Cyclic Proofs and jumping automata

Denis Kuperberg    Laureline Pinault    Damien Pous

LIP, ENS Lyon

Séminaire MOVE
Marseille

Thursday 7th November 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 \( e \vdash f \).

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

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

- 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 [Das, Pous ’17]

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

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

- Validity condition on infinite branches

\[ \exists \text{ proof of } e \vdash f \iff L(e) \subseteq L(f). \]
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 \( \Leftrightarrow \)

Several programs of the same type

Example:

\[
\begin{align*}
    a & \vdash a + a \\
    \text{in}_l \text{ or } \text{in}_r
\end{align*}
\]
Computational interpretation

Proof of $e \vdash f$

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

Several proofs of the same statement

$\iff$

Several programs of the same type

Example:

\[
a \vdash a + a
\]

$in_l$ 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*:

  Proof $\pi$ of $A^* \vdash 2$ $\rightarrow$ 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:

$\frac{}{\vdash 2}$ (tt)

$\frac{E, F \vdash 2}{E, e, F \vdash 2}$ (wkn)

$\frac{E, e, e, F \vdash 2}{E, e, F \vdash 2}$ (ctr)

$\frac{(E, F \vdash 2)_{a \in A}}{E, A, F \vdash 2}$ (A)

$\frac{E, F \vdash 2}{E, A, A^*, F \vdash 2}$

$\frac{E, F \vdash 2}{E, A^*, F \vdash 2}$ ($\ast$)
Proofs as language acceptors

What are the languages computed by cyclic proofs?

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

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

What are the languages computed by cyclic proofs?

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

```
\[
\begin{array}{c}
\begin{array}{c}
\vdash 2 \\
\hline
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
\vdash 2 \\
\hline
(\text{f}f)
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(\text{w}kn)
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(\text{A}^* \vdash 2)_a
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(\text{A}^* \vdash 2)_b
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(A)
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(\text{A} \cdot \text{A}^* \vdash 2)_{(\text{tt})}
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(\text{A} \cdot \text{A}^* \vdash 2)_{(\text{A})}
\end{array}
\end{array}
\]
\[
\begin{array}{c}
\begin{array}{c}
(\text{A} \cdot \text{A}^* \vdash 2)_{(*)}
\end{array}
\end{array}
\]
```

Lemma

*Without contraction, the system captures exactly regular languages.*
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 \(\{a, b\}\): \(a^n b^n\)

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

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

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 \( \{a, b\} \): \( a^n b^n \)

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

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 \( \{a, b\} \): \( a^n b^n \)

\[
\frac{\vdash 2}{(A^* \vdash 2)_a} \quad \frac{\vdash 2}{(A^* \vdash 2)_b} \quad \frac{A, A^* \vdash 2}{A^* \vdash 2}
\]

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

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

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

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

Jumping Multihead Automata

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

**Transitions:** $Q \times (A \cup \{\leftarrow\})^k \rightarrow Q \times \{\text{\textbullet}, \text{\textcircled{C}}, J_1, \ldots, J_k\}^k$

- \text{\textbullet}: advance one step
- \text{\textcircled{C}}: stay in place
- $J_i$: jump to the position of head $i$
With contractions: a new automaton model

Jumping Multihead Automata

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

Transitions: $Q \times (A \cup \{\langle\})^k \rightarrow Q \times \{\uparrow, \bigcirc, J_1, \ldots, J_k\}^k$

- $\uparrow$: advance one step
- $\bigcirc$: stay in place
- $J_i$: jump to the position of head $i$

$+$ Equivalent of the validity criterion
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: \( \{a^{2^n} \mid n \in \mathbb{N}\} \) 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.
### Equivalence Theorem

**Theorem**

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

<table>
<thead>
<tr>
<th>States of the automaton</th>
<th>Positions in the proof tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepting / Rejecting state</td>
<td>True / False axiom</td>
</tr>
<tr>
<td>Multiple heads</td>
<td>Multiple copies of $A^*$</td>
</tr>
<tr>
<td>Reading a letter</td>
<td>Applying $*$ and $(A)$ rules</td>
</tr>
</tbody>
</table>
Expressive power of JMA

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

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

Emptiness Undecidable

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

\text{e.g. Palindroms?}
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]

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

Emptiness Undecidable

LogSpace
Expressive power of JMA

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

\[ \forall k, \ JMA(2) \not\subseteq 1DFA(k) \]

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

Emptiness Undecidable

LogSpace
Expressive power of JMA

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

\( \forall k, \ JMA(2) \not\subset 1DFA(k) \Rightarrow ? \)

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

Emptiness Undecidable

\( \text{LogSpace} \)
Expressive power of JMA

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

\[ \forall k, \ JMA(2) \not\subseteq 1DFA(k) \]

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

Emptiness Undecidable

\text{LogSpace}
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

---

| Difficulty: simulate a left move of some head. |
| Example: Palindroms $= \{ u \in \Sigma^* | u = u^R \}$ is accepted by a JMA. |

---

D. Kuperberg  
Cyclic Proofs and jumping automata  
Thursday 7th November 2019  
12 / 13
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.
Simulating 2-ways automata

Theorem (unpublished)

JMA have same expressive power as 2-way Multihead automata.

Difficulty: simulate a left move of some head.

Example: Palindroms = \( \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

Example: *Palindroms* = \( \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** *Palindroms* $= \{ u \in \Sigma^* \mid u = u^R \}$ is accepted by a JMA.

\[ \begin{array}{c}
\uparrow & a & a & b & c & c & b & a & a & \downarrow \\
\end{array} \]
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** *Palindroms* $= \{ u \in \Sigma^* \mid u = u^R \}$ is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

Example: \( \text{Palindroms} = \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

JMA have same expressive power as 2-way Multihead automata.

Difficulty: simulate a left move of some head.

Example: Palindroms = \{u \in \Sigma^* \mid u = u^R\} is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

Example: *Palindroms* $\{u \in \Sigma^* \mid u = u^R\}$ is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** *Palindroms* $= \{ u \in \Sigma^* \mid u = u^R \}$ is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** \( \text{Palindroms} = \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.

\[
\begin{array}{ccccccc}
\triangleright & a & a & b & c & c & b & a & a & \triangleright \\
\end{array}
\]
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** $\text{Palindroms} = \{ u \in \Sigma^* \mid u = u^R \}$ is accepted by a JMA.

- $a \quad a \quad b \quad c \quad c \quad b \quad a \quad a \quad \leftarrow$
- $\downarrow$
- $\uparrow$
- $\uparrow$

D. Kuperberg

Cyclic Proofs and jumping automata

Thursday 7th November 2019

12 / 13
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** *Palindroms* = \( \{u \in \Sigma^* \mid u = u^R\} \) is accepted by a JMA.

\[ \begin{array}{ccccccccc}
\triangleright & a & a & b & c & c & b & a & a & \downarrow \\
\end{array} \]
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

 Difficulty: simulate a left move of some head.

**Example:** *Palindroms* = \( \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** $\textit{Palindroms} = \{ u \in \Sigma^* \mid u = u^R \}$ is accepted by a JMA.

![Diagram of JMA simulation](image)
Simulating 2-ways automata

Theorem (unpublished)

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

Example: *Palindroms* = \{ u ∈ Σ* | u = u^R \} is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

**JMA have same expressive power as 2-way Multihead automata.**

Difficulty: simulate a left move of some head.

Example: *Palindroms* = \{u ∈ Σ* | u = u^R\} is accepted by a JMA.

![Diagram of JMA simulating a left move](image-url)
Simulating 2-ways automata

Theorem (unpublished)

JMA have same expressive power as 2-way Multihead automata.

Difficulty: simulate a left move of some head.

Example: Palindroms = \{ u \in \Sigma^* \mid u = u^R \} is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** Palindroms = \{ u \in \Sigma^* \mid u = u^R \} is accepted by a JMA.

\[
\begin{array}{cccccccc}
\Large{\triangleright} & a & a & b & c & c & b & a & a \Large{\triangleleft}
\end{array}
\]
Simulating 2-ways automata

Theorem (unpublished)

JMA have same expressive power as 2-way Multihead automata.

Difficulty: simulate a left move of some head.

Example: Palindroms = \{u \in \Sigma^* \mid u = u^R\} is accepted by a JMA.

\[\begin{array}{cccccccc}
\triangleright & a & a & b & c & c & b & a & a \\
\end{array}\]
Simulating 2-ways automata

Theorem (unpublished)

JMA have same expressive power as 2-way Multihead automata.

Difficulty: simulate a left move of some head.

Example: Palindroms = \{u \in \Sigma^* \mid u = u^R\} is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

\textit{JMA have same expressive power as 2-way Multihead automata.}

Difficulty: simulate a left move of some head.

**Example**: \textit{Palindroms} = \{\(u \in \Sigma^* \mid u = u^R\}\} is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

*JMA have same expressive power as 2-way Multihead automata.*

**Difficulty:** simulate a left move of some head.

**Example:** *Palindroms* $= \{ u \in \Sigma^* \mid u = u^R \}$ is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** *Palindroms* = \( \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.
Simulating 2-ways automata

**Theorem (unpublished)**

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

**Example:** *Palindroms* = \( \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

\textit{JMA have same expressive power as 2-way Multihead automata.}

Difficulty: simulate a left move of some head.

Example: \textit{Palindroms} = \{u \in \Sigma^* \mid u = u^R\} is accepted by a JMA.
Simulating 2-ways automata

Theorem (unpublished)

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

Example: *Palindroms* = \( \{ u \in \Sigma^* \mid u = u^R \} \) is accepted by a JMA.

\[
\begin{array}{cccccccc}
\triangleright & a & a & b & c & c & b & a & a
\end{array}
\]

Generalization of this idea \( \Rightarrow \) Translation from 2DFA to JMA.
Simulating 2-ways automata

Theorem (unpublished)

*JMA have same expressive power as 2-way Multihead automata.*

Difficulty: simulate a left move of some head.

Example: \( \text{Palindroms} = \{u \in \Sigma^* \mid u = u^R\} \) is accepted by a JMA.

\[
\begin{array}{cccccc}
\triangleright & a & a & b & c & c & b & a & a & \triangleleft \\
\end{array}
\]

Generalization of this idea \( \Rightarrow \) Translation from 2DFA to JMA.

Corollary

*Cyclic proofs with contraction characterize \( \text{LogSpace} \).*
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 \(\Rightarrow\) Primitive Recursive
With contraction \(\Rightarrow\) System T \(=\) Peano
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:

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

Thank you for your attention!

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