Cyclic Proofs and jumping automata

Denis Kuperberg  Laureline Pinault  Damien Pous

LIP, ENS Lyon

Highlights of Logic, Games and Automata
Warsaw

Friday 20th 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 \( e \vdash f \).

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

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

- 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** \( e \vdash f \).

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

- **Validity condition on infinite branches**

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

Proof of $e \vdash f$

Several proofs of the same statement $\iff$ Several programs of the same type

Example: $a \vdash a + a$
Computational interpretation

Proof of $e \vdash f$

Several proofs of the same statement $\iff$ Several programs of the same type

Example: $a \vdash a + a$

Curry-Howard isomorphism, typed programming, ... Well-understood for finite proofs, active field for infinite proofs.
This work: Computational content of proofs for regular expressions.

- Boolean type $2 = 1 + 1$
This work: Computational content of proofs for regular expressions.

- Boolean type \( 2 = 1 + 1 \)
- Add *structural* rules corresponding to simple natural programs
This work: Computational content of proofs for regular expressions.

- Boolean type $2 = 1 + 1$
- Add *structural* rules corresponding to simple natural programs
- Study the expressive power of regular proofs (finite graphs)
This work: Computational content of proofs for regular expressions.

- 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 $A^* \vdash 2 \leadsto$ 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:

$$
\begin{align*}
\vdash 2 & \quad \text{(tt)} \\
E, F \vdash 2 & \quad \text{(wkn)} \\
E, e, F \vdash 2 & \quad \text{(ctr)} \\
(E, F \vdash 2)_{a \in A} & \quad \text{(A)} \\
E, A, F \vdash 2 & \quad \text{(*)}
\end{align*}
$$
Proofs as language acceptors

What are the languages computed by cyclic proofs?

Example on alphabet \{a, b\}:

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

Language recognized: \(b^*\).
Proofs as language acceptors

What are the languages computed by cyclic proofs?

Example on alphabet \{a, b\}:

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

Language recognized: \(b^*\).

Lemma

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

A JMA is an automaton with \( k \) reading heads.

\[
\text{Transitions: } Q \times A^k \rightarrow Q \times \{\text{\textup{\\downarrow}}, \text{\textup{\circ}}, J_1, \ldots, J_k\}^k
\]

- \( \text{\textup{\\downarrow}} \): advance one step
- \( \text{\textup{\circ}} \): stay in place
- \( J_i \): jump to the position of head \( i \)
Expressive power of JMA

Theorem

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

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

Theorem

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

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

Comparison with Multihead Automata:

[Holzer, Kutrib, Malcher 2008]

- 1-way Multihead \( \subseteq \) JMA \( \subseteq \) 2-way Multihead
Expressive power of JMA

Theorem

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

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

Comparison with Multihead Automata:

[Holzer, Kutrib, Malcher 2008]

\[
\begin{array}{ccc}
1\text{-way Multihead} & \subseteq & \text{JMA} & \subseteq & 2\text{-way Multihead} \\
\text{Emptiness Undecidable} & = & \text{DLogSpace}
\end{array}
\]
What next?

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

Work in progress:

- No contraction = Primitive Recursive
- With contraction = System T
What next?

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

Work in progress:

No contraction $=$ Primitive Recursive
With contraction $=$ System T

Thank you for your attention!