Algebras for automata: reasoning with regularity

Anupam Das

University of Birmingham

42^{nd} International Symposium on Theoretical Aspects of Computer Science

Jena, Germany
6^{th} March 2025

Based on joint work [DD24b, DD24c] with Abhishek De.

Outline

1. Systems for regular expressions
2. Inlining fixed points: algebras for automata
3. Cyclic proofs and completeness
4. Up to ω: reasoning about parity automata
5. Conclusions and further directions

Regular expressions over $\mathcal{A}$:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad a \in \mathcal{A} \quad | \quad e + f \quad | \quad ef \quad | \quad e^*$$

Write $\mathcal{L}(e) \subseteq \mathcal{A}^*$ for the regular language computed by e.

Regular expressions over $\mathcal{A}$:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad a \in \mathcal{A} \quad | \quad e + f \quad | \quad ef \quad | \quad e^*$$

Write $\mathcal{L}(e) \subseteq \mathcal{A}^*$ for the regular language computed by e.

Definition (Kleene Algebra)

A **Kleene Algebra** (KA) is an idempotent semiring equipped with an operation $*$ s.t.:

$$\begin{array}{c|c} e^* = 1 + ee^* & e^* = 1 + e^*e \\ ef \leqslant f \implies e^*f \leqslant f & ef \leqslant e \implies ef^* \leqslant f \end{array}$$

$(e \leqslant f := e + f = f)$

Regular expressions over $\mathcal{A}$:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad a \in \mathcal{A} \quad | \quad e + f \quad | \quad ef \quad | \quad e^*$$

Write $\mathcal{L}(e) \subseteq \mathcal{A}^*$ for the regular language computed by e.

Definition (Kleene Algebra)

A **Kleene Algebra** (KA) is an idempotent semiring equipped with an operation $*$ s.t.:

$$\begin{array}{c|c} e^* = 1 + ee^* & e^* = 1 + e^*e \\ ef \leqslant f \implies e^*f \leqslant f & ef \leqslant e \implies ef^* \leqslant f \end{array}$$

$(e \leqslant f := e + f = f)$

A **left-handed Kleene Algebra** (ℓKA) is defined like KA but without second column.

NB: in a ℓKA we have $e^* = \mathrm{LFP}\,[X \mapsto 1 + eX]$.

The image of $\mathcal{L}$ (i.e. the regular languages) forms a KA.

Examples

The image of $\mathcal{L}$ (i.e. the regular languages) forms a KA. The following are also KAs:

Algebra **Lang** of languages

The set of languages $\mathcal{P}(\mathcal{A}^*)$ where:

- 0 is $\varnothing$ and 1 is $\{\varepsilon\}$.
- $+$ is union and $\cdot$ is concatenation.
- $*$ is the usual Kleene $*$ of a language.

Algebra **Rel** of relations

The set of binary relations $\mathcal{P}(\mathcal{A} \times \mathcal{A})$ where:

- 0 is $\varnothing$ and 1 is Id.
- $+$ is union and $\cdot$ is composition.
- $*$ is reflexive transitive closure.

The image of $\mathcal{L}$ (i.e. the regular languages) forms a KA. The following are also KAs:

Algebra **Lang** of languages

The set of languages $\mathcal{P}(\mathcal{A}^*)$ where:

- 0 is $\varnothing$ and 1 is $\{\varepsilon\}$.
- $+$ is union and $\cdot$ is concatenation.
- $*$ is the usual Kleene $*$ of a language.

Algebra **Rel** of relations

The set of binary relations $\mathcal{P}(\mathcal{A} \times \mathcal{A})$ where:

- 0 is $\varnothing$ and 1 is Id.
- $+$ is union and $\cdot$ is composition.
- $*$ is reflexive transitive closure.

Proposition (Folklore)

Lang, **Rel** *and* $\mathcal{L}$ *have the same equational theory*.

Theorem ([Koz94, Kro90])

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{KA} \models e = f$

Theorem ([Koz94, Kro90])

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{KA} \models e = f$

- A celebrated but nontrivial result: Krob's proof is 137 pages!
- A crucial tool: $\ell\mathsf{KA}$ can solve 'right-linear systems' of (in)equations.

Theorem ([Koz94, Kro90])

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{KA} \models e = f$

- A celebrated but nontrivial result: Krob's proof is 137 pages!
- A crucial tool: $\ell\mathsf{KA}$ can solve 'right-linear systems' of (in)equations.

NB: this means the equational theory of **Rel** is decidable.

$\leadsto$ reasoning about *imperative programs*.

Theorem ([Koz94, Kro90])

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{KA} \models e = f$

- A celebrated but nontrivial result: Krob's proof is 137 pages!
- A crucial tool: $\ell\mathsf{KA}$ can solve 'right-linear systems' of (in)equations.

NB: this means the equational theory of **Rel** is decidable.

$\rightsquigarrow$ reasoning about *imperative programs*.

In fact, via Krob's argument, we can obtain a stronger result:

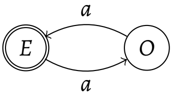
Theorem (Left-handed completeness [Bof95])

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \ell\mathsf{KA} \models e = f.$

Alternative (and shorter!) proofs are now known, [KS12, DDP18].

Digging deeper: NFAs as right-linear systems

An NFA can be construed as a (right-linear) system of inequalities. E.g.:

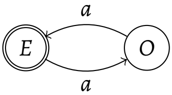


$\sim$

$$\begin{aligned} E &\geqslant 1 \\ E &\geqslant aO \\ O &\geqslant aE \end{aligned}$$

NB: $\begin{aligned} \mathcal{L}(E) &= \{a^{2n}\}_n \\ \mathcal{L}(O) &= \{a^{2n+1}\}_n \end{aligned}$ are the *least solutions* in $\mathcal{L}$.

An NFA can be construed as a (right-linear) system of inequalities. E.g.:



$\sim$

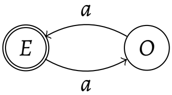
$$E \geqslant 1$$
$$E \geqslant aO$$
$$O \geqslant aE$$

NB: $\mathcal{L}(E) = \{a^{2n}\}_n$, $\mathcal{L}(O) = \{a^{2n+1}\}_n$ are the *least solutions* in $\mathcal{L}$.

Formalised Kleene Theorem

Every right-linear system has *least solutions* in any ℓKA.

An NFA can be construed as a (right-linear) system of inequalities. E.g.:



$$\sim \qquad \begin{aligned} E &\geqslant 1 \\ E &\geqslant aO \\ O &\geqslant aE \end{aligned}$$

NB: $\begin{aligned} \mathcal{L}(E) &= \{a^{2n}\}_n \\ \mathcal{L}(O) &= \{a^{2n+1}\}_n \end{aligned}$ are the *least solutions* in $\mathcal{L}$.

Formalised Kleene Theorem

Every right-linear system has *least solutions* in any ℓKA.

Motto: ℓKAs are *just* idempotent semirings with least solutions to NFAs.

Question

Can we accommodate NFAs natively within syntax?

The theory of idempotent semirings admits a natural proof calculus.

Sequents: $\Gamma \to e$, where Γ is a list of regular expressions (read $\prod \Gamma \leqslant e$)

Rules: *Lambek calculus* (equivalently, non-commutative IMALL)

The theory of idempotent semirings admits a natural proof calculus.

Sequents: $\Gamma \to e$, where Γ is a list of regular expressions (read $\prod \Gamma \leqslant e$)

Rules: *Lambek calculus* (equivalently, non-commutative IMALL)

Possibilities for $*$

[Jip04] Induction rule:
$$\frac{\Gamma \to f \quad e, f \to f}{e^*, \Gamma \to f}$$

Not complete without cut

[Pal07] ω-rule:
$$\frac{\Gamma \to f \quad e, \Gamma \to f \quad e, e, \Gamma \to f \quad \cdots}{e^*, \Gamma \to f}$$

Proofs are necessarily infinite

[DP17] 'Cyclic proofs' with unfoldings:
$$\frac{\Gamma \to f \quad e, e^*, \Gamma \to f}{e^*, \Gamma \to f}.$$

Not regular complete without cut

The only finitary approach we have is via complex 'hypersequents', $\Gamma \to S$ where now S is a set of lists (read $\prod \Gamma \leqslant \sum_{\Delta \in S} \prod \Delta$)

Theorem ([DP17])

There is a hypersequential calculus HKA *regularly complete for* $\mathcal{L}$. *(without cut)*

The only finitary approach we have is via complex 'hypersequents', $\Gamma \to S$ where now S is a set of lists (read $\prod \Gamma \leqslant \sum_{\Delta \in S} \prod \Delta$)

Theorem ([DP17])

There is a hypersequential calculus HKA *regularly complete for* $\mathcal{L}$. *(without cut)*

This can be extended to ω-regular languages too!

Definition

$A \subseteq \mathcal{A}^\omega$ is ω-**regular** if $A = \bigcup_{i<n} B_i C_i^\omega$, where all $B_i, C_i \subseteq \mathcal{A}^+$ are regular.

The only finitary approach we have is via complex 'hypersequents', $\Gamma \to S$ where now S is a set of lists (read $\prod \Gamma \leqslant \sum_{\Delta \in S} \prod \Delta$)

Theorem ([DP17])

There is a hypersequential calculus HKA *regularly complete for $\mathcal{L}$.* *(without cut)*

This can be extended to ω-regular languages too!

Definition

$A \subseteq \mathcal{A}^\omega$ is ω-**regular** if $A = \bigcup_{i<n} B_i C_i^\omega$, where all $B_i, C_i \subseteq \mathcal{A}^+$ are regular.

Theorem ([HK22])

There is an extension of HKA *regularly complete for ω-regular inclusions.* *(without cut)*

Outline

1. Systems for regular expressions
2. Inlining fixed points: algebras for automata
3. Cyclic proofs and completeness
4. Up to ω: reasoning about parity automata
5. Conclusions and further directions

(Right-linear) **μ-expressions**, $e, f, \ldots$, are generated by:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e$$

NB: there is no native product.

(Right-linear) μ-**expressions**, $e, f, \ldots$, are generated by:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e$$

NB: there is no native product.

Language semantics:

- $\mathcal{L}(\mu X e(X)) := \text{LFP}\,[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\implies$ $\mathcal{L}(e)$ is well-defined.

(Right-linear) μ-**expressions**, $e, f, \ldots$, are generated by:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e$$

NB: there is no native product.

Language semantics:

- $\mathcal{L}(\mu X e(X)) := \mathrm{LFP}\,[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\implies \mathcal{L}(e)$ is well-defined.

μ-expressions are a notation for NFAs:

- Expressions give rise to a canonical NFA.
- Right-linear systems can be solved by expressions via *Bekić's Lemma*.

(Right-linear) μ-expressions, $e, f, \ldots$, are generated by:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e$$

NB: there is no native product.

Language semantics:

- $\mathcal{L}(\mu X e(X)) := \mathrm{LFP}\,[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\implies$ $\mathcal{L}(e)$ is well-defined.

μ-expressions are a notation for NFAs:

- Expressions give rise to a canonical NFA.
- Right-linear systems can be solved by expressions via *Bekić's Lemma*.

Example

We can solve the previous systems for E, O in two (equivalent) ways:

$$\begin{aligned} E &= \mu X(1 + aaX) \\ O &= aE \end{aligned} \qquad \begin{aligned} O &= \mu X(a + aaX) \\ E &= 1 + aO \end{aligned}$$

Right-linear expressions admit a natural algebraic theory analogous to ℓKA:

Definition

A **right-linear algebra** (RLA) is a structure $\mathfrak{L} = (L, 0, 1, +, \mathcal{A})$ where:

- $(L, 0, +)$ is a bounded join-semilattice.
- Each $a \in \mathcal{A}$ is a bounded semilattice homomorphism.
- Each right-linear system has unique least solutions.

NB: no axioms for 1.

Right-linear expressions admit a natural algebraic theory analogous to ℓKA:

Definition

A **right-linear algebra (RLA)** is a structure $\mathfrak{L} = (L, 0, 1, +, \mathcal{A})$ where:

- $(L, 0, +)$ is a bounded join-semilattice.
- Each $a \in \mathcal{A}$ is a bounded semilattice homomorphism.
- Each right-linear system has unique least solutions.

NB: no axioms for 1.

Motto: RLAs are *just* join-semilattices with least solutions to NFAs.

Right-linear expressions admit a natural algebraic theory analogous to ℓKA:

Definition

A **right-linear algebra** (RLA) is a structure $\mathfrak{L} = (L, 0, 1, +, \mathcal{A})$ where:

- $(L, 0, +)$ is a bounded join-semilattice.
- Each $a \in \mathcal{A}$ is a bounded semilattice homomorphism.
- Each right-linear system has unique least solutions.

NB: no axioms for 1.

Motto: RLAs are *just* join-semilattices with least solutions to NFAs.

Example

Each ℓKA is a RLA, but not vice-versa!

- The structures **Lang** and **Rel** form RLAs in the expected way.
- $\mathcal{P}(\mathcal{A}^\omega)$ forms an RLA in the expected way, but not a ℓKA.

Outline

The absence of native products simplifies the resulting proof systems:

Sequents: $e \to \Gamma$, where Γ is a set of expressions (read $e \leqslant \sum \Gamma$)

The absence of native products simplifies the resulting proof systems:

Sequents: $e \to \Gamma$, where Γ is a set of expressions (read $e \leqslant \sum \Gamma$)

Non-logical rules:

$$\mathsf{id}\ \frac{}{e \to e} \qquad k_a\ \frac{e \to \Gamma}{ae \to a\Gamma} \qquad \mathsf{wk}\ \frac{e \to \Gamma}{e \to \Gamma, f}$$

Left logical rules:

$$0\text{-}l\ \frac{}{0 \to \Gamma} \qquad +\text{-}l\ \frac{e \to \Gamma \quad f \to \Gamma}{e + f \to \Gamma} \qquad \mu\text{-}l\ \frac{e(\mu X e(X)) \to \Gamma}{\mu X e(X) \to \Gamma}$$

Right logical rules:

$$0\text{-}r\ \frac{e \to \Gamma}{e \to \Gamma, 0} \qquad +\text{-}r\ \frac{e \to \Gamma, f_i}{e \to \Gamma, f_0 + f_1} \qquad \mu\text{-}r\ \frac{e \to \Gamma, f(\mu X f(X))}{e \to \Gamma, \mu X f(X)}$$

The absence of native products simplifies the resulting proof systems:

Sequents: $e \to \Gamma$, where Γ is a set of expressions (read $e \leqslant \sum \Gamma$)

Non-logical rules:

$$\mathsf{id}\ \frac{}{e \to e} \qquad k_a\ \frac{e \to \Gamma}{ae \to a\Gamma} \qquad \mathsf{wk}\ \frac{e \to \Gamma}{e \to \Gamma, f}$$

Left logical rules:

$$0\text{-}l\ \frac{}{0 \to \Gamma} \qquad +\text{-}l\ \frac{e \to \Gamma \quad f \to \Gamma}{e + f \to \Gamma} \qquad \mu\text{-}l\ \frac{e(\mu X e(X)) \to \Gamma}{\mu X e(X) \to \Gamma}$$

Right logical rules:

$$0\text{-}r\ \frac{e \to \Gamma}{e \to \Gamma, 0} \qquad +\text{-}r\ \frac{e \to \Gamma, f_i}{e \to \Gamma, f_0 + f_1} \qquad \mu\text{-}r\ \frac{e \to \Gamma, f(\mu X f(X))}{e \to \Gamma, \mu X f(X)}$$

NB: the fixed point rules do *not* guarantee leastness of μ...

Definition

- **Preproofs** are generated coinductively from the inference rules.
- A preproof is **cyclic/regular** if it has only finitely many distinct sub-preproofs.
- A **proof** is a preproof where each infinite branch has a 'good formula trace'.

Write CRLA for the class of cyclic $\widehat{\mathsf{RLA}}$ proofs.

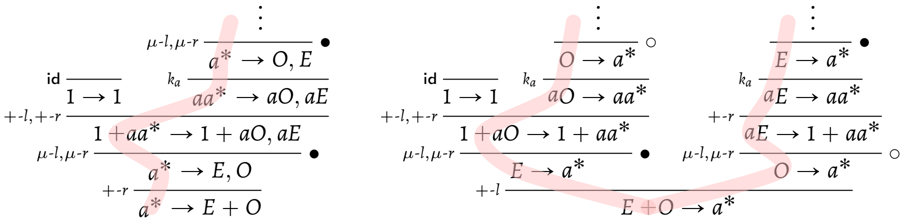
Definition

- **Preproofs** are generated coinductively from the inference rules.
- A preproof is **cyclic/regular** if it has only finitely many distinct sub-preproofs.
- A **proof** is a preproof where each infinite branch has a 'good formula trace'.

Write CRLA for the class of cyclic $\widehat{\mathsf{RLA}}$ proofs.

Example

Write $a^* := \mu X(1 + aX)$. We can show $a^* = E + O$:



Theorem (Soundness)

$\mathsf{CRLA} \vdash e \to f \implies \mathcal{L}(e) \subseteq \mathcal{L}(f)$.

Proof idea.

- For each $w \in \mathcal{L}(e)$ we take its finite 'run' along a cyclic proof.
- By induction on the run we show $w \in \mathcal{L}(f)$. □

Theorem (Soundness)
$\mathsf{CRLA} \vdash e \rightarrow f \implies \mathcal{L}(e) \subseteq \mathcal{L}(f)$.

Proof idea.

- For each $w \in \mathcal{L}(e)$ we take its finite 'run' along a cyclic proof.
- By induction on the run we show $w \in \mathcal{L}(f)$. □

Theorem (Regular completeness)
$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{CRLA} \vdash e \rightarrow f$.

Proof idea.

- Define a bottom-up validity-preserving proof search strategy.
- Critical loop-check for weakenings.
- analyticity $\rightsquigarrow$ finite state proof search $\rightsquigarrow$ regularity. □

Theorem (Soundness)

$\mathsf{CRLA} \vdash e \to f \implies \mathcal{L}(e) \subseteq \mathcal{L}(f)$.

Proof idea.

- For each $w \in \mathcal{L}(e)$ we take its finite 'run' along a cyclic proof.
- By induction on the run we show $w \in \mathcal{L}(f)$. □

Theorem (Regular completeness)

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{CRLA} \vdash e \to f$.

Proof idea.

- Define a bottom-up validity-preserving proof search strategy.
- Critical loop-check for weakenings.
- analyticity $\rightsquigarrow$ finite state proof search $\rightsquigarrow$ regularity. □

NB: This gives us an effective algorithm for proof search.

We can extract inductive invariants from cyclic proofs:

Theorem
$\mathsf{CRLA} \vdash e \rightarrow f \implies \mathsf{RLA} \vdash e \leqslant f$.

We can extract inductive invariants from cyclic proofs:

Theorem

$\mathsf{CRLA} \vdash e \to f \implies \mathsf{RLA} \vdash e \leqslant f$.

Proof idea.

- Inspired by previous approaches for $\ell\mathsf{KA}$ [KS12, DDP18].
- Can compute intersections of languages via right-linear systems.
  $\rightsquigarrow$ a product construction on NFAs.
- Appropriate local properties by analysis of cyclic proofs. □

We can extract inductive invariants from cyclic proofs:

Theorem
$\mathsf{CRLA} \vdash e \to f \implies \mathsf{RLA} \vdash e \leqslant f$.

Proof idea.

- Inspired by previous approaches for $\ell\mathsf{KA}$ [KS12, DDP18].
- Can compute intersections of languages via right-linear systems.
  $\rightsquigarrow$ a product construction on NFAs.
- Appropriate local properties by analysis of cyclic proofs. □

Corollary (Algebraic completeness)
$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \mathsf{RLA} \models e \leqslant f$.

Outline

Infinite words via greatest fixed points

We can reason about ω-regular languages via greatest fixed points:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e \quad | \quad \nu X e$$

NB: we omit 1 to limit to strictly infinite words, for simplicity.

Infinite words via greatest fixed points

We can reason about ω-regular languages via greatest fixed points:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e \quad | \quad \nu X e$$

NB: we omit 1 to limit to strictly infinite words, for simplicity.

Language semantics:

- $\mathcal{L}(\nu X e(X)) := \mathrm{GFP}[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\Longrightarrow$ $\mathcal{L}(e)$ is well-defined.

We can reason about ω-regular languages via greatest fixed points:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e \quad | \quad \nu X e$$

NB: we omit 1 to limit to strictly infinite words, for simplicity.

Language semantics:

- $\mathcal{L}(\nu X e(X)) := \mathrm{GFP}[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\implies$ $\mathcal{L}(e)$ is well-defined.

$\mu\nu$-expressions are a notation for parity automata (NDPA).

We can reason about ω-regular languages via greatest fixed points:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad X \quad | \quad ae \quad | \quad e+f \quad | \quad \mu X e \quad | \quad \nu X e$$

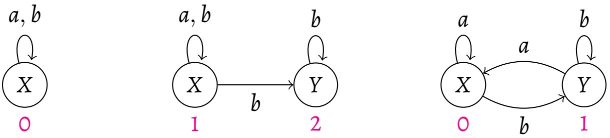
NB: we omit 1 to limit to strictly infinite words, for simplicity.

Language semantics:

- $\mathcal{L}(\nu X e(X)) := \mathrm{GFP}[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\implies$ $\mathcal{L}(e)$ is well-defined.

$\mu\nu$-expressions are a notation for parity automata (NDPA). E.g.:

$\top : \{a, b\}^\omega$ $\quad$ f_a : *"finitely many as"* $\quad$ i_a : *"infinitely many as"*



$\nu X(aX + bX)$ $\quad$ $\mu X(aX + bX + \nu Y(bY))$ $\quad$ $\nu X \mu Y(aX + bY)$

We can reason about ω-regular languages via greatest fixed points:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad X \quad | \quad ae \quad | \quad e + f \quad | \quad \mu X e \quad | \quad \nu X e$$

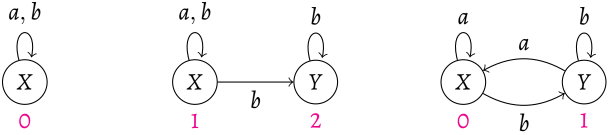
NB: we omit 1 to limit to strictly infinite words, for simplicity.

Language semantics:

- $\mathcal{L}(\nu X e(X)) := \mathrm{GFP}[A \mapsto \mathcal{L}(e(A))]$
- *Knaster-Tarski Theorem* $\implies$ $\mathcal{L}(e)$ is well-defined.

$\mu\nu$-expressions are a notation for parity automata (NDPA). E.g.:

$\top : \{a, b\}^\omega$ | f_a : *"finitely many as"* | i_a : *"infinitely many as"*



$\nu X(aX + bX)$ | $\mu X(aX + bX + \nu Y(bY))$ | $\nu X \mu Y(aX + bY)$

NB: equivalence between an expression and associated NDPA is not immediate...

Segue: the evaluation puzzle

Positions: pairs (w, e) where $w \in \mathcal{A}^\omega$ and e an expression.

Position	Available moves
(aw, ae)	(w, e)
$(w, e + f)$	$(w, e), (w, f)$
$(w, \mu Xe(X))$	$(w, e(\mu Xe(X)))$
$(w, \nu Xe(X))$	$(w, e(\nu Xe(X)))$

Winning: smallest expression occurring infinitely often is a ν-expression.

Positions: pairs (w, e) where $w \in \mathcal{A}^\omega$ and e an expression.

Position	**Available moves**
(aw, ae)	(w, e)
$(w, e + f)$	(w, e), (w, f)
$(w, \mu X e(X))$	$(w, e(\mu X e(X)))$
$(w, \nu X e(X))$	$(w, e(\nu X e(X)))$

Winning: smallest expression occurring infinitely often is a ν-expression.

Theorem (Adequacy)

$w \in \mathcal{L}(e) \iff$ *there is a winning play from* (w, e).

Proof idea.

- *Approximate* (non-)membership by ordinals ('signatures', cf. [SE89]).
- 'Least' signatures induce/exclude winning plays. □

Positions: pairs (w, e) where $w \in \mathcal{A}^\omega$ and e an expression.

Position	Available moves
(aw, ae)	(w, e)
$(w, e + f)$	(w, e), (w, f)
$(w, \mu X e(X))$	$(w, e(\mu X e(X)))$
$(w, \nu X e(X))$	$(w, e(\nu X e(X)))$

Winning: smallest expression occurring infinitely often is a ν-expression.

Theorem (Adequacy)

$w \in \mathcal{L}(e) \iff$ *there is a winning play from* (w, e).

Proof idea.

- *Approximate* (non-)membership by ordinals ('signatures', cf. [SE89]).
- 'Least' signatures induce/exclude winning plays. □

Corollary

A $\mu\nu$-expression and its associated NDPA compute the same ω-language.

νCRLA extends CRLA by the rules:

$$\nu\text{-}l\ \frac{e(\nu X e(X)) \to \Gamma}{\nu X e(X) \to \Gamma} \qquad \nu\text{-}r\ \frac{e \to \Gamma, f(\nu X f(X))}{e \to \Gamma, \nu X f(X)}$$

'Good formula traces' now given by winning plays (on RHS, or losing plays on LHS)

νCRLA extends CRLA by the rules:

$$\nu\text{-}l\ \frac{e(\nu X e(X)) \to \Gamma}{\nu X e(X) \to \Gamma} \qquad \nu\text{-}r\ \frac{e \to \Gamma, f(\nu X f(X))}{e \to \Gamma, \nu X f(X)}$$

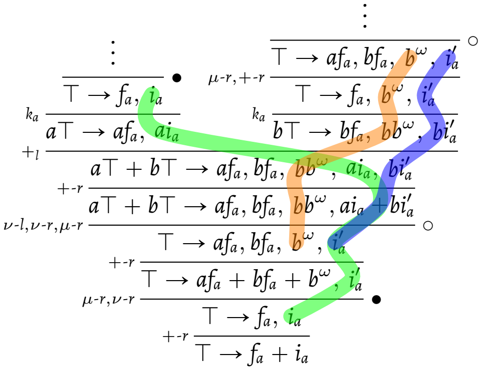
'Good formula traces' now given by winning plays (on RHS, or losing plays on LHS)

Theorem (Soundness)
$\nu\mathsf{CRLA} \vdash e \to f \implies \mathcal{L}(e) \subseteq \mathcal{L}(f)$.

Proof idea.
Let P be a νCRLA proof of $e \to f$.

- Suppose $w \in \mathcal{L}(e)$ and let π be a winning play from (w, e).
- π determines an infinite branch B_π of P.
- By construction, B_π in turn induces a winning play from (w, f). □

"Any ω-word over $\{a, b\}$ has finitely many as or infinitely many as"



Key

$b^\omega := \nu Y(bY)$
$i'_a := \mu Y(ai_a + bY)$

Correctness

$-\circ^\omega$: orange trace good
$-\bullet^\omega$: green trace good
$(-\bullet-\circ)^\omega$: green/blue trace good

Construe proof search as a 2-player game between `Prover` and `Refuter`.

Construe proof search as a 2-player game between `Prover` and `Refuter`.

By ω-regularity of the correctness condition for νCRLA:

Lemma (cf. Büchi-Landweber)

The proof search game for νCRLA is finite memory determined.

Construe proof search as a 2-player game between `Prover` and `Refuter`.

By ω-regularity of the correctness condition for νCRLA:

Lemma (cf. Büchi-Landweber)

The proof search game for νCRLA is finite memory determined.

Theorem (Completeness)

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \nu\mathsf{CRLA} \vdash e \to f$ *(for e, f guarded)*

Construe proof search as a 2-player game between `Prover` and `Refuter`.

By ω-regularity of the correctness condition for νCRLA:

Lemma (cf. Büchi-Landweber)

The proof search game for νCRLA is finite memory determined.

Theorem (Completeness)

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \implies \nu\mathsf{CRLA} \vdash e \to f$ *(for e, f guarded)*

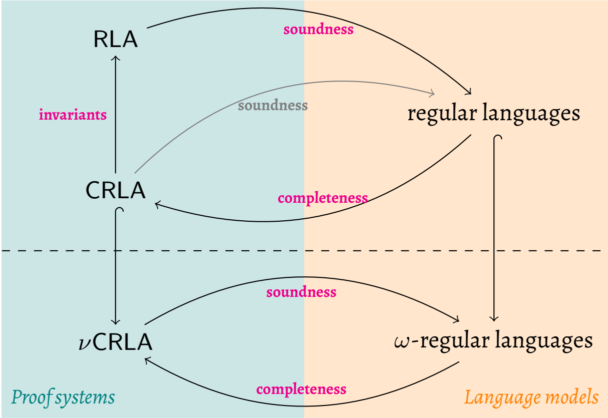
Proof idea.

- Similar proof search strategy to CRLA, but guardedness $\implies$ fairness.
- $\nu\mathsf{CRLA} \nvdash e \to f \implies$ there is a 'bad branch' of proof search, by determinacy.
- By reduction to Evaluation Puzzle, we extract a word $w \in \mathcal{L}(e) \setminus \mathcal{L}(f)$. □

Outline

1. Systems for regular expressions
2. Inlining fixed points: algebras for automata
3. Cyclic proofs and completeness
4. Up to ω: reasoning about parity automata
5. Conclusions and further directions

Summary



RLA
soundness
invariants
soundness
regular languages
CRLA
completeness
soundness
νCRLA
ω-regular languages
completeness
Proof systems
Language models

Towards Alternating Parity Automata: Right-Linear Lattices (w.i.p.)

We can go futher and consider a fully dualised syntax:

$$
\begin{array}{rcllllll}
e, f, \dots & ::= & X & \mid\ ae & \mid\ 0 & \mid\ e + f & \mid\ \mu X\, e \\
& & & & \mid\ \top & \mid\ e \cap f & \mid\ \nu X\, e
\end{array}
$$

This comprises a notation for Alternating Parity Automata (APA).

We can go futher and consider a fully dualised syntax:

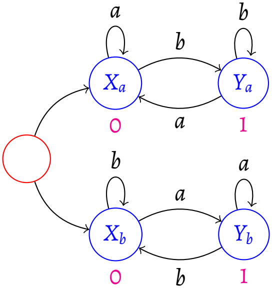
$$
\begin{array}{lclcl}
e, f, \ldots & ::= & X \;|\; ae & | & 0 \;|\; e+f \;|\; \mu X\, e \\
 & & & | & \top \;|\; e \cap f \;|\; \nu X\, e
\end{array}
$$

This comprises a notation for Alternating Parity Automata (APA). E.g.:

$\nu X_a \mu Y_a \, (aX_a + bY_a)$

$\cap$

$\nu X_b \mu Y_b \, (aY_b + bX_b)$



Key

- ◯ : universal state
- ◯ : existential state
- n : priority

We can go futher and consider a fully dualised syntax:

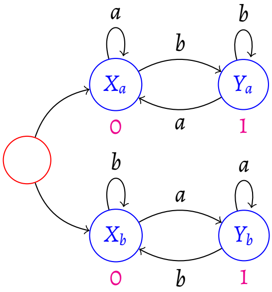
$$e, f, \ldots \quad ::= \quad X \;\mid\; ae \;\mid\; 0 \;\mid\; e+f \;\mid\; \mu X\, e$$
$$\qquad\qquad\qquad\qquad\qquad\quad \mid\; \top \;\mid\; e \cap f \;\mid\; \nu X\, e$$

This comprises a notation for Alternating Parity Automata (APA). E.g.:

$$\nu X_a \mu Y_a\, (aX_a + bY_a)$$

$$\cap$$

$$\nu X_b \mu Y_b\, (aY_b + bX_b)$$



Key

- ◯ : universal state
- ◯ : existential state
- n : priority

NB: the resulting theory is much more symmetric!

We can go futher and consider a fully dualised syntax:

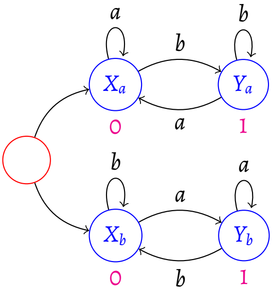
$$e, f, \ldots \quad ::= \quad X \mid ae \mid 0 \mid e+f \mid \mu X\, e$$
$$\mid \top \mid e \cap f \mid \nu X\, e$$

This comprises a notation for Alternating Parity Automata (APA). E.g.:

$$\nu X_a \mu Y_a\, (aX_a + bY_a)$$

$$\cap$$

$$\nu X_b \mu Y_b\, (aY_b + bX_b)$$



Key

- ◯ : universal state
- ◯ : existential state
- n : priority

NB: the resulting theory is much more symmetric!

Theorem

- CRLA *has an extension complete for APA expressions.*
- RLA *has an extension complete for APA expressions.* ⇝ ***Right-Linear Lattices***

We can combine all the linguistic features we have seen so far:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad a \quad | \quad e+f \quad | \quad ef \quad | \quad \mu X e \quad | \quad \nu X e$$

Such expressions compute just the $\leqslant \omega$-**context-free** languages.

We can combine all the linguistic features we have seen so far:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad a \quad | \quad e+f \quad | \quad ef \quad | \quad \mu X e \quad | \quad \nu X e$$

Such expressions compute just the $\leqslant \omega$-**context-free** languages.

We obtain an extension $\mu\nu\ell$HKA of HKA with:

Theorem (Assuming $\exists 0\#$)

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \iff \mu\nu\ell\mathsf{HKA}$ *has a (not necessarily regular) proof of* $e \to f$.

- Note that proofs are inherently irregular! Universality of CFLs is not r.e.
- We thus need analytic determinacy for completeness, which is beyond ZFC.

We can combine all the linguistic features we have seen so far:

$$e, f, \ldots \quad ::= \quad 0 \quad | \quad 1 \quad | \quad X \quad | \quad a \quad | \quad e+f \quad | \quad ef \quad | \quad \mu X e \quad | \quad \nu X e$$

Such expressions compute just the $\leqslant \omega$-**context-free** languages.

We obtain an extension $\mu\nu\ell$HKA of HKA with:

Theorem (Assuming $\exists 0\#$)

$\mathcal{L}(e) \subseteq \mathcal{L}(f) \iff \mu\nu\ell$HKA *has a (not necessarily regular) proof of* $e \to f$.

- Note that proofs are inherently irregular! Universality of CFLs is not r.e.
- We thus need analytic determinacy for completeness, which is beyond ZFC.

Corollary

Completeness for an infinitary axiomatisation of CFLs (cf. [GH13]).

Perspectives

- Equational theories of automata via fixed point notations.
- Cyclic proofs a natural and powerful technique.
- Right-linear syntax much more amenable to proof theory.
- Completeness for regular languages is independent of multiplication!

Perspectives

- Equational theories of automata via fixed point notations.
- Cyclic proofs a natural and powerful technique.
- Right-linear syntax much more amenable to proof theory.
- Completeness for regular languages is independent of multiplication!

Future directions

- Further models e.g. visibly pushdown and tree automata.
- What about computational interpretations of proofs?

  NB: beware non-constructivity!

Perspectives

- Equational theories of automata via fixed point notations.
- Cyclic proofs a natural and powerful technique.
- Right-linear syntax much more amenable to proof theory.
- Completeness for regular languages is independent of multiplication!

Future directions

- Further models e.g. visibly pushdown and tree automata.
- What about computational interpretations of proofs?

  NB: beware non-constructivity!

THANK YOU.

References I

M. Boffa.
Une condition impliquant toutes les identités rationnelles.
RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications, 29(6):515–518, 1995.

Anupam Das and Abhishek De.
A proof theory of (ω-)context-free languages, via non-wellfounded proofs.
In Christoph Benzmüller, Marijn J. H. Heule, and Renate A. Schmidt, editors, *Automated Reasoning - 12th International Joint Conference, IJCAR 2024, Nancy, France, July 3-6, 2024, Proceedings, Part II*, volume 14740 of *Lecture Notes in Computer Science*, pages 237–256. Springer, 2024.

Anupam Das and Abhishek De.
A proof theory of right-linear (ω-)grammars via cyclic proofs.
In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 30:1–30:14. ACM, 2024.

Anupam Das and Abhishek De.
A proof theory of right-linear (omega-)grammars via cyclic proofs, 2024.

Anupam Das, Amina Doumane, and Damien Pous.

Left-handed completeness for kleene algebra, via cyclic proofs.

In Gilles Barthe, Geoff Sutcliffe, and Margus Veanes, editors, *LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 57 of *EPiC Series in Computing*, pages 271–289, Awassa, Ethiopia, 2018. EasyChair.

Anupam Das and Damien Pous.

A cut-free cyclic proof system for kleene algebra.

In Renate A. Schmidt and Cláudia Nalon, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 26th International Conference, TABLEAUX 2017, Brasília, Brazil, September 25-28, 2017, Proceedings*, volume 10501 of *Lecture Notes in Computer Science*, pages 261–277, Brazil, 2017. Springer.

Niels Bjørn Bugge Grathwohl, Fritz Henglein, and Dexter .

Infinitary axiomatization of the equational theory of context-free languages.

Electronic Proceedings in Theoretical Computer Science, 126:44–55, August 2013.

Emile Hazard and Denis Kuperberg.

Cyclic proofs for transfinite expressions.

In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference)*, volume 216 of *LIPIcs*, pages 23:1–23:18, Göttingen, Germany, 2022. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Peter Jipsen.

From semirings to residuated kleene lattices.

Studia Logica, 76:291–303, 03 2004.

D. Kozen.

A completeness theorem for kleene algebras and the algebra of regular events.

Information and Computation, 110(2):366–390, 1994.

Daniel Krob.

A complete system of b-rational identities.

In Michael S. Paterson, editor, *Automata, Languages and Programming*, pages 60–73, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

References IV

Dexter Kozen and Alexandra Silva.
Left-handed completeness.
In Wolfram Kahl and Timothy G. Griffin, editors, *Relational and Algebraic Methods in Computer Science*, pages 162–178, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

Ewa Palka.
An infinitary sequent system for the equational theory of *-continuous action lattices.
Fundamenta Informaticae, 78(2):295–309, 2007.

Robert S. Streett and E. Allen Emerson.
An automata theoretic decision procedure for the propositional mu-calculus.
Inf. Comput., 81(3):249–264, 1989.