Complexity of Reasoning in Kleene and Action Algebras

Stepan Kuznetsov

ESSLLI 2022 · Galway, Ireland · Gallimh, Éire

Lecture 5

Recap

- Yesterday we defined action lattices (= residuated Kleene lattices), and the subclass of *-continuous action lattices.

Recap

- Yesterday we defined action lattices (= residuated Kleene lattices), and the subclass of *-continuous action lattices.
- Their inequational theories are **ACT** and $\mathbf{ACT}_\omega$ respectively.

Recap

- Yesterday we defined action lattices (= residuated Kleene lattices), and the subclass of *-continuous action lattices.
- Their inequational theories are **ACT** and $\mathbf{ACT}_\omega$ respectively.
- For the corresponding classes of commutative action algebras, the logics are **CommACT** and $\mathbf{CommACT}_\omega$.

Recap

- Yesterday we defined action lattices (= residuated Kleene lattices), and the subclass of *-continuous action lattices.
- Their inequational theories are **ACT** and $\mathbf{ACT}_\omega$ respectively.
- For the corresponding classes of commutative action algebras, the logics are **CommACT** and $\mathbf{CommACT}_\omega$.
- We proved that $\mathbf{CommACT}_\omega$ is Π^0_1-hard and **CommACT** is Σ^0_1-hard.

Recap

- Yesterday we defined action lattices (= residuated Kleene lattices), and the subclass of *-continuous action lattices.
- Their inequational theories are **ACT** and $\mathbf{ACT}_\omega$ respectively.
- For the corresponding classes of commutative action algebras, the logics are **CommACT** and $\mathbf{CommACT}_\omega$.
- We proved that $\mathbf{CommACT}_\omega$ is Π^0_1-hard and **CommACT** is Σ^0_1-hard.
- This was done by encoding infinite and cyclic (resp.) behaviour of counter machines.

- From a bird-eye view, the instruction set of a machine is encoded by formula E.

Encoding Infinite Computations

- From a bird-eye view, the instruction set of a machine is encoded by formula E.
- This formula is put under iteration (E^*) into the antecedent.

Encoding Infinite Computations

- From a bird-eye view, the instruction set of a machine is encoded by formula E.
- This formula is put under iteration (E^*) into the antecedent.
- This licenses, by the ω-rule, extracting an arbitrary number of copies of E, which means infinite execution.

Encoding Infinite Computations

- From a bird-eye view, the instruction set of a machine is encoded by formula E.
- This formula is put under iteration (E^*) into the antecedent.
- This licenses, by the ω-rule, extracting an arbitrary number of copies of E, which means infinite execution.
- For a circular run, this proof can be made circular and rendered into **CommACT**.

Encoding Infinite Computations

- From a bird-eye view, the instruction set of a machine is encoded by formula E.
- This formula is put under iteration (E^*) into the antecedent.
- This licenses, by the ω-rule, extracting an arbitrary number of copies of E, which means infinite execution.
- For a circular run, this proof can be made circular and rendered into **CommACT**.
- In the non-commutative setting, the following issue arises: now we have no way to move E to the desired place in the antecedent.

Encoding Infinite Computations

- From a bird-eye view, the instruction set of a machine is encoded by formula E.
- This formula is put under iteration (E^*) into the antecedent.
- This licenses, by the ω-rule, extracting an arbitrary number of copies of E, which means infinite execution.
- For a circular run, this proof can be made circular and rendered into **CommACT**.
- In the non-commutative setting, the following issue arises: now we have no way to move E to the desired place in the antecedent.
- For example, we need to modify counter c, but we have a and b in between: $E, q, \mathrm{a}^a, \mathrm{b}^b, \mathrm{c}^c$.

Handling Noncommutativity

- Buszkowski (2007) suggested an indirect method of encoding the non-halting problem in $\mathbf{ACT}_{\omega}$.

Handling Noncommutativity

- Buszkowski (2007) suggested an indirect method of encoding the non-halting problem in $\mathbf{ACT}_\omega$.
- This method goes via **totality of context-free grammars.**

Handling Noncommutativity

- Buszkowski (2007) suggested an indirect method of encoding the non-halting problem in $\mathbf{ACT}_\omega$.
- This method goes via **totality of context-free grammars.**
- The workflow is as follows:

$$\begin{aligned} \mathscr{M} \text{ does not halt on } x \iff \mathscr{G}_{\mathscr{M},x} &\text{ generates all non-empty words} \\ &\iff \mathbf{ACT}_\omega \vdash E^+_{\mathscr{M},x} \to S \end{aligned}$$

Handling Noncommutativity

- Buszkowski (2007) suggested an indirect method of encoding the non-halting problem in $\mathbf{ACT}_\omega$.
- This method goes via **totality of context-free grammars.**
- The workflow is as follows:

$$\begin{aligned} \mathscr{M} \text{ does not halt on } x \iff \mathscr{G}_{\mathscr{M},x} &\text{ generates all non-empty words} \\ &\iff \mathbf{ACT}_\omega \vdash E^+_{\mathscr{M},x} \to S \end{aligned}$$

  - Recall that $E^+ = E \cdot E^*$.

Handling Noncommutativity

- Buszkowski (2007) suggested an indirect method of encoding the non-halting problem in $\mathbf{ACT}_\omega$.
- This method goes via **totality of context-free grammars.**
- The workflow is as follows:

$$\begin{aligned} \mathscr{M} \text{ does not halt on } x \iff \mathscr{G}_{\mathscr{M},x} \text{ generates all non-empty words} \\ \iff \mathbf{ACT}_\omega \vdash E^+_{\mathscr{M},x} \to S \end{aligned}$$

  - Recall that $E^+ = E \cdot E^*$.
- The construction of $\mathscr{G}_{\mathscr{M},x}$ is standard: this grammar generates all words which are **not** the halting protocol of $\mathscr{M}$ on x.

Let us recall sequent calculi for $\mathbf{ACT}_\omega$ and $\mathbf{ACT}$. The core system is **MALC**, the multiplicative-additive Lambek calculus

$$\frac{}{A \to A}\ Id \qquad \frac{}{\Gamma, 0, \Delta \to B}\ 0L$$

$$\frac{\Gamma, \Delta \to B}{\Gamma, 1, \Delta \to B}\ 1L \qquad \frac{}{\to 1}\ 1R \qquad \frac{\Gamma, A, B, \Delta \to C}{\Gamma, A \cdot B, \Delta \to C}\ {\cdot}L \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma, \Delta \to A \cdot B}\ {\cdot}R$$

$$\frac{\Gamma, A, \Delta \to C \quad \Gamma, B, \Delta \to C}{\Gamma, A \vee B, \Delta \to C}\ {\vee}L \qquad \frac{\Pi \to A}{\Pi \to A \vee B}\ {\vee}R_1 \qquad \frac{\Pi \to B}{\Pi \to A \vee B}\ {\vee}R_2$$

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, A \wedge B, \Delta \to C}\ {\wedge}L_1 \qquad \frac{\Gamma, B, \Delta \to C}{\Gamma, A \wedge B, \Delta \to C}\ {\wedge}L_2 \qquad \frac{\Pi \to A \quad \Pi \to B}{\Pi \to A \wedge B}\ {\wedge}R$$

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

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

$\mathbf{ACT}_\omega$ and ACT

- $\mathbf{ACT}_\omega$ is obtained from **MALC** by adding the following rules:

$$\frac{\left(\Gamma, A^n, \Delta \to C\right)_{n=0}^{\infty}}{\Gamma, A^*, \Delta \to C}\ {*}L_\omega \qquad \frac{\Pi_1 \to A \quad \dots \quad \Pi_n \to A}{\Pi_1, \dots, \Pi_n \to A^*}\ {*}R_n$$

$\mathbf{ACT}_\omega$ and ACT

- $\mathbf{ACT}_\omega$ is obtained from **MALC** by adding the following rules:

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

- The rules for * in **ACT** are as follows:

$$\frac{\to B \quad A, B \to B}{A^* \to B} \qquad \frac{}{\to A^*} \qquad \frac{\Gamma \to A \quad \Delta \to A^*}{\Gamma, \Delta \to A^*}$$

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

$\mathbf{ACT}_\omega$ and ACT

- $\mathbf{ACT}_\omega$ is obtained from **MALC** by adding the following rules:

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

- The rules for * in **ACT** are as follows:

$$\frac{\to B \quad A, B \to B}{A^* \to B} \qquad \frac{}{\to A^*} \qquad \frac{\Gamma \to A \quad \Delta \to A^*}{\Gamma, \Delta \to A^*}$$

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

- In **ACT**, the cut rule is crucial; in contrast, in $\mathbf{ACT}_\omega$ it is admissible, which is shown by a transfinite version of the standard argument (Palka 2007).

Encoding Context-Free Grammars

- Recall that a context-free grammar is a rewriting system in which left-hand sides of rewriting rules are just letter: $A \Rightarrow \alpha$.

Encoding Context-Free Grammars

- Recall that a context-free grammar is a rewriting system in which left-hand sides of rewriting rules are just letter: $A \Rightarrow \alpha$.
- The context-free grammar gets transformed into Greibach normal form (Greibach 1965), with rules of the form $A \Rightarrow aB_1 \dots B_\ell$.

Encoding Context-Free Grammars

- Recall that a context-free grammar is a rewriting system in which left-hand sides of rewriting rules are just letter: $A \Rightarrow \alpha$.
- The context-free grammar gets transformed into Greibach normal form (Greibach 1965), with rules of the form $A \Rightarrow aB_1 \dots B_\ell$.
- Now $E_{\mathscr{M},x}$ is constructed as follows:

$$F_a = \bigwedge \{A \big/ (B_1 \cdot \ldots \cdot B_\ell) \mid (A \Rightarrow B_1 \dots B_\ell) \text{ is a rule of } \mathscr{G}_{\mathscr{M},x}\}$$

$$E_{\mathscr{M},x} = \bigvee_a F_a.$$

Encoding Context-Free Grammars

- Recall that a context-free grammar is a rewriting system in which left-hand sides of rewriting rules are just letter: $A \Rightarrow \alpha$.
- The context-free grammar gets transformed into Greibach normal form (Greibach 1965), with rules of the form $A \Rightarrow aB_1 \dots B_\ell$.
- Now $E_{\mathscr{M},x}$ is constructed as follows:

$$F_a = \bigwedge \{A \,/\, (B_1 \cdot \ldots \cdot B_\ell) \mid (A \Rightarrow B_1 \dots B_\ell) \text{ is a rule of } \mathscr{G}_{\mathscr{M},x}\}$$

$$E_{\mathscr{M},x} = \bigvee_a F_a.$$

- By construction, a word $a_1 \dots a_n$ is derivable in $\mathscr{G}_{\mathscr{M},x}$ iff $F_{a_1}, \dots, F_{a_n} \to S$ is derivable in MALC.

Encoding Context-Free Grammars

- Recall that a context-free grammar is a rewriting system in which left-hand sides of rewriting rules are just letter: $A \Rightarrow \alpha$.
- The context-free grammar gets transformed into Greibach normal form (Greibach 1965), with rules of the form $A \Rightarrow aB_1 \dots B_\ell$.
- Now $E_{\mathscr{M},x}$ is constructed as follows:

$$F_a = \bigwedge \{ A \,/\, (B_1 \cdot \ldots \cdot B_\ell) \mid (A \Rightarrow B_1 \dots B_\ell) \text{ is a rule of } \mathscr{G}_{\mathscr{M},x} \}$$
$$E_{\mathscr{M},x} = \bigvee_a F_a.$$

- By construction, a word $a_1 \dots a_n$ is derivable in $\mathscr{G}_{\mathscr{M},x}$ iff $F_{a_1}, \dots, F_{a_n} \to S$ is derivable in MALC.
- This finishes the proof of Π^0_1-hardness of $\mathbf{ACT}_\omega$ (Buszkowski 2007).

Encoding Circularity

- In order to encode circular runs of $\mathscr{M}$ in **ACT**, we need some extra work to be done.

Encoding Circularity

- In order to encode circular runs of $\mathscr{M}$ in **ACT**, we need some extra work to be done.
- We consider only **trivial cycling** (looping), which means that $\mathscr{M}$ gets stuck at a designated state q_c.

Encoding Circularity

- In order to encode circular runs of $\mathscr{M}$ in **ACT**, we need some extra work to be done.
- We consider only **trivial cycling** (looping), which means that $\mathscr{M}$ gets stuck at a designated state q_c.
- Thus, if a word includes q_c, it cannot be a terminating protocol of $\mathscr{M}$.

Encoding Circularity

- In order to encode circular runs of $\mathscr{M}$ in **ACT**, we need some extra work to be done.
- We consider only **trivial cycling** (looping), which means that $\mathscr{M}$ gets stuck at a designated state q_c.
- Thus, if a word includes q_c, it cannot be a terminating protocol of $\mathscr{M}$.
- The rules which make this explicit will be added to $\mathscr{G}_{\mathscr{M},x}$:

$$S \Rightarrow aXU \qquad U \Rightarrow aU \qquad U \Rightarrow a$$

(for each a), where X generates all words which either include q_c or locally violate the instructions of $\mathscr{M}$.

Encoding Circularity

- In order to encode circular runs of $\mathscr{M}$ in **ACT**, we need some extra work to be done.
- We consider only **trivial cycling** (looping), which means that $\mathscr{M}$ gets stuck at a designated state q_c.
- Thus, if a word includes q_c, it cannot be a terminating protocol of $\mathscr{M}$.
- The rules which make this explicit will be added to $\mathscr{G}_{\mathscr{M},x}$:

$$S \Rightarrow aXU \qquad U \Rightarrow aU \qquad U \Rightarrow a$$

  (for each a), where X generates all words which either include q_c or locally violate the instructions of $\mathscr{M}$.
- If $\mathscr{M}$ does not halt on x, then any sufficiently long word ($\geq n$ letters) will be derived by $S \Rightarrow aXU$ as the first rule.

Encoding Circularity

- Now we are ready to construct the **ACT** proof for $E^{+}_{\mathscr{M},x} \rightarrow S$. (Let $E = E_{\mathscr{M},x}$)

Encoding Circularity

- Now we are ready to construct the **ACT** proof for $E^+_{\mathscr{M},x} \to S$. (Let $E = E_{\mathscr{M},x}$)
- First we establish $E^+ \to U$:

$$\dfrac{\dfrac{\dfrac{U \to U}{(F_a \to U)_{a\in\Sigma}}}{E \to U} \qquad \dfrac{\dfrac{U \,/\, U, U \to U}{(F_a, U \to U)_{a\in\Sigma}}}{E, U \to U}}{E^+ \to U}$$

Encoding Circularity

- Now we are ready to construct the **ACT** proof for $E^+_{\mathscr{M},x} \to S$. (Let $E = E_{\mathscr{M},x}$)
- First we establish $E^+ \to U$:

$$\dfrac{\dfrac{\dfrac{U \to U}{(F_a \to U)_{a\in\Sigma}}}{E \to U} \qquad \dfrac{\dfrac{U \,/\, U, U \to U}{(F_a, U \to U)_{a\in\Sigma}}}{E, U \to U}}{E^+ \to U}$$

- Next, we use the “long decomposition” rule:

$$\frac{E \to S \quad E^2 \to S \quad \dots \quad E^n \to S \quad E^n, E^+ \to S}{E^+ \to S}$$

which follows from $E^+ \equiv E \vee E^2 \vee \dots \vee E^n \vee (E^n \cdot E^+)$.

Encoding Circularity

- Sequents $E^i \to S$ are derivable in the same way as in Buszkowski's proof (they do not include Kleene star).

Encoding Circularity

- Sequents $E^i \to S$ are derivable in the same way as in Buszkowski's proof (they do not include Kleene star).
- For $E^n, E^+ \to S$, the derivation is as follows:

$$\dfrac{E^+ \to U \qquad \dfrac{\dfrac{\dfrac{(F_{a_2}, \ldots, F_{a_n} \to X)_{a_2 \ldots a_n \in \Sigma^*} \quad U \to U \quad S \to S}{(S\,/(X \cdot U), F_{a_2}, \ldots, F_{a_n}, U \to S)_{a_2 \ldots a_n \in \Sigma^*}}}{(F_{a_1}, F_{a_2}, \ldots, F_{a_n}, U \to S)_{a_1 \ldots a_n \in \Sigma^*}}}{E^n, U \to S}}{E^n, E^+ \to S}$$

Encoding Circularity

- Sequents $E^i \to S$ are derivable in the same way as in Buszkowski's proof (they do not include Kleene star).
- For $E^n, E^+ \to S$, the derivation is as follows:

$$\dfrac{E^+ \to U \qquad \dfrac{\dfrac{\dfrac{(F_{a_2}, \ldots, F_{a_n} \to X)_{a_2 \ldots a_n \in \Sigma^*} \quad U \to U \quad S \to S}{(S\,/(X \cdot U), F_{a_2}, \ldots, F_{a_n}, U \to S)_{a_2 \ldots a_n \in \Sigma^*}}}{(F_{a_1}, F_{a_2}, \ldots, F_{a_n}, U \to S)_{a_1 \ldots a_n \in \Sigma^*}}}{E^n, U \to S}}{E^n, E^+ \to S}$$

- Since n is sufficiently big, $X \Rightarrow^* a_2 \ldots a_n$, thus the sequent on top is derivable.

Inseparability and Intermediate Systems

- Let $\mathscr{C}_T$ denote the set of machines and inputs which enjoy trivial cycling.

Inseparability and Intermediate Systems

- Let $\mathscr{C}_T$ denote the set of machines and inputs which enjoy trivial cycling.
- As for $\mathscr{C}$, $\mathscr{C}_T$ and $\mathscr{H}$ are effectively inseparable.

Inseparability and Intermediate Systems

- Let $\mathscr{C}_T$ denote the set of machines and inputs which enjoy trivial cycling.
- As for $\mathscr{C}$, $\mathscr{C}_T$ and $\mathscr{H}$ are effectively inseparable.
- This yields the desired complexity result:

Inseparability and Intermediate Systems

- Let $\mathscr{C}_T$ denote the set of machines and inputs which enjoy trivial cycling.
- As for $\mathscr{C}$, $\mathscr{C}_T$ and $\mathscr{H}$ are effectively inseparable.
- This yields the desired complexity result:

Theorem (K. 2020)

If a logic $\mathscr{L}$ is between **ACT** *and* $\mathbf{ACT}_\omega$ *and it is r.e., then it is* Σ^0_1*-complete.*

Inseparability and Intermediate Systems

- Let $\mathscr{C}_T$ denote the set of machines and inputs which enjoy trivial cycling.
- As for $\mathscr{C}$, $\mathscr{C}_T$ and $\mathscr{H}$ are effectively inseparable.
- This yields the desired complexity result:

Theorem (K. 2020)

If a logic $\mathscr{L}$ is between **ACT** *and* $\mathbf{ACT}_\omega$ *and it is r.e., then it is* Σ^0_1*-complete.*

- In particular, so is **ACT** itself.

Inseparability and Intermediate Systems

- Let $\mathscr{C}_T$ denote the set of machines and inputs which enjoy trivial cycling.
- As for $\mathscr{C}$, $\mathscr{C}_T$ and $\mathscr{H}$ are effectively inseparable.
- This yields the desired complexity result:

Theorem (K. 2020)

If a logic $\mathscr{L}$ is between **ACT** *and* $\mathbf{ACT}_\omega$ *and it is r.e., then it is* Σ^0_1*-complete.*

- In particular, so is **ACT** itself.
- Another interesting logic is **ACT** extended by the "induction-in-the-middle" rule:

$$\frac{\to B \quad A \to B \quad A, B, A \to B}{A^* \to B} \; {*L_{\mathrm{mid}}}$$

Upper Π^0_1 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π^0_1.

Upper Π^0_1 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π^0_1.
- There are several approaches to proving this upper bound, due to Palka (2007), Das & Pous (2018), K. & Speranski (2022).

Upper Π_1^0 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π_1^0.
- There are several approaches to proving this upper bound, due to Palka (2007), Das & Pous (2018), K. & Speranski (2022).
- Palka's approach is based on replacing each negative occurrence of A^* by its n-th approximation $1 \vee A \vee A^2 \vee \ldots \vee A^n$.

Upper Π^0_1 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π^0_1.
- There are several approaches to proving this upper bound, due to Palka (2007), Das & Pous (2018), K. & Speranski (2022).
- Palka's approach is based on replacing each negative occurrence of A^* by its n-th approximation $1 \vee A \vee A^2 \vee \ldots \vee A^n$.
- A sequent is provable iff so is its n-th approximation for each n. This yields Π^0_1.

Upper Π^0_1 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π^0_1.
- There are several approaches to proving this upper bound, due to Palka (2007), Das & Pous (2018), K. & Speranski (2022).
- Palka's approach is based on replacing each negative occurrence of A^* by its n-th approximation $1 \vee A \vee A^2 \vee \ldots \vee A^n$.
- A sequent is provable iff so is its n-th approximation for each n. This yields Π^0_1.
- The "if" direction here is non-trivial.

Upper Π_1^0 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π_1^0.
- There are several approaches to proving this upper bound, due to Palka (2007), Das & Pous (2018), K. & Speranski (2022).
- Palka's approach is based on replacing each negative occurrence of A^* by its n-th approximation $1 \vee A \vee A^2 \vee \ldots \vee A^n$.
- A sequent is provable iff so is its n-th approximation for each n. This yields Π_1^0.
- The "if" direction here is non-trivial.
- Moreover, Palka (2007) proved the **finite-model property (FMP)** for $\mathbf{ACT}_\omega$.

Upper Π_1^0 Bound

- For $\mathbf{ACT}_\omega$, it is not obvious that it is not harder than Π_1^0.
- There are several approaches to proving this upper bound, due to Palka (2007), Das & Pous (2018), K. & Speranski (2022).
- Palka's approach is based on replacing each negative occurrence of A^* by its n-th approximation $1 \vee A \vee A^2 \vee \ldots \vee A^n$.
- A sequent is provable iff so is its n-th approximation for each n. This yields Π_1^0.
- The "if" direction here is non-trivial.
- Moreover, Palka (2007) proved the **finite-model property (FMP)** for $\mathbf{ACT}_\omega$.
- Therefore, FMP does not hold for **ACT** (as all finite action lattices are *-continuous).

Upper Π^0_1 Bound

- Another approach (Das & Pous 2018) is based on non-well-founded proofs.

Upper Π^0_1 Bound

- Another approach (Das & Pous 2018) is based on non-well-founded proofs.
- Recall that $\mathbf{ACT}_\omega$ can be reformulated to a system with non-well-founded proofs, having the following rules for Kleene star:

$$\frac{\Gamma, \Delta \to B \quad \Gamma, A, A^*, \Delta \to B}{\Gamma, A^*, \Delta \to B}\ {*L} \qquad \frac{}{\to A^*}\ {*R_0} \qquad \frac{\Gamma \to A \quad \Delta \to A^*}{\Gamma, \Delta \to A^*}\ {*R}$$

where in $*R$ we require Γ to be non-empty.

Upper Π_1^0 Bound

- Another approach (Das & Pous 2018) is based on non-well-founded proofs.
- Recall that $\mathbf{ACT}_\omega$ can be reformulated to a system with non-well-founded proofs, having the following rules for Kleene star:

$$\frac{\Gamma, \Delta \to B \quad \Gamma, A, A^*, \Delta \to B}{\Gamma, A^*, \Delta \to B}\ {*L} \qquad \frac{}{\to A^*}\ {*R_0} \qquad \frac{\Gamma \to A \quad \Delta \to A^*}{\Gamma, \Delta \to A^*}\ {*R}$$

  where in $*R$ we require Γ to be non-empty.
- Under this restriction, and in the absence of cut, no correctness conditions are necessary.

Upper Π^0_1 Bound

- By an n-**preproof** let us denote a finite tree in which each path either reaches an axiom leaf or traverses $*L$ to the right n times.

Upper Π^0_1 Bound

- By an ***n*-preproof** let us denote a finite tree in which each path either reaches an axiom leaf or traverses $*L$ to the right n times.
- A non-well-founded proof π can be considered as an increasing chain of n-preproofs $\pi_0 \subset \pi_1 \subset \pi_2 \subset \dots$

Upper Π^0_1 Bound

- By an ***n*-preproof** let us denote a finite tree in which each path either reaches an axiom leaf or traverses $*L$ to the right n times.
- A non-well-founded proof π can be considered as an increasing chain of n-preproofs $\pi_0 \subset \pi_1 \subset \pi_2 \subset \ldots$
- Thus, if the sequent is provable, then for any n it has an n-preproof.

Upper Π^0_1 Bound

- By an ***n*-preproof** let us denote a finite tree in which each path either reaches an axiom leaf or traverses $*L$ to the right n times.
- A non-well-founded proof π can be considered as an increasing chain of n-preproofs $\pi_0 \subset \pi_1 \subset \pi_2 \subset \ldots$
- Thus, if the sequent is provable, then for any n it has an n-preproof.
- The key idea is that for each n-preproof π_n there is a **finite** number of its possible extensions to an $(n+1)$-preproof π_{n+1}.

Upper Π^0_1 Bound

- By an ***n*-preproof** let us denote a finite tree in which each path either reaches an axiom leaf or traverses $*L$ to the right n times.
- A non-well-founded proof π can be considered as an increasing chain of n-preproofs $\pi_0 \subset \pi_1 \subset \pi_2 \subset \ldots$
- Thus, if the sequent is provable, then for any n it has an n-preproof.
- The key idea is that for each n-preproof π_n there is a **finite** number of its possible extensions to an $(n+1)$-preproof π_{n+1}.
- Now by Kőnig's theorem if for any n there exists an n-preproof π_n, then there exists an increasing chain $\pi_0 \subset \pi_1 \subset \pi_2 \subset \ldots$, i.e., a non-well-founded proof π.

Complexity of $\mathbf{ACT}_\omega$

Theorem (Buszkowski & Palka 2007)

$\mathbf{ACT}_\omega$ *is* Π^0_1*-complete.*

Complexity of Fragments

- The Π_1^0 lower bound keeps valid without additives, $\wedge$ and $\vee$ (K. 2021).

Complexity of Fragments

- The Π_1^0 lower bound keeps valid without additives, $\wedge$ and $\vee$ (K. 2021).
- The proof is based on Safiullin's (2007) construction of Lambek grammars with unique type assignment.

Complexity of Fragments

- The Π_1^0 lower bound keeps valid without additives, $\wedge$ and $\vee$ (K. 2021).
- The proof is based on Safiullin's (2007) construction of Lambek grammars with unique type assignment.
- For any context-free grammar $\mathcal{G}$ over alphabet $\Sigma = \{a_1, \ldots, a_m\}$, there exist Lambek formulae $K_1, \ldots, K_m, H$ such that $a_{i_1} \ldots a_{i_n}$ is generated by $\mathcal{G}$ iff the sequent $K_{i_1}, \ldots, K_{i_n} \to H$ is derivable.

Complexity of Fragments

- The Π_1^0 lower bound keeps valid without additives, $\wedge$ and $\vee$ (K. 2021).
- The proof is based on Safiullin's (2007) construction of Lambek grammars with unique type assignment.
- For any context-free grammar $\mathcal{G}$ over alphabet $\Sigma = \{a_1, \dots, a_m\}$, there exist Lambek formulae $K_1, \dots, K_m, H$ such that $a_{i_1} \dots a_{i_n}$ is generated by $\mathcal{G}$ iff the sequent $K_{i_1}, \dots, K_{i_n} \to H$ is derivable.
- Now the grammar generates all words starting with a_1 iff $K_1, (K_1 \vee \dots \vee K_m)^+ \to H$ is derivable in $\mathbf{ACT}_\omega$, and $\vee$ is removed by applying $(A \vee B)^* \equiv A^* \cdot (B \cdot A^*)^*$ several times.

Complexity of Fragments

- The Π^0_1 lower bound keeps valid without additives, $\wedge$ and $\vee$ (K. 2021).
- The proof is based on Safiullin's (2007) construction of Lambek grammars with unique type assignment.
- For any context-free grammar $\mathcal{G}$ over alphabet $\Sigma = \{a_1, \dots, a_m\}$, there exist Lambek formulae $K_1, \dots, K_m, H$ such that $a_{i_1} \dots a_{i_n}$ is generated by $\mathcal{G}$ iff the sequent $K_{i_1}, \dots, K_{i_n} \to H$ is derivable.
- Now the grammar generates all words starting with a_1 iff $K_1, (K_1 \vee \dots \vee K_m)^+ \to H$ is derivable in $\mathbf{ACT}_\omega$, and $\vee$ is removed by applying $(A \vee B)^* \equiv A^* \cdot (B \cdot A^*)^*$ several times.
- There are techniques of getting rid of product also, thus getting Π^0_1-hardness of the fragment of $\mathbf{ACT}_\omega$ with $\backslash, /, ^*$ only.

Complexity of Fragments

- **Open problem:** complexity of the fragment of $\mathbf{ACT}_\omega$ with $\backslash, {}^*$ only (or $\backslash, \cdot, {}^*$).

Complexity of Fragments

- **Open problem:** complexity of the fragment of $\mathbf{ACT}_\omega$ with $\backslash, {}^*$ only (or $\backslash, \cdot, {}^*$).
- If it happens to be decidable, this would yield impossibility of Safiullin's theorem for the one-division fragment of the Lambek calculus.

Complexity of Fragments

- **Open problem:** complexity of the fragment of $\mathbf{ACT}_\omega$ with $\backslash, ^*$ only (or $\backslash, \cdot, ^*$).
- If it happens to be decidable, this would yield impossibility of Safiullin's theorem for the one-division fragment of the Lambek calculus.
- For fragments of **ACT**, the situation is a bit trickier.

Complexity of Fragments

- **Open problem:** complexity of the fragment of $\mathbf{ACT}_\omega$ with $\backslash, {}^*$ only (or $\backslash, \cdot, {}^*$).
- If it happens to be decidable, this would yield impossibility of Safiullin's theorem for the one-division fragment of the Lambek calculus.
- For fragments of **ACT**, the situation is a bit trickier.
- The best result known by now is undecidability (Σ^0_1-completeness) of the multiplicative-only fragment of **ACT** extended with the **decomposition rule** (K. 2020)

$$\frac{\Gamma, \Delta \to B \quad \Gamma, A, A^*, \Delta \to B}{\Gamma, A^*, \Delta \to B}\ {*L_{\mathrm{dec}}}$$

Complexity of Fragments

- **Open problem:** complexity of the fragment of $\mathbf{ACT}_\omega$ with $\backslash, {}^*$ only (or $\backslash, \cdot, {}^*$).
- If it happens to be decidable, this would yield impossibility of Safiullin's theorem for the one-division fragment of the Lambek calculus.
- For fragments of **ACT**, the situation is a bit trickier.
- The best result known by now is undecidability (Σ_1^0-completeness) of the multiplicative-only fragment of **ACT** extended with the **decomposition rule** (K. 2020)

$$\frac{\Gamma, \Delta \to B \quad \Gamma, A, A^*, \Delta \to B}{\Gamma, A^*, \Delta \to B} \; {*}L_{\mathrm{dec}}$$

- In the absence of additives, this rule is not *derivable*, but its *admissibility* is an open question.

Complexity of Fragments

- The decomposition rule is derivable using $\vee$, by cut with $A^* \equiv 1 \vee A^+$ (recall the notation $A^+ = A \cdot A^*$).

Complexity of Fragments

- The decomposition rule is derivable using $\vee$, by cut with $A^* \equiv 1 \vee A^+$ (recall the notation $A^+ = A \cdot A^*$).
- Moreover, it is also derivable using $\wedge$, as follows:

$$
\dfrac{
  \dfrac{
    \dfrac{\to B \qquad \dfrac{A, A^* \to B}{\to B \,/\, A^+}}{\to B \wedge (B \,/\, A^+)}
    \qquad
    \dfrac{\dfrac{\dfrac{
      \dfrac{\dfrac{B \to B}{B \wedge (B \,/\, A^+) \to B}}{\to (B \wedge (B \,/\, A^+)) \setminus B}
      \qquad
      \dfrac{\dfrac{\dfrac{
        \dfrac{
          \dfrac{A \to A^+ \quad B \to B}{B \,/\, A^+, A \to B}
          \qquad
          \dfrac{\dfrac{A, A^+ \to A^+ \quad B \to B}{B \,/\, A^+, A, A^+ \to B}}{B \,/\, A^+, A \to B \,/\, A^+}
        }{B \,/\, A^+, A \to B \wedge (B \,/\, A^+)}
        \qquad B \to B
      }{B \,/\, A^+, A, (B \wedge (B \,/\, A^+)) \setminus B \to B}}{B \wedge (B \,/\, A^+), A, (B \wedge (B \,/\, A^+)) \setminus B \to B}}{A, (B \wedge (B \,/\, A^+)) \setminus B \to (B \wedge (B \,/\, A^+)) \setminus B}
    }{A^* \to (B \wedge (B \,/\, A^+)) \setminus B}}{B \wedge (B \,/\, A^+), A^* \to B}}{B \wedge (B \,/\, A^+) \to B \,/\, A^*}
  }{\to B \,/\, A^*}
}{A^* \to B}
$$

(Γ and Δ are internalised into B by divisions).

Complexity of Fragments

- Thus, we have undecidability for all fragments between fragments of **ACT** with $\vee$ or with $\wedge$, and $\mathbf{ACT}_{\omega}$.

Complexity of Fragments

- Thus, we have undecidability for all fragments between fragments of **ACT** with $\vee$ or with $\wedge$, and $\mathbf{ACT}_\omega$.
- We also get undecidability for a multiplicative-only fragment of **ACT**, but extended with the decomposition rule.

- Thus, we have undecidability for all fragments between fragments of **ACT** with $\vee$ or with $\wedge$, and $\mathbf{ACT}_\omega$.
- We also get undecidability for a multiplicative-only fragment of **ACT**, but extended with the decomposition rule.
- For the case of $\vee$, we can do with only one division; in other situations, two divisions are required.

Distributivity

- Standard set-theoretic examples of action algebras, i.e., algebras of formal languages and algebras of binary relations, are **distributive** as lattices:

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$
$$(A \vee B) \wedge (A \vee C) \equiv A \vee (B \wedge C)$$

Distributivity

- Standard set-theoretic examples of action algebras, i.e., algebras of formal languages and algebras of binary relations, are **distributive** as lattices:

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$
$$(A \vee B) \wedge (A \vee C) \equiv A \vee (B \wedge C)$$

- While the $\leftarrow$ directions of these distributivity laws are derivable in **MALC**, the $\rightarrow$ ones are not (though they entail one another).

Distributivity

- Standard set-theoretic examples of action algebras, i.e., algebras of formal languages and algebras of binary relations, are **distributive** as lattices:

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$
$$(A \vee B) \wedge (A \vee C) \equiv A \vee (B \wedge C)$$

- While the $\leftarrow$ directions of these distributivity laws are derivable in **MALC**, the $\rightarrow$ ones are not (though they entail one another).
- Let us denote the systems with distributivity (added as an axiom scheme, with *Cut* enabled) by **MALCD**, **ACTD**, $\mathbf{ACTD}_\omega$.

Distributivity

- Obviously, the multiplicative-only fragment of $\mathbf{ACTD}_\omega$ coincides with that of $\mathbf{ACT}_\omega$.

Distributivity

- Obviously, the multiplicative-only fragment of $\mathbf{ACTD}_\omega$ coincides with that of $\mathbf{ACT}_\omega$.
- However, having only $\vee$ is sufficient to feel distributivity (Kanovich, K., Scedrov 2019):

$$((x \mathbin{/} y) \vee w) \mathbin{/} ((x \mathbin{/} y) \vee (x \mathbin{/} z) \vee w), (x \mathbin{/} y) \vee w, \\ ((x \mathbin{/} y) \vee w) \mathbin{\backslash} ((x \mathbin{/} z) \vee w) \to (x \mathbin{/} (y \vee z)) \vee w.$$

Distributivity

- Obviously, the multiplicative-only fragment of $\mathbf{ACTD}_\omega$ coincides with that of $\mathbf{ACT}_\omega$.
- However, having only $\vee$ is sufficient to feel distributivity (Kanovich, K., Scedrov 2019):

$$((x \,/\, y) \vee w) \,/\, ((x \,/\, y) \vee (x \,/\, z) \vee w), (x \,/\, y) \vee w, \\ ((x \,/\, y) \vee w) \setminus ((x \,/\, z) \vee w) \to (x \,/\, (y \vee z)) \vee w.$$

- Thus, we get the lower bound, Π_1^0-hardness.

Distributivity

- Obviously, the multiplicative-only fragment of $\mathbf{ACTD}_\omega$ coincides with that of $\mathbf{ACT}_\omega$.
- However, having only $\vee$ is sufficient to feel distributivity (Kanovich, K., Scedrov 2019):

$$((x \,/\, y) \vee w) \,/((x \,/\, y) \vee (x \,/\, z) \vee w), (x \,/\, y) \vee w,$$
$$((x \,/\, y) \vee w) \backslash ((x \,/\, z) \vee w) \to (x \,/(y \vee z)) \vee w.$$

- Thus, we get the lower bound, Π_1^0-hardness.
- For **ACTD**, we get Σ_1^0-completeness, since it includes the multiplicative-only fragment of **ACT** extended with $*L_{\mathrm{dec}}$ and is included into $\mathbf{ACTD}_\omega$.

Distributivity

- Obviously, the multiplicative-only fragment of $\mathbf{ACTD}_\omega$ coincides with that of $\mathbf{ACT}_\omega$.
- However, having only $\vee$ is sufficient to feel distributivity (Kanovich, K., Scedrov 2019):

$$((x \,/\, y) \vee w) \,/\, ((x \,/\, y) \vee (x \,/\, z) \vee w), (x \,/\, y) \vee w, \\ ((x \,/\, y) \vee w) \backslash ((x \,/\, z) \vee w) \to (x \,/\, (y \vee z)) \vee w.$$

- Thus, we get the lower bound, Π_1^0-hardness.
- For **ACTD**, we get Σ_1^0-completeness, since it includes the multiplicative-only fragment of **ACT** extended with $*L_{\mathrm{dec}}$ and is included into $\mathbf{ACTD}_\omega$.
- **Open problem:** is there a Π_1^0 upper bound on $\mathbf{ACTD}_\omega$?

- For **MALCD**, Kozak (2009) presented a cut-free sequent system.

Axiomatising Distributivity

- For **MALCD**, Kozak (2009) presented a cut-free sequent system.
- In Kozak's system, antecedents have both $\cdot$ and $\vee$ on the meta-syntactic level.

Axiomatising Distributivity

- For **MALCD**, Kozak (2009) presented a cut-free sequent system.
- In Kozak's system, antecedents have both $\cdot$ and $\vee$ on the meta-syntactic level.
- Thus, meta-formulae are constructed from formulae and the empty structure Λ using two meta-operators: $\odot$ and $\owedge$.

Axiomatising Distributivity

- For **MALCD**, Kozak (2009) presented a cut-free sequent system.
- In Kozak's system, antecedents have both $\cdot$ and $\vee$ on the meta-syntactic level.
- Thus, meta-formulae are constructed from formulae and the empty structure Λ using two meta-operators: $\odot$ and $\owedge$.
- Sequents are of the form $\Pi \to B$, where Π is a meta-formula and B is a formula.

Axiomatising Distributivity

- For **MALCD**, Kozak (2009) presented a cut-free sequent system.
- In Kozak's system, antecedents have both $\cdot$ and $\vee$ on the meta-syntactic level.
- Thus, meta-formulae are constructed from formulae and the empty structure Λ using two meta-operators: $\odot$ and $\owedge$.
- Sequents are of the form $\Pi \to B$, where Π is a meta-formula and B is a formula.
- There will be structural rules operating the meta-operators.

Kozak's Calculus for MALCD

$$\frac{}{A \to A}\ Id \qquad \frac{\Gamma[\Lambda] \to A}{\Gamma[1] \to A}\ 1L \qquad \frac{}{\Lambda \to 1}\ 1R \qquad \frac{}{\Gamma[0] \to A}\ 0L$$

$$\frac{\Gamma[A \owedge B] \to C}{\Gamma[A \wedge B] \to C}\ \wedge L \qquad \frac{\Pi \to A \quad \Pi \to B}{\Pi \to A \wedge B}\ \wedge R$$

$$\frac{\Gamma[A] \to C \quad \Gamma[B] \to C}{\Gamma[A \vee B] \to C}\ \vee L \qquad \frac{\Pi \to A}{\Pi \to A \vee B}\ \vee R_1 \qquad \frac{\Pi \to B}{\Pi \to A \vee B}\ \vee R_2$$

$$\frac{\Gamma[A \odot B] \to C}{\Gamma[A \cdot B] \to C}\ \cdot L \qquad \frac{\Gamma \to A \quad \Delta \to B}{\Gamma \odot \Delta \to A \cdot B}\ \cdot R$$

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

$$\frac{\Pi \to A \quad \Gamma[B] \to C}{\Gamma[(B / A) \odot \Pi] \to C}\ / L \qquad \frac{\Pi \odot A \to B}{\Pi \to B / A}\ / R$$

Kozak's Calculus for MALCD

Structural rules:

$$\frac{\Gamma[(\Pi \odot \Phi) \odot \Psi] \to A}{\Gamma[\Pi \odot (\Phi \odot \Psi)] \to A}\ \odot A_1 \qquad \frac{\Gamma[\Pi \odot (\Phi \odot \Psi)] \to A}{\Gamma[(\Pi \odot \Phi) \odot \Psi] \to A}\ \odot A_2$$

$$\frac{\Gamma[\Pi] \to A}{\Gamma[\Pi \odot \Lambda] \to A}\ \Lambda W_1 \qquad \frac{\Gamma[\Pi] \to A}{\Gamma[\Lambda \odot \Pi] \to A}\ \Lambda W_2$$

$$\frac{\Gamma[\Pi \odot \Lambda] \to A}{\Gamma[\Pi] \to A}\ \Lambda C_1 \qquad \frac{\Gamma[\Lambda \odot \Pi] \to A}{\Gamma[\Pi] \to A}\ \Lambda C_2$$

$$\frac{\Gamma[(\Pi \owedge \Phi) \owedge \Psi] \to A}{\Gamma[\Pi \owedge (\Phi \owedge \Psi)] \to A}\ \owedge A_1 \qquad \frac{\Gamma[\Pi \owedge (\Phi \owedge \Psi)] \to A}{\Gamma[(\Pi \owedge \Phi) \owedge \Psi] \to A}\ \odot A_2$$

$$\frac{\Gamma[\Pi] \to A}{\Gamma[\Pi \owedge \Phi] \to A}\ \owedge W \qquad \frac{\Gamma[\Phi \owedge \Pi] \to A}{\Gamma[\Pi \owedge \Phi] \to A}\ \owedge E \qquad \frac{\Gamma[\Pi \owedge \Pi] \to A}{\Gamma[\Pi] \to A}\ \owedge C$$

Kozak's Calculi for MALCD and ACTD$_\omega$

- Kozak's calculus admits cut:

$$\frac{\Pi \to A \quad \Gamma[A] \to B}{\Gamma[\Pi] \to B}\ Cut$$

Kozak's Calculi for MALCD and ACTD$_\omega$

- Kozak's calculus admits cut:

$$\frac{\Pi \to A \quad \Gamma[A] \to B}{\Gamma[\Pi] \to B}\ Cut$$

- However, in the absence of cut the $\owedge C$ rule becomes unavoidable.

Kozak's Calculi for MALCD and ACTD$_\omega$

- Kozak's calculus admits cut:

$$\frac{\Pi \to A \quad \Gamma[A] \to B}{\Gamma[\Pi] \to B}\ Cut$$

- However, in the absence of cut the $\owedge C$ rule becomes unavoidable.
- This is a contraction rule, and it makes proof search harder.

Kozak's Calculi for MALCD and $\mathbf{ACTD}_\omega$

- Kozak's calculus admits cut:

$$\frac{\Pi \to A \quad \Gamma[A] \to B}{\Gamma[\Pi] \to B}\ Cut$$

- However, in the absence of cut the $\owedge C$ rule becomes unavoidable.
- This is a contraction rule, and it makes proof search harder.
- Nevertheless, Kozak managed to prove decidability (however, without a PSPACE upper bound) and FMP for **MALCD**.

Kozak's Calculi for MALCD and ACTD$_\omega$

- Kozak's calculus admits cut:

$$\frac{\Pi \to A \quad \Gamma[A] \to B}{\Gamma[\Pi] \to B}\ \mathit{Cut}$$

- However, in the absence of cut the $\oslash C$ rule becomes unavoidable.
- This is a contraction rule, and it makes proof search harder.
- Nevertheless, Kozak managed to prove decidability (however, without a PSPACE upper bound) and FMP for **MALCD**.
- This calculus can be extended by Kleene star in a natural way:

$$\frac{\left(\Gamma[A^{\odot n}] \to B\right)_{n=0}^{\infty}}{\Gamma[A^*] \to B}\ {*L_\omega} \qquad \frac{\Pi_1 \to A \quad \ldots \quad \Pi_n \to A}{\Pi_1 \odot \ldots \odot \Pi_n \to A^*}\ {*R_n}$$

Kozak's Calculi for MALCD and ACTD$_\omega$

- Constructing a system with non-well-founded proofs is also possible:

$$\frac{\Gamma[\Lambda] \to B \quad \Gamma[A \odot A^*] \to B}{\Gamma[A^*] \to B}\ *L \qquad \frac{}{\Lambda \to A^*}\ *R_0 \qquad \frac{\Pi \to A \quad \Delta \to A^*}{\Pi \odot \Delta \to A^*}\ *R$$

Kozak's Calculi for MALCD and ACTD_ω

- Constructing a system with non-well-founded proofs is also possible:

$$\frac{\Gamma[\Lambda] \to B \quad \Gamma[A \odot A^*] \to B}{\Gamma[A^*] \to B} *L \qquad \frac{}{\Lambda \to A^*} *R_0 \qquad \frac{\Pi \to A \quad \Delta \to A^*}{\Pi \odot \Delta \to A^*} *R$$

- In the presence of $\oslash C$, however, now we require correctness conditions to be in place.

Kozak's Calculi for MALCD and ACTD$_\omega$

- Constructing a system with non-well-founded proofs is also possible:

$$\frac{\Gamma[\Lambda] \to B \quad \Gamma[A \odot A^*] \to B}{\Gamma[A^*] \to B}\ {*L} \qquad \frac{}{\Lambda \to A^*}\ {*R_0} \qquad \frac{\Pi \to A \quad \Delta \to A^*}{\Pi \odot \Delta \to A^*}\ {*R}$$

- In the presence of $\owedge C$, however, now we require correctness conditions to be in place.
- Namely, each infinite path should have a trace of A^* which traverses $*L$ infinitely often.

Kozak's Calculi for MALCD and $\mathbf{ACTD}_\omega$

- Constructing a system with non-well-founded proofs is also possible:

$$\frac{\Gamma[\Lambda] \to B \quad \Gamma[A \odot A^*] \to B}{\Gamma[A^*] \to B} *L \qquad \frac{}{\Lambda \to A^*} *R_0 \qquad \frac{\Pi \to A \quad \Delta \to A^*}{\Pi \odot \Delta \to A^*} *R$$

- In the presence of $\owedge C$, however, now we require correctness conditions to be in place.
- Namely, each infinite path should have a trace of A^* which traverses $*L$ infinitely often.
- This ruins the Π^0_1 upper bound argument of Das & Pous for $\mathbf{ACT}_\omega$.

Non-Associativity and Iterative Divisions

- An interesting variation of action logic is the non-associative one.

Non-Associativity and Iterative Divisions

- An interesting variation of action logic is the non-associative one.
- It is based on the non-associative version of **MALC** or **MALCD**, if we wish distributivity.

Non-Associativity and Iterative Divisions

- An interesting variation of action logic is the non-associative one.
- It is based on the non-associative version of **MALC** or **MALCD**, if we wish distributivity.
- In the non-associative case, defining Kleene star is problematic, as we do not know what A^n is.

- An interesting variation of action logic is the non-associative one.
- It is based on the non-associative version of **MALC** or **MALCD**, if we wish distributivity.
- In the non-associative case, defining Kleene star is problematic, as we do not know what A^n is.
- Instead, Sedlár (2020) suggested **iterated divisions,** $A \mathbin{\backslash\!\backslash} B$ and $B \mathbin{/\!/} A$, roughly meaning $A^+ \setminus B$ and $B \mathbin{/} A^+$.

Non-Associativity and Iterative Divisions

- For the extension of non-associative **MALCD** by iterated divisions, Sedlár proved decidability and FMP.

Non-Associativity and Iterative Divisions

- For the extension of non-associative **MALCD** by iterated divisions, Sedlár proved decidability and FMP.
- In particular, FMP means completeness w.r.t. *-continuous non-associative residuated lattices with iterated divisions (i.e., admissibility of the ω-rule).

Non-Associativity and Iterative Divisions

- For the extension of non-associative **MALCD** by iterated divisions, Sedlár proved decidability and FMP.
- In particular, FMP means completeness w.r.t. *-continuous non-associative residuated lattices with iterated divisions (i.e., admissibility of the ω-rule).
- In the associative situation, iterated divisions still give Π_1^0-hardness, at least in the *-continuous case (K. & Ryzhkova 2020).

(Sub)exponentials

- Finally, let us extend the system with subexponential modalities.

- Finally, let us extend the system with subexponential modalities.
- These modalities locally allow structural rules, which are absent in the original system.

- Finally, let us extend the system with subexponential modalities.
- These modalities locally allow structural rules, which are absent in the original system.
- In particular, the exponential allows both structural rules of contraction, weakening, and permutation.

(Sub)exponentials

- Finally, let us extend the system with subexponential modalities.
- These modalities locally allow structural rules, which are absent in the original system.
- In particular, the exponential allows both structural rules of contraction, weakening, and permutation.
- In general, we have a polymodal logic, with $!^i$ modalities, $i \in \mathscr{I}$.

Subexponentials

- The *subexponential signature* $\Sigma = (\mathscr{I}, \mathscr{C}, \mathscr{W}, \mathscr{E}, \preceq)$, where $\preceq$ is a partial order on $\mathscr{I}$, and $\mathscr{C}$ and $\mathscr{W}$ are subsets of $\mathscr{I}$, closed upward under $\preceq$.

Subexponentials

- The *subexponential signature* $\Sigma = (\mathscr{I}, \mathscr{C}, \mathscr{W}, \mathscr{E}, \preceq)$, where $\preceq$ is a partial order on $\mathscr{I}$, and $\mathscr{C}$ and $\mathscr{W}$ are subsets of $\mathscr{I}$, closed upward under $\preceq$.
- Rules for subexponentials:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !^i A, \Delta \to C}\ !L \qquad \frac{!^{i_1} A_1, \dots, !^{i_n} A_n \to B}{!^{i_1} A_1, \dots, !^{i_n} A_n \to !^j B}\ !R,\ i_k \succeq j$$

$$\frac{\Gamma, !^i A, !^i A \to C}{\Gamma, !^i A, \Delta \to C}\ !C,\ i \in \mathscr{C} \qquad \frac{\Gamma, \Delta \to C}{\Gamma, !^i A, \Delta \to C}\ !W,\ i \in \mathscr{W}$$

$$\frac{\Gamma, B, !^i A, \Delta \to C}{\Gamma, !^i A, B, \Delta \to C}\ !E_1,\ i \in \mathscr{E} \qquad \frac{\Gamma, !^i A, B, \Delta \to C}{\Gamma, B, !^i A, \Delta \to C}\ !E_2,\ i \in \mathscr{E}$$

Subexponentials

- The *subexponential signature* $\Sigma = (\mathcal{I}, \mathcal{C}, \mathcal{W}, \mathcal{E}, \preceq)$, where $\preceq$ is a partial order on $\mathcal{I}$, and $\mathcal{C}$ and $\mathcal{W}$ are subsets of $\mathcal{I}$, closed upward under $\preceq$.
- Rules for subexponentials:

$$\frac{\Gamma, A, \Delta \to C}{\Gamma, !^i A, \Delta \to C}\ !L \qquad \frac{!^{i_1} A_1, \dots, !^{i_n} A_n \to B}{!^{i_1} A_1, \dots, !^{i_n} A_n \to !^j B}\ !R,\ i_k \succeq j$$

$$\frac{\Gamma, !^i A, !^i A \to C}{\Gamma, !^i A, \Delta \to C}\ !C,\ i \in \mathcal{C} \qquad \frac{\Gamma, \Delta \to C}{\Gamma, !^i A, \Delta \to C}\ !W,\ i \in \mathcal{W}$$

$$\frac{\Gamma, B, !^i A, \Delta \to C}{\Gamma, !^i A, B, \Delta \to C}\ !E_1,\ i \in \mathcal{E} \qquad \frac{\Gamma, !^i A, B, \Delta \to C}{\Gamma, B, !^i A, \Delta \to C}\ !E_2,\ i \in \mathcal{E}$$

- For the sake of cut elimination, we require $\mathcal{C} \subseteq \mathcal{E}$.

Closure Ordinal

- Let $\mathscr{D} : \mathscr{P}(\mathrm{Seq}) \to \mathscr{P}(\mathrm{Seq})$ denote the *immediate derivability* operator on sequents, for the given calculus.

Closure Ordinal

- Let $\mathscr{D} : \mathscr{P}(\mathrm{Seq}) \to \mathscr{P}(\mathrm{Seq})$ denote the *immediate derivability* operator on sequents, for the given calculus.
- Let $S_\alpha = \mathscr{D}^\alpha(\emptyset)$, where $\alpha \in \mathrm{Ord}$, form the transfinite sequence of iterations of $\mathscr{D}$.

Closure Ordinal

- Let $\mathscr{D} : \mathscr{P}(\mathrm{Seq}) \to \mathscr{P}(\mathrm{Seq})$ denote the *immediate derivability* operator on sequents, for the given calculus.
- Let $S_\alpha = \mathscr{D}^\alpha(\emptyset)$, where $\alpha \in \mathrm{Ord}$, form the transfinite sequence of iterations of $\mathscr{D}$.
- By Knaster – Tarski, this sequence has a fixed point.

Closure Ordinal

- Let $\mathscr{D} : \mathscr{P}(\mathrm{Seq}) \to \mathscr{P}(\mathrm{Seq})$ denote the *immediate derivability* operator on sequents, for the given calculus.
- Let $S_\alpha = \mathscr{D}^\alpha(\emptyset)$, where $\alpha \in \mathrm{Ord}$, form the transfinite sequence of iterations of $\mathscr{D}$.
- By Knaster – Tarski, this sequence has a fixed point.
- The ordinal α_0 which is the smallest fixed point is called the *closure ordinal* for $\mathscr{D}$.

Closure Ordinal

- Let $\mathscr{D} : \mathscr{P}(\mathrm{Seq}) \to \mathscr{P}(\mathrm{Seq})$ denote the *immediate derivability* operator on sequents, for the given calculus.
- Let $S_\alpha = \mathscr{D}^\alpha(\emptyset)$, where $\alpha \in \mathrm{Ord}$, form the transfinite sequence of iterations of $\mathscr{D}$.
- By Knaster – Tarski, this sequence has a fixed point.
- The ordinal α_0 which is the smallest fixed point is called the *closure ordinal* for $\mathscr{D}$.
- Closure ordinals for variations of infinitary action logic with subexponentials are being studied in joint work of K. and Stanislav Speranski.

Closure Ordinal and Complexity

- A general folklore result: for any monotone Π^1_1 operator $F: \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ its least fixed point is Π^1_1-bounded and its closure ordinal is $\leq \omega_1^{\mathrm{CK}}$.

Closure Ordinal and Complexity

- A general folklore result: for any monotone Π^1_1 operator $F : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ its least fixed point is Π^1_1-bounded and its closure ordinal is $\leq \omega_1^{\mathrm{CK}}$.
- This applies to our $\mathscr{D}$, which gives upper complexity bounds for infinitary action logic with subexponentials.

Closure Ordinal and Complexity

- A general folklore result: for any monotone Π^1_1 operator $F : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ its least fixed point is Π^1_1-bounded and its closure ordinal is $\leq \omega_1^{\mathrm{CK}}$.
- This applies to our $\mathscr{D}$, which gives upper complexity bounds for infinitary action logic with subexponentials.
- On the other hand, there are lower bounds.

Closure Ordinal and Complexity

- A general folklore result: for any monotone Π^1_1 operator $F : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ its least fixed point is Π^1_1-bounded and its closure ordinal is $\leq \omega_1^{\mathrm{CK}}$.
- This applies to our $\mathscr{D}$, which gives upper complexity bounds for infinitary action logic with subexponentials.
- On the other hand, there are lower bounds.
- In our system, if $\mathscr{C} \neq \varnothing$, we can encode well-foundedness of recursively defined graphs on $\mathbb{N}$ (which is Π^1_1-complete).

Closure Ordinal and Complexity

- A general folklore result: for any monotone Π^1_1 operator $F : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ its least fixed point is Π^1_1-bounded and its closure ordinal is $\leq \omega_1^{\mathrm{CK}}$.
- This applies to our $\mathscr{D}$, which gives upper complexity bounds for infinitary action logic with subexponentials.
- On the other hand, there are lower bounds.
- In our system, if $\mathscr{C} \neq \varnothing$, we can encode well-foundedness of recursively defined graphs on $\mathbb{N}$ (which is Π^1_1-complete).
    - In the non-commutative case, this is due to Kozen 2002, and we conjecture the same for the commutative case (encoding of Minsky machines).

Closure Ordinal and Complexity

- A general folklore result: for any monotone Π^1_1 operator $F : \mathscr{P}(\mathbb{N}) \to \mathscr{P}(\mathbb{N})$ its least fixed point is Π^1_1-bounded and its closure ordinal is $\leq \omega_1^{\mathrm{CK}}$.
- This applies to our $\mathscr{D}$, which gives upper complexity bounds for infinitary action logic with subexponentials.
- On the other hand, there are lower bounds.
- In our system, if $\mathscr{C} \neq \varnothing$, we can encode well-foundedness of recursively defined graphs on $\mathbb{N}$ (which is Π^1_1-complete).
    - In the non-commutative case, this is due to Kozen 2002, and we conjecture the same for the commutative case (encoding of Minsky machines).
- This yields that the closure ordinal is *exactly* ω_1^{CK}: otherwise complexity would have been hyperarithmetical.

Lower Closure Ordinals

- If $\mathscr{C} = \emptyset$, we can define a complexity measure η on sequents, such that $\eta(\Gamma \to A) < \omega^{\omega}$ for any $\Gamma \to A$.

Lower Closure Ordinals

- If $\mathscr{C} = \emptyset$, we can define a complexity measure η on sequents, such that $\eta(\Gamma \to A) < \omega^{\omega}$ for any $\Gamma \to A$.
  - Kleene star is multiplying by ω, others just add (componentwise addition in Cantor normal form).

Lower Closure Ordinals

- If $\mathscr{C} = \emptyset$, we can define a complexity measure η on sequents, such that $\eta(\Gamma \to A) < \omega^\omega$ for any $\Gamma \to A$.
  - Kleene star is multiplying by ω, others just add (componentwise addition in Cantor normal form).
- For each rule, its premises have strictly smaller η than the conclusion.

Lower Closure Ordinals

- If $\mathscr{C} = \emptyset$, we can define a complexity measure η on sequents, such that $\eta(\Gamma \to A) < \omega^\omega$ for any $\Gamma \to A$.
  - Kleene star is multiplying by ω, others just add (componentwise addition in Cantor normal form).
- For each rule, its premises have strictly smaller η than the conclusion.
- This gives an ω^ω upper bound on the closure ordinal.

Lower Closure Ordinals

- If $\mathscr{C} = \emptyset$, we can define a complexity measure η on sequents, such that $\eta(\Gamma \to A) < \omega^\omega$ for any $\Gamma \to A$.
  - Kleene star is multiplying by ω, others just add (componentwise addition in Cantor normal form).
- For each rule, its premises have strictly smaller η than the conclusion.
- This gives an ω^ω upper bound on the closure ordinal.
- Next, this quickly shows that the system is at the ω^ω-th level of the hyperarithmetical hierarchy.

Lower Closure Ordinals

- If $\mathscr{C} = \emptyset$, we can define a complexity measure η on sequents, such that $\eta(\Gamma \to A) < \omega^\omega$ for any $\Gamma \to A$.
  - Kleene star is multiplying by ω, others just add (componentwise addition in Cantor normal form).
- For each rule, its premises have strictly smaller η than the conclusion.
- This gives an ω^ω upper bound on the closure ordinal.
- Next, this quickly shows that the system is at the ω^ω-th level of the hyperarithmetical hierarchy.
- However, at each step we have only "$\forall$" quantifiers, or something finite and computable, therefore we get a Π^0_1 upper bound.

He will settle disputes among the nations and provide arbitration for many peoples. They will beat their swords into plows and their spears into pruning knives. Nation will not take up the sword against nation, and they will never again train for war.

Isaiah 2:4