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: $\mathcal{F} = (W, \sigma)$
- $W \neq \emptyset$, a domain;
- $\sigma : W \mapsto 2^W$, a nbd function.

Model: $M = (\mathcal{F}, V)$
- $\mathcal{F}$, a nbd frame;
- $V : W \mapsto 2^P$, 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. $\emptyset$, $W$, ...),
  - each nbd is non-empty,
  - each nbd of $w$ contains $w$,
  - each state has exactly 1 nbd,
  - nbd is closed under ... .
Frame: $\mathcal{F} = (W, \sigma)$
- $W \neq \emptyset$, a domain;
- $\sigma : W \mapsto 2^W$, a nbd function.

Model: $\mathcal{M} = (\mathcal{F}, V)$
- $\mathcal{F}$, a nbd frame;
- $V : W \mapsto 2^P$, 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. $\emptyset$, $W$, ...),
  - each nbd is non-empty,
  - each nbd of $w$ contains $w$,
  - each state has exactly 1 nbd,
  - nbd is closed under ... .
Nbd semantics

Frame: $\mathcal{F} = (W, \sigma)$
- $W \neq \emptyset$, a domain;
- $\sigma : W \mapsto 2^W$, a nbd function.

Model: $\mathcal{M} = (\mathcal{F}, V)$
- $\mathcal{F}$, a nbd frame;
- $V : W \mapsto 2^P$, 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. $\emptyset$, $W$, ...),
  - each nbd is non-empty,
  - each nbd of $w$ contains $w$,
  - each state has exactly 1 nbd,
  - nbd is closed under ... .
Basic modal language: unary operator $\square$ ($\Diamond$ as defined).

Truth definition - a $\exists \forall$ reading of $\square$:
- $\mathcal{M}, w \models \square \alpha$ iff $(\exists N \in \sigma(w))(\forall n \in N) \mathcal{M}, n \models \alpha$.
- A neighborhood (of the current state) has $\alpha$ true everywhere inside.

Some schemes of normal K are NOT valid:
- $\not\models \square(p \rightarrow q) \rightarrow (\square p \rightarrow \square q)$,
- $\not\models (\square p \land \square q) \rightarrow \square(p \land q)$,
- $(Nec)$ $(\models \phi) \not\models (\models \square \phi)$.
Basic modal language: unary operator $\Box$ (◊ as defined).

Truth definition - a $\exists\forall$ reading of $\Box$:

- $M, w \models \Box \alpha$ iff $(\exists N \in \sigma(w))(\forall n \in N) M, n \models \alpha$.
- A neighborhood (of the current state) has $\alpha$ true everywhere inside.

Some schemes of normal K are NOT valid:

- $\not\models \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$,
- $\not\models (\Box p \land \Box q) \rightarrow \Box(p \land q)$,
- $(\text{Nec}) (\models \phi) \not\Rightarrow (\models \Box \phi)$.
Basic modal language: unary operator $\square$ ($\Diamond$ as defined).

Truth definition - a $\exists \forall$ reading of $\square$:

- $\mathcal{M}, w \models \square \alpha$ iff $(\exists N \in \sigma(w))(\forall n \in N)\mathcal{M}, n \models \alpha$.
- a neighborhood (of the current state) has $\alpha$ true everywhere inside.

Some schemes of normal K are NOT valid:

- $\not\models \square(p \rightarrow q) \rightarrow (\square p \rightarrow \square q)$,
- $\not\models (\square p \land \square q) \rightarrow \square(p \land q)$,
- $(\text{Nec}) \quad (\models \phi) \not\Rightarrow (\models \square \phi)$. 
Basic nbd logic NL

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

\[
\frac{\alpha \leftrightarrow \beta}{\phi} \phi'
\]

where \( \phi' \) is \( \phi \) with an occurrence of \( \alpha \) replaced by \( \beta \).

- \( \square (\alpha \land \beta) \rightarrow \square \alpha \land \square \beta \).
  - An \( \alpha \land \beta \) neighborhood is also an \( \alpha \) neighborhood.
Axiomatization:

• (axiom and rule) Schemes of classical propositional calculus.
• Rule scheme RE (rule of replacement)

\[
\frac{\alpha \leftrightarrow \beta}{\phi \phi'}
\]

where \( \phi' \) is \( \phi \) with an occurrence of \( \alpha \) replaced by \( \beta \).

• \( \square (\alpha \land \beta) \rightarrow \square \alpha \land \square \beta \).
  • An \( \alpha \land \beta \) neighborhood is also an \( \alpha \) neighborhood.
Same frames/models with an “instantial” language:

- Operator (with any positive finite arity) \( \Box(\alpha_1, ..., \alpha_j; \alpha_0) \).

Truth definition - a “\( \exists(\exists, ..., \exists; \forall) \)” reading of \( \Box \):

- \( M, w \models \Box(\alpha_1, ..., \alpha_j; \alpha_0) \) iff

\[
(\exists N \in \sigma(w)) \left\{ \begin{array}{l}
(\forall n \in N) M, n \models \alpha_0 \\
(\exists n_1 \in N) M, n_1 \models \alpha_1 \\
\vdots \\
(\exists n_j \in N) M, n_j \models \alpha_j
\end{array} \right.
\]

- a neighborhood (of the current state) has
  - \( \alpha_0 \) true everywhere inside, and
  - \( \alpha_j \) true somewhere inside (resp. for each \( i \in \{1, ..., j\} \)).
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 $\Box$:
  $$\mathcal{M}, w \models \Box(\alpha_1, ..., \alpha_j; \alpha_0) \iff (\exists N \in \sigma(w)) \begin{cases} (\forall n \in N) \mathcal{M}, n \models \alpha_0 \\ (\exists n_1 \in N) \mathcal{M}, n_1 \models \alpha_1 \\ \vdots \\ (\exists n_j \in N) \mathcal{M}, n_j \models \alpha_j \end{cases}$$
- a neighborhood (of the current state) has
  - $\alpha_0$ true everywhere inside, and
  - $\alpha_i$ true somewhere inside (resp. for each $i \in \{1, ..., j\}$).
Some invalid schemes:

- \( \not\equiv \neg \Box (\bot) \) (empty neighborhoods are permitted)
  - cf. a validity: \( \models \neg \Box (\alpha; \bot) \).
- \( \not\equiv \Box (\top) \) (a state can have no neighborhoods).
- \( \not\equiv \Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi) \) (neighborhoods given by premises may be distinct).

Also, there are valid schemes.

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

Also, there are valid schemes.
- An axiomatization later.
- Reducible to NL? NO.
Some invalid schemes:

- $\not\models \neg \Box(\perp)$ (empty neighborhoods are permitted)
- cf. a validity: $\models \neg \Box(\alpha; \perp)$.
- $\not\models \Box(\top)$ (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.
Some invalid schemes:

- $\not\models \Box (\bot)$ (empty neighborhoods are permitted)
- cf. a validity: $\models \neg \Box (\alpha; \bot)$.
- $\not\models \Box (\top)$ (a state can have no neighborhoods).
- $\not\models \Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi)$ (neighborhoods given by premises may be distinct).

Also, there are valid schemes.

- An axiomatization later.
- Reducible to NL? NO.
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).
- $\not\models \Box(\alpha; \psi) \land \Box(\beta; \psi) \to \Box(\alpha, \beta; \psi)$ (neighborhoods given by premises may be distinct).

Also, there are valid schemes.
- An axiomatization later.
- Reducible to NL? NO.
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).

- $\not\models \Box(\alpha; \psi) \land \Box(\beta; \psi) \rightarrow \Box(\alpha, \beta; \psi)$
  (neighborhoods given by premises may be distinct).

Also, there are valid schemes.

- An axiomatization later.
- Reducible to NL? NO.
\( \Box \phi \) in the basic language can be written as \( \Box(\,; \phi) \).

- Let \( n = 0 \) in \( \Box(\phi_1, \ldots, \phi_n; \phi) \).

- 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.
\( \square \phi \) in the basic language can be written as \( \square(\; \phi) \).
- Let \( n = 0 \) in \( \square(\phi_1, ..., \phi_n; \phi) \).
- 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.
(Basic bisimulation test): - if \( w \equiv w' \), i.e.:

- \( V(w) = V'(w') \),
- \( \forall N \in \sigma(w). \exists N' \in \sigma(w'). \forall n' \in N'. \exists n \in N. (n \equiv n') \),
- \( \forall N' \in \sigma(w'). \exists N \in \sigma(w). \forall n \in N. \exists n' \in N'. (n \equiv n') \);

then \( w \) and \( w' \) agree on all formulas in the basic language.

No longer capable in the instantiable setting:
(Basic bisimulation test): - if $w \bisim w'$, i.e.:

- $V(w) = V'(w')$,
- $\forall N \in \sigma(w). \exists N' \in \sigma(w'). \forall n' \in N'. \exists n \in N. (n \bisim n')$,
- $\forall N' \in \sigma(w'). \exists N \in \sigma(w). \forall n \in N. \exists n' \in N'. (n \bisim n')$;

then $w$ and $w'$ agree on all formulas in the basic language.

No longer capable in the instantiaable setting:
(Basic bisimulation test): - if $w \leadsto w'$, i.e.:
- $V(w) = V'(w')$,
- $\forall N \in \sigma(w). \exists N' \in \sigma(w'). \forall n' \in N'. \exists n \in N. (n \leadsto n')$,
- $\forall N' \in \sigma(w'). \exists N \in \sigma(w). \forall n \in N. \exists n' \in N'. (n \leadsto n')$;

then $w$ and $w'$ agree on all formulas in the basic language.

No longer capable in the instantialbe setting:

\[
0 \vdash \square(\neg p; \top) \\
\downarrow \\
1 \vdash p \\
2
\]

\[
0' \not\vdash \square(\neg p; \top) \\
\downarrow \\
1' \vdash p
\]
B.t.w., an **instantial** bisimulation should should take care of both directions:

- \( V(w) = V'(w') \),
- if \( \forall N \in \sigma(w). \exists N' \in \sigma(w') \).
- \( \forall n' \in N'. \exists n \in N. (n \leftrightarrow n') \) \& \( \forall n \in N. \exists n' \in N'. (n \leftrightarrow n') \),
- if \( \forall N' \in \sigma(w'). \exists N \in \sigma(w) \).
- \( \forall n \in N. \exists n' \in N'. (n \leftrightarrow n') \) \& \( \forall n' \in N'. \exists n \in N. (n \leftrightarrow n') \).
Classical propositional logic with rule scheme $RE$;

Additional schemes:

- $R$ – mon:
  $\Box(\alpha_1,\ldots,\alpha_j;\alpha_0) \rightarrow \Box(\alpha_1,\ldots,\alpha_j;\alpha_0 \lor \eta)$

- $L$ – mon:
  $\Box(\alpha_1,\ldots,\alpha_j,\phi;\alpha_0) \rightarrow \Box(\alpha_1,\ldots,\alpha_j,\phi \lor \psi;\alpha_0)$

- Inst:
  $\Box(\alpha_1,\ldots,\alpha_j,\eta;\alpha_0) \rightarrow \Box(\alpha_1,\ldots,\alpha_j,\eta \land \alpha_0;\alpha_0)$

- Norm:
  $\neg \Box(\alpha_1,\ldots,\alpha_j,\bot;\alpha_0)$

- Case:
  $\Box(\alpha_1,\ldots,\alpha_j;\alpha_0) \rightarrow (\Box(\alpha_1,\ldots,\alpha_j,\delta;\alpha_0) \lor \Box(\alpha_1,\ldots,\alpha_j;\alpha_0 \land \neg \delta))$

- Weak:
  $\Box(\alpha_1,\alpha_2,\ldots,\alpha_j;\alpha_0) \rightarrow \Box(\alpha_2,\ldots,\alpha_j;\alpha_0)$

- Dupl:
  $\Box(\alpha_1,\ldots,\alpha_j;\alpha_0) \rightarrow \Box(\alpha_1,\ldots,\alpha_{j-1},\alpha_i;\alpha_0)$ where $i \in \{1,\ldots,j\}$
Classical propositional logic with rule scheme $RE$;

Additional schemes:

- **$R – mon$:**
  \[ \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \]

- **$L – mon$:**
  \[ \Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0) \]

- **$Inst$:**
  \[ \Box (\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \]

- **$Norm$:**
  \[ \neg \Box (\alpha_1, \ldots, \alpha_j, \bot; \alpha_0) \]

- **$Case$:**
  \[ \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\Box (\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \]

- **$Weak$:**
  \[ \Box (\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_2, \ldots, \alpha_j; \alpha_0) \]

- **$Dupl$:**
  \[ \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0) \quad \text{where } i \in \{1, \ldots, j\} \]
Classical propositional logic with rule scheme \( RE \);

**Additional schemes:**

- **\( R – \text{mon} \):**
  \[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \]

- **\( L – \text{mon} \):**
  \[ \Box(\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0) \]

- **\( \text{Inst} \):**
  \[ \Box(\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \]

- **\( \text{Norm} \):**
  \[ \neg \Box(\alpha_1, \ldots, \alpha_j, \bot; \alpha_0) \]

- **\( \text{Case} \):**
  \[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \Box(\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \]

- **\( \text{Weak} \):**
  \[ \Box(\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, \ldots, \alpha_j; \alpha_0) \]

- **\( \text{Dupl} \):**
  \[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0) \quad \text{where } i \in \{1, \ldots, j\} \]
Classical propositional logic with rule scheme \(RE\);

**Additional schemes:**

- **\(R - mon\):**
  \[\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \to \square(\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta)\]

- **\(L - mon\):**
  \[\square(\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \to \square(\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0)\]

- **Inst:**
  \[\square(\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \to \square(\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0)\]

- **Norm:**
  \[\neg \square(\alpha_1, \ldots, \alpha_j, \bot; \alpha_0)\]

- **Case:**
  \[\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \to (\square(\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \square(\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta))\]

- **Weak:**
  \[\square(\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \to \square(\alpha_2, \ldots, \alpha_j; \alpha_0)\]

- **Dupl:**
  \[\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \to \square(\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0)\]
  where \(i \in \{1, \ldots, j\}\)
INL - axiomatization

- Classical propositional logic with rule scheme $RE$;
- Additional schemes:
  - $R - \text{mon}$:
    \[
    \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \lor \eta)
    \]
  - $L - \text{mon}$:
    \[
    \Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \lor \psi; \alpha_0)
    \]
  - $\text{Inst}$:
    \[
    \Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \land \alpha_0; \alpha_0)
    \]
  - $\text{Norm}$:
    \[
    \neg \Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0)
    \]
  - $\text{Case}$:
    \[
    \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \lor \Box(\alpha_1, ..., \alpha_j; \alpha_0 \land \neg \delta))
    \]
  - $\text{Weak}$:
    \[
    \Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)
    \]
  - $\text{Dupl}$:
    \[
    \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0) \quad \text{where } i \in \{1, ..., j\}
    \]
INL - axiomatization

- Classical propositional logic with rule scheme $RE$;
- Additional schemes:
  - $R$ – mon:
    \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \]
  - $L$ – mon:
    \[ \square(\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0) \]
  - $Inst$:
    \[ \square(\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \]
  - $Norm$:
    \[ \neg \square(\alpha_1, \ldots, \alpha_j, \bot; \alpha_0) \]
  - $Case$:
    \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\square(\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \square(\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \]
  - $Weak$:
    \[ \square(\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_2, \ldots, \alpha_j; \alpha_0) \]
  - $Dupl$:
    \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0) \quad \text{where } i \in \{1, \ldots, j\} \]
INL - axiomatization

- Classical propositional logic with rule scheme $RE$;
- Additional schemes:
  - $R - mon$:
    \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \]
  - $L - mon$:
    \[ \square(\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0) \]
  - $Inst$:
    \[ \square(\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \]
  - $Norm$:
    \[ \neg \square(\alpha_1, \ldots, \alpha_j, \bot; \alpha_0) \]
  - $Case$:
    \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\square(\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \square(\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \]
  - $Weak$:
    \[ \square(\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_2, \ldots, \alpha_j; \alpha_0) \]
  - $Dupl$:
    \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0) \quad \text{where } i \in \{1, \ldots, j\} \]
Classical propositional logic with rule scheme $RE$;

Additional schemes:

- $R - mon$:
  \[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \]

- $L - mon$:
  \[ \Box(\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0) \]

- $Inst$:
  \[ \Box(\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \]

- $Norm$:
  \[ \neg \Box(\alpha_1, \ldots, \alpha_j, \bot; \alpha_0) \]

- $Case$:
  \[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \Box(\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \]

- $Weak$:
  \[ \Box(\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, \ldots, \alpha_j; \alpha_0) \]

- $Dupl$:
  \[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0) \quad \text{where } i \in \{1, \ldots, j\} \]
Some Derivable Schemes:

\[\vdash \square(\alpha_1, \ldots, \alpha_i, \gamma, \delta, \beta_1, \ldots, \beta_j; \psi) \rightarrow \square(\alpha_1, \ldots, \alpha_i, \delta, \gamma, \beta_1, \ldots, \beta_j; \psi)\]

Together with *Weak* and *Dupl*, we can read ‘instance-formulas’ as a finite set.

\[\vdash \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j; \top; \alpha_0), \text{ when } j > 0\]

Not valid when \(j = 0\).

\[\phi \rightarrow \psi\]
\[\square(\alpha_1, \ldots, \alpha_j; \phi) \rightarrow \square(\alpha_1, \ldots, \alpha_j; \psi)\]

*R - mon* as a rule scheme.

\[\phi \rightarrow \psi\]
\[\square(\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \square(\alpha_1, \ldots, \alpha_j, \psi; \alpha_0)\]

*L - mon* as a rule scheme.
Some Derivable Schemes:

\[ \vdash □(α_1, ..., α_i, γ, δ, β_1, ..., β_j; ψ) \rightarrow □(α_1, ..., α_i, δ, γ, β_1, ..., β_j; ψ) \]

Together with Weak and Dupl, we can read ‘instance-formulas’ as a finite set.

\[ \vdash □(α_1, ..., α_j; α_0) \rightarrow □(α_1, ..., α_j, T; α_0), \text{ when } j > 0 \]

Not valid when \( j = 0 \).

\[ \phi \rightarrow ψ \]

\[ □(α_1, ..., α_j; ψ) \rightarrow □(α_1, ..., α_j; ψ) \]

\( R \) – mon as a rule scheme.

\[ \phi \rightarrow ψ \]

\[ □(α_1, ..., α_j, φ; α_0) \rightarrow □(α_1, ..., α_j, ψ; α_0) \]

\( L \) – mon as a rule scheme.
Some Derivable Schemes:

\[ \vdash \Box (\alpha_1, \ldots, \alpha_i, \gamma, \delta, \beta_1, \ldots, \beta_j; \psi) \rightarrow \Box (\alpha_1, \ldots, \alpha_i, \delta, \gamma, \beta_1, \ldots, \beta_j; \psi) \]

Together with \textit{Weak} and \textit{Dupl}, we can read ‘instance-formulas’ as a finite set.

\[ \vdash \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \top; \alpha_0), \text{ when } j > 0 \]

Not valid when \( j = 0 \).

\[ \phi \rightarrow \psi \]

\[ \Box (\alpha_1, \ldots, \alpha_j; \phi) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \psi) \]

\textit{R – mon} as a rule scheme.

\[ \phi \rightarrow \psi \]

\[ \Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \psi; \alpha_0) \]

\textit{L – mon} as a rule scheme.
Some Derivable Schemes:

\[ \vdash \Box (\alpha_1, \ldots, \alpha_i, \gamma, \delta, \beta_1, \ldots, \beta_j; \psi) \rightarrow \Box (\alpha_1, \ldots, \alpha_i, \delta, \gamma, \beta_1, \ldots, \beta_j; \psi) \]

Together with \textit{Weak} and \textit{Dupl}, we can read ‘instance-formulas’ as a finite set.

\[ \vdash \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \top; \alpha_0), \text{ when } j > 0 \]

Not valid when \( j = 0 \).

\[ \phi \rightarrow \psi \]

\[ \Box (\alpha_1, \ldots, \alpha_j; \phi) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \psi) \]

\textit{R} – \textit{mon} as a rule scheme.

\[ \phi \rightarrow \psi \]

\[ \Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \psi; \alpha_0) \]

\textit{L} – \textit{mon} as a rule scheme.
Some Derivable Schemes:

- $\vdash \Box (\alpha_1, \ldots, \alpha_i, \gamma, \delta, \beta_1, \ldots, \beta_j; \psi) \rightarrow \Box (\alpha_1, \ldots, \alpha_i, \delta, \gamma, \beta_1, \ldots, \beta_j; \psi)$
  - Together with *Weak* and *Dupl*, we can read ‘instance-formulas’ as a finite set.

- $\vdash \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \top; \alpha_0)$, when $j > 0$
  - Not valid when $j = 0$. 

\[ \phi \rightarrow \psi \]

$\Box (\alpha_1, \ldots, \alpha_j; \phi) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \psi)$

- $R – mon$ as a rule scheme.

$\phi \rightarrow \psi$

$\Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \psi; \alpha_0)$

- $L – mon$ as a rule scheme.
Satisfiability problem of INL is \textit{PSPACE}-complete.

- Faithful embeddings $K \leftrightarrow \text{INL} \leftrightarrow K \oplus K$;
- Both $K$ and $K \oplus K$ are \textit{PSPACE}-complete.
Proof Theory
General idea of semantic tableau

- In order to prove $\phi$, 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 $\alpha$ and $\neg \alpha$’;
  - “Open” tableaus provide hints to counter-models (of $\phi$);
  - “Closed” tableaus are defined as proofs (of $\phi$).

Rules for classical propositional logic

$\neg \neg \phi \quad \alpha \land \beta \quad \neg (\alpha \lor \beta) \quad \neg (\alpha \rightarrow \beta) \quad \neg (\alpha \land \beta) \quad \alpha \lor \beta \quad \alpha \rightarrow \beta$

$\phi \quad \alpha \quad \neg \alpha \quad \beta \quad \neg \beta \quad \neg \alpha \quad \neg \beta \quad \alpha \quad \beta \quad \alpha \quad \beta$
General idea of semantic tableau
- In order to prove $\phi$, 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 $\alpha$ and $\neg \alpha$’;
  - “Open” tableaus provide hints to counter-models (of $\phi$);
  - “Closed” tableaus are defined as proofs (of $\phi$).

Rules for classical propositional logic

<table>
<thead>
<tr>
<th>$\neg \phi$</th>
<th>$\alpha \land \beta$</th>
<th>$\neg (\alpha \lor \beta)$</th>
<th>$\neg (\alpha \rightarrow \beta)$</th>
<th>$\neg (\alpha \land \beta)$</th>
<th>$\alpha \lor \beta$</th>
<th>$\alpha \rightarrow \beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\phi$</td>
<td>$\alpha$</td>
<td>$\neg \alpha$</td>
<td>$\alpha$</td>
<td>$\neg \alpha$</td>
<td>$\neg \beta$</td>
<td>$\neg \beta$</td>
</tr>
</tbody>
</table>
General idea of semantic tableau

- In order to prove $\phi$, 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 $\alpha$ and $\neg\alpha$’;
  - “Open” tableaus provide hints to counter-models (of $\phi$);
  - “Closed” tableaus are defined as proofs (of $\phi$).

Rules for classical propositional logic

$\neg\neg\phi \quad \alpha \land \beta \quad \neg(\alpha \lor \beta) \quad \neg(\alpha \rightarrow \beta) \quad \neg(\alpha \land \beta) \quad \alpha \lor \beta \quad \alpha \rightarrow \beta$

$\phi \quad \alpha \quad \neg\alpha \quad \beta \quad \neg\beta \quad \alpha \quad \neg\beta \quad \alpha \quad \neg\alpha \quad \beta \quad \neg\beta \quad \alpha \lor \beta \quad \neg\alpha \lor \neg\beta \quad \alpha \lor \beta \quad \neg\alpha \lor \neg\beta \quad \alpha \lor \beta \quad \neg\alpha \lor \neg\beta$
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):
    $\Box(\alpha_1, \ldots, \alpha_j; \alpha_0)$
    $\neg\Box(\beta^1_1, \ldots, \beta^1_{j_1}; \beta^1_0)$
    $\vdots$
    $\neg\Box(\beta^k_1, \ldots, \beta^k_{j_k}; \beta^k_0)$
INL needs (at least) a modal rule.

- A $\square$-formula requires a nbd (with certain properties);
  A $\neg \square$-formula refutes any nbd (with certain properties).
- $\square$’s do not work together to close a goal;
  they each does, together with all $\neg \square$’s in the same goal.

The rule takes from a goal:
- one $\square$-formula, and
- and any number of $\neg \square$-formulas
  (with variant numbers of instances):

$\square(\alpha_1, \ldots, \alpha_j; \alpha_0)$
$\neg \square(\beta_1^1, \ldots, \beta_{j_1}^1; \beta_0^1)$
  
  
  
  
  $\vdots$
$\neg \square(\beta_k^k, \ldots, \beta_{j_k}^k; \beta_0^k)$
INL needs (at least) a modal rule.

- A □-formula requires a nbd (with certain properties);
  A ¬□-formula refutes any nbd (with certain properties).

- □’s do not work together to close a goal;
  they each does, together with all ¬□’s in the same goal.

The rule takes from a goal:

- one □-formula, and
- and any number of ¬□-formulas

(with variant numbers of instances):

\[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \]
\[ \neg \square(\beta_{11}, \ldots, \beta_{1j}; \beta_{10}) \]
\[ \vdots \]
\[ \neg \square(\beta_{k1}, \ldots, \beta_{kj}; \beta_{k0}) \]
\[ \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \]
\[ \neg \Box(\beta_1^1, \ldots, \beta_{j_1}^1; \beta_0^1) \]
\[ \vdots \]
\[ \neg \Box(\beta_k^k, \ldots, \beta_{j_k}^k; \beta_0^k) \]

\[ \mid \alpha_0 \land \sigma \]
\[ \sigma \in \{\alpha_x\}_{x=1}^j \]

- \( \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \) **requires a nbd** with (generally) \( j \) states. Each nbd is consistent, if all its states are.

- \( \forall i \in \{1, \ldots, k\}, \neg \Box(\beta_i^1, \ldots, \beta_{j_i}^i; \beta_0^i) \) requires that either - \( \beta_0^i \) fails at some state, or - \( \beta_h^i \) fails at each state for some \( h \in \{1, \ldots, j_i\} \).

- \( \prod_{z=1}^k (j_z + 1) \) options in total.
  Index possible nbd's by the option it takes, e.g., \( \langle l(1), \ldots, l(k) \rangle \).
\[\Box (\alpha_1, \ldots, \alpha_j; \alpha_0)\]
\[\neg \Box (\beta^1_1, \ldots, \beta^1_{j_1}; \beta^1_0)\]
\[\vdots\]
\[\neg \Box (\beta^k_1, \ldots, \beta^k_{j_k}; \beta^k_0)\]

\begin{align*}
| \alpha_0 \land \sigma & \quad | \sigma \in \{\alpha_x\}^j_{x=1} \cup \{-\beta^i_0\} \quad | \\
\end{align*}

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

\[ \square(\alpha_1, ..., \alpha_j; \alpha_0) \]
\[ \neg \square(\beta_1^1, ..., \beta_j^1; \beta_0^1) \]
\[ \vdots \]
\[ \neg \square(\beta_k^k, ..., \beta_{j_k}^k; \beta_0^k) \]

\[ |\alpha_0 \land \sigma \land \neg \beta_i^h |_{\sigma \in \{\alpha_x\}_{x=1}^j \cup \{-\beta_0^i\}} \]

- **\( \square(\alpha_1, ..., \alpha_j; \alpha_0) \)** requires a nbd with (generally) \( j \) states. Each nbd is consistent, if all its states are.
- \( \forall i \in \{1, ..., k\}, \neg \square(\beta_1^i, ..., \beta_{j_i}^i; \beta_0^i) \)** requires that either - \( \beta_0^i \) fails at some state, or - \( \beta_h^i \) fails at each state for some \( h \in \{1, ..., j_i\} \).
- \( \prod_{z=1}^k (j_z + 1) \) options in total.

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

---

Junhua Yu

A Hyper-sequent Calculus for INL
Semantic tableau

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

\[ \alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}} \neg \beta^i_{l(i)} \big| \sigma \in \{\alpha_x\}^i_{x=1} \cup \{-\beta^y_{l(y)}\}^i_{y=1} \big| l \in \bigotimes_{z=1}^k \{0, \ldots, j_z\} \]

- \( \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \) requires a nbd with (generally) \( j \) states. Each nbd is consistent, if all its states are.

- \( \forall i \in \{1, \ldots, k\}, \neg \Box (\beta^i_1, \ldots, \beta^i_{j_i}; \beta^i_0) \) requires that either - \( \beta^0_0 \) fails at some state, or - \( \beta^i_h \) fails at each state for some \( h \in \{1, \ldots, j_i\} \).

- \( \prod_{z=1}^k (j_z + 1) \) options in total. Index possible nbd’s by the option it takes, e.g., \( \langle l(1), \ldots, l(k) \rangle \).
Semantic tableau

\[\Box(\alpha_1, \ldots, \alpha_j; \alpha_0)\]
\[-\Box (\beta^1_1, \ldots, \beta^1_j; \beta^1_0)\]
\[\vdots\]
\[-\Box (\beta^k_1, \ldots, \beta^k_j; \beta^k_0)\]

\(\alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}} \neg \beta^i_{l(i)} \big|_{\sigma \in \{\alpha_x\}^j_{x=1} \cup \{-\beta^y_0\}^j_{y=1} \cup \{\neg \beta^y_0\}^j_{y \in \{1, \ldots, k\}}} \big|_{l \in \bigotimes_{z=1}^k \{0, \ldots, j_z\}}\)

- 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.
A Hyper-sequent Calculus for INL

Semantic tableau

\[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \]
\[ \neg \square(\beta_1^1, \ldots, \beta_j^1; \beta_0^1) \]
\[ \vdots \]
\[ \neg \square(\beta_1^k, \ldots, \beta_j^k; \beta_0^k) \]

\[ | \alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}} \neg \beta_i^i \bigm|_{\sigma \in \{\alpha_x\}^j_x \cup \{-\beta_y^y\}^l_y \in \{1, \ldots, k\}} |_{l \in \bigotimes_{z=1}^k \{0, \ldots, j_z\}} \]

- 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.
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** \(\vdash \Box (\phi \lor \chi; \theta) \rightarrow \Box (\phi; \theta) \lor \Box (\chi; \theta)\)
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
Call the above mentioned tableau system \textit{TABinl}

- \textit{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
  - \textit{TABinl} offers a decision procedure
  - \textit{TABinl} indicates a way to some real proof-theory
    - a hyper sequent calculus
Call the above mentioned tableau system \textit{TABinl}

- \textit{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
- \textit{TABinl} offers a decision procedure
- \textit{TABinl} indicates a way to some real proof-theory
  - a hyper sequent calculus
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
  - \(\Gamma_1 \Rightarrow \Delta_1 | \ldots | \Gamma_n \Rightarrow \Delta_n\) - 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

Junhua Yu  A Hyper-sequent Calculus for INL
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
  - \([\Gamma_1 \Rightarrow \Delta_1] \ldots [\Gamma_n \Rightarrow \Delta_n]\) - 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, \to, \Box\} (classical)
- Multi-set-based, G3-style
  - No Exchange
  - Built-in Weakening and Contraction
  - Easier proofs of \(\text{Cut}\)-admissibility
- Hyper-sequent
  - \(|\Gamma_1 \Rightarrow \Delta_1| \ldots |\Gamma_n \Rightarrow \Delta_n|\) - 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

Junhua Yu
A Hyper-sequent Calculus for INL
Hyper-sequent calculus HSinl

• Primitive connectives: \{\bot, \rightarrow, \square\} (classical)

• Multi-set-based, G3-style
  • No Exchange
  • Built-in Weakening and Contraction
  • Easier proofs of \textit{(Cut)}-admissibility

• Hyper-sequent
  \[|\Gamma_1 \Rightarrow \Delta_1| \ldots |\Gamma_n \Rightarrow \Delta_n|\] - 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) \textit{(Weakening & Contraction)}
    no Exchange

• Intuitive correspondence
  • regular sequents \sim states
  • hyper-sequents \sim nbd’s
  - sufficient to prove (close) one sequent (state) in

Junhua Yu  A Hyper-sequent Calculus for INL
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
  - $|\Gamma_1 \Rightarrow \Delta_1| \ldots |\Gamma_n \Rightarrow \Delta_n|$ - finite multi-set of regular sequents
    - ‘standing for’ $\vee_{i=1}^{n}((\wedge \Gamma_i) \to (\vee \Delta_i))$
  - 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

- \(G|\Pi, p \Rightarrow p, \Sigma (Ax)\) and \(G|\Pi, \bot \Rightarrow \Sigma (L\bot)\)

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

- \(\frac{G|\Pi, \alpha \Rightarrow \beta, \Sigma}{G|\Pi \Rightarrow \alpha \rightarrow \beta, \Sigma} (R\rightarrow)\)

- \(\frac{G|\Pi \Rightarrow \alpha \rightarrow \beta, \Sigma}{[G|\Pi \Rightarrow \alpha, \Sigma][G|\Pi, \beta \Rightarrow \Sigma]} (L\rightarrow)\)

[...] stands for branches

- \(\alpha_0, \alpha_x \Rightarrow \{\beta_i^{I(i)_i} \}_{i \in \{1, \ldots, k\}}^{l(i)_i \neq 0} \mid x \in \{1, \ldots, j\}\)

- \(\frac{\alpha_0 \Rightarrow \beta_0^y, \{\beta_i^{l(i)_i} \}_{i \in \{1, \ldots, k\}}^{l(i)_i \neq 0} \mid y \in \{1, \ldots, k\}}{l \in \bigotimes_{i=1}^{k} \{0, 1, \ldots, j_i\}} (\square)\)

- \(G|\Pi, \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_i^{l(i)_i} \mid \beta_i^{l(i)_i} \}_{i \in \{1, \ldots, k\}}^{k} \mid i=1, \Sigma\)
Hyper-sequent calculus HSinl

- $G \mid \Pi, p \Rightarrow p, \Sigma (Ax)$ and $G \mid \Pi, \bot \Rightarrow \Sigma (L\bot)$

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

- $G \mid \Pi, \alpha \Rightarrow \beta, \Sigma (R \rightarrow)$

- $G \mid \Pi \Rightarrow \alpha \rightarrow \beta, \Sigma (L \rightarrow)$

  $[...]$ stands for branches

- $G \mid \Pi, \square (\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square (\beta^{i_1}_1, \ldots, \beta^{i_j}_{j_i})\}_{i=1}^{k}, \Sigma (\square)$
Hyper-sequent calculus HSinl

- \( G|\Pi, p \Rightarrow p, \Sigma (Ax) \) and \( G|\Pi, \bot \Rightarrow \Sigma (L\bot) \)
  
  - \( G \): meta-variable for sequent multi-sets (hyper-sequents)

- \( G|\Pi, \alpha \Rightarrow \beta, \Sigma (R\rightarrow) \)

- \( G|\Pi \Rightarrow \alpha \rightarrow \beta, \Sigma (L\rightarrow) \)

- \([G|\Pi \Rightarrow \alpha, \Sigma][G|\Pi, \beta \Rightarrow \Sigma]\) [... stands for branches]

\[
\begin{align*}
\alpha_0, \alpha_x & \Rightarrow \left\{ \beta^i_{l(i)} \right\}_{i \in \{1,\ldots,k\}}^l(i) \neq 0 \quad \left| \alpha_x \in \{1,\ldots,j\} \right. \\
\alpha_0 & \Rightarrow \beta^y_0, \left\{ \beta^i_{l(i)} \right\}_{i \in \{1,\ldots,k\}}^l(i) \neq 0 \quad \left| l(y) = 0 \right. \\
& \Rightarrow \quad \left| l(y) \in \bigotimes_{i=1}^k \{0,1,\ldots,j_i\} \right. \\
G|\Pi, \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) & \Rightarrow \left\{ \Box(\beta^i_1, \ldots, \beta^i_{j_i}) \right\}_{i=1}^k \cup \Sigma
\end{align*}
\]
Hyper-sequent calculus HSinl

- HSinl is sound.
  - If $\text{HSinl} \vdash |\Gamma_1 \Rightarrow \Delta_1|...|\Gamma_n \Rightarrow \Delta_n|$, then $\text{INL} \vdash \bigvee_{i=1}^{n} (\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the ($\Box$) rule, a sub-induction gives a stronger form of what we need.

- HSinl is complete.
  - If $\text{INL} \vdash \phi$, then $\text{HSinl} \vdash \Rightarrow \phi$.
  - $\{\phi | \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi | \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under $MP$.
    - A corollary of ($Cut$)-admissibility.
  - $\{\phi | \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under $RE$. 
Hyper-sequent calculus HSinl

- HSinl is sound.
  - If $\text{HSinl} \vdash |\Gamma_1 \Rightarrow \Delta_1| ... |\Gamma_n \Rightarrow \Delta_n|$, then $\text{INL} \vdash \bigvee_{i=1}^{n} (\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the ($\square$) rule, a sub-induction gives a stronger form of what we need.

- HSinl is complete.
  - If $\text{INL} \vdash \phi$, then $\text{HSinl} \vdash \Rightarrow \phi$.
  - $\{\phi | \text{HSinl} \vdash \Rightarrow \phi\}$ includes all axioms of INL.
  - $\{\phi | \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under $MP$.
    - A corollary of $(\text{Cut})$-admissibility.
  - $\{\phi | \text{HSinl} \vdash \Rightarrow \phi\}$ is closed under $RE$. 
Hyper-sequent calculus HSinl

- HSinl is sound.
  - If $\text{HSinl} \vdash \vdash_{\Gamma_1} \Rightarrow \Delta_1 | \ldots | \vdash_{\Gamma_n} \Rightarrow \Delta_n |$
    then $\text{INL} \vdash \bigvee_{i=1}^{n} (\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
  - Proved by an induction.
  - For the ($\Box$) rule, a sub-induction gives a stronger form of what we need.

- HSinl is complete.
  - If $\text{INL} \vdash \phi$, then $\text{HSinl} \vdash \Rightarrow \phi$.
    - $\{ \phi \mid \text{HSinl} \vdash \Rightarrow \phi \}$ 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 ⊢ |Γ₁ ⇒ Δ₁|...|Γₙ ⇒ Δₙ|, then INL ⊢ ∨ⁿᵢ₌₁ (∧ Γᵢ → ∨ Δᵢ).
  - Proved by an induction.
  - For the (□) rule, a sub-induction gives a stronger form of what we need.

- HSinl is complete.
  - If INL ⊢ φ, then HSinl ⊢ ⇒ φ.
  - {φ | HSinl ⊢ ⇒ φ} includes all axioms of INL.
    - A corollary of (Cut)-admissibility.
  - {φ | HSinl ⊢ ⇒ φ} is closed under MP
    - {φ | HSinl ⊢ ⇒ φ} is closed under RE.
HSinl is sound.

- If $\text{HSinl} \vdash \Gamma_1 \Rightarrow \Delta_1 | \ldots | \Gamma_n \Rightarrow \Delta_n$, then $\text{INL} \vdash \bigvee_{i=1}^n (\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$.
- Proved by an induction.
- For the ($\Box$) rule, a sub-induction gives a stronger form of what we need.

HSinl is complete.

- If $\text{INL} \vdash \phi$, then $\text{HSinl} \vdash \Rightarrow \phi$.
- $\{ \phi \mid \text{HSinl} \vdash \Rightarrow \phi \}$ includes all axioms of INL.
- $\{ \phi \mid \text{HSinl} \vdash \Rightarrow \phi \}$ is closed under MP
  - A corollary of ($\text{Cut}$)-admissibility.
- $\{ \phi \mid \text{HSinl} \vdash \Rightarrow \phi \}$ is closed under $\text{RE}$. 
HSinl is sound.
- If HSinl ⊢ |Γ₁ ⇒ Δ₁|...|Γₙ ⇒ Δₙ|,
  then INL ⊢ ∨ⁿᵢ₌₁ (∧ Γᵢ → ∨ Δᵢ).
- Proved by an induction.
- For the (□) rule, a sub-induction gives a stronger form of what we need.

HSinl is complete.
- If INL ⊢ φ, then HSinl ⊢ ⇒ φ.
- {φ | HSinl ⊢ ⇒ φ} includes all axioms of INL.
- {φ | HSinl ⊢ ⇒ φ} is closed under MP
  - A corollary of (Cut)-admissibility.
- {φ | HSinl ⊢ ⇒ φ} is closed under RE.
Hyper-sequent calculus HSinl

- **Admissibility of (Cut)** (no matter (Cut⁺) or (Cutₓ))
  - In HSinl resp. “HSinl ⊕ (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 α ⇒ α 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 (Cutₓ) (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 \(\alpha\), (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 \((\text{Cut})\) (no matter \((\text{Cut}_+)\) or \((\text{Cut}_\times)\))

- In \(\text{HSinl}\) resp. “\(\text{HSinl} \oplus (\text{Cut}_+)\) 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 \(\alpha\), (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 \(\text{HSinl}\), rules \((\text{Cut}_+)\) and \((\text{Cut}_\times)\) (at any same ‘degree’) are inter-derivable.

- Then, a standard double-induction works.

Subformula property of \(\text{HSinl}\) as a corollary.
Admissibility of \((\text{Cut})\) (no matter \((\text{Cut}_+)\) or \((\text{Cut}_\times)\))

In HSinl resp. “HSinl \(\oplus (\text{Cut}_+)\) 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 \(\alpha\), (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 \((\text{Cut}_+)\) and \((\text{Cut}_\times)\) (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. \(\text{“HSinl} \oplus (Cut_+)\) 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 \(\alpha\), (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:**
  
  \[ (\text{Let } \nu^+(\alpha)/\nu^-(\alpha) \text{ denotes positive/negative atoms in } \alpha) \]
  
  If \( \text{INL} \vdash \phi \rightarrow \psi \), then there is a formula \( \epsilon \) s.t.:
  
  - \( \nu^\pm(\epsilon) \subseteq \nu^\pm(\phi) \cap \nu^\pm(\psi) \)
  
  - \( \text{INL} \vdash \phi \rightarrow \epsilon \) and \( \text{INL} \vdash \epsilon \rightarrow \psi \).

  (a ‘polar generalization’ of Craig interpolation)

- **A general form:**
  
  If \( \text{HSinl} \vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R \), then there is a formula \( \epsilon \) s.t.:
  
  - \( \nu^\pm(\epsilon) \subseteq (\nu^\pm(\Pi_R, \Sigma_L)) \cap (\nu^\pm(\Pi_L, \Sigma_R)) \)
  
  - \( \text{HSinl} \vdash \Pi_L \Rightarrow \Sigma_L, \epsilon \) and \( \text{HSinl} \vdash \epsilon, \Pi_R \Rightarrow \Sigma_R \).

- **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 $\nu^+(\alpha) / \nu^-(\alpha)$ denotes positive/negative atoms in $\alpha$)

If $\text{INL} \vdash \phi \rightarrow \psi$, then there is a formula $\epsilon$ s.t.:

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

(a ‘polar generalization’ of Craig interpolation)

A general form:

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

- $\nu^\pm(\epsilon) \subseteq (\nu^+(\Pi_R, \Sigma_L)) \cap (\nu^+(\Pi_L, \Sigma_R))$
- $\text{HSinl} \vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and $\text{HSinl} \vdash \epsilon, \Pi_R \Rightarrow \Sigma_R$.

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:
  
  \[ \text{(Let } \mathcal{V}^{+}(\alpha)/\mathcal{V}^{-}(\alpha) \text{ denotes positive/negative atoms in } \alpha) \]

  If INL \( \vdash \phi \rightarrow \psi \), then there is a formula \( \epsilon \) 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 \( \epsilon \) s.t.:

  - \( \mathcal{V}^{\pm}(\epsilon) \subseteq (\mathcal{V}^{\pm}(\Pi_R, \Sigma_L)) \cap (\mathcal{V}^{\pm}(\Pi_L, \Sigma_R)) \)
  - 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 theorem:

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

If $\text{INL} \vdash \phi \rightarrow \psi$, then there is a formula $\epsilon$ s.t.:

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

(a ‘polar generalization’ of Craig interpolation)

A general form:

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

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

Employ a ‘splitting version’ of $\text{HSinl}$

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

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

  (a ‘polar generalization’ of Craig interpolation)

- **A general form:**

  If $\text{HSinl} \vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula $\epsilon$ s.t.

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

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

(a ‘polar generalization’ of Craig interpolation)

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

- 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 $\nu^+ (\alpha)/\nu^- (\alpha)$ denotes positive/negative atoms in $\alpha$)

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

  (a ‘polar generalization’ of Craig interpolation)

- **A general form:**
  
  If $\text{HSinl} \vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula $\epsilon$ s.t.:
  
  - $\nu^\pm (\epsilon) \subseteq (\nu^\mp (\Pi_R, \Sigma_L)) \cap (\nu^\pm (\Pi_L, \Sigma_R))$
  - $\text{HSinl} \vdash \Pi_L \Rightarrow \Sigma_L, \epsilon$ and $\text{HSinl} \vdash \epsilon, \Pi_R \Rightarrow \Sigma_R$.

- **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!