} \not\subseteq \{Y_2\}$$

$$\frac{\{X_1\}, X_2 \vdash Y_2}{X_1 \vdash Y_1, (X_2 > Y_2)} > \{X_1\} \not\subseteq \{X_2\}$$

$$\frac{A \vdash B, \{Y\}, (X, A \prec B > Y)}{X, A \prec B \vdash Y} \prec_{L2}$$

$$\frac{(X < Y, A \rightarrow B), \{X\}, A \vdash B}{X \vdash Y, A \rightarrow B} \rightarrow_{R2}$$

Uustalu's Example Revisited

Using cut:

$$\frac{\frac{\frac{}{p \vdash q, p} \text{Id}}{p \vdash q, p \prec q} \text{Id} \quad \frac{\frac{}{q \vdash q} \text{Id}}{p \vdash q, p \prec q} \prec_R}{p \vdash q, r \rightarrow ((p \prec q) \wedge r)} \text{cut} \quad \frac{\frac{\frac{\frac{}{p \prec q, r \vdash p \prec q} \text{Id}}{p \prec q, r \vdash (p \prec q) \wedge r} \text{Id}}{p \prec q \vdash r \rightarrow ((p \prec q) \wedge r)} \rightarrow_R}{p \vdash q, r \rightarrow ((p \prec q) \wedge r)} \wedge_R}{p \vdash q, r \rightarrow ((p \prec q) \wedge r)} \text{cut}$$

Using LBIInt:

$$\frac{\frac{\frac{}{p \vdash q, \dots, p} \text{Id}}{p \vdash q, \dots, p \prec q} \text{Id} \quad \frac{\frac{}{p, q \vdash q, \dots} \text{Id}}{p, p \vdash q, \dots} \prec_R}{(p \prec q, \dots), p, r \vdash p \prec q} \prec \quad \frac{\frac{}{(p \prec q, \dots), p, r \vdash r} \text{Id}}{(p \prec q, \dots), p, r \vdash (p \prec q) \wedge r} \wedge_R}{p \vdash q, r \rightarrow ((p \prec q) \wedge r)} \rightarrow_{R2}$$

Save/Restore

- Rauszer's G1 vs LBIInt:

Lose context:

$$\frac{X, A \vdash B}{X \vdash A \rightarrow B} \rightarrow_R \quad \frac{}{X \vdash Y, A \rightarrow B} W_R$$

Save context:

$$\frac{(X < Y, A \rightarrow B), \{X\}, A \vdash B}{X \vdash Y, A \rightarrow B} \rightarrow_{R2}$$

- Restore context:

$$\frac{X_2 \vdash Y_2, \{Y_1\}}{X_1, (X_2 < Y_2) \vdash Y_1} < \{Y_1\} \not\subseteq \{Y_2\}$$

- Cut-elimination: instance of cut \rightsquigarrow pair of save/restore rules

Soundness

- Relates syntax to semantics
- Formula-translation τ maps sequents to formulae

$$\begin{array}{llll}
 \tau_N(\emptyset) & = & \top & \tau_P(\emptyset) & = & \perp \\
 \tau_N(A) & = & A & \tau_P(A) & = & A \\
 \tau_N(X, Y) & = & \tau_N(X) \wedge \tau_N(Y) & \tau_P(X, Y) & = & \tau_P(X) \vee \tau_P(Y) \\
 \tau_N(X < Y) & = & \tau_N(X) \multimap \tau_P(Y) & \tau_P(X > Y) & = & \tau_N(X) \rightarrow \tau_P(Y)
 \end{array}$$

- $<$ is a proxy for \multimap
- $>$ is a proxy for \rightarrow
- Soundness: for every rule, show:
 - if τ of premises is valid then τ of conclusion is valid

Saturation and Blocking

Definition

We classify the rules of LBIInt_2 into three groups:

Static Rules: = $\{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \rightarrow_L, \neg_R, \neg_{L1}, \rightarrow_{R1}\}$;

Jump Rules: = $\{\neg_{L2}, \rightarrow_{R2}\}$; and

Return Rules: = $\{<, >\}$.

Saturation and Blocking

Definition

We classify the rules of LBIInt_2 into three groups:

Static Rules: $= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \rightarrow_L, \neg\langle_R, \neg\langle_{L1}, \rightarrow_{R1}\}$;

Jump Rules: $= \{\neg\langle_{L2}, \rightarrow_{R2}\}$; and

Return Rules: $= \{\langle, \rangle\}$.

We call a sequence of static rule applications a *saturation*.

Definition

A sequent $X_0 \vdash Y_0$ is **saturated** if applying a static rule would yield at least one premise $X_1 \vdash Y_1$ such that $\{X_1\} \subseteq \{X_0\}$ and $\{Y_1\} \subseteq \{Y_0\}$.

Saturation and Blocking

Definition

We classify the rules of LBilnt_2 into three groups:

Static Rules: $= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \rightarrow_L, \leftarrow_R, \leftarrow_{L1}, \rightarrow_{R1}\};$

Jump Rules: $= \{\leftarrow_{L2}, \rightarrow_{R2}\};$ and

Return Rules: $= \{<, >\}.$

We call a sequence of static rule applications a *saturation*.

Definition

A sequent $X_0 \vdash Y_0$ is **saturated** if applying a static rule would yield at least one premise $X_1 \vdash Y_1$ such that $\{X_1\} \subseteq \{X_0\}$ and $\{Y_1\} \subseteq \{Y_0\}$.

Definition

A LBilnt_2 rule ρ is **applicable** to a sequent $X_0 \vdash Y_0$ if for every premise $X_i \vdash Y_i$ of ρ , $\{X_i\} \not\subseteq \{X_0\}$ or $\{Y_i\} \not\subseteq \{Y_0\}$.

Corollary

Only jump and return rules are applicable to saturated sequents.

Proof Search Strategy

Function Prove

Input: sequent γ_0

Output: *true* (i.e. γ_0 is derivable) or *false* (i.e. γ_0 is not derivable)

- 1 If *id* is applicable to γ_0 then return *true*
- 2 Else if a static rule ρ is applicable to γ_0 then
 - 1 Let $\gamma_1, \dots, \gamma_n$ be the premises of ρ obtained from γ_0
 - 2 Return $\bigwedge_{i=1}^n \text{Prove}(\gamma_i)$
- 3 Else if $\text{Prove}(\gamma_1) = \text{true}$ for some premise instance γ_1 obtained from γ_0 by applying $\rho \in \{\leftarrow_{L2}, \rightarrow_{R2}, <, >\}$ then return *true*
- 4 Else return *false*.

Linear Sequents

Definition

- 1 If Γ/Δ are sets of formulae, then $\Gamma \vdash \Delta$ is a linear sequent.
- 2 If $X \vdash Y$ is a linear sequent and Γ/Δ are sets of formulae, then
 - 1 $(X < Y), \Gamma \vdash \Delta$ and
 - 2 $\Gamma \vdash \Delta, (X > Y)$
 are linear sequents.

Example

$$C \vdash B, A \rightarrow B$$

$$(C < B, A \rightarrow B), C, A \vdash B$$

$$D \vdash E, ((C < B, A \rightarrow B), C, A > B)$$

Lemma

Every LBIInt-derivation of a linear end-sequent contains only linear sequents.

Lists

Definition (Linear Sequent to List)

$$\begin{aligned}
 list(\Gamma \vdash \Delta) &= \langle \Gamma, \Delta \rangle \\
 list((X < Y), \Gamma \vdash \Delta) &= list(X \vdash Y) \leq \langle \Gamma, \Delta \rangle \\
 list(\Gamma \vdash \Delta, (X > Y)) &= list(X \vdash Y) \geq \langle \Gamma, \Delta \rangle
 \end{aligned}$$

Example

$$\begin{aligned}
 list((C < B, A \rightarrow B), C, A \vdash B) &= list(C \vdash B, A \rightarrow B) \leq \langle \{C, A\}, \{B\} \rangle \\
 &= \langle \{C\}, \{B, A \rightarrow B\} \rangle \leq \langle \{C, A\}, \{B\} \rangle
 \end{aligned}$$

Remark

Lists encoded in sequents \rightsquigarrow branches in counter-model.

List Operations

Corollary

A backward $LBIInt_2$ rule application to a linear sequent $X \vdash Y$ can be viewed as an operation on $list(X \vdash Y)$:

- Conclusion/premise is the list before/after the operation
- Jump rules: append a node to the list
- Static rules: saturate the end node
- Return rules: remove end node, update penultimate node.

Example

$$\frac{(C < B, A \rightarrow B), C, A \vdash B}{C \vdash B, A \rightarrow B} \rightarrow_{R2} \frac{\langle \{C\}, \{B, A \rightarrow B\} \rangle \leq \langle \{C, A\}, \{B\} \rangle}{\langle \{C\}, \{B, A \rightarrow B\} \rangle}$$

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, $\text{list}(X \vdash Y)$ can be extended at most $\mathcal{O}(m^2)$ times.

Lemma (Saturation)

Let $X \vdash Y$ be any sequent encountered during proof search. Then the saturation process for $X \vdash Y$ terminates after $\mathcal{O}(m)$ steps.

Theorem

The proof search strategy terminates.

Proof.

- Nodes/lists are bounded in size/length
- Jump/return rules cannot repeatedly create/remove nodes
 - Every update adds one more (sub)formula to a node
 - Eventually no subformulae can be added to any node
 - Return rules are blocked

Countermodel Construction

- Want to show: if A valid then $\emptyset \vdash A$ derivable
- Usually show contrapositive:
 - if $\emptyset \vdash A$ not derivable then A not valid
 - if $\text{Prove}(\emptyset \vdash A) = \text{false}$, then $\emptyset \vdash A$ has a counter-model
- More generally:
 - if $\text{Prove}(\Gamma \vdash \Delta) = \text{false}$, then $w \Vdash \Gamma$ and $w \not\Vdash \Delta$ for some $\langle W, \leq, V \rangle$ and $w \in W$
- Extract countermodel from failed proof search
- Usually done inductively by combining sub-models
- We use the notion of a pre-model

Pre-model: a “model in progress”

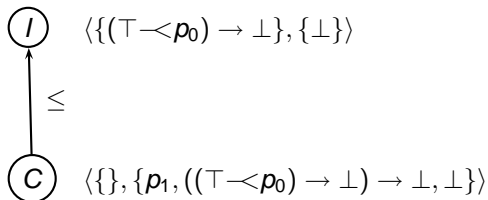
- Tree of nodes
 - Nodes are pairs $\langle \Gamma, \Delta \rangle$ of sets of formulae
 - Nodes are linked by labels \leq or \geq
- Nodes are marked either I or C
 - Each inner node is marked C
 - Each C-node is saturated
- Satisfies “pre-model properties”
- When all nodes are marked C, the root has a counter-model
 - By induction on the size of the pre-model
 - Pre-model properties ensure forcing and persistence
- Aim: build a pre-model with all nodes marked C
 - Use failed proof search to guide us

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $Prove(\emptyset \vdash A) = false$
- Use failed proof search to update pre-model
 - Static rules: saturate leaf nodes
 - Jump rules: mark node C, create new I-nodes
 - Return rules: remove I-node, update previous node and mark it I
 - No rule applicable: mark leaf C
- Concatenate lists obtained from linear sequents to obtain a tree
- When all leaves are marked C, the root has a model
 - $w \not\models \{A\}$ for some $\langle W, \leq, V \rangle$ and $w \in W$
 - A is not valid
- By contrapositive, if A is valid, then $Prove(\emptyset \vdash A) = true$

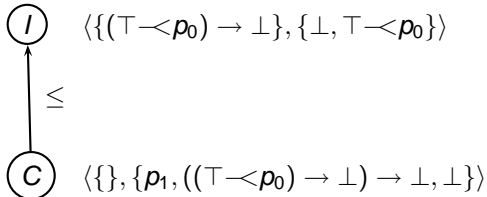
Restart Example

$$\begin{array}{c}
 \vdots \\
 \frac{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_L \\
 \vdots \\
 \frac{\vdots}{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \rightarrow_{R2} \\
 \frac{\vdots}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \multimap_R \\
 \frac{\vdots}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0} \langle \\
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_{R2} \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp} \rightarrow_{R1}
 \end{array}$$



Restart Example

$$\begin{array}{c}
 \vdots \\
 \frac{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_L \\
 \vdots \\
 \frac{\vdots}{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \rightarrow_{R2} \\
 \frac{\vdots}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \multimap_R \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp, \top \multimap p_0} < \\
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp, \top \multimap p_0} \rightarrow_L \\
 \frac{\vdots}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp} \rightarrow_{R2} \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp} \rightarrow_{R1}
 \end{array}$$



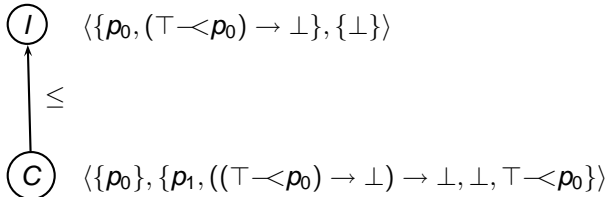
Restart Example

$$\begin{array}{c}
 \vdots \\
 \frac{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_L \\
 \vdots \\
 \frac{\vdots}{\vdots} \\
 \frac{\vdots}{\vdots} \\
 \frac{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \multimap_R \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0} < \\
 \vdots \\
 \frac{\vdots}{\vdots} \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_{R2} \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp} \rightarrow_{R1}
 \end{array}$$

$$\textcircled{1} \quad \langle \{\}, \{p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0\} \rangle$$

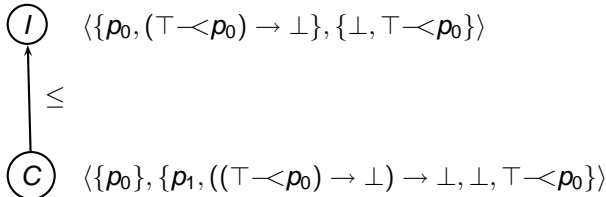
Restart Example

$$\begin{array}{c}
 \vdots \\
 \frac{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_L \\
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp}{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \rightarrow_{R2} \\
 \vdots \\
 \frac{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \multimap_R \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0} < \\
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, (\top \multimap p_0) \rightarrow \perp \vdash \perp}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp} \rightarrow_L \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp} \rightarrow_{R2} \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp} \rightarrow_{R1}
 \end{array}$$



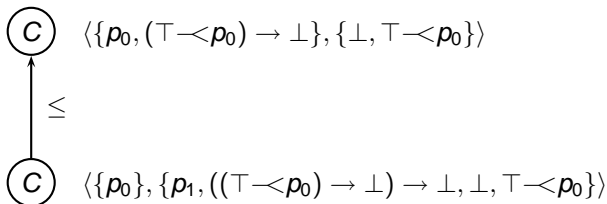
Restart Example

$$\begin{array}{c}
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0 \rangle}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp \rangle} \rightarrow_L \\
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp \rangle}{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \rightarrow_{R2} \\
 \vdots \\
 \frac{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \multimap_R \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0 \rangle}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp, \top \multimap p_0 \rangle} < \\
 \vdots \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp, \top \multimap p_0 \rangle}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp, \perp \rangle} \rightarrow_L \\
 \frac{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \perp, \perp \rangle}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp} \rightarrow_{R2} \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp} \rightarrow_{R1}
 \end{array}$$



Restart Example

$$\begin{array}{c}
 \vdots \\
 \frac{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0), p_0, (\top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_L \\
 \vdots \\
 \frac{\vdots}{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \rightarrow_{R2} \\
 \frac{p_0 \vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0} \multimap_R \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0}{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0} < \\
 \vdots \\
 \frac{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0) \rightarrow \perp \vdash \perp, \top \multimap p_0}{(\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_L \\
 \frac{\vdots}{\langle p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp, \top \multimap p_0) \rightarrow \perp \vdash \perp} \rightarrow_{R2} \\
 \frac{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp, \perp}{\vdash p_1, ((\top \multimap p_0) \rightarrow \perp) \rightarrow \perp} \rightarrow_{R1}
 \end{array}$$



Conclusions and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
 - a cut-free sequent calculus LBIInt
 - a complete, terminating proof search strategy
- Further work:
 - Generalise LBIInt to other similar logics (KtS4, S5, bi-Lambek, ...)
 - Explore connection with type theory