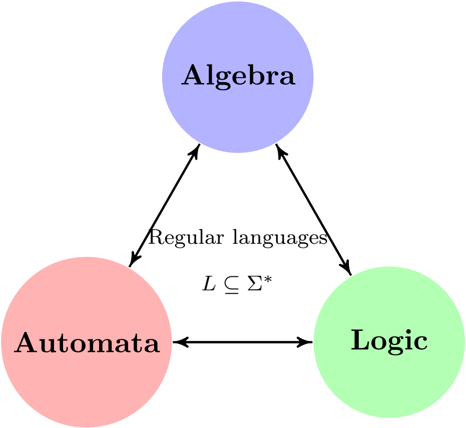
Automata, Logic and Algebra for (Finite) Word Transductions

Emmanuel Filiot

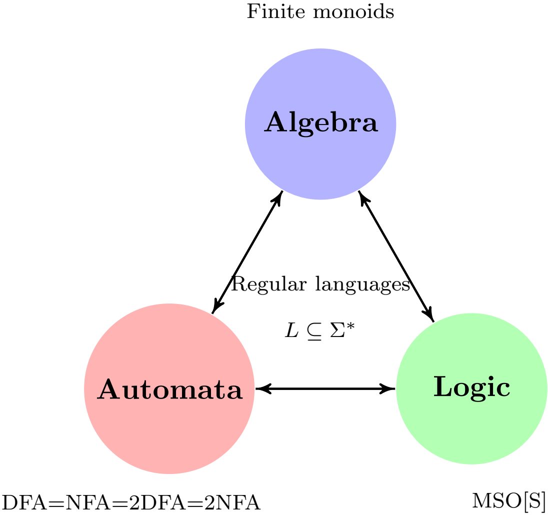
Université libre de Bruxelles & FNRS

ACTS 2017, Chennai

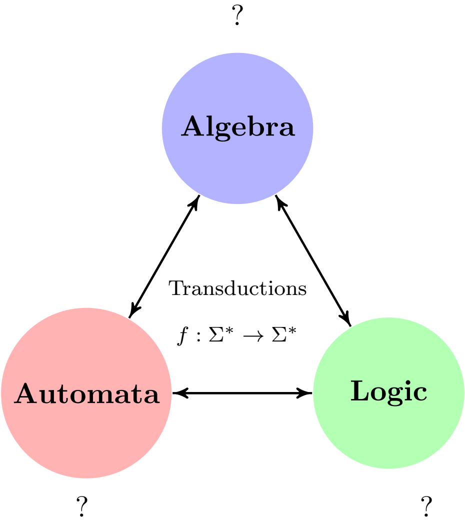
Trinity for Regular Languages



Trinity for Regular Languages



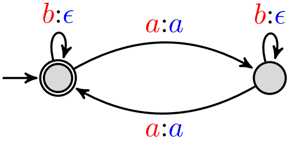
Objective of the talk



Automata models for transductions

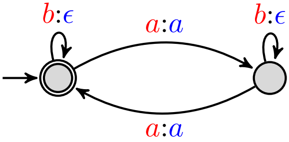
Automata for transductions: transducers

f_{del} :



Automata for transductions: transducers

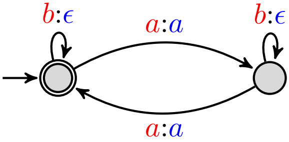
f_{del} :



$aabaa \mapsto aaaa$

Automata for transductions: transducers

f_{del} :

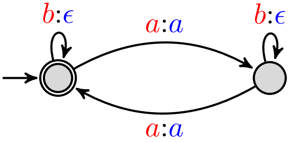


$$aabaa \mapsto aaaa$$

$$aaba \mapsto \text{undefined}$$

Automata for transductions: transducers

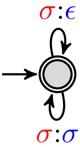
f_{del} :



$$
\begin{aligned}
aabaa &\mapsto aaaa \\
aaba &\mapsto \text{undefined} \\
dom(f_{del}) &= \text{'even number of } a\text{'}
\end{aligned}
$$

Non-determinism

In general, transducers define binary *relations* in $\Sigma^* \times \Sigma^*$

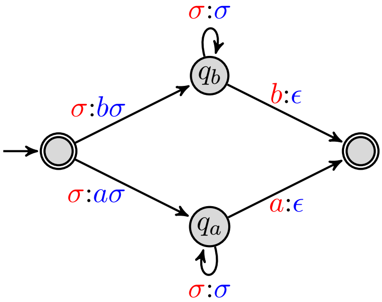


realizes $\{(u, v) \mid v \text{ is a subword of } u\}$

Sequential vs Non-deterministic functional

Non-deterministic transducers may define functions:

f_{sw} :

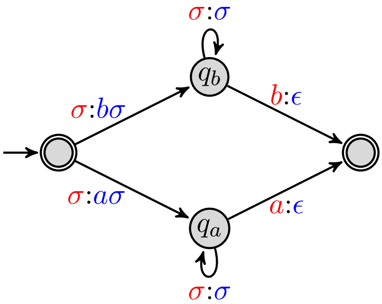


for all $\sigma \in \Sigma$

Sequential vs Non-deterministic functional

Non-deterministic transducers may define functions:

f_{sw} :



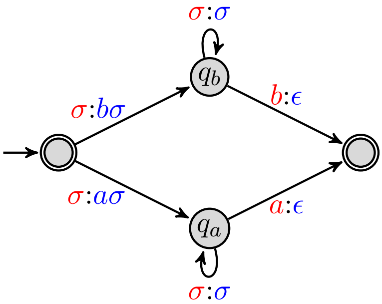
for all $\sigma \in \Sigma$

$$babaa \mapsto ababa$$

Sequential vs Non-deterministic functional

Non-deterministic transducers may define functions:

f_{sw} :



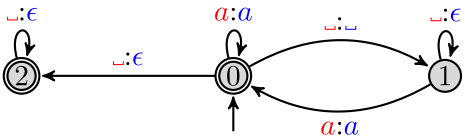
for all $\sigma \in \Sigma$

$$babaa \quad \mapsto \quad ababa$$

$$u\sigma \quad \mapsto \quad \sigma u \qquad |u| \geq 1$$

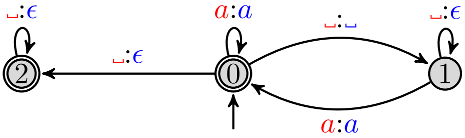
input-determinism (aka sequential) $<$ non-determinism $\cap$ functions

Determinizability



␣ = white space

Determinizability

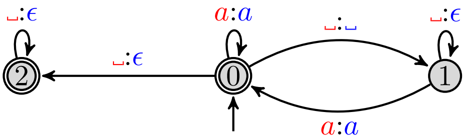


␣ = white space

␣␣aa␣␣␣a␣␣ ↦ ␣aa␣a

Determinizability

␣ = white space

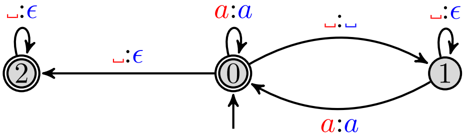


␣␣*aa*␣␣␣*a*␣␣ ↦ ␣*aa*␣*a*

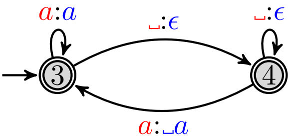
Is non-determinism needed ?

Determinizability

␣ = white space

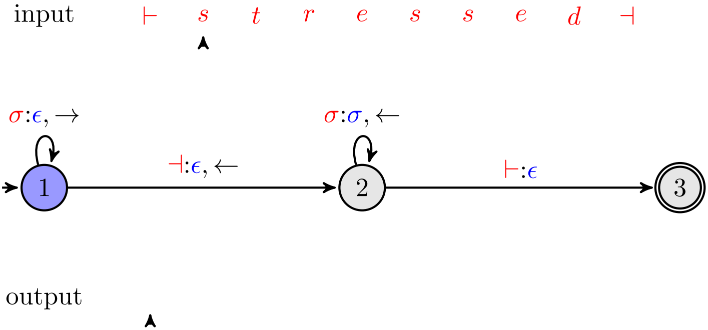


␣␣aa␣␣␣a␣␣ ↦ ␣aa␣a

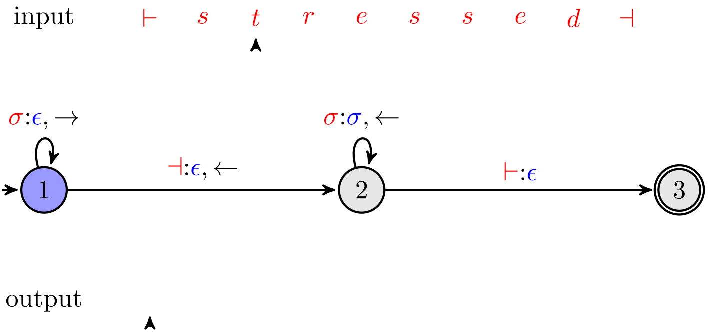


Is non-determinism needed ? No.

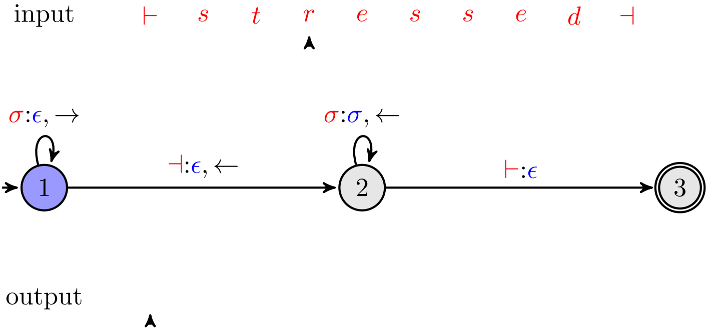
Two-way transducers



Two-way transducers



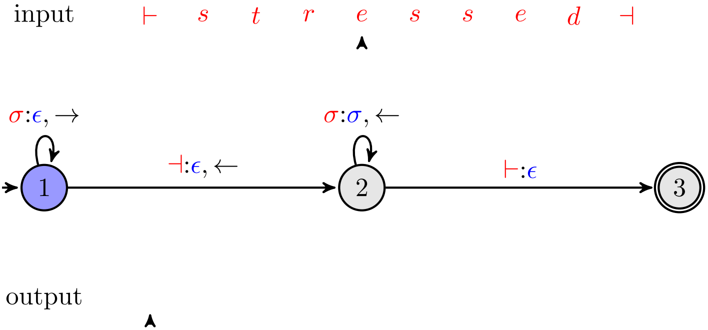
Two-way transducers



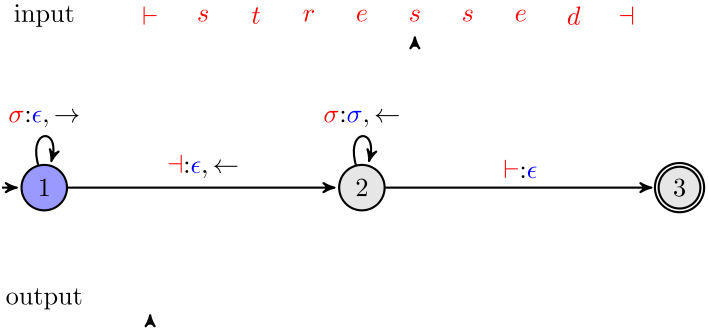
input $\vdash$ s t r e s s e d $\dashv$

output

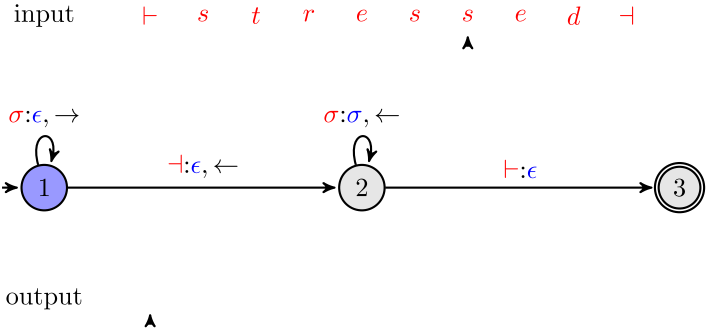
Two-way transducers



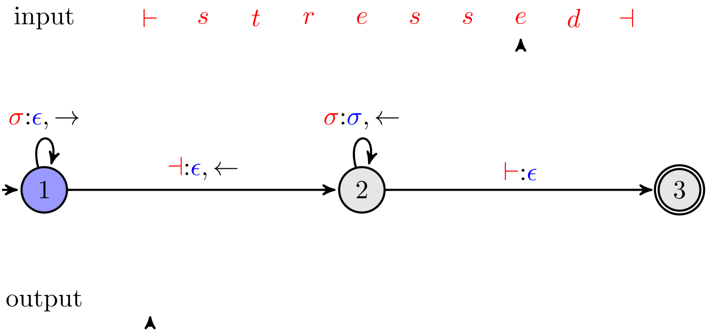
Two-way transducers



Two-way transducers

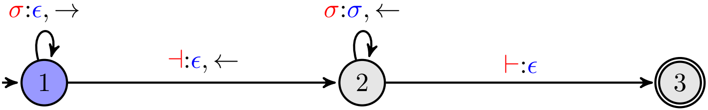


Two-way transducers



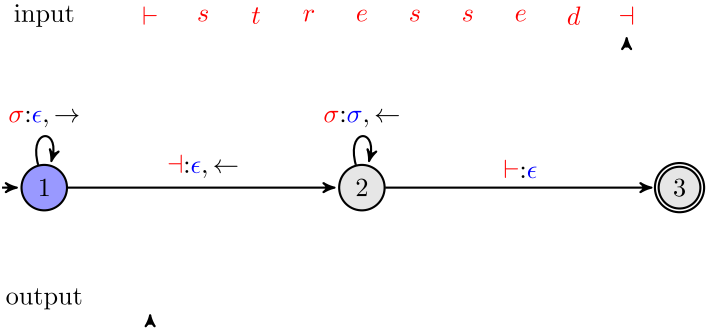
Two-way transducers

input $\vdash$ s t r e s s e d $\dashv$



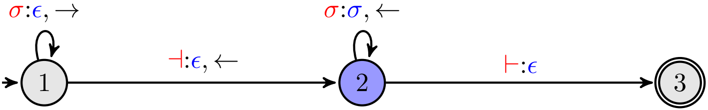
output

Two-way transducers



Two-way transducers

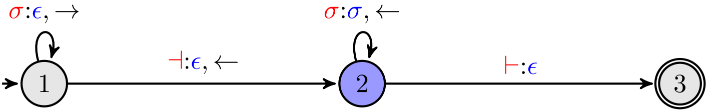
input $\vdash$ s t r e s s e d $\dashv$



output

Two-way transducers

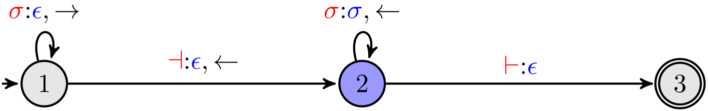
input $\vdash$ s t r e s s e d $\dashv$



output d

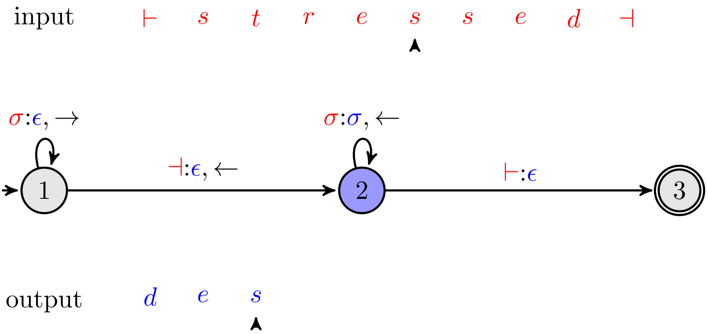
Two-way transducers

input $\vdash$ s t r e s s e d $\dashv$



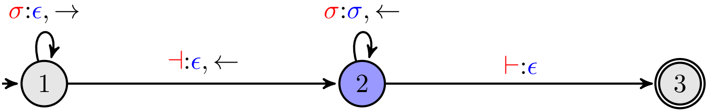
output d e

Two-way transducers



Two-way transducers

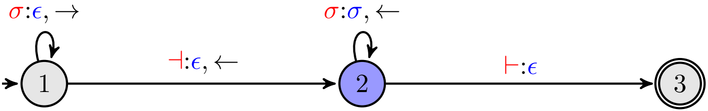
input $\vdash$ s t r e s s e d $\dashv$



output d e s s

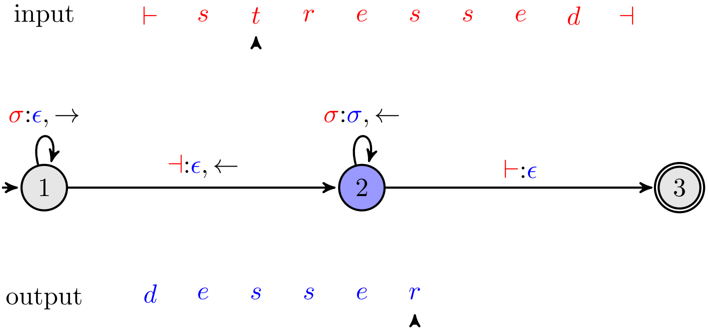
Two-way transducers

input $\vdash \; s \; t \; r \; e \; s \; s \; e \; d \; \dashv$



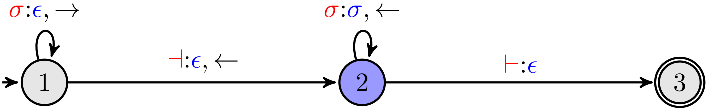
output $d \; e \; s \; s \; e$

Two-way transducers



Two-way transducers

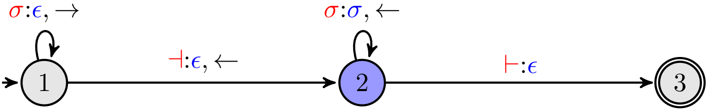
input $\vdash$ s t r e s s e d $\dashv$



output d e s s e r t

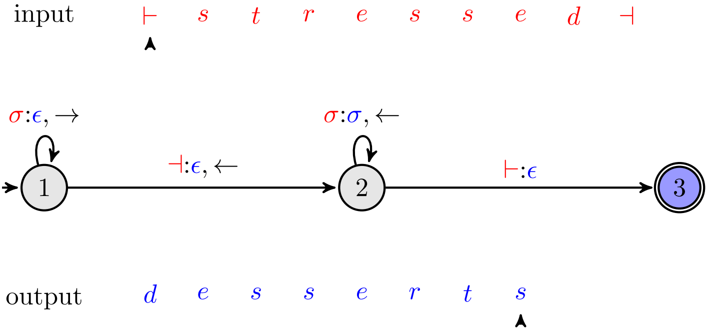
Two-way transducers

input $\vdash$ s t r e s s e d $\dashv$

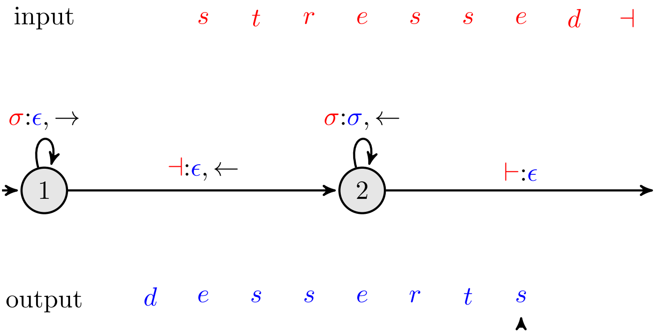


output d e s s e r t s

Two-way transducers



Two-way transducers



one-way < two-way

☺ decidable equivalence problem (Culik, Karhumaki, 87).

☺ closed under composition ∘ (Chytil, Jakl, 77)

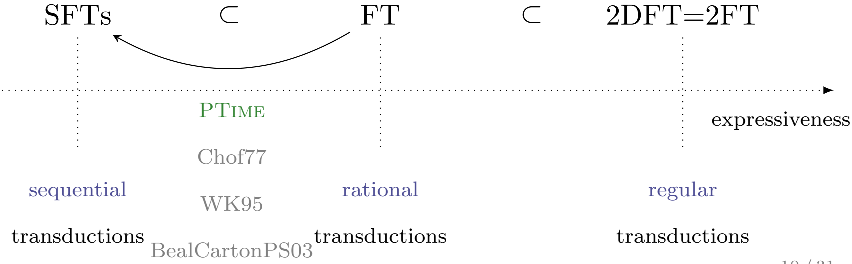
Landscape of Transducer Classes

SFTs ⊂ FT ⊂ 2DFT=2FT

expressiveness

sequential transductions	rational transductions	regular transductions

Landscape of Transducer Classes



SFTs ⊂ FT ⊂ 2DFT=2FT

expressiveness

PTime

Chof77

WK95

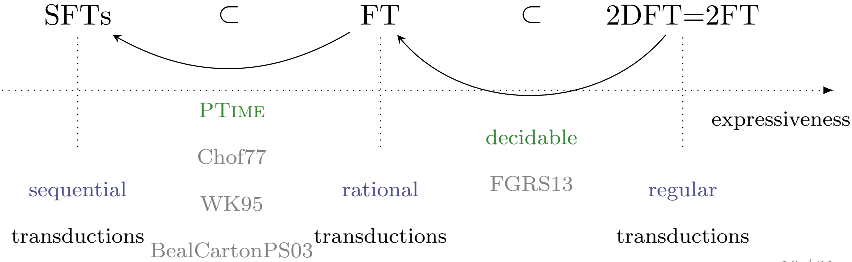
BealCartonPS03

sequential transductions

rational transductions

regular transductions

Landscape of Transducer Classes



SFTs ⊂ FT ⊂ 2DFT=2FT

PTime

Chof77

WK95

BealCartonPS03

decidable

FGRS13

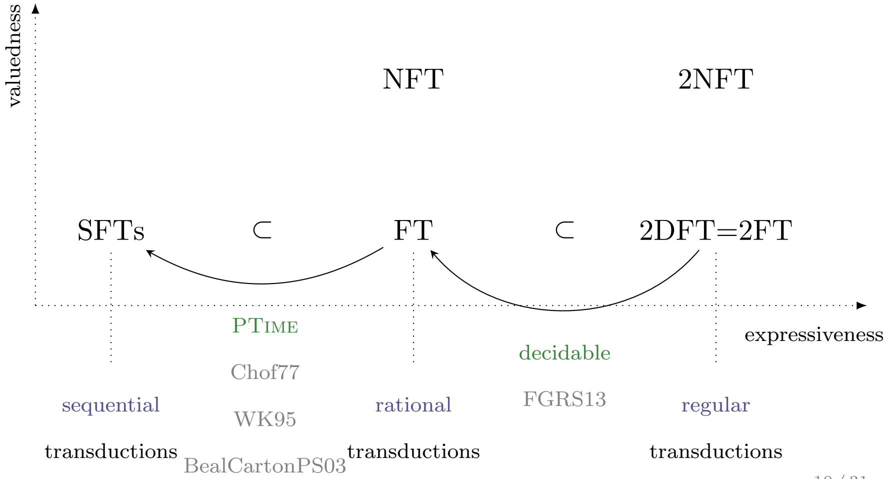
expressiveness

sequential transductions

rational transductions

regular transductions

Landscape of Transducer Classes



valuedness

NFT 2NFT

SFTs ⊂ FT ⊂ 2DFT=2FT

expressiveness

sequential transductions

PTime

Chof77

WK95

BealCartonPS03

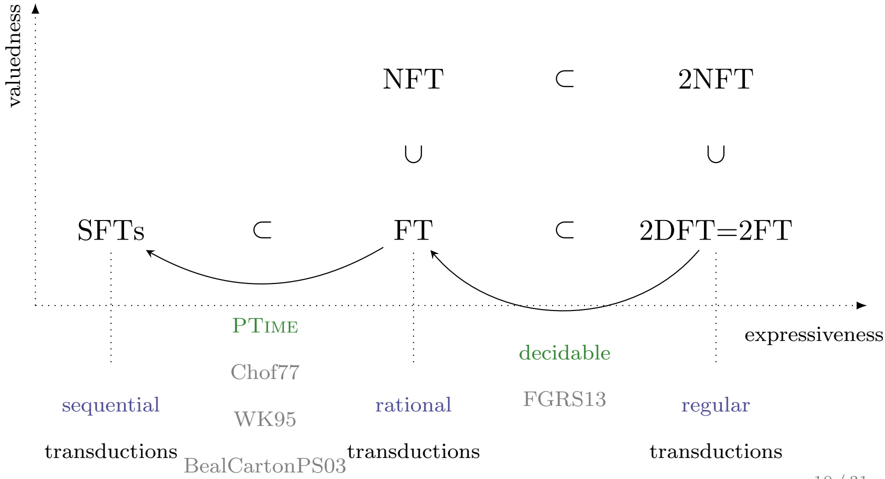
rational transductions

decidable

FGRS13

regular transductions

Landscape of Transducer Classes



valuedness

NFT ⊂ 2NFT

∪ ∪

SFTs ⊂ FT ⊂ 2DFT=2FT

expressiveness

PTIME

Chof77

WK95

BealCartonPS03

decidable

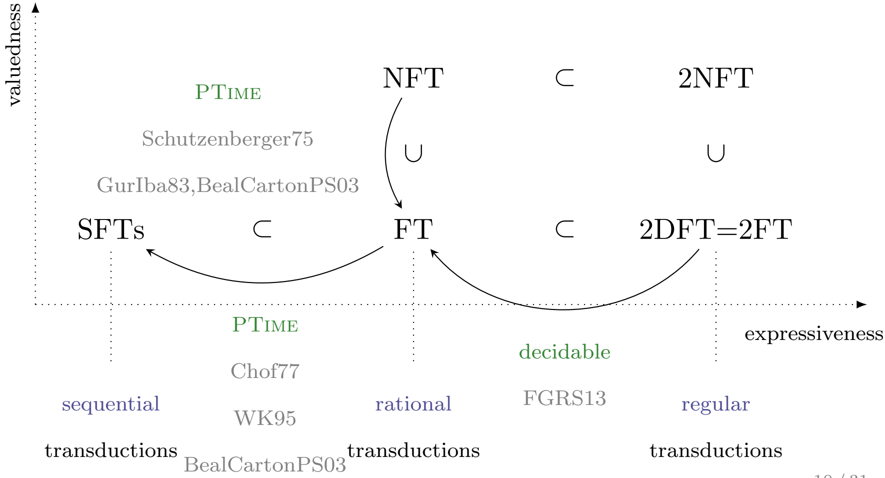
FGRS13

sequential transductions

rational transductions

regular transductions

Landscape of Transducer Classes



valuedness

PTIME

Schutzenberger75

GurIba83,BealCartonPS03

NFT ⊂ 2NFT

∪ ∪

SFTs ⊂ FT ⊂ 2DFT=2FT

PTIME

Chof77

WK95

BealCartonPS03

decidable

FGRS13

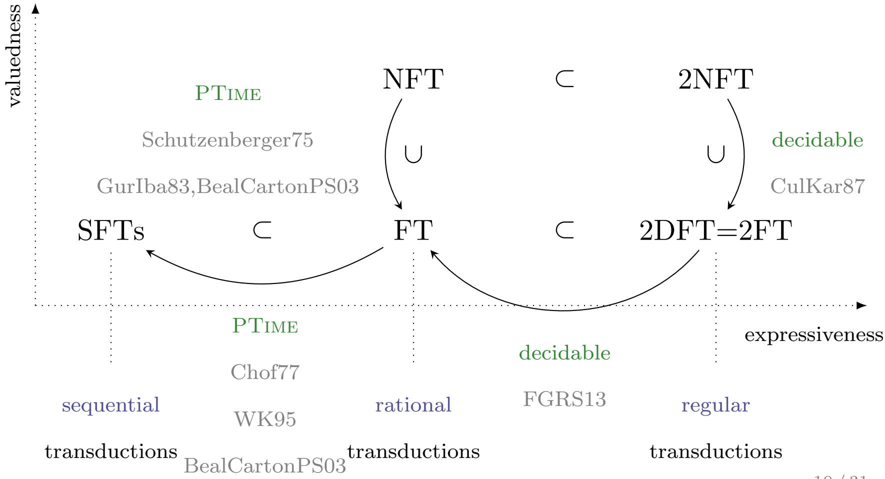
expressiveness

sequential transductions

rational transductions

regular transductions

Landscape of Transducer Classes



valuedness

PTime

Schutzenberger75

GurIba83,BealCartonPS03

NFT ⊂ 2NFT

∪ ∪

decidable

CulKar87

SFTs ⊂ FT ⊂ 2DFT=2FT

expressiveness

PTime

Chof77

WK95

BealCartonPS03

decidable

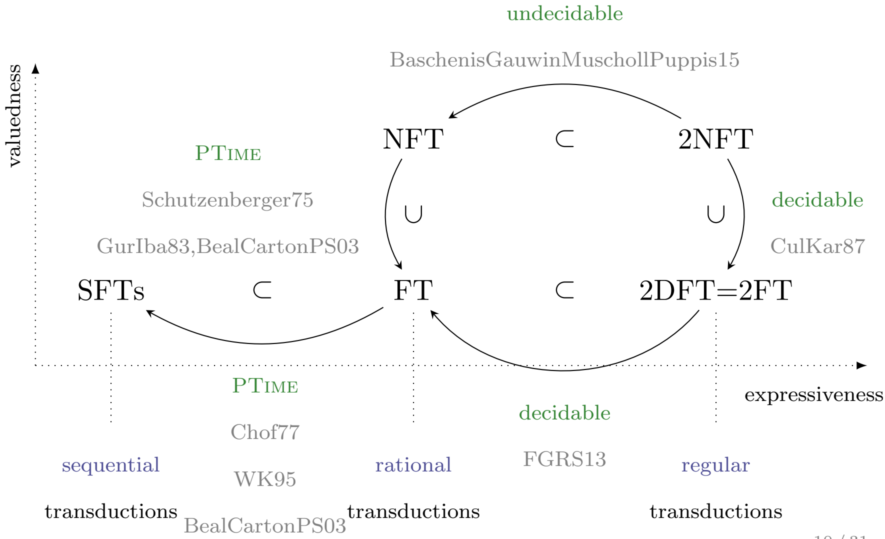
FGRS13

sequential transductions

rational transductions

regular transductions

Landscape of Transducer Classes



undecidable

BaschenisGauwinMuschollPuppis15

NFT ⊂ 2NFT

PTime

Schutzenberger75

GurIba83,BealCartonPS03

∪ ∪

decidable

CulKar87

SFTs ⊂ FT ⊂ 2DFT=2FT

valuedness

expressiveness

PTime

Chof77

WK95

BealCartonPS03

decidable

FGRS13

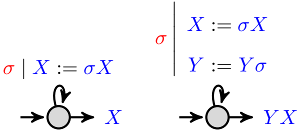
sequential transductions

rational transductions

regular transductions

Other recent results

Transducers with registers

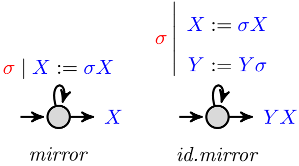


mirror *id.mirror*

- deterministic one-way
- equivalent to 2DFT if copyless updates (Alur, Cerny, 10)
- decidable equivalence problem (F., Reynier) ∼ HDT0L

Other recent results

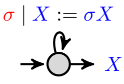
Transducers with registers



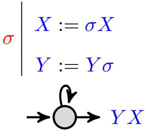
- deterministic one-way
- equivalent to 2DFT if copyless updates (Alur, Cerny, 10)
- decidable equivalence problem (F., Reynier) $\sim$ HDT0L
- regular expressions to register transducer, implemented in DReX (Alur, D'Antoni, Raghothaman, 2015)
- register minimization for a subclass (Baschenis, Gauwin, Muscholl, Puppis, 16)

Other recent results

Transducers with registers



mirror



id.mirror

- deterministic one-way
- equivalent to 2DFT if copyless updates (Alur, Cerny, 10)
- decidable equivalence problem (F., Reynier) $\sim$ HDT0L
- regular expressions to register transducer, implemented in DReX (Alur, D'Antoni, Raghothaman, 2015)
- register minimization for a subclass (Baschenis, Gauwin, Muscholl, Puppis, 16)

Two-way to one-way transducers

- decidable, but non-elementary complexity in (FGRS13)
- elementary complexity first obtained for subclasses (sweeping) by (BGMP15)
- recently for the full class (BGMP17)

Logic for transductions

(Courcelle) MSO Transformations

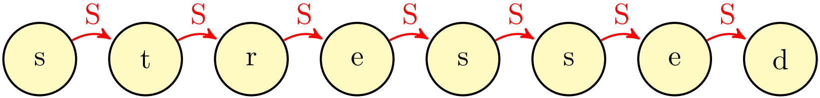
"interpreting the output structure in the input structure"

- output predicates defined by MSO[S] formulas interpreted over the input structure

(Courcelle) MSO Transformations

"interpreting the output structure in the input structure"

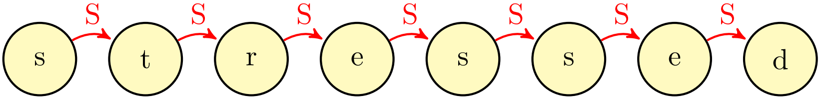
- output predicates defined by MSO[S] formulas interpreted over the input structure



(Courcelle) MSO Transformations

“interpreting the output structure in the input structure”

- output predicates defined by MSO[S] formulas interpreted over the input structure



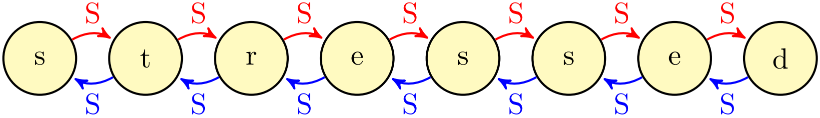
$$\phi_S(x,y) \equiv S(y,x)$$

$$\phi_\sigma(x) \equiv \sigma(x)$$

(Courcelle) MSO Transformations

"interpreting the output structure in the input structure"

- output predicates defined by MSO[S] formulas interpreted over the input structure



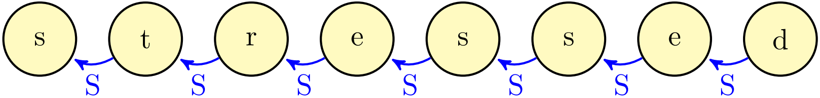
$$\phi_S(x, y) \equiv S(y, x)$$

$$\phi_\sigma(x) \equiv \sigma(x)$$

(Courcelle) MSO Transformations

"interpreting the output structure in the input structure"

- output predicates defined by MSO[S] formulas interpreted over the input structure



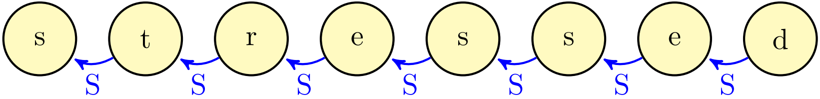
$$\phi_S(x, y) \quad \equiv \quad S(y, x)$$

$$\phi_\sigma(x) \quad \equiv \quad \sigma(x)$$

(Courcelle) MSO Transformations

"interpreting the output structure in the input structure"

- output predicates defined by MSO[S] formulas interpreted over the input structure



$$\phi_S(x,y) \quad \equiv \quad S(y,x)$$

$$\phi_\sigma(x) \quad \equiv \quad \sigma(x)$$

- input structure can be copied a fixed number of times: $u \mapsto uu$, or $u \mapsto u.\mathrm{mirror}(u)$.

Büchi Theorems for Word Transductions

Let $f : \Sigma^* \to \Sigma^*$.

Theorem (Engelfriet, Hoogeboom, 01)

f is 2FT-definable iff f is MSO-definable.

Büchi Theorems for Word Transductions

Let $f : \Sigma^* \to \Sigma^*$.

Theorem (Engelfriet, Hoogeboom, 01)

f is 2FT-definable iff f is MSO-definable.

Consequence Equivalence is decidable for MSO-transducers.

Büchi Theorems for Word Transductions

Let $f : \Sigma^* \to \Sigma^*$.

Theorem (Engelfriet, Hoogeboom, 01)

f is 2FT-definable iff f is MSO-definable.

Consequence Equivalence is decidable for MSO-transducers.

Theorem (Bojanczyk 14, F. 15)

f is (1)FT-definable iff f is order-preserving MSO-definable.

Order-preserving MSO: $\phi_S^{i,j}(x,y) \models x \preceq y$.

First-order transductions

Replace MSO by FO formulas.

Results

- equivalent to aperiodic transducers with registers (F., Trivedi, Krishna S., 14)
- and to aperiodic 2DFT (Carton, Dartois, 15) (Dartois, Jecker, Reynier, 16)

Algebraic characterizations of transductions

Myhill-Nerode congruence for

- $u \sim_L v$ if: for all $w \in \Sigma^*$, $uw \in L$ iff $vw \in L$
- u and v have the same “effect” on continuations w
- **Myhill-Nerode's Thm**: L is regular iff $\Sigma^*/_{\sim_L}$ is finite
- canonical (and minimal) deterministic automaton for L, $\Sigma^*/_{\sim_L}$ as set of states

Myhill-Nerode congruence for

- $u \sim_L v$ if: for all $w \in \Sigma^*$, $uw \in L$ iff $vw \in L$
- u and v have the same "effect" on continuations w
- **Myhill-Nerode's Thm**: L is regular iff $\Sigma^*/_{\sim_L}$ is finite
- canonical (and minimal) deterministic automaton for L, $\Sigma^*/_{\sim_L}$ as set of states

Goal

Extend Myhill-Nerode's theorem to classes of transductions

Sequential transductions (Choffrut)

Refinement of the MN congruence.

Two ideas

1. produce asap: $F(u) = LCP\{f(uw) \mid uw \in dom(f)\}$

Sequential transductions (Choffrut)

Refinement of the MN congruence.

Two ideas

1. produce asap: $F(u) = LCP\{f(uw) \mid uw \in dom(f)\}$
2. $u \sim_f v$ if
   2.1 $u \sim_{dom(f)} v$
   2.2 $F(u)^{-1}f(uw) = F(v)^{-1}f(vw)$ $\forall w \in u^{-1}dom(f)$

"u and v have the same effect on continuations w w.r.t. domain membership and produced outputs"

Sequential transductions (Choffrut)

Refinement of the MN congruence.

Two ideas

1. produce asap: $F(u) = LCP\{f(uw) \mid uw \in dom(f)\}$
2. $u \sim_f v$ if
   - 2.1 $u \sim_{dom(f)} v$
   - 2.2 $F(u)^{-1}f(uw) = F(v)^{-1}f(vw) \qquad \forall w \in u^{-1}dom(f)$

"u and v have the same effect on continuations w w.r.t. domain membership and produced outputs"

Theorem (Choffrut)

f is sequential iff $\sim_f$ has finite index

$\sim_f$ is a right congruence ⇝ canonical and minimal transducer !

Transitions: $[u] \xrightarrow{\sigma|F(u)^{-1}F(u\sigma)} [u\sigma]$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$a\,b\,b\,a\,a\,a\,a\,b\,b\,b\,b\,a\,b$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$a\,b\,b\,a\,a\,a\,a\,b\,b\,b\,b\,a\,\overset{\epsilon}{b}$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$$a\,b\,b\,a\,a\,a\,a\,b\,b\,b\,b\,\overset{b}{a}\,\overset{\epsilon}{b}$$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$$a\,b\,b\,a\,a\,a\,a\,b\,b\,b\,\overset{b}{b}\,\overset{b}{a}\,\overset{\epsilon}{b}$$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$$a\,b\,b\,a\,a\,a\,a\,b\,b\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{a}\,\overset{\epsilon}{b}$$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$$a\,b\,b\,a\,a\,a\,a\,b\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{a}\,\overset{\epsilon}{b}$$

Rational transductions are almost sequential

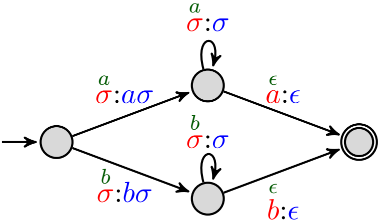
- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$\overset{b}{a}\overset{b}{b}\overset{b}{b}\overset{b}{a}\overset{b}{a}\overset{b}{a}\overset{b}{a}\overset{b}{b}\overset{b}{b}\overset{b}{b}\overset{b}{b}\overset{b}{a}\overset{\epsilon}{b}$

Rational transductions are almost sequential

- $f_{sw} : u\sigma \mapsto \sigma u$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

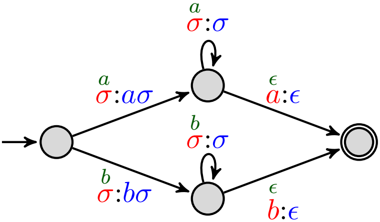
$\overset{b}{a}\overset{b}{b}\overset{b}{b}\overset{b}{a}\overset{b}{a}\overset{b}{a}\overset{b}{a}\overset{b}{b}\overset{b}{b}\overset{b}{b}\overset{b}{b}\overset{b}{a}\overset{\epsilon}{b}$



Rational transductions are almost sequential

- $f_{sw} : {\color{red}u\sigma} \mapsto {\color{blue}\sigma u}$ is not sequential
- but sequential modulo *look-ahead information* $\mathcal{I} = \{a, b, \epsilon\}$.

$$\overset{b}{a}\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{a}\,\overset{b}{a}\,\overset{b}{a}\,\overset{b}{a}\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{b}\,\overset{b}{a}\,\overset{\epsilon}{b}$$



- look-ahead information: $\mathcal{L} : \Sigma^* \to \mathcal{I}$
- $f[\mathcal{L}]$: f with input words extended with look-ahead information

Results

Theorem (Elgot, Mezei, 65)

f is rational iff $f[\mathcal{L}]$ is sequential, for some finite look-ahead information $\mathcal{L}$ computable by a right sequential transducer.

Original statement: $RAT = SEQ \circ RightSEQ$.

Results

Theorem (Elgot, Mezei, 65)

f is rational iff $f[\mathcal{L}]$ is sequential, for some finite look-ahead information $\mathcal{L}$ computable by a right sequential transducer.

Original statement: $RAT = SEQ \circ RightSEQ$.

Reutenauer, Schützenberger, 91

- *canonical look-ahead* given by a congruence $\equiv_f$
- identify suffixes with a 'bounded' effect on the transduction of prefixes
- characterization of rational transductions
  - f is rational
  - $\equiv_f$ has finite index and $f[\equiv_f]$ is sequential
  - $\equiv_f$ and $\sim_{f[\equiv_f]}$ have finite index.

Definability problems

Rational Transductions

Given f defined by T, is it definable by some $\mathcal{C}$-transducer ?

- sufficient conditions on $\mathcal{C}$ to get decidability (F., Gauwin, Lhote, LICS'16)
- includes aperiodic congruences: decidable FO-definability
- even PSpace-c (F., Gauwin, Lhote, FSTTCS'16)

Regular Transductions

- existence of a canonical transducer if *origin* is taken into account (Bojanczyk, ICALP'14)

- decidable FO-definability *with* origin, open without

A new logic for transductions

joint with Luc Dartois and Nathan Lhote

MOTIVATIONS

- specify properties of transductions in a high-level formalism: a logic
- decidable model-checking

$$T \models \varphi$$

a transducer (NFT, 2DFT, ...)

a formula

$$[\![T]\!] \subseteq [\![\varphi]\!]$$

a transduction $\Sigma^* \to \Sigma^*$ (function)

a relation $\subseteq \Sigma^* \times \Sigma^*$

EXAMPLES

- "there exists at least one 'a' in the output"

  $$\{ (u, v_1 a v_2) \mid u, v_1, v_2 \in \Sigma^* \}$$

- "every request is processed exactly one"

  $$\{ (r_{i_1} \cdots r_{i_k}, g_{\pi(i_1)} \cdots g_{\pi(i_k)}) \mid \pi \text{ permutation} \}$$

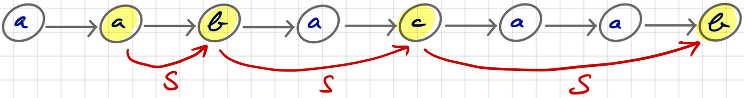
- more generally : shuffle

  $$\{ (\sigma_1 \cdots \sigma_n, \sigma_{\pi(1)} \cdots \sigma_{\pi(n)}) \mid \pi \text{ permutation} \}$$

NON-DETERMINISTIC MSOT (NMSOT)   COURCELLE

use second-order parameters $X_1, \dots, X_n$

$\{ (u, v) \mid v \text{ is a subword of } u, |v| \text{ even} \}$



$$\varphi_{dom}(X) \equiv even(X)$$

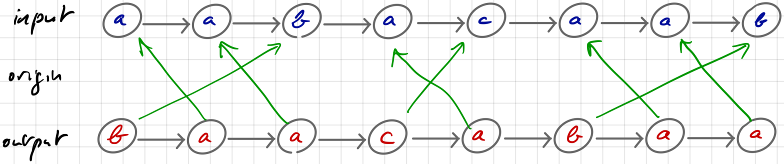
$$\varphi_S(x,y,X) \equiv x,y \in X \wedge x < y \wedge \neg(\exists z \in X \, . \, x < z < y)$$

FROM NMSOT to NEW LOGIC

NMSOT is not satisfactory as a specification language

- * too "operational"
- * the previous examples are not NMSOT

IDEA: SEE TRANSDUCTIONS AS SINGLE STRUCTURES WITH ORIGIN



Predicates: $\leq_{in}$, $\leq_{out}$, o

EXAMPLES

- "there exists at least one 'a' in the output"

$$\exists^{out} x \,.\, a(x)$$

- shuffle

$$BIJ(\sigma) \wedge \underbrace{\bigwedge_{\sigma \in \Sigma} \forall^{in} x \, \forall^{out} y \,.\, \begin{array}{c} \sigma(x,y) \wedge \\ \sigma(x) \end{array} \rightarrow \sigma(y)}_{LABPRES(\sigma)}$$

- identity

$$BIJ(\sigma) \wedge LABPRES(\sigma) \wedge ORDERPRES(\sigma)$$

where $ORDERPRES(\sigma) \equiv \forall^{in}_{x,x'} \forall^{out}_{y,y'} \begin{array}{c} \sigma(x,y) \\ \sigma(x',y') \\ x \leq_{in} x' \end{array} \rightarrow y \leq_{out} y'$

RESULTS

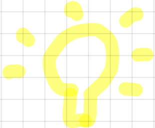
$FO[\leq_{in}, \leq_{out}, \theta]$ is undecidable

$FO_2[\leq_{out}, \theta, MSO_{bin}[\leq_{in}]]$ is decidable

→ capture MSOT (for functions)

→ all previous examples are definable

→ $T \models \varphi$ decidable for T: 2DFT



Reduction to a data word logic

$$FO_2[\leq, MSO_{bin}[\lesssim]]$$

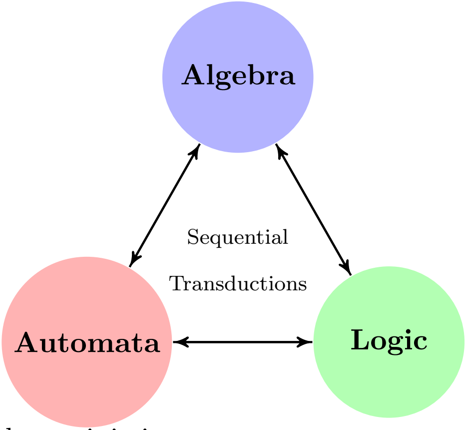
linear-order

total pre-order (data comparison)

"origin = data"

Summary: sequential transductions

finiteness of $\sim_f$ (Choffrut)

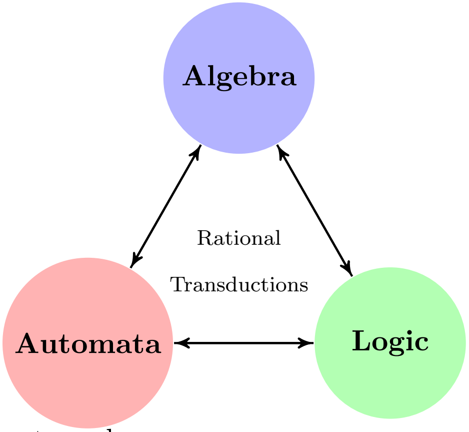


input-deterministic
one-way transducers

prefix-independent MSO, ?

Summary: rational transductions

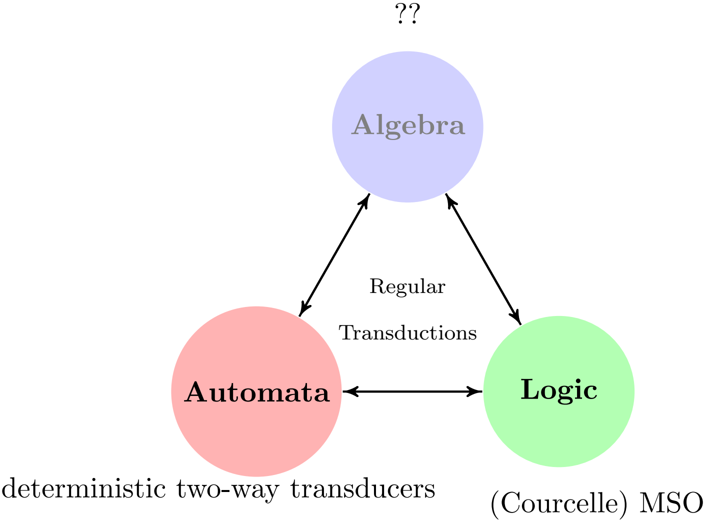
finiteness of $\equiv_f$ and $\sim_{f[\equiv_f]}$ (Reutenauer, Schützenberger)



one-way transducers

order-preserving MSO

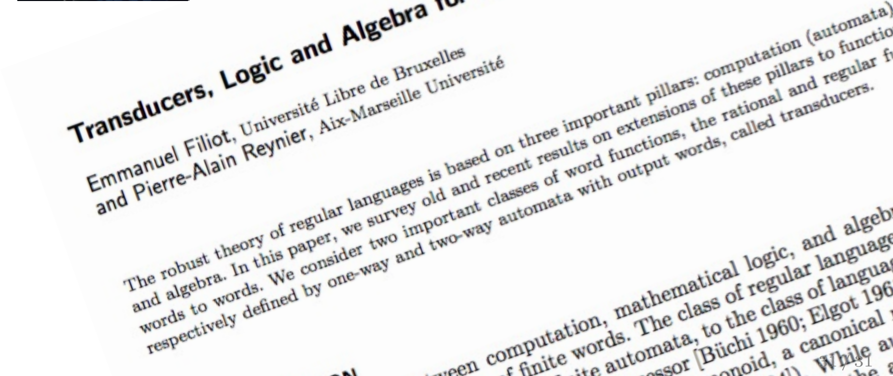
Summary: regular transductions



Other works

- AC^0 transductions (Cadilhac,Krebs,Ludwig,Paperman,15)
- variants of two-way transducers (Guillon, Choffrut, 14,15,16), (Carton, 12) (McKenzie, Schwentick, Thérien, Vollmer, 06)
- model-checking and synthesis problems for rational transductions with "similar origins" (F., Jecker, Löding, Winter, 16)
- non-determinism
- infinite words, nested words, trees

SIGLOG News 9th



Transducers, Logic and Algebra for Functions of Finite Words

Emmanuel Filiot, Université Libre de Bruxelles
and Pierre-Alain Reynier, Aix-Marseille Université

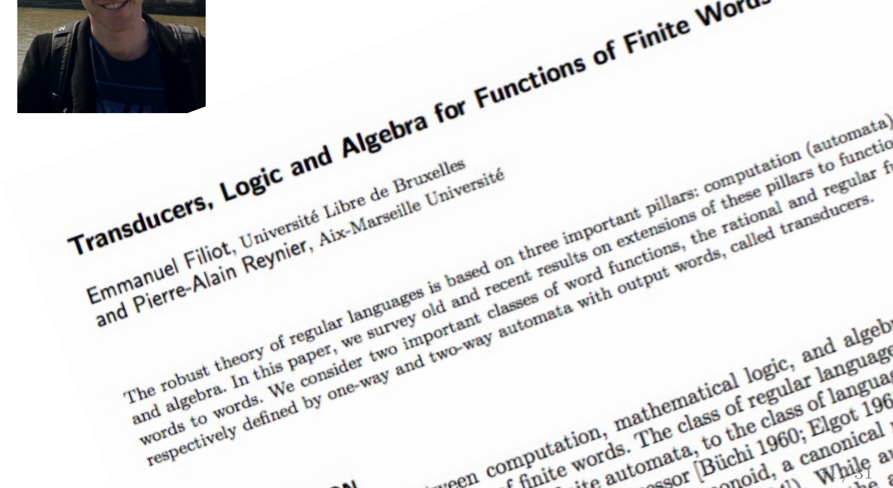
The robust theory of regular languages is based on three important pillars: computation (automata)
and algebra. In this paper, we survey old and recent results on extensions of these pillars to functio
words to words. We consider two important classes of word functions, the rational and regular f
respectively defined by one-way and two-way automata with output words, called transducers.

ON

ween computation, mathematical logic, and algeb
f finite words. The class of regular language
ite automata, to the class of langua
essor [Büchi 1960; Elgot 196
onoid, a canonical
While a

SIGLOG News 9th

Thank You.



Transducers, Logic and Algebra for Functions of Finite Words

Emmanuel Filiot, Université Libre de Bruxelles
and Pierre-Alain Reynier, Aix-Marseille Université

The robust theory of regular languages is based on three important pillars: computation (automata)
and algebra. In this paper, we survey old and recent results on extensions of these pillars to functio
words to words. We consider two important classes of word functions, the rational and regular f
respectively defined by one-way and two-way automata with output words, called transducers.

ween computation, mathematical logic, and algeb
f finite words. The class of regular language
ite automata, to the class of langua
essor [Büchi 1960; Elgot 196
onoid, a canonical
While a