Syntactic Concept Lattice Models for Infinitary Action Logic

Stepan L. Kuznetsov[1,2]

[1]Steklov Mathematical Institute, RAS [2]HSE University
Moscow, Russia

WoLLIC 2024 · Bern, Switzerland, June 10–13, 2024

HSE University Computer Science Dept. Logic Seminar 2024

The Lambek Calculus

- The **Lambek calculus** was introduced by J. Lambek in 1958 to provide a mathematical description of natural language syntax.

The Lambek Calculus

- The **Lambek calculus** was introduced by J. Lambek in 1958 to provide a mathematical description of natural language syntax.
- Algebraically, it is the logic of residuated semigroups.

The Lambek Calculus

- The **Lambek calculus** was introduced by J. Lambek in 1958 to provide a mathematical description of natural language syntax.
- Algebraically, it is the logic of residuated semigroups.
- The Gentzen-style rules for divisions (residuals) are as follows:

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

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

The axiom is $A \to A$.

The Lambek Calculus

- The **Lambek calculus** was introduced by J. Lambek in 1958 to provide a mathematical description of natural language syntax.
- Algebraically, it is the logic of residuated semigroups.
- The Gentzen-style rules for divisions (residuals) are as follows:

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

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

  The axiom is $A \to A$.
- There is also the Cut rule, which is important for reasoning from hypotheses:

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

L-models

- The linguistic usage of the Lambek calculus suggests its natural interpretation on **the algebra of formal languages** $\mathscr{P}(\Sigma^*)$, called **L-models.**

L-models

- The linguistic usage of the Lambek calculus suggests its natural interpretation on **the algebra of formal languages** $\mathscr{P}(\Sigma^*)$, called **L-models.**
- Residuals are interpreted as **division of languages:**

$$A \setminus B = \{ v \in \Sigma^* \mid (\forall u \in A)\, uv \in B \}$$
$$B \mathbin{/} A = \{ u \in \Sigma^* \mid (\forall v \in A)\, uv \in B \}$$

L-models

- The linguistic usage of the Lambek calculus suggests its natural interpretation on **the algebra of formal languages** $\mathscr{P}(\Sigma^*)$, called **L-models.**
- Residuals are interpreted as **division of languages:**

$$A \setminus B = \{ v \in \Sigma^* \mid (\forall u \in A)\, uv \in B \}$$
$$B \mathbin{/} A = \{ u \in \Sigma^* \mid (\forall v \in A)\, uv \in B \}$$

- The original Lambek calculus has the **non-emptiness restriction** on antecedents. To accommodate it in L-models, one disallows the empty word (changes Σ^* to Σ^+).

L-models

- The linguistic usage of the Lambek calculus suggests its natural interpretation on **the algebra of formal languages** $\mathscr{P}(\Sigma^*)$, called **L-models.**
- Residuals are interpreted as **division of languages:**

$$A \setminus B = \{ v \in \Sigma^* \mid (\forall u \in A)\, uv \in B \}$$
$$B \mathbin{/} A = \{ u \in \Sigma^* \mid (\forall v \in A)\, uv \in B \}$$

- The original Lambek calculus has the **non-emptiness restriction** on antecedents. To accommodate it in L-models, one disallows the empty word (changes Σ^* to Σ^+).
- A straightforward canonical model argument [Buszkowski 1982] shows that the Lambek calculus is **strongly complete** w.r.t. L-models. (This means that semantic entailment of a set of hypotheses is equivalent to derivability from it.)

Extensions of the Lambek Calculus

- Buszkowski's strong completeness result, however, does not easily generalise to extensions of the Lambek calculus.

Extensions of the Lambek Calculus

- Buszkowski's strong completeness result, however, does not easily generalise to extensions of the Lambek calculus.
- The **product** operation (multiplicative conjunction) is axiomatised as follows:

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

For the Lambek calculus with product, weak completeness holds [Pentus 1995, 1998], but strong completeness fails [Buszkowski 2010, K. 2024].

Extensions of the Lambek Calculus

- Buszkowski's strong completeness result, however, does not easily generalise to extensions of the Lambek calculus.
- The **product** operation (multiplicative conjunction) is axiomatised as follows:

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

  For the Lambek calculus with product, weak completeness holds [Pentus 1995, 1998], but strong completeness fails [Buszkowski 2010, K. 2024].
- **Additive conjunction** (intersection), axiomatised as follows, keeps Buszkowski's completeness argument:

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

Extensions of the Lambek Calculus

- In contrast, for the dual operation of **additive disjunction** (union):

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

  completeness fails even in the weak sense.

Extensions of the Lambek Calculus

- In contrast, for the dual operation of **additive disjunction** (union):

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

  completeness fails even in the weak sense.
- This is due to the distributivity law: $(A \vee B) \wedge C \to (A \wedge C) \vee (B \wedge C)$, and can be propagated to the $\{\backslash, /, \vee\}$ fragment [Kanovich et al. 2002].

Extensions of the Lambek Calculus

- In contrast, for the dual operation of **additive disjunction** (union):

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

  completeness fails even in the weak sense.
- This is due to the distributivity law: $(A \vee B) \wedge C \to (A \wedge C) \vee (B \wedge C)$, and can be propagated to the $\{\backslash, /, \vee\}$ fragment [Kanovich et al. 2002].
- Issues with distributivity also affect the infinitary operation of **Kleene star:**

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

  Here the $\{/, \cdot, \wedge, {}^*\}$ fragment is not (weakly) complete.

Extensions of the Lambek Calculus

- **Constants** $\mathbf{0}$ and $\mathbf{1}$ are axiomatised as follows:

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

Extensions of the Lambek Calculus

- **Constants** $\mathbf{0}$ and $\mathbf{1}$ are axiomatised as follows:

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

- $\mathbf{0}$ and $\mathbf{1}$ are interpreted as, resp., $\varnothing$ and $\{\varepsilon\}$.

Extensions of the Lambek Calculus

- **Constants** $\mathbf{0}$ and $\mathbf{1}$ are axiomatised as follows:

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

- $\mathbf{0}$ and $\mathbf{1}$ are interpreted as, resp., $\varnothing$ and $\{\varepsilon\}$.
- There is also constant $\top$ (maximal element), expressible using $\mathbf{0}$: $\top = \mathbf{0} \mathbin{/} \mathbf{0}$.

Extensions of the Lambek Calculus

- **Constants** **0** and **1** are axiomatised as follows:

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

- **0** and **1** are interpreted as, resp., $\varnothing$ and $\{\varepsilon\}$.
- There is also constant $\top$ (maximal element), expressible using **0**: $\top = \mathbf{0} \mathbin{/} \mathbf{0}$.
- Constants **0** and **1** also lead to (weak) incompleteness: formulae of the form $\mathbf{1} \mathbin{/} A$, $\mathbf{0} \mathbin{/} A$ obey some finite-valued logic rules (e.g., commutativity).

Extensions of the Lambek Calculus

- **Constants** $\mathbf{0}$ and $\mathbf{1}$ are axiomatised as follows:

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

- $\mathbf{0}$ and $\mathbf{1}$ are interpreted as, resp., $\varnothing$ and $\{\varepsilon\}$.
- There is also constant $\top$ (maximal element), expressible using $\mathbf{0}$: $\top = \mathbf{0} \,/\, \mathbf{0}$.
- Constants $\mathbf{0}$ and $\mathbf{1}$ also lead to (weak) incompleteness: formulae of the form $\mathbf{1} \,/\, A$, $\mathbf{0} \,/\, A$ obey some finite-valued logic rules (e.g., commutativity).
- All the operations together form **infinitary aciton logic** $\mathbf{ACT}_\omega$; the fragment without Kleene star is **multiplicative-additive Lambek calculus** **MALC**.

Syntactic Concept Lattices

- **Syntactic concept lattices** (SCL) are a modification of language algebras, designed for better completeness properties and arguably more adequacy for linguistic intuitions [Clark 2009; Wurm 2012, 2017].

Syntactic Concept Lattices

- **Syntactic concept lattices** (SCL) are a modification of language algebras, designed for better completeness properties and arguably more adequacy for linguistic intuitions [Clark 2009; Wurm 2012, 2017].
- Linguistically, each Lambek formula, as a syntactic type is interpreted by the language of all phrases (words) of this type.

Syntactic Concept Lattices

- **Syntactic concept lattices** (SCL) are a modification of language algebras, designed for better completeness properties and arguably more adequacy for linguistic intuitions [Clark 2009; Wurm 2012, 2017].
- Linguistically, each Lambek formula, as a syntactic type is interpreted by the language of all phrases (words) of this type.
- In SCL, this language A is augmented with all words which can appear in the same **contexts** as the words of A.

Syntactic Concept Lattices

- **Syntactic concept lattices** (SCL) are a modification of language algebras, designed for better completeness properties and arguably more adequacy for linguistic intuitions [Clark 2009; Wurm 2012, 2017].
- Linguistically, each Lambek formula, as a syntactic type is interpreted by the language of all phrases (words) of this type.
- In SCL, this language A is augmented with all words which can appear in the same **contexts** as the words of A.
- E.g., if *John* is of type np (noun phrase), and may appear in contexts like __ *likes Mary* or Mary met __ yesterday, then the words and phrases which may appear in the same contexts (Pete or the boy whom Ann likes) should also be added to the language for np.

Syntactic Concept Lattices

- This forms the **closure** of language A, denoted by $A^{\triangleright\triangleleft}$.

Syntactic Concept Lattices

- This forms the **closure** of language A, denoted by $A^{\triangleright\triangleleft}$.
- More precisely, we fix a language L_0 ("all correct sentences") and define two operations:

$$A^{\triangleright} = \{(x,y) \in \Sigma^* \times \Sigma^* \mid (\forall w \in A)\, xwy \in L_0\}, \text{ for } A \subseteq \Sigma^*;$$
$$C^{\triangleleft} = \{v \in \Sigma^* \mid (\forall (x,y) \in C)\, xvy \in L_0\}, \text{ for } C \subseteq \Sigma^* \times \Sigma^*.$$

Syntactic Concept Lattices

- This forms the **closure** of language A, denoted by $A^{\triangleright\triangleleft}$.
- More precisely, we fix a language L_0 ("all correct sentences") and define two operations:

$$A^{\triangleright} = \{(x,y) \in \Sigma^* \times \Sigma^* \mid (\forall w \in A)\; xwy \in L_0\}, \text{ for } A \subseteq \Sigma^*;$$
$$C^{\triangleleft} = \{v \in \Sigma^* \mid (\forall (x,y) \in C)\; xvy \in L_0\}, \text{ for } C \subseteq \Sigma^* \times \Sigma^*.$$

- The mappings $A \mapsto A^{\triangleright\triangleleft}$ and $C \mapsto C^{\triangleleft\triangleright}$ are closure operators. The SCL, denoted by $\mathscr{B}_{L_0}$, is the set of all closed languages (or, by Galois isomorphism, closed sets of contexts).

Syntactic Concept Lattices

- This forms the **closure** of language A, denoted by $A^{\triangleright\triangleleft}$.
- More precisely, we fix a language L_0 ("all correct sentences") and define two operations:

$$A^{\triangleright} = \{(x,y) \in \Sigma^* \times \Sigma^* \mid (\forall w \in A)\, xwy \in L_0\}, \text{ for } A \subseteq \Sigma^*;$$
$$C^{\triangleleft} = \{v \in \Sigma^* \mid (\forall (x,y) \in C)\, xvy \in L_0\}, \text{ for } C \subseteq \Sigma^* \times \Sigma^*.$$

- The mappings $A \mapsto A^{\triangleright\triangleleft}$ and $C \mapsto C^{\triangleleft\triangleright}$ are closure operators. The SCL, denoted by $\mathscr{B}_{L_0}$, is the set of all closed languages (or, by Galois isomorphism, closed sets of contexts).
- In SCL-models of the Lambek calculus, after each operation, the resulting language should be replaced by each closure.

Syntactic Concept Lattices

- The general idea of concept lattice comes from **formal concept analysis** (FCA), an area on the border of information science and philosophy [Wille 1982, 1992; S.O. Kuznetsov 1996; ...].

Syntactic Concept Lattices

- The general idea of concept lattice comes from **formal concept analysis** (FCA), an area on the border of information science and philosophy [Wille 1982, 1992; S.O. Kuznetsov 1996; ...].
- In FCA, each **concept** is described by both its **extent,** which consists of all objects of the concept, and its **intent,** the set of all attributes shared by these objects.

Syntactic Concept Lattices

- The general idea of concept lattice comes from **formal concept analysis** (FCA), an area on the border of information science and philosophy [Wille 1982, 1992; S.O. Kuznetsov 1996; ...].
- In FCA, each **concept** is described by both its **extent,** which consists of all objects of the concept, and its **intent,** the set of all attributes shared by these objects.
- The extent and the intent are connected by Galois connection ($^{\triangleright}$ and $^{\triangleleft}$), and both should be closed.

Syntactic Concept Lattices

- The general idea of concept lattice comes from **formal concept analysis** (FCA), an area on the border of information science and philosophy [Wille 1982, 1992; S.O. Kuznetsov 1996; ...].
- In FCA, each **concept** is described by both its **extent,** which consists of all objects of the concept, and its **intent,** the set of all attributes shared by these objects.
- The extent and the intent are connected by Galois connection ($^{\triangleright}$ and $^{\triangleleft}$), and both should be closed.
- In SCL, the concept is a syntactic category. Its intent is the set of phrases of the given category, and its extent is the set of contexts in which these phrases may appear.

Syntactic Concept Lattices

- The general idea of concept lattice comes from **formal concept analysis** (FCA), an area on the border of information science and philosophy [Wille 1982, 1992; S.O. Kuznetsov 1996; ...].
- In FCA, each **concept** is described by both its **extent,** which consists of all objects of the concept, and its **intent,** the set of all attributes shared by these objects.
- The extent and the intent are connected by Galois connection ($^{\triangleright}$ and $^{\triangleleft}$), and both should be closed.
- In SCL, the concept is a syntactic category. Its intent is the set of phrases of the given category, and its extent is the set of contexts in which these phrases may appear.
- There is also a connection to **distributional semantics:** *"a word is characterized by the company it keeps"* [Firth 1957].

SCL-models

- For $\backslash, /, \wedge$, closure properties are preserved, so the operations are defined exactly in L-models.

SCL-models

- For $\backslash, /, \wedge$, closure properties are preserved, so the operations are defined exactly in L-models.
- Thus, Buszkowski's completeness actually follows from the general SCL-completeness below.

SCL-models

- For $\backslash$, $/$, $\wedge$, closure properties are preserved, so the operations are defined exactly in L-models.
- Thus, Buszkowski's completeness actually follows from the general SCL-completeness below.
- In contrast, for $\cdot$, $\wedge$, and *, closure should be added:

$$A_1 \circ A_2 = (A_1 \cdot A_2)^{\triangleright\triangleleft}, \qquad A_1 \sqcup A_2 = (A_1 \cup A_2)^{\triangleright\triangleleft}, \qquad A^{\circledast} = (A^*)^{\triangleright\triangleleft}.$$

SCL-models

- For $\backslash$, $/$, $\wedge$, closure properties are preserved, so the operations are defined exactly in L-models.
- Thus, Buszkowski's completeness actually follows from the general SCL-completeness below.
- In contrast, for $\cdot$, $\wedge$, and *, closure should be added:

$$A_1 \circ A_2 = (A_1 \cdot A_2)^{\triangleright\triangleleft}, \qquad A_1 \sqcup A_2 = (A_1 \cup A_2)^{\triangleright\triangleleft}, \qquad A^{\circledast} = (A^*)^{\triangleright\triangleleft}.$$

- The unit and zero are also replaced with their closures: $\varnothing^{\triangleright\triangleleft}$ and $\{\varepsilon\}^{\triangleright\triangleleft}$.

SCL-models

- For $\backslash$, $/$, $\wedge$, closure properties are preserved, so the operations are defined exactly in L-models.
- Thus, Buszkowski's completeness actually follows from the general SCL-completeness below.
- In contrast, for $\cdot$, $\wedge$, and *, closure should be added:

$$A_1 \circ A_2 = (A_1 \cdot A_2)^{\triangleright\triangleleft}, \qquad A_1 \sqcup A_2 = (A_1 \cup A_2)^{\triangleright\triangleleft}, \qquad A^{\circledast} = (A^*)^{\triangleright\triangleleft}.$$

- The unit and zero are also replaced with their closures: $\varnothing^{\triangleright\triangleleft}$ and $\{\varepsilon\}^{\triangleright\triangleleft}$.

Theorem (Wurm 2017)

MALC *without* **0** *is (weakly) complete w.r.t. SCL-models.*

SCL-models

- For $\backslash$, $/$, $\wedge$, closure properties are preserved, so the operations are defined exactly in L-models.
- Thus, Buszkowski's completeness actually follows from the general SCL-completeness below.
- In contrast, for $\cdot$, $\wedge$, and *, closure should be added:

$$A_1 \circ A_2 = (A_1 \cdot A_2)^{\triangleright\triangleleft}, \qquad A_1 \sqcup A_2 = (A_1 \cup A_2)^{\triangleright\triangleleft}, \qquad A^{\circledast} = (A^*)^{\triangleright\triangleleft}.$$

- The unit and zero are also replaced with their closures: $\varnothing^{\triangleright\triangleleft}$ and $\{\varepsilon\}^{\triangleright\triangleleft}$.

Theorem (Wurm 2017)

MALC *without* **0** *is (weakly) complete w.r.t. SCL-models.*

- For **0**, Wurm uses a non-standard interpretation to gain completeness.

Going Infinitary and Strong

- We extend Wurm's result to $\mathbf{ACT}_\omega$ and make completeness strong.

Going Infinitary and Strong

- We extend Wurm's result to $\mathbf{ACT}_\omega$ and make completeness strong.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *is strongly complete w.r.t. SCL-models.*

Going Infinitary and Strong

- We extend Wurm's result to $\mathbf{ACT}_\omega$ and make completeness strong.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *is strongly complete w.r.t. SCL-models.*

- The construction is a representation function from arbitrary residuated Kleene lattice (algebraic model for $\mathbf{ACT}_\omega$) to SCL. (The idea goes back to Wurm.)

Going Infinitary and Strong

- We extend Wurm's result to $\mathbf{ACT}_\omega$ and make completeness strong.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *is strongly complete w.r.t. SCL-models.*

- The construction is a representation function from arbitrary residuated Kleene lattice (algebraic model for $\mathbf{ACT}_\omega$) to SCL. (The idea goes back to Wurm.)
- This yields strong completeness, for SCL-models over a countable Σ.

Going Infinitary and Strong

- We extend Wurm's result to $\mathbf{ACT}_\omega$ and make completeness strong.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *is strongly complete w.r.t. SCL-models.*

- The construction is a representation function from arbitrary residuated Kleene lattice (algebraic model for $\mathbf{ACT}_\omega$) to SCL. (The idea goes back to Wurm.)
- This yields strong completeness, for SCL-models over a countable Σ.
- For operations which preserve closure, we get strong completeness w.r.t. standard L-models. These include, besides $\backslash$, $/$, $\wedge$, also composite operations ${}^*\backslash$ and $/{}^*$ (associate versions of **iterative divisions** [Sedlár 2019]).

Representation Function

- We start with a residuated Kleene lattice **K** (we shall take the Lindenbaum–Tarski algebra, relativised to hypotheses) and map it injectively to an SCL $\mathscr{B}_{L_0}$.

Representation Function

- We start with a residuated Kleene lattice $\mathbf{K}$ (we shall take the Lindenbaum–Tarski algebra, relativised to hypotheses) and map it injectively to an SCL $\mathscr{B}_{L_0}$.
- Define two alphabets, $\overline{\Sigma} = \{\overline{b} \mid b \in \mathbf{K}\}$ and $\underline{\Sigma} = \{\underline{b} \mid b \in \mathbf{K}\}$; let $\Sigma = \overline{\Sigma} \cup \underline{\Sigma}$.

Representation Function

- We start with a residuated Kleene lattice $\mathbf{K}$ (we shall take the Lindenbaum–Tarski algebra, relativised to hypotheses) and map it injectively to an SCL $\mathscr{B}_{L_0}$.
- Define two alphabets, $\overline{\Sigma} = \{\overline{b} \mid b \in \mathbf{K}\}$ and $\underline{\Sigma} = \{\underline{b} \mid b \in \mathbf{K}\}$; let $\Sigma = \overline{\Sigma} \cup \underline{\Sigma}$.
- For each $w = \overline{b}_1 \ldots \overline{b}_n$ let $w^{\bullet} = b_1 \cdot \ldots \cdot b_n$ (in $\mathbf{K}$).

Representation Function

- We start with a residuated Kleene lattice **K** (we shall take the Lindenbaum–Tarski algebra, relativised to hypotheses) and map it injectively to an SCL $\mathscr{B}_{L_0}$.
- Define two alphabets, $\overline{\Sigma} = \{\overline{b} \mid b \in \mathbf{K}\}$ and $\underline{\Sigma} = \{\underline{b} \mid b \in \mathbf{K}\}$; let $\Sigma = \overline{\Sigma} \cup \underline{\Sigma}$.
- For each $w = \overline{b}_1 \dots \overline{b}_n$ let $w^\bullet = b_1 \cdot \ldots \cdot b_n$ (in **K**).
- Finally, define L_0 and the mapping h:

$$L_0 = \{w\underline{b}u \mid w, u \in \overline{\Sigma}^* \text{ and } w^\bullet \leq b, u^\bullet \leq 1 \text{ in } \mathbf{K}\};$$
$$h(b) = \{(\varepsilon, \underline{b})\}^{\lhd} = \{w \in \Sigma^* \mid w\underline{b} \in L_0\}.$$

Representation Function

- We start with a residuated Kleene lattice $\mathbf{K}$ (we shall take the Lindenbaum–Tarski algebra, relativised to hypotheses) and map it injectively to an SCL $\mathscr{B}_{L_0}$.
- Define two alphabets, $\overline{\Sigma} = \{\overline{b} \mid b \in \mathbf{K}\}$ and $\underline{\Sigma} = \{\underline{b} \mid b \in \mathbf{K}\}$; let $\Sigma = \overline{\Sigma} \cup \underline{\Sigma}$.
- For each $w = \overline{b}_1 \dots \overline{b}_n$ let $w^\bullet = b_1 \cdot \ldots \cdot b_n$ (in $\mathbf{K}$).
- Finally, define L_0 and the mapping h:

$$L_0 = \{w\underline{b}u \mid w, u \in \overline{\Sigma}^* \text{ and } w^\bullet \leq b, u^\bullet \leq 1 \text{ in } \mathbf{K}\};$$
$$h(b) = \{(\varepsilon, \underline{b})\}^{\triangleleft} = \{w \in \Sigma^* \mid w\underline{b} \in L_0\}.$$

- This mapping h commutes with all operations, including Kleene star: $h(a^*) = \big(h(a)\big)^{\circledast}$.

Representation Function

- We start with a residuated Kleene lattice $\mathbf{K}$ (we shall take the Lindenbaum–Tarski algebra, relativised to hypotheses) and map it injectively to an SCL $\mathscr{B}_{L_0}$.
- Define two alphabets, $\overline{\Sigma} = \{\overline{b} \mid b \in \mathbf{K}\}$ and $\underline{\Sigma} = \{\underline{b} \mid b \in \mathbf{K}\}$; let $\Sigma = \overline{\Sigma} \cup \underline{\Sigma}$.
- For each $w = \overline{b}_1 \dots \overline{b}_n$ let $w^\bullet = b_1 \cdot \ldots \cdot b_n$ (in $\mathbf{K}$).
- Finally, define L_0 and the mapping h:

$$L_0 = \{w\underline{b}u \mid w, u \in \overline{\Sigma}^* \text{ and } w^\bullet \leq b, u^\bullet \leq 1 \text{ in } \mathbf{K}\};$$
$$h(b) = \{(\varepsilon, \underline{b})\}^{\triangleleft} = \{w \in \Sigma^* \mid w\underline{b} \in L_0\}.$$

- This mapping h commutes with all operations, including Kleene star: $h(a^*) = \big(h(a)\big)^{\circledast}$.
- Also, $h(1) = \{\varepsilon\}^{\triangleright\triangleleft}$, but $h(0) = \{\overline{0}\}^{\triangleright\triangleleft} \neq \varnothing^{\triangleright\triangleleft}$ (non-standard interpretation). Thus, we get completeness only for the $\mathbf{0}$-free fragment.

Finite Alphabet

- The alphabet Σ in the SCL-model constructed above is infinite (countable). In linguistic applications, however, alphabets are finite.

Finite Alphabet

- The alphabet Σ in the SCL-model constructed above is infinite (countable). In linguistic applications, however, alphabets are finite.
- We reduce the strong completeness result to a 2-letter alphabet $\Sigma_2 = \{e, f\}$.

Finite Alphabet

- The alphabet Σ in the SCL-model constructed above is infinite (countable). In linguistic applications, however, alphabets are finite.
- We reduce the strong completeness result to a 2-letter alphabet $\Sigma_2 = \{e, f\}$.
- For $\Sigma = \{a_1, a_2, \ldots\}$, the reduction function here is $g(a_i) = e f^i e$ (due to [Pentus 1995]).

Finite Alphabet

- The alphabet Σ in the SCL-model constructed above is infinite (countable). In linguistic applications, however, alphabets are finite.
- We reduce the strong completeness result to a 2-letter alphabet $\Sigma_2 = \{e, f\}$.
- For $\Sigma = \{a_1, a_2, \ldots\}$, the reduction function here is $g(a_i) = ef^ie$ (due to [Pentus 1995]).
- This is almost a homomorphism of SCLs, up to some emptiness issues.

Finite Alphabet

- The alphabet Σ in the SCL-model constructed above is infinite (countable). In linguistic applications, however, alphabets are finite.
- We reduce the strong completeness result to a 2-letter alphabet $\Sigma_2 = \{e, f\}$.
- For $\Sigma = \{a_1, a_2, \ldots\}$, the reduction function here is $g(a_i) = ef^ie$ (due to [Pentus 1995]).
- This is almost a homomorphism of SCLs, up to some emptiness issues.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *and* $\mathbf{1}$ *is strongly complete w.r.t. SCL-models over* Σ_2.

Finite Alphabet

- The alphabet Σ in the SCL-model constructed above is infinite (countable). In linguistic applications, however, alphabets are finite.
- We reduce the strong completeness result to a 2-letter alphabet $\Sigma_2 = \{e, f\}$.
- For $\Sigma = \{a_1, a_2, \ldots\}$, the reduction function here is $g(a_i) = ef^ie$ (due to [Pentus 1995]).
- This is almost a homomorphism of SCLs, up to some emptiness issues.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *and* $\mathbf{1}$ *is strongly complete w.r.t. SCL-models over* Σ_2*.*

- Here we had to sacrifice $\mathbf{1}$ also, due to emptiness issues.

Regular SCL-models

- An SCL is called **regular,** if so is the language L_0.

Regular SCL-models

- An SCL is called **regular,** if so is the language L_0.
- Wurm proved that an SCL is regular iff it is finite (i.e., $\mathscr{B}_{L_0}$ is a finite set).

Regular SCL-models

- An SCL is called **regular,** if so is the language L_0.
- Wurm proved that an SCL is regular iff it is finite (i.e., $\mathscr{B}_{L_0}$ is a finite set).

Theorem

Neither $\mathbf{ACT}_\omega$, *nor even* **MALC** *is strongly complete w.r.t. regular SCL-models.*

Regular SCL-models

- An SCL is called **regular,** if so is the language L_0.
- Wurm proved that an SCL is regular iff it is finite (i.e., $\mathscr{B}_{L_0}$ is a finite set).

Theorem

Neither $\mathbf{ACT}_\omega$, *nor even* **MALC** *is strongly complete w.r.t. regular SCL-models.*

- The argument comes from complexity: strong completeness w.r.t. finite SCL-models would yield a Π^0_1 upper bound, while reasoning from finite sets of hypotheses in **MALC** is Σ^0_1-hard [Buszkowski 1982].

Regular SCL-models

- An SCL is called **regular,** if so is the language L_0.
- Wurm proved that an SCL is regular iff it is finite (i.e., $\mathscr{B}_{L_0}$ is a finite set).

Theorem

Neither $\mathbf{ACT}_\omega$*, nor even* **MALC** *is strongly complete w.r.t. regular SCL-models.*

- The argument comes from complexity: strong completeness w.r.t. finite SCL-models would yield a Π_1^0 upper bound, while reasoning from finite sets of hypotheses in **MALC** is Σ_1^0-hard [Buszkowski 1982].
- Weak complexity, however, is easily extended to $\mathbf{ACT}_\omega$.

Regular SCL-models

- An SCL is called **regular,** if so is the language L_0.
- Wurm proved that an SCL is regular iff it is finite (i.e., $\mathscr{B}_{L_0}$ is a finite set).

Theorem

Neither $\mathbf{ACT}_\omega$*, nor even* **MALC** *is strongly complete w.r.t. regular SCL-models.*

- The argument comes from complexity: strong completeness w.r.t. finite SCL-models would yield a Π^0_1 upper bound, while reasoning from finite sets of hypotheses in **MALC** is Σ^0_1-hard [Buszkowski 1982].
- Weak complexity, however, is easily extended to $\mathbf{ACT}_\omega$.

Theorem

The fragment of $\mathbf{ACT}_\omega$ *without* $\mathbf{0}$ *is weakly complete w.r.t. regular SCL-models.*

Future Work

- Deal with constants.

Future Work

- Deal with constants.
- Concept lattice models on binary relations (also proposed by Wurm, as a version of R-models).

Future Work

- Deal with constants.
- Concept lattice models on binary relations (also proposed by Wurm, as a version of R-models).
- Study connections of SCL semantics with phase semantics (in particular, **0** is the only thing which makes a difference btw. intuitionistic and classical systems).

Future Work

- Deal with constants.
- Concept lattice models on binary relations (also proposed by Wurm, as a version of R-models).
- Study connections of SCL semantics with phase semantics (in particular, **0** is the only thing which makes a difference btw. intuitionistic and classical systems).
- Provide a concrete example for strong incompleteness w.r.t. regular SCL-models.

Some References

- J. Lambek. The mathematics of sentence structure. Amer. Math. Monthly 65, 154–170 (1958).
- R. Wille. Concept lattices and conceptual knowledge systems. Computers Math. Applic. 23(6–9), 493–515 (1992).
- C. Wurm. Language-theoretic and finite relation models for the (full) Lambek calculus. J. Logic Lang. Inform. 26(2), 179–214 (2017).
- S.L. Kuznetsov. Syntactic concept lattice models for infinitary action logic. In: G. Metcalfe et al. (eds). Logic, Language, Information, and Computation. WoLLIC 2024. Lect. Notes Comput. Sci. vol. 14672, Springer, 2024, pp. 93–107.

Thank you!