The planar geometry of first-order string transductions

Nguyễn Lê Thành Dũng (a.k.a. Tito) — `nltd@nguyentito.eu`

Updated version of two old talks, both online
(Links seminar, Inria Lille, January 2021; Journées du GT ALGA, June 2021)

One possible motivation

Star-free languages are equivalently defined by:

- Star-free regexps: $E, E' ::= \varnothing \mid \{a\} \mid E \cup E' \mid E \cdot E' \mid E^{\mathrm{c}}$ (complement)
  e.g. $(ab)^* = (b\varnothing^{\mathrm{c}} \cup \varnothing^{\mathrm{c}}a \cup \varnothing^{\mathrm{c}}aa\varnothing^{\mathrm{c}} \cup \varnothing^{\mathrm{c}}bb\varnothing^{\mathrm{c}})^{\mathrm{c}}$ over the alphabet $\{a, b\}$
- $\varphi^{-1}(P)$ for some morphism φ to a finite and *aperiodic* monoid $M \supseteq P$
- counter-free automata (aperiodicity condition), first-order logic (FO), ...

Definition

A monoid M is *aperiodic* when $\forall x \in M,\ \exists n \in \mathbb{N} : x^n = x^{n+1}$.

One possible motivation

Star-free languages are equivalently defined by:

- Star-free regexps: $E, E' ::= \varnothing \mid \{a\} \mid E \cup E' \mid E \cdot E' \mid E^{\mathrm{c}}$ (complement)
  e.g. $(ab)^* = (b\varnothing^{\mathrm{c}} \cup \varnothing^{\mathrm{c}}a \cup \varnothing^{\mathrm{c}}aa\varnothing^{\mathrm{c}} \cup \varnothing^{\mathrm{c}}bb\varnothing^{\mathrm{c}})^{\mathrm{c}}$ over the alphabet $\{a, b\}$
- $\varphi^{-1}(P)$ for some morphism φ to a finite and *aperiodic* monoid $M \supseteq P$
- counter-free automata (aperiodicity condition), first-order logic (FO), ...

Definition

A monoid M is *aperiodic* when $\forall x \in M,\ \exists n \in \mathbb{N} : x^n = x^{n+1}$.

Lack of *compositionality*: $\{x \in M \mid \exists n \in \mathbb{N} : x^n = x^{n+1}\}$ not a submonoid
$\implies$ lack of *locality*: aperiodicity cannot be checked just on the generators

One possible motivation

Star-free languages are equivalently defined by:

- Star-free regexps: $E, E' ::= \varnothing \mid \{a\} \mid E \cup E' \mid E \cdot E' \mid E^{\mathrm{c}}$ (complement)
  e.g. $(ab)^* = (b\varnothing^{\mathrm{c}} \cup \varnothing^{\mathrm{c}}a \cup \varnothing^{\mathrm{c}}aa\varnothing^{\mathrm{c}} \cup \varnothing^{\mathrm{c}}bb\varnothing^{\mathrm{c}})^{\mathrm{c}}$ over the alphabet $\{a, b\}$
- $\varphi^{-1}(P)$ for some morphism φ to a finite and *aperiodic* monoid $M \supseteq P$
- counter-free automata (aperiodicity condition), first-order logic (FO), ...

Definition

A monoid M is *aperiodic* when $\forall x \in M,\ \exists n \in \mathbb{N} : x^n = x^{n+1}$.

Lack of *compositionality*: $\{x \in M \mid \exists n \in \mathbb{N} : x^n = x^{n+1}\}$ not a submonoid
$\Longrightarrow$ lack of *locality*: aperiodicity cannot be checked just on the generators

We characterize star-free languages (and FO transductions)
by "compositional/local" conditions on *behaviors of two-way automata*

Drawback: no counterpart to syntactic monoid / minimal DFA

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$$

Directed states are an old idea[1], more convenient
+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

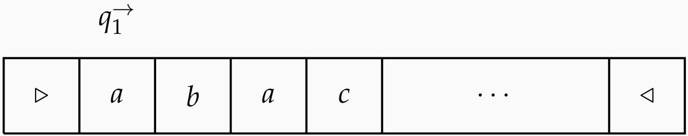
[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Reminder: two-way automata (1)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$$



Directed states are an old idea[1], more convenient
+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

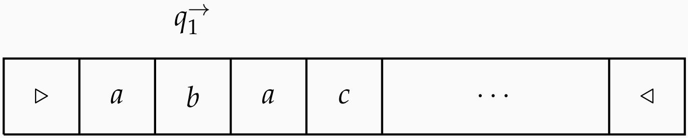
[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Reminder: two-way automata (1)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow}$ $\qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow}$ $\qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow}$ $\qquad q_3^{\leftarrow}, b \mapsto$ accept



Directed states are an old idea[1], more convenient
+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow}$ $\quad$ $q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow}$ $\quad$ $q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow}$ $\quad$ $q_3^{\leftarrow}, b \mapsto$ accept

$q_1^{\rightarrow}$

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

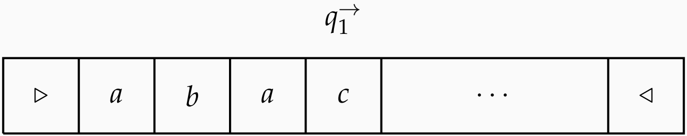
Directed states are an old idea[1], more convenient
+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow}$ $\quad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow}$ $\quad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow}$ $\quad q_3^{\leftarrow}, b \mapsto \text{accept}$



Directed states are an old idea[1], more convenient

+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

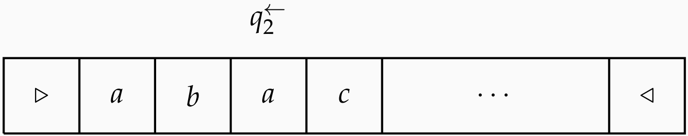
[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Reminder: two-way automata (1)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow}$ $q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow}$ $q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow}$ $q_3^{\leftarrow}, b \mapsto$ accept



Directed states are an old idea[1], more convenient

+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

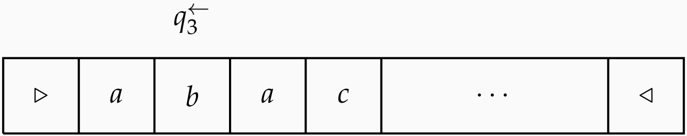
[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Reminder: two-way automata (1)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow}$ $\quad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow}$ $\quad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow}$ $\quad q_3^{\leftarrow}, b \mapsto$ accept



Directed states are an old idea[1], more convenient

+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

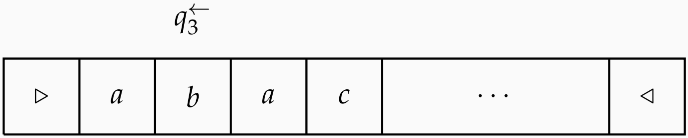
[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Reminder: two-way automata (1)

Transitions: update finite state + move left/right depending on new state

Example: states $Q = \{q_1^{\rightarrow}, q_2^{\leftarrow}, q_3^{\leftarrow}\}$, initial state $q_1^{\rightarrow}$

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow}$ $\quad$ $q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow}$ $\quad$ $q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow}$ $\quad$ $q_3^{\leftarrow}, b \mapsto \text{accept}$



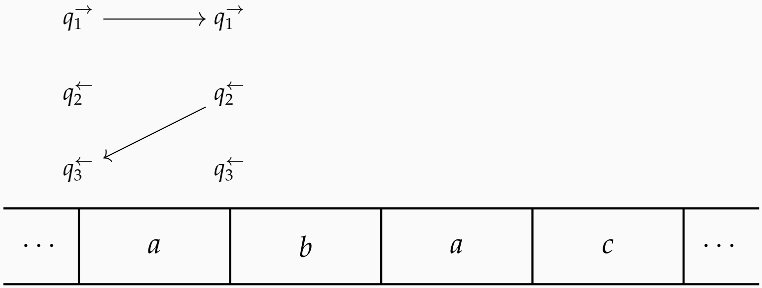
Directed states are an old idea[1], more convenient
+ needed to define *reversible* 2DFAs (Dartois et al. ICALP'17)

[1] cf. e.g. J.-C. Birget, *Concatenation of Inputs in a Two-Way Automaton* (1989)

Reminder: two-way automata (2)

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$

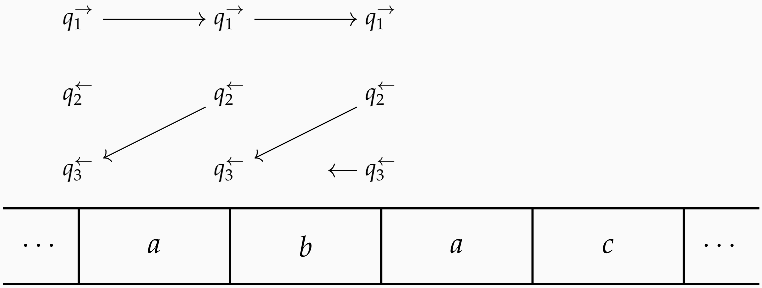
Graphical representation of transitions for each letter:



Reminder: two-way automata (2)

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$

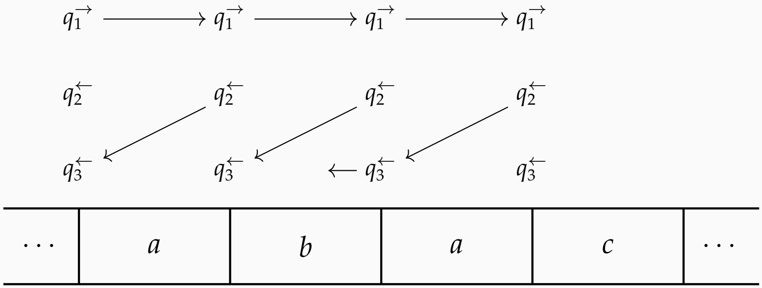
Graphical representation of transitions for each letter:



Reminder: two-way automata (2)

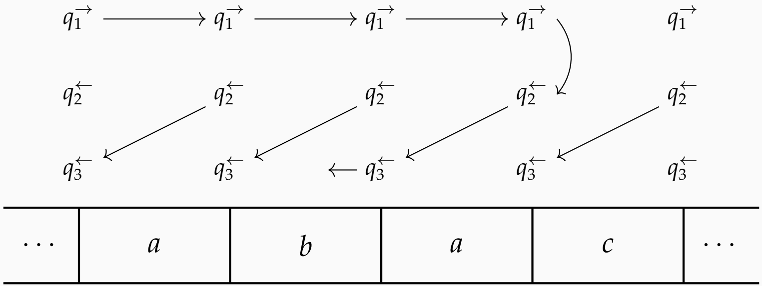
$$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$$

Graphical representation of transitions for each letter:



$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$

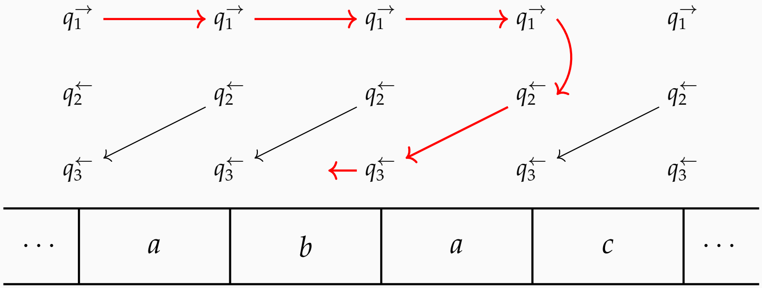
Graphical representation of transitions for each letter:



Reminder: two-way automata (2)

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$

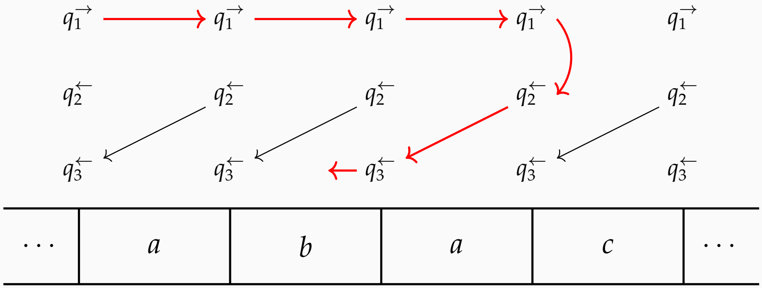
Graphical representation of transitions for each letter:



Reminder: two-way automata (2)

$q_1^{\rightarrow}, (a|b) \mapsto q_1^{\rightarrow} \qquad q_1^{\rightarrow}, c \mapsto q_2^{\leftarrow} \qquad q_2^{\leftarrow}, (a|b|c) \mapsto q_3^{\leftarrow} \qquad q_3^{\leftarrow}, b \mapsto \text{accept}$

Graphical representation of transitions for each letter:

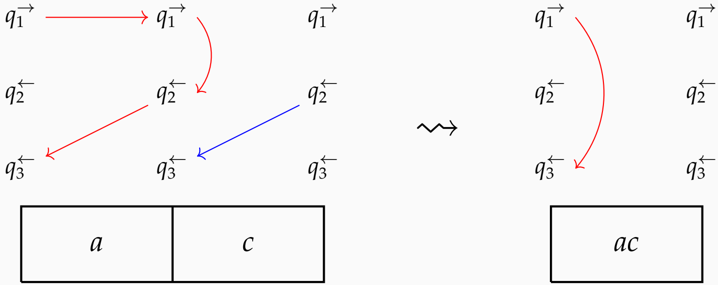


This two-way automaton is *deterministic*: outdegree ≤ 1

reversible: deterministic + indegree ≤ 1

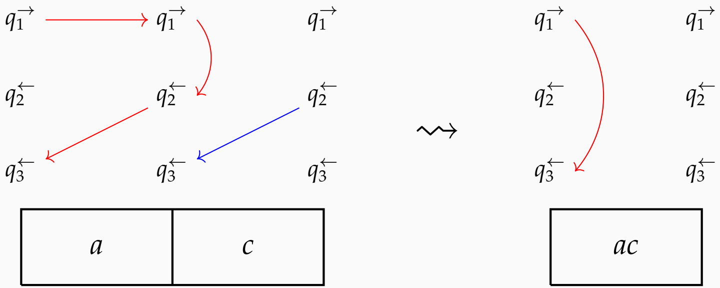
Reminder: two-way automata (3)

Behaviors (or crossing types) form a monoid:



Reminder: two-way automata (3)

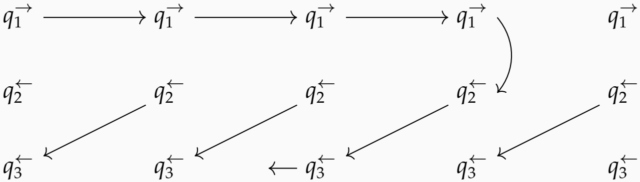
Behaviors (or crossing types) form a monoid:



This monoid is finite, therefore 2DFA recognize regular languages
(modern account of Shepherdson's construction (1959))

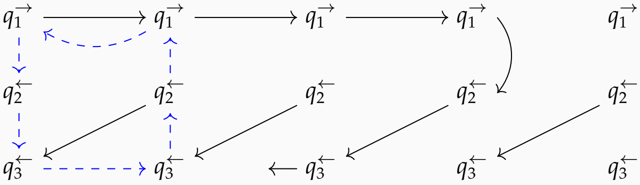
Reversible behaviors are closed under product, and
reversible 2DFA can recognize all regular languages (Dartois et al. ICALP'17)

Combinatorial planarity



This drawing is *planar*, i.e. without crossed edges.

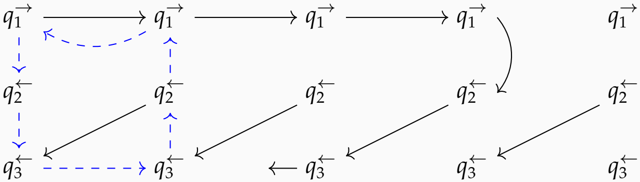
Combinatorial planarity



This drawing is *planar*, i.e. without crossed edges.
Formally: for each of these 4 behaviors, the cyclic order

$$q_1^{\text{left}} \prec q_2^{\text{left}} \prec q_3^{\text{left}} \prec q_3^{\text{right}} \prec q_2^{\text{right}} \prec q_1^{\text{right}} \prec q_1^{\text{left}}$$

Combinatorial planarity



This drawing is *planar*, i.e. without crossed edges.
Formally: for each of these 4 behaviors, the cyclic order

$$q_1^{\text{left}} \prec q_2^{\text{left}} \prec q_3^{\text{left}} \prec q_3^{\text{right}} \prec q_2^{\text{right}} \prec q_1^{\text{right}} \prec q_1^{\text{left}}$$

does not contain any sub-cyclic-order $x \prec y \prec z \prec w \prec x$ such that

- x and z are connected by an edge (either $x \to z$ or $z \to x$)
- and y and w are also connected by an edge

$\longrightarrow$ depends on the choice of total order $q_1 < q_2 < q_3$

(More like planar combinatorial maps than planar graphs...)

The main theorem

Theorem

Let $L \subseteq \Sigma^*$. The following are equivalent:

- *L is a star-free language.*
- *L is recognized by some* planar *2DFA.*
- *L is recognized by some* planar reversible *2DFA.*

Our example of planar 2DFA recognizes $(\varnothing^c c \varnothing^c)^c b (a \cup b) c \varnothing^c$

The main theorem

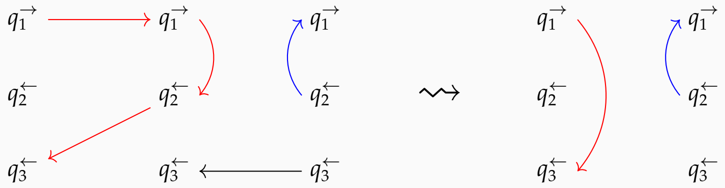
Theorem

Let $L \subseteq \Sigma^*$. The following are equivalent:

- *L is a star-free language.*
- *L is recognized by some* planar *2DFA.*
- *L is recognized by some* planar reversible *2DFA.*

Our example of planar 2DFA recognizes $(\varnothing^c c \varnothing^c)^c b(a \cup b)c\varnothing^c$

Compositionality/locality: planar behaviors are closed under product



Planar *one-way* automata = *monotone* transitions, not powerful enough

A stronger theorem on transducers

Two-way deterministic *transducers* (2DFT) = 2DFA with output
(each transition can append a suffix to the output log)

2DFTs compute *regular functions* a.k.a. *MSO transductions*,
a well-behaved class of functions with many different characterizations
example: $w \mapsto w \cdot \texttt{reverse}(w)$

Ask for the monoid of behaviors to be *aperiodic*,
and you get *first-order transductions* (Carton & Dartois, CSL'15)

A stronger theorem on transducers

Two-way deterministic *transducers* (2DFT) = 2DFA with output
(each transition can append a suffix to the output log)

2DFTs compute *regular functions* a.k.a. *MSO transductions*,
a well-behaved class of functions with many different characterizations
example: $w \mapsto w \cdot \texttt{reverse}(w)$

Ask for the monoid of behaviors to be *aperiodic*,
and you get *first-order transductions* (Carton & Dartois, CSL'15)

Theorem

Let $f : \Gamma^* \rightharpoonup \Sigma^*$. *The following are equivalent:*

- f *is a first-order transduction.*
- f *is computed by some* planar *2DFT.*
- f *is computed by some* planar reversible *2DFT.*

Next: proofs!

Aperiodicity of planar behaviors (1)

To show that planar 2DFA can recognize *only* star-free languages, we use:

Lemma

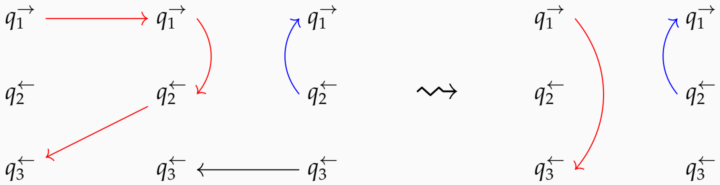
Let Q be a finite set of directed states. The finite monoid $\mathfrak{P}_Q$ of all possible planar behaviors over Q is aperiodic*:* $\forall x \in \mathfrak{P}_Q,\ \exists n \in \mathbb{N} : x^n = x^{n+1}$.

Aperiodicity of planar behaviors (1)

To show that planar 2DFA can recognize *only* star-free languages, we use:

Lemma

Let Q be a finite set of directed states. The finite monoid $\mathfrak{P}_Q$ of all possible planar behaviors over Q is aperiodic*:* $\forall x \in \mathfrak{P}_Q,\ \exists n \in \mathbb{N} : x^n = x^{n+1}$.



The blue edge is conserved: right-right edges of $y \subseteq$ right-right edges of xy

(right-right edges of $x^n)_{n \in \mathbb{N}}$ monotone, hence eventually constant

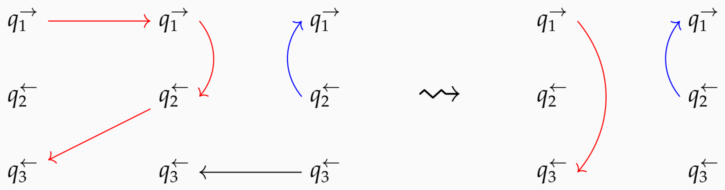
$x \leq xy$ for left-left edges, cf. new red edge.

Aperiodicity of planar behaviors (1)

To show that planar 2DFA can recognize *only* star-free languages, we use:

Lemma

Let Q be a finite set of directed states. The finite monoid $\mathfrak{P}_Q$ of all possible planar behaviors over Q is aperiodic*:* $\forall x \in \mathfrak{P}_Q,\ \exists n \in \mathbb{N} : x^n = x^{n+1}$.



The blue edge is conserved: right-right edges of $y \subseteq$ right-right edges of xy

(right-right edges of $x^n)_{n\in\mathbb{N}}$ monotone, hence eventually constant

$x \leq xy$ for left-left edges, cf. new red edge. What about left-right edges?

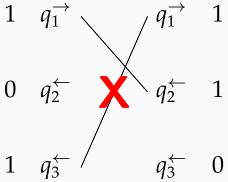
Aperiodicity of planar behaviors (2)

Left-right edges are entirely determined by *degrees*:

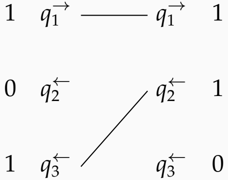
1	$\overrightarrow{q_1}$	$\overrightarrow{q_1}$	1
0	$\overleftarrow{q_2}$	$\overleftarrow{q_2}$	1
1	$\overleftarrow{q_3}$	$\overleftarrow{q_3}$	0

Aperiodicity of planar behaviors (2)

Left-right edges are entirely determined by *degrees*:

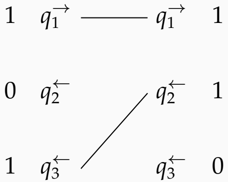


Left-right edges are entirely determined by *degrees*:



Aperiodicity of planar behaviors (2)

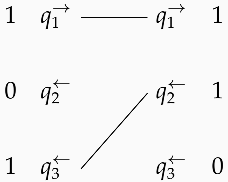
Left-right edges are entirely determined by *degrees*:



degrees of right half of $y \geq$ degrees of right half of xy
$\longrightarrow$ (degrees of $x^n)_{n \in \mathbb{N}}$ non-increasing, hence eventually constant
Combine with previous slide: $\exists n \in \mathbb{N} : x^n = x^{n+1}$!

Aperiodicity of planar behaviors (2)

Left-right edges are entirely determined by *degrees*:



degrees of right half of $y \geq$ degrees of right half of xy

$\longrightarrow$ (degrees of $x^n)_{n\in\mathbb{N}}$ non-increasing, hence eventually constant

Combine with previous slide: $\exists n \in \mathbb{N} : x^n = x^{n+1}$!

More conceptual POV: Green's relations on the monoid $\mathfrak{P}_Q$

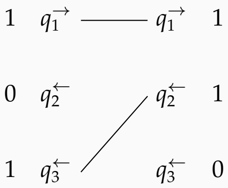
behavior $\nearrow$ for $\preccurlyeq_{\mathcal{L}} \implies$ right degrees $\nearrow$ and right-right edges $\searrow$

$\longrightarrow$ $\mathcal{L}$-class invariant + $\mathcal{R}$-class invariant determine element of $\mathfrak{P}_Q$

$\longrightarrow$ $\mathfrak{P}_Q$ is $\mathcal{H}$-trivial i.e. aperiodic (an idea by Cécilia Pradic)

Aperiodicity of planar behaviors (2)

Left-right edges are entirely determined by *degrees*:



degrees of right half of $y \geq$ degrees of right half of xy
$\longrightarrow$ (degrees of $x^n)_{n \in \mathbb{N}}$ non-increasing, hence eventually constant
Combine with previous slide: $\exists n \in \mathbb{N} : x^n = x^{n+1}$!

More conceptual POV: Green's relations on the monoid $\mathfrak{P}_Q$

behavior $\nearrow$ for $\preccurlyeq_{\mathcal{L}} \implies$ right degrees $\nearrow$ and right-right edges $\searrow$
$\longrightarrow$ $\mathcal{L}$-class invariant + $\mathcal{R}$-class invariant determine element of $\mathfrak{P}_Q$
$\longrightarrow$ $\mathfrak{P}_Q$ is $\mathcal{H}$-trivial i.e. aperiodic (an idea by Cécilia Pradic)

Next: the converse direction of the main theorem

Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

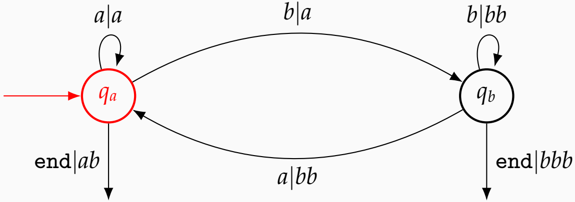
Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)

Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)
Sequential transducers (see below) with aperiodic transition monoids



$abba \mapsto$

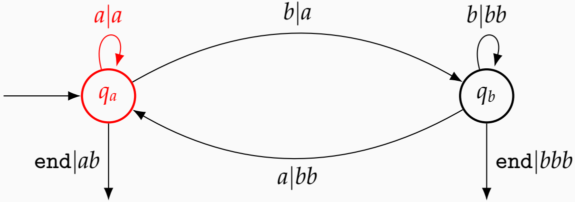
Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)

Sequential transducers (see below) with aperiodic transition monoids



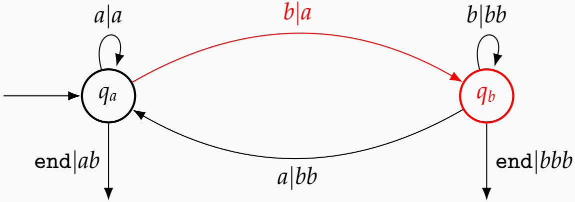
$abba \mapsto a$

Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)
Sequential transducers (see below) with aperiodic transition monoids



$abba \mapsto aa$

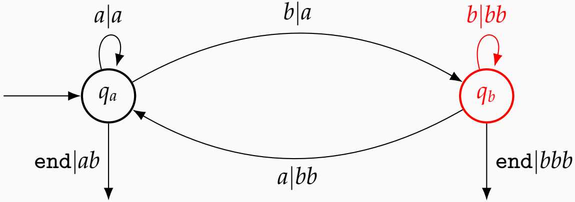
Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)

Sequential transducers (see below) with aperiodic transition monoids



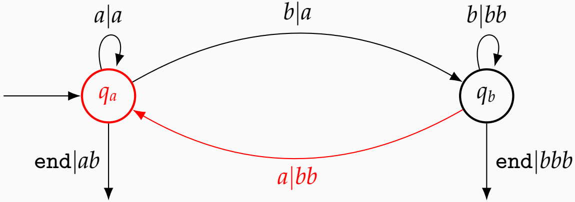
$abba \mapsto aabb$

Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)
Sequential transducers (see below) with aperiodic transition monoids



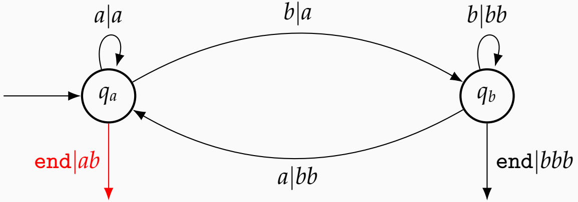
$abba \mapsto aabbbb$

Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)
Sequential transducers (see below) with aperiodic transition monoids



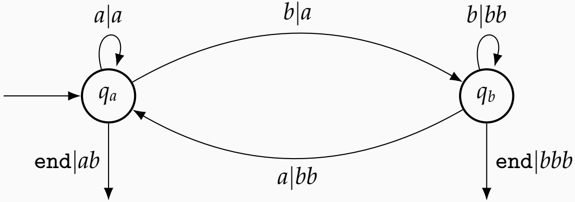
$abba \mapsto aabbbbab$

Expressiveness of reversible planar 2DFTs (1)

Theorem (Part of the main theorem on transducers)

Any first-order transduction can be computed by a reversible planar 2DFT.

Let's start with *aperiodic sequential functions* ($\subsetneq$ FO transductions)
Sequential transducers (see below) with aperiodic transition monoids



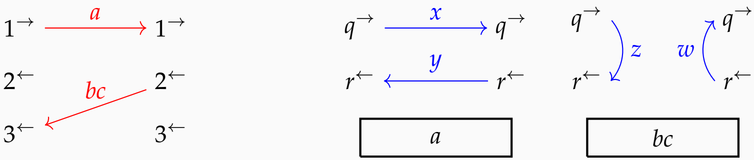
$$abba \mapsto aabbbbab$$

Reminder: Krohn–Rhodes decomposition theorem

Aperiodic sequential functions are generated by aper. seq. transducers *with 2 states* (like the one above) + *function composition*.

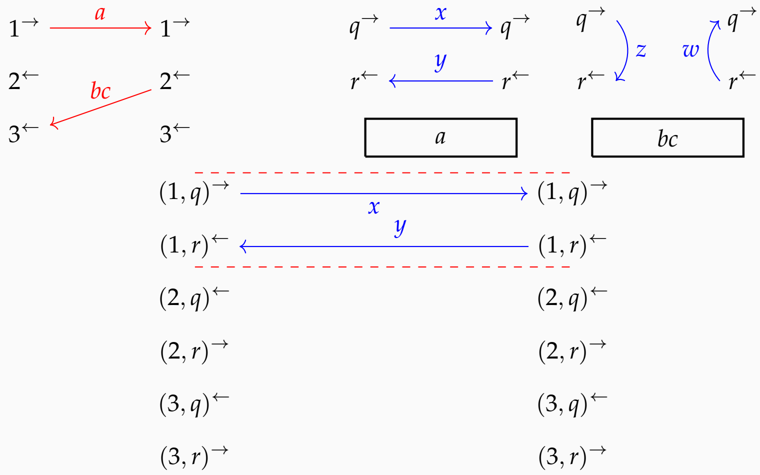
Expressiveness of reversible planar 2DFTs (2): composition

Composition of reversible 2DFTs uses a wreath-product-like construction (do you see why reversibility is needed?) *preserving planarity*



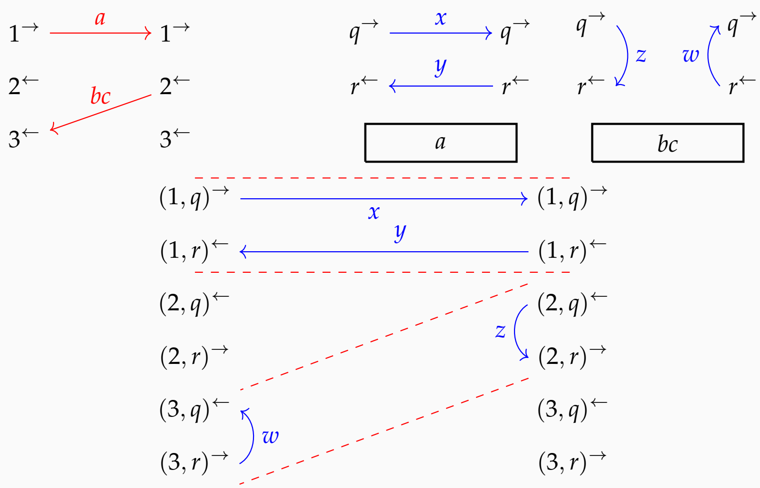
Expressiveness of reversible planar 2DFTs (2): composition

Composition of reversible 2DFTs uses a wreath-product-like construction (do you see why reversibility is needed?) *preserving planarity*

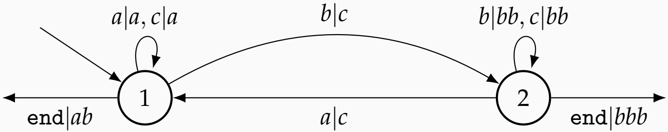


Expressiveness of reversible planar 2DFTs (2): composition

Composition of reversible 2DFTs uses a wreath-product-like construction (do you see why reversibility is needed?) *preserving planarity*

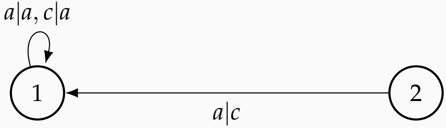


The Krohn–Rhodes decomposition involves aperiodic sequential transducers *with 2 states*, such as:



Expressiveness of reversible planar 2DFTs (3): flip-flops

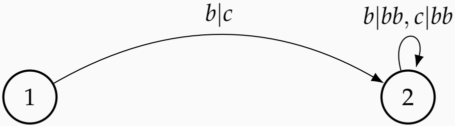
The Krohn–Rhodes decomposition involves aperiodic sequential transducers *with 2 states*, such as:



$\delta_a : \_ \mapsto 1$

Expressiveness of reversible planar 2DFTs (3): flip-flops

The Krohn–Rhodes decomposition involves aperiodic sequential transducers *with 2 states*, such as:



$\delta_a : \_ \mapsto 1 \qquad \delta_b : \_ \mapsto 2$

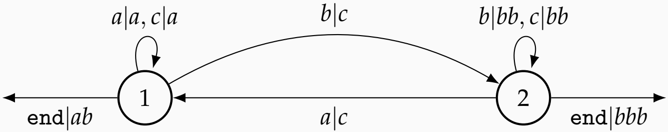
Expressiveness of reversible planar 2DFTs (3): flip-flops

The Krohn–Rhodes decomposition involves aperiodic sequential transducers *with 2 states*, such as:

$\delta_a : \_ \mapsto 1$ $\delta_b : \_ \mapsto 2$ $\delta_c = \mathrm{id}$

Expressiveness of reversible planar 2DFTs (3): flip-flops

The Krohn–Rhodes decomposition involves aperiodic sequential transducers *with 2 states*, such as:

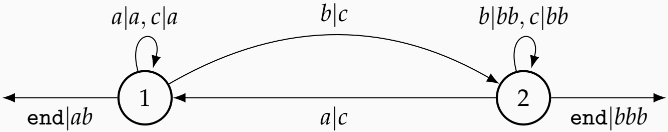


$\delta_a : \_ \mapsto 1 \qquad \delta_b : \_ \mapsto 2 \qquad \delta_c = \mathrm{id}$

For $Q = \{1, 2\}$, aperiodicity is equivalent to excluding $q \mapsto 3 - q$
$\{\delta_a, \delta_b, \delta_c\}$ is the largest aperiodic submonoid of $Q \to Q$

Expressiveness of reversible planar 2DFTs (3): flip-flops

The Krohn–Rhodes decomposition involves aperiodic sequential transducers *with 2 states*, such as:



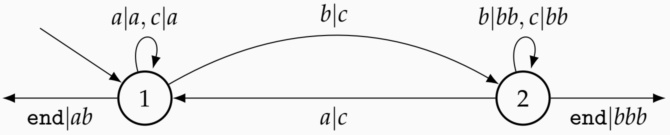
$\delta_a : \_ \mapsto 1$ $\delta_b : \_ \mapsto 2$ $\delta_c = \mathrm{id}$

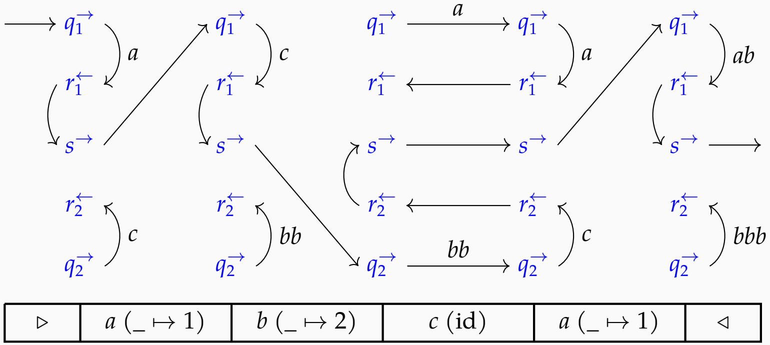
For $Q = \{1, 2\}$, aperiodicity is equivalent to excluding $q \mapsto 3 - q$

$\{\delta_a, \delta_b, \delta_c\}$ is the largest aperiodic submonoid of $Q \to Q$

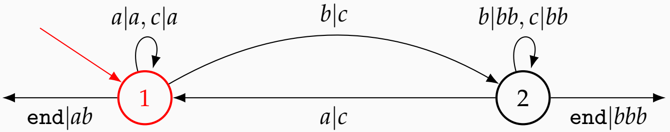
Let's translate this into a reversible planar 2DFT! (next slide)

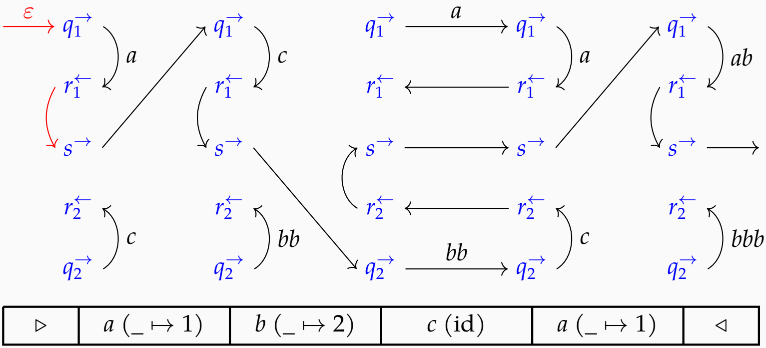
Expressiveness of reversible planar 2DFTs (4): encoding flip-flops



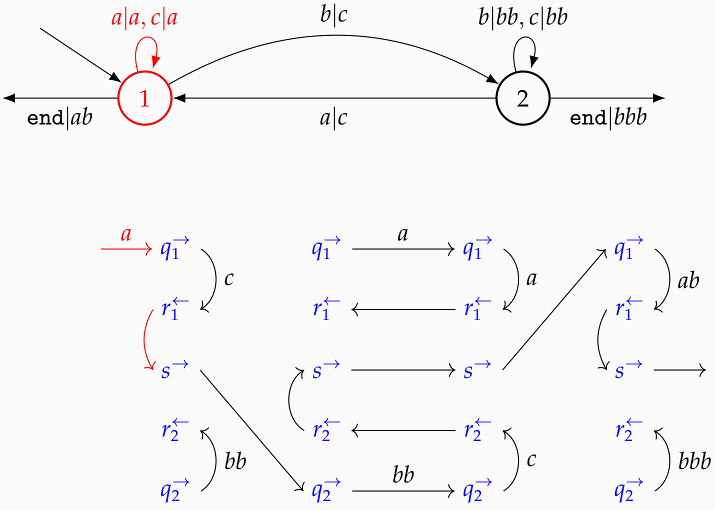


Expressiveness of reversible planar 2DFTs (4): encoding flip-flops



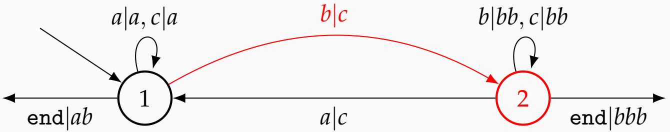


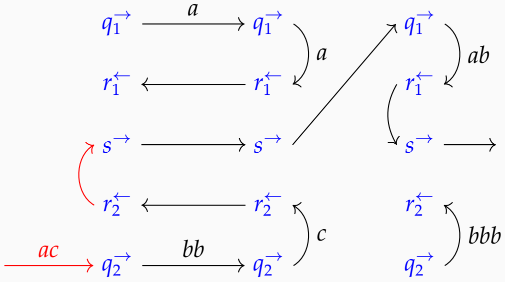
Expressiveness of reversible planar 2DFTs (4): encoding flip-flops



▷	a ($\_ \mapsto 1$)	b ($\_ \mapsto 2$)	c (id)	a ($\_ \mapsto 1$)	◁

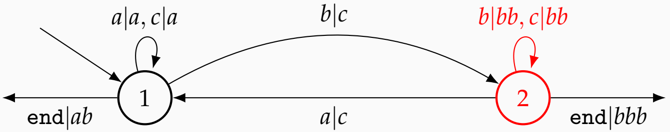
Expressiveness of reversible planar 2DFTs (4): encoding flip-flops

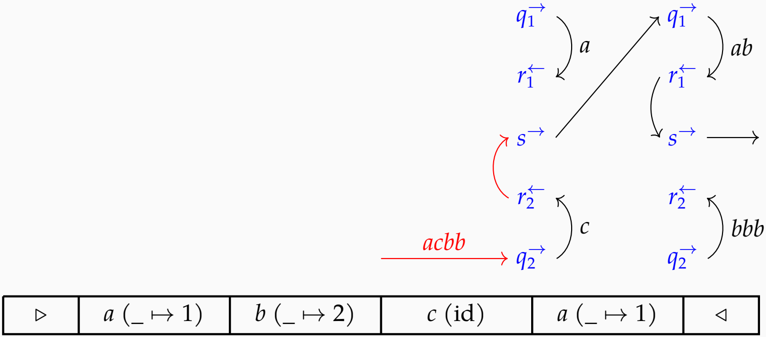




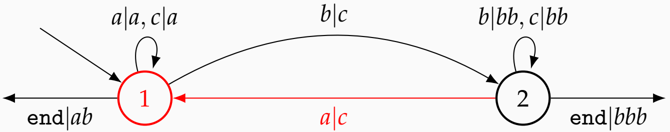
▷	a ($\_ \mapsto 1$)	b ($\_ \mapsto 2$)	c (id)	a ($\_ \mapsto 1$)	◁

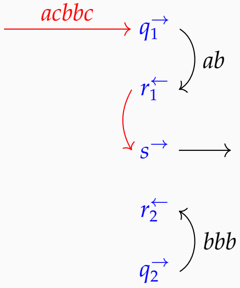
Expressiveness of reversible planar 2DFTs (4): encoding flip-flops





Expressiveness of reversible planar 2DFTs (4): encoding flip-flops





▷	a (_ ↦ 1)	b (_ ↦ 2)	c (id)	a (_ ↦ 1)	◁

Expressiveness of reversible planar 2DFTs (5): FO transductions

We just proved that reversible planar 2DFTs are *closed under composition* and can simulate *two-state aperiodic sequential transducers*.

By Krohn–Rhodes, we get all aper. seq. functions.

Expressiveness of reversible planar 2DFTs (5): FO transductions

We just proved that reversible planar 2DFTs are *closed under composition* and can simulate *two-state aperiodic sequential transducers*.

By Krohn–Rhodes, we get all aper. seq. functions. To go further, we use:

Theorem (Bojańczyk et al.[2])

Any first-order transduction *can be obtained as a composition of:*

- *aperiodic sequential functions;*
- $\texttt{mapReverse}_\Sigma, \texttt{mapDuplicate}_\Sigma : (\Sigma \cup \{\#\})^* \to (\Sigma \cup \{\#\})^*$ *for* $\# \notin \Sigma$

For $w_1, \dots, w_n \in \Sigma^*$, $\texttt{mapReverse}_\Sigma(w_1\# \dots \#w_n) = \texttt{rev}(w_1)\# \cdots \#\texttt{rev}(w_n)$

$$\texttt{mapDuplicate}_\Sigma(w_1\# \dots \#w_n) = w_1w_1\# \cdots \#w_nw_n$$

With this, we can conclude the proof of the main theorems.

[2]Not entirely explicit in the literature; variants can be found in:
Bojańczyk, Daviaud & Krishna, *Regular and first-order list functions* (2018)
Bojańczyk & Stefański, *Single-use automata and transducers for infinite alphabets* (2020)

Important related work

Our main inspirations: λ-calculus and category theory

- Hines, *A categorical framework for finite state machines* (2003)
  Relates monoid of 2DFA behaviors to *geometry of interaction* (GoI),
  a family of semantics for linear λ-calculi (as in *linear logic*)[3]
- N. & Pradic, *Implicit automata in typed λ-calculi I: Aperiodicity in a non-commutative logic* (ICALP'20)
  Characterizes star-free languages using a linear non-comm. λ-calculus

[3] See also: T. Seiller, *Interaction graphs: non-deterministic automata* (2018)

Important related work

Our main inspirations: λ-calculus and category theory

- Hines, *A categorical framework for finite state machines* (2003)
  Relates monoid of 2DFA behaviors to *geometry of interaction* (GoI),
  a family of semantics for linear λ-calculi (as in *linear logic*)[3]
- N. & Pradic, *Implicit automata in typed λ-calculi I: Aperiodicity in a non-commutative logic* (ICALP'20)
  Characterizes star-free languages using a linear non-comm. λ-calculus

Connection: non-commutativity in λ-calculi $\iff$ planarity in GoI

Reversible planar DFA were considered by Hines in a talk
(*Temperley-Lieb algebras as two-way automata*, QNET[4] Workshop 2006)
but he did not characterize their expressive power

[3]See also: T. Seiller, *Interaction graphs: non-deterministic automata* (2018)
[4]UK Network on Semantics of Quantum Computation

Important related work

Our main inspirations: λ-calculus and category theory

- Hines, *A categorical framework for finite state machines* (2003)
  Relates monoid of 2DFA behaviors to *geometry of interaction* (GoI),
  a family of semantics for linear λ-calculi (as in *linear logic*)[3]
- N. & Pradic, *Implicit automata in typed λ-calculi I: Aperiodicity in a non-commutative logic* (ICALP'20)
  Characterizes star-free languages using a linear non-comm. λ-calculus

Connection: non-commutativity in λ-calculi $\iff$ planarity in GoI

Reversible planar DFA were considered by Hines in a talk
(*Temperley-Lieb algebras as two-way automata*, QNET[4] Workshop 2006)
but he did not characterize their expressive power

(Papers on GoI are often named "The geometry of X" $\longrightarrow$ this talk's title)

[3] See also: T. Seiller, *Interaction graphs: non-deterministic automata* (2018)
[4] UK Network on Semantics of Quantum Computation

Conclusion

We introduced a notion of *planarity* of two-way transducers,
based on the graphical representation of their behavior, and showed:

Main theorem

star-free language	$\Longleftrightarrow$	planar 2DFA	$\Longleftrightarrow$	planar reversible 2DFA
FO transduction	$\Longleftrightarrow$	planar 2DFT	$\Longleftrightarrow$	planar reversible 2DFT

- Planar behaviors form an aperiodic submonoid of all behaviors
  $\longrightarrow$ unlike aperiodicity, planarity is compositional
- Expressivity established via factorization theorems
  (Krohn–Rhodes + extension to FO transductions)
- Inspiration from other areas of "logic in computer science"

Conclusion

We introduced a notion of *planarity* of two-way transducers,
based on the graphical representation of their behavior, and showed:

Main theorem

star-free language	$\iff$	planar 2DFA	$\iff$	planar reversible 2DFA
FO transduction	$\iff$	planar 2DFT	$\iff$	planar reversible 2DFT

- Planar behaviors form an aperiodic submonoid of all behaviors
  $\longrightarrow$ unlike aperiodicity, planarity is compositional
- Expressivity established via factorization theorems
  (Krohn–Rhodes + extension to FO transductions)
- Inspiration from other areas of "logic in computer science"

Thanks for your attention! Any questions?