INL (instantial neighborhood logic)
- tableau, sequent calculus, interpolation

Junhua Yu
(junhua.yu.5036@outlook.com)

Tsinghua University (Beijing, China)

2017.10.18 @ Steklov Mathematical Institute
Outline

- Instantial neighborhood logic INL
- Semantic tableau system \textit{TABinl} (presented at TSU in 2016)
- Sequent calculus G3inl
- Lyndon interpolation theorem
- Future directions
Abbreviation: “nbd” means “neighborhood”
Instantial neighborhood logic INL
Nbd structures

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

- Possible properties of nbd functions
  - nbd’s are non-empty,
  - nbd’s of $w$ always contain $w$,
  - each point has exactly 1 nbd
    - degenerates to relational semantics,
  - closure properties...

None of above assumed for the current work.

- **Model**: $M = (\mathcal{F}, V)$
  - $\mathcal{F}$, a nbd frame;
  - $V : W \rightarrow 2^P$, a propositional valuation.
Nbd structures

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

- Possible properties of nbd functions
  - nbd’s are non-empty,
  - nbd’s of $w$ always contain $w$,
  - each point has exactly 1 nbd
    - degenerates to relational semantics,
  - closure properties...

None of above assumed for the current work.

- Model: $\mathcal{M} = (\mathcal{F}, V)$
  - $\mathcal{F}$, a nbd frame;
  - $V : W \mapsto 2^P$, a propositional valuation.
Nbd structures

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

- Possible properties of nbd functions
  - nbd’s are non-empty,
  - nbd’s of $w$ always contain $w$,
  - each point has exactly 1 nbd
    - degenerates to relational semantics,
  - closure properties...

None of above assumed for the current work.

- Model: $\mathcal{M} = (\mathfrak{F}, V)$
  - $\mathfrak{F}$, a nbd frame;
  - $V : W \mapsto 2^P$, a propositional valuation.
Nbd logic

- **Basic modal language:** unary operator $\square$.
- **Truth definition - a $\exists \forall$ reading of $\square$:**
  - $\mathcal{M}, w \models \exists \forall \alpha$ iff $(\exists N \in \mathcal{N}(w))(\forall n \in N) \mathcal{M}, n \models \alpha$.
  - A neighborhood (of the current point) has $\alpha$ true everywhere inside.
- B.t.w., alternative truth definition exist.
- Some schemes as examples:
  - **Invalid**
    - $\square \alpha \land \square \beta \rightarrow \square (\alpha \land \beta)$
    - $\models \phi$
    - $\models \square \phi$
  - **Valid**
    - $\square (\alpha \land \beta) \rightarrow \square \alpha \land \square \beta$
    - $\square \bot \rightarrow \square \alpha$
Nbd logic

- **Basic modal language**: unary operator □.
- **Truth definition** - a ∃∀ reading of □:
  - \( M, w \models □\alpha \) iff \( (\exists N \in \sigma(w))(\forall n \in N) M, n \models \alpha. \)
  - A neighborhood (of the current point) has \( \alpha \) true everywhere inside.
  - B.t.w., alternative truth definition exist.
- Some schemes as examples:

<table>
<thead>
<tr>
<th>Invalid</th>
<th>Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>□\alpha \land □\beta \rightarrow □(\alpha \land \beta)</td>
<td>□(\alpha \land \beta) \rightarrow □\alpha \land □\beta</td>
</tr>
<tr>
<td>\models \phi</td>
<td>□\bot \rightarrow □\alpha</td>
</tr>
<tr>
<td>□\phi</td>
<td>\models □\phi</td>
</tr>
</tbody>
</table>
Nbd logic

- Basic modal language: unary operator □.
- Truth definition - a ∃∀ reading of □:
  - \( M, w \models □\alpha \) iff
    - (\( ∃N ∈ \sigma(w) \)) (\( ∀n ∈ N \) M, n \( \models α \)).
  - a neighborhood (of the current point) has \( α \) true everywhere inside.
  - B.t.w., alternative truth definition exist.

- Some schemes as examples:

  invalid
  \[ □\alpha ∧ □\beta → □(\alpha ∧ \beta) \]
  \[ \models \phi \]
  \[ \models □\phi \]

  valid
  \[ □(\alpha ∧ \beta) → □\alpha ∧ □\beta \]
  \[ □⊥ → □\alpha \]
Basic modal language: unary operator □.

Truth definition - a ∃∀ reading of □:

- \( \mathcal{M}, w \models \square \alpha \)
- iff
- \((\exists N \in \sigma(w))(\forall n \in N) \mathcal{M}, n \models \alpha.\)
- a neighborhood (of the current point) has \( \alpha \) true everywhere inside.
- B.t.w., alternative truth definition exist.

Some schemes as examples:

- invalid:
  - \( \square \alpha \land \square \beta \rightarrow \square (\alpha \land \beta) \)
  - \( \models \phi \)
  - \( \models \square \phi \)

- valid:
  - \( \square (\alpha \land \beta) \rightarrow \square \alpha \land \square \beta \)
  - \( \square \bot \rightarrow \square \alpha \)
Describe nbd structures using an “instantial” language:

- 2-sorted, \((j; 1)\)-ary (where \(j \in \mathbb{N}\)) operator:
  \[ \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \]

- Subformulas \(\alpha_1, \ldots, \alpha_j\) are called instances.

Truth definition - a “\(\exists(\exists; \forall)\)” reading of \(\square\):

\[ M, w \models \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \text{ iff } \]

\[
\begin{cases} 
(\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{cases}
\)

- a neighborhood (of the current point) in which
  - \(\alpha_0\) holds everywhere, and
  - each instance (respectively) holds somewhere.
van Bentham et.al. 2017: *RSL* 10(1).

**Describe nbd structures using an “instantial” language:**

- 2-sorted, \((j; 1)\)-ary (where \(j \in \mathbb{N}\)) operator:
  \[
  \Box(\alpha_1, \ldots, \alpha_j; \alpha_0)
  \]

- Subformulas \(\alpha_1, \ldots, \alpha_j\) are called **instances**.

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

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

- **a neighborhood** (of the current point) in which
  - \(\alpha_0\) holds everywhere, and
  - each instance (respectively) holds somewhere.

Junhua Yu (junhua.yu.5036@outlook.com)  
INL - tableau, sequent calculus, interpolation
van Benthem et.al. 2017: RSL 10(1).

Describe nbd structures using an “instantial” language:

- 2-sorted, \((j; 1)\)-ary (where \(j \in \mathbb{N}\)) operator:

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

- Subformulas \(\alpha_1, \ldots, \alpha_j\) are called instances.

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

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

- a neighborhood (of the current point) in which
  - \(\alpha_0\) holds everywhere, and
  - each instance (respectively) holds somewhere.
Some **invalid** schemes:

- \( \neg \Box (\alpha; \bot) \)  
  (cf. \( \neg \Box \bot \) in nbd logic)
  - but, \( \neg \Box (\alpha; \bot) \) is valid
  - \( \models \phi \)
  - and also \( \Box (\alpha) \land \Box (\beta) \rightarrow \Box (\alpha \land \beta) \)
  - \( \Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi) \)

Some **valid** schemes:

- \( \Box (\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, ..., \alpha_j; \alpha_0 \lor \eta) \)
- \( \Box (\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, ..., \alpha_j, \phi \lor \psi; \alpha_0) \)
- \( \Box (\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box (\alpha_2, ..., \alpha_j; \alpha_0) \)
- \( \Box (\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0) \)  
  where \( i \in \{1, ..., j\} \)
- \( \Box (\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box (\alpha_1, ..., \alpha_j, \eta \land \alpha_0; \alpha_0) \)
- \( \neg \Box (\alpha_1, ..., \alpha_j; \bot; \alpha_0) \)
- \( \Box (\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box (\alpha_1, ..., \alpha_j, \delta; \alpha_0) \lor \Box (\alpha_1, ..., \alpha_j; \alpha_0 \land \neg \delta)) \)

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some **invalid** schemes:

- $\neg \Box (\; \bot)$  
  (cf. $\neg \Box \bot$ in nbd logic)
- but, $\neg \Box (\alpha; \bot)$ is valid

- $\models \phi$ and also $\Box (\; \alpha) \land \Box (\; \beta) \rightarrow \Box (\; \alpha \land \beta)$
- $\Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi)$

Some **valid** schemes:

- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta)$
- $\Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0)$
- $\Box (\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_2, \ldots, \alpha_j; \alpha_0)$
- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_i; \alpha_0)$ where $i \in \{1, \ldots, j\}$
- $\Box (\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0)$
- $\neg \Box (\alpha_1, \ldots, \alpha_j, \bot; \alpha_0)$
- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\Box (\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta))$

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some invalid schemes:

- $\neg \Box (\alpha; \bot)$ (cf. $\neg \Box \bot$ in nbd logic)
- but, $\neg \Box (\alpha; \bot)$ is valid

$$\vdash \phi$$
and also $\Box (\alpha) \land \Box (\beta) \rightarrow \Box (\alpha \land \beta)$

$$\Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi)$$

Some valid schemes:

- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta)$
- $\Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0)$
- $\Box (\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_2, \ldots, \alpha_j; \alpha_0)$
- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_i; \alpha_0)$ where $i \in \{1, \ldots, j\}$
- $\Box (\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0)$
- $\neg \Box (\alpha_1, \ldots, \alpha_j; \bot; \alpha_0)$
- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\Box (\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \neg \delta))$

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some invalid schemes:

- $\neg \Box(\bot)$  
  (cf. $\neg \Box \bot$ in nbd logic)
- but, $\neg \Box(\alpha; \bot)$ is valid
- $\models \phi$  
  and also $\Box(\alpha) \land \Box(\beta) \rightarrow \Box(\alpha \land \beta)$
- $\Box(\alpha; \psi) \land \Box(\beta; \psi) \rightarrow \Box(\alpha, \beta; \psi)$

Some valid schemes:

- $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \lor \eta)$
- $\Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \lor \psi; \alpha_0)$
- $\Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0)$
- $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0)$  
  where $i \in \{1, ..., j\}$
- $\Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \land \alpha_0; \alpha_0)$
- $\neg \Box(\alpha_1, ..., \alpha_j; \bot; \alpha_0)$
- $\Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \lor \Box(\alpha_1, ..., \alpha_j; \alpha_0 \land \neg \delta))$

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some invalid schemes:

- \( \neg \Box (\perp ; \perp) \) (cf. \( \neg \Box \perp \) in nbd logic)
  - but, \( \neg \Box (\alpha; \perp) \) is valid
  - \( \models \phi \) and also \( \Box (\alpha ; \phi) \land \Box (\beta ; \phi) \rightarrow \Box (\alpha \land \beta) \)
  - \( \Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi) \)

Some valid schemes:

- \( \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \)
- \( \Box (\alpha_1, \ldots, \alpha_j, \phi; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \phi \lor \psi; \alpha_0) \)
- \( \Box (\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_2, \ldots, \alpha_j; \alpha_0) \)
- \( \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0) \) where \( i \in \{1, \ldots, j\} \)
- \( \Box (\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \)
- \( \neg \Box (\alpha_1, \ldots, \alpha_j, \perp; \alpha_0) \)
- \( \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\Box (\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \)

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some invalid schemes:

\[ \neg \square (; \bot) \quad \text{(cf. } \neg \square \bot \text{ in nbd logic)} \]

but, \( \neg \square (\alpha; \bot) \) is valid

\[ \models \phi \quad \text{and also } \square (; \alpha) \land \square (; \beta) \rightarrow \square (; \alpha \land \beta) \]

\[ \square (\alpha; \psi) \land \square (\beta; \psi) \rightarrow \square (\alpha, \beta; \psi) \]

Some valid schemes:

\[ \square (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta) \]

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

\[ \square (\alpha_1, \alpha_2, \ldots, \alpha_j; \alpha_0) \rightarrow \square (\alpha_2, \ldots, \alpha_j; \alpha_0) \]

\[ \square (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \square (\alpha_1, \ldots, \alpha_j, \alpha_j; \alpha_0) \quad \text{where } i \in \{1, \ldots, j\} \]

\[ \square (\alpha_1, \ldots, \alpha_j, \eta; \alpha_0) \rightarrow \square (\alpha_1, \ldots, \alpha_j, \eta \land \alpha_0; \alpha_0) \]

\[ \neg \square (\alpha_1, \ldots, \alpha_j, \bot; \alpha_0) \]

\[ \square (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow (\square (\alpha_1, \ldots, \alpha_j, \delta; \alpha_0) \lor \square (\alpha_1, \ldots, \alpha_j; \alpha_0 \land \neg \delta)) \]

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some **invalid** schemes:

- $\neg \Box (\bot)$ (cf. $\neg \Box \bot$ in nbd logic)
  - but, $\neg \Box (\alpha; \bot)$ is valid

- $\Box \phi$ and also $\Box (\alpha) \land \Box (\beta) \rightarrow \Box (\alpha \land \beta)$

- $\Box (\alpha; \psi) \land \Box (\beta; \psi) \rightarrow \Box (\alpha, \beta; \psi)$

Some **valid** schemes:

- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j; \alpha_0 \lor \eta)$

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

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

- $\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \rightarrow \Box (\alpha_1, \ldots, \alpha_j, \alpha_i; \alpha_0)$ where $i \in \{1, \ldots, j\}$

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

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

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

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Some invalid schemes:

- \( \neg \Box(\alpha; \bot) \) (cf. \( \neg \Box \bot \) in nbd logic)
- but, \( \neg \Box(\alpha; \bot) \) is valid

\[ \models \phi \]
\[ \models \Box(\phi) \]

and also \( \Box(\alpha) \land \Box(\beta) \rightarrow \Box(\alpha \land \beta) \)

\[ \Box(\alpha; \psi) \land \Box(\beta; \psi) \rightarrow \Box(\alpha, \beta; \psi) \]

Some valid schemes:

- \( \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j; \alpha_0 \lor \eta) \)
- \( \Box(\alpha_1, ..., \alpha_j, \phi; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \phi \lor \psi; \alpha_0) \)
- \( \Box(\alpha_1, \alpha_2, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_2, ..., \alpha_j; \alpha_0) \)
- \( \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \alpha_i; \alpha_0) \) where \( i \in \{1, ..., j\} \)
- \( \Box(\alpha_1, ..., \alpha_j, \eta; \alpha_0) \rightarrow \Box(\alpha_1, ..., \alpha_j, \eta \land \alpha_0; \alpha_0) \)
- \( \neg \Box(\alpha_1, ..., \alpha_j, \bot; \alpha_0) \)
- \( \Box(\alpha_1, ..., \alpha_j; \alpha_0) \rightarrow (\Box(\alpha_1, ..., \alpha_j, \delta; \alpha_0) \lor \Box(\alpha_1, ..., \alpha_j; \alpha_0 \land \neg \delta)) \)

Together with propositional logic and replacement of equivalence, these constitutes Hinl, a sound-and-complete axiomatization of INL.
Semantic tableau system TABinl
General idea: satisfiability reduction

- Start with the main goal of satisfying the negation;
- Reduce goals to subgoals (by rules), extend potential model upon request;
- Impossible goals are “closed”, otherwise “open” (hints of models).

Rules for classical propositional logic

||...|| means branching

\[
\begin{array}{cccccccc}
\neg\neg\phi & \alpha \land \beta & \neg(\alpha \lor \beta) & \neg(\alpha \rightarrow \beta) & \neg(\alpha \land \beta) & \alpha \lor \beta & \alpha \rightarrow \beta \\
\phi & \alpha & \neg\alpha & \alpha & \neg\alpha \lor \neg\beta & \neg\alpha \land \neg\beta & \neg\alpha \lor \beta \\
\beta & \neg\beta & \neg\beta & \neg\beta & \\
\end{array}
\]

Still missing for INL are rule(s) for nbd operator □.
General idea: satisfiability reduction

- Start with the main goal of satisfying the negation;
- Reduce goals to subgoals (by rules),
- extend potential model upon request;
- Impossible goals are “closed”,
  otherwise “open” (hints of models).

Rules for classical propositional logic

||...|| means branching

\[
\begin{array}{cccc}
\neg\neg\phi & \alpha \land \beta & \neg(\alpha \lor \beta) & \neg(\alpha \to \beta) \\
\phi & \alpha & \neg\alpha & \alpha \\
\beta & \neg\beta & \neg\beta &
\end{array}
\]

Still missing for INL are rule(s) for nbd operator $\Box$. 
General idea: satisfiability reduction
- Start with the main goal of satisfying the negation;
- Reduce goals to subgoals (by rules),
  extend potential model upon request;
- Impossible goals are “closed”,
  otherwise “open” (hints of models).

Rules for classical propositional logic
||...|| means branching

Still missing for INL are rule(s) for nbd operator □.
General idea: satisfiability reduction
- Start with the main goal of satisfying the negation;
- Reduce goals to subgoals (by rules), extend potential model upon request;
- Impossible goals are “closed”, otherwise “open” (hints of models).

Rules for classical propositional logic
\[
\begin{align*}
\neg\neg\phi & \quad \alpha \land \beta & \quad \neg(\alpha \lor \beta) & \quad \neg(\alpha \rightarrow \beta) & \quad \neg(\alpha \land \beta) & \quad \alpha \lor \beta & \quad \alpha \rightarrow \beta \\
\phi & \quad \alpha & \quad \neg\alpha & \quad \alpha & \quad \vert\neg\alpha\vert \neg\beta & \quad \vert\alpha\vert \beta & \quad \vert\neg\alpha\vert \beta
\end{align*}
\]

Still missing for INL are rule(s) for nbd operator □.
Semantic tableau system TABinl

- General idea: satisfiability reduction
  - Start with the main goal of satisfying the negation;
  - Reduce goals to subgoals (by rules), extend potential model upon request;
  - Impossible goals are “closed”, otherwise “open” (hints of models).

- Rules for classical propositional logic
  - \( \lnot \lnot \phi \quad \alpha \land \beta \quad \lnot (\alpha \lor \beta) \quad \lnot (\alpha \rightarrow \beta) \quad \lnot (\alpha \land \beta) \quad \alpha \lor \beta \quad \alpha \rightarrow \beta \)

- Still missing for INL are rule(s) for nbd operator \( \square \).
In order to be satisfied:

- A $\Box$-formula requires a nbd (of certain type);
- A $\neg\Box$-formula refutes any nbd (of certain type);
- Other formulas govern locally.

- $\Box$-formulas do not work together to close a goal; they each does, together with all $\neg\Box$-formulas in the same goal.

The rule takes from a goal:

- one $\Box$-formula (with $j$-many instances), and
- any finite number (say $k$) of $\neg\Box$-formulas (resp. with $j_1, \ldots, j_k$-many instances).

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

In order to be satisfied:
- A $\Box$-formula requires a nbd (of certain type);
  A $\neg\Box$-formula refutes any nbd (of certain type);
  Other formulas govern locally.
- $\Box$-formulas do not work together to close a goal; they each does,
  together with all $\neg\Box$-formulas in the same goal.

The rule takes from a goal:
- one $\Box$-formula (with $j$-many instances), and
- any finite number (say $k$) of $\neg\Box$-formulas
  (resp. with $j_1, \ldots, j_k$-many instances).

\[
\begin{align*}
\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \\
\neg\Box(\beta^1_1, \ldots, \beta^1_{j_1}; \beta^1_0) \\
\vdots \\
\neg\Box(\beta^k_1, \ldots, \beta^k_{j_k}; \beta^k_0)
\end{align*}
\]
In order to be satisfied:

- A $\Box$-formula requires a nbd (of certain type);
- A $\neg\Box$-formula refutes any nbd (of certain type);
- Other formulas govern locally.

$\Box$-formulas do not work together to close a goal; they each does, together with all $\neg\Box$-formulas in the same goal.

The rule takes from a goal:

- one $\Box$-formula (with $j$-many instances), and
- any finite number (say $k$) of $\neg\Box$-formulas (resp. with $j_1, \ldots, j_k$-many instances).

\[
\begin{align*}
\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \\
\neg\Box(\beta^1_1, \ldots, \beta^1_{j_1}; \beta^1_0) \\
\vdots \\
\neg\Box(\beta^k_1, \ldots, \beta^k_{j_k}; \beta^k_0)
\end{align*}
\]
| \[\Box(\alpha_1, ..., \alpha_j; \alpha_0)\] \text{ requires a nbd with (generally) } j \text{ points.} \\
| A \text{ nbd is consistent, if all its points are.} \\
| \forall i \in \{1, ..., k\}, \neg \Box(\beta_i^1, ..., \beta_i^{j_i}; \beta_i^0) \text{ requires} \\
| \text{ either } - \beta_i^0 \text{ fails at some point,} \\
| \text{ or } - \beta_i^0 \text{ fails at each point for some } h \in \{1, ..., j_i\}. \\
| j_i + 1 \text{ options for each } i, \text{ so } \prod_{z=1}^{k} (j_z + 1) \text{ options in total.} \\
| \text{ Index possible nbd's by the option it takes, e.g., } l = \langle I(1), ..., I(k) \rangle.
Semantic tableau system TABinl

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

\[ | \alpha_0 \land \sigma \quad | \sigma \in \{ \alpha_x \}^j_x=1 \cup \{ \neg \beta^i_0 \} \]

- \( \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \) requires a nbd with (generally) \( j \) points. A nbd is consistent, if all its points are.
- \( \forall i \in \{ 1, \ldots, k \}, \neg \Box(\beta^i_1, \ldots, \beta^i_{j_i}; \beta^i_0) \) requires either - \( \beta^i_0 \) fails at some point, or - \( \beta^i_h \) fails at each point for some \( h \in \{ 1, \ldots, j_i \} \).
- \( j_i + 1 \) options for each \( i \), so \( \prod_{z=1}^k (j_z + 1) \) options in total.

Index possible nbd’s by the option it takes, e.g., \( I = \langle I(1), \ldots, I(k) \rangle \).
\(|\Box (\alpha_1, \ldots, \alpha_j; \alpha_0)\) requires a nbd with (generally) \(j\) points.

A nbd is consistent, if all its points are.

\(|\forall i \in \{1, \ldots, k\}, \neg \Box (\beta_i^1, \ldots, \beta_{ji}^i; \beta_0^i)\) requires that

either - \(\beta_0^i\) fails at some point,
or - \(\beta_h^i\) fails at each point for some \(h \in \{1, \ldots, j_i\}\).

\(|j_i + 1\) options for each \(i\), so \(\prod_{z=1}^k (j_z + 1)\) options in total.

Index possible nbd's by the option it takes, e.g., \(|I = \langle I(1), \ldots, I(k)\rangle|\).
\[ \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \]
\[ \neg \Box (\beta_1^1, \ldots, \beta_{j_i}^1; \beta_0^1) \]
\[ \vdots \]
\[ \neg \Box (\beta_k^k, \ldots, \beta_{j_k}^k; \beta_0^k) \]

\[ \big| \alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}} \neg \beta_i^{l(i) \neq 0} \big| \sigma \in \{\alpha_x\}_{x=1}^j \cup \{-\beta_0^y\}_{y \in \{1, \ldots, k\}} \big| l \in \bigotimes_{z=1}^k \{0, \ldots, j_z\} \]

- \( \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \) requires a nbd with (generally) \( j \) points. A nbd is consistent, if all its points are.
- \( \forall i \in \{1, \ldots, k\}, \neg \Box (\beta_1^i, \ldots, \beta_{j_i}^i; \beta_0^i) \) requires that either - \( \beta_0^i \) fails at some point - “\( l(i) = 0 \)” or - \( \beta_h^i \) fails at each point for some \( h \in \{1, \ldots, j_i\} \) - “\( l(i) \neq 0 \)”.
- \( j_i + 1 \) options for each \( i \), so \( \prod_{z=1}^k (j_z + 1) \) options in total.

Index possible nbd’s by the option it takes, e.g., \( l = \langle l(1), \ldots, l(k) \rangle \).
\[ \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \]
\[ \neg \Box (\beta^1_1, \ldots, \beta^1_j; \beta^1_0) \]
\[ \vdots \]
\[ \neg \Box (\beta^k_1, \ldots, \beta^k_j; \beta^k_0) \]
\[ | \alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}} \neg \beta^i_{l(i)} | \sigma \in \{ \alpha_x \}^j_{x=1} \cup \{ \neg \beta^y_0 \}^j_{y=1} \cup \{ \neg \beta^y_0 \}^j_{y=1} | l \in \bigotimes_{z=1}^{k} \{0, \ldots, j_z\} \]

- Each of these options a possible nbd to satisfy the goal,
- In order to close, each option has to be closed.
- Each possible nbd is a collection of points (subgoals),
  To close a nbd, it is enough to close one subgoal in it.
- It is destructive,
  No formulas (used or not) above the line are effective below the line.
- It degenerates to special cases where certain parameters are 0.
Each of these options a possible nbd to satisfy the goal,

In order to close, each option has to be closed.

Each possible nbd is a collection of points (subgoals),

To close a nbd, it is enough to close one subgoal in it.

It is destructive,

No formulas (used or not) above the line are effective below the line.

It degenerates to special cases where certain parameters are 0.
Each of these options a possible nbd to satisfy the goal.

In order to close, each option has to be closed.

Each possible nbd is a collection of points (subgoals),

To close a nbd, it is enough to close one subgoal in it.

It is destructive,

No formulas (used or not) above the line are effective below the line.

It degenerates to special cases where certain parameters are 0.
Each of these options a possible nbd to satisfy the goal,

In order to close, each option has to be closed.

Each possible nbd is a collection of points (subgoals),

To close a nbd, it is enough to close one subgoal in it.

It is destructive,

No formulas (used or not) above the line are effective below the line.

It degenerates to special cases where certain parameters are 0.
TABinl is sound and complete.

TABinl offers to INL a mechanical procedure of proof/counter-model search.

B.t.w., known from van Benthem et.al. 2017: RSL 10(1), INL is PSPACE-complete.

A manuscript introducing TABinl will be available in the proceeding of LFCS 2018.
TABinl is sound and complete.

TABinl offers to INL a mechanical procedure of proof/counter-model search.

B.t.w., known from van Benthem et.al. 2017: *RSL* 10(1), INL is PSPACE-complete.

A manuscript introducing TABinl will be available in the proceeding of LFCS 2018.
TABinl is sound and complete.

TABinl offers to INL a mechanical procedure of proof/counter-model search.

B.t.w., known from van Benthem et.al. 2017: *RSL* 10(1), INL is PSPACE-complete.

A manuscript introducing TABinl will be available in the proceeding of LFCS 2018.
Sequent calculus G3inl
A duality

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

\[ | \alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}} I(i) \neq 0 | \neg \beta_i^{\sigma} \big| I(i) \in \{\alpha_x\}_{x = 1} \cup \{\neg \beta_0^y\}_{y \in \{1, \ldots, k\}} \big| l \in \otimes_{z = 1}^{k} \{0, \ldots, j_z\} \]

- Sequent/tableau proofs are usually dual of each other.
- Sequents \( \leftrightarrow \) branches where closure is tested.
- (Left/right) sides to “\( \Rightarrow \)” \( \leftrightarrow \) (positive/negative) signs of formulas.
- In the nbd rule of TABinl:
  - a goal is reduced to groups (instead of a group) of subgoals,
  - it is sufficient to close one subgoal in each group.
A duality

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

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

- Sequent/tableau proofs are usually dual of each other.
  - Sequents \(\leftrightarrow\) branches where closure is tested.
  - (Left/right) sides to “\(\Rightarrow\)” \(\leftrightarrow\) (positive/negative) signs of formulas.
- In the nbd rule of TABinl:
  - a goal is reduced to groups (instead of a group) of subgoals,
  - it is sufficient to close one subgoal in each group.
Sequent/tableau proofs are usually dual of each other.

- Sequents $\leftrightarrow$ branches where closure is tested.
- (Left/right) sides to “$\Rightarrow$” $\leftrightarrow$ (positive/negative) signs of formulas.

In the nbd rule of TABinl:

- a goal is reduced to groups (instead of a group) of subgoals,
- it is sufficient to close one subgoal in each group.
A duality

\[
\begin{align*}
\Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \\
\neg \Box (\beta_1^1, \ldots, \beta_j^1; \beta_0^1) \\
\vdots \\
\neg \Box (\beta_k^1, \ldots, \beta_j^k; \beta_0^k)
\end{align*}
\]

\[
\begin{array}{c}
| \alpha_0 \land \sigma \land \bigwedge_{i \in \{1, \ldots, k\}}^{l(i) \neq 0} \neg \beta_i^l(i) \\
| \sigma \in \{\alpha_x\}_{x=1}^j \cup \{-\beta_0^y\}_{y \in \{1, \ldots, k\}}^{l(y)=0}
\end{array}
\]

\[
l \in \bigotimes_{z=1}^k \{0, \ldots, j_z\}
\]

- The dual is a **hyper-sequent rule**.
  - Hyper-sequents := finite multi-sets of (regular) sequents;
  - like \( |\Gamma_1 \Rightarrow \Delta_1| \ldots |\Gamma_n \Rightarrow \Delta_n| \),
    where “|” is read disjunctively.
  - In order to prove a hyper-seq.,
    it is **sufficient to prove one** of its sequents.
The hyper-sequent rule

\[ \square(\alpha_1, ..., \alpha_j; \alpha_0) \]

\[ \neg \square(\beta_1^1, ..., \beta_{j_1}^1; \beta_0^1) \]

\[ \vdots \]

\[ \neg \square(\beta_k^1, ..., \beta_{j_k}^1; \beta_0^k) \]

\[ |\alpha_0 \land \sigma \land \bigwedge_{i \in \{1, ..., k\}} \neg \beta_i^{l(i)} | \sigma \in \{\alpha_x\}_{x=1}^j \cup \{\neg \beta_0^y\}_{y=1}^k \]  

\[ l \in \bigotimes_{i=1}^k \{0, ..., j_i\} \]

\[ \begin{align*}
\alpha_0, \alpha_{-x} & \Rightarrow \{\beta_i^{l(i)}\}_{i \in \{1, ..., k\}} | x \in \{-j, ..., -1\} \\
\alpha_0 & \Rightarrow \beta_0^y, \{\beta_i^{l(i)}\}_{i \in \{1, ..., k\}} | y \in \{1, ..., k\} \\
\square(\alpha_1, ..., \alpha_j; \alpha_0) & \Rightarrow \{\square(\beta_1^i, ..., \beta_{j_i}^i; \beta_0^i)\}_{i=1}^k
\end{align*} \]
The hyper-sequent rule

For a better appearance, let

\[
\begin{align*}
J &= \{-j, \ldots, -1\} \\
K &= \{1, \ldots, k\} \\
K(I) &= \{y \in K \mid I(y) = 0\} \\
\Omega_K^l &= \{\beta_i^l(i)\}_{i \in K}^{l(i) \neq 0} \\
\Omega_D\left(\bigotimes_{i \in K}\{0, 1, \ldots, j_i\}\right) &= \{\beta_i^l(i)\}_{i \in K}^{l(i) \neq 0}
\end{align*}
\]

\[
\begin{array}{c}
\alpha_0, \alpha_x \Rightarrow \{\beta_i^l(i)\}_{i \in \{1, \ldots, k\}}^{l(i) \neq 0} \mid x \in \{-j, \ldots, -1\} \\
\alpha_0 \Rightarrow \beta_0^y, \{\beta_i^l(i)\}_{i \in \{1, \ldots, k\}}^{l(i) \neq 0} \mid y \in \{1, \ldots, k\} \\
\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_i^l(i) \mid i = 1, \ldots, k)\}_{i = 1}^k
\end{array}
\]
The hyper-sequent rule

For a better appearance, let

\[
\begin{align*}
J &= \{-j, \ldots, -1\} \\
K &= \{1, \ldots, k\} \\
K^{(l)} &= \{y \in K \mid l(y) = 0\} \\
\Omega_{K}^l &= \{\beta_{l(i)}^i\}_{i \in K} \\
D &= \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\end{align*}
\]

\[
\vdash \alpha_0, \alpha_x \Rightarrow \Omega_{K}^l \mid x \in J \mid \alpha_0 \Rightarrow \beta_0^y, \Omega_{K}^l \mid y \in K^{(l)} \mid l \in D
\]

\[
\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_1^i, \ldots, \beta_{j_i}^i; \beta_0^i)\}_{i \in K}
\]

Unfold to regular-sequent rules.

- not genuine ‘hyper’
  - only 1 seq. in the conclusion (in this only nbd rule).
- specify for each premise (hyper-sequent) with “name” \(l\), index (from \(J \cup K^{(l)}\)) of the regular seq. that exemplifies its provability.
- unfold to infinitely many regular seq. rules, with \(j, k, j_1, \ldots, j_k\) and the choice of indexes as parameters.
The hyper-sequent rule

For a better appearance, let

\[
\begin{align*}
J &= \{-j, ..., -1\} \\
K &= \{1, ..., k\} \\
K^{(l)} &= \{y \in K \mid l(y) = 0\} \\
\Omega^K_l &= \{\beta^i_{l(i)}\}_{i \in K}^{l(i) \neq 0} \\
D &= \bigotimes_{i \in K} \{0, 1, ..., j_i\}
\end{align*}
\]

\[
\left[ \left| \alpha_0, \alpha_x \Rightarrow \Omega^K_l \right|_{x \in J} \mid \alpha_0 \Rightarrow \beta^y_0, \Omega^K_l \right|_{y \in K^{(l)}} \right]_{l \in D}
\]

\[
\square(\alpha_1, ..., \alpha_j; \alpha_0) \Rightarrow \{\square(\beta^i_1, ..., \beta^i_{j_i}; \beta^i_0)\}_{i \in K}
\]

Unfold to regular-sequent rules.

- not genuine ‘hyper’
  - only 1 seq. in the conclusion (in this only nbd rule).
- specify for each premise (hyper-sequent) with “name” \( l \),
  index (from \( J \cup K^{(l)} \))
  of the regular seq. that exemplifies its provability.
- unfold to infinitely many regular seq. rules,
  with \( j, k, j_1, ..., j_k \) and the choice of indexes as parameters.
For a better appearance, let

\[
\begin{align*}
J &= \{-j, \ldots, -1\} & K &= \{1, \ldots, k\} \\
K^{(l)} &= \{y \in K \mid l(y) = 0\} & \Omega_K^l &= \{\beta_{l(i)}^i\}_{i \in K}^{l(i) \neq 0} & D &= \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\end{align*}
\]

\[
\left[\left|\alpha_0, \alpha_x \Rightarrow \Omega_K^l\right|_{x \in J} \mid \alpha_0 \Rightarrow \beta_0^y, \Omega_K^l \mid y \in K^{(l)}\right]_{l \in D}
\]

\[
\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta_1^i, \ldots, \beta_{j_i}^i; \beta_0^i)\}_{i \in K}
\]

**Unfold** to regular-sequent rules.

- not genuine ‘hyper’
  - only 1 seq. in the conclusion (in this only nbd rule).
- specify for each premise (hyper-sequent) with “name” \(l\),
  index (from \(J \uplus K^{(l)}\))
  of the regular seq. that exemplifies its provability.
- unfold to infinitely many regular seq. rules,
  with \(j, k, j_1, \ldots, j_k\) and the choice of indexes as parameters.
The hyper-sequent rule

- For a better appearance, let

\[
\begin{align*}
J &= \{-j, \ldots, -1\} \quad K = \{1, \ldots, k\} \\
K^{(l)} &= \{y \in K \mid l(y) = 0\} \quad \Omega^l_K = \{\beta^i_{l(i)}\}_{i \in K} \\
D &= \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\end{align*}
\]

\[
\left[\left|\alpha_0, \alpha_x \Rightarrow \Omega^l_K\right|_{x \in J} \left| \alpha_0 \Rightarrow \beta^y_0, \Omega^l_K\right|_{y \in K^{(l)}}\right]_{l \in D}
\]

\[
\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta^i_1, \ldots, \beta^i_j; \beta^i_0)\}_{i \in K}
\]

- **Unfold** to regular-sequent rules.
  - not genuine ‘hyper’
    - only 1 seq. in the conclusion (in this only nbd rule).
  - specify for each premise (hyper-sequent) with “name” \(l\), index (from \(J \uplus K^{(l)}\)) of the regular seq. that exemplifies its provability.
  - unfold to infinitely many regular seq. rules,
    with \(j, k, j_1, \ldots, j_k\) and the choice of indexes as parameters.
Start by the nbd rule of TABinl.

\[
\begin{align*}
J &= \{-j, \ldots, -1\} & K &= \{1, \ldots, k\} \\
K^{(i)} &= \{y \in K \mid l(y) = 0\} & \Omega^l_K &= \left\{ \beta^i_{l(i)} \right\}_{i \in K}^{l(i) \neq 0} \\
D &= \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\end{align*}
\]

Choice of indexes presented by a function \( f \) on names:

\[
\forall I \in D \left\{ \begin{array}{ll}
f(I) = x & \text{if premise } I \text{ is } \vdash \alpha_0, \alpha_{-x} \Rightarrow \Omega^l_K, \\
f(I) = y & \text{if premise } I \text{ is } \vdash \alpha_0 \Rightarrow \beta^y_0, \Omega^l_K.
\end{array} \right.
\]

We write \( f_I \) instead of \( f(I) \) for a better appearance...

\( f : D \mapsto J \cup K \) s.t.

(adequacy) \( \forall I \in D \)(\( f_I \in K \) implies \( f_I \in K^{(l)} \), e.g., \( l(f_I) = 0 \)).
Start by the nbd rule of TABinl.

\[
\begin{align*}
J &= \{-j, \ldots, -1\} & K &= \{1, \ldots, k\} \\
K^{(l)} &= \{y \in K \mid l(y) = 0\} & \Omega_K^l &= \left\{ \beta_i^l \right\}_{i \in K}^{l(i) \neq 0} & D &= \bigotimes_{i \in K} \{0,1,\ldots,j_i\}
\end{align*}
\]

\[
\frac{\vert \alpha_0, \alpha_{-x} \Rightarrow \Omega_K^l \mid_{x \in J} \mid \alpha_0 \Rightarrow \beta_0^y, \Omega_K^l \mid_{y \in K^{(l)}}}{\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta_1^i, \ldots, \beta_j^i; \beta_0^i)\}_{i \in K}}
\]

Choice of indexes presented by a function \( f \) on names:

\[
\forall l \in D \left\{ \begin{array}{ll}
  f(l) = x & \text{if premise } l \text{ is } \vdash \alpha_0, \alpha_{-x} \Rightarrow \Omega_K^l, \\
  f(l) = y & \text{if premise } l \text{ is } \vdash \alpha_0 \Rightarrow \beta_0^y, \Omega_K^l.
\end{array} \right.
\]

We write \( f_i \) instead of \( f(l) \) for a better appearance...

\[ f : D \mapsto J \cup K \text{ s.t.} \]

(aeadqacy) \( \forall l \in D)(f_i \in K \text{ implies } f_i \in K^{(l)} \text{, e.g., } l(f_i) = 0) \).
Start by the nbd rule of TABinl.

\[ J = \{-j, \ldots, -1\} \quad K = \{1, \ldots, k\} \]

\[ K^{(l)} = \{y \in K \mid l(y) = 0\} \quad \Omega_K^l = \{\beta^i_{l(i)}\}_{i \in K}^{l(i) \neq 0} \quad D = \bigotimes_{i \in K} \{0, 1, \ldots, j_i\} \]

\[ \alpha_0, \alpha_{-x} \Rightarrow \Omega_K^l \quad x \in J \quad \alpha_0 \Rightarrow \beta_0^y, \Omega_K^l \quad y \in K^{(l)} \]

\[ \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box (\beta^i_1, \ldots, \beta^i_j; \beta_0^i)\}_{i \in K} \]

Choice of indexes presented by a function \( f \) on names:

\[ \forall l \in D \left\{ \begin{array}{ll} f(l) = x \in J & \text{if premise } l \text{ is } \vdash \alpha_0, \alpha_{-x} \Rightarrow \Omega_K^l, \\
 f(l) = y \in K^{(l)} & \text{if premise } l \text{ is } \vdash \alpha_0 \Rightarrow \beta_0^y, \Omega_K. \end{array} \right. \]

We write \( f_i \) instead of \( f(l) \) for a better appearance...

\[ f : D \mapsto J \cup K \text{ s.t.} \]

(adequacy) \( \forall l \in D \) \( (f_i \in K \text{ implies } f_i \in K^{(l)}, \text{ e.g., } l(f_i) = 0) \).
Sequent calculus G3inl

- Group premises w.r.t. \( f \)-images of their names.

\[
J = \{-j, \ldots, -1\} \quad \quad \quad \quad K = \{1, \ldots, k\} \\
K^{(l)} = \{y \in K \mid l(y) = 0\} \quad \quad \Omega^l_K = \{\beta^l_{i(i)}\}_{i \in K}^{l(i) \neq 0} \quad D = \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\]

\[
[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^l_K]_{f_i \in J}^{l \in D} \quad [\alpha_0 \Rightarrow \beta^f_i, \Omega^l_K]_{f_i \in K^{(l)}}^{l \in D} \\
\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta^i_1, \ldots, \beta^i_{j_i}; \beta^i_0)\}_{i \in K}
\]

- Choice of indexes presented by a function \( f \) on names:

\[
\forall l \in D \left\{ \begin{array}{ll}
f(l) = x \in J & \text{if premise } l \text{ is } \vdash \alpha_0, \alpha_{-x} \Rightarrow \Omega^l_K, \\
f(l) = y \in K^{(l)} & \text{if premise } l \text{ is } \vdash \alpha_0 \Rightarrow \beta^y_0, \Omega^l_K.
\end{array} \right.
\]

We write \( f_i \) instead of \( f(l) \) for a better appearance...

- \( f : D \mapsto J \cup K \text{ s.t.} \)

\( (\text{adequacy}) \ (\forall l \in D)(f_i \in K \text{ implies } f_i \in K^{(l)}, \text{ e.g., } l(f_i) = 0). \)
Sequent calculus G3inl

- Group premises w.r.t. $f$-images of their names.

$$
\left(\begin{array}{c}
J = \{-j, \ldots, -1\} \\
K^{(l)} = \{y \in K \mid l(y) = 0\} \\
K = \{1, \ldots, k\} \\
\Omega^l_K = \left\{\beta^i_{l(i)}\right\}_{i \in K}^{l(i) \neq 0} \\
D = \prod_{i \in K} \{0, 1, \ldots, j_i\}
\end{array}\right)
$$

$$\left[\frac{\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^l_K}{f_i \in J} \right]_l \left[\frac{\alpha_0 \Rightarrow \beta^f_0, \Omega^l_K}{f_i \in K^{(l)}} \right]_l \prod, \Box (\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \left\{\Box (\beta^i_1, \ldots, \beta^i_{j_i}; \beta^i_0)\right\}_{i \in K}, \Sigma$$

- In multi-set based G3, built-in weakening is necessary.

  - This is $\left(\Box^{j,k,f}_{\langle j_1, \ldots, j_k \rangle}\right)$, nbd rule with parameters $j, k, j_1, \ldots, j_k, f$.
  - It respects the proper sub-formula property (no built-in contraction).
  - G3inl is G3cp (in the language of INL) extended by all $\left(\Box^{j,k,f}_{\langle j_1, \ldots, j_k \rangle}\right)$ where $f$ is adequate w.r.t. its other parameters.
Sequent calculus G3inl

- Group premises w.r.t. \( f \)-images of their names.

\[
\begin{align*}
J &= \{-j, \ldots, -1\} \quad K = \{1, \ldots, k\} \\
K^{(I)} &= \{y \in K \mid I(y) = 0\} \quad \Omega_K^I = \left\{ \beta_{l(i)}^i \right\}_{i \in K}^{l(i) \neq 0} \\
D &= \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\end{align*}
\]

\[
[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega_K^I]_{f_i \in J} \quad I \in D \\
[\alpha_0 \Rightarrow \beta_0^f, \Omega_K^I]_{f_i \in K^{(I)}} \\
\prod, \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_1^i, \ldots, \beta_{j_i}^i; \beta_0^i)\}_{i \in K, \Sigma}
\]

- In multi-set based G3, built-in weakening is necessary.
- This is \( \square_{\langle j_1, \ldots, j_k \rangle} \), nbd rule with parameters \( j, k, j_1, \ldots, j_k, f \).
- It respects the proper sub-formula property (no built-in contraction).
- G3inl is G3cp (in the language of INL) extended by all \( \square_{\langle j_1, \ldots, j_k \rangle} \) where \( f \) is adequate w.r.t. its other parameters.
Sequent calculus G3inl

- Group premises w.r.t. $f$-images of their names.

\[
\begin{align*}
J &= \{-j, \ldots, -1\} \quad \quad K = \{1, \ldots, k\} \\
K^{(l)} &= \{y \in K \mid l(y) = 0\} \quad \Omega_K^l = \left\{ \beta_{l(i)}^i \right\}_{i \in K}^{l(i) \neq 0} \\
D &= \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}
\end{align*}
\]

\[
[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega_K^l]_{f_i \in J}^{f_i \in K^{(l)}} [\alpha_0 \Rightarrow \beta_0^f, \Omega_K^l]_{f_i \in D}^{f_i \in K^{(l)}} \Pi, \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\Box(\beta_1^i, \ldots, \beta_{j_i}^i; \beta_0^i)\}_{i \in K, \Sigma}
\]

- In multi-set based G3, built-in weakening is necessary.
- This is $\left(\Box_{\langle j_1, \ldots, j_k \rangle}^{j, k, f}\right)$, nbd rule with parameters $j$, $k$, $j_1$, $\ldots$, $j_k$, $f$.
- It respects the proper sub-formula property (no built-in contraction).
- G3inl is G3cp (in the language of INL) extended by all $\left(\Box_{\langle j_1, \ldots, j_k \rangle}^{j, k, f}\right)$ where $f$ is adequate w.r.t. its other parameters.
G3inl
- admits Weakening, Contraction, and Cut;
- is sound and complete (via equivalent with Hinl).
- enjoys the sub-formula property,
supports mechanical proof-search.

A splitting version of G3inl leads to a constructive proof of
INL’s Lyndon interpolation theorem.
Sequent calculus G3inl

- G3inl
  - admits Weakening, Contraction, and Cut;
  - is sound and complete (via equivalent with Hinl).
  - enjoys the sub-formula property,
    supports mechanical proof-search.

- A splitting version of G3inl leads to a constructive proof of
  INL’s Lyndon interpolation theorem.
G3inl

- admits Weakening, Contraction, and Cut;
- is sound and complete (via equivalent with Hinl).
- enjoys the sub-formula property,
  supports mechanical proof-search.

A splitting version of G3inl leads to a constructive proof of
INL’s Lyndon interpolation theorem.
Lyndon interpolation theorem
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 $\theta$ s.t.:

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

(a ‘polarized’ Craig interpolation)

A general form on sequents:

If $G3\text{inl} \vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R$, then there is a formula $\theta$ s.t.:

- $V^\pm(\theta) \subseteq (V^\pm(\Pi_R, \Sigma_L)) \cap (V^\pm(\Pi_L, \Sigma_R))$
- $G3\text{inl} \vdash \Pi_L \Rightarrow \Sigma_L, \theta$ and $G3\text{inl} \vdash \theta, \Pi_R \Rightarrow \Sigma_R$.

The method of splitting sequent calculus (K. Schütte) works.
Lyndon interpolation of INL

- Lydon interpolation theorem:
  \( (\text{Let } V^+(\alpha)/V^-(\alpha) \text{ denotes positive/negative atoms in } \alpha) \)
  
  If INL \( \vdash \phi \rightarrow \psi \), then there is a formula \( \theta \) s.t.:
  
  - \( V^\pm(\theta) \subseteq V^\pm(\phi) \cap V^\pm(\psi) \)
  - INL \( \vdash \phi \rightarrow \theta \) and INL \( \vdash \theta \rightarrow \psi \).

  (a ‘polarized’ Craig interpolation)

- A general form on sequents:
  
  If G3inl \( \vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R \), then there is a formula \( \theta \) s.t.:
  
  - \( V^\pm(\theta) \subseteq (V^\pm(\Pi_R, \Sigma_L)) \cap (V^\pm(\Pi_L, \Sigma_R)) \)
  - G3inl \( \vdash \Pi_L \Rightarrow \Sigma_L, \theta \) and G3inl \( \vdash \theta, \Pi_R \Rightarrow \Sigma_R \).

- The method of splitting sequent calculus (K. Schütte) works.
Lyndon 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 \( \theta \) s.t.:

- \( \mathcal{V}^\pm (\theta) \subseteq \mathcal{V}^\pm (\phi) \cap \mathcal{V}^\pm (\psi) \)
- \( \text{INL} \vdash \phi \rightarrow \theta \), and \( \text{INL} \vdash \theta \rightarrow \psi \).

(a ‘polarized’ Craig interpolation)

A general form on sequents:

If \( \text{G3inl} \vdash \Pi_L, \Pi_R \Rightarrow \Sigma_L, \Sigma_R \), then there is a formula \( \theta \) s.t.:

- \( \mathcal{V}^\pm (\theta) \subseteq (\mathcal{V}^\pm (\Pi_R, \Sigma_L)) \cap (\mathcal{V}^\pm (\Pi_L, \Sigma_R)) \)
- \( \text{G3inl} \vdash \Pi_L \Rightarrow \Sigma_L, \theta \), and \( \text{G3inl} \vdash \theta, \Pi_R \Rightarrow \Sigma_R \).

The method of splitting sequent calculus (K. Schütte) works.
A splitting version of G3inl

- Antecedent and succedent of eachsequent split into two parts (left | right), like $\Gamma_L | \Gamma_R \Rightarrow \Delta_L | \Delta_R$.
- For each rule, given an arbitrary splitting conclusion, depending on locations of the principals, the splitting calculus should offer a rule with splitting premises where actives respect polarities, and also specify a construction of the conclusion’s interpolant based on assumed interpolants of premises.
A splitting version of G3inl

- Antecedent and succedent of each sequent split into two parts (left | right), like $\Gamma_L | \Gamma_R \Rightarrow \Delta_L | \Delta_R$.
- For each rule, given an arbitrary splitting conclusion, depending on locations of the principals, the splitting calculus should offer a rule with splitting premises where actives respect polarities, and also specify a construction of the conclusion’s interpolant based on assumed interpolants of premises.
Lyndon interpolation of INL

- For the nbe rule \((\Box^{j,k,f}_{\langle j_1,\ldots,j_k \rangle})\):

\[
\begin{align*}
\left[ \alpha_0, \alpha_{-f_I} \Rightarrow \Omega^I_K \right]_{I \in D} & \left[ \alpha_0 \Rightarrow \beta_0^I, \Omega^I_K \right]_{I \in D} \\
\Pi, \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) & \Rightarrow \{ \Box(\beta_1^i, \ldots, \beta_j^i; \beta_0^i) \}_{i \in K, \Sigma}
\end{align*}
\]

- Weakening \(\Pi, \Sigma\) never matter.
  - Assume the negative principal \(\Box(\alpha_1, \ldots, \alpha_j; \alpha_0)\) goes left, the case that it goes right is quite similar.
  - Recall: \(J = \{-j, \ldots, -1\}, K = \{1, \ldots, k\}, D = \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}\).
  - W.l.o.g., assume that positive principals split to \(\{\Box(\beta_1^i, \ldots, \beta_j^i)\}_{i \in L} \{\Box(\beta_1^i, \ldots, \beta_j^i)\}_{i \in R}\) (where \(L \cup R = K\)).
  - Also let \(D_L = \bigotimes_{i \in L} \{0, 1, \ldots, j_i\}, D_R = \bigotimes_{i \in R} \{0, 1, \ldots, j_i\}\).
  - For \(I = \langle I(1), \ldots, I(k) \rangle \in D\), let \(I_L := I \upharpoonright L \in D_L\), and \(I_R := I \upharpoonright R \in D_R\).
  - Let \(0^R\) be a list of 0's with index-set \(R\), similar for \(0^L\).
  - Denote the index-disjoint merge of lists \(M\) and \(N\) by \(M \circ N\).
Lyndon interpolation of INL

- For the nbe rule (\(\square^{j,k,f}_{\langle j_1,\ldots,j_k \rangle}\)):

\[
\frac{[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^I_{f_i \in J}]}{} \quad \Pi, \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_1^i, \ldots, \beta_{j_i}^i; \beta_0^i)\}^i \in K, \Sigma
\]

- Weakening \(\Pi, \Sigma\) never matter.
- Assume the negative principal \(\square(\alpha_1, \ldots, \alpha_j; \alpha_0)\) goes left, the case that it goes right is quite similar.

- Recall: \(J = \{-j, \ldots, -1\}\), \(K = \{1, \ldots, k\}\), \(D = \bigotimes_{i \in K}\{0, 1, \ldots, j_i\}\).
- W.l.o.g., assume that positive principals split to \(\{\square(\beta_1^i, \ldots, \beta_{j_i}^i)\}^i \in L\}{\square(\beta_1^i, \ldots, \beta_{j_i}^i)\}^i \in R\) (where \(L \sqcup R = K\)).
- Also let \(D_L = \bigotimes_{i \in L}\{0, 1, \ldots, j_i\}\), \(D_R = \bigotimes_{i \in R}\{0, 1, \ldots, j_i\}\).
  - For \(I = \langle I(1), \ldots, I(k) \rangle\) \(\in D\), let \(I_L := I \upharpoonright L \in D_L\), and \(I_R := I \upharpoonright R \in D_R\).
  - Let \(0^R\) be a list of 0's with index-set \(R\), similar for \(0^L\).
  - Denote the index-disjoint merge of lists \(M\) and \(N\) by \(M \odot N\).
For the nbe rule \((\square_{\langle j_1, \ldots, j_k \rangle})\):

\[
\frac{[\alpha_0, \alpha_{-f_I} \Rightarrow \Omega_I^l]_{l \in D}}{\Pi, \square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_{i_1}^1, \ldots, \beta_{i_j}^j; \beta_0^1)\}_{i \in K}, \Sigma}
\]

- Weakening \(\Pi, \Sigma\) never matter.
- Assume the negative principal \(\square(\alpha_1, \ldots, \alpha_j; \alpha_0)\) goes left, the case that it goes right is quite similar.
- Recall: \(J = \{-j, \ldots, -1\}\), \(K = \{1, \ldots, k\}\), \(D = \bigotimes_{i \in K} \{0, 1, \ldots, j_i\}\).
- W.l.o.g., assume that positive principals split to \(\{\square(\beta_{i_1}^1, \ldots, \beta_{i_j}^j)\}_{i \in L} \cup \{\square(\beta_{i_1}^1, \ldots, \beta_{i_j}^j)\}_{i \in R}\) (where \(L \uplus R = K\)).
- Also let \(D_L = \bigotimes_{i \in L} \{0, 1, \ldots, j_i\}\), \(D_R = \bigotimes_{i \in R} \{0, 1, \ldots, j_i\}\).
- For \(I = \langle I(1), \ldots, I(k) \rangle \in D\), let \(I_L := I \upharpoonright L \in D_L\), and \(I_R := I \upharpoonright R \in D_R\). Let \(0^R\) be a list of 0's with index-set \(R\), similar for \(0^L\).
- Denote the index-disjoint merge of lists \(M\) and \(N\) by \(M \circ N\).
Lyndon interpolation of INL

- For the nbe rule \((\Box^{j,k,f}_{\langle j_1,\ldots,j_k \rangle})\):

\[
\begin{align*}
\left[ \alpha_0, \alpha_{-f_i} \Rightarrow \Omega^I_K \right]_{f_i \in J} & \left[ \alpha_0 \Rightarrow \beta^f_0, \Omega^I_K \right]_{f_i \in K} \\
\Pi, \Box(\alpha_1, \ldots, \alpha_j; \alpha_0) & \Rightarrow \{ \Box(\beta^i_1, \ldots, \beta^i_j; \beta^i_0) \}_{i \in K}, \Sigma
\end{align*}
\]

- Weakening \(\Pi, \Sigma\) never matter.
- Assume the negative principal \(\Box(\alpha_1, \ldots, \alpha_j; \alpha_0)\) goes left, the case that it goes right is quite similar.
- Recall: \(J = \{-j, \ldots, -1\}\), \(K = \{1, \ldots, k\}\), \(D = \bigotimes_{i \in K}\{0, 1, \ldots, j_i\}\).
- W.l.o.g., assume that positive principals split to \(\{ \Box(\beta^i_1, \ldots, \beta^i_j) \}_{i \in L}|\{ \Box(\beta^i_1, \ldots, \beta^i_j) \}_{i \in R}\) (where \(L \uplus R = K\)).
- Also let \(D_L = \bigotimes_{i \in L}\{0, 1, \ldots, j_i\}\), \(D_R = \bigotimes_{i \in R}\{0, 1, \ldots, j_i\}\).
  - For \(l = \langle l(1), \ldots, l(k) \rangle \in D\), let \(l_L := l|L \in D_L\), and \(l_R := l|R \in D_R\).
  - Let \(0^R\) be a list of 0’s with index-set \(R\), similar for \(0^L\).
  - Denote the index-disjoint merge of lists \(M\) and \(N\) by \(M \circ N\).
Lyndon interpolation of INL

- When the only negative principal goes left:

\[
\left[ \alpha_0, \alpha_{-f_i} \Rightarrow \Omega_I^L \right]_{I \in D} \left[ \alpha_0 \Rightarrow \beta_0^{f_i}, \Omega_K^I \right]_{I \in D} \\
\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{ \square(\beta_1^i, \ldots, \beta_j^i; \beta_0^i) \}_{i \in L} \{ \square(\beta_1^i, \ldots, \beta_j^i; \beta_0^i) \}_{i \in R}
\]

- For premise with name \( I \in D \), choose split premise:
  - \( \alpha_0, \alpha_{-f_i} \Rightarrow \Omega_I^L | \Omega_I^R \), when \( f_i \in J \);
  - \( \alpha_0 \Rightarrow \Omega_I^L | \beta_0^{f_i} | \Omega_I^R \), when \( f_i \in L \subseteq K \);
  - \( \alpha_0 \Rightarrow \Omega_I^L | \beta_0^{f_i} | \Omega_I^R \), when \( f_i \in R \subseteq K \).

and denote its assumed interpolant by \( \theta_I \).

- The desired interpolant for the conclusion is

\[
\bigvee_{M \in D_L} \square \left( \left\{ \theta_{M \circ N} \right\}_{N \in D_L} \right) ; \bigwedge_{N \in D_R} \theta_{M \circ N}
\]
Lyndon interpolation of INL

When the only negative principal goes left:

\[
[\alpha_0, \alpha_{-f_I} \Rightarrow \Omega^I_K]_{f_I \in J} \quad [\alpha_0 \Rightarrow \beta_0^{f_I}, \Omega^I_K]_{f_I \in K}
\]

\[
\square(\alpha_1, \ldots, \alpha_j; \alpha_0) \Rightarrow \{\square(\beta_1^{f_I}, \ldots, \beta_j^{f_I}; \beta_0^{f_I})\}_{i \in L} \{\square(\beta_1^{f_I}, \ldots, \beta_j^{f_I}; \beta_0^{f_I})\}_{i \in R}
\]

For premise with name \( I \in D \), choose split premise:

- \( \alpha_0, \alpha_{-f_I} \Rightarrow \Omega^I_L \Omega^I_R \), when \( f_I \in J \);
- \( \alpha_0 \Rightarrow \Omega^I_L \beta_0^{f_I} \Omega^I_R \), when \( f_I \in L \subseteq K \);
- \( \alpha_0 \Rightarrow \Omega^I_L \beta_0^{f_I} \Omega^I_R \), when \( f_I \in R \subseteq K \).

and denote its assumed interpolant by \( \theta_I \).

The desired interpolant for the conclusion is

\[
f_{M \circ 0 R} \in R \bigvee_{M \in D_L} \square \left( \left\{ \theta_{M \circ N} \right\}_{N \in D_R} : f_{M \circ N} \in J \cup L \right) ; \bigwedge_{N \in D_R} \theta_{M \circ N}
\]
Lyndon interpolation of INL

- When the only negative principal goes left:

\[
\begin{align*}
[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^i_K]_{i \in D} \quad & [\alpha_0 \Rightarrow \beta^i_0, \Omega^i_K]_{i \in D} \\
\Box(\alpha_1, \ldots, \alpha_j; \alpha_0) & \Rightarrow \{\Box(\beta^i_1, \ldots, \beta^i_j; \beta^i_0)\}_{i \in L} \{\Box(\beta^i_1, \ldots, \beta^i_j; \beta^i_0)\}_{i \in R}
\end{align*}
\]

- For premise with name \(i \in D\), choose split premise:
  \begin{itemize}
  \item \(\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^i_L|\Omega^i_R\), when \(f_i \in J\);
  \item \(\alpha_0 \Rightarrow \Omega^i_L, \beta^i_0 |\Omega^i_R\), when \(f_i \in L \subseteq K\);
  \item \(\alpha_0 \Rightarrow \Omega^i_L|\beta^i_0, \Omega^i_R\), when \(f_i \in R \subseteq K\).
  \end{itemize}

and denote its assumed interpolant by \(\theta_i\).

- The desired interpolant for the conclusion is

\[
\bigvee_{M \in D_L} \Box \left( \{\theta^M_{\alpha N}\}_{N \in D_L} ; \bigwedge_{N \in D_R} \theta^M_{\alpha N} \right)
\]

Junhua Yu (junhua.yu.5036@outlook.com)

INL - tableau, sequent calculus, interpolation
When the only negative principal goes left:

\[
\begin{align*}
[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^I_K]_{f_i \in J} & \Rightarrow \Omega^I_K \bigcup_{f_i \in K} \\
\square(a_1, \ldots, a_j; a_0) & \Rightarrow \{\square(b^i_1, \ldots, b^i_j; b^i_0)\}_{i \in L} \bigcup \{\square(b^i_1, \ldots, b^i_j; b^i_0)\}_{i \in R}
\end{align*}
\]

For premise with name \( l \in D \), choose split premise:

- \( \alpha_0, \alpha_{-f_i} \Rightarrow \Omega^I_L | \Omega^I_R \), when \( f_i \in J \);
- \( \alpha_0 \Rightarrow \Omega^I_L | \beta^i_0 | \Omega^I_R \), when \( f_i \in L \subseteq K \);
- \( \alpha_0 \Rightarrow \Omega^I_L | \beta^i_0, \Omega^I_R \), when \( f_i \in R \subseteq K \).

and denote its assumed interpolant by \( \theta_i \).

The desired interpolant for the conclusion is

\[
\bigvee_{M \in D_L} \square \left( \left\{ \theta_{M \circ N} \right\}_{N \in D_R} \bigcup_{M \circ N \in J \cup L} f_{M \circ N \in R} \bigcup_{N \in D_R} \theta_{M \circ N} \right)
\]
Lyndon interpolation of INL

When the only negative principal goes left:

\[
\begin{align*}
[\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^l_I]_{f_i \in J} & \quad [\alpha_0 \Rightarrow \beta^j_0, \Omega^l_K]_{f_i \in K} \\
\square(\alpha_1, \ldots, \alpha_j; \alpha_0) & \Rightarrow \{\square(\beta^i_1, \ldots, \beta^i_j; \beta^i_0)\}_{i \in L} \{\square(\beta^i_1, \ldots, \beta^i_j; \beta^i_0)\}_{i \in R}
\end{align*}
\]

- For premise with name \(l \in D\), choose split premise:
  - \(\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^l_L | \Omega^l_R\), when \(f_i \in J\);
  - \(\alpha_0 \Rightarrow \Omega^l_L, \beta^f_0 | \Omega^l_R\), when \(f_i \in L \subseteq K\);
  - \(\alpha_0 \Rightarrow \Omega^l_L | \beta^f_0, \Omega^l_R\), when \(f_i \in R \subseteq K\).

and denote its assumed interpolant by \(\theta_i\).

- The desired interpolant for the conclusion is

\[
\bigvee_{M \in D_L} \square \left( \left\{ \theta_{M \circ N} \right\}_{N \in D_R} \quad \bigwedge_{N \in D_R} \theta_{M \circ N} \right)
\]
When the only negative principal goes right:

\[
\begin{align*}
\left[ \alpha_0, \alpha_{-f_i} \Rightarrow \Omega^L_I \right]_{f_i \in J} & \left[ \alpha_0 \Rightarrow \beta_0^f, \Omega^L_K \right]_{f_i \in K} \\
\square (\alpha_1, \ldots, \alpha_j ; \alpha_0) & \Rightarrow \left\{ \square (\beta_1^i, \ldots, \beta_j^i ; \beta_0^i) \right\}_{i \in L} \left\{ \square (\beta_1^i, \ldots, \beta_j^i ; \beta_0^i) \right\}_{i \in R}
\end{align*}
\]

For premise with name \( I \in D \), choose split premise:

- \( |\alpha_0, \alpha_{-f_i} \Rightarrow \Omega^L_I | \Omega^R_I \), when \( f_i \in J \);
- \( |\alpha_0 \Rightarrow \Omega^L_I | \beta_0^f | \Omega^R_I \), when \( f_i \in L \subseteq K \);
- \( |\alpha_0 \Rightarrow \Omega^L_I | \beta_0^f | \Omega^R_I \), when \( f_i \in R \subseteq K \).

And denote its assumed interpolant by \( \theta_I \).

The desired interpolant for the conclusion is

\[
\neg \bigvee_{N \in D_R} \square \left( \left\{ \neg \theta_M \right\}_{M \in D_L} ; \bigwedge_{M \in D_L} \neg \theta_M \right)
\]
Future directions
More properties of the nbd function leads to genuine hyper-sequent?
Modification on language
  bound on number of instances
  infinite language (conjunctions, disjunctions, instances).
Nabla operator v.s. $\Box(\alpha_1, \ldots, \alpha_j; \alpha_1 \lor \ldots \lor \alpha_j)$.
Uniform interpolation
  the method of R. Iemhoff seems to work.
More properties of the nbd function
  leads to genuine hyper-sequent ?
Modification on language
  bound on number of instances
    infinite language (conjunctions, disjunctions, instances).
Nabla operator v.s. □(α₁, ..., αⱼ; α₁ ∨ ... ∨ αⱼ).
Uniform interpolation
  the method of R. Iemhoff seems to work.
Future Directions

- More properties of the nbd function
  - leads to genuine hyper-sequent?
- Modification on language
  - bound on number of instances
  - infinite language (conjunctions, disjunctions, instances).

- Nabla operator v.s. $\Box(\alpha_1, \ldots, \alpha_j; \alpha_1 \lor \ldots \lor \alpha_j)$.
- Uniform interpolation
  - the method of R. Iemhoff seems to work.
Future Directions

- More properties of the nbd function
  - leads to genuine hyper-sequent?
- Modification on language
  - bound on number of instances
  - infinite language (conjunctions, disjunctions, instances).
- Nabla operator v.s. $\Box(\alpha_1, \ldots, \alpha_j; \alpha_1 \lor \ldots \lor \alpha_j)$.
- Uniform interpolation
  - the method of R. Iemhoff seems to work.
Future Directions

- More properties of the nbd function
  - leads to genuine hyper-sequent?
- Modification on language
  - bound on number of instances
  - infinite language (conjunctions, disjunctions, instances).
- Nabla operator v.s. $\Box(\alpha_1, \ldots, \alpha_j; \alpha_1 \lor \ldots \lor \alpha_j)$.
- Uniform interpolation
  - the method of R. Iemhoff seems to work.
Thanks!