

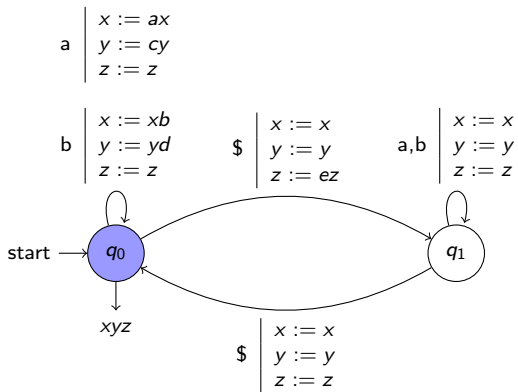
# Towards the decomposition of $k$ -valued NSST

Paul Gallot, Anca Muscholl, Gabriele Puppis, Sylvain Salvati

Université de Lille, INRIA Links

# Streaming String Transducers

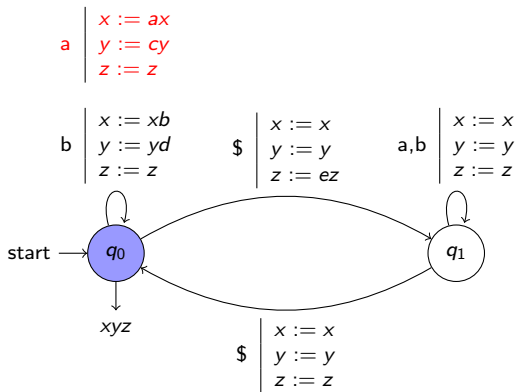
# Streaming String Transducers [Alur, Cerny 11]



$abb\$aa\$ba$   
 ↑

$x = \epsilon$   
 $y = \epsilon$   
 $z = \epsilon$

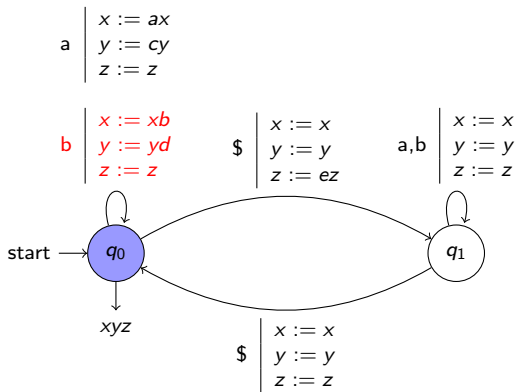
# Streaming String Transducers [Alur, Cerny 11]



a bb\$aa\$ba  
 ↑

x = a  
 y = c  
 z = ε

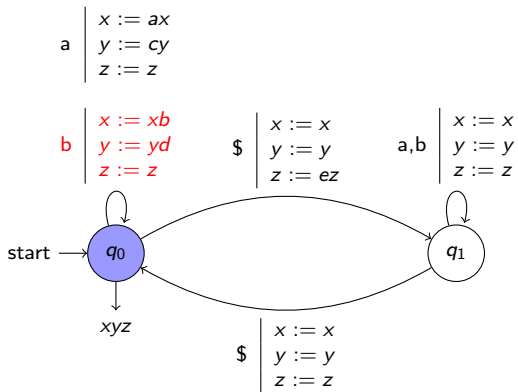
# Streaming String Transducers [Alur, Cerny 11]



a b b \$ a a \$ b a  
 ↑

$x = ab$   
 $y = cd$   
 $z = \epsilon$

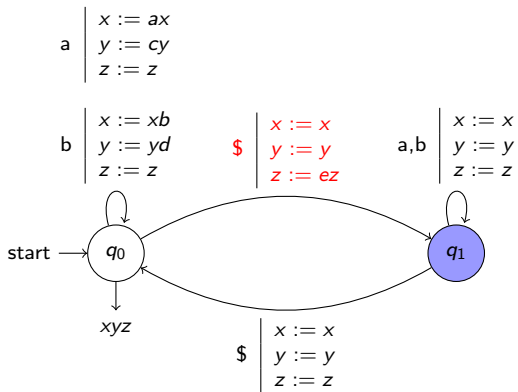
# Streaming String Transducers [Alur, Cerny 11]



ab**b**\$aa\$ba  
 ↑

$x = abb$   
 $y = cdd$   
 $z = \epsilon$

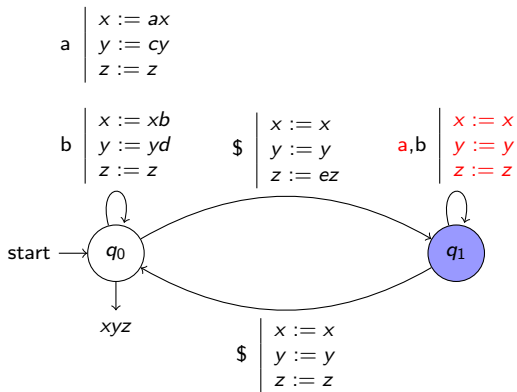
# Streaming String Transducers [Alur, Cerny 11]



abb\$aa\$ba  
 ↑

$x = abb$   
 $y = cdd$   
 $z = e$

# Streaming String Transducers [Alur, Cerny 11]

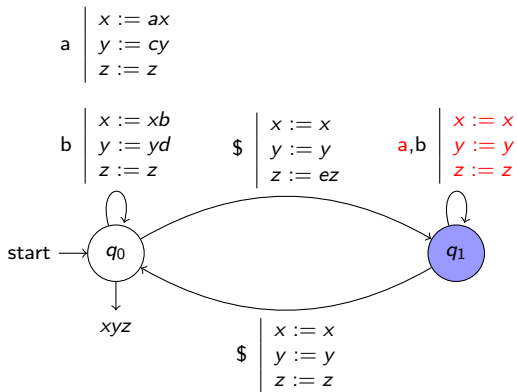


abb\$ **a** a\$ba  
 ↑

$x = abb$   
 $y = cdd$   
 $z = e$



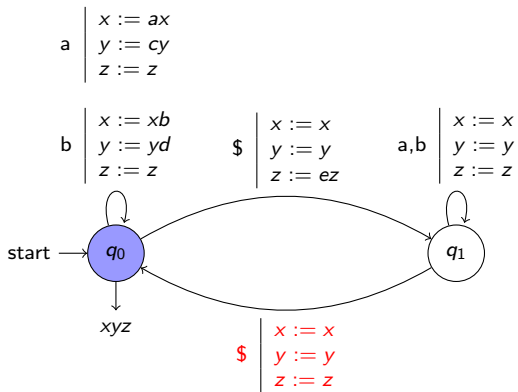
# Streaming String Transducers [Alur, Cerny 11]



abb\$a**a**\$ba  
 ↑

$x = abb$   
 $y = cdd$   
 $z = e$

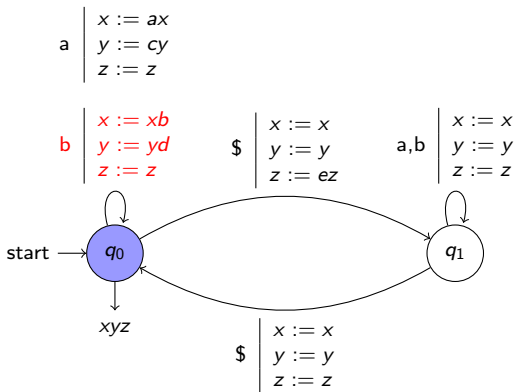
# Streaming String Transducers [Alur, Cerny 11]



abb\$aa\$ba  
 ↑

$x = abb$   
 $y = cdd$   
 $z = e$

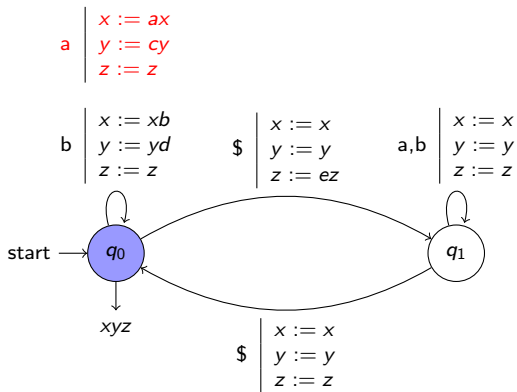
# Streaming String Transducers [Alur, Cerny 11]



abb\$aa\$ba  
 ↑

$x = abbb$   
 $y = cddd$   
 $z = e$

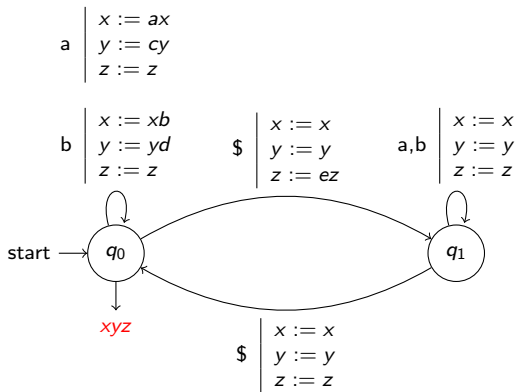
# Streaming String Transducers [Alur, Cerny 11]



abb\$aa\$ba  
 ↑

$x = aabbb$   
 $y = cdddd$   
 $z = e$

# Streaming String Transducers [Alur, Cerny 11]



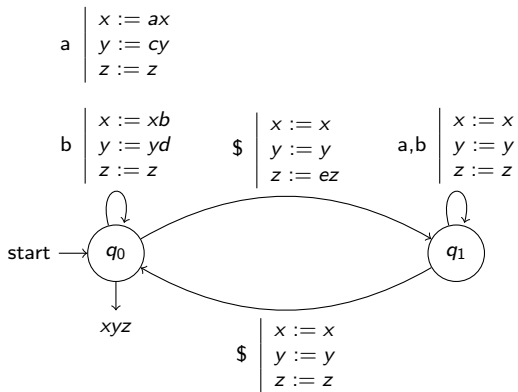
abb\$aa\$ba



$x = aabbb$   
 $y = ccddd$   
 $z = e$

**Output:** aabbbccddde

# Streaming String Transducers [Alur, Cerny 11]



$abb$aa$ba$

$x = aabbb$   
 $y = ccddd$   
 $z = e$

**Output:** aabbbccdde

$$\{(w_0\$u_0\$w_1\$u_1\$ \dots \$u_{p-1}\$w_p, a^n b^m c^n d^m e^p) \mid |w_1 \dots w_p|_a = n \wedge |w_1 \dots w_p|_b = m\}$$

## Known properties

### Theorem (Alur Cerny 10)

*deterministic/functional SST (DSST) = deterministic MSOT = 2-DGSM*

### Theorem (Alur, Deshumukh 11)

- ▶ *non-deterministic SST (NSST) = MSOT*
- ▶ *NSST  $\neq$  2-NGSM*
- ▶ *equivalence of functional NSST is decidable,*
- ▶ *k-valuedness of NSST is decidable.*

## Open problems

We have:

- ▶  $DSST = 2\text{-DGSM}$  and  $NSST \neq 2\text{-NGSM}$  (also hold in the functional case)
- ▶ equivalence of DSST is decidable and equivalence of NSST is undecidable



## Open problems

We have:

- ▶  $DSST = 2\text{-DGSM}$  and  $NSST \neq 2\text{-NGSM}$  (also hold in the functional case)
- ▶ equivalence of DSST is decidable and equivalence of NSST is undecidable

For GSM we have:

- ▶ equivalence of DGSM is decidable

# Open problems

We have:

- ▶  $DSST = 2\text{-DGSM}$  and  $NSST \neq 2\text{-NGSM}$  (also hold in the functional case)
- ▶ equivalence of DSST is decidable and equivalence of NSST is undecidable

For GSM we have:

- ▶ equivalence of DGSM is decidable
- ▶ equivalence of NGSM is undecidable

# Open problems

We have:

- ▶  $DSST = 2\text{-DGSM}$  and  $NSST \neq 2\text{-NGSM}$  (also hold in the functional case)
- ▶ equivalence of DSST is decidable and equivalence of NSST is undecidable

For GSM we have:

- ▶ equivalence of DGSM is decidable
- ▶ equivalence of NGSM is undecidable
- ▶ equivalence of  $k$ -valued NGSM is decidable

# Open problems

We have:

- ▶  $DSST = 2\text{-DGSM}$  and  $NSST \neq 2\text{-NGSM}$  (also hold in the functional case)
- ▶ equivalence of DSST is decidable and equivalence of NSST is undecidable

For GSM we have:

- ▶ equivalence of DGSM is decidable
- ▶ equivalence of NGSM is undecidable
- ▶ equivalence of  $k$ -valued NGSM is decidable

Questions:

- ▶ Is the effect of  $k$ -valuedness similar on NSST and NGSM?
- ▶ Are  $k$ -valued NSST and  $k$ -valued 2-NGSM equivalent?

## Conjecture coming from $k$ -valued GSM

Theorem (Weber 96, Sakarovitch de Souza 08)

*If  $\mathcal{T}$  is a  $k$ -valued GSM then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional GSMs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .*

## Conjecture coming from $k$ -valued GSM

Theorem (Weber 96, Sakarovitch de Souza 08)

*If  $\mathcal{T}$  is a  $k$ -valued GSM then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional GSMs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .*

Conjecture

If  $\mathcal{T}$  is a  $k$ -valued NSST then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional NSSTs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .

# The conjecture and the open problems

## Conjecture

If  $\mathcal{T}$  is a  $k$ -valued NSST then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional NSSTs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .

# The conjecture and the open problems

## Conjecture

If  $\mathcal{T}$  is a  $k$ -valued NSST then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional NSSTs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .

## $k$ -valued NSST and $k$ -valued 2-NGSM

- ▶  $k$ -valued 2-NGSM  $\subseteq$   $k$ -valued NSST: easy,
- ▶  $k$ -valued NSST  $\subseteq$   $k$ -valued 2-NGSM: based on the equivalence of functional NSSTs and functional 2-NGSM.



# The conjecture and the open problems

## Conjecture

If  $\mathcal{T}$  is a  $k$ -valued NSST then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional NSSTs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .

## $k$ -valued NSST and $k$ -valued 2-NGSM

- ▶  $k$ -valued 2-NGSM  $\subseteq$   $k$ -valued NSST: easy,
- ▶  $k$ -valued NSST  $\subseteq$   $k$ -valued 2-NGSM: based on the equivalence of functional NSSTs and functional 2-NGSM.

## Equivalence problem for of $k$ -valued NSST

It reduces to the decidability of  $k$ -valuedness.

# The conjecture and the open problems

## Conjecture

If  $\mathcal{T}$  is a  $k$ -valued NSST then there are  $\mathcal{T}_1, \dots, \mathcal{T}_k$  functional NSSTs so that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$ .

## $k$ -valued NSST and $k$ -valued 2-NGSM

- ▶  $k$ -valued 2-NGSM  $\subseteq$   $k$ -valued NSST: easy,
- ▶  $k$ -valued NSST  $\subseteq$   $k$ -valued 2-NGSM: based on the equivalence of functional NSSTs and functional 2-NGSM.

## Equivalence problem for of $k$ -valued NSST

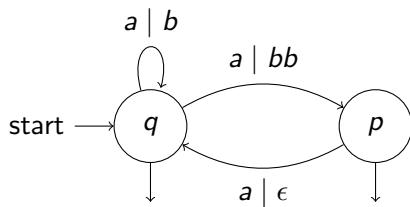
It reduces to the decidability of  $k$ -valuedness.

If  $\mathcal{T}$  and  $\mathcal{T}'$  are  $k$ -valued, the conjecture implies that  $\mathcal{T} = \mathcal{T}_1 \cup \dots \cup \mathcal{T}_k$  and  $\mathcal{T}' = \mathcal{T}'_1 \cup \dots \cup \mathcal{T}'_k$ :

$\mathcal{T} = \mathcal{T}'$  iff for every  $i \in [1, k]$ , both  $\mathcal{T}'_i \subseteq \mathcal{T}$  and  $\mathcal{T}_i \subseteq \mathcal{T}'$  (inclusion of functional NSSTs into NSSTs is decidable).

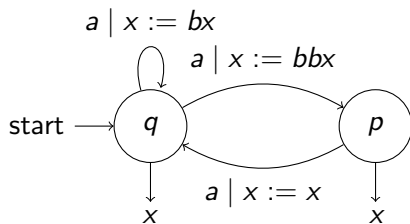
Decomposition of  $k$ -valued rational transducers,  
by Sakarovitch and de Souza

# Rational transducers



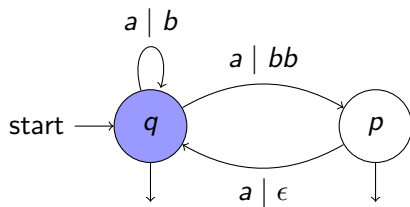
*aaaaaa*

# Rational transducers



*aaaaaa*

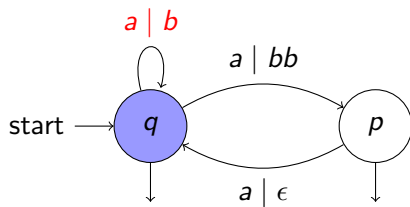
# Rational transducers



*aaaaaa*

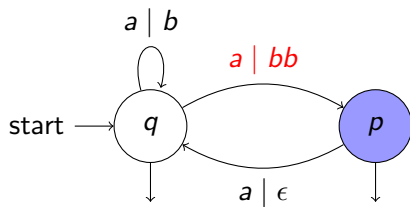
↑

# Rational transducers



$a$   $aaaaa$        $b$   
↑

# Rational transducers



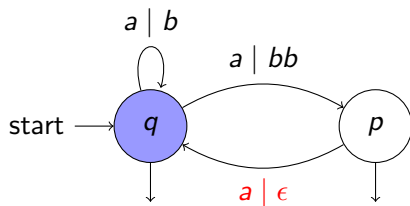
$a$   $a$   $aaaa$

↑

$bbb$



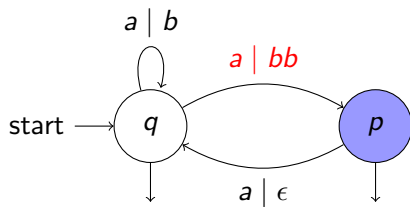
# Rational transducers



$aa$   $a$   $aaa$   
↑

$bbb$

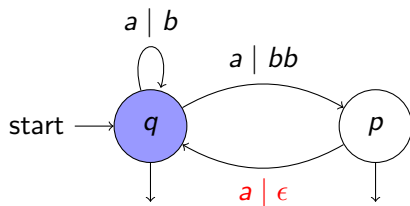
# Rational transducers



*aaa* *a* *aa*  
↑

*bbb* *bb*

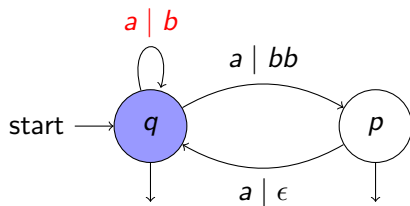
# Rational transducers



aaaa **a** a  
↑

bbbb

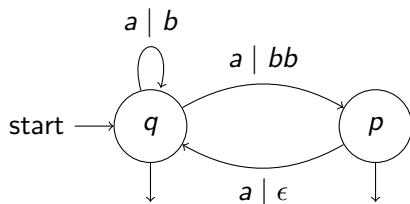
# Rational transducers



aaaaa **a**  
↑

bbbbbb **b**

## Rational transducers



*aaaaaa*

*bbbbbb*

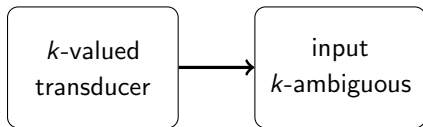
The relation defined by this transducer is:

$$\{(a^k, b^k) \mid k \in \mathbb{N}\} \cup \{(a^k, b^{k+1}) \mid k \in \mathbb{N}\} .$$

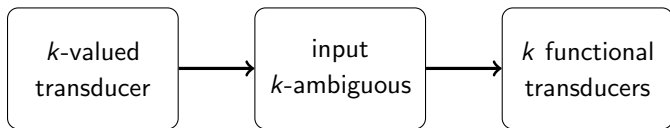
# The steps of the proof

$k$ -valued  
transducer

## The steps of the proof

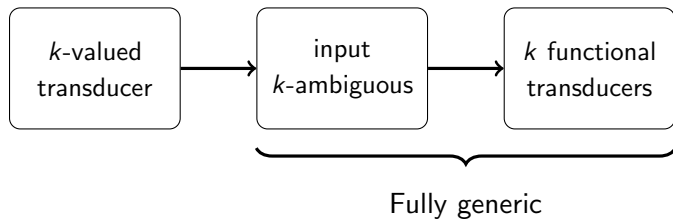


## The steps of the proof

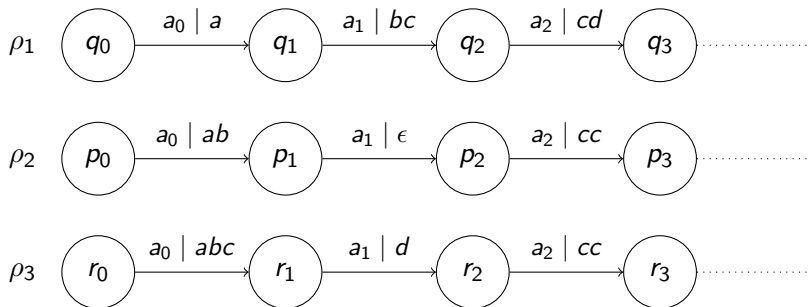




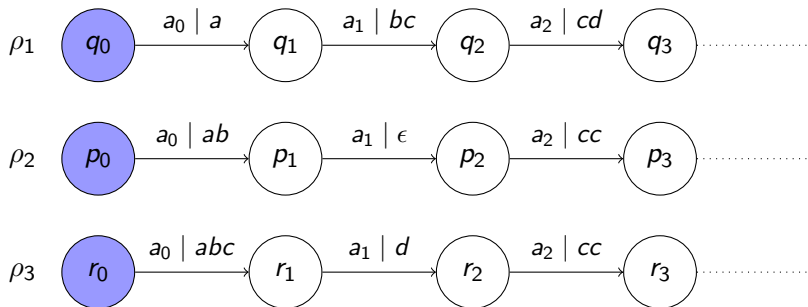
## The steps of the proof



## Lead and Delay between runs

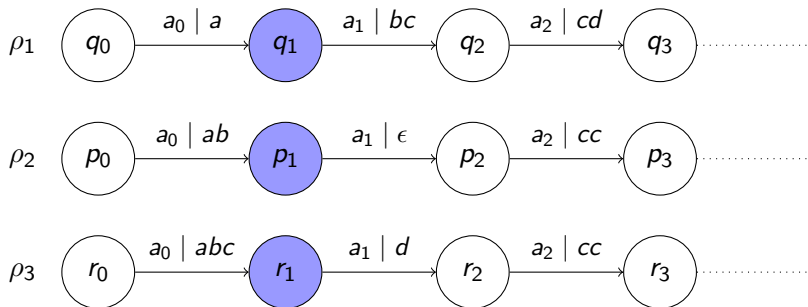


## Lead and Delay between runs



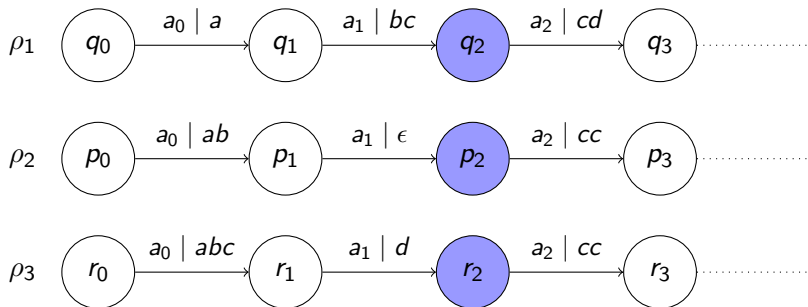
$LD(\rho_1, \rho_2)$	$LD(\rho_1, \rho_3)$	$LD(\rho_2, \rho_3)$
$\epsilon$	$\epsilon$	$\epsilon$

## Lead and Delay between runs



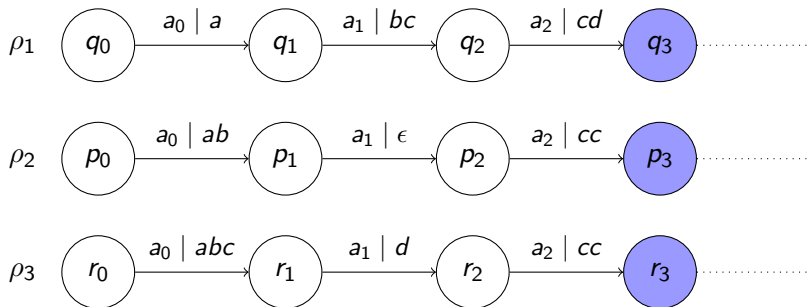
$LD(\rho_1, \rho_2)$	$LD(\rho_1, \rho_3)$	$LD(\rho_2, \rho_3)$
$\epsilon$	$\epsilon$	$\epsilon$
$b$	$bc$	$c$

## Lead and Delay between runs



$LD(\rho_1, \rho_2)$	$LD(\rho_1, \rho_3)$	$LD(\rho_2, \rho_3)$
$\epsilon$	$\epsilon$	$\epsilon$
$b$	$bc$	$c$
$c^{-1}$	$d$	$cd$

## Lead and Delay between runs



$LD(\rho_1, \rho_2)$	$LD(\rho_1, \rho_3)$	$LD(\rho_2, \rho_3)$
$\epsilon$	$\epsilon$	$\epsilon$
$b$	$bc$	$c$
$c^{-1}$	$d$	$cd$
$d^{-1}$	$\perp$	$\perp$

## A theorem by Weber

Fix a  $k$ -valued transducer  $\mathcal{T}$ .

### Theorem

*If  $\rho_1, \dots, \rho_{k+1}$  are runs of  $\mathcal{T}$  on a word  $w$ , then there is  $i$  and  $j$  so that  $i \neq j$  and  $\text{LD}(\rho_i, \rho_j)$  is always smaller than  $Cn^{k+1}$  where:*

- ▶  $n$  is the number of states of  $\mathcal{T}$ ,
- ▶  $C$  is the maximal number of symbols  $\mathcal{T}$  outputs in one transition.

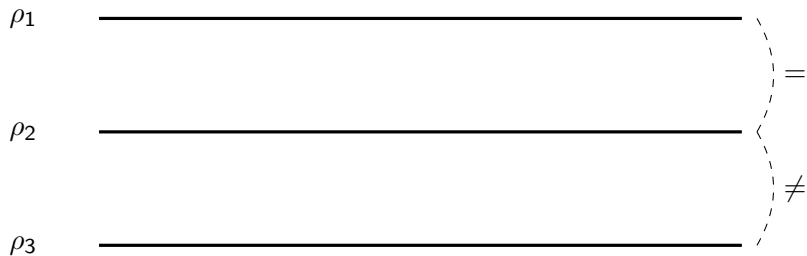
## Results of iterations

Let  $\rho = \rho_1\rho_2\rho_3$  be a run of a rational transducer so that  $\rho_2$  is a loop:

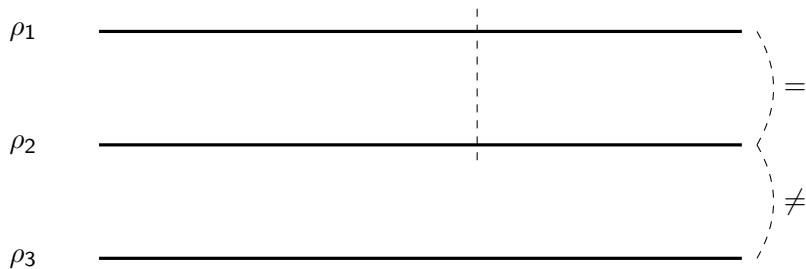
- ▶ if  $u$  is the output of  $\rho$ , then  $u = u_1u_2u_3$  and,
- ▶ for every  $i \in \mathbb{N}$ , the output of  $\rho[i] = \rho_1\rho_2^i\rho_3$  is  $u_1u_2^i u_3$ .



## Sketch of the proof of Weber's Theorem

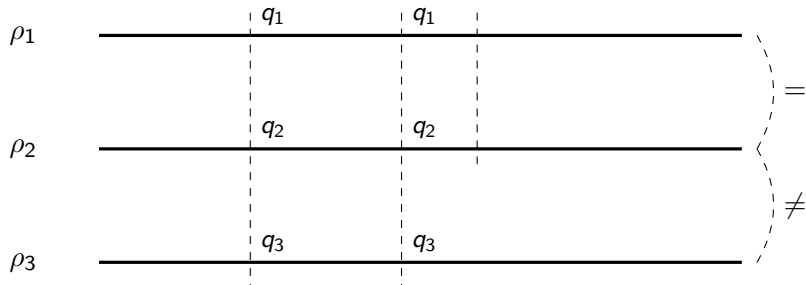


## Sketch of the proof of Weber's Theorem



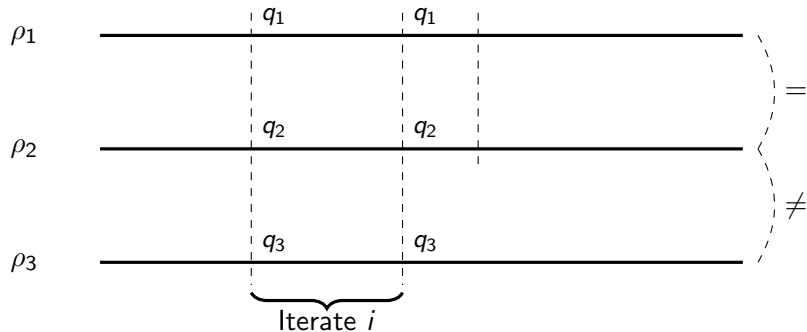
- ▶ At some point  $LD(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,

## Sketch of the proof of Weber's Theorem



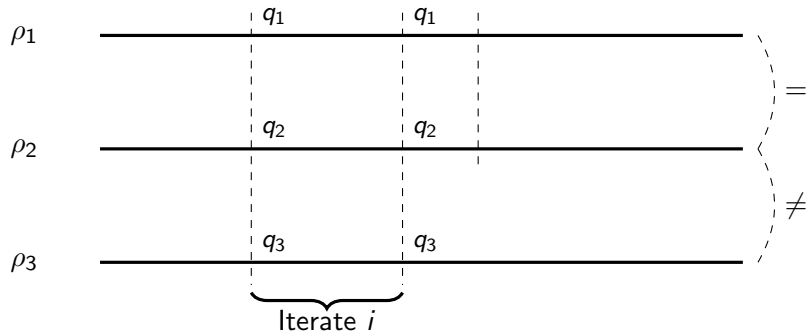
- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,

## Sketch of the proof of Weber's Theorem



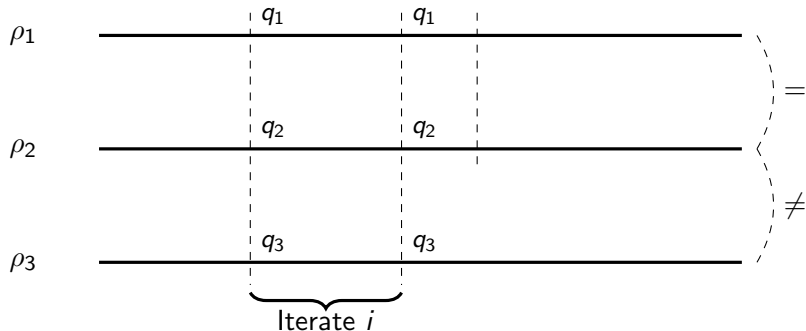
- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,

## Sketch of the proof of Weber's Theorem



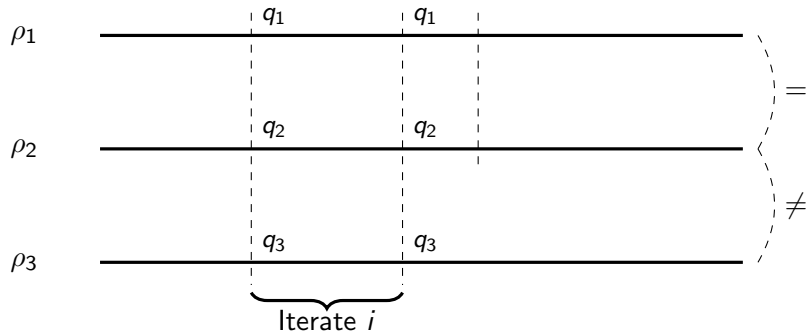
- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ for some  $N$ , for every  $i > N$ ,  $\text{out}(\rho_1[i]) \neq \text{out}(\rho_2[i])$ ,

## Sketch of the proof of Weber's Theorem



- ▶ At some point  $LD(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ for some  $N$ , for every  $i > N$ ,  $\text{out}(\rho_1[i]) \neq \text{out}(\rho_2[i])$ ,
- ▶ for some  $M$ , for every  $i > M$ ,  $\text{out}(\rho_2[i]) \neq \text{out}(\rho_3[i])$ ,

## Sketch of the proof of Weber's Theorem



- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ for some  $N$ , for every  $i > N$ ,  $\text{out}(\rho_1[i]) \neq \text{out}(\rho_2[i])$ ,
- ▶ for some  $M$ , for every  $i > M$ ,  $\text{out}(\rho_2[i]) \neq \text{out}(\rho_3[i])$ ,
- ▶ indeed  $\text{out}(\rho_2[i]) = u_1 u_2^i u_3$  and  $\text{out}(\rho_3[i]) = v_1 v_2^i v_3$ , and if for infinitely many  $i$ 's,  $u_1 u_2^i u_3 = v_1 v_2^i v_3$ , then the identity holds for every  $i$ .

## Making a $k$ ambiguous transducer

Let  $\mathcal{T}$  be a  $k$ -valued transducer

- ▶ put an arbitrary order on the transitions of  $\mathcal{T}$ ,



## Making a $k$ ambiguous transducer

Let  $\mathcal{T}$  be a  $k$ -valued transducer

- ▶ put an arbitrary order on the transitions of  $\mathcal{T}$ ,
- ▶ the runs are ordered lexicographically with respect to that order,

## Making a $k$ ambiguous transducer

Let  $\mathcal{T}$  be a  $k$ -valued transducer

- ▶ put an arbitrary order on the transitions of  $\mathcal{T}$ ,
- ▶ the runs are ordered lexicographically with respect to that order,
- ▶ execute a possible run of  $\mathcal{T}$ ,

## Making a $k$ ambiguous transducer

Let  $\mathcal{T}$  be a  $k$ -valued transducer

- ▶ put an arbitrary order on the transitions of  $\mathcal{T}$ ,
- ▶ the runs are ordered lexicographically with respect to that order,
- ▶ execute a possible run of  $\mathcal{T}$ ,
- ▶ for every state maintain the possible leads and delays bounded by  $Cn^{k+1}$  of every run:
  - ▶ on the same input and
  - ▶ that is lexicographically smaller than the run being executed

## Making a $k$ ambiguous transducer

Let  $\mathcal{T}$  be a  $k$ -valued transducer

- ▶ put an arbitrary order on the transitions of  $\mathcal{T}$ ,
- ▶ the runs are ordered lexicographically with respect to that order,
- ▶ execute a possible run of  $\mathcal{T}$ ,
- ▶ for every state maintain the possible leads and delays bounded by  $Cn^{k+1}$  of every run:
  - ▶ on the same input and
  - ▶ that is lexicographically smaller than the run being executed
- ▶ accept only in the end the leads and delays of the accepting states are empty.

## Recap

- ▶ the key ingredient of the proof is to find a way of comparing runs s.t.:

## Recap

- ▶ the key ingredient of the proof is to find a way of comparing runs s.t.:
  - ▶ this comparison can be done in finite memory,

# Recap

- ▶ the key ingredient of the proof is to find a way of comparing runs s.t.:
  - ▶ this comparison can be done in finite memory,
  - ▶ when runs are **far away**, and their output is equal, we should be able to use iterations to make the output of those runs distinct.

Decomposition of  $k$ -valued NSST with 1 register



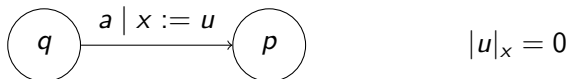
## Normalizing NSST with one register

In an NSST with 1 register, a transition is **deleting** when it is of the form:



## Normalizing NSST with one register

In an NSST with 1 register, a transition is **deleting** when it is of the form:

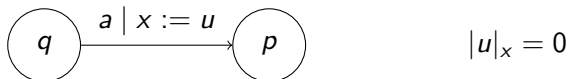


### Lemma

*For every NSST with one register there is an equivalent NSST with no deleting transitions.*

## Normalizing NSST with one register

In an NSST with 1 register, a transition is **deleting** when it is of the form:



### Lemma

*For every NSST with one register there is an equivalent NSST with no deleting transitions.*

From now on we only consider NSSTs that have no deleting rules.

## Defining lead and delay

The main difference between NSSTs with one register and rational transducers is that:

- ▶ rational transducers write from left to right,

## Defining lead and delay

The main difference between NSSTs with one register and rational transducers is that:

- ▶ rational transducers write from left to right,
- ▶ NSSTs with one register can write both to the left and to the right.

## Defining lead and delay

The main difference between NSSTs with one register and rational transducers is that:

- ▶ rational transducers write from left to right,
- ▶ NSSTs with one register can write both to the left and to the right.

A natural way of comparing the contents  $x$  and  $y$  of registers of two runs is as follows:

$$\text{LD}(x, y) = \{(u_1, u_2, v_1, v_2) \mid u_1xu_2 = v_1yv_2\}$$

$$\Delta(x, y) = \min\{|u_1u_2v_1v_2| \mid (u_1, u_2, v_1, v_2) \in \text{LD}(x, y)\}$$

# Weber Theorem for NSST with 1 register

Fix a  $k$ -valued NSST with 1 register  $\mathcal{T}$ .

## Theorem

*If  $\rho_1, \dots, \rho_{k+1}$  are runs of  $\mathcal{T}$  on a word  $w$ , then there is  $i$  and  $j$  so that  $i \neq j$  and  $\text{LD}(\rho_i, \rho_j)$  is always smaller than  $Cn^{k+1}$  where:*

- ▶  $n$  is the number of states of  $\mathcal{T}$ ,
- ▶  $C$  is the maximal number of symbols  $\mathcal{T}$  outputs in one transition.

# A technical lemma by Saarela

## Lemma

*If the identity:*

$$u_1 v_1^i u_2 v_2^i \dots u_n v_n^i u_{n+1} = s_1 t_1^i s_2 t_2^i \dots s_p t_p^i s_{p+1}$$

*holds for more than  $n + p$  values of  $i$ , then it holds for all  $i$ .*

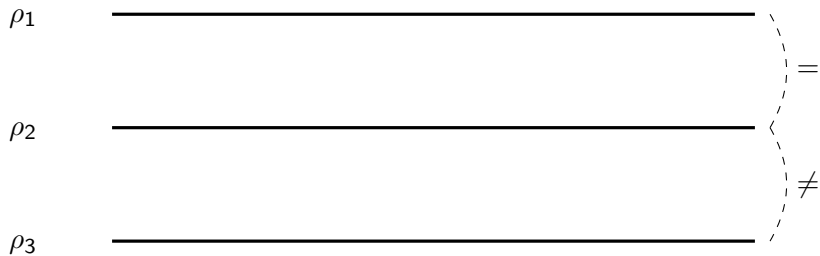


## Simultaneous iterations of loop

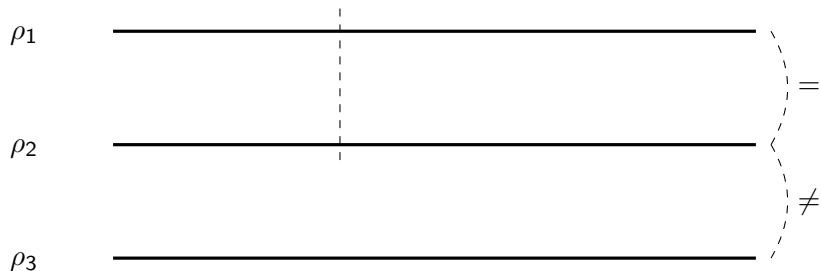
If  $\rho = \rho_1 \rho'_1 \dots \rho_n \rho'_n \rho_{n+1}$  is a run of an NSST with one register so that the  $\rho'_i$  are loops,

- ▶ if the output of  $\rho$  is  $u$ , then  $u = u_1 v_1 \dots u_{2n} v_{2n} u_{2n+1}$  so that:
- ▶ for every  $i \in \mathbb{N}$ , the output of  $\rho[i] = \rho_1 \rho_1^i \dots \rho_n \rho_n^i \rho_{n+1}$  is  $u_1 v_1^i \dots u_{2n} v_{2n}^i u_{2n+1}$ .

## Sketch of the proof

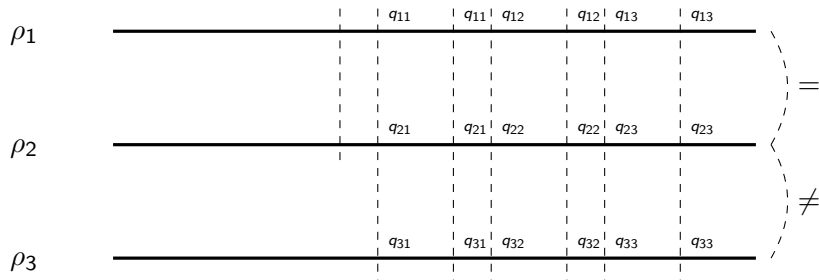


## Sketch of the proof



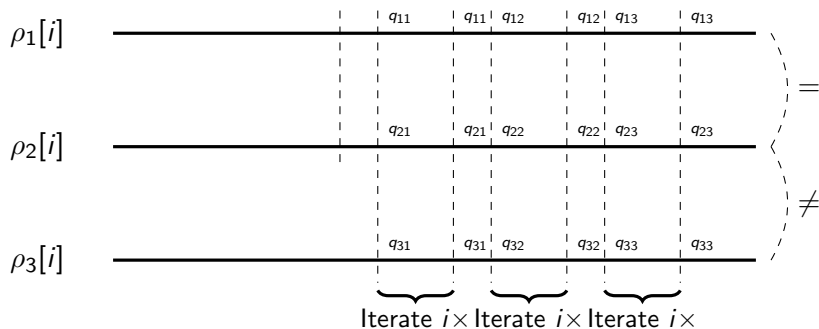
- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,

## Sketch of the proof



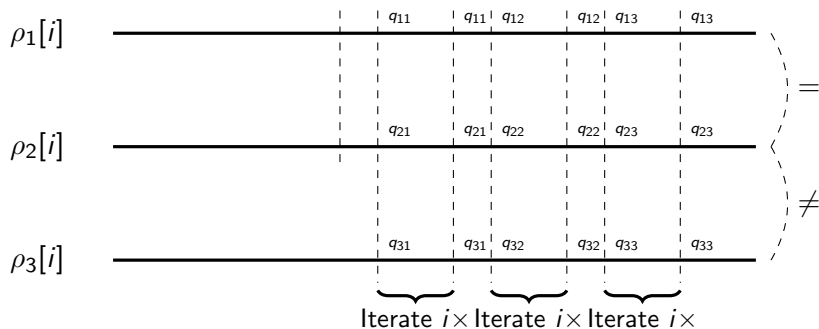
- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ take all the loops in the second half,

## Sketch of the proof



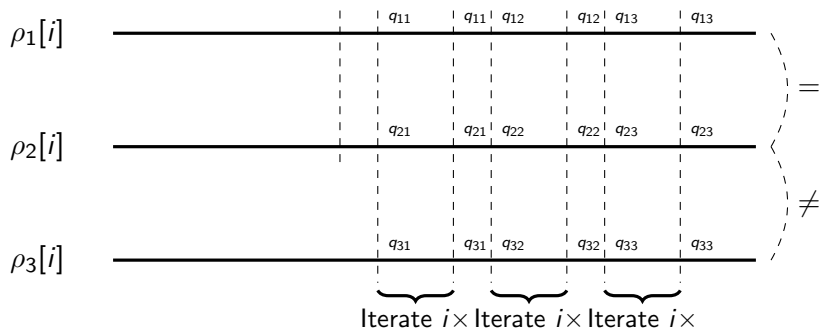
- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ take all the loops in the second half,

## Sketch of the proof



- ▶ At some point  $LD(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ take all the loops in the second half,
- ▶ Using Saarela's Lemma:
  - ▶ for some  $N$ , for every  $i > N$ ,  $\text{out}(\rho_1[i]) \neq \text{out}(\rho_2[i])$ ,

## Sketch of the proof



- ▶ At some point  $\text{LD}(\rho_1, \rho_2)$  has size greater than  $Cn^3$ ,
- ▶ take all the loops in the second half,
- ▶ Using Saarela's Lemma:
  - ▶ for some  $N$ , for every  $i > N$ ,  $\text{out}(\rho_1[i]) \neq \text{out}(\rho_2[i])$ ,
  - ▶ for some  $M$ , for every  $i > M$ ,  $\text{out}(\rho_2[i]) \neq \text{out}(\rho_3[i])$ ,

## A technical problem

Just knowing the set

$$\text{LD}^{\leq N}(x, y) = \{(u_1, u_2, v_1, v_2) \mid (u_1, u_2, v_1, v_2) \in \text{LD}(x, y) \wedge |u_1 u_2 v_1 v_2| \leq N\}$$

is not enough to compute

$$\text{LD}^{\leq N}(uxw, syt)$$



## A technical problem

Just knowing the set

$$\text{LD}^{\leq N}(x, y) = \{(u_1, u_2, v_1, v_2) \mid (u_1, u_2, v_1, v_2) \in \text{LD}(x, y) \wedge |u_1 u_2 v_1 v_2| \leq N\}$$

is not enough to compute

$$\text{LD}^{\leq N}(uxw, syt)$$

For constructing the  $k$  ambiguous transducer we save the information:

$$\text{LD}^{\leq N}(x, y), \text{Per}(x, y)$$

where  $\text{Per}(x, y)$  contains information about the periodicities inside  $x$  and  $y$ .

Towards the general result

## The natural generalization of lead and delay

When dealing with NSSTs with  $n$  registers, it is natural to define the lead and delay as follows:

$$\text{LD}((u_1, \dots, u_n), (v_1, \dots, v_n)) = \{ ((s_1, \dots, s_{n+1}), (t_1, \dots, t_{n+1})) \mid s_1 u_1 \dots u_n s_{n+1} = t_1 v_1 \dots v_n t_{n+1} \}$$

## The natural generalization of lead and delay

When dealing with NSSTs with  $n$  registers, it is natural to define the lead and delay as follows:

$$\text{LD}((u_1, \dots, u_n), (v_1, \dots, v_n)) = \{ ((s_1, \dots, s_{n+1}), (t_1, \dots, t_{n+1})) \mid s_1 u_1 \dots u_n s_{n+1} = t_1 v_1 \dots v_n t_{n+1} \}$$

- ▶ Even for bounded sizes, these sets cannot be maintained with a finite memory.

## The natural generalization of lead and delay

When dealing with NSSTs with  $n$  registers, it is natural to define the lead and delay as follows:

$$\text{LD}((u_1, \dots, u_n), (v_1, \dots, v_n)) = \{ ((s_1, \dots, s_{n+1}), (t_1, \dots, t_{n+1})) \mid s_1 u_1 \dots u_n s_{n+1} = t_1 v_1 \dots v_n t_{n+1} \}$$

- ▶ Even for bounded sizes, these sets cannot be maintained with a finite memory.
- ▶ To overcome this difficulty, we try to consider weaker notions of leads and delays.

## Difficulties with iterations

If  $\rho = \rho_1\rho_2\rho_3$  is a run of a NSST with  $k$  registers, and  $\rho_2$  is a loop, then it is not the case that the output of  $\rho[i] = \rho_1\rho_2^i\rho_3$  can be represented by iterating the output of  $\rho$ .

## Difficulties with iterations

If  $\rho = \rho_1\rho_2\rho_3$  is a run of a NSST with  $k$  registers, and  $\rho_2$  is a loop, then it is not the case that the output of  $\rho[i] = \rho_1\rho_2^i\rho_3$  can be represented by iterating the output of  $\rho$ .

The ideas we use to overcome this problem:

- ▶ use loops with idempotent flow,

## Difficulties with iterations

If  $\rho = \rho_1\rho_2\rho_3$  is a run of a NSST with  $k$  registers, and  $\rho_2$  is a loop, then it is not the case that the output of  $\rho[i] = \rho_1\rho_2^i\rho_3$  can be represented by iterating the output of  $\rho$ .

The ideas we use to overcome this problem:

- ▶ use loops with idempotent flow,
- ▶ use an induction on the number of registers (c.f. [Kanazawa 09])



## Difficulties with iterations

If  $\rho = \rho_1\rho_2\rho_3$  is a run of a NSST with  $k$  registers, and  $\rho_2$  is a loop, then it is not the case that the output of  $\rho[i] = \rho_1\rho_2^i\rho_3$  can be represented by iterating the output of  $\rho$ .

The ideas we use to overcome this problem:

- ▶ use loops with idempotent flow,
- ▶ use an induction on the number of registers (c.f. [Kanazawa 09])
- ▶ we are then able to prove a generalization of Weber theorem for NSSTs with  $k$  registers for the natural generalization of lead and delay.

Conclusion

# Conclusion

- ▶ we have obtained a decomposition theorem for  $k$ -valued NSST with 1 registers

# Conclusion

- ▶ we have obtained a decomposition theorem for  $k$ -valued NSST with 1 registers
- ▶ we are on the way to solve the general case, but coming up with the right notion of Lead and Delay requires heavy combinatorics on words,

# Conclusion

- ▶ we have obtained a decomposition theorem for  $k$ -valued NSST with 1 registers
- ▶ we are on the way to solve the general case, but coming up with the right notion of Lead and Delay requires heavy combinatorics on words,
- ▶ this decomposition property should resist many extensions of NSSTs:
  - ▶ adding copies,

# Conclusion

- ▶ we have obtained a decomposition theorem for  $k$ -valued NSST with 1 registers
- ▶ we are on the way to solve the general case, but coming up with the right notion of Lead and Delay requires heavy combinatorics on words,
- ▶ this decomposition property should resist many extensions of NSSTs:
  - ▶ adding copies,
  - ▶ translating trees to strings,

# Conclusion

- ▶ we have obtained a decomposition theorem for  $k$ -valued NSST with 1 registers
- ▶ we are on the way to solve the general case, but coming up with the right notion of Lead and Delay requires heavy combinatorics on words,
- ▶ this decomposition property should resist many extensions of NSSTs:
  - ▶ adding copies,
  - ▶ translating trees to strings,
- ▶ but the combinatorics seems to become harder and harder. . .