Université Libre de Bruxelles

Aperiodic Two-way Transducers and FO-Transductions

Olivier Carton [1], Luc Dartois [2]

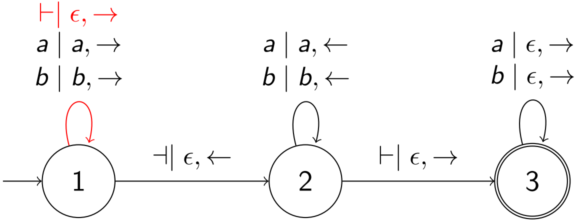
October 22^{th}, 2015

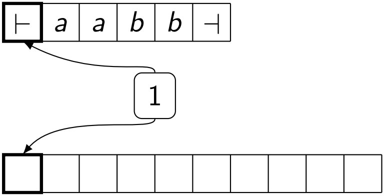
1. LIAFA, Université Paris-Diderot, France
2. Université Libre de Bruxelles, Belgium

1. Two-way transducers
   - Example
   - Relevance of two-way
2. Logic transductions
   - **MSO**-transductions
   - Case of **FO**
3. Transition monoid of a 2-way automata
4. Equivalence result and proof ideas
5. Related work and Conclusion

A two-way transducer

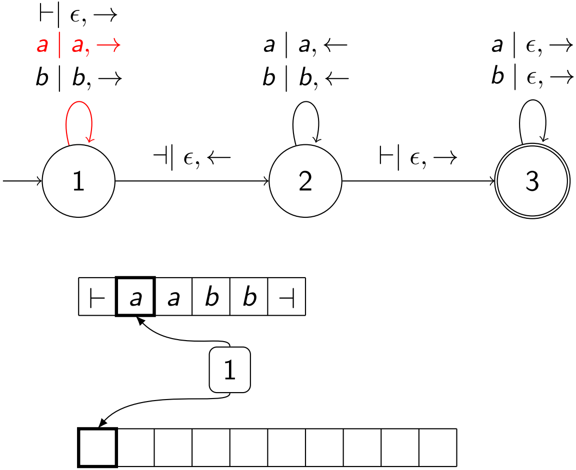
This two-way transducer realizes the function : $w \rightarrow w\tilde{w}$.





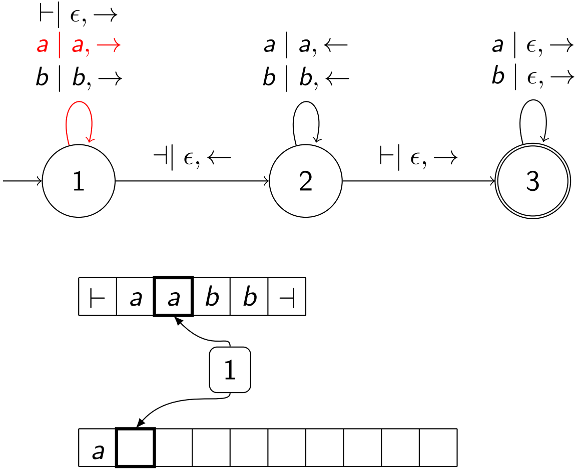
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



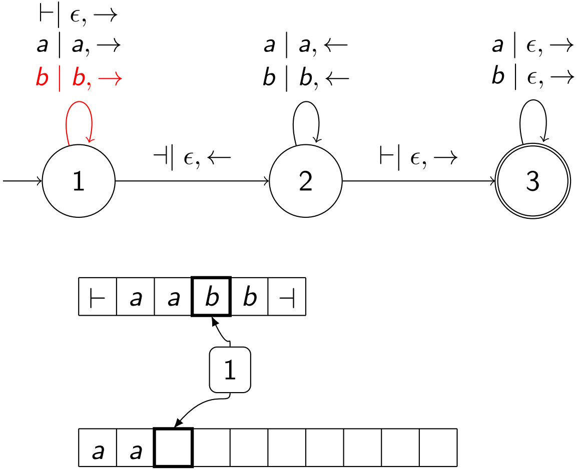
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



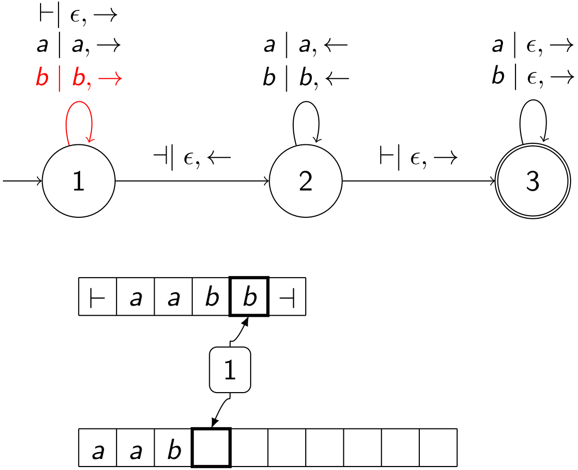
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



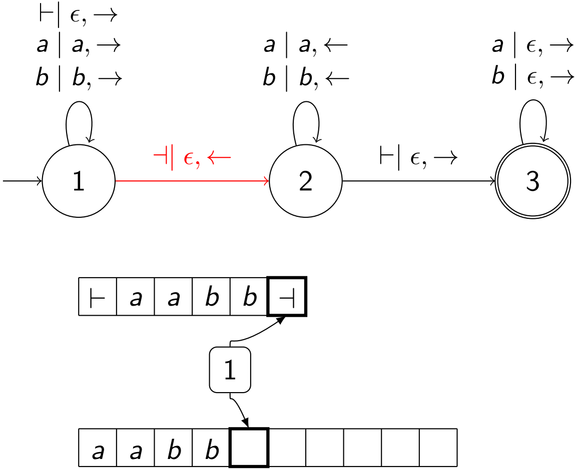
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



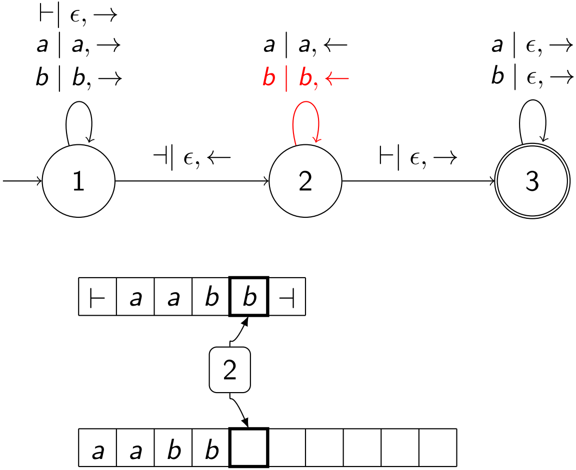
A two-way transducer

This two-way transducer realizes the function : $w \rightarrow w\tilde{w}$.



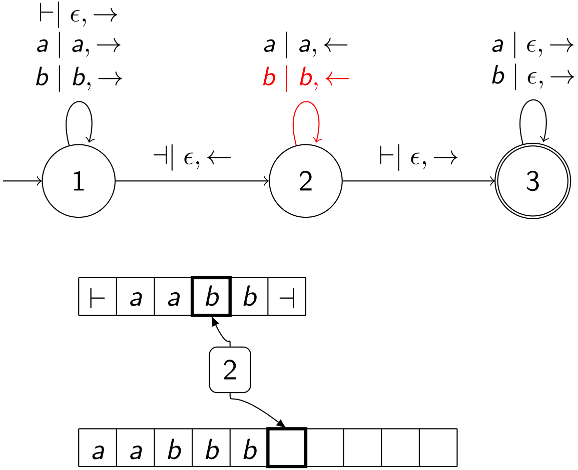
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



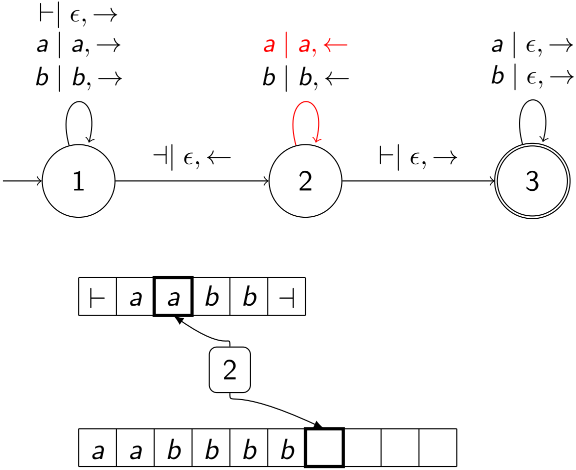
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



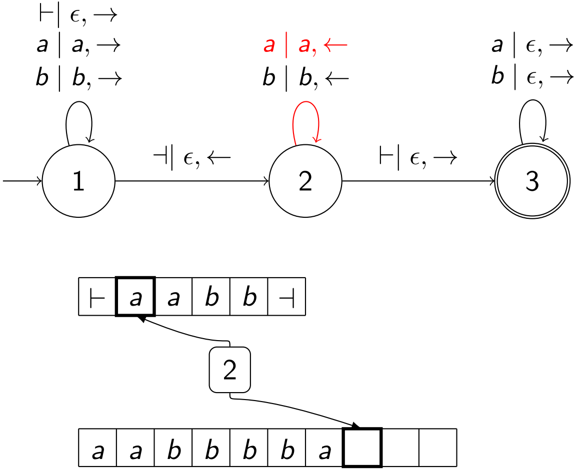
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



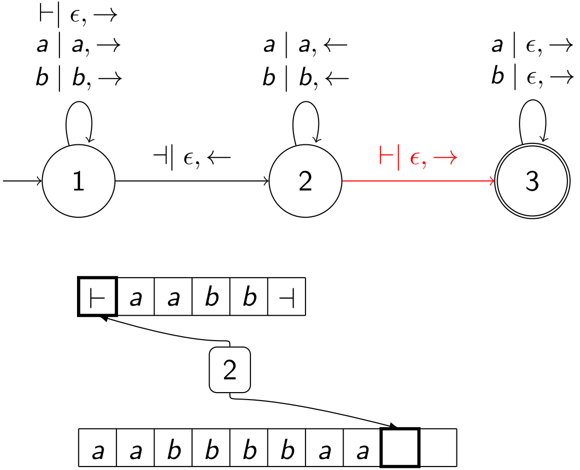
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



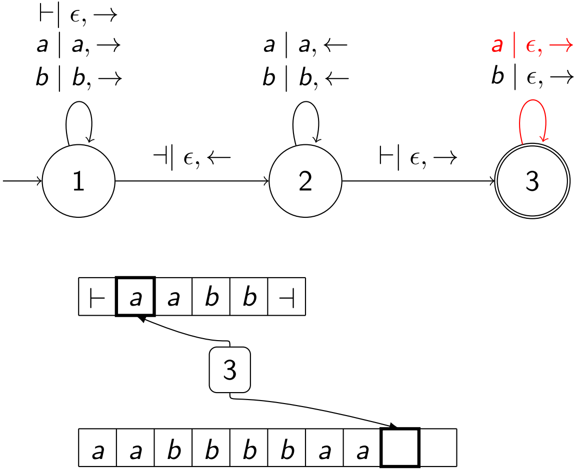
A two-way transducer

This two-way transducer realizes the function : $w \rightarrow w\tilde{w}$.



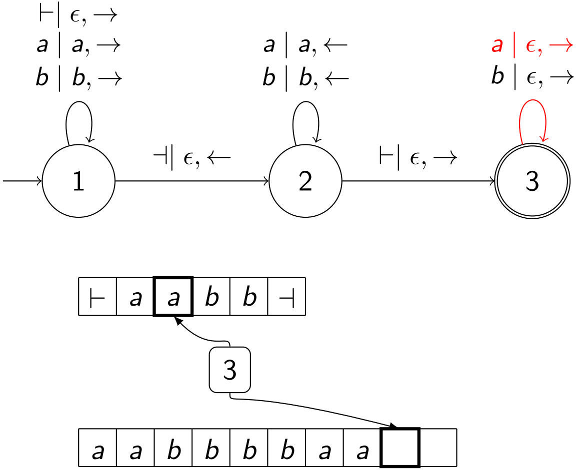
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



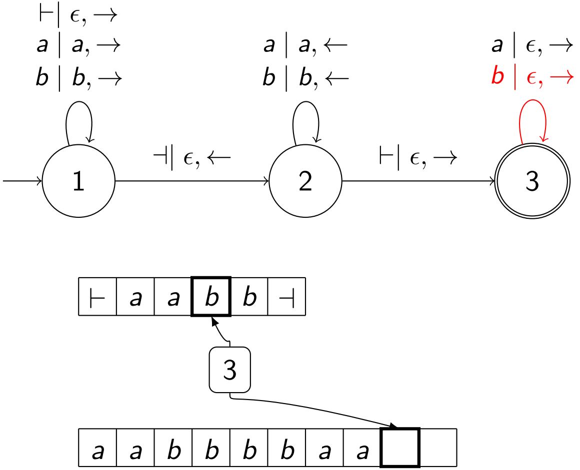
A two-way transducer

This two-way transducer realizes the function : $w \rightarrow w\tilde{w}$.



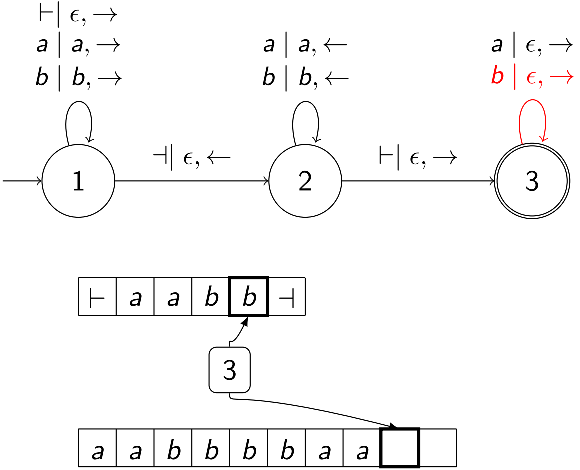
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



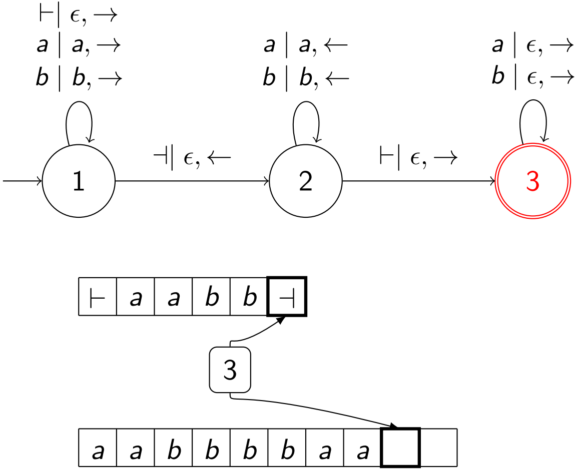
A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.

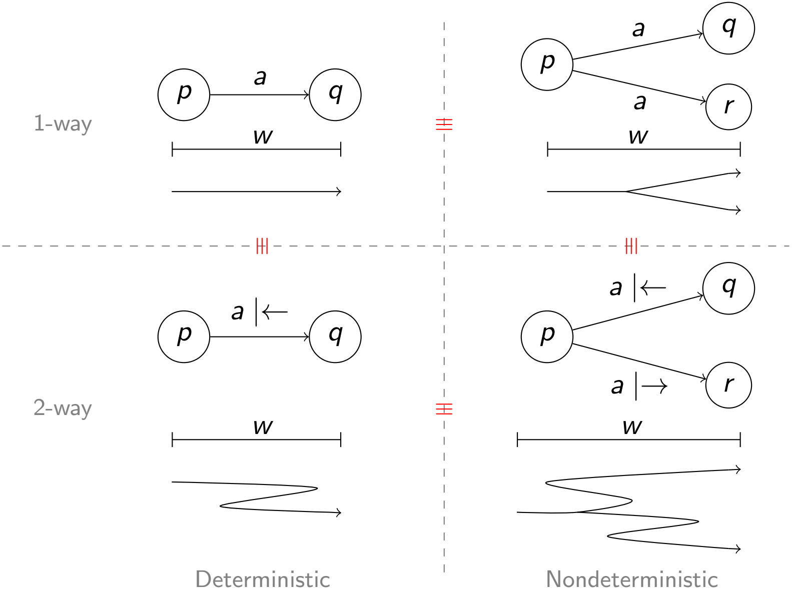


A two-way transducer

This two-way transducer realizes the function : $w \to w\tilde{w}$.



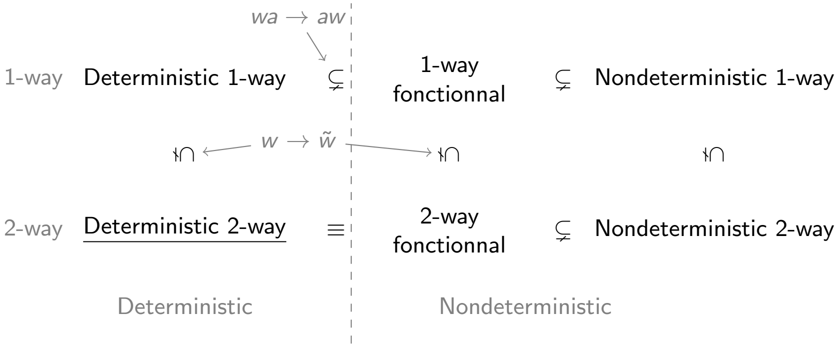
Formalisms of automata



Formalisms of automata with outputs (Transducers)

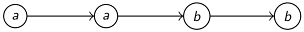
	Deterministic		Nondeterministic
1-way	Deterministic 1-way	$\subsetneq$	Nondeterministic 1-way
	$\not\supseteq$ ← $w \to \tilde{w}$ →		$\not\supseteq$
2-way	Deterministic 2-way	$\subsetneq$	Nondeterministic 2-way

Formalisms of automata



MSO-definable function

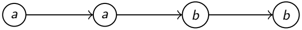
word w $\rightsquigarrow$ linear graph G_w

$aabb$ $\rightsquigarrow$ 

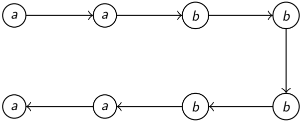
MSO-definable function

word w $\rightsquigarrow$ linear graph G_w

$aabb$ $\rightsquigarrow$



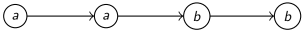
$f(aabb) = aabbbbaa$ $\rightsquigarrow$



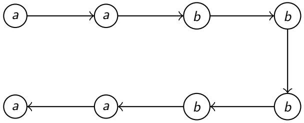
MSO-definable function

word w $\rightsquigarrow$ linear graph G_w

$aabb$ $\rightsquigarrow$



$f(aabb) = aabbbbaa$ $\rightsquigarrow$



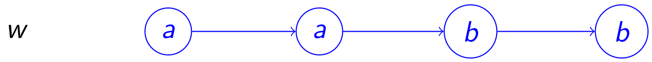
MSO-interpretation [Courcelle94]

MSO formulas over the input graph to reinterpret C copies of it :

- φ_{dom}, the domain formula,
- $\varphi_b^c(x)$, the c copy of x is labelled by b,
- $\varphi^{c,c'}(x,y)$, an edge between the c copy of x and the c' copy of y.

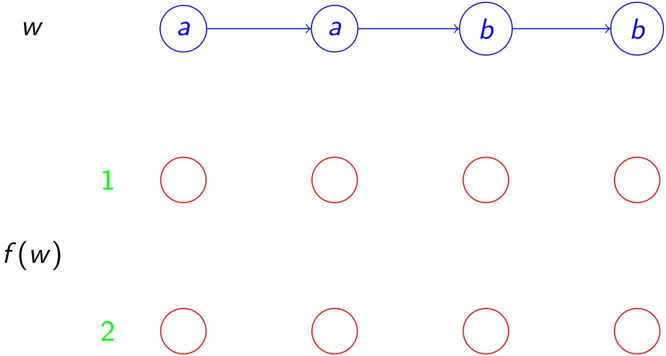
Example of *MSO*-definable function

The same function $f : w \to w\tilde{w}$ is a *MSO* definable function. We construct the image of the word *aabb*.



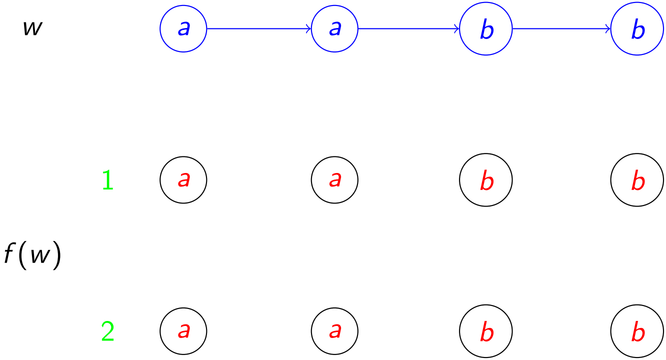
Example of *MSO*-definable function

The same function $f : w \rightarrow w\tilde{w}$ is a *MSO* definable function. We construct the image of the word *aabb*.



Example of *MSO*-definable function

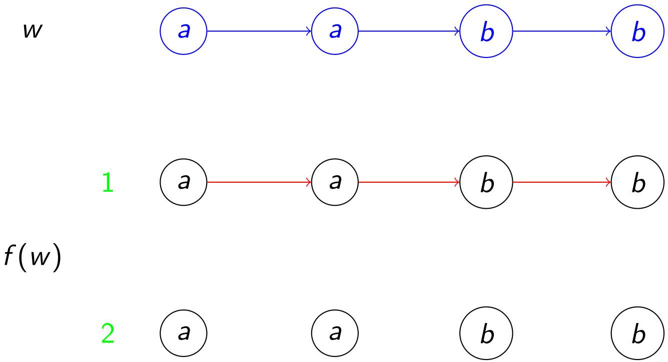
The same function $f : w \to w\tilde{w}$ is a *MSO* definable function. We construct the image of the word *aabb*.



$$lab^1(x) \equiv lab^2(x) \equiv lab(x)$$

Example of *MSO*-definable function

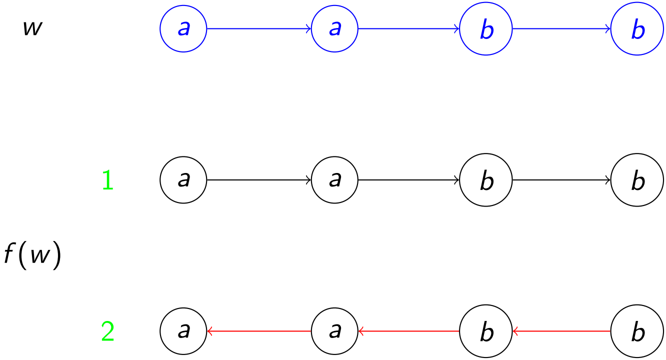
The same function $f : w \to w\tilde{w}$ is a *MSO* definable function. We construct the image of the word *aabb*.



$$edge^{1,1}(x, y) \equiv edge(x, y)$$

Example of *MSO*-definable function

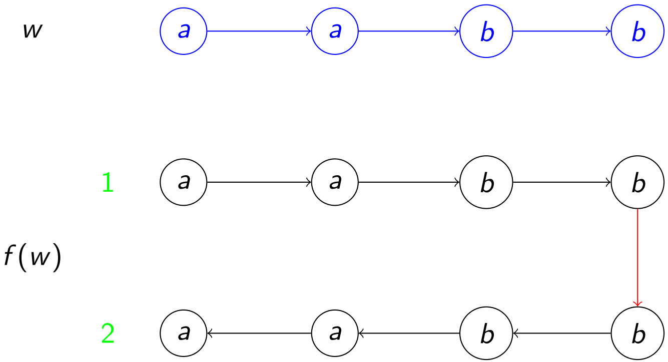
The same function $f : w \rightarrow w\tilde{w}$ is a *MSO* definable function. We construct the image of the word *aabb*.



$$edge^{2,2}(x, y) \equiv edge(y, x)$$

Example of *MSO*-definable function

The same function $f : w \to w\tilde{w}$ is a *MSO* definable function. We construct the image of the word *aabb*.



$$edge^{1,2}(x, y) \equiv (x = y) \wedge \neg(\exists z \; edge(x, z))$$

First-order transductions and aperiodic transducers

Theorem [EH01]

Given a word function f,

- f is defined by an **MSO**-transduction iff
- f is recognized by a deterministic 2-way transducer.

Theorem [MP71,Schutz65]

Given a regular language L,

- L is definable by an **FO**$[<]$ formula iff
- The syntactic monoid of L is aperiodic.

First-order transductions and aperiodic transducers

Theorem [EH01]

Given a word function f,

- f is defined by an **MSO**-transduction iff
- f is recognized by a deterministic 2-way transducer.

Theorem [MP71,Schutz65]

Given a regular language L,

- L is definable by an **FO**$[<]$ formula iff
- The syntactic monoid of L is aperiodic.

Theorem [Carton, D.]

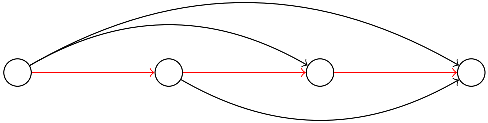
Given a word function f,

- f is defined by an **FO**$[<]$-transduction iff
- f is recognized by an aperiodic and deterministic 2-way transducer.

FO-transductions

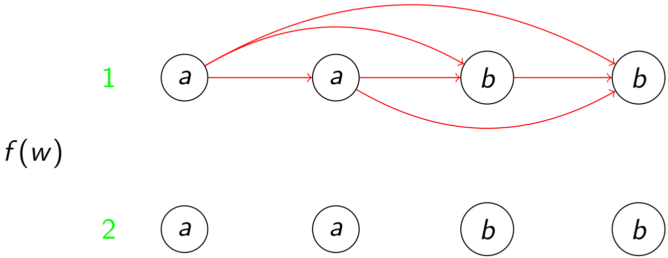
We consider **FO**$[<]$ instead of **FO**$[+1]$, the order being the transitive closure of the successor.
Formally, we replace the *edge* predicate by the *path* predicate in both the input and the output structures.

$$edge^{i,j}(x,y) \rightarrow path^{i,j}(x,y)$$



FO-transductions

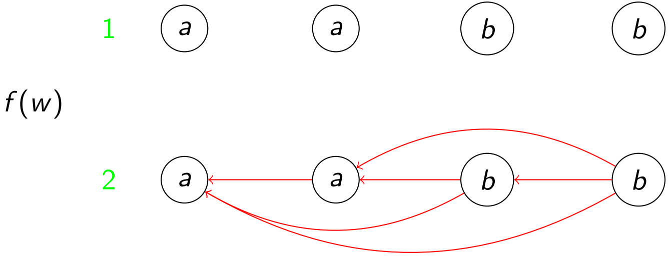
We consider **FO**[<] instead of **FO**[+1], the order being the transitive closure of the successor.
Formally, we replace the *edge* predicate by the *path* predicate in both the input and the output structures.



$$path^{1,1}(x,y) \equiv path(x,y)$$

FO-transductions

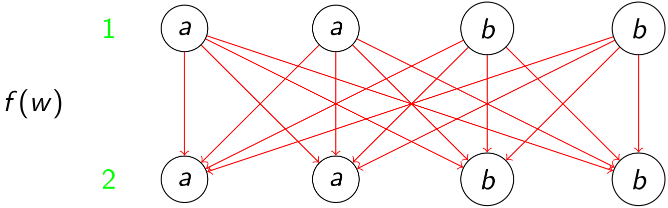
We consider **FO**[$<$] instead of **FO**[$+1$], the order being the transitive closure of the successor.
Formally, we replace the *edge* predicate by the *path* predicate in both the input and the output structures.



$$path^{2,2}(x, y) \equiv path(y, x)$$

FO-transductions

We consider **FO**$[<]$ instead of **FO**$[+1]$, the order being the transitive closure of the successor.
Formally, we replace the *edge* predicate by the *path* predicate in both the input and the output structures.

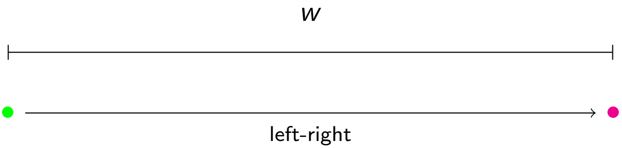


$path^{1,2}(x, y) \equiv true$

Elements of the transition monoid

1-way case :

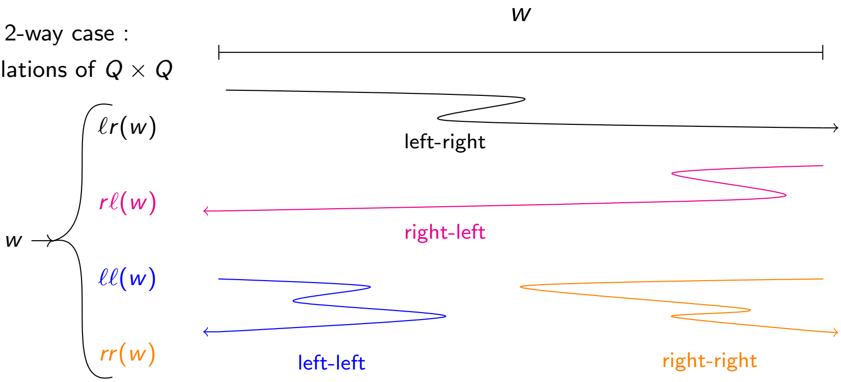
$w \to R_w \subseteq Q \times Q$



Elements of the transition monoid

2-way case :

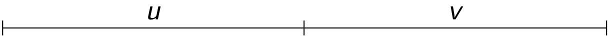
4 relations of $Q \times Q$



4 kinds of partial runs

[Shepherdson 59], [Pécuchet 85], [Birget 89]

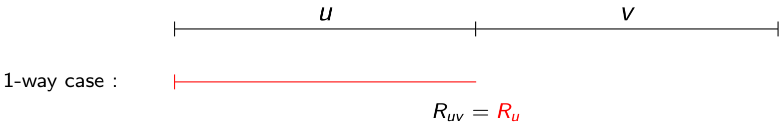
Composition law for runs



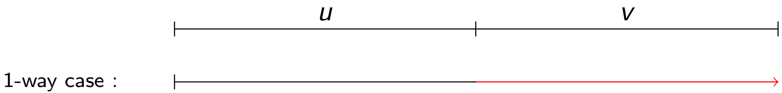
1-way case :

$$R_{uv} =$$

Composition law for runs

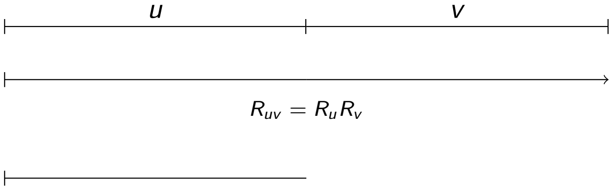


Composition law for runs



$R_{uv} = R_u R_v$

Composition law for runs



1-way case :

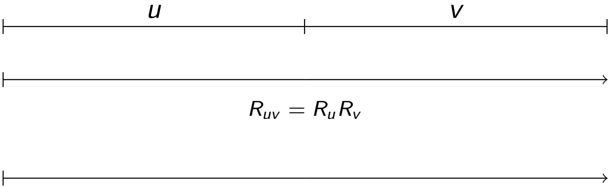
$$R_{uv} = R_u R_v$$

2-way case :

$$\ell r(uv) = \ell r(u)$$

[Shepherdson 59], [Pécuchet 85], [Birget 89]

Composition law for runs



1-way case :

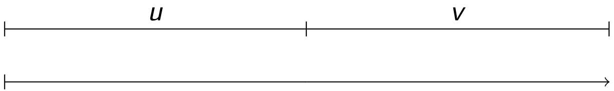
$$R_{uv} = R_u R_v$$

2-way case :

$$\ell r(uv) = \ell r(u) \qquad \ell r(v)$$

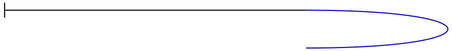
[Shepherdson 59], [Pécuchet 85], [Birget 89]

Composition law for runs



1-way case :

$$R_{uv} = R_u R_v$$

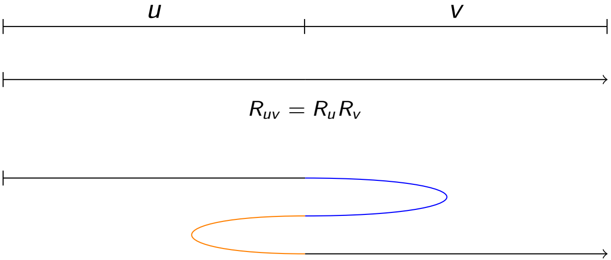


2-way case :

$$\ell r(uv) = \ell r(u)\ \ell\ell(v)$$

[Shepherdson 59], [Pécuchet 85], [Birget 89]

Composition law for runs



1-way case :

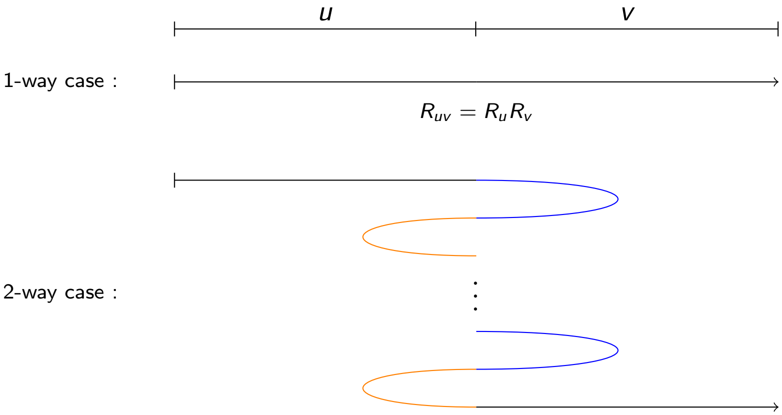
$R_{uv} = R_u R_v$

2-way case :

$\ell r(uv) = \ell r(u)\ \ell\ell(v) rr(u)\quad \ell r(v)$

[Shepherdson 59], [Pécuchet 85], [Birget 89]

Composition law for runs



$$R_{uv} = R_u R_v$$

$$\ell r(uv) = \ell r(u)\big(\ell\ell(v)\, rr(u)\big)^{*}\ell r(v)$$

[Shepherdson 59], [Pécuchet 85], [Birget 89]

Equivalence result

Definition

A 2-way transducer is *aperiodic* iff its transition monoid is aperiodic, i.e. there exists an integer n such that $x^n = x^{n+1}$ for all elements x.
For every word u, the same runs occur on u^n and u^{n+1}.

Theorem [Carton, D.]

Given a word function f,

- f is defined by an **FO**$[<]$-transduction iff
- f is recognized by an aperiodic 2-way transducer.

Encoding into a **FO**[<]-transduction

Given an aperiodic 2-way transducer $T = (A, B, Q, \delta, \gamma, q_i, Q_f)$, we have to define :

- φ_{dom}
- C
- $lab_b^c(x)$
- $path^{c,c'}(x, y)$

Encoding into a **FO**$[<]$-transduction

Given an aperiodic 2-way transducer $T = (A, B, Q, \delta, \gamma, q_i, Q_f)$, we have to define :

- $\varphi_{dom} = \varphi_{L(T)} \wedge \varphi_{lin}$
- C
- $lab_b^c(x)$
- $path^{c,c'}(x, y)$

The domain formula is defined as the conjunction of :

- $\varphi_{L(T)}$, a formula recognizing the input domain of T,
- φ_{lin}, a formula satisfied if the input graph is linear.

Both are **FO**$[<]$-definable.

Encoding into a **FO**$[<]$-transduction

Given an aperiodic 2-way transducer $T = (A, B, Q, \delta, \gamma, q_i, Q_f)$, we have to define :

- $\varphi_{dom} = \varphi_{L(T)} \wedge \varphi_{lin}$
- $C = Q$
- $lab_b^c(x)$
- $path^{c,c'}(x, y)$

The set of copies is defined as the set of states of T.

Encoding into a **FO**[<]-transduction

Given an aperiodic 2-way transducer $T = (A, B, Q, \delta, \gamma, q_i, Q_f)$, we have to define :

- $\varphi_{dom} = \varphi_{L(T)} \wedge \varphi_{lin}$
- $C = Q$
- $lab_b^q(x) = \bigvee_a a(x)$ such that $\gamma(q, a) = b$
- $path^{q,q'}(x, y)$

The production of a position i over a state q is stored in the q copy.

Encoding into a FO[<]-transduction

Given an aperiodic 2-way transducer $T = (A, B, Q, \delta, \gamma, q_i, Q_f)$, we have to define :

- $\varphi_{dom} = \varphi_{L(T)} \wedge \varphi_{lin}$
- $C = Q$
- $lab_b^q(x) = \bigvee_a a(x)$ such that $\gamma(q, a) = b$
- $path^{q,q'}(x, y)$

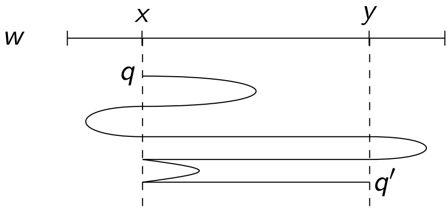
The formula $path^{q,q'}(x, y)$ has to be satisfied iff the run passes on position x in state q, then on position y in state q'.

Encoding into a **FO**[<]-transduction

Given an aperiodic 2-way transducer $T = (A, B, Q, \delta, \gamma, q_i, Q_f)$, we have to define :

- $\varphi_{dom} \; = \varphi_{L(T)} \wedge \varphi_{lin}$
- $C \; = Q$
- $lab_b^q(x) \; = \bigvee_a a(x)$ such that $\gamma(q, a) = b$
- $path^{q,q'}(x, y) \in$ **FO**[<]

This property is **FO**[<]-definable since it depends on sequences of partial runs over subwords the $w[1, x]$, $w[x, y]$ and $w[y, |u|]$, which depends on their images in the *aperiodic* monoid $M(T)$.



Coding **FO**[$<$]-transductions in aperiodic transducers

FO[$<$] as look-around tests

- The *path* predicate allows us to recover the *edge* predicate.
- The *edge* formulas are then encoded in (star-free) look-around tests, allowing to decide where to go.

Coding **FO**[$<$]-transductions in aperiodic transducers

FO[$<$] as look-around tests

- The *path* predicate allows us to recover the *edge* predicate.
- The *edge* formulas are then encoded in (star-free) look-around tests, allowing to decide where to go.

Removing the look-around

- Star-free look-around tests can be decided by aperiodic and functional 1-way automata,
- Transducers can preprocess the input to add the results of the tests to the letters,
- It only remains to prove that the deterministic and aperiodic transducers are closed under composition.

Composition of transducers

Theorem [Hopcroft-Ullman 67]

Let $\mathcal{A}$ be a 1-way transducer and $\mathcal{B}$ be a 2-way transducer, both deterministic.
Then we can effectively construct a deterministic 2-way transducer $\mathcal{D}$ such that $\mathcal{D} = \mathcal{B} \circ \mathcal{A}$.

Composition of transducers

Theorem [Hopcroft-Ullman 67]

Let $\mathcal{A}$ be a 1-way transducer and $\mathcal{B}$ be a 2-way transducer, both deterministic.
Then we can effectively construct a deterministic 2-way transducer $\mathcal{D}$ such that $\mathcal{D} = \mathcal{B} \circ \mathcal{A}$.

Main argument

- A two-way automaton can be enriched to keep track of the run of a given one-way automaton over the same input.

Composition of transducers

Theorem

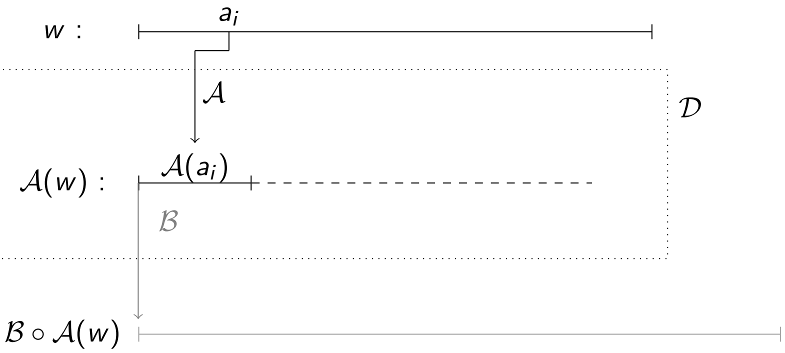
Let $\mathcal{A}$ be a 1-way transducer and $\mathcal{B}$ be a 2-way transducer, both deterministic and aperiodic.
Then we can effectively construct an aperiodic and deterministic 2-way transducer $\mathcal{D}$ such that $\mathcal{D} = \mathcal{B} \circ \mathcal{A}$.

Main argument

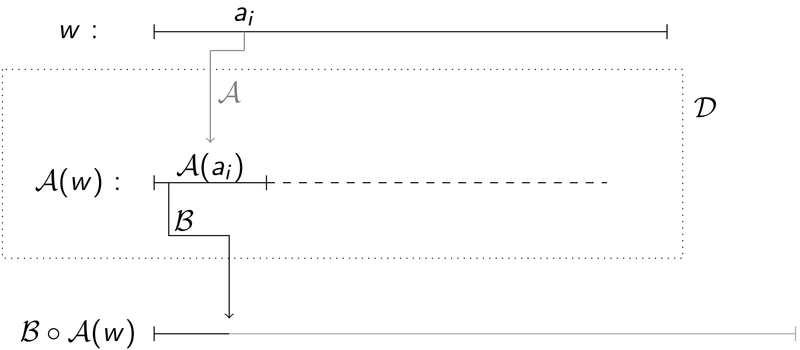
- A two-way automaton can be enriched to keep track of the run of a given one-way automaton over the same input.
- Aperiodicity is preserved by this procedure.

How the transducer $\mathcal{D}$ works

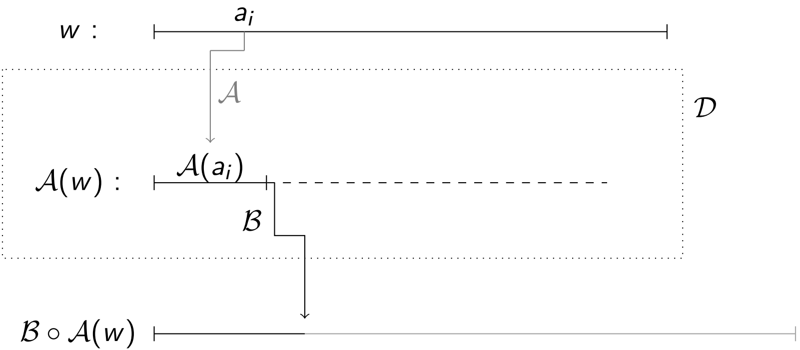
How the transducer $\mathcal{D}$ works



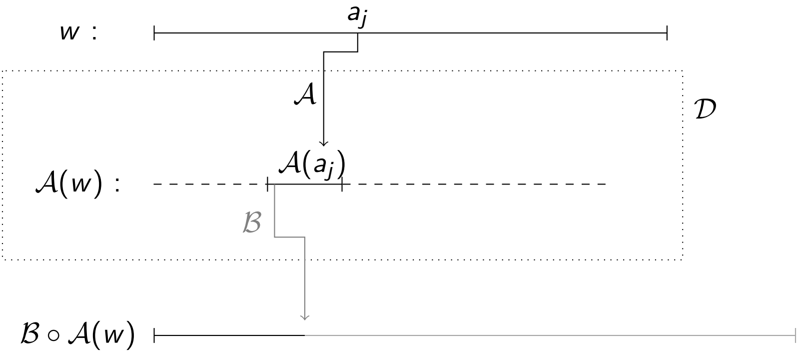
How the transducer $\mathcal{D}$ works



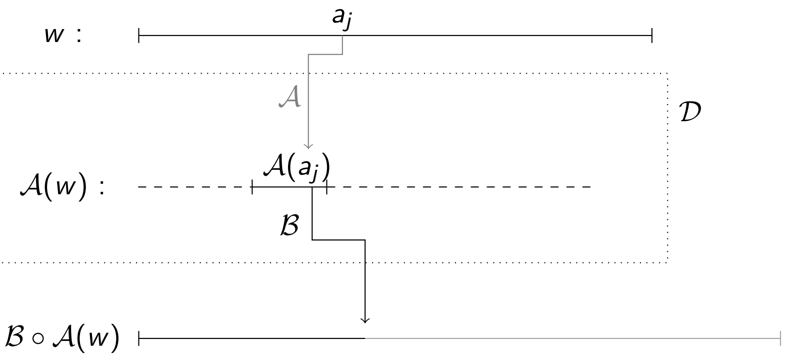
How the transducer $\mathcal{D}$ works



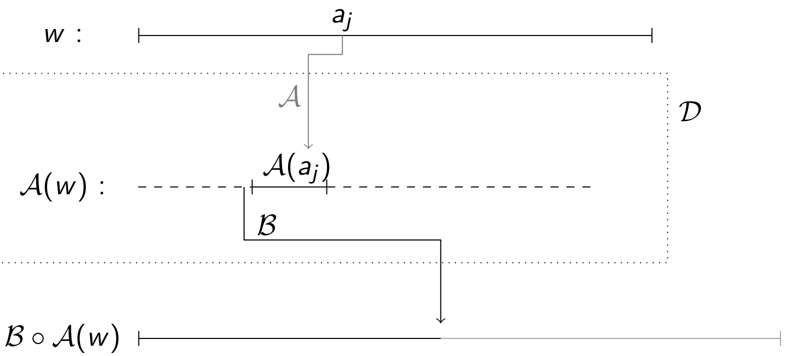
How the transducer $\mathcal{D}$ works



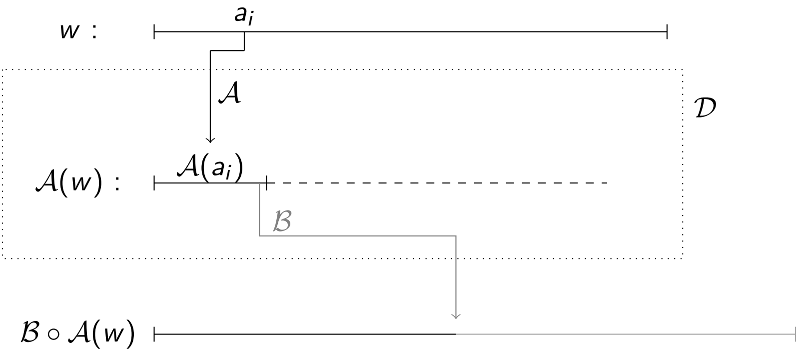
How the transducer $\mathcal{D}$ works



How the transducer $\mathcal{D}$ works



How the transducer $\mathcal{D}$ works

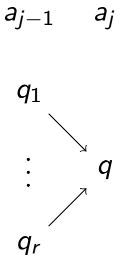


The Hopcroft-Ullman construction

When the two-way moves to the right, we keep track of the state of the one-way by simply following the transition function.

The Hopcroft-Ullman construction

When the two-way moves to the left, we have to rewind the one-way automata by one step.
Problem : *nondeterminism* arises.

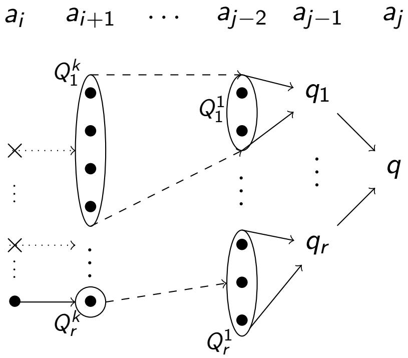


The Hopcroft-Ullman construction

When the two-way moves to the left, we have to rewind the one-way automata by one step.

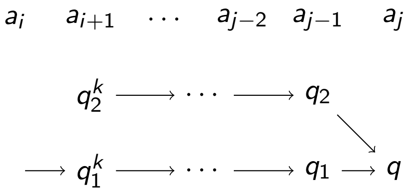
Problem : *nondeterminism* arises.

Solution : Rewind the run further to clear the nondeterminism,



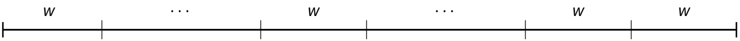
The Hopcroft-Ullman construction

When the two-way moves to the left, we have to rewind the one-way automata by one step.
Problem : *nondeterminism* arises.
Solution : Rewind the run further to clear the nondeterminism, then move back to the required position.



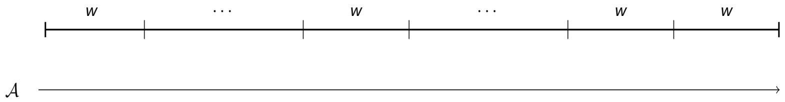
Aperiodicity of $\mathcal{D}$

$n = O(n_{\mathcal{A}} + n_{\mathcal{B}})$
$\ell r_{\mathcal{D}}(w^n) = \ell r_{\mathcal{D}}(w^{n+1})$



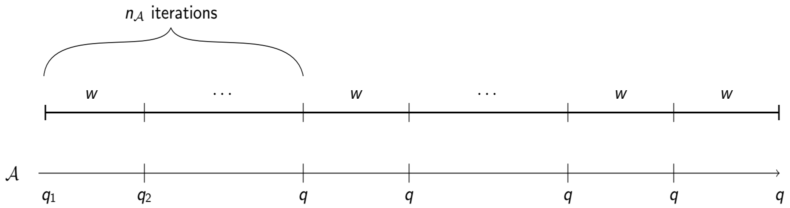
Aperiodicity of $\mathcal{D}$

$n = O(n_{\mathcal{A}} + n_{\mathcal{B}})$
$\ell r_{\mathcal{D}}(w^n) = \ell r_{\mathcal{D}}(w^{n+1})$



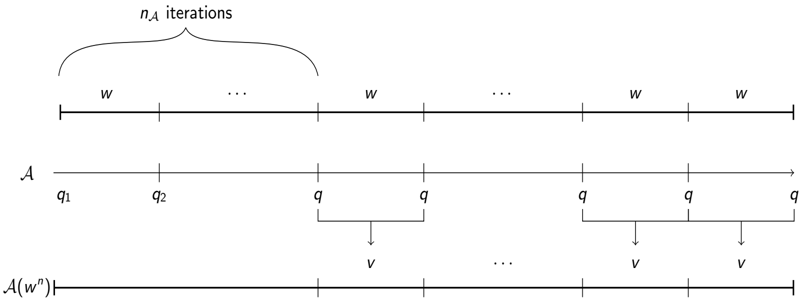
Aperiodicity of $\mathcal{D}$

$n = O(n_{\mathcal{A}} + n_{\mathcal{B}})$
$\ell r_{\mathcal{D}}(w^n) = \ell r_{\mathcal{D}}(w^{n+1})$



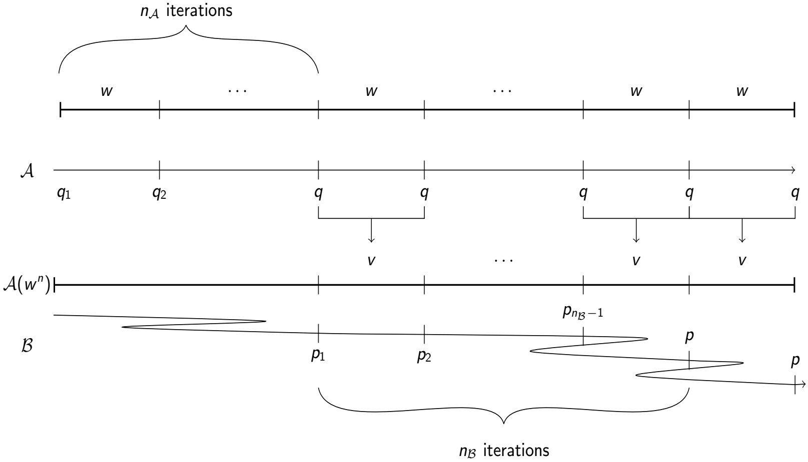
Aperiodicity of $\mathcal{D}$

$n = O(n_{\mathcal{A}} + n_{\mathcal{B}})$
$\ell r_{\mathcal{D}}(w^n) = \ell r_{\mathcal{D}}(w^{n+1})$



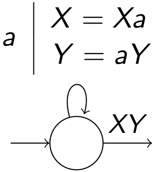
Aperiodicity of $\mathcal{D}$

$n = O(n_{\mathcal{A}} + n_{\mathcal{B}})$
$\ell r_{\mathcal{D}}(w^n) = \ell r_{\mathcal{D}}(w^{n+1})$



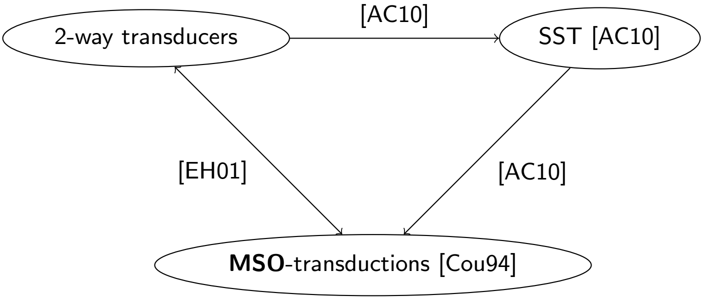
Streaming String Transducers

A *Streaming String Transducer* is a 1-way finite state transducer with variables used only for output.



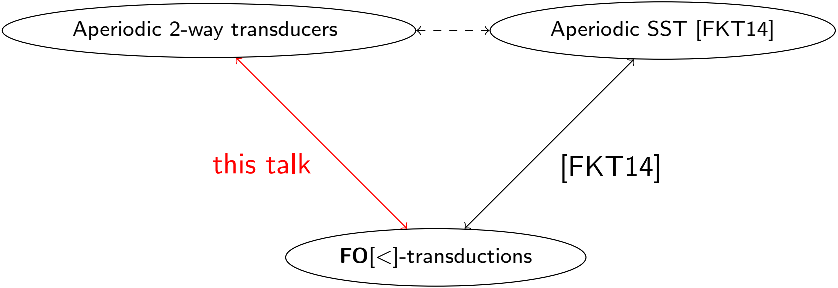
This SST realizes the function $w \to w\tilde{w}$.

Streaming String Transducers



[Courcelle 94], [Engelfriet, Hoogeboom 01], [Alur,Černý 10]

Streaming String Transducers



[Filiot, Krishna, Trivedi 14]

Conclusion and open questions

What we have

- Notion of a computable transition monoid for 2-way machines,
- Equivalence with a first-order logic,
- Aperiodicity of 2-way transducers corresponds to the notion of aperiodicity of SSTs.

Conclusion and open questions

What we have

- Notion of a computable transition monoid for 2-way machines,
- Equivalence with a first-order logic,
- Aperiodicity of 2-way transducers corresponds to the notion of aperiodicity of SSTs.

What's next ?

- A decidable and machine independent notion of aperiodicity.
- Notion of Star-free expression for functions (Regular expressions described in [Alur,Freilich,Raghothaman14]).
- Study of other fragments of logic ($\mathbf{FO}^2, \mathbf{FO}[Reg], (\mathbf{FO} + \mathbf{MOD})$).