

# One-way definability of two-way transducers

F. Baschenis, O. Gauwin, A. Muscholl and G. Puppis

DeLTA meeting. LaBRI

March 10, 2017

Mapping from words to words:

- delta  $\mapsto$  dlt

erase vowels

- delta  $\mapsto$   $\overleftarrow{\text{delta}} = \text{atled}$

reverse

Mapping from words to words:

- $\text{delta} \mapsto \text{dlt}$                       erase vowels

- $\text{delta} \mapsto \overleftarrow{\text{delta}} = \text{atled}$                       reverse

Or relations:

- $\text{delta} \longrightarrow \{(\text{delta})^i \mid i \leq k\}$                       k-bounded duplication

# How to model transductions

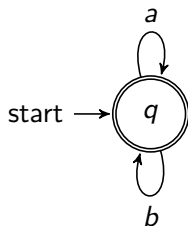
- Logic : *MSOT*, *NMSOT* (Courcelle)

# How to model transductions

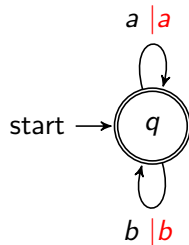
- Logic : *MSOT*, *NMSOT* (Courcelle)
- Write-only registers : *DSST*, *NSST* (Alur)

# How to model transductions

- Logic : *MSOT*, *NMSOT* (Courcelle)
- Write-only registers : *DSST*, *NSST* (Alur)
- Finite machines : *1NFT*, *2DFT*, etc...
- etc...

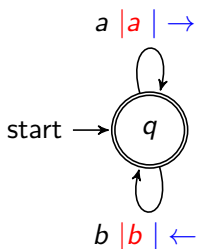


- automata :  $L = \Sigma^*$

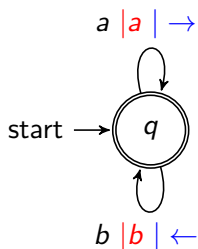


- automata :  $L = \Sigma^*$
- **transducer** (1DFT) :  $f : u \mapsto u \quad \forall u \in \Sigma^*$



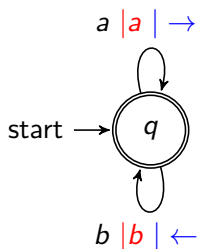


- automata :  $L = \Sigma^*$
- **transducer** (1DFT) :  $f : u \mapsto u \quad \forall u \in \Sigma^*$
- **two-way transducer** (2DFT) :  $f : u \mapsto u \quad \forall u \in \{a\}^*$



- automata :  $L = \Sigma^*$
- **transducer** (1DFT) :  $f : u \mapsto u \quad \forall u \in \Sigma^*$
- **two-way transducer** (2DFT) :  $f : u \mapsto u \quad \forall u \in \{a\}^*$

We can add non determinism : 1NFT, 2NFT.



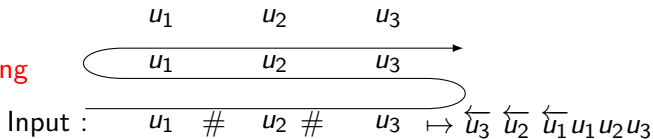
- automata :  $L = \Sigma^*$
- **transducer** (1DFT) :  $f : u \mapsto u \quad \forall u \in \Sigma^*$
- **two-way transducer** (2DFT) :  $f : u \mapsto u \quad \forall u \in \{a\}^*$

We can add non determinism : 1NFT, 2NFT.

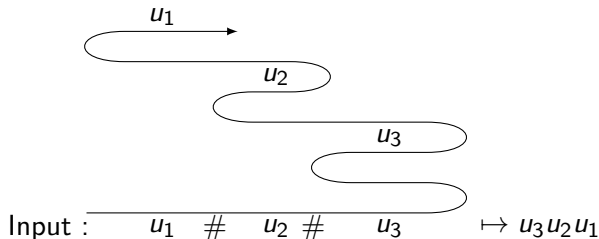
**functional**: the transduction is a function.

# Sweeping transducers

Sweeping



Non-sweeping



For automata:

- Determinism = Non-determinism (subset construction)

For automata:

- Determinism = Non-determinism (subset construction)
- 2-way = 1-way (crossing sequence construction)

# For transducers

	Det	Functional	Non-Det
1W			
Sweeping			
2W			

# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$		
Sweeping			
2W			



# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$		
Sweeping	<i>SweepingDFT</i> $u \mapsto u u$		
2W			

# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$		
Sweeping	<i>SweepingDFT</i> $u \mapsto u u$		
2W	<i>2DFT</i> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$		

# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$	<i>1fNFT</i> $ua \mapsto a^{ u }$	
Sweeping	<i>SweepingDFT</i> $u \mapsto u u$		
2W	<i>2DFT</i> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$		

# For transducers

	Det	Functional	Non-Det
1W	<p><i>1DFT</i></p> $au \mapsto a^{ u }$	<p><i>1fNFT</i></p> $ua \mapsto a^{ u }$	
Sweeping	<p><i>SweepingDFT</i></p> $u \mapsto u u$	<p><i>Sweeping fNFT</i></p> $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	
2W	<p><i>2DFT</i></p> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$		

# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$	<i>1fNFT</i> $ua \mapsto a^{ u }$	
Sweeping	<i>SweepingDFT</i> $u \mapsto u u$	<i>Sweeping fNFT</i> $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	
2W	<i>2DFT</i> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	<i>2fNFT</i> $= 2DFT$	

# For transducers

	Det	Functional	Non-Det
1W	<p><i>1DFT</i></p> $au \mapsto a^{ u }$	<p><i>1fNFT</i></p> $ua \mapsto a^{ u }$	<p><i>1NFT</i></p> $u \mapsto \Sigma^{ u }$
Sweeping	<p><i>SweepingDFT</i></p> $u \mapsto u u$	<p><i>Sweeping fNFT</i></p> $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	
2W	<p><i>2DFT</i></p> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	<p><i>2fNFT</i></p> <p><i>= 2DFT</i></p>	

# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$	<i>1fNFT</i> $ua \mapsto a^{ u }$	<i>1NFT</i> $u \mapsto \Sigma^{ u }$
Sweeping	<i>SweepingDFT</i> $u \mapsto u u$	<i>Sweeping fNFT</i> $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	<i>SweepingNFT</i> $u \mapsto u^*$
2W	<i>2DFT</i> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	<i>2fNFT</i> $= 2DFT$	

# For transducers

	Det	Functional	Non-Det
1W	<i>1DFT</i> $au \mapsto a^{ u }$	<i>1fNFT</i> $ua \mapsto a^{ u }$	<i>1NFT</i> $u \mapsto \Sigma^{ u }$
Sweeping	<i>SweepingDFT</i> $u \mapsto u u$	<i>Sweeping fNFT</i> $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	<i>SweepingNFT</i> $u \mapsto u^*$
2W	<i>2DFT</i> $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	<i>2fNFT</i> $= 2DFT$	<i>2NFT</i> $u_1 \cdots u_n \mapsto$ $(u_n \cdots u_1)^*$



# For transducers

	Det	Functional	Non-Det
1W	$1DFT$ $au \mapsto a^{ u }$	$1fNFT$ $ua \mapsto a^{ u }$	$1NFT$ $u \mapsto \Sigma^{ u }$
Sweeping	$SweepingDFT$ $u \mapsto u u$	$Sweeping fNFT$ $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	$SweepingNFT$ $u \mapsto u^*$
2W	$2DFT$ $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	$2fNFT$ $= 2DFT$	$2NFT$ $u_1 \cdots u_n \mapsto$ $(u_n \cdots u_1)^*$

Polynomial

# For transducers

	Det	Functional	Non-Det
1W	$1DFT$ $au \mapsto a^{ u }$	$1fNFT$ $ua \mapsto a^{ u }$	$1NFT$ $u \mapsto \Sigma^{ u }$
Sweeping	$SweepingDFT$ $u \mapsto u u$	$Sweeping fNFT$ $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	$SweepingNFT$ $u \mapsto u^*$
2W	$2DFT$ $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	$2fNFT$ $= 2DFT$	$2NFT$ $u_1 \cdots u_n \mapsto$ $(u_n \cdots u_1)^*$

Polynomial

# For transducers

	Det	Functional	Non-Det
1W	$1DFT$ $au \mapsto a^{ u }$	$1fNFT$ $ua \mapsto a^{ u }$	$1NFT$ $u \mapsto \sum^{ u }$
Sweeping	$SweepingDFT$ $u \mapsto u u$	$Sweeping fNFT$ $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $\mapsto$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	$SweepingNFT$ $u \mapsto u^*$
2W	$2DFT$ $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	$2fNFT$ $= 2DFT$	$2NFT$ $u_1 \cdots u_n \mapsto$ $(u_n \cdots u_1)^*$

Polynomial

Undecidable

# For transducers

	Det	Functional	Non-Det
1W	$1DFT$ $au \mapsto a^{ u }$	$1fNFT$ $ua \mapsto a^{ u }$	$1NFT$ $u \mapsto \sum^{ u }$
Sweeping	$SweepingDFT$ $u \mapsto u u$	$Sweeping fNFT$ $a_1 u_1 a'_1 \cdot a_n u_n a'_n$ $a'_1 u_1 a_1 \cdot a'_n u_n a_n$	$SweepingNFT$ $u \mapsto u^*$
2W	$2DFT$ $u_1 \cdots u_n \mapsto$ $u_n \cdots u_1$	$2fNFT$ $= 2DFT$	$2NFT$ $u_1 \cdots u_n \mapsto$ $(u_n \cdots u_1)^*$

Polynomial

Undecidable

2-Exponential

## Problem (One-way definability)

*Given a 2NFT, is it equivalent to some NFT ?*

## Problem (One-way definability)

*Given a 2NFT, is it equivalent to some NFT ?*

- Undecidable in the general case <sup>1</sup>

---

<sup>1</sup> B., Gauwin, Muscholl, Puppis, FSTTCS 2015

## Problem (One-way definability)

*Given a 2NFT, is it equivalent to some NFT ?*

- Undecidable in the general case <sup>1</sup>
- Decidable for **functional** transducers in **non-elementary** complexity <sup>2</sup>

---

<sup>1</sup> B., Gauwin, Muscholl, Puppis, FSTTCS 2015

<sup>2</sup>Filiot, Gauwin, Reynier, Servais, LICS 2013

## Problem (One-way definability)

*Given a 2NFT, is it equivalent to some NFT ?*

- Undecidable in the general case <sup>1</sup>
- Decidable for **functional** transducers in **non-elementary** complexity <sup>2</sup>
- Decidable for **functional sweeping** transducers in **elementary** complexity <sup>1</sup>

---

<sup>1</sup> B., Gauwin, Muscholl, Puppis, FSTTCS 2015

<sup>2</sup>Filiot, Gauwin, Reynier, Servais, LICS 2013



## Problem (One-way definability)

*Given a 2NFT, is it equivalent to some NFT ?*

- Undecidable in the general case <sup>1</sup>
- Decidable for **functional** transducers in **non-elementary** complexity <sup>2</sup>
- Decidable for **functional sweeping** transducers in **elementary** complexity <sup>1</sup>

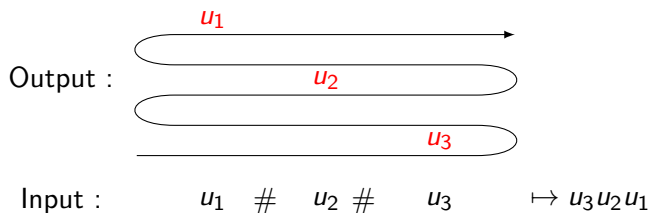
Goal : **elementary** complexity for **functional** transducers ?

---

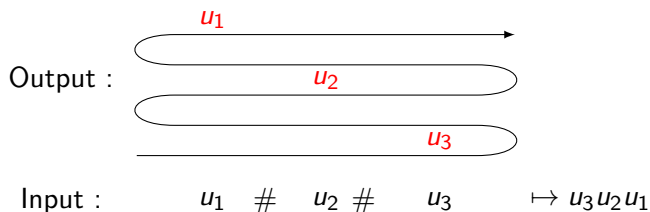
<sup>1</sup> B., Gauwin, Muscholl, Puppis, FSTTCS 2015

<sup>2</sup>Filiot, Gauwin, Reynier, Servais, LICS 2013

# Example (1)



# Example (1)



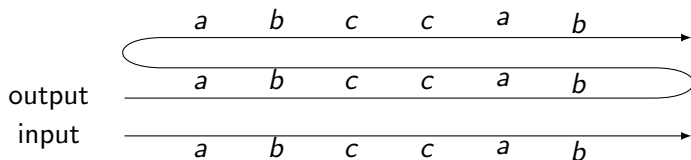
If  $u_1, u_3$  bounded:

guess  $u_3$  store  $u_1$   $u_3$   $u_2$   $u_1$  check  $u_3$

## Example (2)

$R$  regular language .  $F : u \in R \mapsto uu$

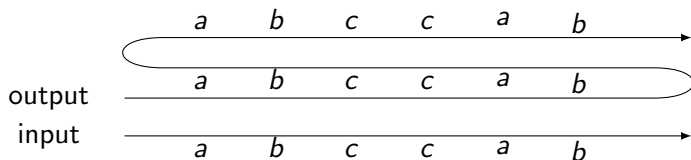
- $F$  is definable by a 2NFT :



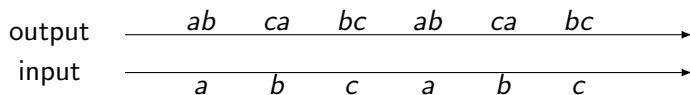
## Example (2)

$R$  regular language .  $F : u \in R \mapsto uu$

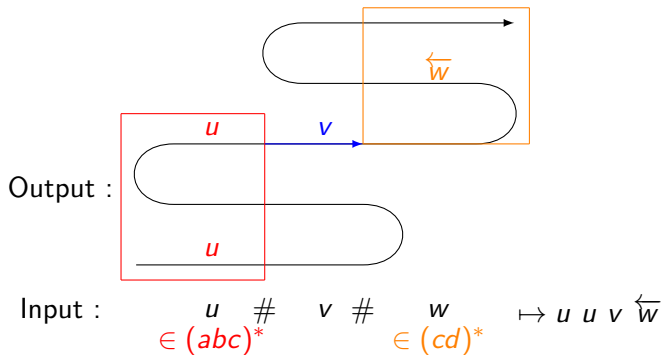
- $F$  is definable by a 2NFT :



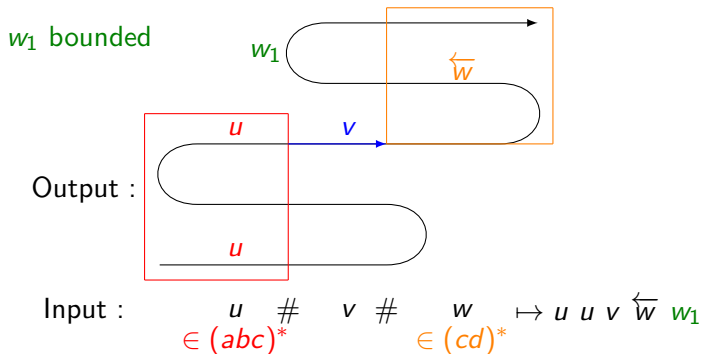
- If  $R = (abc)^*$  : one-way definable :



# Run decomposition



# Run decomposition

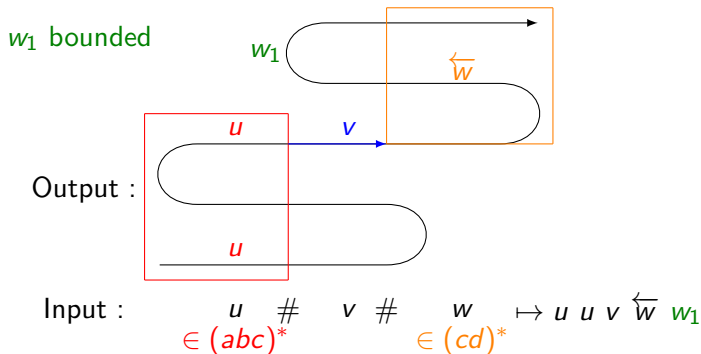


## Theorem

*T is one-way definable iff all runs have a decomposition.*

$\Leftarrow$ : Construction of the equivalent transducer.

# Run decomposition



## Theorem

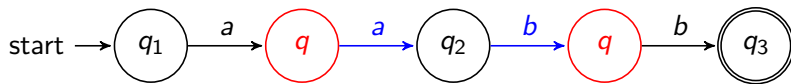
$T$  is one-way definable iff all runs have a decomposition.

$\Leftarrow$ : Construction of the equivalent transducer. We focus now on  $\Rightarrow$

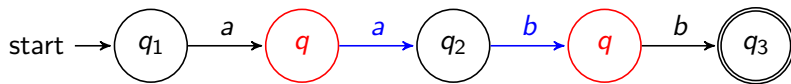


- 1 Introduction on transductions
- 2 Decomposing a run in feasible parts
- 3 Obtaining periodic outputs
  - Inversions
  - Idempotent loops
- 4 Minimisation and Conclusion

# Pumping: automaton

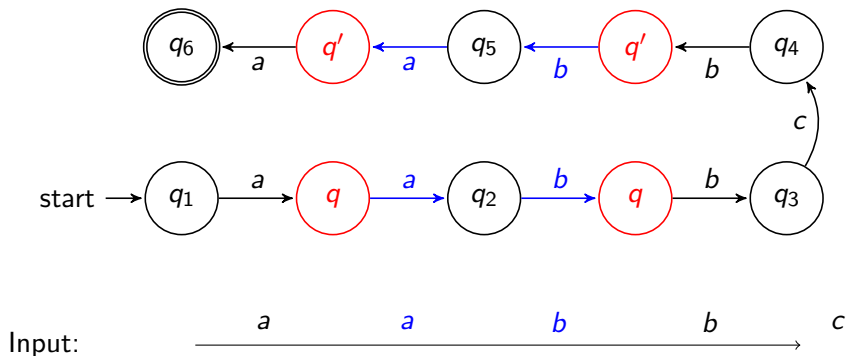


# Pumping: automaton

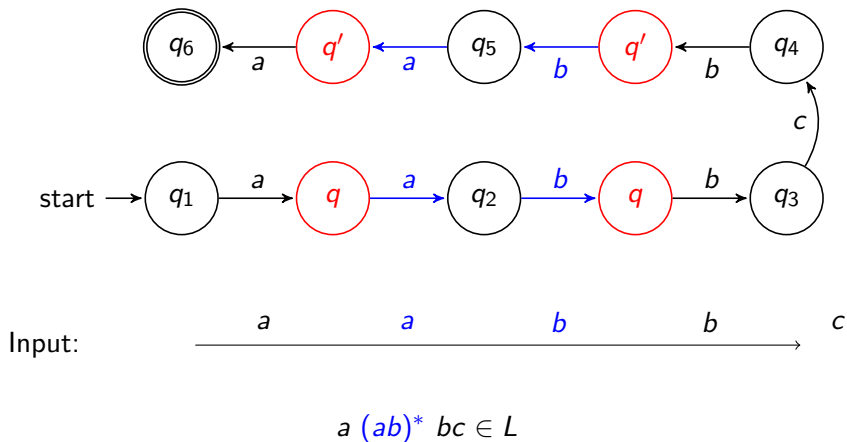


$$a (ab)^* b \in L$$

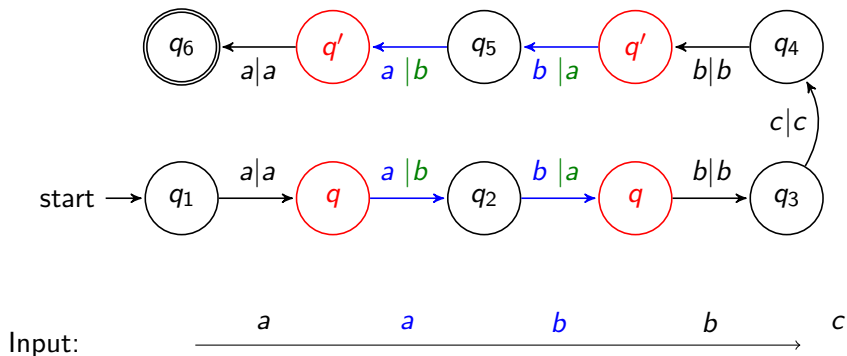
# Pumping: two-way automaton



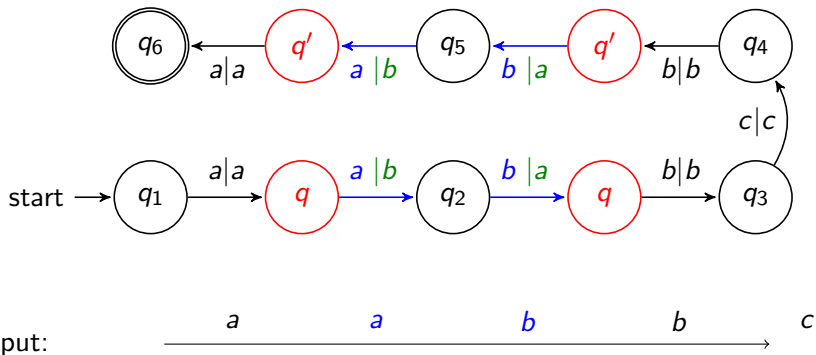
# Pumping: two-way automaton



# Pumping: two-way transducer

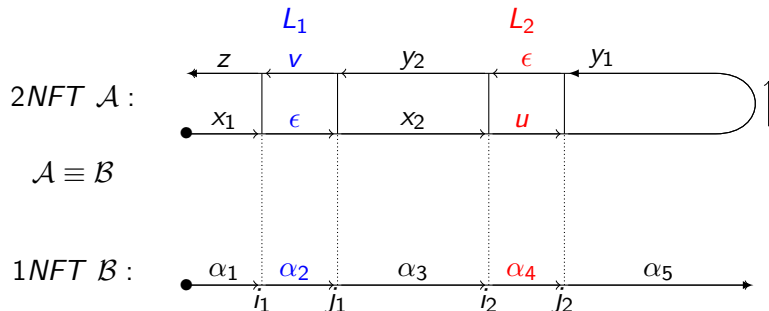


# Pumping: two-way transducer



$$a (ab)^n bc \mapsto a (ba)^n bcb (ab)^n a$$

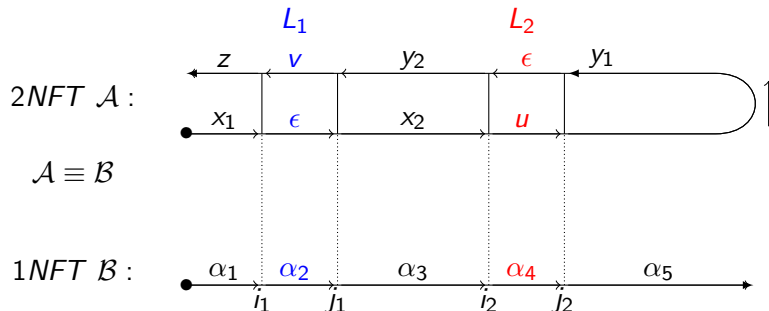
# Obtaining periodic parts of the output



<sup>3</sup>Fine and Wilf theorem



# Obtaining periodic parts of the output

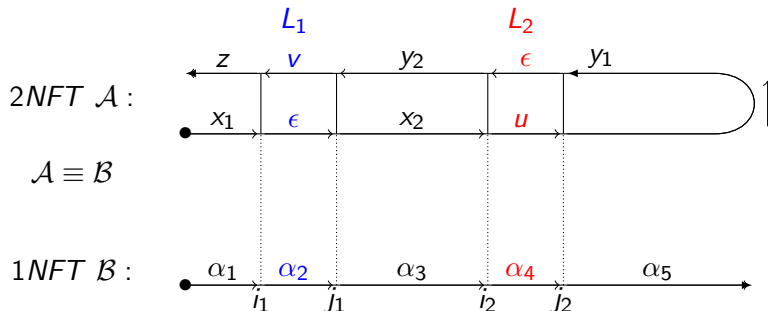


We can pump  $u[i_1, j_1]$  and  $u[i_2, j_2]$  in the input, and obtain the equation

$$x_1 x_2 u^K y_1 y_2 v^P z = \alpha_1 \alpha_2^P \alpha_3 \alpha_4^K \alpha_5$$

<sup>3</sup>Fine and Wilf theorem

# Obtaining periodic parts of the output



We can pump  $u[i_1, j_1]$  and  $u[i_2, j_2]$  in the input, and obtain the equation

$$x_1 x_2 u^K y_1 y_2 v^P z = \alpha_1 \alpha_2^P \alpha_3 \alpha_4^K \alpha_5$$

Combinatorics<sup>3</sup>  $\Rightarrow$  the output from  $u$  to  $v$  has bounded period

<sup>3</sup>Fine and Wilf theorem

What happens when the loops are not sweeping ?

What happens when the loops are not sweeping ?

⇒ pumping by groups of factors

- $\# \text{Right reversals} = \# \text{Left reversals}$

## Second difficulty

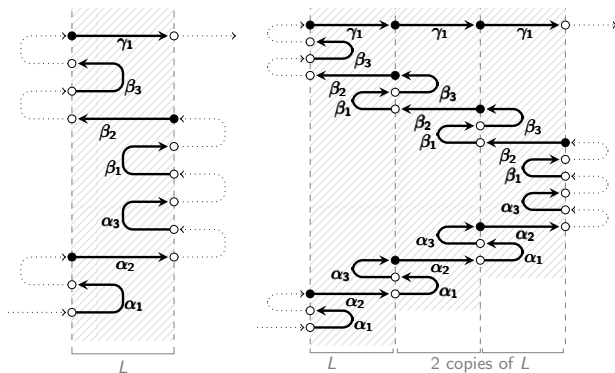
How to get periodic parts for the output of the **original** run ?

## Second difficulty

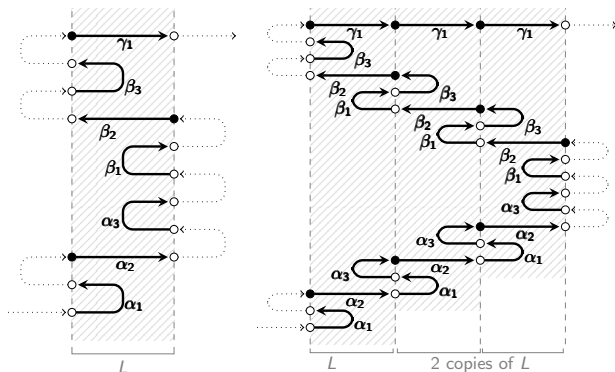
How to get periodic parts for the output of the **original** run ?

⇒ Focus on loops with "good" components.

# Good behavior of loops



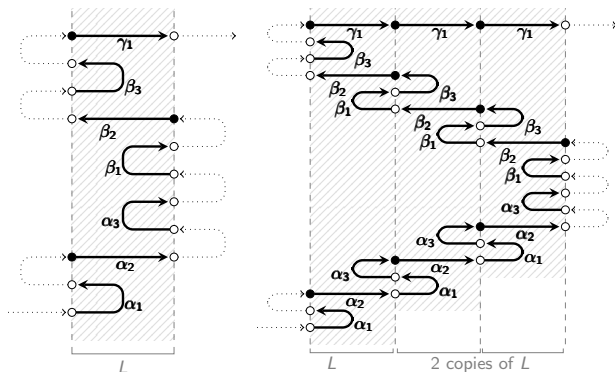
# Good behavior of loops



Good loop = One flat factor by component



# Good behavior of loops



Good loop = One flat factor by component = Idempotent loops

# Pumping idempotent loops

- Components: group of factors pumped together.

# Pumping idempotent loops

- Components: group of factors pumped together.
- Location of a component: location of the flat factor

# Pumping idempotent loops

- Components: group of factors pumped together.
- Location of a component: location of the flat factor
- Trace: factors of a component in the pumped order

# Pumping idempotent loops

- Components: group of factors pumped together.
- Location of a component: location of the flat factor
- Trace: factors of a component in the pumped order

## Theorem

$L$  an idempotent loop.  $C_1, \dots, C_k$  the components.  $\rho = \rho_0 \cdot \dots \cdot \rho_k$ .  
 $\text{pump}_L^{m+1}(\rho) = \rho_0 \text{tr}(C_1)^m \rho_1 \dots \rho_{k-1} \text{tr}(C_k)^m \rho_k$

# Pumping idempotent loops

- Components: group of factors pumped together.
- Location of a component: location of the flat factor
- Trace: factors of a component in the pumped order

## Theorem

$L$  an idempotent loop.  $C_1, \dots, C_k$  the components.  $\rho = \rho_0 \cdot \dots \cdot \rho_k$ .  
 $\text{pump}_L^{m+1}(\rho) = \rho_0 \text{tr}(C_1)^m \rho_1 \dots \rho_{k-1} \text{tr}(C_k)^m \rho_k$

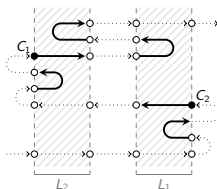
Same equation as the sweeping case !

# How to get periodic blocks

## Definition (Inversion)

$(L_1, C_1), (L_2, C_2)$  with  $\text{tr}(C_i) \neq \epsilon$ .

- $L_1 < L_2$  for the input order
- $C_2 < C_1$  for the run order

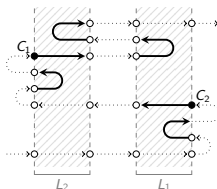


# How to get periodic blocks

## Definition (Inversion)

$(L_1, C_1), (L_2, C_2)$  with  $\text{tr}(C_i) \neq \epsilon$ .

- $L_1 < L_2$  for the input order
- $C_2 < C_1$  for the run order



## Proposition

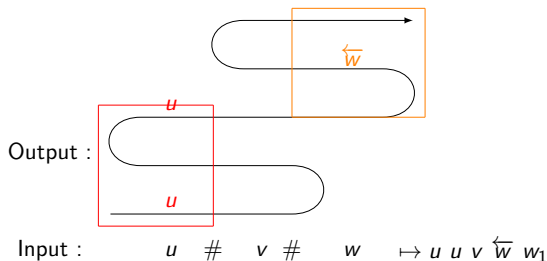
$\text{out}[C_2, C_1]$  has a bounded period <sup>a</sup>

<sup>a</sup>Fine and Wilf and previous theorem





# Obtaining idempotent loops

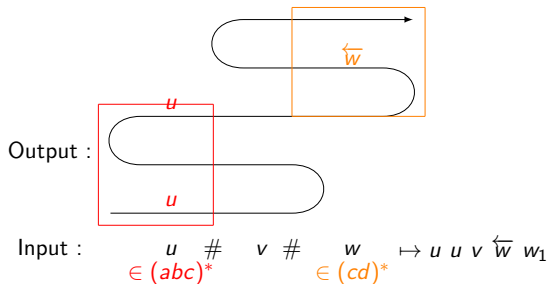


Ramsey theory <sup>4</sup> :

output unbounded  $\Rightarrow$  idempotent loop with non-empty component.

<sup>4</sup>Simon's Factorization Forest Theorem

# Obtaining idempotent loops



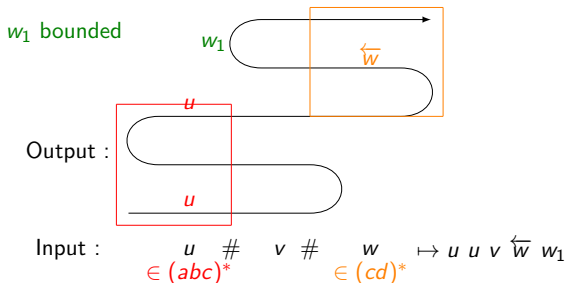
Ramsey theory <sup>4</sup> :

output unbounded  $\Rightarrow$  idempotent loop with non-empty component.

- Bounds the output of the period of a block.

<sup>4</sup>Simon's Factorization Forest Theorem

# Obtaining idempotent loops



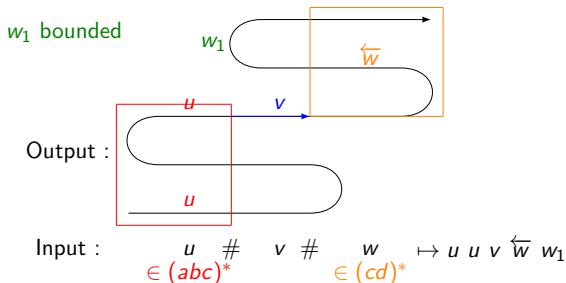
Ramsey theory<sup>4</sup> :

output unbounded  $\Rightarrow$  idempotent loop with non-empty component.

- Bounds the output of the period of a block.
- Bounds the output outside a block.

<sup>4</sup>Simon's Factorization Forest Theorem

# Obtaining idempotent loops



Ramsey theory<sup>4</sup> :

output unbounded  $\Rightarrow$  idempotent loop with non-empty component.

- Bounds the output of the period of a block.
- Bounds the output outside a block.
- Unbounded  $\Rightarrow$  produced from left to right.

<sup>4</sup>Simon's Factorization Forest Theorem

## Theorem

*The following are equivalent:*

- *$T$  one-way definable*
- *All inversions have a bounded period*
- *$\exists$  a decomposition for all run*

## Theorem

*The following are equivalent:*

- *$T$  one-way definable*
- *All inversions have a bounded period*
- *$\exists$  a decomposition for all run*

Decision procedure: check that all inversions have a bounded period :  
 $2\text{EXPSPACE}$

## Theorem

*We build  $T'$  such that:*

- $T' \subseteq T$ ,
- $\text{dom}(T) = \text{dom}(T')$  iff  $\mathcal{T}$  is one-way definable.
- $T'$  is in  $3\text{EXP}$  size.



## Theorem

We build  $T'$  such that:

- $T' \subseteq T$ ,
- $\text{dom}(T) = \text{dom}(T')$  iff  $\mathcal{T}$  is one-way definable.
- $T'$  is in  $3EXP$  size.

$T'$  guesses a word of  $2EXP$  length:

- Crossing sequence construction
- Idempotent loops

$\Rightarrow 3EXP$  size

- 1 Introduction on transductions
- 2 Decomposing a run in feasible parts
- 3 Obtaining periodic outputs
  - Inversions
  - Idempotent loops
- 4 Minimisation and Conclusion

## Problem (k-pass definability)

*Given a 2NFT, is it equivalent to some k-pass sweeping transducer ?*

---

<sup>5</sup>B., Gauwin, Muscholl, Puppis ICALP 2016

## Problem (k-pass definability)

*Given a 2NFT, is it equivalent to some k-pass sweeping transducer ?*

EXPSPACE for sweeping transducers <sup>5</sup>

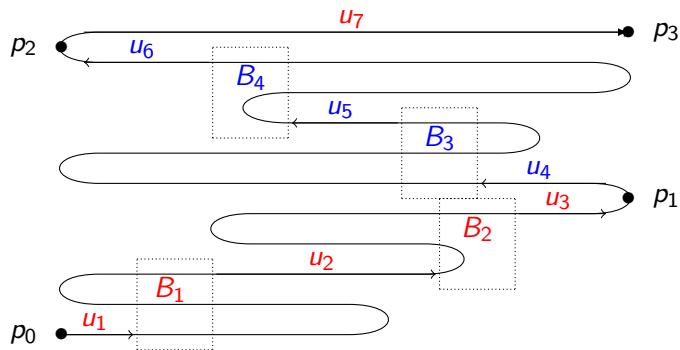
One-way definability as a black box

---

<sup>5</sup>B., Gauwin, Muscholl, Puppis ICALP 2016

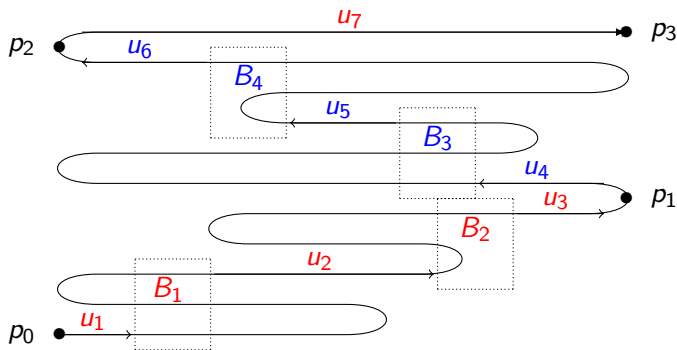
# k-decomposition

Same notion of  $k$ -decomposition:



# k-decomposition

Same notion of  $k$ -decomposition:



$2EXP$  for any 2-way transducer.

## Proposition

*If  $T$  is  $k$ -pass sweeping  $\rightarrow k < 2EXP$*

## Proposition

If  $T$  is  $k$ -pass sweeping  $\rightarrow k < 2EXP$

$\Rightarrow$  Sweeping-definability in  $2EXPSPACE$ .

We can build a  $3EXP$  equivalent transducer



- 1 In  $2EXPSPACE$  for functional  $2NFT$ :
  - One-way definability
  - k-pass definability
  - Sweeping definability
  - Minimization of the number of reversals/passes

- 1 In  $2EXPSPACE$  for functional  $2NFT$ :
  - One-way definability
  - k-pass definability
  - Sweeping definability
  - Minimization of the number of reversals/passes
- 2 Equivalent transducer of 3 exponential size

- 1 In  $2EXPSPACE$  for functional  $2NFT$ :
  - One-way definability
  - k-pass definability
  - Sweeping definability
  - Minimization of the number of reversals/passes
- 2 Equivalent transducer of 3 exponential size
- 3 Only 2 exponential lower bound for the size.

- 1 In  $2EXPSPACE$  for functional  $2NFT$ :
  - One-way definability
  - $k$ -pass definability
  - Sweeping definability
  - Minimization of the number of reversals/passes
- 2 Equivalent transducer of 3 exponential size
- 3 Only 2 exponential lower bound for the size.
- 4 What about  $k$ -valued transducers ?

