A Hyper-sequent Calculus for INL

Yu, Junhua

Tsinghua University

2017.04.13 @ Zhejiang University

Outline

- Backgrounds
  - Neighborhood semantics & ‘Basic’ neighborhood logic NL
  - ‘Instantial’ neighborhood logic INL
  - Expressive power & Axiomatization
- Proof Theory
  - Semantic tableau & Hyper-sequent calculus HSinl
  - Soundness, (*Cut*)-admissibility, & Completeness
  - Lyndon interpolation
- Future directions

Abbreviation: “nbd” means “neighborhood”

Background

Joint work with

Johan van Benthem, Nick Bezhanishvili, Sebastian Enqvist

Nbd semantics

- Frame: $\mathfrak{F} = (W, \sigma)$
  - $W \neq \varnothing$, a domain;
  - $\sigma : W \mapsto 2^{2^W}$, a nbd function.
- Model: $\mathfrak{M} = (\mathfrak{F}, V)$
  - $\mathfrak{F}$, a nbd frame;
  - $V : W \mapsto 2^{\mathcal{P}}$, a propositional valuation.
- Remarks:
  - Nbd semantics is general
  - Specified properties of nbd functions
    - each state has a nbd,
    - $\{w\}$ is a nbd of w (resp. $\varnothing$, W, ...),
    - each nbd is non-empty,
    - each nbd of w contains w,
    - each state has exactly 1 nbd,
    - nbd is closed under ... .

Nbd semantics

- Frame: $\mathfrak{F} = (W, \sigma)$
  - $W \neq \varnothing$, a domain;
  - $\sigma : W \mapsto 2^{2^W}$, a nbd function.
- Model: $\mathfrak{M} = (\mathfrak{F}, V)$
  - $\mathfrak{F}$, a nbd frame;
  - $V : W \mapsto 2^{\mathcal{P}}$, a propositional valuation.
- Remarks:
  - Nbd semantics is general
  - Specified properties of nbd functions
    - each state has a nbd,
    - $\{w\}$ is a nbd of w (resp. $\varnothing$, W, ...),
    - each nbd is non-empty,
    - each nbd of w contains w,
    - each state has exactly 1 nbd,
    - nbd is closed under ... .

Nbd semantics

- Frame: $\mathfrak{F} = (W, \sigma)$
  - $W \neq \varnothing$, a domain;
  - $\sigma : W \mapsto 2^{2^W}$, a nbd function.
- Model: $\mathfrak{M} = (\mathfrak{F}, V)$
  - $\mathfrak{F}$, a nbd frame;
  - $V : W \mapsto 2^{\mathcal{P}}$, a propositional valuation.
- Remarks:
  - Nbd semantics is general
  - Specified properties of nbd functions
    - each state has a nbd,
    - $\{w\}$ is a nbd of w (resp. $\varnothing$, W, ...),
    - each nbd is non-empty,
    - each nbd of w contains w,
    - each state has exactly 1 nbd,
    - nbd is closed under ... .

Basic nbd logic NL

- Basic modal language: unary operator $\Box$ ($\Diamond$ as defined).
- Truth definition - a $\exists\forall$ reading of $\Box$:
  - $\mathfrak{M}, w \models \Box\alpha$ iff $(\exists N \in \sigma(w))(\forall n \in N)\, \mathfrak{M}, n \models \alpha$.
  - a neighborhood (of the current state) has α true everywhere inside.
- Some schemes of normal K are NOT valid:
  - $\not\models \Box(p \to q) \to (\Box p \to \Box q)$,
  - $\not\models (\Box p \wedge \Box q) \to \Box(p \wedge q)$,
  - (Nec) $\quad (\models \phi) \not\Rightarrow (\models \Box\phi)$.

Basic nbd logic NL

- Basic modal language: unary operator $\Box$ ($\Diamond$ as defined).
- Truth definition - a $\exists\forall$ reading of $\Box$:
  - $\mathfrak{M}, w \models \Box\alpha$ iff $(\exists N \in \sigma(w))(\forall n \in N)\, \mathfrak{M}, n \models \alpha$.
  - a neighborhood (of the current state) has α true everywhere inside.
- Some schemes of normal K are NOT valid:
  - $\not\models \Box(p \to q) \to (\Box p \to \Box q)$,
  - $\not\models (\Box p \wedge \Box q) \to \Box(p \wedge q)$,
  - $(Nec) \quad (\models \phi) \not\Rightarrow (\models \Box\phi)$.

Basic nbd logic NL

- Basic modal language: unary operator $\Box$ ($\Diamond$ as defined).
- Truth definition - a $\exists\forall$ reading of $\Box$:
  - $\mathfrak{M}, w \models \Box\alpha$ iff $(\exists N \in \sigma(w))(\forall n \in N)\, \mathfrak{M}, n \models \alpha$.
  - a neighborhood (of the current state) has α true everywhere inside.
- Some schemes of normal K are NOT valid:
  - $\not\models \Box(p \to q) \to (\Box p \to \Box q)$,
  - $\not\models (\Box p \wedge \Box q) \to \Box(p \wedge q)$,
  - $(Nec) \quad (\models \phi) \not\Rightarrow (\models \Box\phi)$.

Basic nbd logic NL

- Axiomatization:
  - (axiom and rule) Schemes of classical propositional calculus.
  - Rule scheme *RE* (rule of replacement)

$$\frac{\alpha \leftrightarrow \beta \qquad \phi}{\phi'}$$

where ϕ' is ϕ with an occurrance of α replaced by β

  - $\Box(\alpha \wedge \beta) \rightarrow \Box\alpha \wedge \Box\beta$.
    - An $\alpha \wedge \beta$ neighborhood is also an α neighborhood.

- Axiomatization:
  - (axiom and rule) Schemes of classical propositional calculus.
  - Rule scheme *RE* (rule of replacement)

$$\frac{\alpha \leftrightarrow \beta \qquad \phi}{\phi'}$$

where ϕ' is ϕ with an occurrance of α replaced by β

  - $\Box(\alpha \wedge \beta) \rightarrow \Box\alpha \wedge \Box\beta$.
    - An $\alpha \wedge \beta$ neighborhood is also an α neighborhood.

Instantial nbd logic INL

- Same frames/models with an "instantial" language:
  - Operator (with any positive finite arity) $\Box(\alpha_i, \ldots, \alpha_j; \alpha_0)$.
- Truth definition - a "$\exists(\exists, \ldots, \exists; \forall)$" reading of $\Box$:
  - $\mathfrak{M}, w \vDash \Box(\alpha_1, \ldots, \alpha_j; \alpha_0)$ iff

$$(\exists N \in \sigma(w)) \begin{cases} (\forall n \in N)\, \mathfrak{M}, n \vDash \alpha_0 \\ (\exists n_1 \in N)\, \mathfrak{M}, n_1 \vDash \alpha_1 \\ \vdots \\ (\exists n_j \in N)\, \mathfrak{M}, n_j \vDash \alpha_j \end{cases}$$

  - a neighborhood (of the current state) has
    - α_0 true everywhere inside, and
    - α_i true somewhere inside (resp. for each $i \in \{1, \ldots, j\}$).

Instantial nbd logic INL

- Same frames/models with an "instantial" language:
  - Operator (with any positive finite arity) $\Box(\alpha_i, ..., \alpha_j; \alpha_0)$.
- Truth definition - a "$\exists(\exists, ..., \exists; \forall)$" reading of $\Box$:
  - $\mathfrak{M}, w \vDash \Box(\alpha_1, ..., \alpha_j; \alpha_0)$ iff

$$(\exists N \in \sigma(w)) \begin{cases} (\forall n \in N)\, \mathfrak{M}, n \vDash \alpha_0 \\ (\exists n_1 \in N)\, \mathfrak{M}, n_1 \vDash \alpha_1 \\ \vdots \\ (\exists n_j \in N)\, \mathfrak{M}, n_j \vDash \alpha_j \end{cases}$$

  - a neighborhood (of the current state) has
    - α_0 true everywhere inside, and
    - α_i true somewhere inside (resp. for each $i \in \{1, ..., j\}$).

Instantial nbd logic INL

- Some invalid schemes:
  - $\not\models \neg\Box(;\bot)$ (empty neighborhoods are permitted)
    - cf. a validity: $\models \neg\Box(\alpha;\bot)$.
  - $\not\models \Box(;\top)$ (a state can have no neighborhoods).
  - $\not\models \Box(\alpha;\psi)\wedge\Box(\beta;\psi)\rightarrow\Box(\alpha,\beta;\psi)$ (neighborhoods given by premises may be distinct).
- Also, there are valid schemes.
  - An axiomatization later.
  - Reducible to NL? NO.

Instantial nbd logic INL

- Some invalid schemes:
  - $\not\models \neg\Box(;\bot)$ (empty neighborhoods are permitted)
    - cf. a validity: $\models \neg\Box(\alpha;\bot)$.
  - $\not\models \Box(;\top)$ (a state can have no neighborhoods).
  - $\not\models \Box(\alpha;\psi) \wedge \Box(\beta;\psi) \to \Box(\alpha,\beta;\psi)$ (neighborhoods given by premises may be distinct).
- Also, there are valid schemes.
  - An axiomatization later.
  - Reducible to NL? NO.

Instantial nbd logic INL

- Some invalid schemes:
  - $\not\models \neg\Box(;\bot)$ (empty neighborhoods are permitted)
    - cf. a validity: $\models \neg\Box(\alpha;\bot)$.
  - $\not\models \Box(;\top)$ (a state can have no neighborhoods).
  - $\not\models \Box(\alpha;\psi) \wedge \Box(\beta;\psi) \rightarrow \Box(\alpha,\beta;\psi)$

    (neighborhoods given by premises may be distinct).
- Also, there are valid schemes.
  - An axiomatization later.
  - Reducible to NL? NO.

Instantial nbd logic INL

- Some invalid schemes:
  - $\not\models \neg\Box(;\bot)$ (empty neighborhoods are permitted)
    - cf. a validity: $\models \neg\Box(\alpha;\bot)$.
  - $\not\models \Box(;\top)$ (a state can have no neighborhoods).
  - $\not\models \Box(\alpha;\psi) \wedge \Box(\beta;\psi) \rightarrow \Box(\alpha,\beta;\psi)$

    (neighborhoods given by premises may be distinct).
- Also, there are valid schemes.
  - An axiomatization later.
  - Reducible to NL? NO.

Instantial nbd logic INL

- Some invalid schemes:
  - $\not\models \neg\Box(;\bot)$ (empty neighborhoods are permitted)
    - cf. a validity: $\models \neg\Box(\alpha;\bot)$.
  - $\not\models \Box(;\top)$ (a state can have no neighborhoods).
  - $\not\models \Box(\alpha;\psi) \wedge \Box(\beta;\psi) \rightarrow \Box(\alpha,\beta;\psi)$

    (neighborhoods given by premises may be distinct).
- Also, there are valid schemes.
  - An axiomatization later.
  - Reducible to NL? NO.

Instantial nbd logic INL

- Some invalid schemes:
  - $\nvDash \neg\Box(;\bot)$ (empty neighborhoods are permitted)
    - cf. a validity: $\vDash \neg\Box(\alpha;\bot)$.
  - $\nvDash \Box(;\top)$ (a state can have no neighborhoods).
  - $\nvDash \Box(\alpha;\psi) \wedge \Box(\beta;\psi) \rightarrow \Box(\alpha,\beta;\psi)$

    (neighborhoods given by premises may be distinct).
- Also, there are valid schemes.
  - An axiomatization later.
  - Reducible to NL? NO.

INL - expressive power

- $\Box\phi$ in the basic language can be written as $\Box(;\phi)$.
  - Let $n = 0$ in $\Box(\phi_1, ..., \phi_n; \phi)$.
  - Expressive power of the new language is not weaker than the basic language.
- The new language is

  strictly more expressive

  than the basic one.
  - So axiomatization of INL is not trivial.

INL - expressive power

- $\Box\phi$ in the basic language can be written as $\Box(;\phi)$.
  - Let $n = 0$ in $\Box(\phi_1, ..., \phi_n; \phi)$.
  - Expressive power of the new language is not weaker than the basic language.
- The new language is

  strictly more expressive

  than the basic one.
  - So axiomatization of INL is not trivial.

INL - expressive power

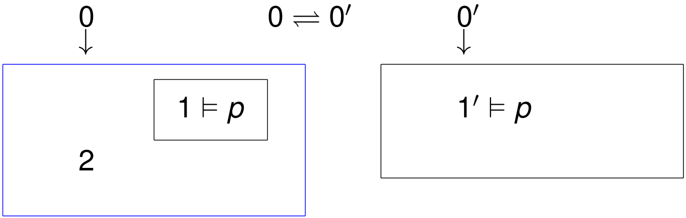
- (Basic bisimulation test): - if $w \rightleftharpoons w'$, i.e.:
  - $V(w) = V'(w')$,
  - $\forall N \in \sigma(w).\exists N' \in \sigma(w').\forall n' \in N'.\exists n \in N.(n \rightleftharpoons n')$,
  - $\forall N' \in \sigma(w').\exists N \in \sigma(w).\forall n \in N.\exists n' \in N'.(n \rightleftharpoons n')$;

  then w and w' agree on all formulas in the basic language.
- No longer capable in the instantialbe setting:

INL - expressive power

- (Basic bisimulation test): - if $w \rightleftharpoons w'$, i.e.:
  - $V(w) = V'(w')$,
  - $\forall N \in \sigma(w).\exists N' \in \sigma(w').\forall n' \in N'.\exists n \in N.(n \rightleftharpoons n')$,
  - $\forall N' \in \sigma(w').\exists N \in \sigma(w).\forall n \in N.\exists n' \in N'.(n \rightleftharpoons n')$;

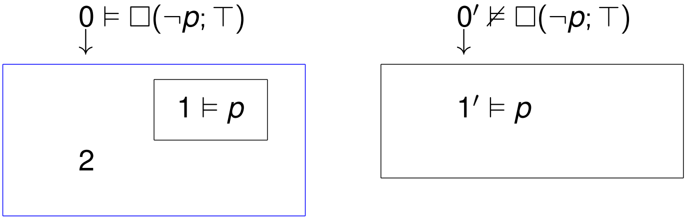
  then w and w' agree on all formulas in the basic language.
- No longer capable in the instantialbe setting:



INL - expressive power

- (Basic bisimulation test): - if $w \rightleftharpoons w'$, i.e.:
  - $V(w) = V'(w')$,
  - $\forall N \in \sigma(w).\exists N' \in \sigma(w').\forall n' \in N'.\exists n \in N.(n \rightleftharpoons n')$,
  - $\forall N' \in \sigma(w').\exists N \in \sigma(w).\forall n \in N.\exists n' \in N'.(n \rightleftharpoons n')$;

  then w and w' agree on all formulas in the basic language.
- No longer capable in the instantialbe setting:



INL - expressive power

- B.t.w., an instantial bisimulation should should take care of both directions:
  - $V(w) = V'(w')$,
  - if $\forall N \in \sigma(w).\exists N' \in \sigma(w')$.
    $[[\forall n' \in N'.\exists n \in N.(n \rightleftarrows n')]\&[\forall n \in N.\exists n' \in N'.(n \rightleftarrows n')]]$,
  - if $\forall N' \in \sigma(w').\exists N \in \sigma(w)$.
    $[[\forall n \in N.\exists n' \in N'.(n \rightleftarrows n')]\&[\forall n' \in N'.\exists n \in N.(n \rightleftarrows n')]]$.

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \to \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \to \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \to \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R – mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L – mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Classical propositional logic with rule scheme *RE*;
- Additional schemes:
  - *R − mon*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \vee \eta)$
  - *L − mon*:
    $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \vee \psi; \alpha_0)$
  - *Inst*:
    $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \wedge \alpha_0; \alpha_0)$
  - *Norm*:
    $\neg\Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)$
  - *Case*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \vee \Box(\alpha_1, ..., \alpha_j; \alpha_0 \wedge \neg\delta))$
  - *Weak*:
    $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
  - *Dupl*:
    $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, ..., j\}$

INL - axiomatization

- Some Derivable Schemes:
  - $\vdash \Box(\alpha_1, ..., \alpha_i, \gamma, \delta, \beta_1, ..., \beta_j; \psi) \rightarrow \Box(\alpha_1, ..., \alpha_i, \delta, \gamma, \beta_1, ..., \beta_j; \psi)$
    - Together with *Weak* and *Dupl*, we can read 'instance-formulas' as a finite set.
  - $\vdash \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \top; \alpha_0)$, when $j > 0$
    - Not valid when $j = 0$.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j; \phi) \rightarrow \Box(\alpha_1, ..., \alpha_j; \psi)}$
    - $R - mon$ as a rule scheme.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \psi; \alpha_0)}$
    - $L - mon$ as a rule scheme.

INL - axiomatization

- Some Derivable Schemes:
  - $\vdash \Box(\alpha_1, ..., \alpha_i, \gamma, \delta, \beta_1, ..., \beta_j; \psi) \rightarrow \Box(\alpha_1, ..., \alpha_i, \delta, \gamma, \beta_1, ..., \beta_j; \psi)$
    - Together with *Weak* and *Dupl*, we can read 'instance-formulas' as a finite set.
  - $\vdash \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \top; \alpha_0)$, when $j > 0$
    - Not valid when $j = 0$.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j; \phi) \rightarrow \Box(\alpha_1, ..., \alpha_j; \psi)}$
    - $R - mon$ as a rule scheme.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \psi; \alpha_0)}$
    - $L - mon$ as a rule scheme.

INL - axiomatization

- Some Derivable Schemes:
  - $\vdash \Box(\alpha_1, ..., \alpha_i, \gamma, \delta, \beta_1, ..., \beta_j; \psi) \rightarrow \Box(\alpha_1, ..., \alpha_i, \delta, \gamma, \beta_1, ..., \beta_j; \psi)$
    - Together with *Weak* and *Dupl*, we can read 'instance-formulas' as a finite set.
  - $\vdash \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \top; \alpha_0)$, when $j > 0$
    - Not valid when $j = 0$.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j; \phi) \rightarrow \Box(\alpha_1, ..., \alpha_j; \psi)}$
    - $R - mon$ as a rule scheme.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \psi; \alpha_0)}$
    - $L - mon$ as a rule scheme.

INL - axiomatization

- Some Derivable Schemes:
  - $\vdash \Box(\alpha_1, ..., \alpha_i, \gamma, \delta, \beta_1, ..., \beta_j; \psi) \to \Box(\alpha_1, ..., \alpha_i, \delta, \gamma, \beta_1, ..., \beta_j; \psi)$
    - Together with *Weak* and *Dupl*, we can read 'instance-formulas' as a finite set.
  - $\vdash \Box(\alpha_1, ..., \alpha_j; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \top; \alpha_0)$, when $j > 0$
    - Not valid when $j = 0$.
  - $\dfrac{\phi \to \psi}{\Box(\alpha_1, ..., \alpha_j; \phi) \to \Box(\alpha_1, ..., \alpha_j; \psi)}$
    - $R - mon$ as a rule scheme.
  - $\dfrac{\phi \to \psi}{\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \to \Box(\alpha_1, ..., \alpha_j, \psi; \alpha_0)}$
    - $L - mon$ as a rule scheme.

INL - axiomatization

- Some Derivable Schemes:
  - $\vdash \Box(\alpha_1, ..., \alpha_i, \gamma, \delta, \beta_1, ..., \beta_j; \psi) \rightarrow \Box(\alpha_1, ..., \alpha_i, \delta, \gamma, \beta_1, ..., \beta_j; \psi)$
    - Together with *Weak* and *Dupl*, we can read ‘instance-formulas’ as a finite set.
  - $\vdash \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \top; \alpha_0)$, when $j > 0$
    - Not valid when $j = 0$.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j; \phi) \rightarrow \Box(\alpha_1, ..., \alpha_j; \psi)}$
    - $R - mon$ as a rule scheme.
  - $\dfrac{\phi \rightarrow \psi}{\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \psi; \alpha_0)}$
    - $L - mon$ as a rule scheme.

Complexity

- Satisfiability problem of INL is *PSPACE*-complete.
  - Faithful embeddings $\mathsf{K} \hookrightarrow \mathsf{INL} \hookrightarrow \mathsf{K} \oplus \mathsf{K}$;
  - Both K and $\mathsf{K} \oplus \mathsf{K}$ are *PSPACE*-complete.

Proof Theory

Semantic tableau

- General idea of semantic tableau
  - In order to prove ϕ, start with the goal of satisfying $\neg\phi$
  - Reduce goals to subgoals (usually on subformulas)
    - Rules
  - Impossible goals are "closed", otherwise "open"
    - Impossible - have $\perp$ or 'both α and $\neg\alpha$';
    - "Open" tableaus provide hints to counter-models (of ϕ);
    - "Closed" tableaus are defined as proofs (of ϕ).
- Rules for classical propositional logic

  ||...|| means branching

$$\frac{\neg\neg\phi}{\phi} \qquad \frac{\alpha \wedge \beta}{\begin{array}{c}\alpha\\ \beta\end{array}} \qquad \frac{\neg(\alpha \vee \beta)}{\begin{array}{c}\neg\alpha\\ \neg\beta\end{array}} \qquad \frac{\neg(\alpha \rightarrow \beta)}{\begin{array}{c}\alpha\\ \neg\beta\end{array}} \qquad \frac{\neg(\alpha \wedge \beta)}{||\neg\alpha\,||\,\neg\beta||} \qquad \frac{\alpha \vee \beta}{||\alpha\,||\,\beta||} \qquad \frac{\alpha \rightarrow \beta}{||\neg\alpha\,||\,\beta||}$$

- General idea of semantic tableau
  - In order to prove ϕ, start with the goal of satisfying $\neg\phi$
  - Reduce goals to subgoals (usually on subformulas)
    - Rules
  - Impossible goals are “closed”, otherwise “open”
    - Impossible - have $\bot$ or ‘both α and $\neg\alpha$’;
    - “Open” tableaus provide hints to counter-models (of ϕ);
    - “Closed” tableaus are defined as proofs (of ϕ).
- Rules for classical propositional logic

  ||...|| means branching

$$\frac{\neg\neg\phi}{\phi} \quad \frac{\alpha\wedge\beta}{\begin{matrix}\alpha\\ \beta\end{matrix}} \quad \frac{\neg(\alpha\vee\beta)}{\begin{matrix}\neg\alpha\\ \neg\beta\end{matrix}} \quad \frac{\neg(\alpha\to\beta)}{\begin{matrix}\alpha\\ \neg\beta\end{matrix}} \quad \frac{\neg(\alpha\wedge\beta)}{||\neg\alpha||\neg\beta||} \quad \frac{\alpha\vee\beta}{||\alpha||\beta||} \quad \frac{\alpha\to\beta}{||\neg\alpha||\beta||}$$

- General idea of semantic tableau
  - In order to prove ϕ, start with the goal of satisfying $\neg\phi$
  - Reduce goals to subgoals (usually on subformulas)
    - Rules
  - Impossible goals are “closed”, otherwise “open”
    - Impossible - have $\perp$ or ‘both α and $\neg\alpha$’;
    - “Open” tableaus provide hints to counter-models (of ϕ);
    - “Closed” tableaus are defined as proofs (of ϕ).
- Rules for classical propositional logic

  ||...|| means branching

$$\frac{\neg\neg\phi}{\phi} \qquad \frac{\alpha\wedge\beta}{\begin{matrix}\alpha\\ \beta\end{matrix}} \qquad \frac{\neg(\alpha\vee\beta)}{\begin{matrix}\neg\alpha\\ \neg\beta\end{matrix}} \qquad \frac{\neg(\alpha\rightarrow\beta)}{\begin{matrix}\alpha\\ \neg\beta\end{matrix}} \qquad \frac{\neg(\alpha\wedge\beta)}{||\neg\alpha\,||\,\neg\beta||} \qquad \frac{\alpha\vee\beta}{||\alpha\,||\,\beta||} \qquad \frac{\alpha\rightarrow\beta}{||\neg\alpha\,||\,\beta||}$$

Semantic tableau

- INL needs (at least) a modal rule.
  - A $\Box$-formula requires a nbd (with certain properties);
    A $\neg\Box$-formula refutes any nbd (with certain properties).
  - $\Box$'s do not work together to close a goal;
    they each does, together with all $\neg\Box$'s in the same goal.
- The rule takes from a goal:
  - one $\Box$-formula, and
  - and any number of $\neg\Box$-formulas
    (with variant numbers of instances):

$$\Box(\alpha_1, ..., \alpha_j; \alpha_0)$$
$$\neg\Box(\beta_1^1, ..., \beta_{j_1}^1; \beta_0^1)$$
$$\vdots$$
$$\neg\Box(\beta_1^k, ..., \beta_{j_k}^k; \beta_0^k)$$

Semantic tableau

- INL needs (at least) a modal rule.
  - A $\Box$-formula requires a nbd (with certain properties);
    A $\neg\Box$-formula refutes any nbd (with certain properties).
  - $\Box$'s do not work together to close a goal;
    they each does, together with all $\neg\Box$'s in the same goal.
- The rule takes from a goal:
  - one $\Box$-formula, and
  - and any number of $\neg\Box$-formulas

(with variant numbers of instances):

$$\begin{array}{c} \Box(\alpha_1, ..., \alpha_j; \alpha_0) \\ \neg\Box(\beta_1^1, ..., \beta_{j_1}^1; \beta_0^1) \\ \vdots \\ \neg\Box(\beta_1^k, ..., \beta_{j_k}^k; \beta_0^k) \\ \hline \end{array}$$

Semantic tableau

- INL needs (at least) a modal rule.
  - A $\Box$-formula requires a nbd (with certain properties);
    A $\neg\Box$-formula refutes any nbd (with certain properties).
  - $\Box$'s do not work together to close a goal;
    they each does, together with all $\neg\Box$'s in the same goal.
- The rule takes from a goal:
  - one $\Box$-formula, and
  - and any number of $\neg\Box$-formulas

(with variant numbers of instances):

$$\begin{array}{c} \Box(\alpha_1, ..., \alpha_j; \alpha_0) \\ \neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0) \\ \vdots \\ \neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0) \\ \hline \end{array}$$

Semantic tableau

$$\frac{\begin{array}{c}\Box(\alpha_1, ..., \alpha_j; \alpha_0)\\ \neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)\\ \vdots \\ \neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)\end{array}}{\Big\| \; \Big|\alpha_0 \wedge \sigma \qquad \Big|_{\sigma \in \{\alpha_x\}^j_{x=1}} \; \Big\|}$$

- $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ requires a nbd with (generally) j states.
  Each nbd is consistent, if all its states are.
- $\forall i \in \{1, ..., k\}$, $\neg\Box(\beta^i_1, ..., \beta^i_{j_i}; \beta^i_0)$ requires that
  either - β^i_0 fails at some state,
  or - β^i_h fails at each state for some $h \in \{1, ..., j_i\}$.
- $\prod^k_{z=1}(j_z + 1)$ options in total.
  Index possible nbd's by the option it takes, e.g., $\langle I(1), ..., I(k)\rangle$.

$$\Box(\alpha_1, ..., \alpha_j; \alpha_0)$$
$$\neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)$$
$$\vdots$$
$$\neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)$$

$$\overline{\left|\left|\ |\alpha_0 \wedge \sigma \qquad |_{\sigma\in\{\alpha_x\}^j_{x=1}\cup\{\neg\beta^i_0\}}\ \right|\right|}$$

- $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ requires a nbd with (generally) j states.
  Each nbd is consistent, if all its states are.
- $\forall i \in \{1, ..., k\}$, $\neg\Box(\beta^i_1, ..., \beta^i_{j_i}; \beta^i_0)$ requires that
  either - β^i_0 fails at some state,
  or - β^i_h fails at each state for some $h \in \{1, ..., j_i\}$.
- $\prod^k_{z=1}(j_z + 1)$ options in total.
  Index possible nbd's by the option it takes, e.g., $\langle I(1), ..., I(k)\rangle$.

Semantic tableau

$$\Box(\alpha_1, ..., \alpha_j; \alpha_0)$$
$$\neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)$$
$$\vdots$$
$$\neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)$$

$$\Big|\Big| \; |\alpha_0 \wedge \sigma \wedge \quad \neg{\beta^i}_h \,|_{\sigma \in \{\alpha_x\}^j_{x=1} \cup \{\neg\beta^i_0\}} \; \Big|\Big|$$

- $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ requires a nbd with (generally) j states.
  Each nbd is consistent, if all its states are.
- $\forall i \in \{1, ..., k\}$, $\neg\Box(\beta^i_1, ..., \beta^i_{j_i}; \beta^i_0)$ requires that
  either - β^i_0 fails at some state,
  or - β^i_h fails at each state for some $h \in \{1, ..., j_i\}$.
- $\prod^k_{z=1}(j_z + 1)$ options in total.
  Index possible nbd's by the option it takes, e.g., $\langle I(1), ..., I(k)\rangle$.

Semantic tableau

$$\frac{\begin{array}{c}\Box(\alpha_1, ..., \alpha_j; \alpha_0)\\ \neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)\\ \vdots \\ \neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)\end{array}}{\left|\left|\ \left|\alpha_0 \wedge \sigma \wedge \bigwedge_{i\in\{1,...,k\}}^{I(i)\neq 0} \neg\beta^i_{I(i)}\right|_{\sigma\in\{\alpha_x\}^j_{x=1}\cup\{\neg\beta^y_0\}^{I(y)=0}_{y\in\{1,...,k\}}}\ \right|\right|_{I\in\bigotimes^k_{z=1}\{0,...,j_z\}}}$$

- $\Box(\alpha_1, ..., \alpha_j; \alpha_0)$ requires a nbd with (generally) j states.

  Each nbd is consistent, if all its states are.
- $\forall i \in \{1, ..., k\}$, $\neg\Box(\beta^i_1, ..., \beta^i_{j_i}; \beta^i_0)$ requires that

  either - β^i_0 fails at some state,

  or - β^i_h fails at each state for some $h \in \{1, ..., j_i\}$.
- $\prod^k_{z=1}(j_z + 1)$ options in total.

  Index possible nbd's by the option it takes, e.g., $\langle I(1), ..., I(k)\rangle$.

Semantic tableau

$$\frac{\begin{array}{c}\Box(\alpha_1, \ldots, \alpha_j; \alpha_0)\\ \neg\Box(\beta^1_1, \ldots, \beta^1_{j_1}; \beta^1_0)\\ \vdots \\ \neg\Box(\beta^k_1, \ldots, \beta^k_{j_k}; \beta^k_0)\end{array}}{\left|\left| \left|\alpha_0 \wedge \sigma \wedge \bigwedge_{i\in\{1,\ldots,k\}}^{l(i)\neq 0} \neg\beta^i_{l(i)}\right|_{\sigma\in\{\alpha_x\}_{x=1}^{j}\cup\{\neg\beta^y_0\}_{y\in\{1,\ldots,k\}}^{l(y)=0}} \right|\right|_{l\in\bigotimes_{z=1}^{k}\{0,\ldots,j_z\}}}$$

- It is $\prod_{z=1}^{k}(j_z + 1)$-branching

  In order to close a tableau, each branch has to be closed.

  Branch correspond to neighborhoods of the current state.

- Each branch offers a hyper-node

  A collection of regular nodes (labeled by formulas).

  To close a branch, it is enough to close one node in the hyper-node.

  Nodes correspond to states in the neighborhood.

Semantic tableau

$$\frac{\begin{array}{c}\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \\ \neg\Box(\beta_1^1, \ldots, \beta_{j_1}^1; \beta_0^1) \\ \vdots \\ \neg\Box(\beta_1^k, \ldots, \beta_{j_k}^k; \beta_0^k)\end{array}}{\left|\left| \left|\alpha_0 \wedge \sigma \wedge \bigwedge_{i\in\{1,\ldots,k\}}^{I(i)\neq 0} \neg\beta_{I(i)}^i\right|_{\sigma\in\{\alpha_x\}_{x=1}^j \cup \{\neg\beta_0^y\}_{y\in\{1,\ldots,k\}}^{I(y)=0}} \right|\right|_{I\in\bigotimes_{z=1}^k\{0,\ldots,j_z\}}}$$

- It is $\prod_{z=1}^k (j_z + 1)$-branching

  In order to close a tableau, each branch has to be closed.

  Branch correspond to neighborhoods of the current state.

- Each branch offers a hyper-node

  A collection of regular nodes (labeled by formulas).

  To close a branch, it is enough to close one node in the hyper-node.

  Nodes correspond to states in the neighborhood.

Semantic tableau

$$\frac{\begin{array}{c}\Box(\alpha_1, ..., \alpha_j; \alpha_0)\\ \neg\Box(\beta^1_1, ..., \beta^1_{j_1}; \beta^1_0)\\ \vdots \\ \neg\Box(\beta^k_1, ..., \beta^k_{j_k}; \beta^k_0)\end{array}}{\left|\left| \; \left|\alpha_0 \wedge \sigma \wedge \bigwedge_{i\in\{1,...,k\}}^{l(i)\neq 0} \neg\beta^i_{l(i)}\right|_{\sigma\in\{\alpha_x\}^j_{x=1}\cup\{\neg\beta^y_0\}^{l(y)=0}_{y\in\{1,...,k\}}} \; \right|\right|_{l\in\bigotimes^k_{z=1}\{0,...,j_z\}}}$$

- It is destructive

  Formulas (used or not) above the line cannot be used any longer (on this branch) to trigger a rule or to close a branch.
- An example $\vdash \Box(\phi \vee \chi; \theta) \rightarrow \Box(\phi; \theta) \vee \Box(\chi; \theta)$

Semantic tableau

- Call the above mentioned tableau system *TABinl*
  - *TABinl* is sound and complete
    - The direct proof of completeness requires an extraction of counter-model out of a 'systematical-yet-failed' implement of rules, and hence is ugly
  - *TABinl* offers a decision procedure
  - *TABinl* indicates a way to some real proof-theory
    - a hyper sequent calculus

Semantic tableau

- Call the above mentioned tableau system *TABinl*
    - *TABinl* is sound and complete
        - The direct proof of completeness requires an extraction of counter-model out of a ‘systematical-yet-failed’ implement of rules, and hence is ugly
    - *TABinl* offers a decision procedure
    - *TABinl* indicates a way to some real proof-theory

      - a hyper sequent calculus

Semantic tableau

- Call the above mentioned tableau system *TABinl*
  - *TABinl* is sound and complete
    - The direct proof of completeness requires an extraction of counter-model out of a ‘systematical-yet-failed’ implement of rules, and hence is ugly
  - *TABinl* offers a decision procedure
  - *TABinl* indicates a way to some real proof-theory

    - a hyper sequent calculus

Hyper-sequent calculus HSinl

- Primitive connectives: $\{\bot, \rightarrow, \Box\}$ (classical)
- Multi-set-based, G3-style
  - No Exchange
  - Built-in Weakening and Contraction
  - Easier proofs of (*Cut*)-admissibility
- Hyper-sequent
  - $|\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$ - finite multi-set of regular sequents
    'standing for' $\bigvee_{i=1}^{n}((\bigwedge \Gamma_i) \rightarrow (\bigvee \Delta_i))$
  - Two groups of (admissible) structural rules
    (internal & external) (Weakening & Contraction)
    no Exchange
- Intuitive correspondence
  - regular sequents $\sim$ states
  - hyper-sequents $\sim$ nbd's
    - sufficient to prove (close) one sequent (state) in

Hyper-sequent calculus HSinl

- Primitive connectives: $\{\bot, \to, \Box\}$ (classical)
- Multi-set-based, G3-style
  - No Exchange
  - Built-in Weakening and Contraction
  - Easier proofs of (*Cut*)-admissibility
- Hyper-sequent
  - $|\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$ - finite multi-set of regular sequents
    'standing for' $\bigvee_{i=1}^{n}((\bigwedge \Gamma_i) \to (\bigvee \Delta_i))$
  - Two groups of (admissible) structural rules
    (internal & external) (Weakening & Contraction)
    no Exchange
- Intuitive correspondence
  - regular sequents $\sim$ states
  - hyper-sequents $\sim$ nbd's
    - sufficient to prove (close) one sequent (state) in

Hyper-sequent calculus HSinl

- Primitive connectives: $\{\bot, \rightarrow, \Box\}$ (classical)
- Multi-set-based, G3-style
  - No Exchange
  - Built-in Weakening and Contraction
  - Easier proofs of (*Cut*)-admissibility
- Hyper-sequent
  - $|\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$ - finite multi-set of regular sequents
    'standing for' $\bigvee_{i=1}^{n}((\bigwedge \Gamma_i) \rightarrow (\bigvee \Delta_i))$
  - Two groups of (admissible) structural rules
    (internal & external) (Weakening & Contraction)
    no Exchange
- Intuitive correspondence
  - regular sequents $\sim$ states
  - hyper-sequents $\sim$ nbd's
    - sufficient to prove (close) one sequent (state) in

Hyper-sequent calculus HSinl

- Primitive connectives: $\{\bot, \rightarrow, \Box\}$ (classical)
- Multi-set-based, G3-style
  - No Exchange
  - Built-in Weakening and Contraction
  - Easier proofs of (*Cut*)-admissibility
- Hyper-sequent
  - $|\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$ - finite multi-set of regular sequents 'standing for' $\bigvee_{i=1}^{n}((\bigwedge \Gamma_i) \rightarrow (\bigvee \Delta_i))$
  - Two groups of (admissible) structural rules (internal & external) (Weakening & Contraction) no Exchange
- Intuitive correspondence
  - regular sequents $\sim$ states
  - hyper-sequents $\sim$ nbd's
    - sufficient to prove (close) one sequent (state) in

Hyper-sequent calculus HSinl

- Primitive connectives: $\{\bot, \rightarrow, \Box\}$ (classical)
- Multi-set-based, G3-style
  - No Exchange
  - Built-in Weakening and Contraction
  - Easier proofs of (*Cut*)-admissibility
- Hyper-sequent
  - $|\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$ - finite multi-set of regular sequents
    'standing for' $\bigvee_{i=1}^{n}((\bigwedge \Gamma_i) \rightarrow (\bigvee \Delta_i))$
  - Two groups of (admissible) structural rules (internal & external) (Weakening & Contraction) no Exchange
- Intuitive correspondence
  - regular sequents $\sim$ states
  - hyper-sequents $\sim$ nbd's
    - sufficient to prove (close) one sequent (state) in

Hyper-sequent calculus HSinl

- $\dfrac{}{G|\Pi, p \Rightarrow p, \Sigma}(Ax)$ and $\dfrac{}{G|\Pi, \bot \Rightarrow \Sigma}(L\bot)$

  G: meta-variable for sequent multi-sets (hyper-sequents)

- $\dfrac{G|\Pi, \alpha \Rightarrow \beta, \Sigma}{G|\Pi \Rightarrow \alpha \to \beta, \Sigma}(R\to)$

- $\dfrac{[G|\Pi \Rightarrow \alpha, \Sigma][G|\Pi, \beta \Rightarrow \Sigma]}{G|\Pi, \alpha \to \beta \Rightarrow \Sigma}(L\to)$

  [...] stands for branches

- $$\dfrac{\left[\begin{array}{c} \left| \alpha_0, \alpha_x \Rightarrow \left\{\beta^i_{l(i)}\right\}^{l(i)\neq 0}_{i\in\{1,\dots,k\}} \right|_{x\in\{1,\dots,j\}} \\ \left| \alpha_0 \Rightarrow \beta^y_0, \left\{\beta^i_{l(i)}\right\}^{l(i)\neq 0}_{i\in\{1,\dots,k\}} \right|^{l(y)=0}_{y\in\{1,\dots,k\}} \end{array} \right]_{l\in\bigotimes^k_{i=1}\{0,1,\dots,j_i\}}}{G|\Pi, \Box(\alpha_1, \dots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta^i_1, \dots, \beta^i_{j_i})\}^k_{i=1}, \Sigma}(\Box)$$

Hyper-sequent calculus HSInl

- $$\frac{}{G|\Pi, p \Rightarrow p, \Sigma}(Ax) \quad \text{and} \quad \frac{}{G|\Pi, \bot \Rightarrow \Sigma}(L\bot)$$
  G: meta-variable for sequent multi-sets (hyper-sequents)
- $$\frac{G|\Pi, \alpha \Rightarrow \beta, \Sigma}{G|\Pi \Rightarrow \alpha \to \beta, \Sigma}(R\to)$$
- $$\frac{[G|\Pi \Rightarrow \alpha, \Sigma][G|\Pi, \beta \Rightarrow \Sigma]}{G|\Pi, \alpha \to \beta \Rightarrow \Sigma}(L\to)$$
  [...] stands for branches
- $$\frac{\left[\left| \alpha_0, \alpha_x \Rightarrow \left\{ \beta^i_{l(i)} \right\}^{l(i)\neq 0}_{i\in\{1,\ldots,k\}} \right|_{x\in\{1,\ldots,j\}} \quad \left| \alpha_0 \Rightarrow \beta^y_0, \left\{ \beta^i_{l(i)} \right\}^{l(i)\neq 0}_{i\in\{1,\ldots,k\}} \right|^{l(y)=0}_{y\in\{1,\ldots,k\}} \right]_{l\in\bigotimes^k_{i=1}\{0,1,\ldots,j_i\}}}{G|\Pi, \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta^i_1, \ldots, \beta^i_{j_i})\}^k_{i=1}, \Sigma}(\Box)$$

Hyper-sequent calculus HSinl

- $$\frac{}{G|\Pi, p \Rightarrow p, \Sigma}(Ax) \quad \text{and} \quad \frac{}{G|\Pi, \perp \Rightarrow \Sigma}(L\perp)$$
  G: meta-variable for sequent multi-sets (hyper-sequents)
- $$\frac{G|\Pi, \alpha \Rightarrow \beta, \Sigma}{G|\Pi \Rightarrow \alpha \to \beta, \Sigma}(R\to)$$
- $$\frac{[G|\Pi \Rightarrow \alpha, \Sigma][G|\Pi, \beta \Rightarrow \Sigma]}{G|\Pi, \alpha \to \beta \Rightarrow \Sigma}(L\to)$$
  [...] stands for branches
- $$\frac{\left[\begin{array}{c} \left|\alpha_0, \alpha_x \Rightarrow \left\{\beta^i_{l(i)}\right\}^{l(i)\neq 0}_{i\in\{1,\ldots,k\}}\right|_{x\in\{1,\ldots,j\}} \\ \\ \left|\alpha_0 \Rightarrow \beta^y_0, \left\{\beta^i_{l(i)}\right\}^{l(i)\neq 0}_{i\in\{1,\ldots,k\}}\right|^{l(y)=0}_{y\in\{1,\ldots,k\}} \end{array}\right]_{l\in\bigotimes^k_{i=1}\{0,1,\ldots,j_i\}}}{G|\Pi, \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta^i_1, \ldots, \beta^i_{j_i})\}^k_{i=1}, \Sigma}(\Box)$$

Hyper-sequent calculus HSinl

- HSinl is sound.
  - If HSinl $\vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$,
    then INL $\vdash \bigvee_{i=1}^{n}(\bigwedge \Gamma_i \to \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the $(\Box)$ rule, a sub-induction gives a stronger form of what we need.
- HSinl is complete.
  - If INL $\vdash \phi$, then HSinl $\vdash \Rightarrow \phi$.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *MP*
    - A corollary of (*Cut*)-admissibility.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *RE*.

Hyper-sequent calculus HSinl

- HSinl is sound.
  - If HSinl $\vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$,
    then INL $\vdash \bigvee_{i=1}^{n}(\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the $(\Box)$ rule, a sub-induction gives a stronger form of what we need.
- HSinl is complete.
  - If INL $\vdash \phi$, then HSinl $\vdash \Rightarrow \phi$.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *MP*
    - A corollary of (*Cut*)-admissibility.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *RE*.

Hyper-sequent calculus HSinl

- HSinl is sound.
  - If HSinl $\vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$,
    then INL $\vdash \bigvee_{i=1}^{n}(\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the $(\Box)$ rule, a sub-induction gives a stronger form of what we need.
- HSinl is complete.
  - If INL $\vdash \phi$, then HSinl $\vdash \Rightarrow \phi$.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *MP*
    - A corollary of (*Cut*)-admissibility.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *RE*.

Hyper-sequent calculus HSinl

- HSinl is sound.
  - If HSinl $\vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$,
    then INL $\vdash \bigvee_{i=1}^{n}(\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the ($\Box$) rule, a sub-induction gives a stronger form of what we need.
- HSinl is complete.
  - If INL $\vdash \phi$, then HSinl $\vdash \Rightarrow \phi$.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *MP*
    - A corollary of (*Cut*)-admissibility.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *RE*.

Hyper-sequent calculus HSinl

- HSinl is sound.
  - If HSinl $\vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$,
    then INL $\vdash \bigvee_{i=1}^{n}(\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the $(\Box)$ rule, a sub-induction gives a stronger form of what we need.
- HSinl is complete.
  - If INL $\vdash \phi$, then HSinl $\vdash \Rightarrow \phi$.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *MP*
    - A corollary of (*Cut*)-admissibility.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *RE*.

Hyper-sequent calculus HSinl

- HSinl is sound.
  - If HSinl $\vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$,
    then INL $\vdash \bigvee_{i=1}^{n}(\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the $(\Box)$ rule, a sub-induction gives a stronger form of what we need.
- HSinl is complete.
  - If INL $\vdash \phi$, then HSinl $\vdash \Rightarrow \phi$.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *MP*
    - A corollary of (*Cut*)-admissibility.
  - $\{\phi \mid \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under *RE*.

Hyper-sequent calculus HSInl

- Admissibility of (Cut) (no matter (Cut_+) or ($Cut_\times$))
  - In HSInl resp. "HSInl $\oplus$ (Cut_+) of a certain 'degree' ":
    - Internal/External Weakening is d.p.a. (depth-preserved admissible). Actually, in each provable hyper-sequent there is a provable sequent.
    - For each formula α, (hyper-)sequent $\alpha \Rightarrow \alpha$ is provable.
    - External/Internal Contraction is d.p.a.. D.p.a. of External Contraction is used when showing that of Internal Contraction.
  - Based on HSInl, rules (Cut_+) and ($Cut_\times$) (at any same 'degree') are inter-derivable.
  - Then, a standard double-induction works.
- Subformula property of HSInl as a corollary.

Hyper-sequent calculus HSInl

- Admissibility of (Cut) (no matter (Cut_+) or $(Cut_\times)$)
  - In HSInl resp. "HSInl $\oplus$ (Cut_+) of a certain 'degree' ":
    - Internal/External Weakening is d.p.a.
      (depth-preserved admissible).
      Actually, in each provable hyper-sequent there is a provable sequent.
    - For each formula α, (hyper-)sequent $\alpha \Rightarrow \alpha$ is provable.
    - External/Internal Contraction is d.p.a..
      D.p.a. of External Contraction is used when showing that of Internal Contraction.
  - Based on HSInl, rules (Cut_+) and $(Cut_\times)$ (at any same 'degree') are inter-derivable.
  - Then, a standard double-induction works.
- Subformula property of HSInl as a corollary.

Hyper-sequent calculus HSInl

- Admissibility of (Cut) (no matter (Cut_+) or ($Cut_\times$))
  - In HSInl resp. "HSInl $\oplus$ (Cut_+) of a certain 'degree' ":
    - Internal/External Weakening is d.p.a.
      (depth-preserved admissible).
      Actually, in each provable hyper-sequent
      there is a provable sequent.
    - For each formula α, (hyper-)sequent $\alpha \Rightarrow \alpha$ is provable.
    - External/Internal Contraction is d.p.a..
      D.p.a. of External Contraction is used
      when showing that of Internal Contraction.
  - Based on HSInl,
    rules (Cut_+) and ($Cut_\times$) (at any same 'degree') are
    inter-derivable.
  - Then, a standard double-induction works.
- Subformula property of HSInl as a corollary.

- Admissibility of (Cut) (no matter (Cut_{+}) or $(Cut_{\times})$)
  - In HSinl resp. "HSinl $\oplus$ (Cut_{+}) of a certain 'degree' ":
    - Internal/External Weakening is d.p.a.
      (depth-preserved admissible).
      Actually, in each provable hyper-sequent
      there is a provable sequent.
    - For each formula α, (hyper-)sequent $\alpha \Rightarrow \alpha$ is provable.
    - External/Internal Contraction is d.p.a..
      D.p.a. of External Contraction is used
      when showing that of Internal Contraction.
  - Based on HSinl,
    rules (Cut_{+}) and $(Cut_{\times})$ (at any same 'degree') are
    inter-derivable.
  - Then, a standard double-induction works.
- Subformula property of HSinl as a corollary.

Hyper-sequent calculus HSInl

- Admissibility of (Cut) (no matter (Cut_+) or $(Cut_\times)$)
  - In HSInl resp. "HSInl $\oplus$ (Cut_+) of a certain 'degree' ":
    - Internal/External Weakening is d.p.a.
      (depth-preserved admissible).
      Actually, in each provable hyper-sequent
      there is a provable sequent.
    - For each formula α, (hyper-)sequent $\alpha \Rightarrow \alpha$ is provable.
    - External/Internal Contraction is d.p.a..
      D.p.a. of External Contraction is used
      when showing that of Internal Contraction.
  - Based on HSInl,
    rules (Cut_+) and $(Cut_\times)$ (at any same 'degree') are
    inter-derivable.
  - Then, a standard double-induction works.
- Subformula property of HSInl as a corollary.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^{+}(\alpha)/\mathcal{V}^{-}(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \rightarrow \psi$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq \mathcal{V}^{\pm}(\phi) \cap \mathcal{V}^{\pm}(\psi)$
  - INL $\vdash \phi \rightarrow \epsilon$ and INL $\vdash \epsilon \rightarrow \psi$.

  (a 'polar generalization' of Craig interpolation)
- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\mp}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \rightarrow \Sigma_R$.
- Employ a 'splitting version' of HSinl
  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^{+}(\alpha)/\mathcal{V}^{-}(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \rightarrow \psi$, then there is a formula ϵ s.t.:

  - $\mathcal{V}^{\pm}(\epsilon) \subseteq \mathcal{V}^{\pm}(\phi) \cap \mathcal{V}^{\pm}(\psi)$
  - INL $\vdash \phi \rightarrow \epsilon$ and INL $\vdash \epsilon \rightarrow \psi$.

  (a 'polar generalization' of Craig interpolation)

- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:

  - $\mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\mp}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \rightarrow \Sigma_R$.

- Employ a 'splitting version' of HSinl

  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^{+}(\alpha)/\mathcal{V}^{-}(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \rightarrow \psi$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq \mathcal{V}^{\pm}(\phi) \cap \mathcal{V}^{\pm}(\psi)$
  - INL $\vdash \phi \rightarrow \epsilon$ and INL $\vdash \epsilon \rightarrow \psi$.

  (a 'polar generalization' of Craig interpolation)
- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\mp}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \rightarrow \Sigma_R$.
- Employ a 'splitting version' of HSinl
  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^+(\alpha)/\mathcal{V}^-(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \to \psi$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^\pm(\epsilon) \subseteq \mathcal{V}^\pm(\phi) \cap \mathcal{V}^\pm(\psi)$
  - INL $\vdash \phi \to \epsilon$ and INL $\vdash \epsilon \to \psi$.

  (a 'polar generalization' of Craig interpolation)
- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^\pm(\epsilon) \subseteq (\mathcal{V}^\mp(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^\pm(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \to \Sigma_R$.
- Employ a 'splitting version' of HSinl
  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^+(\alpha)/\mathcal{V}^-(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \rightarrow \psi$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq \mathcal{V}^{\pm}(\phi) \cap \mathcal{V}^{\pm}(\psi)$
  - INL $\vdash \phi \rightarrow \epsilon$ and INL $\vdash \epsilon \rightarrow \psi$.

  (a 'polar generalization' of Craig interpolation)

- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\mp}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \rightarrow \Sigma_R$.

- Employ a 'splitting version' of HSinl
  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^+(\alpha)/\mathcal{V}^-(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \to \psi$, then there is a formula ϵ s.t.:

  - $\mathcal{V}^{\pm}(\epsilon) \subseteq \mathcal{V}^{\pm}(\phi) \cap \mathcal{V}^{\pm}(\psi)$
  - INL $\vdash \phi \to \epsilon$ and INL $\vdash \epsilon \to \psi$.

  (a 'polar generalization' of Craig interpolation)

- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:

  - $\mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\mp}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \to \Sigma_R$.

- Employ a 'splitting version' of HSinl
  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

Lyndon interpolation of INL

- Lydon interpolation theorem:

  (Let $\mathcal{V}^{+}(\alpha)/\mathcal{V}^{-}(\alpha)$ denotes positive/negative atoms in α)

  If INL $\vdash \phi \rightarrow \psi$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq \mathcal{V}^{\pm}(\phi) \cap \mathcal{V}^{\pm}(\psi)$
  - INL $\vdash \phi \rightarrow \epsilon$ and INL $\vdash \epsilon \rightarrow \psi$.

  (a 'polar generalization' of Craig interpolation)
- A general form:

  If HSinl $\vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula ϵ s.t.:
  - $\mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\mp}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R))$
  - HSinl $\vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and HSinl $\vdash \epsilon, \Pi_R \rightarrow \Sigma_R$.
- Employ a 'splitting version' of HSinl
  - each rule offers an interpolant of its conclusion built up from those of its premises;
  - cannot be included here in a readable manner.

- Thanks !