Automata, Logic and Algebra for (Finite) Word Transductions

Emmanuel Filiot

Université libre de Bruxelles & FNRS

ACTS 2017, Chennai
Trinity for Regular Languages

\[ L \subseteq \Sigma^* \]
Trinity for Regular Languages

Finite monoids

Algebra

Regular languages

Automata

Logic

\( L \subseteq \Sigma^{\star} \)

DFA = NFA = 2DFA = 2NFA

MSO\([S]\)
Objective of the talk

Algebra

Transductions

$f : \Sigma^* \rightarrow \Sigma^*$

Automata

Logic
Automata models for transductions
Automata for transductions: transducers

\[ f_{\text{del}} : \]

\[
\begin{array}{ccc}
  b & \epsilon & a \times a \\
  \downarrow & \updownarrow & \updownarrow \\
  a \times a & \epsilon & b
\end{array}
\]

\[ \text{dom}(f_{\text{del}}) = \text{'even number of } a \text{'} \]
Automata for transductions: transducers

\[ f_{del} : \]

\[ aabaa \rightarrow aaaa \]
Automata for transductions: transducers

\[ f_{\text{del}} : \]

\[
\begin{align*}
    b &: \epsilon \\
    a &: a \\
    b &: \epsilon \\
\end{align*}
\]

\[
aabaa \; \rightarrow \; aaaa
\]

\[
aaba \; \rightarrow \; \text{undefined}
\]
Automata for transductions: transducers

$f_{del}:$

- $b: \epsilon$
- $a:a$
- $b: \epsilon$

$$aabaa \mapsto aaaa$$

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

$dom(f_{del}) = \text{'even number of } a\text{' }$
Non-determinism

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

\(\sigma : \epsilon\)

realizes \(\{(u, v) \mid v \text{ is a subword of } u\}\)
Sequential vs Non-deterministic functional

Non-deterministic transducers may define functions:

$$f_{sw} : a \sigma \rightarrow b \sigma$$

for all $$\sigma \in \Sigma$$
Non-deterministic transducers may define functions:

\[ f_{sw} : q_a \leftrightarrow q_b \]

for all \( \sigma \in \Sigma \)

\[ \sigma : \sigma \]

\[ \sigma : a\sigma \]

\[ a : \epsilon \]

\[ \sigma : b\sigma \]

\[ b : \epsilon \]

\[ \sigma : \sigma \]

Example:

\[ \text{babaa} \rightarrow \text{ababa} \]
Sequential vs Non-deterministic functional

Non-deterministic transducers may define functions:

\[ f_{sw} : \]

\[ \sigma : \sigma \]
\[ \sigma : b\sigma \]
\[ \sigma : a\sigma \]
\[ b : \epsilon \]
\[ a : \epsilon \]
\[ \sigma : \sigma \]

for all \( \sigma \in \Sigma \)

\( baba \alpha \rightarrow ababa \)

\( u\sigma \rightarrow \sigma u \quad |u| \geq 1 \)

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

\[ \epsilon \quad a \cdot a \quad \epsilon \]

\[ \epsilon \quad a \cdot a \quad \epsilon \]

\[ \ldots \epsilon \quad a \cdot a \quad \ldots \epsilon \]

\[ \epsilon = \text{white space} \]
Determinizability

\[
\begin{array}{c}
\epsilon \\
\downarrow \\
2
\end{array}
\quad
\begin{array}{c}
a \cdot a \\
\downarrow \\
0
\end{array}
\quad
\begin{array}{c}
\epsilon \\
\downarrow \\
1
\end{array}
\quad
\begin{array}{c}
\epsilon \\
\downarrow \\
\epsilon
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\epsilon \\
\downarrow \\
0
\end{array}
\quad
\begin{array}{c}
a \cdot a \\
\downarrow \\
1
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\epsilon \\
\downarrow \\
0
\end{array}
\quad
\begin{array}{c}
a \cdot a \\
\downarrow \\
1
\end{array}
\quad
\begin{array}{c}
\epsilon \\
\downarrow \\
\epsilon
\end{array}
\end{array}
\]

Is non-determinism needed? No.

\[
\text{white space}
\]
Determinizability

Is non-determinism needed?

$\epsilon = \text{white space}$

$aa \mapsto aa_\epsilon$
Determinizability

Is non-determinism needed? No.
Two-way transducers

\[ \sigma: \epsilon, \rightarrow \quad \neg: \epsilon, \leftarrow \quad \sigma: \sigma, \leftarrow \]

\[
\text{input} \quad \vdash s \quad t \quad r \quad e \quad s \quad s \quad e \quad d \quad \vdash \\
\]

\[
\text{output} \\
\]

© decidable equivalence problem (Culik, Karhumaki, 87).
© closed under composition (Chytil, Jakl, 77)
Two-way transducers

\[
\begin{array}{cccccccc}
\text{input} & \vdash & s & t & r & e & s & e & d & \vdash \\
\end{array}
\]

\[
\sigma : \epsilon, \rightarrow \quad \vdash : \epsilon, \leftarrow 
\]

\[
\sigma : \sigma, \leftarrow 
\]

\[
\begin{array}{c}
1 \\
2 \\
3
\end{array}
\]
Two-way transducers

input \[\vdash s t r e s s e d \vdash\]

- \(\sigma:\epsilon, \rightarrow\)
- \(\vdash:\epsilon, \leftarrow\)
- \(\sigma:\sigma, \leftarrow\)

output

\(\uparrow\)
Two-way transducers

\[ \sigma: \epsilon, \rightarrow \]

\[ \vdash: \epsilon, \leftarrow \]

\[ \sigma: \sigma, \leftarrow \]

\[ \vdash: \epsilon \]

\[ \vdash: s \quad t \quad r \quad e \quad s \quad s \quad e \quad d \quad \vdash \]

\[ \vdash: \epsilon \]

\[ \vdash: \epsilon \]

\[ \vdash: \epsilon \]
Two-way transducers

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

$\sigma: \epsilon, \rightarrow$

$\vdash: \epsilon, \leftarrow$

$\sigma: \sigma, \leftarrow$

output

$\vdash: \epsilon$
Two-way transducers

\[ \sigma : \varepsilon, \rightarrow \quad \vdash : \varepsilon, \leftarrow \quad \sigma : \sigma, \leftarrow \]

\[
\begin{array}{c}
1 \\
\rightarrow \\
\end{array}
\quad
\begin{array}{c}
2 \\
\rightarrow \\
\end{array}
\quad
\begin{array}{c}
3 \\
\uparrow \\
\end{array}
\]

input \quad \vdash s t r e s s e d ~ \vdash

output
Two-way transducers

\[
\begin{array}{l}
\text{input} \quad \vdash s t r e s s e d \quad \dashv \\
\sigma:\epsilon, \rightarrow \\
\sigma:\sigma, \leftarrow
\end{array}
\]

\[
\begin{array}{l}
1 \quad \vdash: \epsilon, \leftarrow \\
2 \quad \vdash: \epsilon
\end{array}
\]

output
Two-way transducers

\[
\text{input} \quad \vdash s \ t \ r \ e \ s \ s \ e \ d \ \vdash
\]

\[\sigma : \epsilon, \rightarrow\]

\[
\sigma : \sigma, \leftarrow
\]

output

\[\vdash : \epsilon\]

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

© closed under composition (Chytil, Jakl, 77)
Two-way transducers

\[
\begin{align*}
\text{input} & \quad \vdash s \quad t \quad r \quad e \quad s \quad s \quad e \quad d \quad \vdash \\
\sigma : \varepsilon , \rightarrow & \\
\sigma : \sigma , \leftarrow & \\
\text{output} & \\
\end{align*}
\]
Two-way transducers

input \( \vdash s t r e s s e d \vdash \)

\[ \sigma: \varepsilon, \rightarrow \quad \downarrow: \varepsilon, \leftarrow \quad \sigma: \sigma, \leftarrow \]

output
Two-way transducers

input

\[ \sigma: \epsilon, \rightarrow \]

\[ \sigma: \sigma, \leftarrow \]

output

\[ d \]
Two-way transducers

input

\[ \sigma : \epsilon, \rightarrow \]

\[ \vdash : \epsilon, \leftarrow \]

\[ \sigma : \sigma, \leftarrow \]

output

\[ d \quad e \]

© decidable equivalence problem (Culik, Karhumaki, 87).
© closed under composition (Chytil, Jakl, 77)
Two-way transducers

\[\vdash s \tau r e s s e d \vdash\]

\[\sigma: \epsilon, \rightarrow\]
\[\vdash: \epsilon, \leftarrow\]
\[\sigma: \sigma, \leftarrow\]

\[\vdash: \epsilon\]

output \(d e s\)

input \(\vdash\)
Two-way transducers

\[
\begin{array}{cccccccc}
\sigma: \epsilon, \rightarrow & \downarrow: \epsilon, \leftarrow & \sigma: \sigma, \leftarrow & \downarrow: \epsilon
\end{array}
\]

input $s t r e s s e d \rightarrow$

output $d e s s$
Two-way transducers

input \[\vdash s \quad t \quad r \quad e \quad s \quad s \quad e \quad d \quad \vdash\]

\[\sigma : \epsilon, \rightarrow\]

\[\vdash : \epsilon, \leftarrow\]

\[\sigma : \sigma, \leftarrow\]

\[\vdash : \epsilon\]

output \[d \quad e \quad s \quad s \quad e\]
Two-way transducers

<table>
<thead>
<tr>
<th>Input</th>
<th>s t r e s s e d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>d e s s e r</td>
</tr>
</tbody>
</table>

σ:ε, →

σ:σ, ←

© decidable equivalence problem (Culik, Karhumaki, 87).
© closed under composition (Chytil, Jakl, 77)
Two-way transducers

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

$\sigma : \epsilon, \rightarrow$

output: $d \ e \ s \ s \ e \ r \ t$

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

© closed under composition (Chytil, Jakl, 77).
Two-way transducers

input: \[
\sigma: \epsilon, \rightarrow \\
\sigma: \sigma, \leftarrow
\]

output: \[
d \ e \ s \ s \ e \ r \ t \ s
\]
Two-way transducers

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

\sigma: \epsilon, \rightarrow

\vdash: \epsilon, \leftarrow

\sigma: \sigma, \leftarrow

output \quad d e s s e r t s

\sigma: \epsilon, \rightarrow

\vdash: \epsilon, \leftarrow

\sigma: \epsilon, \rightarrow
Two-way transducers

input \( s \ t \ r \ e \ s \ s \ e \ d \ \vdash \)

\[ \sigma : \epsilon, \rightarrow \]

\[ \vdash : \epsilon, \leftarrow \]

\[ \sigma : \sigma, \leftarrow \]

\[ \vdash : \epsilon \]

output \( d \ e \ s \ s \ e \ r \ t \ s \)

one-way \(<\) two-way

😊 decidable equivalence problem (Culik, Karhumaki, 87).
😊 closed under composition \( \circ \) (Chytil, Jakl, 77)
Landscape of Transducer Classes

SFTs ⊂ FT ⊂ 2DFT=2FT

- sequential transductions
- rational transductions
- regular transductions
Landscape of Transducer Classes

\[ \text{SFTs} \subset \text{FT} \subset 2\text{DFT}=2\text{FT} \]

- Sequential transductions
- Rational transductions
- Regular transductions

Expressiveness:
- PTIME
- Chof77
- WK95

Decidability:
- BealCartonPS03
- GurIba83
- BealCartonPS03
- CulKar87
- Schutzenberger75
- BaschenisGauwinMuschollPuppis15
- FGRS13
Landscape of Transducer Classes

\[ \text{SFTs} \subset \text{FT} \subset 2\text{DFT}=2\text{FT} \]

- \( \subset \) denotes inclusion
- \( \text{PTime} \) for decidable problems
- \( \text{PTIME} \) for \( \text{Chof77} \)
- \( \text{WK95} \) for rational transductions
- \( \text{decidable} \) for \( \text{FGRS13} \)
- \( \text{regular} \) for \( \text{BealCartonPS03} \)

expressiveness
Landscape of Transducer Classes

- SFTs
- FT
- 2DFT = 2FT

Expressiveness:
- Sequential transductions
- Rational transductions
- Regular transductions

Valuedness:
- PTIME
- Chof77
- WK95
- BealCartonPS03
- Decidable: FGRS13
Landscape of Transducer Classes
Landscape of Transducer Classes

- **PTime**
  - Schützenberger75
  - Gur1ba83, BealCartonPS03

- **SFTs**

- **FT**
  - **NFT**
    - **2NFT**

- **2DFT=2FT**

- **sequential transductions**
  - Chof77
  - WK95

- **rational transductions**
  - BealCartonPS03

- **regular transductions**
  - decidable
    - FGRS13
Landscape of Transducer Classes

- **SFTs** ⊂ **FT** ⊂ **2NFT**
- **PTIME** ⊂ **NFT** ⊂ **2DFT=2FT**

**Valuelessness**
- Sequential transductions: BealCartonPS03
- Rational transductions: Schutzenberger75, GurIba83, BealCartonPS03
- Regular transductions: Chof77, WK95

**Expressiveness**
- Decidable transductions: CulKar87, FGRS13
- Undecidable transductions: BaschenisGauwinMuschollPuppis15

Figure: A landscape of transducers of finite words.
Landscape of Transducer Classes

- PTIME
- SFTs
- FT
- NFT
- 2NFT
- 2DFT = 2FT

Values:
- sequential transductions: BealCartonPS03
- rational transductions: WK95, Chof77
- regular transductions: Schutzenberger75, GurIba83, BealCartonPS03

Expressiveness:
- PTIME
- decidable: FGRS13, CulKar87
- undecidable: BaschenisGauwinMuschollPuppis15

Intersection:
- FT ⊂ NFT ⊂ 2NFT ⊂ 2DFT = 2FT
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
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

- 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[\mathcal{S}] formulas interpreted over the input structure

\[
\phi_{\mathcal{S}}(x, y) \equiv \mathcal{S}(y, x)
\]

\[
\phi_{\sigma}(x) \equiv \sigma(x)
\]
(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

\[ \phi_S(x, y) \equiv S(y, x) \]
\[ \phi_\sigma(x) \equiv \sigma(x) \]

- input structure can be copied a fixed number of times:
  \[ u \mapsto uu, \text{ or } u \mapsto u.\text{mirror}(u). \]
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 \( f : \Sigma^* \rightarrow \Sigma^* \).

**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 \( f : \Sigma^* \rightarrow \Sigma^* \).

**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: \( \phi^{i,j}_S(x, y) \models x \leq y \).
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: \( F(u) = LCP\{ f(uw) \mid uw \in \text{dom}(f) \} \)
Sequential transductions (Choffrut)

Refinement of the MN congruence.

Two ideas

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

“\( u \) 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: \( F(u) = LCP\{f(uw) \mid uw \in \text{dom}(f)\} \)

2. \( u \sim_f v \) if
   
   2.1 \( u \sim_{\text{dom}(f)} v \)
   
   2.2 \( F(u)^{-1}f(uw) = F(v)^{-1}f(vw) \quad \forall w \in u^{-1}\text{dom}(f) \)

“\( u \) and \( v \) have the same effect on continuations \( w \) w.r.t. domain membership and produced outputs”

Theorem (Choffrut)

\[ f \text{ is sequential iff } \sim_f \text{ has finite index} \]

\( \sim_f \) is a right congruence \( \sim \) 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* $I = \{a, b, \epsilon\}$. 

```
abbaaabbbab
```
Rational transductions are almost sequential

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

$$a b b a a a a b b b b a \epsilon b$$
Rational transductions are almost sequential

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

\[
\begin{array}{cccccc}
ab & ba & a & a & ab & b \\
\end{array}
\]

\[
\begin{array}{cccccc}
b & bab & b & \epsilon \\
\end{array}
\]
Rational transductions are almost sequential

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

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

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

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

\[
\begin{array}{c}
b b b b b b b b b \epsilon \\
a b b a a a a b b b b a b
\end{array}
\]
Rational transductions are almost sequential

- \( f_{sw} : u\sigma \mapsto \sigma u \) is not sequential
- but sequential modulo \textit{look-ahead information} \( \mathcal{I} = \{ a, b, \epsilon \} \).
Rational transductions are almost sequential

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

- look-ahead information: $\mathcal{L} : \Sigma^* \rightarrow I$
- $f[\mathcal{L}]$: $f$ 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 \( C \)-transducer?

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

Regular Transductions

- existence of a canonical transducer if \( \text{origin} \) is taken into account (Bojanczyk, ICALP’14)
- decidable FO-definability \( \text{with} \) \( \text{origin} \), 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

\[ T \models \varphi \]

\[
\uparrow \quad \uparrow
\]

a transducer \quad a formula

\[(\text{NFT, EDF, } \ldots)\]

\[ [T] \subseteq [\varphi] \]

\[
\uparrow \quad \uparrow
\]

a transduction \quad \Sigma^* \rightarrow \Sigma^* \quad a \text{ relation } \subseteq \Sigma^* \times \Sigma^* (\text{function})
EXAMPLES

1. "there exists at least one \( a \) in the output"
   \[
   \{ (u, v_1 a v_2) \mid u, v_1, v_2 \in \Sigma^* \}
   \]

2. "every request is processed exactly once"
   \[
   \{ (\Gamma_1 \cdots \Gamma_k, g_{\pi(i_1)} \cdots g_{\pi(i_k)}) \mid \pi \text{ permutation} \}
   \]

3. 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 $X_1, \ldots, X_n$

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

$\mathcal{F}_{\text{dom}}(X) = \text{even}(X)$

$\mathcal{F}_S(x, y, X) = \{ x, y \in X \land x < y \land (\exists z \in X : x < z < y) \}$
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}, \sigma
**EXAMPLES**

- "there exists at least one $\alpha'$ in the output"

  $$\exists x . \alpha(x)$$

- shuffle

  $$\text{BIJ}(\sigma) \land \bigwedge_{\tau \in \Sigma} \forall x, y . \sigma(x, y)^{\tau} \rightarrow \sigma(y)$$

- identity

  $$\text{BIJ}(\sigma) \land \text{LABPRES}(\sigma) \land \text{ORDERPRES}(\sigma)$$

  where

  $$\text{ORDERPRES}(\sigma) \equiv \forall x, x', y, y' . \sigma(x, y) \land \sigma(x', y') \rightarrow y \leq_{\sigma'} y'$$
RESULTS

\[ \text{FO}[\leq_{\text{in}}, \leq_{\text{out}}, \sigma] \text{ is } \text{undecidable} \]

\[ \text{FO}_2[\leq_{\text{out}}, \sigma, \text{MSO}_{\text{bin}}[\leq_{\text{in}}]] \text{ is decidable} \]

\[ \rightarrow \text{ capture MSOT (for functions)} \]
\[ \rightarrow \text{ all previous examples are definable} \]
\[ \rightarrow \text{ T is \textit{de}cidable for T: EFDT} \]

Reduction to a data word logic

\[ \text{FO}_2[\leq, \text{MSO}_{\text{bin}}[\leq]] \]

(Origin = data)
Summary: sequential transductions

finiteness of $\sim_f$ (Choffrut)

- Algebra
  - Sequential
  - Transductions
- Automata
  - input-deterministic
  - one-way transducers
- Logic
  - prefix-independent MSO, ?
Summary: rational transductions

finiteness of $\equiv_f$ and $\sim_f[\equiv_f]$ (Reutenauer, Schützenberger)
Summary: regular transductions

- Algebra
- Automata
- Logic

??

Deterministic two-way transducers

(Courcelle) MSO
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
Transducers, Logic and Algebra for Functions of Finite Words

Emmanuel Filot, 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 functions of words to words. We consider two important classes of word functions, the rational and regular functions respectively defined by one-way and two-way automata with output words, called transducers.
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 function words to words. We consider two important classes of word functions, the rational and regular functions respectively defined by one-way and two-way automata with output words, called transducers.