Exponentiation and Iteration in the Lambek Calculus and Its Variants

Stepan L. Kuznetsov

Steklov Mathematical Institute of RAS, Moscow, Russia

Logic and Applications · Dubrovnik, Croatia, 24–28.09.2025

The Lambek Calculus

The Lambek calculus **L** comes from several sources in logic and applications.

The Lambek Calculus

The Lambek calculus **L** comes from several sources in logic and applications.

1. The logical basis for **Lambek categorial grammars** [Lambek 1958], supported by completeness w.r.t. models on formal languages [Pentus 1995].

The Lambek Calculus

The Lambek calculus **L** comes from several sources in logic and applications.

1. The logical basis for **Lambek categorial grammars** [Lambek 1958], supported by completeness w.r.t. models on formal languages [Pentus 1995].
2. **L** is one of the **substructural logics** (see [Došen 1992; Restall 2000]): in the sequent form, it has the same **logical** rules as **Int**, but lacks **structural** rules: weakening, contraction, permutation.

The Lambek Calculus

The Lambek calculus **L** comes from several sources in logic and applications.

1. The logical basis for **Lambek categorial grammars** [Lambek 1958], supported by completeness w.r.t. models on formal languages [Pentus 1995].
2. **L** is one of the **substructural logics** (see [Došen 1992; Restall 2000]): in the sequent form, it has the same **logical** rules as **Int**, but lacks **structural** rules: weakening, contraction, permutation.
3. As **Int** is the logic of Heyting lattices, the Lambek calculus (extended with lattice-theoretic, so-called **additive** operations) is the logic of a broader class—**residuated lattices** (see [Galatos, Jipsen, Kowalski, Ono 2007]).

The Lambek Calculus

4. Another natural class of residuated lattices (besides algebras of formal languages) are formed by algebras on **binary relations;** see completeness results [Andréka, Mikulás 1994]. This corresponds to interpreting Lambek formulae as **actions** (e.g., in a computational system). Extending the language with **Kleene star** [Kleene 1956] yields **action logic** [Pratt 1991; Kozen 1994] and its infinitary version [Buszkowski, Palka 2007].

The Lambek Calculus

4. Another natural class of residuated lattices (besides algebras of formal languages) are formed by algebras on **binary relations;** see completeness results [Andréka, Mikulás 1994]. This corresponds to interpreting Lambek formulae as **actions** (e.g., in a computational system). Extending the language with **Kleene star** [Kleene 1956] yields **action logic** [Pratt 1991; Kozen 1994] and its infinitary version [Buszkowski, Palka 2007].
5. Finally, the Lambek calculus can be viewed as a version of non-commutative intuitionistic **linear logic** [Girard 1987; Abrusci 1990], with its informal resource interpretation. This enables adding **exponential** and **subexponential** modalities to the Lambek calculus.

The Lambek Calculus

$$\frac{}{A \to A}\ \mathsf{Id}$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \cdot B, \Delta \to C}\ \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma, \Delta \to A \cdot B}\ \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, \Pi, A \setminus B, \Delta \to C}\ \setminus\mathsf{L} \qquad \frac{A, \Pi \to B}{\Pi \to A \setminus B}\ \setminus\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, B \,/\, A, \Pi, \Delta \to C}\ /\mathsf{L} \qquad \frac{\Pi, A \to B}{\Pi \to B \,/\, A}\ /\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, A, \Delta \to C}{\Gamma, \Pi, \Delta \to C}\ \mathsf{Cut}$$

The Lambek Calculus

$$\frac{}{A \to A}\ \mathsf{Id}$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \cdot B, \Delta \to C}\ \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma, \Delta \to A \cdot B}\ \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, \Pi, A \setminus B, \Delta \to C}\ \setminus\mathsf{L} \qquad \frac{A, \Pi \to B}{\Pi \to A \setminus B}\ \setminus\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, B / A, \Pi, \Delta \to C}\ /\mathsf{L} \qquad \frac{\Pi, A \to B}{\Pi \to B / A}\ /\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, A, \Delta \to C}{\Gamma, \Pi, \Delta \to C}\ \mathsf{Cut}$$

- Product is **multiplicative conjunction**, and divisions (residuals) are directed **implications**.

The Lambek Calculus

$$\frac{}{A \to A}\ \mathsf{Id}$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \cdot B, \Delta \to C}\ \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma, \Delta \to A \cdot B}\ \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, \Pi, A \setminus B, \Delta \to C}\ \setminus\mathsf{L} \qquad \frac{A, \Pi \to B}{\Pi \to A \setminus B}\ \setminus\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, B / A, \Pi, \Delta \to C}\ /\mathsf{L} \qquad \frac{\Pi, A \to B}{\Pi \to B / A}\ /\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, A, \Delta \to C}{\Gamma, \Pi, \Delta \to C}\ \mathsf{Cut}$$

- Product is **multiplicative conjunction**, and divisions (residuals) are directed **implications**.
- The Cut rule is eliminable.

The Lambek Calculus

$$\frac{}{A \to A}\ \mathsf{Id}$$

$$\frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \cdot B, \Delta \to C}\ \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma, \Delta \to A \cdot B}\ \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, \Pi, A \setminus B, \Delta \to C}\ \setminus\mathsf{L} \qquad \frac{A, \Pi \to B}{\Pi \to A \setminus B}\ \setminus\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, B, \Delta \to C}{\Gamma, B / A, \Pi, \Delta \to C}\ /\mathsf{L} \qquad \frac{\Pi, A \to B}{\Pi \to B / A}\ /\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma, A, \Delta \to C}{\Gamma, \Pi, \Delta \to C}\ \mathsf{Cut}$$

- Product is **multiplicative conjunction**, and divisions (residuals) are directed **implications**.
- The Cut rule is eliminable.
- We allow empty left-hand sides of sequents.

Lambek Grammars

N,	$(N \setminus S) \,/\, N$,	N	$\rightarrow S$
John	likes	Mary	

Lambek Grammars

$$N, \qquad (N \setminus S) / N, \qquad N \qquad \rightarrow S$$

John likes Mary

- Here only \L and /L are used.

Lambek Grammars

$$N, \quad (N \setminus S) / N, \quad N \quad \rightarrow S$$

John likes Mary

- Here only \L and /L are used.

$$N / CN, \quad CN, \quad (CN \setminus CN) / (S / N), \quad N, \quad (N \setminus S) / N, \quad (N \setminus S) / N, \quad N \quad \rightarrow S$$

The girl whom John likes hates John

Lambek Grammars

$$\begin{array}{llll} N, & (N \setminus S) \,/\, N, & N & \to S \\ \text{John} & \text{likes} & \text{Mary} & \end{array}$$

- Here only \L and /L are used.

$$\begin{array}{llllllll} N \,/\, CN, & CN, & (CN \setminus CN) \,/\, (S \,/\, N), & N, & (N \setminus S) \,/\, N, & (N \setminus S) \,/\, N, & N & \to S \\ \text{The} & \text{girl} & \text{whom} & \text{John} & \text{likes} & \text{hates} & \text{John} & \end{array}$$

- Here we also use /R, to show that "John likes" is of type $(S \,/\, N)$.

Lambek Grammars

$$N, \quad (N \setminus S) \,/\, N, \quad N \quad \to S$$

John likes Mary

- Here only \L and /L are used.

$$N \,/\, CN, \quad CN, \quad (CN \setminus CN) \,/\, (S \,/\, N), \quad N, \quad (N \setminus S) \,/\, N, \quad (N \setminus S) \,/\, N, \quad N \quad \to S$$

The girl whom John likes hates John

- Here we also use /R, to show that "John likes" is of type $(S \,/\, N)$.

the girl whom John met yesterday $\quad \ldots \to N$

Lambek Grammars

$$N, \quad (N \setminus S) \,/\, N, \quad N \quad \to S$$

John likes Mary

- Here only \L and /L are used.

$$N \,/\, CN, \quad CN, \quad (CN \setminus CN) \,/\, (S \,/\, N), \quad N, \quad (N \setminus S) \,/\, N, \quad (N \setminus S) \,/\, N, \quad N \quad \to S$$

The girl whom John likes hates John

- Here we also use /R, to show that "John likes" is of type $(S \,/\, N)$.

the girl whom John met yesterday $\quad \ldots \to N$

- This is a more complicated phrase, for which an extension of the Lambek calculus is used.

Models for the Lambek Calculus

- The Lambek calculus is the algebraic logic of **residuated monoids**, which a partially ordered monoids with residuals:

$$a \preccurlyeq c \,/\, b \iff a \cdot b \preccurlyeq c \iff b \preccurlyeq a \setminus c.$$

Models for the Lambek Calculus

- The Lambek calculus is the algebraic logic of **residuated monoids**, which a partially ordered monoids with residuals:

$$a \preccurlyeq c \,/\, b \iff a \cdot b \preccurlyeq c \iff b \preccurlyeq a \setminus c.$$

- Lingustic applications motivate a concrete family of residuated monoids, whose elements are formal languages over an alphabet Σ, where

$$A \cdot B = \{uv \mid u \in A, v \in B\}$$
$$A \setminus B = \{u \in \Sigma^* \mid A \cdot \{u\} \subseteq B\}$$
$$B \,/\, A = \{u \in \Sigma^* \mid \{u\} \cdot A \subseteq B\}$$

Models for the Lambek Calculus

- The Lambek calculus is the algebraic logic of **residuated monoids**, which a partially ordered monoids with residuals:

$$a \preccurlyeq c \,/\, b \iff a \cdot b \preccurlyeq c \iff b \preccurlyeq a \setminus c.$$

- Lingustic applications motivate a concrete family of residuated monoids, whose elements are formal languages over an alphabet Σ, where

$$\begin{aligned} A \cdot B &= \{uv \mid u \in A, v \in B\} \\ A \setminus B &= \{u \in \Sigma^* \mid A \cdot \{u\} \subseteq B\} \\ B \,/\, A &= \{u \in \Sigma^* \mid \{u\} \cdot A \subseteq B\} \end{aligned}$$

- Completeness was proved by Pentus [1998].

Models for the Lambek Calculus

- Another family of residuated monoids is formed by algebras of binary relations, of the form $\mathscr{P}(W \times W)$ for a non-empty W.

Models for the Lambek Calculus

- Another family of residuated monoids is formed by algebras of binary relations, of the form $\mathscr{P}(W \times W)$ for a non-empty W.
- These are viewed as non-deterministic actions in a computing system.

Models for the Lambek Calculus

- Another family of residuated monoids is formed by algebras of binary relations, of the form $\mathscr{P}(W \times W)$ for a non-empty W.
- These are viewed as non-deterministic actions in a computing system.
- Product is relational composition, and divisions are "conditional actions":

$$R \cdot S = R \circ S = \{(x, z) \in W \times W \mid \exists y \in W\,((x, y) \in R \text{ and } (y, z) \in S)\}$$
$$R \setminus S = \{(y, z) \in W \times W \mid R \circ \{(y, z)\} \subseteq S\}$$
$$S / R = \{(x, y) \in W \times W \mid \{(x, y)\} \circ R \subseteq S\}$$

Models for the Lambek Calculus

- Another family of residuated monoids is formed by algebras of binary relations, of the form $\mathscr{P}(W \times W)$ for a non-empty W.
- These are viewed as non-deterministic actions in a computing system.
- Product is relational composition, and divisions are "conditional actions":

$$R \cdot S = R \circ S = \{(x, z) \in W \times W \mid \exists y \in W\,((x, y) \in R \text{ and } (y, z) \in S)\}$$
$$R \setminus S = \{(y, z) \in W \times W \mid R \circ \{(y, z)\} \subseteq S\}$$
$$S \,/\, R = \{(x, y) \in W \times W \mid \{(x, y)\} \circ R \subseteq S\}$$

- Completeness proved by Andréka and Mikulás [1994].

Models for the Lambek Calculus

- Another family of residuated monoids is formed by algebras of binary relations, of the form $\mathscr{P}(W \times W)$ for a non-empty W.
- These are viewed as non-deterministic actions in a computing system.
- Product is relational composition, and divisions are "conditional actions":

$$R \cdot S = R \circ S = \{(x, z) \in W \times W \mid \exists y \in W\,((x, y) \in R \text{ and } (y, z) \in S)\}$$
$$R \setminus S = \{(y, z) \in W \times W \mid R \circ \{(y, z)\} \subseteq S\}$$
$$S / R = \{(x, y) \in W \times W \mid \{(x, y)\} \circ R \subseteq S\}$$

- Completeness proved by Andréka and Mikulás [1994].
- Notice that completeness (for both classes of models) quickly ruins when one extends the set of operations!

Additive Operations and Constants

MALC, the **multiplicative-additive Lambek calculus**, is obtained from **L** by means of additive conjunction $\wedge$, additive disjunction $\vee$, and constants $\mathbf{0}$ and $\mathbf{1}$.

$$\frac{\Gamma, A, \Delta \to C \quad \Gamma, B, \Delta \to C}{\Gamma, A \vee B, \Delta \to C}\ \vee\mathsf{L} \qquad \frac{\Pi \to A}{\Pi \to A \vee B}\ \frac{\Pi \to B}{\Pi \to A \vee B}\ \vee\mathsf{R}$$

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, A \wedge B, \Delta \to C}\ \frac{\Gamma, B, \Delta \to C}{\Gamma, A \wedge B, \Delta \to C}\ \wedge\mathsf{L} \qquad \frac{\Pi \to A \quad \Pi \to B}{\Pi \to A \wedge B}\ \wedge\mathsf{R}$$

$$\frac{}{\Gamma, \mathbf{0}, \Delta \to C}\ \mathbf{0}\mathsf{L} \qquad \frac{\Gamma, \Delta \to C}{\Gamma, \mathbf{1}, \Delta \to C}\ \mathbf{1}\mathsf{L} \qquad \frac{}{\to \mathbf{1}}\ \mathbf{1}\mathsf{R}$$

Exponential and Subexponentials

- We further extend **MALC** by means of linear logic **exponential** modality, yielding **!MALC**:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{L} \qquad \frac{!A_1, \ldots, !A_n \to B}{!A_1, \ldots, !A_n \to !B}\ !\mathsf{R} \qquad \frac{\Gamma, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{W}$$

$$\frac{\Gamma, B, !A, \Delta \to C}{\Gamma, !A, B, \Delta \to C} \quad \frac{\Gamma, !A, B, \Delta \to C}{\Gamma, B, !A, \Delta \to C}\ !\mathsf{P} \qquad \frac{\Gamma, !A, !A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{C}$$

Exponential and Subexponentials

- We further extend **MALC** by means of linear logic **exponential** modality, yielding **!MALC**:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{L} \qquad \frac{!A_1, \ldots, !A_n \to B}{!A_1, \ldots, !A_n \to !B}\ !\mathsf{R} \qquad \frac{\Gamma, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{W}$$

$$\frac{\Gamma, B, !A, \Delta \to C}{\Gamma, !A, B, \Delta \to C} \quad \frac{\Gamma, !A, B, \Delta \to C}{\Gamma, B, !A, \Delta \to C}\ !\mathsf{P} \qquad \frac{\Gamma, !A, !A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{C}$$

- A more fine-grained structural control is given by **subexponentials**, in a polymodal system of $!^s$ indexed by $s \in \mathscr{I}$, where $\mathscr{I}$ bears a partial order $\preceq$.

Exponential and Subexponentials

- We further extend **MALC** by means of linear logic **exponential** modality, yielding **!MALC**:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{L} \qquad \frac{!A_1, \ldots, !A_n \to B}{!A_1, \ldots, !A_n \to !B}\ !\mathsf{R} \qquad \frac{\Gamma, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{W}$$

$$\frac{\Gamma, B, !A, \Delta \to C}{\Gamma, !A, B, \Delta \to C}\quad \frac{\Gamma, !A, B, \Delta \to C}{\Gamma, B, !A, \Delta \to C}\ !\mathsf{P} \qquad \frac{\Gamma, !A, !A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{C}$$

- A more fine-grained structural control is given by **subexponentials**, in a polymodal system of $!^s$ indexed by $s \in \mathcal{I}$, where $\mathcal{I}$ bears a partial order $\preceq$.
- Introduction rules:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !^s A, \Delta \to C}\ !\mathsf{L} \qquad \frac{!^{s_1} A_1, \ldots, !^{s_n} A_n \to B}{!^{s_1} A_1, \ldots, !^{s_n} A_n \to !^s B}\ !\mathsf{R},\ s_i \succeq s$$

Exponential and Subexponentials

- In $\mathscr{I}$, we designate three subsets, $\mathscr{W}$, $\mathscr{E}$, and $\mathscr{C}$, upward-closed under $\preceq$.

Exponential and Subexponentials

- In $\mathscr{I}$, we designate three subsets, $\mathscr{W}$, $\mathscr{E}$, and $\mathscr{C}$, upward-closed under $\preceq$.
- These sets control which structural rules are allowed:

$$\frac{\Gamma, \Delta \to C}{\Gamma, !^{w}A, \Delta \to C}\ !\mathsf{W},\ w \in \mathscr{W} \qquad \frac{\Gamma, B, !^{e}A, \Delta \to C}{\Gamma, !^{e}A, B, \Delta \to C} \quad \frac{\Gamma, !^{e}A, B, \Delta \to C}{\Gamma, B, !^{e}A, \Delta \to C}\ !\mathsf{P},\ e \in \mathscr{E}$$

$$\frac{\Gamma, !^{c}A, \Phi, !^{c}A, \Delta \to C}{\Gamma, !^{c}A, \Phi, \Delta \to C} \quad \frac{\Gamma, !^{c}A, \Phi, !^{c}A, \Delta \to C}{\Gamma, \Phi, !^{c}A, \Delta \to C}\ !\mathsf{NC},\ c \in \mathscr{C}$$

Exponential and Subexponentials

- In $\mathscr{I}$, we designate three subsets, $\mathscr{W}$, $\mathscr{E}$, and $\mathscr{C}$, upward-closed under $\preceq$.
- These sets control which structural rules are allowed:

$$\frac{\Gamma, \Delta \to C}{\Gamma, !^{w}A, \Delta \to C}\ !\mathsf{W},\ w \in \mathscr{W} \qquad \frac{\Gamma, B, !^{e}A, \Delta \to C}{\Gamma, !^{e}A, B, \Delta \to C}\ \frac{\Gamma, !^{e}A, B, \Delta \to C}{\Gamma, B, !^{e}A, \Delta \to C}\ !\mathsf{P},\ e \in \mathscr{E}$$

$$\frac{\Gamma, !^{c}A, \Phi, !^{c}A, \Delta \to C}{\Gamma, !^{c}A, \Phi, \Delta \to C}\ \frac{\Gamma, !^{c}A, \Phi, !^{c}A, \Delta \to C}{\Gamma, \Phi, !^{c}A, \Delta \to C}\ !\mathsf{NC},\ c \in \mathscr{C}$$

- Notice that contaction here appears in its **non-local form**, otherwise cut-elimination fails for $c \in \mathscr{C} - \mathscr{E}$
  [Bayu Surarso, Ono 1996; Kanovich, K., Nigam, Scedrov 2018].

Exponential and Subexponentials

- In $\mathscr{I}$, we designate three subsets, $\mathscr{W}$, $\mathscr{E}$, and $\mathscr{C}$, upward-closed under $\preceq$.
- These sets control which structural rules are allowed:

$$\frac{\Gamma, \Delta \to C}{\Gamma, !^{w}A, \Delta \to C}\ !\mathsf{W},\ w \in \mathscr{W} \qquad \frac{\Gamma, B, !^{e}A, \Delta \to C}{\Gamma, !^{e}A, B, \Delta \to C}\quad \frac{\Gamma, !^{e}A, B, \Delta \to C}{\Gamma, B, !^{e}A, \Delta \to C}\ !\mathsf{P},\ e \in \mathscr{E}$$

$$\frac{\Gamma, !^{c}A, \Phi, !^{c}A, \Delta \to C}{\Gamma, !^{c}A, \Phi, \Delta \to C}\quad \frac{\Gamma, !^{c}A, \Phi, !^{c}A, \Delta \to C}{\Gamma, \Phi, !^{c}A, \Delta \to C}\ !\mathsf{NC},\ c \in \mathscr{C}$$

- Notice that contaction here appears in its **non-local form**, otherwise cut-elimination fails for $c \in \mathscr{C} - \mathscr{E}$
  [Bayu Surarso, Ono 1996; Kanovich, K., Nigam, Scedrov 2018].
- Since !W and !NC simulate !P, we postulate $\mathscr{W} \cap \mathscr{C} \subseteq \mathscr{E}$.

Subexponentials in Lambek Grammars

- The permutation modality ($e \in \mathscr{E}$, $e \notin \mathscr{W} \cap \mathscr{C}$) is used for modelling **medial extraction**:

$N \,/\, CN,$	$CN,$	$(CN \setminus CN) \,/\, (S \,/\, !^{e}N),$	$N,$	$(N \setminus S) \,/\, N,$	$(N \setminus S) \setminus (N \setminus S)$	$\to N$
the	girl	whom	John	met	yesterday	

Subexponentials in Lambek Grammars

- The permutation modality ($e \in \mathscr{E}$, $e \notin \mathscr{W} \cap \mathscr{C}$) is used for modelling **medial extraction**:

$$\underset{\text{the}}{N \,/\, CN},\quad \underset{\text{girl}}{CN},\quad \underset{\text{whom}}{(CN \setminus CN) \,/\, (S \,/\, !^{e}N)},\quad \underset{\text{John}}{N},\quad \underset{\text{met}}{(N \setminus S) \,/\, N},\quad \underset{\text{yesterday}}{(N \setminus S) \setminus (N \setminus S)} \;\to N$$

- Contraction is used for modelling **multiple** (parasitic) extraction: “the paper that the reviewer of [] accepted [] without reading []”.

Subexponentials in Lambek Grammars

- The permutation modality ($e \in \mathscr{E}$, $e \notin \mathscr{W} \cap \mathscr{C}$) is used for modelling **medial extraction**:

$$\underset{\text{the}}{N \,/\, CN},\quad \underset{\text{girl}}{CN},\quad \underset{\text{whom}}{(CN \setminus CN) \,/\, (S \,/\, {!^{e}}N)},\quad \underset{\text{John}}{N},\quad \underset{\text{met}}{(N \setminus S) \,/\, N},\quad \underset{\text{yesterday}}{(N \setminus S) \setminus (N \setminus S)} \;\to N$$

- Contraction is used for modelling **multiple** (parasitic) extraction: “the paper that the reviewer of [] accepted [] without reading []”.
- Notice that weakening is linguistically inacceptable, so we have the so-called **relevant modality**: $r \in \mathscr{C} \cap \mathscr{E}$, $r \notin \mathscr{W}$.

Subexponentials in Lambek Grammars

- The permutation modality ($e \in \mathscr{E}$, $e \notin \mathscr{W} \cap \mathscr{C}$) is used for modelling **medial extraction**:

$N \,/\, CN$,	CN,	$(CN \setminus CN) \,/\, (S \,/\, !^{e}N)$,	N,	$(N \setminus S) \,/\, N$,	$(N \setminus S) \setminus (N \setminus S)$	$\to N$
the	girl	whom	John	met	yesterday	

- Contraction is used for modelling **multiple** (parasitic) extraction: “the paper that the reviewer of [] accepted [] without reading []”.
- Notice that weakening is linguistically inacceptable, so we have the so-called **relevant modality**: $r \in \mathscr{C} \cap \mathscr{E}$, $r \notin \mathscr{W}$.
- In real linguistic applications, even more sophisticated, semi-non-associative, systems of structural control are used [Morrill 1992; Moortgat 1996].

Decidability and Undecidability

- The derivability problem in **L** is algorithmically decidable, being NP-complete [Pentus 2006].

Decidability and Undecidability

- The derivability problem in **L** is algorithmically decidable, being NP-complete [Pentus 2006].
  - The one-division fragment of **L** is in P [Savateev 2009], and so is **L** with bounded depth of formulae [Pentus 2010].

Decidability and Undecidability

- The derivability problem in **L** is algorithmically decidable, being NP-complete [Pentus 2006].
  - The one-division fragment of **L** is in P [Savateev 2009], and so is **L** with bounded depth of formulae [Pentus 2010].
- Derivability in **MALC** is also decidable, being PSPACE-complete [Kanovich 1994; ...]

Decidability and Undecidability

- The derivability problem in **L** is algorithmically decidable, being NP-complete [Pentus 2006].
  - The one-division fragment of **L** is in P [Savateev 2009], and so is **L** with bounded depth of formulae [Pentus 2010].
- Derivability in **MALC** is also decidable, being PSPACE-complete [Kanovich 1994; ...]
- In contrast, **!MALC** is undecidable (Σ^0_1-complete).

Decidability and Undecidability

- The derivability problem in **L** is algorithmically decidable, being NP-complete [Pentus 2006].
  - The one-division fragment of **L** is in P [Savateev 2009], and so is **L** with bounded depth of formulae [Pentus 2010].
- Derivability in **MALC** is also decidable, being PSPACE-complete [Kanovich 1994; ...]
- In contrast, **!MALC** is undecidable (Σ^0_1-complete).
- The reason is undecidability of **entailment** from finite sets of hypotheses. The latter is encoded via **modalised deduction theorem**:

$$\to A_1, \dots, \to A_n \vdash_{\mathbf{MALC}} \Pi \to C \iff \vdash_{\mathbf{!MALC}} !A_1, \dots, !A_n, \Pi \to C.$$

Decidability and Undecidability

- Undecidability of entailment, even in the language including only $\cdot$ and $\backslash$, easily follows from undecidability in semi-Thue systems [Thue 1914; Markov 1947; Post 1947].

Decidability and Undecidability

- Undecidability of entailment, even in the language including only $\cdot$ and $\backslash$, easily follows from undecidability in semi-Thue systems [Thue 1914; Markov 1947; Post 1947].
- Buszkowski [1982] introduced a more sophisticated encoding, which shows undecidability of entailment in the one-division fragment of **L**.

Decidability and Undecidability

- Undecidability of entailment, even in the language including only $\cdot$ and $\backslash$, easily follows from undecidability in semi-Thue systems [Thue 1914; Markov 1947; Post 1947].
- Buszkowski [1982] introduced a more sophisticated encoding, which shows undecidability of entailment in the one-division fragment of **L**.
- Therefore, the fragment of **!MALC** with only $\backslash$ and ! is already undecidable.

Decidability and Undecidability

- Undecidability of entailment, even in the language including only $\cdot$ and $\backslash$, easily follows from undecidability in semi-Thue systems [Thue 1914; Markov 1947; Post 1947].
- Buszkowski [1982] introduced a more sophisticated encoding, which shows undecidability of entailment in the one-division fragment of **L**.
- Therefore, the fragment of **!MALC** with only $\backslash$ and ! is already undecidable.
- To compare, in the commutative case undecidability of linear logic uses additive disjunction [Lincoln, Mitchell, Scedrov, Shankar 1992].

Decidability and Undecidability

- Undecidability of entailment, even in the language including only $\cdot$ and $\backslash$, easily follows from undecidability in semi-Thue systems [Thue 1914; Markov 1947; Post 1947].
- Buszkowski [1982] introduced a more sophisticated encoding, which shows undecidability of entailment in the one-division fragment of **L**.
- Therefore, the fragment of **!MALC** with only $\backslash$ and ! is already undecidable.
- To compare, in the commutative case undecidability of linear logic uses additive disjunction [Lincoln, Mitchell, Scedrov, Shankar 1992].
- (Un)decidability of **MELL** (multiplicative-additive commutative linear logic) is a long-standing open question.

Undecidability with Subexponentials

- In fact, for undecidability it is sufficient to have a subexponential which allows non-local contraction ($c \in \mathscr{C}$) [Kanovich, K., Nigam, Scedrov 2018].

Undecidability with Subexponentials

- In fact, for undecidability it is sufficient to have a subexponential which allows non-local contraction ($c \in \mathscr{C}$) [Kanovich, K., Nigam, Scedrov 2018].
  - If no subexponential allows contraction, the system is decidable.

Undecidability with Subexponentials

- In fact, for undecidability it is sufficient to have a subexponential which allows non-local contraction ($c \in \mathscr{C}$) [Kanovich, K., Nigam, Scedrov 2018].
  - If no subexponential allows contraction, the system is decidable.
- With **local contraction**, cut-elimination can be restored in two ways. One either restricts the right rule

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{L} \qquad \frac{!A \to B}{!A \to !B}\ !_1\mathsf{R} \qquad \frac{\Gamma, !A, !A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{C}$$

or allows contraction of sequences of !-formulae:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !A, \Delta \to C}\ !\mathsf{L} \qquad \frac{!\Pi \to B}{!\Pi \to !B}\ !\mathsf{R} \qquad \frac{\Gamma, !\Pi, !\Pi, \Delta \to C}{\Gamma, !\Pi, \Delta \to C}\ !\mathsf{MC}$$

(Here if $\Pi = A_1, \ldots, A_n$, then $!\Pi = !A_1, \ldots, !A_n$.)

Undecidability with Subexponentials

- For the second system, undecidability was proved by Valinkin [2022], building upon the construction of [Chvalovský, Horčík 2016].

Undecidability with Subexponentials

- For the second system, undecidability was proved by Valinkin [2022], building upon the construction of [Chvalovský, Horčík 2016].
  - For the first system, (un)decidability remains an open question.

Undecidability with Subexponentials

- For the second system, undecidability was proved by Valinkin [2022], building upon the construction of [Chvalovský, Horčík 2016].
  - For the first system, (un)decidability remains an open question.
- It is interesting that local contraction subexponentials allow natural interpretations over algebras of binary relations, with completeness theorems [Valinkin, K. 2025].

Undecidability with Subexponentials

- For the second system, undecidability was proved by Valinkin [2022], building upon the construction of [Chvalovský, Horčík 2016].
  - For the first system, (un)decidability remains an open question.
- It is interesting that local contraction subexponentials allow natural interpretations over algebras of binary relations, with completeness theorems [Valinkin, K. 2025].
- Namely, ! is interpreted as a monotone idempotent operation on binary relations which maps any relation to its **dense** subrelation.

Undecidability with Subexponentials

- For the second system, undecidability was proved by Valinkin [2022], building upon the construction of [Chvalovský, Horčík 2016].
  - For the first system, (un)decidability remains an open question.
- It is interesting that local contraction subexponentials allow natural interpretations over algebras of binary relations, with completeness theorems [Valinkin, K. 2025].
- Namely, ! is interpreted as a monotone idempotent operation on binary relations which maps any relation to its **dense** subrelation.
- For the second system, ! should also commute with product: $!R \circ !S \subseteq !(R \circ S)$.

Kleene Star

- An even more intriguing operation on formal languages is the **Kleene star**:

$$A^* = \bigcup_{n=0}^{\infty} A^n.$$

Kleene Star

- An even more intriguing operation on formal languages is the **Kleene star**:

$$A^* = \bigcup_{n=0}^{\infty} A^n.$$

- On binary relations, this is reflexive-transitive closure.

Kleene Star

- An even more intriguing operation on formal languages is the **Kleene star**:

$$A^* = \bigcup_{n=0}^{\infty} A^n.$$

- On binary relations, this is reflexive-transitive closure.
- In abstract residuated lattices, there are two ways of defining the Kleene star—either as a least fixpoint:

$$a^* = \min\{b \mid \mathbf{1} \vee a \cdot b \preccurlyeq b\},$$

or as the supremum of powers:

$$a^* = \sup\{a^n \mid n \in \omega\}.$$

Kleene Star

- An even more intriguing operation on formal languages is the **Kleene star**:

$$A^* = \bigcup_{n=0}^{\infty} A^n.$$

- On binary relations, this is reflexive-transitive closure.
- In abstract residuated lattices, there are two ways of defining the Kleene star—either as a least fixpoint:

$$a^* = \min\{b \mid \mathbf{1} \vee a \cdot b \preccurlyeq b\},$$

  or as the supremum of powers:

$$a^* = \sup\{a^n \mid n \in \omega\}.$$

- This gives **action lattices** and their **$*$-continuous** subclass [Pratt 1991; Kozen 1994].

Kleene Star

- An even more intriguing operation on formal languages is the **Kleene star**:

$$A^* = \bigcup_{n=0}^{\infty} A^n.$$

- On binary relations, this is reflexive-transitive closure.
- In abstract residuated lattices, there are two ways of defining the Kleene star—either as a least fixpoint:

$$a^* = \min\{b \mid \mathbf{1} \vee a \cdot b \preccurlyeq b\},$$

  or as the supremum of powers:

$$a^* = \sup\{a^n \mid n \in \omega\}.$$

- This gives **action lattices** and their **$*$-continuous** subclass [Pratt 1991; Kozen 1994].
- **Positive iteration** ("Kleene plus") is defined: $A^+ = A \cdot A^*$.

(Infinitary) Action Logic

- The algebraic logic of $*$-continuous action lattices is **infinitary action logic** $\mathbf{ACT}_\omega$ [Palka 2007], obtained from **MALC** by the following rules:

$$\frac{\left(\Gamma, A^n, \Delta \to C\right)_{n=0}^{\infty}}{\Gamma, A^*, \Delta \to C}\ *\mathsf{L}_\omega \qquad \frac{\Pi_1 \to A \quad \ldots \quad \Pi_n \to A}{\Pi_1, \ldots, \Pi_n \to A^*}\ *\mathsf{R}_n$$

(Infinitary) Action Logic

- The algebraic logic of $*$-continuous action lattices is **infinitary action logic** $\mathbf{ACT}_\omega$ [Palka 2007], obtained from **MALC** by the following rules:

$$\frac{(\Gamma, A^n, \Delta \to C)_{n=0}^{\infty}}{\Gamma, A^*, \Delta \to C}\ *\mathsf{L}_\omega \qquad \frac{\Pi_1 \to A \quad \dots \quad \Pi_n \to A}{\Pi_1, \dots, \Pi_n \to A^*}\ *\mathsf{R}_n$$

- For the general class of action lattices, the logic is **action logic** **ACT**:

$$\frac{\to B \quad A, B \to B}{A^* \to B}\ *\mathsf{L}_{\mathrm{fix}} \qquad \frac{\Pi \to A \quad \Gamma, A, \Delta \to C}{\Gamma, \Pi, \Delta \to C}\ \mathsf{Cut}$$

$$\frac{}{\to A^*}\ *\mathsf{R}_0 \qquad \frac{\Pi \to A \quad \Delta \to A^*}{\Pi, \Delta \to A^*}\ *\mathsf{R}$$

(Infinitary) Action Logic

- The algebraic logic of $*$-continuous action lattices is **infinitary action logic** $\mathbf{ACT}_\omega$ [Palka 2007], obtained from **MALC** by the following rules:

$$\frac{(\Gamma, A^n, \Delta \to C)_{n=0}^{\infty}}{\Gamma, A^*, \Delta \to C}\ *\mathsf{L}_\omega \qquad \frac{\Pi_1 \to A \quad \dots \quad \Pi_n \to A}{\Pi_1, \dots, \Pi_n \to A^*}\ *\mathsf{R}_n$$

- For the general class of action lattices, the logic is **action logic** **ACT**:

$$\frac{\to B \quad A, B \to B}{A^* \to B}\ *\mathsf{L}_{\mathrm{fix}} \qquad \frac{\Pi \to A \quad \Gamma, A, \Delta \to C}{\Gamma, \Pi, \Delta \to C}\ \mathsf{Cut}$$

$$\frac{}{\to A^*}\ *\mathsf{R}_0 \qquad \frac{\Pi \to A \quad \Delta \to A^*}{\Pi, \Delta \to A^*}\ *\mathsf{R}$$

  - Notice that Cut is eliminable in $\mathbf{ACT}_\omega$, but not in **ACT**.

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.
  - The upper bound proved by Palka [2007].

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.
  - The upper bound proved by Palka [2007].
- A context-free grammar is total, if it generates all non-empty words over Σ.

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.
  - The upper bound proved by Palka [2007].
- A context-free grammar is total, if it generates all non-empty words over Σ.
- Lambek grammars have the same power as context-free ones [Bar-Hillel, Gaifman, Shamir 1960; Pentus 1992].

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.
  - The upper bound proved by Palka [2007].
- A context-free grammar is total, if it generates all non-empty words over Σ.
- Lambek grammars have the same power as context-free ones [Bar-Hillel, Gaifman, Shamir 1960; Pentus 1992].
- In a Lambek grammar, each letter $a \in \Sigma$ gets several formulae associated to it: $a \rhd A$.

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.
  - The upper bound proved by Palka [2007].
- A context-free grammar is total, if it generates all non-empty words over Σ.
- Lambek grammars have the same power as context-free ones [Bar-Hillel, Gaifman, Shamir 1960; Pentus 1992].
- In a Lambek grammar, each letter $a \in \Sigma$ gets several formulae associated to it: $a \rhd A$.
- Let

$$F_a = \bigwedge \{A \mid a \rhd A\} \qquad \text{and} \qquad E = \bigvee_{a \in \Sigma} F_a.$$

Undecidability of Action Logic

- Buszkowski [2007] proved Π^0_1-completeness of derivability in $\mathbf{ACT}_\omega$ by reducing the **totality problem** for context-free grammars.
  - The upper bound proved by Palka [2007].
- A context-free grammar is total, if it generates all non-empty words over Σ.
- Lambek grammars have the same power as context-free ones [Bar-Hillel, Gaifman, Shamir 1960; Pentus 1992].
- In a Lambek grammar, each letter $a \in \Sigma$ gets several formulae associated to it: $a \rhd A$.
- Let

$$F_a = \bigwedge \{A \mid a \rhd A\} \qquad \text{and} \qquad E = \bigvee_{a \in \Sigma} F_a.$$

- Now the grammar is total iff the following sequent is derivable in $\mathbf{ACT}_\omega$:

$$E^+ \to S.$$

Undecidability of Action Logic

- Π^0_1-completeness of the totality problem for context-free grammars is well-known.

Undecidability of Action Logic

- Π^0_1-completeness of the totality problem for context-free grammars is well-known.
- The construction is as follows: for a Turing machine $\mathtt{M}$ one constructs a grammar $G_{\mathtt{M}}$, which generates all non-empty words, except the halting protocol (if such exists).

Undecidability of Action Logic

- Π^0_1-completeness of the totality problem for context-free grammars is well-known.
- The construction is as follows: for a Turing machine $\mathtt{M}$ one constructs a grammar $G_{\mathtt{M}}$, which generates all non-empty words, except the halting protocol (if such exists).
- Thus, $G_{\mathtt{M}}$ is total iff $\mathtt{M}$ **does not halt**.

Undecidability of Action Logic

- Π^0_1-completeness of the totality problem for context-free grammars is well-known.
- The construction is as follows: for a Turing machine $\mathtt{M}$ one constructs a grammar $G_{\mathtt{M}}$, which generates all non-empty words, except the halting protocol (if such exists).
- Thus, $G_{\mathtt{M}}$ is total iff $\mathtt{M}$ **does not halt**.
- For proving undecidability of the weaker system **ACT**, we introduce the notion of **regular totality** for context-free grammars, which captures **circular runs** of Turing machines.

Undecidability of Action Logic

- Π^0_1-completeness of the totality problem for context-free grammars is well-known.
- The construction is as follows: for a Turing machine $\mathtt{M}$ one constructs a grammar $G_{\mathtt{M}}$, which generates all non-empty words, except the halting protocol (if such exists).
- Thus, $G_{\mathtt{M}}$ is total iff $\mathtt{M}$ **does not halt**.
- For proving undecidability of the weaker system **ACT**, we introduce the notion of **regular totality** for context-free grammars, which captures **circular runs** of Turing machines.
- A grammar G is **regularly total**, if it includes the following rules:

$$S \Rightarrow aYT \qquad T \Rightarrow a \qquad T \Rightarrow aT,$$

and the following holds: (1) $Y \Rightarrow^* w$ for each w with $|w| = n$; (2) $S \Rightarrow^* w$ for each w with $|w| \leqslant n + 1$.

Undecidability of Action Logic

- Regular totality means that, starting from some n, the grammar generates all non-empty words of the corresponding length in a regular fashion, using rules for T.

Undecidability of Action Logic

- Regular totality means that, starting from some n, the grammar generates all non-empty words of the corresponding length in a regular fashion, using rules for T.
- Let all machines have a designated "capturing" state q_c, in which the machine gets stuck.

Undecidability of Action Logic

- Regular totality means that, starting from some n, the grammar generates all non-empty words of the corresponding length in a regular fashion, using rules for T.
- Let all machines have a designated "capturing" state q_c, in which the machine gets stuck.
- We construct $G_{\mathtt{M}}$ in such way that if $\mathtt{M}$ gets into q_c (**looping**), then $G_{\mathtt{M}}$ is regularly total.

Undecidability of Action Logic

- Regular totality means that, starting from some n, the grammar generates all non-empty words of the corresponding length in a regular fashion, using rules for T.
- Let all machines have a designated "capturing" state q_c, in which the machine gets stuck.
- We construct $G_{\mathtt{M}}$ in such way that if $\mathtt{M}$ gets into q_c (**looping**), then $G_{\mathtt{M}}$ is regularly total.
- For a regularly total $G_{\mathtt{M}}$, the sequent $E^+ \to S$ is provable already in **ACT**.

Undecidability of Action Logic

- Regular totality means that, starting from some n, the grammar generates all non-empty words of the corresponding length in a regular fashion, using rules for T.
- Let all machines have a designated "capturing" state q_c, in which the machine gets stuck.
- We construct $G_{\mathtt{M}}$ in such way that if $\mathtt{M}$ gets into q_c (**looping**), then $G_{\mathtt{M}}$ is regularly total.
- For a regularly total $G_{\mathtt{M}}$, the sequent $E^+ \to S$ is provable already in **ACT**.
- Unfortunately, the converse does not hold: there could be cases where $E^+ \to S$ is derivable in **ACT**, but $\mathtt{M}$ does not loop.

Effective Inseparability

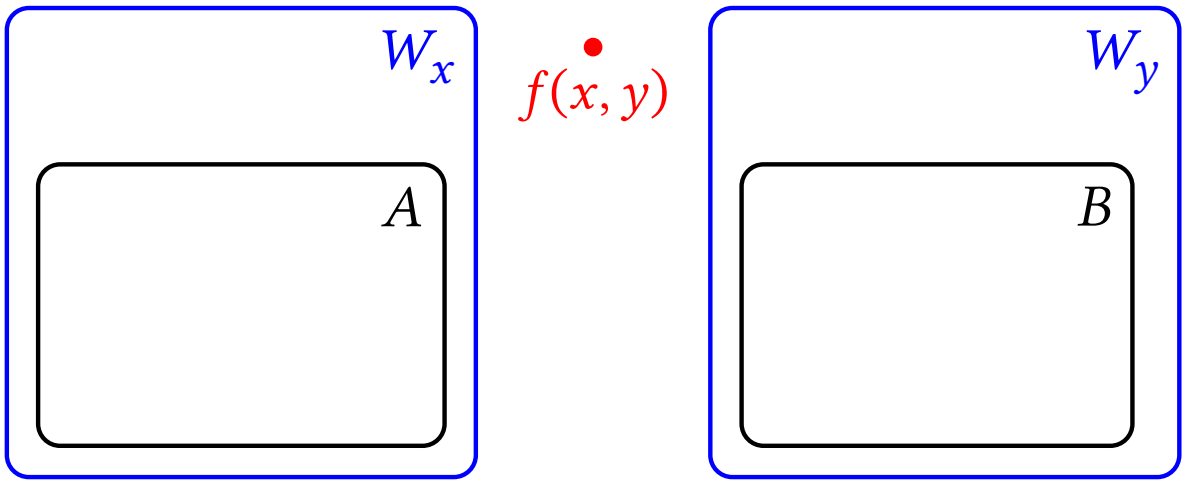
- In order to handle this, the indirect technique of **effective inseparability** is used.

Effective Inseparability

- In order to handle this, the indirect technique of **effective inseparability** is used.
- Let W_x be the r.e. set of index x.

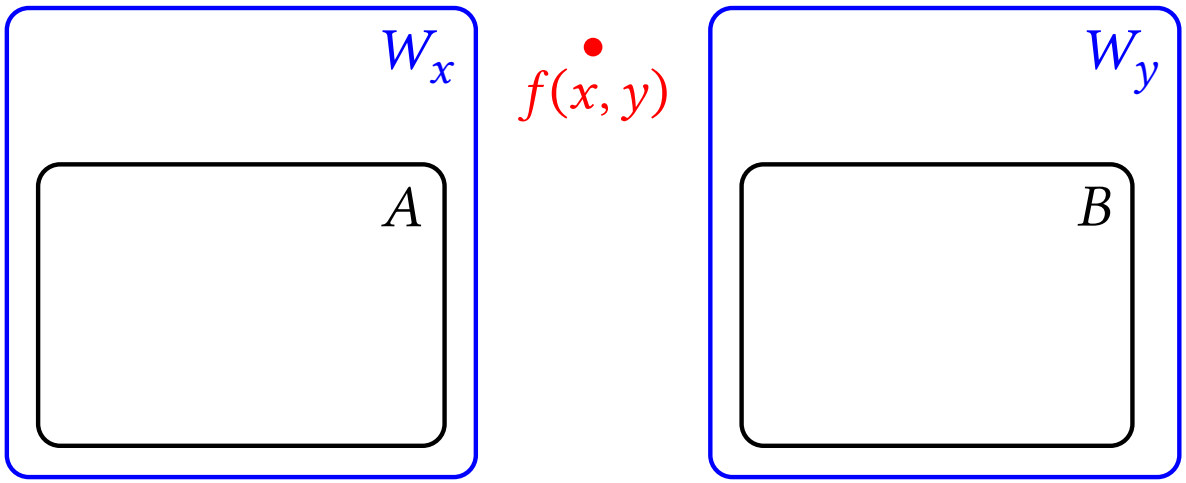
Effective Inseparability

- In order to handle this, the indirect technique of **effective inseparability** is used.
- Let W_x be the r.e. set of index x.
- Two disjoint sets $A, B \subseteq \mathbb{N}$ are effectively inseparable, if there is a computable function $f(x, y)$ with the following property.
  If $W_x \supseteq A$, $W_y \supseteq B$, and $W_x \cap W_y = \emptyset$, then $f(x, y) \notin W_x \cup W_y$.



Effective Inseparability

- In order to handle this, the indirect technique of **effective inseparability** is used.
- Let W_x be the r.e. set of index x.
- Two disjoint sets $A, B \subseteq \mathbb{N}$ are effectively inseparable, if there is a computable function $f(x, y)$ with the following property. If $W_x \supseteq A$, $W_y \supseteq B$, and $W_x \cap W_y = \emptyset$, then $f(x, y) \notin W_x \cup W_y$.



- f effectively prevents A and B from being separated by a decidable set.

Effective Inseparability

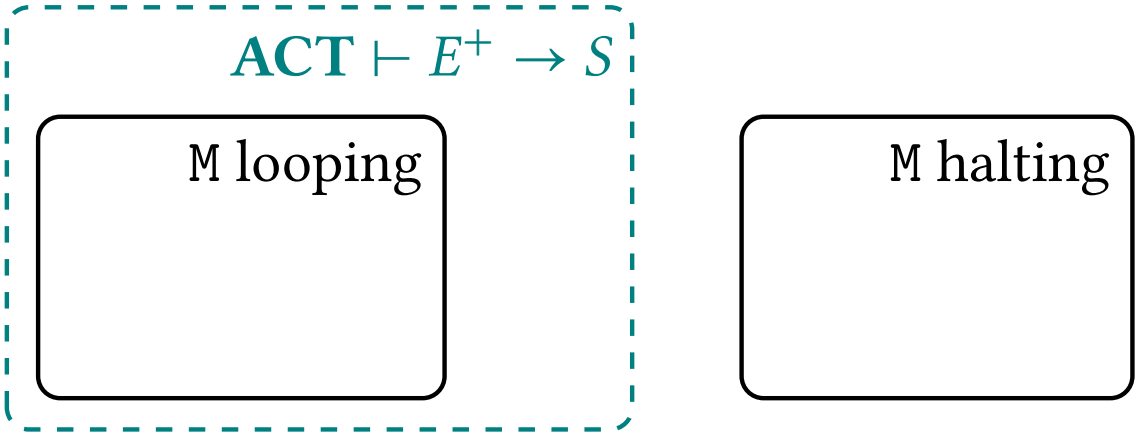
- Let A be the set of (codes of) machines which loop, and B the set of machines which halt.

Effective Inseparability

- Let A be the set of (codes of) machines which loop, and B the set of machines which halt.
- Folklore fact: A and B are effectively inseparable.

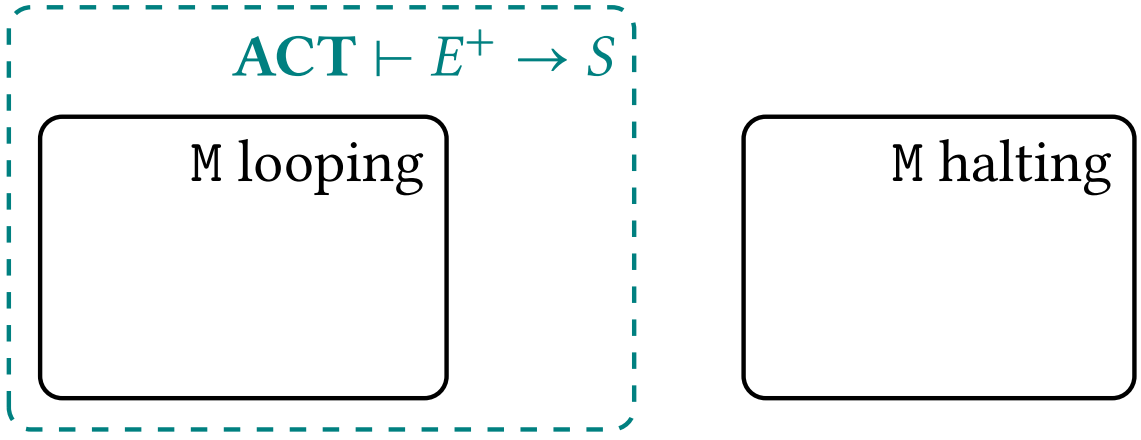
Effective Inseparability

- Let A be the set of (codes of) machines which loop, and B the set of machines which halt.
- Folklore fact: A and B are effectively inseparable.
- Therefore, the set of machines for which $E^+ \to S$ is provable in **ACT**, is undecidable.



Effective Inseparability

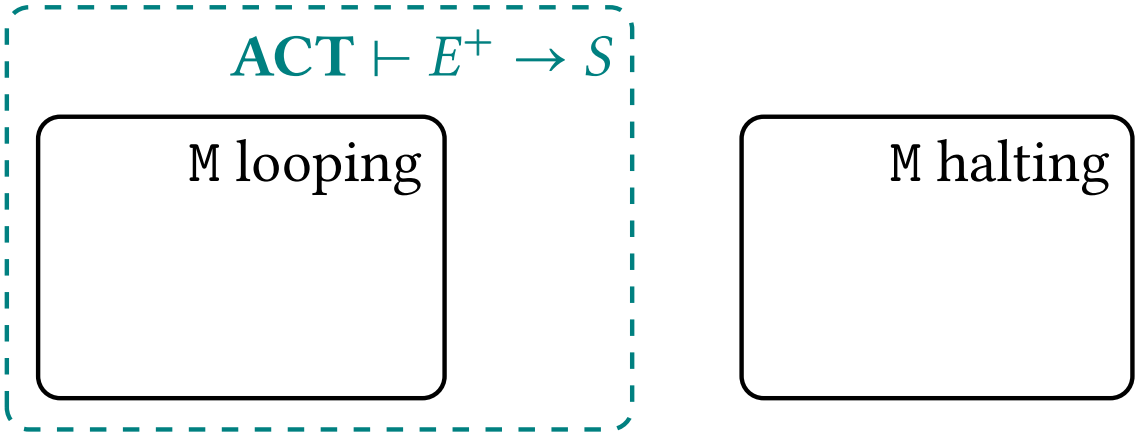
- Let A be the set of (codes of) machines which loop, and B the set of machines which halt.
- Folklore fact: A and B are effectively inseparable.
- Therefore, the set of machines for which $E^+ \to S$ is provable in **ACT**, is undecidable.



- Moreover, as follows from [Myhill 1955], any r.e. set separating two r.e. effectively inseparable sets, is Σ^0_1-complete.

Effective Inseparability

- Let A be the set of (codes of) machines which loop, and B the set of machines which halt.
- Folklore fact: A and B are effectively inseparable.
- Therefore, the set of machines for which $E^+ \to S$ is provable in **ACT**, is undecidable.



- Moreover, as follows from [Myhill 1955], any r.e. set separating two r.e. effectively inseparable sets, is Σ^0_1-complete.
- This proves Σ^0_1-completeness of **ACT** [K. 2021].

Exponential and Kleene Star

- Now let us put things together and introduce $\mathbf{!ACT}_\omega$, a system including both the exponential and the Kleene star (with infinitary axiomatisation).

Exponential and Kleene Star

- Now let us put things together and introduce $\mathbf{!ACT}_\omega$, a system including both the exponential and the Kleene star (with infinitary axiomatisation).
- It turns out that complexity grows dramatically, and $\mathbf{!ACT}_\omega$ is Π^1_1-complete [K., Speranski 2022].

Exponential and Kleene Star

- Now let us put things together and introduce $\mathbf{!ACT}_\omega$, a system including both the exponential and the Kleene star (with infinitary axiomatisation).
- It turns out that complexity grows dramatically, and $\mathbf{!ACT}_\omega$ is Π^1_1-complete [K., Speranski 2022].
- The upper bound here comes from a very general argument for infinitary calculi of the given form.

Exponential and Kleene Star

- Now let us put things together and introduce $\mathbf{!ACT}_\omega$, a system including both the exponential and the Kleene star (with infinitary axiomatisation).
- It turns out that complexity grows dramatically, and $\mathbf{!ACT}_\omega$ is Π^1_1-complete [K., Speranski 2022].
- The upper bound here comes from a very general argument for infinitary calculi of the given form.
- For the lower bound, we use Kozen's result on Π^1_1-completeness for reasoning from hypotheses in Kleene algebra—in the language of * and $\cdot$, without residuals [Kozen 2002].

Exponential and Kleene Star

- Now let us put things together and introduce $\mathbf{!ACT}_\omega$, a system including both the exponential and the Kleene star (with infinitary axiomatisation).
- It turns out that complexity grows dramatically, and $\mathbf{!ACT}_\omega$ is Π^1_1-complete [K., Speranski 2022].
- The upper bound here comes from a very general argument for infinitary calculi of the given form.
- For the lower bound, we use Kozen's result on Π^1_1-completeness for reasoning from hypotheses in Kleene algebra—in the language of * and $\cdot$, without residuals [Kozen 2002].
- Kozen encodes the **well-foundedness problem** for a recursively defined relation $R \subseteq \mathbb{N} \times \mathbb{N}$.

Exponential and Kleene Star

- Now let us put things together and introduce $\mathbf{!ACT}_\omega$, a system including both the exponential and the Kleene star (with infinitary axiomatisation).
- It turns out that complexity grows dramatically, and $\mathbf{!ACT}_\omega$ is Π^1_1-complete [K., Speranski 2022].
- The upper bound here comes from a very general argument for infinitary calculi of the given form.
- For the lower bound, we use Kozen's result on Π^1_1-completeness for reasoning from hypotheses in Kleene algebra—in the language of * and $\cdot$, without residuals [Kozen 2002].
- Kozen encodes the **well-foundedness problem** for a recursively defined relation $R \subseteq \mathbb{N} \times \mathbb{N}$.
- Next we, again, internalise reasoning from hypotheses into $\mathbf{!ACT}_\omega$ using modalised deduction theorem.

Exponential and Kleene Star

- Kozen's construction works as follows. The Turing machine computing R on input (x, y) starts at the configuration $s\, a^y\, b^x$.

Exponential and Kleene Star

- Kozen's construction works as follows. The Turing machine computing R on input (x, y) starts at the configuration $s\, a^y\, b^x$.
- If the answer is "no," $(x, y) \notin R$, then it ends in configuration r ("reject").

Exponential and Kleene Star

- Kozen's construction works as follows. The Turing machine computing R on input (x, y) starts at the configuration $s\, a^y\, b^x$.
- If the answer is "no," $(x, y) \notin R$, then it ends in configuration r ("reject").
- Otherwise, it yields $t\, b^y$.

Exponential and Kleene Star

- Kozen's construction works as follows. The Turing machine computing R on input (x, y) starts at the configuration $s\, a^y\, b^x$.
- If the answer is "no," $(x, y) \notin R$, then it ends in configuration r ("reject").
- Otherwise, it yields $t\, b^y$.
- The machine's commands are presented as a semi-Thue system, whose rules are translated into Lambek hypotheses:
  $x_1, \ldots, x_m \to y_1 \cdot \ldots \cdot y_k$.

Exponential and Kleene Star

- Kozen's construction works as follows. The Turing machine computing R on input (x, y) starts at the configuration $s\,a^y\,b^x$.
- If the answer is "no," $(x, y) \notin R$, then it ends in configuration r ("reject").
- Otherwise, it yields $t\,b^y$.
- The machine's commands are presented as a semi-Thue system, whose rules are translated into Lambek hypotheses:

  $x_1, \ldots, x_m \to y_1 \cdot \ldots \cdot y_k$.
- We add an "infinitary semi-Thue rule" $t \Rightarrow s\,a^*$, which is responsible for restarting the computation on input (y, z), for all z.

Exponential and Kleene Star

- Kozen's construction works as follows. The Turing machine computing R on input (x, y) starts at the configuration $s\,a^y\,b^x$.
- If the answer is "no," $(x, y) \notin R$, then it ends in configuration r ("reject").
- Otherwise, it yields $t\,b^y$.
- The machine's commands are presented as a semi-Thue system, whose rules are translated into Lambek hypotheses:
  $x_1, \ldots, x_m \to y_1 \cdot \ldots \cdot y_k$.
- We add an "infinitary semi-Thue rule" $t \Rightarrow s\,a^*$, which is responsible for restarting the computation on input (y, z), for all z.
- Well-foundedness of R is equivalent to well-foundedness (e.g., correctness) of the proof of $t, b^* \to r$ from the given set of hypotheses in $\mathbf{ACT}_\omega$.

Iterative Divisions

- By combining Kozen's reasoning and the construction from [Buszkowski 1982], we can trade product and iteration for division and **iterative division**—a compound connective of the form $A \setminus\setminus B = A^* \setminus B$ [Kanovich, K., Scedrov 2025].

Iterative Divisions

- By combining Kozen's reasoning and the construction from [Buszkowski 1982], we can trade product and iteration for division and **iterative division**—a compound connective of the form $A \setminus\setminus B = A^* \setminus B$ [Kanovich, K., Scedrov 2025].
  - Iterative divisions were introduced by Sedlár [2019], in a non-associative setting.

Iterative Divisions

- By combining Kozen's reasoning and the construction from [Buszkowski 1982], we can trade product and iteration for division and **iterative division**—a compound connective of the form $A \setminus\setminus B = A^* \setminus B$ [Kanovich, K., Scedrov 2025].
  - Iterative divisions were introduced by Sedlár [2019], in a non-associative setting.
- The advantage of iterative divisions over Kleene star is that in the language of $\setminus, /, \setminus\setminus, //, \wedge$ we have completeness, both for language and relational semantics [K., Ryzhkova 2020; K. 2024].

Iterative Divisions

- By combining Kozen's reasoning and the construction from [Buszkowski 1982], we can trade product and iteration for division and **iterative division**—a compound connective of the form $A \setminus\setminus B = A^* \setminus B$ [Kanovich, K., Scedrov 2025].
  - Iterative divisions were introduced by Sedlár [2019], in a non-associative setting.
- The advantage of iterative divisions over Kleene star is that in the language of $\setminus, /, \setminus\setminus, //, \wedge$ we have completeness, both for language and relational semantics [K., Ryzhkova 2020; K. 2024].
- We have obtained Π^1_1-completeness for derivability in the $(\setminus, \setminus\setminus, !)$-fragment of $\mathbf{!ACT}_\omega$ and for entailment from finite sets of hypotheses in the $(\setminus, \setminus\setminus)$-fragment of $\mathbf{ACT}_\omega$.

Iterative Divisions

- By combining Kozen's reasoning and the construction from [Buszkowski 1982], we can trade product and iteration for division and **iterative division**—a compound connective of the form $A \setminus\setminus B = A^* \setminus B$ [Kanovich, K., Scedrov 2025].
  - Iterative divisions were introduced by Sedlár [2019], in a non-associative setting.
- The advantage of iterative divisions over Kleene star is that in the language of $\setminus, /, \setminus\setminus, //, \wedge$ we have completeness, both for language and relational semantics [K., Ryzhkova 2020; K. 2024].
- We have obtained Π^1_1-completeness for derivability in the $(\setminus, \setminus\setminus, !)$-fragment of $\mathbf{!ACT}_\omega$ and for entailment from finite sets of hypotheses in the $(\setminus, \setminus\setminus)$-fragment of $\mathbf{ACT}_\omega$.
- We shall need this result later on.

Fragments of $\mathbf{!ACT}_\omega$

- A very interesting fragment of $\mathbf{!ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.

Fragments of $!\mathbf{ACT}_\omega$

- A very interesting fragment of $!\mathbf{ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.
- For this fragment, we manage to prove an ω^ω upper bound on its **closure ordinal**.

Fragments of $\mathbf{!ACT}_\omega$

- A very interesting fragment of $\mathbf{!ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.
- For this fragment, we manage to prove an ω^ω upper bound on its **closure ordinal**.
  - In an infinitary calculus, the set of derivable sequents is the limit of sets S_α, "the sequents derived at rank α." The closure ordinal is the supremum of ranks.

Fragments of $\mathbf{!ACT}_\omega$

- A very interesting fragment of $\mathbf{!ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.
- For this fragment, we manage to prove an ω^ω upper bound on its **closure ordinal**.
  - In an infinitary calculus, the set of derivable sequents is the limit of sets S_α, "the sequents derived at rank α." The closure ordinal is the supremum of ranks.
- For each sequent we recursively compute its **ordinal measure**, which is multiplied by ω and increased by 1 at each occurrence of Kleene star (for binary connectives we take the natural sum).

Fragments of $\mathbf{!ACT}_\omega$

- A very interesting fragment of $\mathbf{!ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.
- For this fragment, we manage to prove an ω^ω upper bound on its **closure ordinal**.
  - In an infinitary calculus, the set of derivable sequents is the limit of sets S_α, "the sequents derived at rank α." The closure ordinal is the supremum of ranks.
- For each sequent we recursively compute its **ordinal measure**, which is multiplied by ω and increased by 1 at each occurrence of Kleene star (for binary connectives we take the natural sum).
- Ordinal measures are $< \omega^\omega$, and they control the closure ordinal.

Fragments of $!\mathbf{ACT}_\omega$

- A very interesting fragment of $!\mathbf{ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.
- For this fragment, we manage to prove an ω^ω upper bound on its **closure ordinal**.
  - In an infinitary calculus, the set of derivable sequents is the limit of sets S_α, "the sequents derived at rank α." The closure ordinal is the supremum of ranks.
- For each sequent we recursively compute its **ordinal measure**, which is multiplied by ω and increased by 1 at each occurrence of Kleene star (for binary connectives we take the natural sum).
- Ordinal measures are $< \omega^\omega$, and they control the closure ordinal.
- By $H(\alpha)$ (for $\alpha \leq \omega^\omega$) we denote the α iteration of Turing jump.

Fragments of $\mathbf{!ACT}_\omega$

- A very interesting fragment of $\mathbf{!ACT}_\omega$ is the one where the Kleene star is not allowed to be used under the exponential.
- For this fragment, we manage to prove an ω^ω upper bound on its **closure ordinal**.
  - In an infinitary calculus, the set of derivable sequents is the limit of sets S_α, "the sequents derived at rank α." The closure ordinal is the supremum of ranks.
- For each sequent we recursively compute its **ordinal measure**, which is multiplied by ω and increased by 1 at each occurrence of Kleene star (for binary connectives we take the natural sum).
- Ordinal measures are $< \omega^\omega$, and they control the closure ordinal.
- By $H(\alpha)$ (for $\alpha \leq \omega^\omega$) we denote the α iteration of Turing jump.
  - In particular, $H(n)$ is the standard Σ^0_n-complete set.

Fragments of $!\mathbf{ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.

Fragments of $!\mathbf{ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.
  - If $\alpha = \beta \cdot \omega + k$, then $\alpha' = \beta \cdot \omega + 2k + 1$.

Fragments of $\mathbf{!ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.
  - If $\alpha = \beta \cdot \omega + k$, then $\alpha' = \beta \cdot \omega + 2k + 1$.
- This gives a reduction from derivability in the given fragment of $\mathbf{!ACT}_\omega$ to $H(\omega^\omega)$, i.e., establishes a $\Sigma^0_{\omega^\omega}$ upper bound [K., Pshenitsyn, Speranski 2025].

Fragments of $\mathbf{!ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.
  - If $\alpha = \beta \cdot \omega + k$, then $\alpha' = \beta \cdot \omega + 2k + 1$.
- This gives a reduction from derivability in the given fragment of $\mathbf{!ACT}_\omega$ to $H(\omega^\omega)$, i.e., establishes a $\Sigma^0_{\omega^\omega}$ upper bound [K., Pshenitsyn, Speranski 2025].
  - Our notation here is not entirely standard, as usually (see [Rogers 1967]) on limit steps an extra Turing jump is added.

Fragments of $\mathbf{!ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.
  - If $\alpha = \beta \cdot \omega + k$, then $\alpha' = \beta \cdot \omega + 2k + 1$.
- This gives a reduction from derivability in the given fragment of $\mathbf{!ACT}_\omega$ to $H(\omega^\omega)$, i.e., establishes a $\Sigma^0_{\omega^\omega}$ upper bound [K., Pshenitsyn, Speranski 2025].
  - Our notation here is not entirely standard, as usually (see [Rogers 1967]) on limit steps an extra Turing jump is added.
- Astonishingly, the $\Sigma^0_{\omega^\omega}$ complexity estimation is exact!

Fragments of $\mathbf{!ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.
  - If $\alpha = \beta \cdot \omega + k$, then $\alpha' = \beta \cdot \omega + 2k + 1$.
- This gives a reduction from derivability in the given fragment of $\mathbf{!ACT}_\omega$ to $H(\omega^\omega)$, i.e., establishes a $\Sigma^0_{\omega^\omega}$ upper bound [K., Pshenitsyn, Speranski 2025].
  - Our notation here is not entirely standard, as usually (see [Rogers 1967]) on limit steps an extra Turing jump is added.
- Astonishingly, the $\Sigma^0_{\omega^\omega}$ complexity estimation is exact!
- The lower bound is proved via encoding truth of computable infinitary propositional formulae of rank $< \omega^\omega$ [Ash, Knight 2000; Montalbán 2022].

Fragments of $\mathbf{!ACT}_\omega$

- By **computable transfinite induction** we construct reductions from derivability of sequents of measure $\leq \alpha$ (with the given restriction) to $H(\alpha')$, where α' is an ordinal close to α.
  - If $\alpha = \beta \cdot \omega + k$, then $\alpha' = \beta \cdot \omega + 2k + 1$.
- This gives a reduction from derivability in the given fragment of $\mathbf{!ACT}_\omega$ to $H(\omega^\omega)$, i.e., establishes a $\Sigma^0_{\omega^\omega}$ upper bound [K., Pshenitsyn, Speranski 2025].
  - Our notation here is not entirely standard, as usually (see [Rogers 1967]) on limit steps an extra Turing jump is added.
- Astonishingly, the $\Sigma^0_{\omega^\omega}$ complexity estimation is exact!
- The lower bound is proved via encoding truth of computable infinitary propositional formulae of rank $< \omega^\omega$ [Ash, Knight 2000; Montalbán 2022].
- As a matter of fact, a similar technique applied to $\mathbf{ACT}_\omega$ computes Π^0_1-formulae defining derivability for sequents of given ordinal measure. This gives an alternative Π^0_1 upper bound proof.

Theories of Language and Relational Models

- Constants **0** and **1** allow simulating the exponential modality, in models over algebras of formal languages and algebras of binary relations [Kanovich, K., Scedrov 2025].

Theories of Language and Relational Models

- Constants **0** and **1** allow simulating the exponential modality, in models over algebras of formal languages and algebras of binary relations [Kanovich, K., Scedrov 2025].
- This becomes possible, since in the presence of constants completeness does not hold.

Theories of Language and Relational Models

- Constants **0** and **1** allow simulating the exponential modality, in models over algebras of formal languages and algebras of binary relations [Kanovich, K., Scedrov 2025].
- This becomes possible, since in the presence of constants completeness does not hold.
- In models on languages, the construction $\mathbf{1} \wedge A$ is $\emptyset$ (“zero”), if $\rightarrow A$ is not true in the model, and $\{\varepsilon\}$ (“unit”) otherwise.

Theories of Language and Relational Models

- Constants $\mathbf{0}$ and $\mathbf{1}$ allow simulating the exponential modality, in models over algebras of formal languages and algebras of binary relations [Kanovich, K., Scedrov 2025].
- This becomes possible, since in the presence of constants completeness does not hold.
- In models on languages, the construction $\mathbf{1} \wedge A$ is $\emptyset$ ("zero"), if $\rightarrow A$ is not true in the model, and $\{\varepsilon\}$ ("unit") otherwise.
- This yields modalised deduction theorem: $\rightarrow A \vDash \rightarrow B$ iff $\vDash \mathbf{1} \wedge A \rightarrow B$.

Theories of Language and Relational Models

- Constants $\mathbf{0}$ and $\mathbf{1}$ allow simulating the exponential modality, in models over algebras of formal languages and algebras of binary relations [Kanovich, K., Scedrov 2025].
- This becomes possible, since in the presence of constants completeness does not hold.
- In models on languages, the construction $\mathbf{1} \wedge A$ is $\emptyset$ ("zero"), if $\to A$ is not true in the model, and $\{\varepsilon\}$ ("unit") otherwise.
- This yields modalised deduction theorem: $\to A \vDash \to B$ iff $\vDash \mathbf{1} \wedge A \to B$.
- From this, using strong completeness and complexity results, one derives Π^1_1-completeness for the equational theory of formal language algebras, in the language of $\backslash$, $\backslash\backslash$, $\wedge$, $\mathbf{1}$.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.
- Again, we have modalised deduction theorem: $\rightarrow A \models \rightarrow B$ iff $\models \widehat{A} \rightarrow B$.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.
- Again, we have modalised deduction theorem: $\rightarrow A \models \rightarrow B$ iff $\models \widehat{A} \rightarrow B$.
- Any finite set of hypotheses can be encoded as one of the form $\rightarrow A$.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.
- Again, we have modalised deduction theorem: $\rightarrow A \models \rightarrow B$ iff $\models \widehat{A} \rightarrow B$.
- Any finite set of hypotheses can be encoded as one of the form $\rightarrow A$.
- Now we prove Π^1_1-hardness. For A, B in the language of $\backslash$, $\backslash\backslash$, $\wedge$, we have

$$\rightarrow A \vdash \rightarrow B \iff \rightarrow A \models \rightarrow B \iff \models \widehat{A} \rightarrow B.$$

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.
- Again, we have modalised deduction theorem: $\rightarrow A \models \rightarrow B$ iff $\models \widehat{A} \rightarrow B$.
- Any finite set of hypotheses can be encoded as one of the form $\rightarrow A$.
- Now we prove Π^1_1-hardness. For A, B in the language of $\backslash$, $\backslash\backslash$, $\wedge$, we have

$$\rightarrow A \vdash \rightarrow B \iff \rightarrow A \models \rightarrow B \iff \models \widehat{A} \rightarrow B.$$

- The upper Π^1_1 bounds come from a general Löwenheim–Skolem style reasoning.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.
- Again, we have modalised deduction theorem: $\rightarrow A \models \rightarrow B$ iff $\models \widehat{A} \rightarrow B$.
- Any finite set of hypotheses can be encoded as one of the form $\rightarrow A$.
- Now we prove Π^1_1-hardness. For A, B in the language of $\backslash$, $\backslash\backslash$, $\wedge$, we have

$$\rightarrow A \vdash \rightarrow B \iff \rightarrow A \models \rightarrow B \iff \models \widehat{A} \rightarrow B.$$

- The upper Π^1_1 bounds come from a general Löwenheim–Skolem style reasoning.
- Thus, we get Π^1_1-completeness for the equational theory of algebras of binary relations, in the language of $\backslash$, $\backslash\backslash$, $\wedge$, $\mathbf{0}$, $\mathbf{1}$.

Theories of Language and Relational Models

- For algebras of binary relations, the construction is a bit more involved.
- As a substitute for $!A$, we take $\widehat{A} = \mathbf{1} \wedge (\mathbf{0} \,/\, (\mathbf{0} \,/\, (\mathbf{1} \wedge A)))$.
- Again, we have modalised deduction theorem: $\to A \models \to B$ iff $\models \widehat{A} \to B$.
- Any finite set of hypotheses can be encoded as one of the form $\to A$.
- Now we prove Π^1_1-hardness. For A, B in the language of $\backslash$, $\backslash\backslash$, $\wedge$, we have

$$\to A \vdash \to B \iff \to A \models \to B \iff \models \widehat{A} \to B.$$

- The upper Π^1_1 bounds come from a general Löwenheim–Skolem style reasoning.
- Thus, we get Π^1_1-completeness for the equational theory of algebras of binary relations, in the language of $\backslash$, $\backslash\backslash$, $\wedge$, $\mathbf{0}$, $\mathbf{1}$.
  - This solves an open problem by Buszkowski [1982], who conjectures Π^0_1-completeness.

Non-Associative Lambek Calculus

- The non-associative Lambek calculus **NL** goes even more substructural and abandons the implicit rule of associativity.

Non-Associative Lambek Calculus

- The non-associative Lambek calculus **NL** goes even more substructural and abandons the implicit rule of associativity.
- Left-hand sides of sequents are now bracketed structures (binary trees) of formulae.

Non-Associative Lambek Calculus

- The non-associative Lambek calculus **NL** goes even more substructural and abandons the implicit rule of associativity.
- Left-hand sides of sequents are now bracketed structures (binary trees) of formulae.
- The inference rules are reformulated accordingly:

$$\frac{\Gamma[(A, B)] \to C}{\Gamma[A \cdot B] \to C} \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{(\Gamma, \Delta) \to A \cdot B} \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma(B) \to C}{\Gamma[(\Pi, A \setminus B)] \to C} \setminus\mathsf{L} \qquad \frac{(\Pi, A) \to B}{\Pi \to A \setminus B} \setminus\mathsf{R}$$

Non-Associative Lambek Calculus

- The non-associative Lambek calculus **NL** goes even more substructural and abandons the implicit rule of associativity.
- Left-hand sides of sequents are now bracketed structures (binary trees) of formulae.
- The inference rules are reformulated accordingly:

$$\frac{\Gamma[(A, B)] \to C}{\Gamma[A \cdot B] \to C} \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{(\Gamma, \Delta) \to A \cdot B} \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma(B) \to C}{\Gamma[(\Pi, A \setminus B)] \to C} \setminus\mathsf{L} \qquad \frac{(\Pi, A) \to B}{\Pi \to A \setminus B} \setminus\mathsf{R}$$

- Hopefully we hear more on non-associative systems in the talk by Michael Moortgat.

Non-Associative Lambek Calculus

- The non-associative Lambek calculus **NL** goes even more substructural and abandons the implicit rule of associativity.
- Left-hand sides of sequents are now bracketed structures (binary trees) of formulae.
- The inference rules are reformulated accordingly:

$$\frac{\Gamma[(A, B)] \to C}{\Gamma[A \cdot B] \to C} \cdot\mathsf{L} \qquad \frac{\Gamma \to A \quad \Delta \to B}{(\Gamma, \Delta) \to A \cdot B} \cdot\mathsf{R}$$

$$\frac{\Pi \to A \quad \Gamma(B) \to C}{\Gamma[(\Pi, A \setminus B)] \to C} \setminus\mathsf{L} \qquad \frac{(\Pi, A) \to B}{\Pi \to A \setminus B} \setminus\mathsf{R}$$

- Hopefully we hear more on non-associative systems in the talk by Michael Moortgat.
- In **NL** without additives and in the **distributive** version of **NL** with additives, entailment from finite sets of hypotheses is **decidable** [Buszkowski, Farulewski 2009].

Non-Associativity and Distributivity

- In multiplicative-additive **NL** without distributivity, it is **undecidable** [Chvalovský 2015].

Non-Associativity and Distributivity

- In multiplicative-additive **NL** without distributivity, it is **undecidable** [Chvalovský 2015].
  - This allows proving some undecidability results for subexponential extensions of **NL** [Blaisdell, Kanovich, K., Pimentel, Scedrov 2022].

Non-Associativity and Distributivity

- In multiplicative-additive **NL** without distributivity, it is **undecidable** [Chvalovský 2015].
  - This allows proving some undecidability results for subexponential extensions of **NL** [Blaisdell, Kanovich, K., Pimentel, Scedrov 2022].
- Sedlár [2019] shows decidability of **NL** with additives and iterative divisions, again in the presence of distributivity.

Non-Associativity and Distributivity

- In multiplicative-additive **NL** without distributivity, it is **undecidable** [Chvalovský 2015].
  - This allows proving some undecidability results for subexponential extensions of **NL** [Blaisdell, Kanovich, K., Pimentel, Scedrov 2022].
- Sedlár [2019] shows decidability of **NL** with additives and iterative divisions, again in the presence of distributivity.
- In the associative setting, in contrast, distributivity is usually an obstacle.

Non-Associativity and Distributivity

- In multiplicative-additive **NL** without distributivity, it is **undecidable** [Chvalovský 2015].
  - This allows proving some undecidability results for subexponential extensions of **NL** [Blaisdell, Kanovich, K., Pimentel, Scedrov 2022].
- Sedlár [2019] shows decidability of **NL** with additives and iterative divisions, again in the presence of distributivity.
- In the associative setting, in contrast, distributivity is usually an obstacle.
- In particular, we do not know exact complexity of distributive versions of **MALC** and $\mathbf{ACT}_\omega$.

Non-Associativity and Distributivity

- In multiplicative-additive **NL** without distributivity, it is **undecidable** [Chvalovský 2015].
  - This allows proving some undecidability results for subexponential extensions of **NL** [Blaisdell, Kanovich, K., Pimentel, Scedrov 2022].
- Sedlár [2019] shows decidability of **NL** with additives and iterative divisions, again in the presence of distributivity.
- In the associative setting, in contrast, distributivity is usually an obstacle.
- In particular, we do not know exact complexity of distributive versions of **MALC** and $\mathbf{ACT}_\omega$.
- Decidability of distributive **MALC** shown by Kozak [2009].

Thanks

Thanks!*

- This talk is partially based on joint work with Eben Blaisdell, Max Kanovich, Glyn Morrill, Vivek Nigam, Elaine Pimentel, Tikhon Pshenitsyn, Andre Scedrov, and Stanislav Speranski, all of whom the author is indebted to.
- The work of S. K. was performed at Steklov International Mathematical Centre and supported by the Ministry of Science and Higher Education of the Russian Federation (agreement no. 075-15-2025-303).