Cyclic Proofs and jumping automata

Denis Kuperberg  Laureline Pinault  Damien Pous

LIP, ENS Lyon

Journées 2019 du GT Scalp
Lyon

Friday 18th October 2019
Cyclic proofs

Regular expressions

\[ e, f ::= 1 \mid a \in A \mid e \cdot f \mid e + f \mid e^* \]

Context: Cyclic proofs for inclusion of expressions [Das, Pous ’17]

- Infinite proof trees, with root of the form \( e \vdash f \).

\[
\begin{array}{cccc}
1 \vdash 1 & (Ax) & a \vdash a & (Ax) \\
1 \vdash a^* & a \vdash a & a^* \vdash a^* \\
1 \vdash a^* & a, a^* \vdash a^* & a^* \vdash a^*
\end{array}
\]
Cyclic proofs

Regular expressions

\[ e, f ::= 1 \mid a \in A \mid e \cdot f \mid e + f \mid e^* \]

Context: Cyclic proofs for inclusion of expressions [Das, Pous ’17]

- Infinite proof trees, with root of the form \( e \vdash f \).

\[
\begin{align*}
1 \vdash 1 \quad &\text{(Ax)} \\
1 \vdash a^* \\
\hline
a \vdash a \quad &\text{(Ax)} \\
\hline
a^* \vdash a^* \\
\hline
a, a^* \vdash a^* \\
\hline
a^* \vdash a^*
\end{align*}
\]

- Validity condition on infinite branches
Cyclic proofs

Regular expressions

\[
e, f := 1 \mid a \in A \mid e \cdot f \mid e + f \mid e^*\]

Context: Cyclic proofs for inclusion of expressions \[\text{[Das, Pous '17]}\]

- Infinite proof trees, with root of the form \(e \vdash f\).

\[
\begin{align*}
1 \vdash 1 \quad &\text{(Ax)} \\
1 \vdash a^* \\
a \vdash a \quad &\text{(Ax)} \\
a^* \vdash a^* \\
a, a^* \vdash a^* \\
a^* \vdash a^*
\end{align*}
\]

- Validity condition on infinite branches

- \(\exists\) proof of \(e \vdash f \iff L(e) \subseteq L(f)\).
Computational interpretation

Proof of \( e \vdash f \)

Program with input from \( e \) and output in \( f \).
Computational interpretation

Proof of $e \vdash f$

Program with input from $e$ and output in $f$.

Several proofs of the same statement

$\uparrow$

Several programs of the same type

Example:

$\overline{a \vdash a + a}$

$in_l$ or $in_r$
Computational interpretation

Proof of $e \vdash f$

Program with input from $e$ and output in $f$.

Several proofs of the same statement

\[ a \vdash a + a \]

Several programs of the same type

Example: $in_l$ or $in_r$

Curry-Howard isomorphism, typed programming,…

Well-understood for finite proofs, active field for infinite proofs.
This work

- Boolean type $2 = 1 + 1$

- Add *structural* rules corresponding to simple natural programs

- Study the expressive power of regular proofs (finite graphs)

- Focus on proofs for languages:

  \[ \text{Proof } \pi \text{ of } A^* \vdash 2 \longrightarrow \text{Language } L(\pi) \subseteq A^* \]
Proof system

Expressions \( e := A \mid A^* \)

Sequents \( E, F = e_1, e_2, \ldots, e_n \)

Proof system with extra rules for basic data manipulation:

\[
\begin{array}{c}
E, F \vdash 2 \\
\hline
E, e, F \vdash 2 \quad \text{(wkn)}
\end{array}
\]

\[
\begin{array}{c}
E, F \vdash 2 \\
\hline
E, e, e, F \vdash 2 \quad \text{(ctr)}
\end{array}
\]

\[
\begin{array}{c}
(E, F \vdash 2)_{a \in A} \\
\hline
E, A, F \vdash 2 \quad \text{(A)}
\end{array}
\]

\[
\begin{array}{c}
E, F \vdash 2 \\
\hline
E, A, A^*, F \vdash 2 \quad \text{(*)}
\end{array}
\]

\[
\begin{array}{c}
E, F \vdash 2 \\
\hline
E, A^*, F \vdash 2
\end{array}
\]
Proofs as language acceptors

What are the languages computed by cyclic proofs?

Example on alphabet \( \{a, b\} \): \( b^* \)

\[
\begin{align*}
\vdash 2 \quad (\text{tt}) \\
\vdash 2 \quad (f\bar{f}) \\
(A^* \vdash 2)_a \quad (\text{wkn}) \\
(A^* \vdash 2)_b \quad (A) \\
A, A^* \vdash 2 \\
A^* \vdash 2 \quad (\ast)
\end{align*}
\]
Proofs as language acceptors

What are the languages computed by cyclic proofs?

Example on alphabet \( \{a, b\} \): \( b^* \)

\[
\begin{align*}
\vdash 2 & \quad \text{(tt)} \\
(\vdash 2)_{a} & \quad \text{(ff)} \\
(\vdash 2)_{b} & \quad \text{(wkn)} \\
\vdash 2 & \quad \text{(A)} \\
(\vdash 2)_{a} & \quad \text{(wkn)} \\
\vdash 2 & \quad \text{(A)} \\
\vdash 2 & \quad \text{(A)} \\
A^* & \vdash 2
\end{align*}
\]

Lemma

*Without contraction, the system captures exactly regular languages.*
With contractions: what class of language?

Example on alphabet \{a, b\}: \textit{a}^n\textit{b}^n

1st step: create a copy of the input and delete the first \textit{a}'s.
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

1st step: create a copy of the input and delete the first \( a \)'s.
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

1st step: create a copy of the input and delete the first \( a \)'s.
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

\[
\begin{array}{c}
A^* \vdash 2 \\
(\text{wkn})
\end{array}
\]

\[
\begin{array}{c}
(\text{ff}) \\
(\text{wkn})
\end{array}
\]

2nd step: check that for each \( b \) of the second copy we have an \( a \) in the first one.
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

2nd step: check that for each \( b \) of the second copy we have a \( a \) in the first one.
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

2nd step: check that for each \( b \) of the second copy we have a \( a \) in the first one.
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

2nd step: check that for each \( b \) of the second copy we have a \( a \) in the first one.
With contractions: what class of language?

Example on alphabet \{a, b\}: \textit{a}^n\textit{b}^n

3rd step: checking that we have no more \textit{a}'s
With contractions: what class of language?

Example on alphabet \( \{a, b\} \): \( a^n b^n \)

\[
\begin{align*}
\vdash 2 \quad (\text{ff}) \quad \frac{\quad \vdash 2 \quad (\text{ff}) \quad (\text{wkn}) \quad \vdash 2 \quad (\text{tt}) \quad (\text{wkn})}{(A^* \vdash 2)_a} & \quad \frac{(A^* \vdash 2)_b}{(A^* \vdash 2)} \quad (A)
\end{align*}
\]

3rd step: checking that we have no more \( a \)'s

Example on alphabet \( \{a, b\} \): \( a^n b^n c^n \)
With contractions: a new automaton model

Jumping Multihead Automata

A JMA is an automaton with $k$ reading heads.

Transitions: $Q \times A^k \rightarrow Q \times \{\text{(step forward), } \text{spinner}, J_1, \ldots, J_k\}^k$

- $\text{step forward}$: advance one step
- $\text{spinner}$: stay in place
- $J_i$: jump to the position of head $i$
A JMA is an automaton with $k$ reading heads.

Transitions: $Q \times A^k \rightarrow Q \times \{\text{✓}, \text{_spinner}, J_1, \ldots, J_k\}^k$

- ✓: advance one step
- Spinner: stay in place
- $J_i$: jump to the position of head $i$

+ Equivalent of the validity criterion
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{ a^{2n} \mid n \in \mathbb{N} \} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{ a^{2^n} \mid n \in \mathbb{N} \} \) is accepted by a 2-head JMA.
Example of JMA

Example: $\{a^{2^n} \mid n \in \mathbb{N}\}$ is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: $\{a^{2^n} \mid n \in \mathbb{N}\}$ is accepted by a 2-head JMA.
Example of JMA

Example: \( \{ a^{2n} \mid n \in \mathbb{N} \} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{ a^{2n} \mid n \in \mathbb{N} \} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N} \} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: $\{a^{2^n} \mid n \in \mathbb{N}\}$ is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) is accepted by a 2-head JMA.
Example of JMA

Example: $\{a^{2n} \mid n \in \mathbb{N}\}$ is accepted by a 2-head JMA.
Expressive power of JMA

Comparison with Multihead Automata in Literature:
[Holzer, Kutrib, Malcher 2008]

1-way Multihead \subseteq JMA \subseteq 2-way Multihead

\text{Emptiness Undecidable}\quad \forall k, \text{JMA}(2) \not\subseteq 1\text{DFA}(k)
Expressive power of JMA

Comparison with Multihead Automata in Literature:
[Holzer, Kutrib, Malcher 2008]

1-way Multihead $\subseteq$ JMA $\subseteq$ 2-way Multihead
Expressive power of JMA

Comparison with Multihead Automata in Literature:
[Holzer, Kutrib, Malcher 2008]

\[
\begin{align*}
1\text{-way Multihead} & \subseteq \text{JMA} & \subseteq 2\text{-way Multihead} \\
\text{Emptiness Undecidable} & & \text{DLogSpace}
\end{align*}
\]
Expressive power of JMA

Comparison with Multihead Automata in Literature:
[Holzer, Kutrib, Malcher 2008]

∀k, \( JMA(2) \not\subseteq 1DFA(k) \)

1-way Multihead \( \subseteq \) JMA \( \subseteq \) 2-way Multihead

Emptiness Undecidable

DLogSpace
Expressive power of JMA

Comparison with Multihead Automata in Literature:
[Holzer, Kutrib, Malcher 2008]

∀k, \( \text{JMA}(2) \nsubseteq \text{1DFA}(k) \)

1-way Multihead \( \subseteq \) JMA \( \subseteq \) 2-way Multihead

Emptiness Undecidable

\( \quad \) DLogSpace
Main result

Theorem

*Cyclic proofs and JMA recognize the same class of languages.*

- States of the automaton $\sim$ Positions in the proof tree
- Accepting / Rejecting state $\sim$ True / False axiom
- Multiple heads $\sim$ Multiple copies of $A^*$
- Reading a letter $\sim$ Applying $\ast$ and $(A)$ rules
What next?

- Add the cut rule
- Corresponds to composition of functions
- Sequents \((1^*)^k \vdash 1^*\): functions \(\mathbb{N}^k \rightarrow \mathbb{N}\)

**Work in progress:**

\[
\begin{align*}
\text{No contraction} & \quad = \quad \text{Primitive Recursive} \\
\text{With contraction} & \quad = \quad \text{System T}
\end{align*}
\]
What next?

- Add the cut rule
- Corresponds to composition of functions
- Sequents $(1^*)^k \vdash 1^*$: functions $\mathbb{N}^k \rightarrow \mathbb{N}$

Work in progress:

No contraction \quad = \quad Primitive Recursive

With contraction \quad = \quad System T

Thank you for your attention!

[Denis Kuperberg, Laureline Pinault and Damien Pous, FSTTCS 19]