Computational content of circular proof systems.

Denis Kuperberg   Laureline Pinault   Damien Pous

LIP, ENS Lyon

Séminaire de l'équipe Méthodes Formelles
Bordeaux

Tuesday 21st January 2020

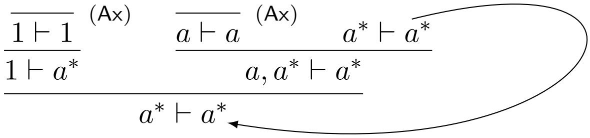
Cyclic proofs

Regular expressions

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

<u>Context:</u> Cyclic proofs for inclusion of expressions [Das, Pous '17]

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



Cyclic proofs

Regular expressions

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

<u>Context:</u> Cyclic proofs for inclusion of expressions [Das, Pous '17]

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

$$\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,(\mathsf{Ax})}{1 \vdash a^*} \qquad \dfrac{\dfrac{}{a \vdash a}\,(\mathsf{Ax}) \qquad a^* \vdash a^*}{a, a^* \vdash a^*}}{a^* \vdash a^*}$$

- Validity condition on infinite branches

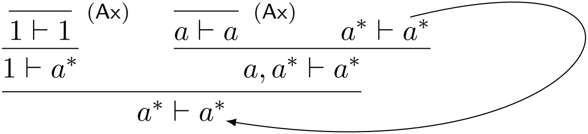
Cyclic proofs

Regular expressions

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

<u>Context:</u> Cyclic proofs for inclusion of expressions [Das, Pous '17]

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



- Validity condition on infinite branches
- $\exists$ proof of $e \vdash f \Leftrightarrow L(e) \subseteq L(f)$.

Computational interpretation

Proof of $e \vdash f$

↝ Computational interpretation

Program with input from e and output in f.

Computational interpretation

Proof of $e \vdash f$

⇝ Computational interpretation

Program with input from e and output in f.

Several proofs of the same statement

⇕

Several programs of the same type

Example:

$a \vdash a + a$

in_l or in_r

Computational interpretation

Proof of $e \vdash f$

↝ Computational interpretation

Program with input from e and output in f.

Several proofs of the same statement

$\Updownarrow$

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.

Computing languages

- 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:

$$\text{Proof } \pi \text{ of } A^* \vdash 2 \rightsquigarrow \text{Language } L(\pi) \subseteq A^*$$

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:

$$\frac{}{\vdash 2}\ (\mathbf{tt}) \qquad \frac{}{\vdash 2}\ (\mathbf{ff})$$

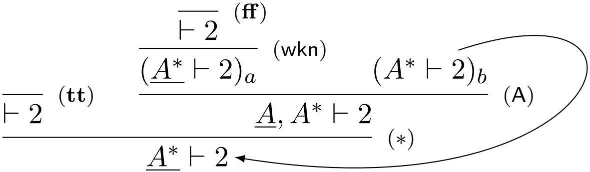
$$\frac{E, F \vdash 2}{E, \underline{e}, F \vdash 2}\ (\mathsf{wkn}) \qquad \frac{E, \underline{e, e}, F \vdash 2}{E, \underline{e}, F \vdash 2}\ (\mathsf{ctr})$$

$$\frac{(E, F \vdash 2)_{\underline{a \in A}}}{E, \underline{A}, F \vdash 2}\ (\mathsf{A}) \qquad \frac{E, F \vdash 2 \qquad E, \underline{A, A^*}, F \vdash 2}{E, \underline{A^*}, F \vdash 2}\ (*)$$

Proofs as language acceptors

What are the languages computed by cyclic proofs ?

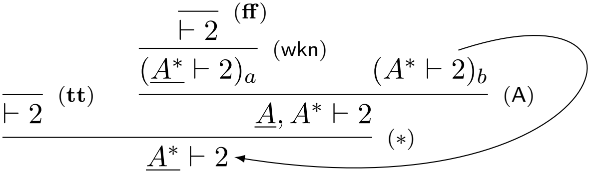
Example on alphabet $\{a, b\}$: b^*



Proofs as language acceptors

What are the languages computed by cyclic proofs ?

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

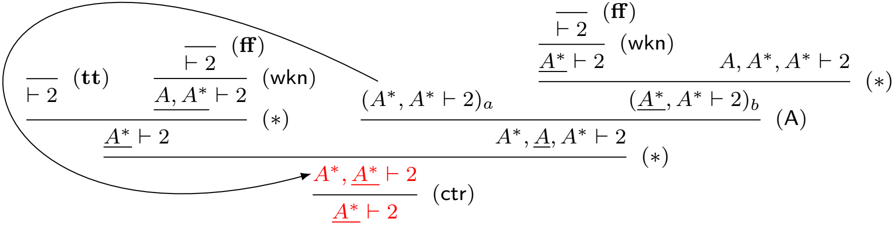


Lemma

Without contraction, the system captures exactly regular languages.

With contractions: what class of language?

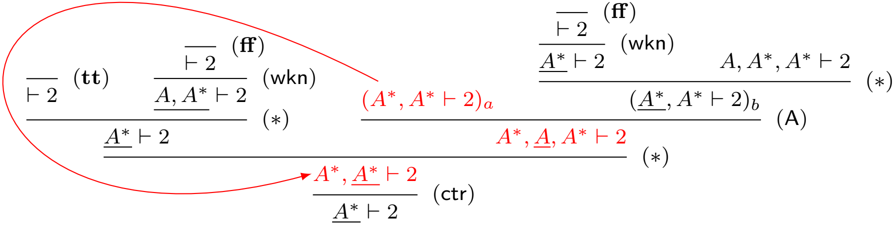
Example on alphabet $\{a, b\}$: $a^n b^n$



1st step : create a copy of the input and delete the first a's.

With contractions: what class of language?

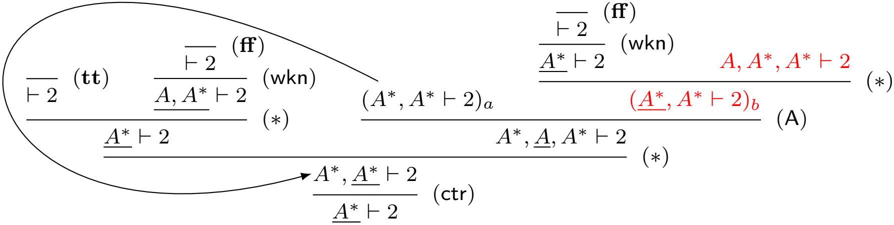
Example on alphabet $\{a, b\}$: $a^n b^n$



1st step : create a copy of the input and delete the first a's.

With contractions: what class of language?

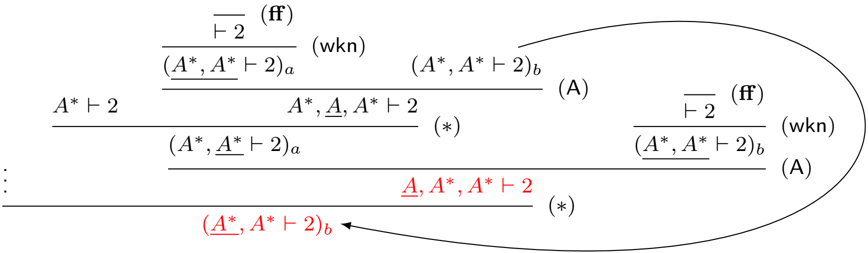
Example on alphabet $\{a, b\}$: $a^n b^n$



1st step : create a copy of the input and delete the first a's.

With contractions: what class of language?

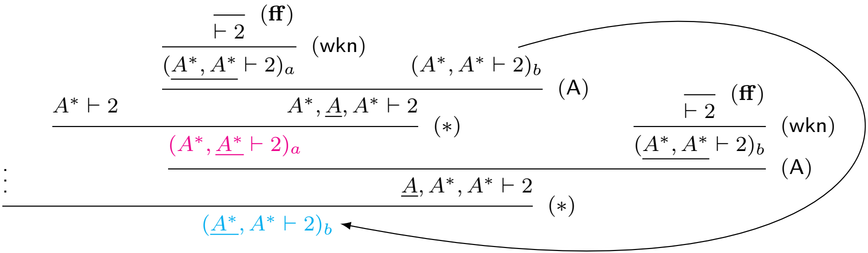
Example on alphabet $\{a, b\}$: $a^n b^n$



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?

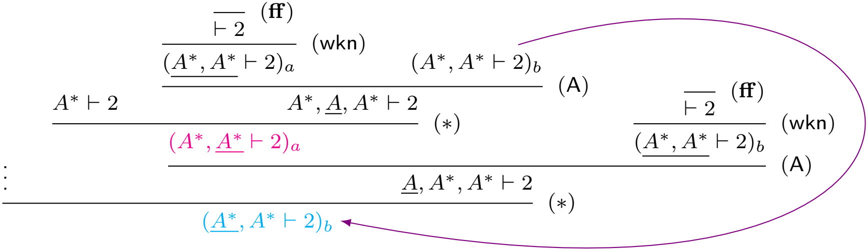
Example on alphabet $\{a, b\}$: $a^n b^n$



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?

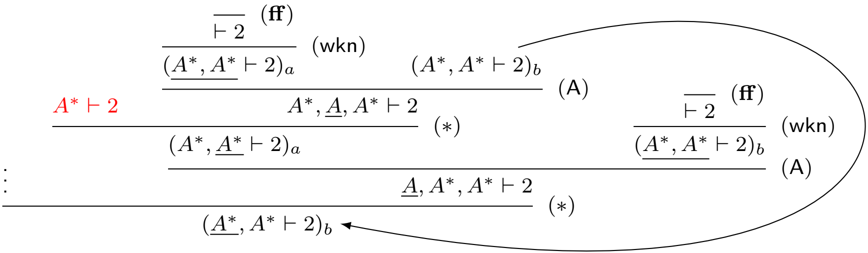
Example on alphabet $\{a, b\}$: $a^n b^n$



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 $\{a, b\}$: $a^n b^n$

$$\dfrac{\dfrac{}{\vdash 2}\,(\mathbf{ff}) \qquad \dfrac{\dfrac{\dfrac{}{\vdash 2}\,(\mathbf{ff})}{(\underline{A^*} \vdash 2)_a}\,(\mathsf{wkn}) \qquad \dfrac{\dfrac{}{\vdash 2}\,(\mathbf{tt})}{(\underline{A^*} \vdash 2)_b}\,(\mathsf{wkn})}{\underline{A}, A^* \vdash 2}\,(\mathsf{A})}{\underline{A^*} \vdash 2}\,(*)$$

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

With contractions: what class of language?

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

$$\dfrac{\dfrac{}{\vdash 2}\,(\mathbf{ff}) \qquad \dfrac{\dfrac{\dfrac{}{\vdash 2}\,(\mathbf{ff})}{(\underline{A^*} \vdash 2)_a}\,(\mathsf{wkn}) \qquad \dfrac{\dfrac{}{\vdash 2}\,(\mathbf{tt})}{(\underline{A^*} \vdash 2)_b}\,(\mathsf{wkn})}{\underline{A}, A^* \vdash 2}\,(\mathsf{A})}{\underline{A^*} \vdash 2}\,(*)$$

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

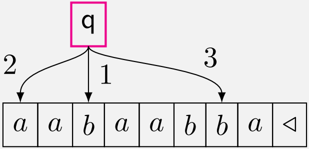
Example on alphabet $\{a, b, c\}$: $a^n b^n c^n$

With contractions: a new automaton model

Jumping Multihead Automata

A JMA is an automaton with k reading heads.

Transitions: $Q \times (A \cup \{\triangleleft\})^k \to Q \times \{▶|, ⁙, J_1, \ldots, J_k\}^k$



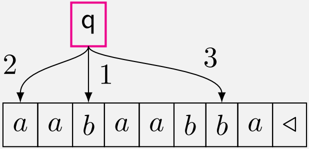
- ▶| : advance one step
- ⁙: stay in place
- J_i: jump to the position of head i

With contractions: a new automaton model

Jumping Multihead Automata

A JMA is an automaton with k reading heads.

Transitions: $Q \times (A \cup \{\triangleleft\})^k \to Q \times \{\blacktriangleright\!|, \text{◌}, J_1, \dots, J_k\}^k$

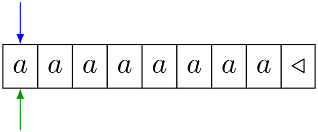


- $\blacktriangleright\!|$: advance one step
- ◌: stay in place
- J_i: jump to the position of head i

+ *Equivalent of the validity criterion*

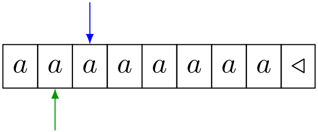
Example of JMA

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



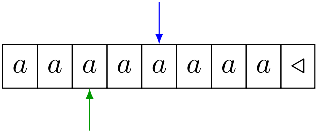
Example of JMA

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



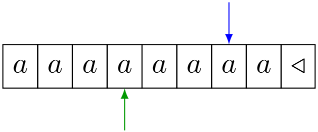
Example of JMA

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



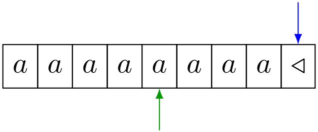
Example of JMA

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



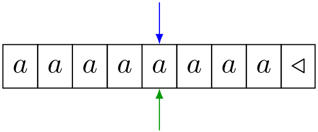
Example of JMA

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



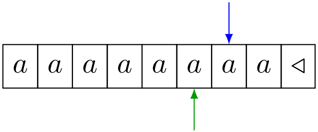
Example of JMA

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



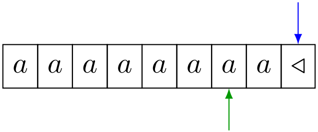
Example of JMA

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



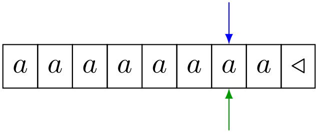
Example of JMA

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



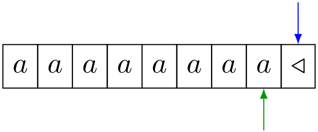
Example of JMA

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



Example of JMA

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



ACCEPT

Equivalence Theorem

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 $*$ and (A) rules

Expressive power of JMA

Comparison with Multihead Automata in Litterature:

[Holzer, Kutrib, Malcher 2008]

1-way Multihead

2-way Multihead

Expressive power of JMA

Comparison with Multihead Automata in Litterature:

[Holzer, Kutrib, Malcher 2008]

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

Expressive power of JMA

Comparison with Multihead Automata in Litterature:

[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 Litterature:

[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 Litterature:

[Holzer, Kutrib, Malcher 2008]

$\forall k,\ JMA(2) \not\subseteq 1DFA(k)$ $\qquad\qquad$ $\stackrel{?}{=}$

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

Emptiness Undecidable $\qquad\qquad$ LogSpace

Expressive power of JMA

Comparison with Multihead Automata in Litterature:

[Holzer, Kutrib, Malcher 2008]

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

$\stackrel{?}{=}$

e.g. Palindroms ?

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

Emptiness Undecidable

LogSpace

Simulating 2-ways automata

Theorem

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

Simulating 2-ways automata

Theorem

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

Difficulty: simulate a left move of some head.

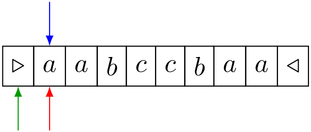
Simulating 2-ways automata

Theorem

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.



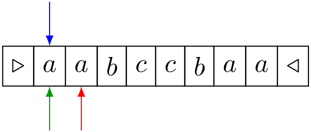
Simulating 2-ways automata

Theorem

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.



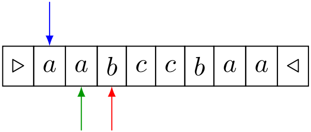
Simulating 2-ways automata

Theorem

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.



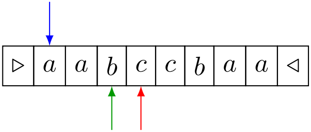
Simulating 2-ways automata

Theorem

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.



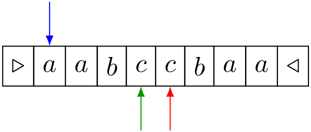
Simulating 2-ways automata

Theorem

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.



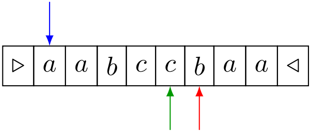
Simulating 2-ways automata

Theorem

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.



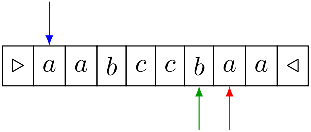
Simulating 2-ways automata

Theorem

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.



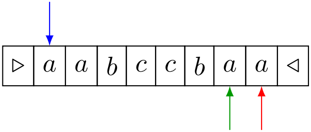
Simulating 2-ways automata

Theorem

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.



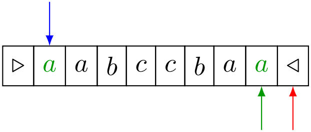
Simulating 2-ways automata

Theorem

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.



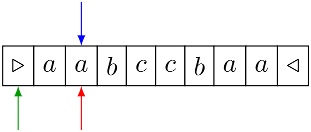
Simulating 2-ways automata

Theorem

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.



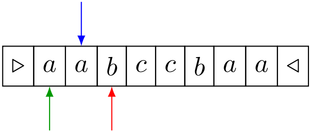
Simulating 2-ways automata

Theorem

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

Difficulty: simulate a left move of some head.

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



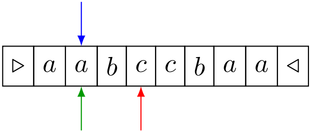
Simulating 2-ways automata

Theorem

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.



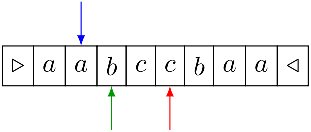
Simulating 2-ways automata

Theorem

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

Difficulty: simulate a left move of some head.

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



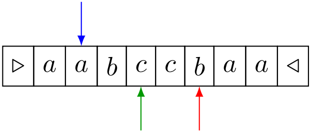
Simulating 2-ways automata

Theorem

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.



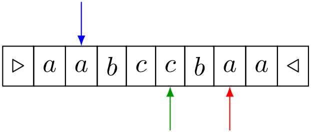
Simulating 2-ways automata

Theorem

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.



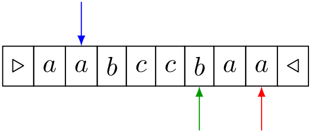
Simulating 2-ways automata

Theorem

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.



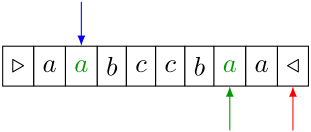
Simulating 2-ways automata

Theorem

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.



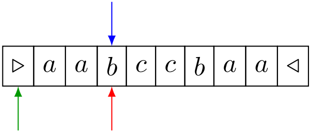
Simulating 2-ways automata

Theorem

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.



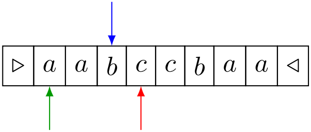
Simulating 2-ways automata

Theorem

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.



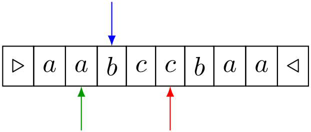
Simulating 2-ways automata

Theorem

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.



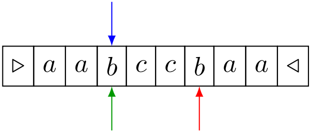
Simulating 2-ways automata

Theorem

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.



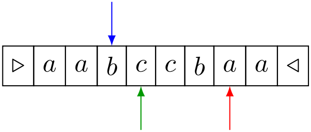
Simulating 2-ways automata

Theorem

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.



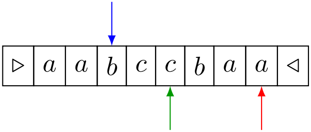
Simulating 2-ways automata

Theorem

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.



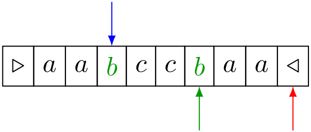
Simulating 2-ways automata

Theorem

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

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.

$\triangleright$	a	a	b	c	c	b	a	a	$\triangleleft$

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

Simulating 2-ways automata

Theorem

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.

$\triangleright$	a	a	b	c	c	b	a	a	$\triangleleft$

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

Corollary

Cyclic proofs with contraction characterize $\textsc{LogSpace}$.

What next ?

- Add the cut rule:

$$\frac{E \vdash e \qquad e, F \vdash g}{E, F \vdash g}$$

- Corresponds to composition of functions
- Enriched expressions:

$$e, f := 1 \mid a \mid e \cdot f \mid e + f \mid e^* \mid e \to f \mid e \cap f$$

- Sequents $(1^*)^k \vdash 1^*$: functions $\mathbb{N}^k \to \mathbb{N}$

What next ?

- Add the cut rule:

$$\frac{E \vdash e \qquad e, F \vdash g}{E, F \vdash g}$$

- Corresponds to composition of functions
- Enriched expressions:

$$e, f := 1 \mid a \mid e \cdot f \mid e + f \mid e^* \mid e \to f \mid e \cap f$$

- Sequents $(1^*)^k \vdash 1^*$: functions $\mathbb{N}^k \to \mathbb{N}$

How does it increase the expressive power ?

An extended, resource-tracking System T

λ-calculus extended with pairs, singletons, sums, lists, and additive pairs ($i \in \{0,1\}$):

$$\begin{array}{llll} M, N, O ::= x & \mid \lambda x.M & \mid M N \\ & \mid \langle M, N \rangle & \mid \text{let } \langle x, y \rangle := M \text{ in } N \\ & \mid \langle\rangle & \mid \text{let } \langle\rangle := M \text{ in } N \\ & \mid \mathbf{i}_i M & \mid \mathbf{D}(M; x.N; x.O) \\ & \mid [] \mid M :: N & \mid \mathbf{R}(M; N; x.y.O) \\ & \mid \langle\!\langle M, N \rangle\!\rangle & \mid \mathbf{p}_i M \end{array}$$

Comes with a *type system*.

Example:

$$\frac{\Gamma \vdash L : e^* \quad \Delta \vdash M : g \quad x : e, y : g \vdash N : g}{\Gamma, \Delta \vdash \mathbf{R}(L; M; x.y.N) : g} \; {*\text{-}e}$$

Affine version T_{aff}

System T_{aff}: Cannot use contraction in typing derivations:

$$c\,\frac{x : e, x : e, \Gamma \vdash M : f}{x : e, \Gamma \vdash M : f}$$

Affine version $\mathsf{T}_{\mathsf{aff}}$

System $\mathsf{T}_{\mathsf{aff}}$: Cannot use contraction in typing derivations:

$$c \frac{x : e, x : e, \Gamma \vdash M : f}{x : e, \Gamma \vdash M : f}$$

Example:

- $\lambda x.\langle x, x \rangle$ is not typable in $\mathsf{T}_{\mathsf{aff}}$
- $\lambda x.\langle\!\langle x, x \rangle\!\rangle$ is typable in $\mathsf{T}_{\mathsf{aff}}$

Results

Theorem

Cyclic proofs without contraction	$\Longleftrightarrow$	*System* T_{aff}		
Cyclic proofs with contraction	$\overset{functions}{\Longleftrightarrow}$	*System* T	$\overset{functions}{\Longleftrightarrow}$	*Peano*

Results

Theorem

Cyclic proofs without contraction	$\iff$	*System T_{aff}*		
Cyclic proofs with contraction	$\overset{functions}{\iff}$	*System T*	$\overset{functions}{\iff}$	*Peano*

System T [affine] $\rightarrow$ proofs: easy.

Results

Theorem

Cyclic proofs without contraction	$\Longleftrightarrow$	*System T_{aff}*		
Cyclic proofs with contraction	$\overset{functions}{\Longleftrightarrow}$	*System T*	$\overset{functions}{\Longleftrightarrow}$	*Peano*

System T [affine] $\rightarrow$ proofs: easy.

Affine proofs $\rightarrow$ T_{aff}:

- normal form for proofs, with explicit hierarchy of cycles
- inductively build T_{aff} terms, crucial use of additive pairs

Results

Theorem

Cyclic proofs without contraction $\iff$ *System* T_{aff}

Cyclic proofs with contraction $\overset{functions}{\iff}$ *System* T $\overset{functions}{\iff}$ *Peano*

System T [affine] $\to$ proofs: easy.

Affine proofs $\to$ T_{aff}:

- normal form for proofs, with explicit hierarchy of cycles
- inductively build T_{aff} terms, crucial use of additive pairs

Proofs $\to$ system T:
Show termination in ACA0 + conservativity results.

Ongoing work

Conjecture

T_{aff} computes exactly primitive recursive functions

Ongoing work

Conjecture

T_{aff} computes exactly primitive recursive functions

Related result:

Theorem (Dal Lago 2009)

T_{aff} without additive pairs $\Leftrightarrow$ primitive recursive functions

Does it still hold with additive pairs $\langle\langle M, N\rangle\rangle$?

Ongoing work

Conjecture

T_{aff} computes exactly primitive recursive functions

Related result:

Theorem (Dal Lago 2009)

T_{aff} without additive pairs $\Leftrightarrow$ primitive recursive functions

Does it still hold with additive pairs $\langle\langle M, N \rangle\rangle$?

Other direction: More constructive proof for general System T ?

Ongoing work

Conjecture

T_{aff} computes exactly primitive recursive functions

Related result:

Theorem (Dal Lago 2009)

T_{aff} without additive pairs $\Leftrightarrow$ primitive recursive functions

Does it still hold with additive pairs $\langle\langle M, N \rangle\rangle$?

Other direction: More constructive proof for general System T ?

Thanks for your attention !