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

Institute of Cybernetics
2 October 2008

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system

Introduction: Sequent Calculus and Proof Search

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

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g. $$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \to (A \wedge B)}\to_R$$

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g. $$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \to (A \wedge B)}\to_R$$
    - Leaf nodes: axioms

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 e.g.

$$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \rightarrow (A \wedge B)}\rightarrow_R$$

  - Leaf nodes: axioms
  - Root: logical consequence to be proven

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g. $$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \to (A \wedge B)}\to_R$$
  - Leaf nodes: axioms
  - Root: logical consequence to be proven
  - Rules link nodes

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g. $$\cfrac{\cfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \to (A \wedge B)}\to_R$$
  - Leaf nodes: axioms
  - Root: logical consequence to be proven
  - Rules link nodes
- Backward proof search: build derivation from root towards leaves

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g. $$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \to (A \wedge B)}\to_R$$
  - 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 $$\dfrac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g. $$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \to (A \wedge B)}\to_R$$
  - 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 $$\dfrac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$
  - Non-deterministic when applied backwards

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 e.g. $$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \wedge B}\wedge_R}{A \vdash B \rightarrow (A \wedge B)}\rightarrow_R$$
  - 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 $$\dfrac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$
  - Non-deterministic when applied backwards
- Cut-elimination: proof that uses cut $\rightsquigarrow$ proof that does not

Introduction: Sequent Calculus and Proof Search

- Sequent calculus: proof system
- Sequent: $\Gamma \vdash \Delta$ read as $\bigwedge \Gamma \to \bigvee \Delta$
- Derivation: tree of sequents e.g.

$$\dfrac{\dfrac{A, B \vdash A \qquad A, B \vdash B}{A, B \vdash A \land B}\land_R}{A \vdash B \to (A \land B)}\to_R$$

  - 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

$$\dfrac{\Gamma_1 \vdash \Delta_1, A \qquad A, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}$$

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

Introduction: Bi-Intuitionistic Logic

- Extension of intuitionistic logic with exclusion connective $\prec$

Introduction: Bi-Intuitionistic Logic

- Extension of intuitionistic logic with exclusion connective $\prec$
- Sequent calculus rules for exclusion dual to implication:

Introduction: Bi-Intuitionistic Logic

- Extension of intuitionistic logic with exclusion connective $-\!\!<$
- Sequent calculus rules for exclusion dual to implication:
  - $\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \to_R \qquad \dfrac{A \vdash B, \Delta}{A -\!\!< B \vdash \Delta} -\!\!<_L$

Introduction: Bi-Intuitionistic Logic

- Extension of intuitionistic logic with exclusion connective $-\!\!<$
- Sequent calculus rules for exclusion dual to implication:
  - $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_R \qquad \frac{A \vdash B, \Delta}{A -\!\!< B \vdash \Delta} -\!\!<_L$$
- Hilbert calculus, algebraic and Kripke semantics (Rauszer 1974)

Introduction: Bi-Intuitionistic Logic

- Extension of intuitionistic logic with exclusion connective $\prec$
- Sequent calculus rules for exclusion dual to implication:
  - $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \to_R \qquad \frac{A \vdash B, \Delta}{A \prec B \vdash \Delta} \prec_L$$
- Hilbert calculus, algebraic and Kripke semantics (Rauszer 1974)
- Type theoretic interpretation of co-routines (Crolard 2004)

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft
  - GBiInt: Buisman/Postniece, Goré 2007

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft
  - GBiInt: Buisman/Postniece, Goré 2007
- Goal: a sequent calculus for BiInt that

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft
  - GBiInt: Buisman/Postniece, Goré 2007
- Goal: a sequent calculus for BiInt that
  - Is complete

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft
  - GBiInt: Buisman/Postniece, Goré 2007
- Goal: a sequent calculus for BiInt that
  - Is complete
  - Allows backward proof search

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft
  - GBiInt: Buisman/Postniece, Goré 2007
- Goal: a sequent calculus for BiInt that
  - Is complete
  - Allows backward proof search
  - Has cut-elimination

Motivation and Related Work

- “Cut-free” sequent calculus for BiInt (Rauszer 1974)
- Rauszer’s cut-elimination fails (Uustalu 2006)
- Display calculus with cut-elimination (Goré 1998)
  - Unsuitable for proof search due to display postulates
- Cut-free proof search calculi with semantic completeness:
  - Labels: Uustalu, Pinto draft
  - GBiInt: Buisman/Postniece, Goré 2007
- Goal: a sequent calculus for BiInt that
  - Is complete
  - Allows backward proof search
  - Has cut-elimination
- Broader goal: proof search in display calculus

Syntax and Kripke Models

- Connectives: $\wedge$ $\vee$ $\rightarrow$ $-\!\!<$

Syntax and Kripke Models

- Connectives: $\wedge \; \vee \; \rightarrow \; -\!\!<$
- Constants: $\top \; \bot$

Syntax and Kripke Models

- Connectives: $\wedge \; \vee \; \rightarrow \; -\!\!<$
- Constants: $\top \; \bot$
- Defined connectives: $\neg \; \sim$

Syntax and Kripke Models

- Connectives: $\wedge \vee \rightarrow \prec$
- Constants: $\top \perp$
- Defined connectives: $\neg \sim$
  - $\neg A := A \rightarrow \perp$ (Int negation)

Syntax and Kripke Models

- Connectives: $\wedge \; \vee \; \rightarrow \; \prec$
- Constants: $\top \; \bot$
- Defined connectives: $\neg \; \sim$
  - $\neg A := A \rightarrow \bot$ (Int negation)
  - $\sim A := \top \prec A$ (dual-Int / paraconsistent negation)

Syntax and Kripke Models

- Connectives: $\wedge \ \vee \ \rightarrow \ \prec$
- Constants: $\top \ \bot$
- Defined connectives: $\neg \ \sim$
  - $\neg A := A \rightarrow \bot$ (Int negation)
  - $\sim A := \top \prec A$ (dual-Int / paraconsistent negation)
- Kripke semantics: model $\langle W, \leq, V \rangle$

Syntax and Kripke Models

- Connectives: $\wedge$ $\vee$ $\rightarrow$ $\prec$
- Constants: $\top$ $\bot$
- Defined connectives: $\neg$ $\sim$
  - $\neg A := A \rightarrow \bot$ (Int negation)
  - $\sim A := \top \prec A$ (dual-Int / paraconsistent negation)
- Kripke semantics: model $\langle W, \leq, V \rangle$
  - $\leq$ is a reflexive and transitive binary relation over W

Syntax and Kripke Models

- Connectives: $\wedge \vee \rightarrow -\!\!<$
- Constants: $\top \perp$
- Defined connectives: $\neg \sim$
  - $\neg A := A \rightarrow \perp$ (Int negation)
  - $\sim A := \top -\!\!< A$ (dual-Int / paraconsistent negation)
- Kripke semantics: model $\langle W, \leq, V \rangle$
  - $\leq$ is a reflexive and transitive binary relation over W
  - V maps atoms to 2^W

Syntax and Kripke Models

- Connectives: $\wedge \; \vee \; \to \; -\!\!<$
- Constants: $\top \; \bot$
- Defined connectives: $\neg \; \sim$
  - $\neg A := A \to \bot$ (Int negation)
  - $\sim A := \top -\!\!< A$ (dual-Int / paraconsistent negation)
- 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 $V(\bot) = \emptyset$

Syntax and Kripke Models

- Connectives: $\wedge \vee \rightarrow -\!\!<$
- Constants: $\top \perp$
- Defined connectives: $\neg \sim$
  - $\neg A := A \rightarrow \perp$ (Int negation)
  - $\sim A := \top -\!\!< A$ (dual-Int / paraconsistent negation)
- 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 $u \in V(p)$

Syntax and Kripke Models

- Connectives: $\wedge \ \vee \ \rightarrow \ \prec$
- Constants: $\top \ \bot$
- Defined connectives: $\neg \ \sim$
  - $\neg A := A \rightarrow \bot$ (Int negation)
  - $\sim A := \top \prec A$ (dual-Int / paraconsistent negation)
- 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 $V(\bot) = \emptyset$
  - V satisfies: if $w \in V(p)$ and $w \leq u$ then $u \in V(p)$
  - $w \Vdash p$ iff $w \in V(p)$

Syntax and Kripke Models

- Connectives: $\wedge \vee \rightarrow -\!\!<$
- Constants: $\top \perp$
- Defined connectives: $\neg \sim$
  - $\neg A := A \rightarrow \perp$ (Int negation)
  - $\sim A := \top -\!\!< A$ (dual-Int / paraconsistent negation)
- 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 $w \in V(p)$
  - Forcing of compound formulae defined w.r.t. $\leq$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ iff $w \Vdash A \quad \& \quad w \Vdash B$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ iff $w \Vdash A$ & $w \Vdash B$
  - $w \Vdash A \vee B$ iff $w \Vdash A$ or $w \Vdash B$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ 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 $\forall u \geq w . u \nVdash A$ or $u \Vdash B$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ iff $w \Vdash A \quad \& \quad w \Vdash B$
  - $w \Vdash A \vee B$ iff $w \Vdash A \;$ or $\; w \Vdash B$
  - $w \Vdash A \rightarrow B$ iff $\forall u \geq w . u \nVdash A$ or $u \Vdash B$
  - $w \Vdash A \prec B$ iff $\exists u \leq w . u \Vdash A \;\; \& \;\; u \nVdash B$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ iff $w \Vdash A \quad \& \quad w \Vdash B$
  - $w \Vdash A \vee B$ iff $w \Vdash A$ or $w \Vdash B$
  - $w \Vdash A \rightarrow B$ iff $\forall u \geq w. u \nVdash A$ or $u \Vdash B$
  - $w \Vdash A \prec B$ iff $\exists u \leq w. u \Vdash A \quad \& \quad u \nVdash B$
- BiInt is a conservative extension of Int

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ 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 $\forall u \geq w.u \nVdash A$ or $u \Vdash B$
  - $w \Vdash A \prec B$ iff $\exists u \leq w.u \Vdash A$ & $u \nVdash B$
- BiInt is a conservative extension of Int
- Persistence: $w \Vdash A \Rightarrow \forall u \geq w.u \Vdash A$

Forcing of Formulae

- Model $\langle W, \leq, V \rangle$
  - $w \Vdash A \wedge B$ 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 $\forall u \geq w . u \nVdash A$ or $u \Vdash B$
  - $w \Vdash A \prec B$ iff $\exists u \leq w . u \Vdash A \ \ \& \ \ u \nVdash B$
- BiInt is a conservative extension of Int
- Persistence: $w \Vdash A \Rightarrow \forall u \geq w . u \Vdash A$
- Reverse Persistence: $w \nVdash A \Rightarrow \forall u \leq w . u \nVdash A$

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$
- A valid iff for all models $\langle W, \leq, V \rangle$ and all worlds $w \in W$, $w \Vdash A$

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$
- 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 $w \in W$, $w \nVdash A$

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$
- 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 $w \in W$, $w \nVdash A$
- $\Gamma \vdash \Delta$ valid iff $\bigwedge \Gamma \to \bigvee \Delta$ valid

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$
- 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 $w \in W$, $w \nVdash A$
- $\Gamma \vdash \Delta$ valid iff $\bigwedge \Gamma \to \bigvee \Delta$ 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 $B \in \Delta$

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$
- 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 $w \in W$, $w \nVdash A$
- $\Gamma \vdash \Delta$ valid iff $\bigwedge \Gamma \to \bigvee \Delta$ 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 $\bigwedge \Gamma \to \bigvee \Delta$ falsifiable

Validity and Falsifiability

- $w \Vdash \Gamma$ iff $w \Vdash A$ for all $A \in \Gamma$
- $w \nVdash \Delta$ iff $w \nVdash A$ for all $A \in \Delta$
- 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 $w \in W$, $w \nVdash A$
- $\Gamma \vdash \Delta$ valid iff $\bigwedge \Gamma \to \bigvee \Delta$ 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 $\bigwedge \Gamma \to \bigvee \Delta$ falsifiable
  - i.e. exists a model $\langle W, \leq, V \rangle$, a world $w \in W$, $w \Vdash \Gamma$ and $w \nVdash \Delta$

Uustalu's Example: Using Cut

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

Uustalu's Example: Using Cut

- Rauszer's $\rightarrow_R$ and $\prec_L$ require singleton succedent/antecedent:
  - $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_R \qquad \frac{A \vdash B, \Delta}{A \prec B \vdash \Delta} \prec_L$$

Uustalu's Example: Using Cut

- Rauszer's $\rightarrow_R$ and $\prec_L$ require singleton succedent/antecedent:
  - $\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \rightarrow_R \qquad \dfrac{A \vdash B, \Delta}{A \prec B \vdash \Delta} \prec_L$
- $p \vdash q, r \rightarrow ((p \prec q) \wedge r)$ is not cut-free derivable in Rauszer's G1

Uustalu's Example: Using Cut

- Rauszer's $\to_R$ and $-\!\!<_L$ require singleton succedent/antecedent:
  - $\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\to_R \qquad \dfrac{A \vdash B, \Delta}{A -\!\!< B \vdash \Delta}-\!\!<_L$
- $p \vdash q, r \to ((p -\!\!< q) \wedge r)$ is not cut-free derivable in Rauszer's G1
- Failed derivation attempt:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\text{Id} \qquad q \vdash ?}{p, r \vdash p -\!\!< q}-\!\!<_R \qquad \dfrac{}{p, r \vdash r}\text{Id}}{p, r \vdash (p -\!\!< q) \wedge r}\wedge_R}{p \vdash r \to ((p -\!\!< q) \wedge r)}\to_R}{p \vdash q, r \to ((p -\!\!< q) \wedge r)}w_R$$

Uustalu's Example: Using Cut

- Rauszer's $\to_R$ and $\mathrel{-\!\!<}_L$ require singleton succedent/antecedent:
  - $\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,\to_R \qquad \dfrac{A \vdash B, \Delta}{A \mathrel{-\!\!<} B \vdash \Delta}\,\mathrel{-\!\!<}_L$
- $p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)$ is not cut-free derivable in Rauszer's G1
- Failed derivation attempt:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash p}\,\text{Id} \qquad q \vdash ?}{p, r \vdash p \mathrel{-\!\!<} q}\,\mathrel{-\!\!<}_R \qquad \dfrac{}{p, r \vdash r}\,\text{Id}}{p, r \vdash (p \mathrel{-\!\!<} q) \wedge r}\,\wedge_R}{p \vdash r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,\to_R}{p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,w_R$$

- Derivation using cut:

$$\dfrac{\dfrac{\dfrac{}{p \vdash q, p}\,\text{Id} \qquad \dfrac{}{q \vdash q}\,\text{Id}}{p \vdash q, p \mathrel{-\!\!<} q}\,\mathrel{-\!\!<}_R \qquad \dfrac{\dfrac{\dfrac{}{p \mathrel{-\!\!<} q, r \vdash p \mathrel{-\!\!<} q}\,\text{Id} \qquad \dfrac{}{p \mathrel{-\!\!<} q, r \vdash r}\,\text{Id}}{p \mathrel{-\!\!<} q, r \vdash (p \mathrel{-\!\!<} q) \wedge r}\,\wedge_R}{p \mathrel{-\!\!<} q \vdash r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,\to_R}{p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,\text{cut}$$

Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p \prec q) \wedge r)$ is valid

Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \prec q) \wedge r)$ is valid
- Counter-model construction

Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p \prec q) \wedge r)$ is valid
- Counter-model construction
  - $w \Vdash p$

Uustalu's Example: Semantic Motivation

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

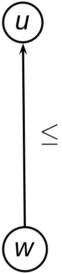
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)$ is valid
- Counter-model construction
  - $w \Vdash p$
  - $w \nVdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)$

$$w \Vdash p$$
$$w \nVdash q$$
$$w \nVdash r \to ((p \mathrel{-\!\!<} q) \wedge r)$$

Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \rightarrow ((p \mathbin{-\!\!<} q) \wedge r)$ is valid
- Counter-model construction
  - $w \Vdash p$
  - $w \nVdash q, r \rightarrow ((p \mathbin{-\!\!<} q) \wedge r)$



$u \Vdash p$

$u \Vdash r$

$u \nVdash (p \mathbin{-\!\!<} q) \wedge r$

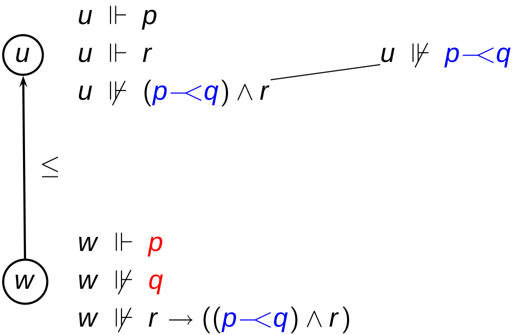
$w \Vdash p$

$w \nVdash q$

$w \nVdash r \rightarrow ((p \mathbin{-\!\!<} q) \wedge r)$

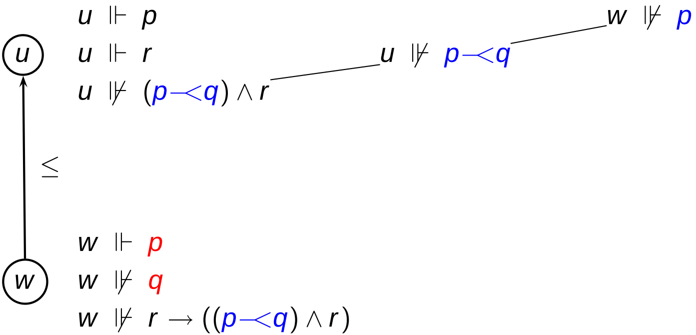
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p \mathrel{-\!\!<} q) \land r)$ is valid
- Counter-model construction
  - $w \Vdash p$
  - $w \not\Vdash q, r \to ((p \mathrel{-\!\!<} q) \land r)$



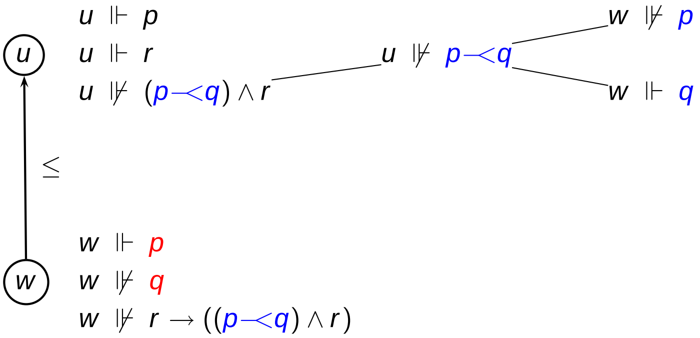
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p -\!\!< q) \wedge r)$ is valid
- Counter-model construction
    - $w \Vdash p$
    - $w \nVdash q, r \to ((p -\!\!< q) \wedge r)$



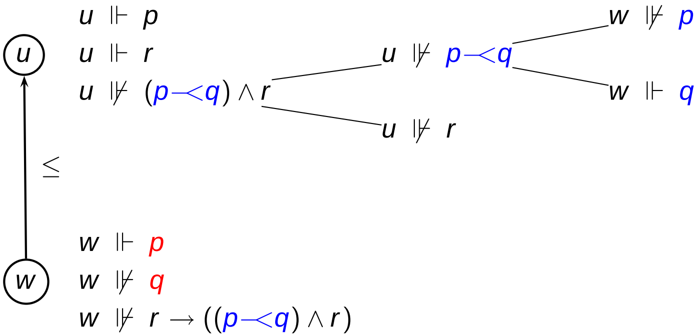
Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)$ is valid
- Counter-model construction
  - $w \Vdash p$
  - $w \nVdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)$



Uustalu's Example: Semantic Motivation

- Sequent $p \vdash q, r \to ((p \mathrel{-\!\!<} q) \land r)$ is valid
- Counter-model construction
  - $w \Vdash p$
  - $w \nVdash q, r \to ((p \mathrel{-\!\!<} q) \land r)$



Nested Sequents

- Negative structures: $N := \emptyset \mid A \mid (N, N) \mid N < P$

Nested Sequents

- Negative structures: $N := \emptyset \mid A \mid (N, N) \mid N < P$
- Positive structures: $P := \emptyset \mid A \mid (P, P) \mid N > P$

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

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

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

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
- Similar, but more restricted than display logic

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
- Similar, but more restricted than display logic
- Examples:

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
- Similar, but more restricted than display logic
- Examples:
  - $r \vdash p -\!\!< q$

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
- Similar, but more restricted than display logic
- Examples:
  - $r \vdash p {-\!\!<} q$
  - $(p < q), r \vdash (q > r), ((p < q) > w)$

Identity, cut and structural rules

Identity and cut:

$$\frac{}{X, A \vdash A, Y}\ id \qquad \frac{X_1 \vdash Y_1, A \quad A, X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

Identity, cut and structural rules

Identity and cut:

$$\frac{}{X, A \vdash A, Y}\ id \qquad \frac{X_1 \vdash Y_1, A \quad A, X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

Structural rules:

$$\frac{X \vdash Y}{X, A \vdash Y}\ w_L \qquad \frac{X \vdash Y}{X \vdash A, Y}\ w_R \qquad \frac{X, A, A \vdash Y}{X, A \vdash Y}\ c_L \qquad \frac{X \vdash A, A, Y}{X \vdash A, Y}\ c_R$$

Identity, cut and structural rules

Identity and cut:

$$\frac{}{X, A \vdash A, Y}\ id \qquad \frac{X_1 \vdash Y_1, A \quad A, X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

Structural rules:

$$\frac{X \vdash Y}{X, A \vdash Y}\ w_L \qquad \frac{X \vdash Y}{X \vdash A, Y}\ w_R \qquad \frac{X, A, A \vdash Y}{X, A \vdash Y}\ c_L \qquad \frac{X \vdash A, A, Y}{X \vdash A, Y}\ c_R$$

Identity, cut and structural rules

Identity and cut:

$$\frac{}{X, A \vdash A, Y}\ id \qquad \frac{X_1 \vdash Y_1, A \quad A, X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ cut$$

Structural rules:

$$\frac{X \vdash Y}{X, A \vdash Y}\ w_L \qquad \frac{X \vdash Y}{X \vdash A, Y}\ w_R \qquad \frac{X, A, A \vdash Y}{X, A \vdash Y}\ c_L \qquad \frac{X \vdash A, A, Y}{X \vdash A, Y}\ c_R$$

$$\frac{(X_1 < Y_1), X_2 \vdash Y_2}{X_1, X_2 \vdash Y_1, Y_2}\ s_L \qquad \frac{X_1 \vdash Y_1, (X_2 > Y_2)}{X_1, X_2 \vdash Y_1, Y_2}\ s_R$$

$$\frac{X_2 \vdash Y_2, Y_1}{X_1, (X_2 < Y_2) \vdash Y_1}\ < \qquad \frac{X_1, X_2 \vdash Y_2}{X_1 \vdash Y_1, (X_2 > Y_2)}\ >$$

Logical rules

Logical rules

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

Logical rules

Logical rules

$$\frac{X \vdash A, Y \quad X, B \vdash Y}{X, A \to B \vdash Y} \to_L \qquad \frac{X, A \vdash B}{X \vdash Y, A \to B} \to_R$$

Logical rules

Logical rules

$$\frac{X \vdash A, Y \quad X, B \vdash Y}{X, A \to B \vdash Y} \to_L \qquad \frac{X, A \vdash B}{X \vdash Y, A \to B} \to_R$$

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

Uustalu's Example Revisited

Using cut:

$$\dfrac{\dfrac{\dfrac{}{p \vdash q, p}\;Id \qquad \dfrac{}{q \vdash q}\;Id}{p \vdash q, p \mathrel{-\!\!<} q}\;{\mathrel{-\!\!<}}_R \qquad \dfrac{\dfrac{\dfrac{}{p \mathrel{-\!\!<} q, r \vdash p \mathrel{-\!\!<} q}\;Id \qquad \dfrac{}{p \mathrel{-\!\!<} q, r \vdash r}\;Id}{p \mathrel{-\!\!<} q, r \vdash (p \mathrel{-\!\!<} q) \wedge r}\;\wedge_R}{p \mathrel{-\!\!<} q \vdash r \to ((p \mathrel{-\!\!<} q) \wedge r)}\;\to_R}{p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)}\;\text{cut}$$

Using LBiInt_1 without cut:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p \vdash q, p}\;Id \qquad \dfrac{}{p, q \vdash q}\;Id}{p \vdash q, p \mathrel{-\!\!<} q}\;{\mathrel{-\!\!<}}_R}{(p < q), r \vdash p \mathrel{-\!\!<} q}\;< \qquad \dfrac{}{(p < q), r \vdash r}\;Id}{(p < q), r \vdash (p \mathrel{-\!\!<} q) \wedge r}\;\wedge_R}{p < q \vdash r \to ((p \mathrel{-\!\!<} q) \wedge r)}\;\to_R}{p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)}\;s_L$$

Soundness and Completeness

- Formula-translation τ maps sequents to formulae

Soundness and Completeness

- Formula-translation τ maps sequents to formulae
  - $>$ maps to $\rightarrow$

Soundness and Completeness

- Formula-translation τ maps sequents to formulae
  - $>$ maps to $\rightarrow$
  - $<$ maps to $\prec$

Soundness and Completeness

- Formula-translation τ maps sequents to formulae
  - $>$ maps to $\rightarrow$
  - $<$ maps to $\prec$
- Soundness: for every rule, show:

Soundness and Completeness

- Formula-translation τ maps sequents to formulae
  - $>$ maps to $\to$
  - $<$ maps to $-\!\!<$
- Soundness: for every rule, show:
  - if τ of premises is valid then τ of conclusion is valid

Soundness and Completeness

- Formula-translation τ maps sequents to formulae
  - $>$ maps to $\rightarrow$
  - $<$ maps to $-\!\!<$
- Soundness: for every rule, show:
  - if τ of premises is valid then τ of conclusion is valid
- Completeness: by embedding Rauszer's G1 into LBiInt_1

General Contraction and Weakening

Lemma

Contraction and weakening on structures admissible:

$$\frac{X , Y , Y \vdash Z}{X , Y \vdash Z}\ gc_L \qquad \frac{X \vdash Y , Y , Z}{X \vdash Y , Z}\ gc_R$$

$$\frac{X \vdash Z}{X , Y \vdash Z}\ gw_L \qquad \frac{X \vdash Z}{X \vdash Y , Z}\ gw_R$$

General Contraction and Weakening

Lemma

Contraction and weakening on structures admissible:

$$\frac{X , Y , Y \vdash Z}{X , Y \vdash Z}\ gc_L \qquad \frac{X \vdash Y , Y , Z}{X \vdash Y , Z}\ gc_R$$

$$\frac{X \vdash Z}{X , Y \vdash Z}\ gw_L \qquad \frac{X \vdash Z}{X \vdash Y , Z}\ gw_R$$

Proof.

By induction on the size of Y. □

Atomic Cuts

We transform

$$\dfrac{\dfrac{\dfrac{}{p \vdash p}\text{id} \quad \cdots \quad \dfrac{}{p \vdash p}\text{id}}{X_1 \vdash Y_1, p}\,\vdots\,\theta \qquad \begin{matrix}\pi \\ p, X_2 \vdash Y_2\end{matrix}}{X_1, X_2 \vdash Y_1, Y_2}\text{cut}$$

Atomic Cuts

We transform

$$\dfrac{\dfrac{\dfrac{}{p \vdash p}\text{id} \quad \cdots \quad \dfrac{}{p \vdash p}\text{id}}{X_1 \vdash Y_1, p}\,\vdots\,\theta \qquad \overset{\pi}{p, X_2 \vdash Y_2}}{X_1, X_2 \vdash Y_1, Y_2}\text{cut}$$

into:

$$\dfrac{\dfrac{\dfrac{\overset{\pi}{p, X_2 \vdash Y_2}}{p \vdash X_2 > Y_2} > \quad \cdots \quad \dfrac{\overset{\pi}{p, X_2 \vdash Y_2}}{p \vdash X_2 > Y_2} >}{X_1 \vdash Y_1, (X_2 > Y_2)}\,\vdots\,\theta[p/X_2 > Y_2]}{X_1, X_2 \vdash Y_1, Y_2}s_R$$

General Cuts: $A \to B$

We transform

$$
\dfrac{
\begin{array}{c}
\pi_1 \\
\dfrac{X_1', A \vdash B}{X_1' \vdash Y_1', A \to B}\,{\to_R} \\
\vdots\;\theta_1 \\
X_1 \vdash Y_1, A \to B
\end{array}
\qquad
\begin{array}{c}
\dfrac{\begin{array}{cc}\pi_2 & \pi_3 \\ X_2' \vdash A, Y_2' & B, X_2' \vdash Y_2'\end{array}}{A \to B, X_2' \vdash Y_2'}\,{\to_L} \\
\vdots\;\theta_2 \\
A \to B, X_2 \vdash Y_2
\end{array}
}{X_1, X_2 \vdash Y_1, Y_2}\;\text{cut}
$$

General Cuts: $A \rightarrow B$

We transform

$$
\dfrac{
\begin{array}{c}
\dfrac{\begin{array}{c}\pi_1 \\ X_1', A \vdash B\end{array}}{X_1' \vdash Y_1', A \rightarrow B} \rightarrow_R \\
\vdots\ \theta_1 \\
X_1 \vdash Y_1, A \rightarrow B
\end{array}
\qquad
\begin{array}{c}
\dfrac{\begin{array}{c}\pi_2 \\ X_2' \vdash A, Y_2'\end{array} \qquad \begin{array}{c}\pi_3 \\ B, X_2' \vdash Y_2'\end{array}}{A \rightarrow B, X_2' \vdash Y_2'} \rightarrow_L \\
\vdots\ \theta_2 \\
A \rightarrow B, X_2 \vdash Y_2
\end{array}
}{X_1, X_2 \vdash Y_1, Y_2} \text{ cut}
$$

into:

$$
\dfrac{
\begin{array}{c}
\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_2 \\ X_2' \vdash A, Y_2'\end{array} \qquad \dfrac{\begin{array}{c}\pi_1 \\ X_1', A \vdash B\end{array} \qquad \begin{array}{c}\pi_3 \\ B, X_2' \vdash Y_2'\end{array}}{X_1', A, X_2' \vdash Y_2'} \text{ cut}}{X_1', X_2', X_2' \vdash Y_2', Y_2'} \text{ cut}}{X_1', X_2' \vdash Y_2'} \ gc_L, gc_R \\
\vdots\ \theta_2[A \rightarrow B / X_1'] \\
\dfrac{X_1', X_2 \vdash Y_2}{X_1' \vdash Y_1', (X_2 > Y_2)} > \\
\vdots\ \theta_1[A \rightarrow B / X_2 > Y_2] \\
X_1 \vdash Y_1, (X_2 > Y_2)
\end{array}
}{X_1, X_2 \vdash Y_1, Y_2} \ s_R
$$

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$
  - Also possible via detour through display calculus

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$
  - Also possible via detour through display calculus
- But LBiInt$_1$ is not suitable for proof search:

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$
  - Also possible via detour through display calculus
- But LBiInt$_1$ is not suitable for proof search:
  - Structural rules allow shuffling of structures ad infinitum

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$
  - Also possible via detour through display calculus
- But LBiInt$_1$ is not suitable for proof search:
  - Structural rules allow shuffling of structures ad infinitum
  - Unlimited contraction

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$
  - Also possible via detour through display calculus
- But LBiInt$_1$ is not suitable for proof search:
  - Structural rules allow shuffling of structures ad infinitum
  - Unlimited contraction
- Solution: absorb structural rules into logical rules

From LBiInt$_1$ to LBiInt$_2$

- LBiInt$_1$ has an elegant direct cut-elimination proof
  - Using structural rules s_L, s_R, $>$ and $<$
  - Also possible via detour through display calculus
- But LBiInt$_1$ is not suitable for proof search:
  - Structural rules allow shuffling of structures ad infinitum
  - Unlimited contraction
- Solution: absorb structural rules into logical rules

$$\cfrac{\cfrac{\cfrac{(X < Y, A \to B), X, A \vdash B}{(X < Y, A \to B), X \vdash Y, A \to B}\ \to_R}{X, X \vdash Y, Y, A \to B, A \to B}\ s_L}{X \vdash Y, A \to B}\ gc_L, gc_R \quad \rightsquigarrow \quad \cfrac{(X < Y, A \to B), \{X\}, A \vdash B}{X \vdash Y, A \to B}\ \to_{R2}$$

$\{X\} = \{A \mid X = (A, Y)$ for some A and $Y\}$

LBiInt$_2$ Rules

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

LBiInt$_2$ Rules

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

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

LBiInt$_2$ Rules

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

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

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

LBiInt$_2$ Rules

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

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

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

$$\frac{X, A \to B \vdash A, Y \quad X, A \to B, B \vdash Y}{X, A \to B \vdash Y} \to_L \qquad \frac{X \vdash Y, A \to B, B}{X \vdash Y, A \to B} \to_{R1}$$

LBiInt$_2$ Rules

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

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

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

$$\frac{X, A \to B \vdash A, Y \quad X, A \to B, B \vdash Y}{X, A \to B \vdash Y}\ \to_L \qquad \frac{X \vdash Y, A \to B, B}{X \vdash Y, A \to B}\ \to_{R1}$$

$$\frac{X, A \prec B, A \vdash Y}{X, A \prec B \vdash Y}\ \prec_{L1} \qquad \frac{X \vdash A, A \prec B, Y \quad X, B \vdash A \prec B, Y}{X \vdash A \prec B, Y}\ \prec_R$$

LBiInt$_2$ Rules

$$\{X\} = \{A \mid X = (A, Y) \text{ for some } A \text{ and } Y\}$$

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

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

$$\frac{X, A \to B \vdash A, Y \quad X, A \to B, B \vdash Y}{X, A \to B \vdash Y} \to_L \qquad \frac{X \vdash Y, A \to B, B}{X \vdash Y, A \to B} \to_{R1}$$

$$\frac{X, A \prec B, A \vdash Y}{X, A \prec B \vdash Y} \prec_{L1} \qquad \frac{X \vdash A, A \prec B, Y \quad X, B \vdash A \prec B, Y}{X \vdash A \prec B, Y} \prec_R$$

$$\frac{A \vdash B, \{Y\}, (X, A \prec B > Y)}{X, A \prec B \vdash Y} \prec_{L2} \qquad \frac{(X < Y, A \to B), \{X\}, A \vdash B}{X \vdash Y, A \to B} \to_{R2}$$

Uustalu's Example Revisited

Using LBiInt$_1$:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{}{p \vdash q, p}\,Id
          \qquad
          \dfrac{}{p, q \vdash q}\,Id
        }{p \vdash q, p \mathrel{-\!\!<} q}\,{-\!\!<}_R
      }{(p < q), r \vdash p \mathrel{-\!\!<} q}\,<
      \qquad
      \dfrac{}{(p < q), r \vdash r}\,Id
    }{(p < q), r \vdash (p \mathrel{-\!\!<} q) \wedge r}\,\wedge_R
  }{p < q \vdash r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,\to_R
}{p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,s_L
$$

Using LBiInt$_2$:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{p \vdash q, \cdots, p}\,Id
        \qquad
        \dfrac{}{p, q \vdash q, \cdots}\,Id
      }{p \vdash q, \cdots, p \mathrel{-\!\!<} q}\,{-\!\!<}_R
    }{(p < q, \cdots), p, r \vdash p \mathrel{-\!\!<} q}\,<
    \qquad
    \dfrac{}{(p < q, \cdots), p, r \vdash r}\,Id
  }{(p < q, \cdots), p, r \vdash (p \mathrel{-\!\!<} q) \wedge r}\,\wedge_R
}{p \vdash q, r \to ((p \mathrel{-\!\!<} q) \wedge r)}\,\to_{R2}
$$

Save/Restore

- LBiInt$_1$ vs LBiInt$_2$:

Lose context:

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

Save context:

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

Save/Restore

- LBiInt$_1$ vs LBiInt$_2$:

Lose context:

$$\frac{X, A \vdash B}{X \vdash Y, A \to B} \to_R$$

Save context:

$$\frac{(X < Y, A \to B), \{X\}, A \vdash B}{X \vdash Y, A \to B} \to_{R2}$$

- Restore context:

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

Save/Restore

- LBiInt$_1$ vs LBiInt$_2$:

Lose context:

$$\frac{X, A \vdash B}{X \vdash Y, A \to B} \to_R$$

Save context:

$$\frac{(X < Y, A \to B), \{X\}, A \vdash B}{X \vdash Y, A \to B} \to_{R2}$$

- Restore context:

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

- LBiInt$_2$ completeness via translation from GBiInt

Save/Restore

- LBiInt$_1$ vs LBiInt$_2$:

Lose context:

$$\frac{X, A \vdash B}{X \vdash Y, A \to B} \to_R$$

Save context:

$$\frac{(X < Y, A \to B), \{X\}, A \vdash B}{X \vdash Y, A \to B} \to_{R2}$$

- Restore context:

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

- LBiInt$_2$ completeness via translation from GBiInt
- GBiInt recompute rule $\rightsquigarrow$ pair of LBiInt$_2$ save/restore rules

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$
5. If $A \vee B \in \{Y\}$ then $A \in \{Y\}$ and $B \in \{Y\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$
5. If $A \vee B \in \{Y\}$ then $A \in \{Y\}$ and $B \in \{Y\}$
6. If $A \rightarrow B \in \{X\}$ then $A \in \{Y\}$ or $B \in \{X\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$
5. If $A \vee B \in \{Y\}$ then $A \in \{Y\}$ and $B \in \{Y\}$
6. If $A \rightarrow B \in \{X\}$ then $A \in \{Y\}$ or $B \in \{X\}$
7. If $A \prec B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{X\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$
5. If $A \vee B \in \{Y\}$ then $A \in \{Y\}$ and $B \in \{Y\}$
6. If $A \to B \in \{X\}$ then $A \in \{Y\}$ or $B \in \{X\}$
7. If $A \prec B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{X\}$
8. If $A \to B \in \{Y\}$ then $B \in \{Y\}$

Saturation

Definition

A sequent $X \vdash Y$ is saturated iff it satisfies:

1. $\{X\} \cap \{Y\} = \emptyset$
2. If $A \wedge B \in \{X\}$ then $A \in \{X\}$ and $B \in \{X\}$
3. If $A \wedge B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{Y\}$
4. If $A \vee B \in \{X\}$ then $A \in \{X\}$ or $B \in \{X\}$
5. If $A \vee B \in \{Y\}$ then $A \in \{Y\}$ and $B \in \{Y\}$
6. If $A \rightarrow B \in \{X\}$ then $A \in \{Y\}$ or $B \in \{X\}$
7. If $A {-\!\!<} B \in \{Y\}$ then $A \in \{Y\}$ or $B \in \{X\}$
8. If $A \rightarrow B \in \{Y\}$ then $B \in \{Y\}$
9. If $A {-\!\!<} B \in \{X\}$ then $A \in \{X\}$

Blocking

Definition

We classify the rules of LBiInt_2 into three groups:

Blocking

Definition

We classify the rules of LBiInt$_2$ into three groups:

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

Blocking

Definition

We classify the rules of LBiInt_2 into three groups:

Static Rules: $= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \to_L, \prec_R, \prec_{L1}, \to_{R1}\}$;

Jump Rules: $= \{\prec_{L2}, \to_{R2}\}$; and

Blocking

Definition

We classify the rules of LBiInt$_2$ into three groups:

$$\text{Static Rules:} = \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \rightarrow_L, -\!\!<_R, -\!\!<_{L1}, \rightarrow_{R1}\};$$

$$\text{Jump Rules:} = \{-\!\!<_{L2}, \rightarrow_{R2}\}; \text{ and}$$

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

Blocking

Definition

We classify the rules of LBiInt_2 into three groups:

$$\begin{aligned} \text{Static Rules:} &= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \to_L, -\!\!<_R, -\!\!<_{L1}, \to_{R1}\}; \\ \text{Jump Rules:} &= \{-\!\!<_{L2}, \to_{R2}\}; \text{ and} \\ \text{Return Rules:} &= \{<, >\}. \end{aligned}$$

Blocking

Definition

We classify the rules of LBiInt$_2$ into three groups:

$$\begin{aligned}\text{Static Rules:} &= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \to_L, -\!\!<_R, -\!\!<_{L1}, \to_{R1}\};\\ \text{Jump Rules:} &= \{-\!\!<_{L2}, \to_{R2}\}; \text{ and}\\ \text{Return Rules:} &= \{<, >\}.\end{aligned}$$

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

Blocking

Definition

We classify the rules of LBiInt$_2$ into three groups:

Static Rules: $= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \to_L, \prec_R, \prec_{L1}, \to_{R1}\}$;

Jump Rules: $= \{\prec_{L2}, \to_{R2}\}$; and

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

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

Definition

A LBiInt$_2$ rule ρ is applicable to a sequent $\gamma_0 = X_0 \vdash Y_0$ if for every premise $X_i \vdash Y_i$ of ρ, $\{X_i\} \not\subseteq \{X_0\}$ or $\{Y_i\} \not\subseteq \{Y_0\}$.

Blocking

Definition

We classify the rules of LBiInt$_2$ into three groups:

$$\begin{aligned}\text{Static Rules: } &= \{id, \wedge_L, \wedge_R, \vee_L, \vee_R, \to_L, -\!\!<_R, -\!\!<_{L1}, \to_{R1}\};\\ \text{Jump Rules: } &= \{-\!\!<_{L2}, \to_{R2}\};\text{ and}\\ \text{Return Rules: } &= \{<, >\}.\end{aligned}$$

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

Definition

A LBiInt$_2$ rule ρ is applicable to a sequent $\gamma_0 = X_0 \vdash Y_0$ 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*

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

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 $\gamma_1, \cdots, \gamma_n$ be the premises of ρ obtained from γ_0

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 $\gamma_1, \cdots, \gamma_n$ be the premises of ρ obtained from γ_0
    2. Return $\bigwedge_{i=1}^{n} \mathit{Prove}(\gamma_i)$

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 $\gamma_1, \cdots, \gamma_n$ be the premises of ρ obtained from γ_0
   2. Return $\bigwedge_{i=1}^{n} Prove(\gamma_i)$
3. Else if $Prove(\gamma_1) = true$ for some premise instance γ_1 obtained from γ_0 by applying $\rho \in \{-\!\!<_{L2}, \rightarrow_{R2}, <, >\}$ then return *true*

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 $\gamma_1, \cdots, \gamma_n$ be the premises of ρ obtained from γ_0
   2. Return $\bigwedge_{i=1}^{n} Prove(\gamma_i)$
3. Else if $Prove(\gamma_1) = true$ for some premise instance γ_1 obtained from γ_0 by applying $\rho \in \{-\!\!<_{L2}, \rightarrow_{R2}, <, >\}$ then return *true*
4. Else return *false*.

Linear Sequents

Definition

Linear Sequents

Definition

1. If Γ/Δ are sets of formulae, then $\Gamma \vdash \Delta$ is a linear sequent.

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

   are linear sequents.

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

    are linear sequents.

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.

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

$C \vdash B, A \rightarrow B$

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

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

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

$C \vdash B, A \to B$

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

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

Lemma

Every LBiInt$_2$-derivation of a linear end-sequent contains only linear sequents.

Lists

Definition (Linear Sequent to List)

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

Lists

Definition (Linear Sequent to List)

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

Example

$$list(C \vdash B, A \rightarrow B) = \langle \{C\}, \{B, A \rightarrow B\} \rangle$$

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

Lists

Definition (Linear Sequent to List)

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

Example

$list(C \vdash B, A \rightarrow B) = \langle \{C\}, \{B, A \rightarrow B\} \rangle$

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

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 $list(X \vdash Y)$:

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*

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*

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*

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.*

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

$$\frac{(C < B, A \to B), C, A \vdash B}{C \vdash B, A \to B} \to_{R2} \qquad \frac{\langle \{C\}, \{B, A \to B\} \rangle \;\; \leq \;\; \langle \{C, A\}, \{B\} \rangle}{\langle \{C\}, \{B, A \to B\} \rangle}$$

Termination

Lemma (Bounded Lists)

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

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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.

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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.

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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.

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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

Termination

Lemma (Bounded Lists)

Let $X \vdash Y$ be any sequent encountered during proof search. Using jump rules, 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

Countermodel Construction

- Want to show: if A valid then $\emptyset \vdash A$ derivable
- Usually show contrapositive:

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

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 $Prove(\emptyset \vdash A) = false$, then $\emptyset \vdash A$ has a counter-model

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 $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$, then $\emptyset \vdash A$ has a counter-model
- More generally:

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 $Prove(\emptyset \vdash A) = false$, then $\emptyset \vdash A$ has a counter-model
- More generally:
  - if $Prove(\Gamma \vdash \Delta) = false$, then $w \Vdash \Gamma$ and $w \not\Vdash \Delta$ for some $\langle W, \leq, V \rangle$ and $w \in W$

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 $Prove(\emptyset \vdash A) = false$, then $\emptyset \vdash A$ has a counter-model
- More generally:
  - if $Prove(\Gamma \vdash \Delta) = false$, then $w \Vdash \Gamma$ and $w \nVdash \Delta$ for some $\langle W, \leq, V \rangle$ and $w \in W$
- Extract countermodel from failed proof search

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 $Prove(\emptyset \vdash A) = false$, then $\emptyset \vdash A$ has a counter-model
- More generally:
  - if $Prove(\Gamma \vdash \Delta) = false$, then $w \Vdash \Gamma$ and $w \nVdash \Delta$ for some $\langle W, \leq, V \rangle$ and $w \in W$
- Extract countermodel from failed proof search
- Usually done inductively by combining sub-models

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 $Prove(\emptyset \vdash A) = false$, then $\emptyset \vdash A$ has a counter-model
- More generally:
  - if $Prove(\Gamma \vdash \Delta) = false$, 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

- Tree of nodes

Pre-model

- Tree of nodes
  - Nodes are pairs $\langle \Gamma, \Delta \rangle$ of sets of formulae

Pre-model

- Tree of nodes
  - Nodes are pairs $\langle \Gamma, \Delta \rangle$ of sets of formulae
  - Nodes are linked by labels $\leq$ or $\geq$

Pre-model

- 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

Pre-model

- 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

Pre-model

- 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

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \rightarrow B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \rightarrow B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A “model in progress”

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A “model in progress”
- When all nodes are marked C, the root has a counter-model

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A "model in progress"
- When all nodes are marked C, the root has a counter-model
  - By induction on the size of the pre-model

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A "model in progress"
- 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

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A "model in progress"
- 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

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A "model in progress"
- 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

Pre-model Properties

- Compatibility of two $\leq$-related C-nodes $\langle \Gamma_1, \Delta_1 \rangle$ and $\langle \Gamma_2, \Delta_2 \rangle$:
  - If $\langle \Gamma_1, \Delta_1 \rangle \leq \langle \Gamma_2, \Delta_2 \rangle$ or $\langle \Gamma_2, \Delta_2 \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$ then
    - $\Gamma_1 \subseteq \Gamma_2$
    - $\Delta_2 \subseteq \Delta_1$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \to B \in \Delta$:
  - $\langle \Gamma, \Delta \rangle \leq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- For every C-node $\langle \Gamma, \Delta \rangle$ and every $A \prec B \in \Gamma$:
  - $\langle \Gamma, \Delta \rangle \geq \langle \Gamma_1, \Delta_1 \rangle$, $A \in \Gamma_1$ and $B \in \Delta_1$ for some C-node $\langle \Gamma_1, \Delta_1 \rangle$
- A “model in progress”
- 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
  - Then we have a proper counter-model

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- Use failed proof search to update pre-model

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- Use failed proof search to update pre-model
  - Static rules: saturate leaf nodes

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- Use failed proof search to update pre-model
  - Static rules: saturate leaf nodes
  - Jump rules: mark node C, create new I-nodes

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- 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

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- 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

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- 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

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- 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

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
  - $w \not\Vdash \{A\}$ for some $\langle W, \leq, V \rangle$ and $w \in W$

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- 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
  - $w \not\Vdash \{A\}$ for some $\langle W, \leq, V \rangle$ and $w \in W$
  - A is not valid

Completeness Proof

- Create pre-model with root I-node $\langle \emptyset, \{A\} \rangle$
- Assume $\mathit{Prove}(\emptyset \vdash A) = \mathit{false}$
- 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
  - $w \not\Vdash \{A\}$ for some $\langle W, \leq, V \rangle$ and $w \in W$
  - A is not valid
- By contrapositive, if A is valid, then $\mathit{Prove}(\emptyset \vdash A) = \mathit{true}$

Restart Example

$$
\dfrac{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0 \qquad \vdots}{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot} \to_L
$$

$$
\dfrac{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot}{p_0 \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0} \to_{R2}
$$

$$
\dfrac{\vdots \qquad p_0 \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0}{\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0} \prec_R
$$

$$
\dfrac{\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0}{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0} <
$$

$$
\dfrac{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0 \qquad \vdots}{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot} \to_L
$$

$$
\dfrac{(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot}{\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot} \to_{R2}
$$

$$
\dfrac{\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot}{\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot} \to_{R1}
$$

(I) $\langle \{\}, \{p_1, ((\top \prec p_0) \to \bot) \to \bot\} \rangle$

Restart Example

$$
\dfrac{\dfrac{\dfrac{\vdots \qquad \dfrac{\dfrac{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0 \qquad \vdots}{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot}\rightarrow_L}{p_0 \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0}\rightarrow_{R2}}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0}-\!\!<_R}{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0}< \qquad \vdots}{\dfrac{\dfrac{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot}\rightarrow_{R2}}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot}\rightarrow_{R1}}\rightarrow_L
$$

(I) $\langle \{\}, \{p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot\} \rangle$

Restart Example

$$(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0 \qquad \vdots$$
$$\to_L$$
$$(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot$$
$$\to_{R2}$$
$$\vdots \qquad p_0 \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0$$
$$\prec_R$$
$$\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0$$
$$<$$
$$(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0 \qquad \vdots$$
$$\to_L$$
$$(< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot$$
$$\to_{R2}$$
$$\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot$$
$$\to_{R1}$$
$$\vdash p_1, ((\top \prec p_0) \to \bot) \to \bot$$

$I \quad \langle \{(\top \prec p_0) \to \bot\}, \{\bot\} \rangle$

$\uparrow \; \leq$

$C \quad \langle \{\}, \{p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot\} \rangle$

Restart Example

$$
\dfrac{
  \vdots
  \qquad
  \dfrac{
    \dfrac{
      (< p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot, \top{-\!\!<}p_0), p_0, (\top{-\!\!<}p_0) \to \bot \vdash \bot, \top{-\!\!<}p_0 \qquad \vdots
    }{
      (< p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot, \top{-\!\!<}p_0), p_0, (\top{-\!\!<}p_0) \to \bot \vdash \bot
    }\ {\to_L}
  }{
    p_0 \vdash p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot, \top{-\!\!<}p_0
  }\ {\to_{R2}}
}{
  \vdash p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot, \top{-\!\!<}p_0
}\ {-\!\!<_R}
$$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \vdash p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot, \top{-\!\!<}p_0
      }{
        \color{blue}{(< p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot), (\top{-\!\!<}p_0) \to \bot \vdash \bot, \top{-\!\!<}p_0}
      }\ {<} \qquad \vdots
    }{
      (< p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot), (\top{-\!\!<}p_0) \to \bot \vdash \bot
    }\ {\to_L}
  }{
    \vdash p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot
  }\ {\to_{R2}}
}{
  \vdash p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot
}\ {\to_{R1}}
$$

I $\quad \langle \{(\top{-\!\!<}p_0) \to \bot\}, \{\bot, \top{-\!\!<}p_0\} \rangle$

$\uparrow \ \leq$

C $\quad \langle \{\}, \{p_1, ((\top{-\!\!<}p_0) \to \bot) \to \bot, \bot\} \rangle$

Restart Example

$$
\dfrac{
  \dfrac{
    \dfrac{
      \vdots
      \qquad
      \dfrac{
        \dfrac{
          \dfrac{
            (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0
            \qquad \vdots
          }{
            (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot
          }\ \to_L
        }{
          p_0 \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0
        }\ \to_{R2}
      }{}
    }{
      \color{blue}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0}
    }\ -\!\!<_R
  }{
    (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0
  }\ <
  \qquad \vdots
}{
  \dfrac{
    \dfrac{
      (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot
    }{
      \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot
    }\ \to_{R2}
  }{
    \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot
  }\ \to_{R1}
}\ \to_L
$$

(I) $\langle \{\}, \{p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0\} \rangle$

Restart Example

$$\dfrac{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0), p_0, (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot, \top \mathrel{-\!\!<} p_0 \qquad \vdots}{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0), p_0, (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot} \to_L$$

$$\dfrac{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0), p_0, (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot}{p_0 \vdash p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0} \to_{R2}$$

$$\dfrac{\vdots \qquad p_0 \vdash p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0}{\vdash p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0} \mathrel{-\!\!<}_R$$

$$\dfrac{\vdash p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0}{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot), (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot, \top \mathrel{-\!\!<} p_0} <$$

$$\dfrac{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot), (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot, \top \mathrel{-\!\!<} p_0 \qquad \vdots}{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot), (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot} \to_L$$

$$\dfrac{\dfrac{(< p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot), (\top \mathrel{-\!\!<} p_0) \to \bot \vdash \bot}{\vdash p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot} \to_{R2}}{\vdash p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot} \to_{R1}$$

(I) $\langle \{p_0\}, \{p_1, ((\top \mathrel{-\!\!<} p_0) \to \bot) \to \bot, \bot, \top \mathrel{-\!\!<} p_0\} \rangle$

Restart Example

$$\dfrac{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0 \qquad \vdots}{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot} \to_L$$

$$\dfrac{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot}{p_0 \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0} \to_{R2}$$

$$\dfrac{\vdots \qquad p_0 \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0} -\!\!<_R$$

$$\dfrac{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0}{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0} <$$

$$\dfrac{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0 \qquad \vdots}{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot} \to_L$$

$$\dfrac{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot} \to_{R2}$$

$$\dfrac{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot}{\vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot} \to_{R1}$$

$$I \quad \langle \{p_0, (\top -\!\!< p_0) \to \bot\}, \{\bot\} \rangle$$

$$C \leq I$$

$$C \quad \langle \{p_0\}, \{p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0\} \rangle$$

Restart Example

$$
\dfrac{
  \dfrac{
    \vdots
    \quad
    \dfrac{
      \dfrac{
        \dfrac{
          \color{blue}{(< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0}
          \quad \vdots
        }{
          (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0), p_0, (\top -\!\!< p_0) \to \bot \vdash \bot
        }\ \to_L
      }{
        p_0 \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0
      }\ \to_{R2}
    }{}
  }{
    \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0
  }\ -\!\!<_R
}{}
$$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0
      }{
        (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot, \top -\!\!< p_0
      }\ <
      \quad \vdots
    }{
      (< p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot), (\top -\!\!< p_0) \to \bot \vdash \bot
    }\ \to_L
  }{
    \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot
  }\ \to_{R2}
}{
  \vdash p_1, ((\top -\!\!< p_0) \to \bot) \to \bot
}\ \to_{R1}
$$

I : $\langle \{p_0, (\top -\!\!< p_0) \to \bot\}, \{\bot, \top -\!\!< p_0\} \rangle$

$\uparrow \leq$

C : $\langle \{p_0\}, \{p_1, ((\top -\!\!< p_0) \to \bot) \to \bot, \bot, \top -\!\!< p_0\} \rangle$

Restart Example

$$
\dfrac{
  \dfrac{
    \dfrac{
      \vdots
      \qquad
      \dfrac{
        \dfrac{
          (< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0
          \qquad \vdots
        }{
          (< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0), p_0, (\top \prec p_0) \to \bot \vdash \bot
        }\to_L
      }{
        p_0 \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0
      }\to_{R2}
    }{
      \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0
    }\prec_R
  }{
    (< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot, \top \prec p_0
  }<
  \qquad \vdots
}{
  \dfrac{
    \dfrac{
      (< p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot), (\top \prec p_0) \to \bot \vdash \bot
    }{
      \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot
    }\to_{R2}
  }{
    \vdash p_1, ((\top \prec p_0) \to \bot) \to \bot
  }\to_{R1}
}\to_L
$$

C $\quad \langle \{p_0, (\top \prec p_0) \to \bot\}, \{\bot, \top \prec p_0\} \rangle$

$\uparrow \; \leq$

C $\quad \langle \{p_0\}, \{p_1, ((\top \prec p_0) \to \bot) \to \bot, \bot, \top \prec p_0\} \rangle$

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
  - a sequent calculus with cut-elimination LBiInt_1

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
  - a sequent calculus with cut-elimination LBiInt_1
  - a cut-free sequent calculus LBiInt_2

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
  - a sequent calculus with cut-elimination LBiInt_1
  - a cut-free sequent calculus LBiInt_2
  - a complete, terminating proof search strategy

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
  - a sequent calculus with cut-elimination LBiInt_1
  - a cut-free sequent calculus LBiInt_2
  - a complete, terminating proof search strategy
- Further work:

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
    - a sequent calculus with cut-elimination LBiInt_1
    - a cut-free sequent calculus LBiInt_2
    - a complete, terminating proof search strategy
- Further work:
    - Bridge gap between LBiInt_1 and LBiInt_2

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
  - a sequent calculus with cut-elimination LBiInt_1
  - a cut-free sequent calculus LBiInt_2
  - a complete, terminating proof search strategy
- Further work:
  - Bridge gap between LBiInt_1 and LBiInt_2
  - Generalise LBiInt to other similar logics (KtS4, S5, bi-Lambek, $\cdots$)

Conclusion and Further Work

- Bi-Intuitionistic logic presents challenges for proof search
- Nested sequents give:
  - a sequent calculus with cut-elimination $LBiInt_1$
  - a cut-free sequent calculus $LBiInt_2$
  - a complete, terminating proof search strategy
- Further work:
  - Bridge gap between $LBiInt_1$ and $LBiInt_2$
  - Generalise LBiInt to other similar logics (KtS4, S5, bi-Lambek, $\cdots$)
  - Explore connection with type theory