Which classes of origin graphs are generated by transducers?

Mikołaj Bojańczyk, Laure Daviaud, Bruno Guillon and Vincent Penelle

Uniwersytet Warszawski – LaBRI

March 26, 2018

— Réunion Delta —

This work is part of a project LIPA that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No.683080).
Definition

A transduction is a binary relation on words:

\[ \mathcal{R} \subseteq \Sigma^* \times \Gamma^*. \]
A *transduction* is a binary relation on words:

\[ \Sigma^* \times \Gamma^* \]

**Example:**
- Input alphabet: `aababbca`
- Output alphabet: `aaaabbbb` and `bbbaaaa`
A transduction is a binary relation on words:

\[ \Sigma^* \times \Gamma^* \times ??? \]

- **input alphabet**: \( \Sigma \)
- **output alphabet**: \( \Gamma \)
- **compute**: \( \Sigma^* \times \Gamma^* \times ??? \)
- **origin information**: [Bojańczyk 2014]
- **aababbbca**: \( \Sigma^* \)
- **aaaabbb**: \( \Gamma^* \)
- **bbbaaaa**: ???
origin graphs

origin: a mapping from output positions into input positions
What is the origin semantic of a transducer?

featuring SST and MSOT
Streaming String Transducers (SST)

A 1-way automaton $\mathcal{A}$

Example: $w \mapsto \cdot a \cdot w \cdot a \cdot b \cdot w \cdot b$
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register
  e.g., $R = \{ X, Y \}$
- a labelling of transitions by copyless register updates
  e.g.,
  \[
  \begin{align*}
  &\{ X \leftarrow X \cdot a \}, \quad \{ Y \leftarrow \varepsilon \},
  \\
  &\{ X \leftarrow a \cdot Y \}, \quad \{ Y \leftarrow b \cdot a \cdot X \},
  \\
  &\{ X \leftarrow b \cdot Y \}, \quad \{ Y \leftarrow X \cdot a \cdot Y \},
  \\
  &\{ X \leftarrow X \cdot a \}, \quad \{ Y \leftarrow X \cdot Y \}.
  \end{align*}
  \]

\[\text{Example: } w \mapsto \|
\begin{array}{c}
a \ b \ c \ a \ b \ b \ a \ c \ b
\end{array}\|
\]

\[\text{register } X: \quad \text{register } Y:
\]
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register
  e.g., $R = \{X, Y\}$

- a labelling of transitions by copyless register updates
  e.g., $\{ X \leftarrow X \cdot a, Y \leftarrow \varepsilon \}, \{ X \leftarrow a \cdot Y, Y \leftarrow b \cdot a \cdot X \}, \{ X \leftarrow b, Y \leftarrow X \cdot a \cdot Y \}, \{ X \leftarrow X \cdot a, Y \leftarrow X \cdot Y \}$

Definition (Origin semantics for streaming string transducers)

- origin of an output position: the position of the input head when the letter was created.
Streaming String Transducers (SST)

- a 1-way automaton \( \mathcal{A} \)
- a finite set \( R \) of registers including a distinguished output register e.g., \( R = \{X, Y\} \)
- a labelling of transitions by copyless register updates e.g., \( \{\ X \leftarrow X \cdot a \ \}, \ \{\ Y \leftarrow \varepsilon \ \}, \ \{\ Y \leftarrow b \cdot a \cdot X \ \}, \ \{\ Y \leftarrow X \cdot a \cdot Y \ \}, \ \{\ X \leftarrow X \cdot a \ \} \)

Example:

\( \rightarrow \quad w \rightarrow a | w |_a \cdot b | w |_b \)

Register \( X \):

Register \( Y \):
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register, e.g., $R = \{X, Y\}$
- a labelling of transitions by copyless register updates
  
  e.g., \[
  \begin{align*}
  &\{X \leftarrow X \cdot a, Y \leftarrow \varepsilon\}, \quad \{X \leftarrow a \cdot Y, Y \leftarrow b \cdot a \cdot X\}, \quad \{X \leftarrow b, Y \leftarrow X \cdot a \cdot Y\}, \quad \{X \leftarrow X \cdot a, Y \leftarrow X \cdot Y\}.
  \end{align*}
  \]

Example:

$w \mapsto a|w|_a \cdot b|w|_b$
Streaming String Transducers (SST)

- a 1-way automaton $A$
- a finite set $R$ of registers including a distinguished output register $e.g.$, $R = \{X, Y\}$
- a labelling of transitions by copyless register updates
  
  $e.g.$, $\{X \leftarrow X \cdot a, Y \leftarrow \varepsilon, Y \leftarrow b \cdot a \cdot X, Y \leftarrow X \cdot a \cdot Y, X \leftarrow X \cdot a, Y \leftarrow X \cdot Y\}$

Example:

$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$
Streaming String Transducers (SST)

- a 1-way automaton \( A \)
- a finite set \( R \) of registers including a distinguished output register
  - e.g., \( R = \{ X, Y \} \)
- a labelling of transitions by copyless register updates
  - e.g., \( \{ X \leftarrow X \cdot a \}, \{ Y \leftarrow \varepsilon \}, \{ X \leftarrow b \}, \{ Y \leftarrow b \cdot a \cdot X \}, \{ Y \leftarrow X \cdot a \cdot Y \}, \{ X \leftarrow X \cdot a \}, \{ Y \leftarrow X \cdot Y \} \)

\[ w \mapsto a|w|_a \cdot b|w|_b \]

Example:

- \( X \) register:
  - a
- \( Y \) register:
  - b
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register e.g., $R = \{X, Y\}$
- a labelling of transitions by copyless register updates
e.g., $\{X \leftarrow X \cdot a, Y \leftarrow \varepsilon, Y \leftarrow b \cdot a \cdot X, Y \leftarrow X \cdot a \cdot Y\}$

Example:
\[ w \mapsto a^{|w|_a} \cdot b^{|w|_b} \]
Streaming String Transducers (SST)

- a 1-way automaton \( \mathcal{A} \)
- a finite set \( R \) of registers including a distinguished output register  
  \[
  e.g., \ R = \{ X, Y \}
  \]
- a labelling of transitions by copyless register updates  
  \[
  e.g., \ \{ X \leftarrow X \cdot a \}, \ \{ Y \leftarrow \varepsilon \}, \ \{ Y \leftarrow b \cdot a \cdot X \}, \ \{ Y \leftarrow X \cdot a \cdot Y \}, \ \{ X \leftarrow X \cdot a \} \quad \text{and} \quad \{ Y \leftarrow X \cdot Y \}.
  \]

Example:

\[
  w \mapsto a^{|w|_a} \cdot b^{|w|_b}
  \]
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register $e.g., R = \{X, Y\}$
- a labelling of transitions by copyless register updates $e.g., \{X \leftarrow X \cdot a, Y \leftarrow \varepsilon, Y \leftarrow b \cdot a \cdot X, Y \leftarrow X \cdot a \cdot Y\}$

**Example:**

$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register e.g., $R = \{X, Y\}$
- a labelling of transitions by copyless register updates e.g.,
  \[
  \begin{align*}
  &X \leftarrow X \cdot a, \quad Y \leftarrow \varepsilon, \quad Y \leftarrow b \cdot a \cdot X, \quad Y \leftarrow X \cdot a \cdot Y, \\
  &\text{and } Y \leftarrow X \cdot Y \text{ is excluded.}
  \end{align*}
  \]

Example:

$w \mapsto a|w|_a \cdot b|w|_b$
Streaming String Transducers (SST)

- a 1-way automaton \( A \)
- a finite set \( R \) of registers including a distinguished output register e.g., \( R = \{ X, Y \} \)
- a labelling of transitions by copyless register updates e.g.,
  \[
  \begin{align*}
  &\{ X \leftarrow X \cdot a \}, \quad \{ X \leftarrow a \cdot Y \}, \quad \{ X \leftarrow b \}, \\
  &\{ Y \leftarrow \varepsilon \}, \quad \{ Y \leftarrow b \cdot a \cdot X \}, \quad \{ Y \leftarrow X \cdot a \cdot Y \}, \quad \{ X \leftarrow X \cdot a \}, \\
  &\quad \{ Y \leftarrow X \cdot Y \}.
  \end{align*}
  \]

Definition (Origin semantics for streaming string transducers)

origin of an output position: the position of the input head when the letter was created.

Example:

\[
\begin{array}{ccccccc}
  a & b & c & a & b & b & a \\
\end{array}
\]

\[
\begin{array}{ccccccc}
  a & a & a & b & b & b & b \\
\end{array}
\]

\[
\begin{array}{ccccccc}
  a & a & a & b & b & b & b \\
\end{array}
\]

Example:

\[
\begin{array}{ccccccc}
  w \mapsto a|w|_a \cdot b|w|_b
\end{array}
\]
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register
e.g., $R = \{X, Y\}$
- a labelling of transitions by copyless register updates
e.g.,
\[
\begin{align*}
X &\leftarrow X \cdot a \\
Y &\leftarrow \varepsilon \\
Y &\leftarrow b \cdot a \cdot X \\
Y &\leftarrow X \cdot a \cdot Y \\
X &\leftarrow X \cdot a \\
Y &\leftarrow X \cdot Y
\end{align*}
\]

Example:

$w \mapsto a | w |_a \cdot b | w |_b$
Streaming String Transducers (SST)

- a 1-way automaton $A$
- a finite set $R$ of registers including a distinguished output register
  e.g., $R = \{X, Y\}$
- a labelling of transitions by copyless register updates
  e.g., $\{X \leftarrow X \cdot a, Y \leftarrow \varepsilon, Y \leftarrow b \cdot a \cdot X, Y \leftarrow X \cdot a \cdot Y\}$

Example:

\[ w \mapsto a^{|w|_a} \cdot b^{|w|_b} \]
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register e.g., $R = \{X, Y\}$
- a labelling of transitions by copyless register updates e.g.,
\[
\begin{align*}
X &\leftarrow X \cdot a, \\
Y &\leftarrow \varepsilon, \\
Y &\leftarrow b \cdot a \cdot X, \\
Y &\leftarrow X \cdot a \cdot Y, \\
X &\leftarrow X \cdot a, \\
Y &\leftarrow X \cdot Y.
\end{align*}
\]

Example:
\[
w \mapsto a|w|_a \cdot b|w|_b
\]
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set $R$ of registers including a distinguished output register $e.g., R = \{X, Y\}$
- a labelling of transitions by copyless register updates $e.g.,$  
  \[
  \begin{align*}
  X &\leftarrow X \cdot a, \\
  Y &\leftarrow \varepsilon, \\
  X &\leftarrow b, \\
  Y &\leftarrow b \cdot a \cdot X, \\
  Y &\leftarrow X \cdot a \cdot Y
  \end{align*}
  \]

Example:

$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$
Streaming String Transducers (SST)

- a 1-way automaton \( A \)
- a finite set \( R \) of registers including a distinguished output register e.g., \( R = \{ X, Y \} \)
- a labelling of transitions by copyless register updates e.g.,
  \[
  \begin{align*}
  X & \leftarrow X \cdot a \\
  Y & \leftarrow \varepsilon \\
  X & \leftarrow a \cdot Y \\
  Y & \leftarrow b \cdot a \cdot X \\
  X & \leftarrow b \\
  Y & \leftarrow X \cdot a \\
  Y & \leftarrow X \cdot Y \\
  \end{align*}
  \]

Example:

\[ w \mapsto a|w|_a \cdot b|w|_b \]
string-to-string MSO-transduction [Courcelle 1991]

Definition (origin semantics of MSO-transduction)

origin of an output position: the input vertex of which it is a copy.
Nondeterministic \textit{MSO}-colouring \hspace{1em} (nondeterministic case);

- \textbf{Copy} \hspace{1em} (finitely many copies of the \textit{input});
string-to-string \textbf{MSO-transduction} [Courcelle 1991]

- **Nondeterministic \textbf{MSO-colouring}** (nondeterministic case);
- **Copy** (finitely many copies of the \textit{input});
- **MSO-Interpretation**
  - a formula for \textit{restricting} the universe;
  - a formula for each predicate of the \textit{output} vocabulary.

\textbf{Definition (origin semantics of MSO-transduction)}

The \textit{origin} of an output position: the input vertex of which it is a copy.

\begin{tikzpicture}
  \node (a) at (0,0) {$a$};
  \node (b) at (1,0) {$b$};
  \node (c) at (2,0) {$c$};
  \node (d) at (3,0) {$a$};
  \node (e) at (4,0) {$b$};
  \node (f) at (5,0) {$b$};
  \node (g) at (6,0) {$a$};
  \node (h) at (7,0) {$c$};
  \node (i) at (8,0) {$b$};
  \draw[->] (a) -- (b);
  \draw[->] (b) -- (c);
  \draw[->] (c) -- (a);
  \draw[->] (a) -- (b);
  \draw[->] (b) -- (b);
  \draw[->] (b) -- (a);
  \draw[->] (a) -- (c);
  \draw[->] (c) -- (b);
\end{tikzpicture}

\begin{tikzpicture}
  \node (a) at (0,0) [fill=blue] {$\text{copy 1}$};
  \node (b) at (1,0) [fill=gray] {$\text{copy 1}$};
  \node (c) at (2,0) [fill=blue] {$\text{copy 1}$};
  \node (d) at (3,0) [fill=gray] {$\text{copy 1}$};
  \node (e) at (4,0) [fill=blue] {$\text{copy 1}$};
  \node (f) at (5,0) [fill=gray] {$\text{copy 1}$};
  \node (g) at (6,0) [fill=blue] {$\text{copy 1}$};
  \node (h) at (7,0) [fill=gray] {$\text{copy 1}$};
  \node (i) at (8,0) [fill=blue] {$\text{copy 1}$};
  \node (j) at (9,0) [fill=gray] {$\text{copy 1}$};
  \node (k) at (10,0) [fill=blue] {$\text{copy 1}$};
  \node (l) at (11,0) [fill=gray] {$\text{copy 1}$};
  \node (m) at (12,0) [fill=blue] {$\text{copy 1}$};
  \node (n) at (13,0) [fill=gray] {$\text{copy 1}$};
  \node (o) at (14,0) [fill=blue] {$\text{copy 1}$};
  \node (p) at (15,0) [fill=gray] {$\text{copy 1}$};
  \node (q) at (16,0) [fill=blue] {$\text{copy 1}$};
  \node (r) at (17,0) [fill=gray] {$\text{copy 1}$};
  \node (s) at (18,0) [fill=blue] {$\text{copy 1}$};
  \node (t) at (19,0) [fill=gray] {$\text{copy 1}$};
  \node (u) at (20,0) [fill=blue] {$\text{copy 1}$};
  \node (v) at (21,0) [fill=gray] {$\text{copy 1}$};
  \node (w) at (22,0) [fill=blue] {$\text{copy 1}$};
  \node (x) at (23,0) [fill=gray] {$\text{copy 1}$};
  \node (y) at (24,0) [fill=blue] {$\text{copy 1}$};
  \node (z) at (25,0) [fill=gray] {$\text{copy 1}$};
  \draw[->] (a) -- (b);
  \draw[->] (b) -- (c);
  \draw[->] (c) -- (a);
  \draw[->] (a) -- (b);
  \draw[->] (b) -- (b);
  \draw[->] (b) -- (a);
  \draw[->] (a) -- (c);
  \draw[->] (c) -- (b);
\end{tikzpicture}
string-to-string MSO-transduction [Courcelle 1991]

- Nondeterministic MSO-colouring (nondeterministic case);
- Copy (finitely many copies of the input);
- MSO-Interpretation
  - a formula for restricting the universe;
  - a formula for each predicate of the output vocabulary.

![Diagram](attachment:image.png)
string-to-string MSO-transduction [Courcelle 1991]

- **Nondeterministic MSO-colouring** (nondeterministic case);
- **Copy** (finitely many copies of the input);
- **MSO-Interpretation**
  - a formula for restricting the universe;
  - a formula for each predicate of the output vocabulary.

---

**Definition (origin semantics of MSO-transduction)**

- **Origin** of an output position: the input vertex of which it is a copy.

---

**Diagram:**

- **Copy 1**
  - Origin sequence: $a b c a b b a c b$
- **Copy 2**
  - Origin sequence: $a b b b b$
string-to-string MSO-transduction [Courcelle 1991]

- Nondeterministic MSO-colouring (nondeterministic case);
- Copy (finitely many copies of the input);
- MSO-Interpretation
  - a formula for restricting the universe;
  - a formula for each predicate of the output vocabulary.

Definition (origin semantics of MSO-transduction)

- origin of an output position: the input vertex of which it is a copy.
string-to-origin graph MSO-transduction

- Nondeterministic MSO-colouring (nondeterministic case);
- Copy (finitely many copies of the input);
- MSO-Interpretation
  - a formula for restricting the universe;
  - a formula for each predicate of the output vocabulary.

---

### Definition (origin semantics of MSO-transduction)

The origin of an output position is the input vertex of which it is a copy.
Nondeterministic MSO-colouring (nondeterministic case);

Copy (finitely many copies of the input);

MSO-Interpretation

- a formula for restricting the universe;
- a formula for each predicate of the output vocabulary.
What do we get from origin information?
Origin semantics is thinner grained

Examples

**unary Identity**

![Graph representing unary identity](image)

**unary Reverse**

![Graph representing unary reverse](image)
Origin semantics is thinner grained

Examples

**unary Identity**

```
a → a → a → a → a
```

**unary Reverse**

```
a → a → a → a → a → a
```

≠
Origin semantics is thinner grained

Examples

unary Identity

unary Reverse

Subword
What is a regular transduction with origin?

This is still true with origin information. [Bojańczyk 2014]
also true for closure under composition, decidability of equivalence...

- **SST**: Streaming String Transducer
- **MSOT**: MSO-transduction
- **2FT**: 2-way finite transducer
Theorem: The following is decidable:

Input
- an NSST $\mathcal{A}$
- an MSO formula $\phi$ over the corresponding origin vocabulary

Question
- Is $\phi$ true in some origin graph in the origin semantics of $\mathcal{A}$?

Example

"the origin mapping is bijective and letter-preserving."

"the output may be split in two parts such that the origin mapping is order-preserving on each part."
Theorem: The following is decidable:

**Input**
- an NSST $\mathcal{A}$
- an MSO formula $\phi$ over the corresponding origin vocabulary

**Question**
- Is $\phi$ true in some origin graph in the origin semantics of $\mathcal{A}$?

**Proof:** Let $\mathcal{A}$ and $\phi$ be fixed.
- there is a string-to-origin graph MSO-transduction $\rho$ equivalent to $\mathcal{A}$
Theorem: The following is decidable:

**Input**
- an NSST $A$
- an MSO formula $\phi$ over the corresponding origin vocabulary

**Question**
- Is $\phi$ true in some origin graph in the origin semantics of $A$?

**Proof:** Let $A$ and $\phi$ be fixed.
- there is a string-to-origin graph MSO-transduction $\rho$ equivalent to $A$
- we consider $\mathcal{G} = \{ G \text{ origin graph} \mid \phi \text{ is true over } G \}$
**Theorem:** The following is decidable:

**Input**
- an NSST $\mathcal{A}$
- an MSO formula $\phi$ over the corresponding origin vocabulary

**Question**
- Is $\phi$ true in some origin graph in the origin semantics of $\mathcal{A}$?

**Proof:** Let $\mathcal{A}$ and $\phi$ be fixed.
- there is a string-to-origin graph MSO-transduction $\rho$ equivalent to $\mathcal{A}$
- we consider $\mathcal{G} = \{ G \text{ origin graph} \mid \phi \text{ is true over } G \}$
- by Backward Translation Theorem [Courcelle91], $\rho^{-1}(\mathcal{G})$ is regular and can be tested for emptiness.  □
Which properties of origin graphs characterise regular sets of origin graphs?
Which sets of origin graphs are generated by transducers?

**Theorem:**
A set of origin graphs is realised by an unambiguous SST if and only if it is

- **mso-definable:**
  an MSO sentence using $\rightarrow_1$, $\rightarrow_2$, $\rightarrow_3$ and labelling in $\Sigma \cup \Gamma$;

- **functional:**
  for each input word, there exists at most one origin graph;

- **bounded degree:**
  each input position is the origin of at most $m$ output positions;

- **bounded crossing:**
crossing of an **input position**
number of maximal factors of the **output**
that originate in the **input prefix** ended by the position

\[
\begin{align*}
&\text{a} \rightarrow \text{b} \rightarrow \text{c} \\
&\text{b} \rightarrow \text{a} \rightarrow \text{b} \\
&\text{b} \rightarrow \text{b} \rightarrow \text{a} \\
&\text{b} \rightarrow \text{c} \rightarrow \text{b}
\end{align*}
\]
crossing of an input position
number of maximal factors of the output
that originate in the input prefix ended by the position
crossing of an input position
number of maximal factors of the output
that originate in the input prefix ended by the position

crossing: 2
crossing of an input position
number of maximal factors of the output
that originate in the input prefix ended by the position

crossing of an origin graph: max of the crossings

(≡ particular path decomposition of bounded width)
Crossing

- crossing of an input position
- number of maximal factors of the output that originate in the input prefix ended by the position
- crossing of an origin graph: max of the crossings

Crossing: 2

(≡ particular path decomposition of bounded width)
- crossing of an **input position**
  number of maximal factors of the **output**
  that **originate** in the **input prefix** ended by the position
- crossing of an origin graph: **max of the crossings**

\[ \text{crossing: 2} \]

(≡ particular path decomposition of bounded width)
crossing of an input position
number of maximal factors of the output
that originate in the input prefix ended by the position

crossing of an origin graph: max of the crossings

(≡ particular path decomposition of bounded width)
Crossing

- crossing of an input position
- number of maximal factors of the output that originate in the input prefix ended by the position
- crossing of an origin graph: max of the crossings

(a b c a b b a c b)

(left1 right1 left2 right1 right2 left1 right1 left2 right2 left2 right2 left1 right1 left1)

crossing: 2

(≡ particular path decomposition of bounded width)
crossing of an input position
number of maximal factors of the output
that originate in the input prefix ended by the position

crossing of an origin graph: \( \text{max of the crossings} \)

\[
\begin{array}{ccccccc}
a & b & c & a & b & b & a & c & b \\
\text{left}_1 & \text{right}_1 & \text{left}_2 & \text{right}_2 & \text{left}_1 & \text{right}_1 & \text{left}_2 & \text{right}_2 & \text{left}_1 & \text{right}_1 & \text{left}_1
\end{array}
\]

\( \text{crossing: 2} \)

\( (\equiv \text{particular path decomposition of bounded width}) \)
crossing of an input position

number of maximal factors of the output

that originate in the input prefix ended by the position

crossing of an origin graph: \( \max \) of the crossings

(\( \equiv \) particular path decomposition of bounded width)
Crossing

- crossing of an **input position**
  number of maximal factors of the **output**
  that **originate** in the **input prefix** ended by the position
- crossing of an origin graph: max of the crossings

(≡ particular path decomposition of bounded width)
- crossing of an input position
- number of maximal factors of the output that originate in the input prefix ended by the position
- crossing of an origin graph: max of the crossings : 2

\[ \equiv \text{particular path decomposition of bounded width} \]
Theorem:
A set of origin graphs is realised by an unambiguous SST if and only if it is

- **mso-definable:**
  an MSO sentence using $\rightarrow, \rightarrow, \rightarrow$ and labelling in $\Sigma \cup \Gamma$;

- **functional:**
  for each input word, there exists at most one origin graph;

- **bounded degree:**
  each input position is the origin of at most $m$ output positions;

- **crossing bounded:**
Theorem:
A set of origin graphs is realised by an unambiguous \(k\)-registers \(\text{SST}\) if and only if it is

- **mso-definable:**
  an MSO sentence using \(\rightarrow, \rightarrow, \rightarrow\) and labelling in \(\Sigma \cup \Gamma\);

- **functional:**
  for each input word, there exists at most one origin graph;

- **bounded degree:**
  each input position is the origin of at most \(m\) output positions;

- **crossing bounded by \(k\):**

![Diagram of origin graphs]

PREVIOUS SLIDE
Which sets of origin graphs are generated by transducers?

**Theorem:**
A set of origin graphs is realised by an unambiguous sst if and only if it is

- **mso-definable:**
  an MSO sentence using $\rightarrow$, $\rightarrow$, $\rightarrow$ and labelling in $\Sigma \cup \Gamma$;

- **bounded degree:**
  each input position is the origin of at most $m$ output positions;

- **crossing bounded by $k$:**

![Diagram of origin graphs]

\[ a \rightarrow b \rightarrow c \rightarrow a \rightarrow b \rightarrow b \rightarrow a \rightarrow c \rightarrow b \]

\[ a \rightarrow a \rightarrow a \rightarrow b \rightarrow b \rightarrow b \rightarrow b \]
Sketch of the proof

- unambiguous $\Rightarrow$ functional
- NSST $\Rightarrow$ bounded degree
Sketch of the proof

- unambiguous $\implies$ functional
- NSST $\implies$ bounded degree
- $k$-register $\implies$ crossing bounded by $k$
Sketch of the proof \[\Rightarrow\]

- unambiguous \[\Rightarrow\] functional
- NSST \[\Rightarrow\] bounded degree
- \(k\)-register \[\Rightarrow\] crossing bounded by \(k\)
- NSST \[\Rightarrow\] string-to-origin graph MSO-transduction

\[
\rho
\]

\[
\begin{array}{c}
a-b-c-a-b-b-a-c-b \\
\end{array}
\]

\[
\begin{array}{c}
a-b-c-a-b-b-a-c-b \\
\end{array}
\]

\[
\begin{array}{c}
a-a-a-b-b-b-b \\
\end{array}
\]

\[
\begin{array}{c}
a-b-c-a-b-b-a-c-b \\
\end{array}
\]

\[
\begin{array}{c}
a-a-a-b-b-b-b \\
\end{array}
\]

\[
\frac{15}{18}
\]
Sketch of the proof

- unambiguous $\implies$ functional
- NSST $\implies$ bounded degree
- $k$-register $\implies$ crossing bounded by $k$
- NSST $\implies$ string-to-origin graph MSO-transduction

**Proposition:** we can inverse this MSO-transduction

![Diagram](image-url)
Sketch of the proof

- unambiguous $\Rightarrow$ functional
- NSST $\Rightarrow$ bounded degree
- $k$-register $\Rightarrow$ crossing bounded by $k$
- NSST $\Rightarrow$ string-to-origin graph MSO-transduction

**Proposition:** we can inverse this MSO-transduction
Sketch of the proof

- unambiguous $\Rightarrow$ functional
- NSST $\Rightarrow$ bounded degree
- $k$-register $\Rightarrow$ crossing bounded by $k$
- NSST $\Rightarrow$ string-to-origin graph MSO-transduction

**Proposition:** we can inverse this MSO-transduction $\Rightarrow$ MSO-definable
Sketch of the proof

- unambiguous $\implies$ functional
- NSST $\implies$ bounded degree
- $k$-register $\implies$ crossing bounded by $k$
- NSST $\implies$ string-to-origin graph MSO-transduction

**Proposition:** we can inverse this MSO-transduction $\implies$ MSO-definable

Note: False when $\varepsilon$-transitions are allowed.
Start with an **MSO-definable** set of origin graphs $G$ with crossing bounded by $k$. 

**Definition** $k$-block origin graphs ($k$-BLOGs): An origin graph with output split in $k$ identified blocks.
Start with an \textbf{MSO-definable} set of origin graphs $G$ with crossing bounded by $k$.

**Definition**

$k$-block origin graphs (\textit{k-BLOGs}):
An origin graph with \textbf{output} split in $k$ identified blocks.
Start with an **MSO-definable** set of origin graphs $G$ with crossing bounded by $k$.

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with output split in $k$ identified blocks.
Sketch of the proof

- Start with an MSO-definable set of origin graphs $G$ with crossing bounded by $k$

Definition

$k$-block origin graphs ($k$-BLOGs):
An origin graph with output split in $k$ identified blocks.
Sketch of the proof

- Start with an **MSO-definable** set of origin graphs $G$
  
  with crossing bounded by $k$

- We define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with output split in $k$ identified blocks.
Start with an **MSO-definable** set of origin graphs $G$ with crossing bounded by $k$

we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with output split in $k$ identified blocks.
Sketch of the proof

- Start with an **MSO-definable** set of origin graphs $G$ with crossing bounded by $k$

- we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with **output** split in $k$ identified blocks.
Sketch of the proof

- Start with an **MSO-definable** set of origin graphs $G$
  - with crossing bounded by $k$
- we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with output split in $k$ identified blocks.
Sketch of the proof

- Start with an **MSO-definable** set of origin graphs $G$
  with crossing bounded by $k$

- we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with **output** split in $k$ identified blocks.
Start with an **MSO-definable** set of origin graphs $G$ with crossing bounded by $k$

we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

**Definition**

$k$-block origin graphs ($k$-BLOGs):
An origin graph with output split in $k$ identified blocks.

![Diagram of $k$-block origin graphs](#)
Start with an \textbf{MSO-definable} set of origin graphs $G$ with crossing bounded by $k$

- we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

- the \textbf{folding} of a word $w$ over $\Omega_k^*$ is the $k$-BLOG $\alpha_k(w)$ obtained from the empty graph by applying the operations. $\alpha_k$ can be realised by an \textbf{MSO-transduction}. 
Start with an \textbf{MSO-definable} set of origin graphs $G$ with crossing bounded by $k$

we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

the \textbf{folding} of a word $w$ over $\Omega_k^*$ is the $k$-BLOG $\alpha_k(w)$ obtained from the empty graph by applying the operations. $\alpha_k$ can be realised by an MSO-transduction.

there exists a \textbf{regular language} $L \subseteq \Omega_k^*$ such that

$$g \in G \iff g = \alpha_k(w) \text{ for some } w \in L$$
Sketch of the proof

- Start with an MSO-definable set of origin graphs $G$ with crossing bounded by $k$

- We define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

- The folding of a word $w$ over $\Omega_k^*$ is the $k$-BLOG $\alpha_k(w)$ obtained from the empty graph by applying the operations. $\alpha_k$ can be realised by an MSO-transduction.

- There exists a regular language $L \subseteq \Omega_k^*$ such that
  \[ g \in G \iff g = \alpha_k(w) \text{ for some } w \in L \]

- From an automaton recognising $L$, we build a NSST with $\varepsilon$-transitions realising $G$
Start with an MSO-definable set of origin graphs $G$ with crossing bounded by $k$

we define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

the folding of a word $w$ over $\Omega_k^*$ is the $k$-BLOG $\alpha_k(w)$ obtained from the empty graph by applying the operations. $\alpha_k$ can be realised by an MSO-transduction.

there exists a regular language $L \subseteq \Omega_k^*$ such that

$$g \in G \iff g = \alpha_k(w) \text{ for some } w \in L$$

from an automaton recognising $L$, we build a NSST with $\varepsilon$-transitions realising $G$

if bounded degree $\implies$ elimination of $\varepsilon$-transition
Sketch of the proof

- Start with an **MSO-definable** set of origin graphs $G$ with crossing bounded by $k$
- We define a finite set of (partial) operations $\Omega_k$ on $k$-BLOGs

  - **The folding** of a word $w$ over $\Omega_k^*$ is the $k$-BLOG $\alpha_k(w)$ obtained from the empty graph by applying the operations. $\alpha_k$ can be realised by an **MSO-transduction**.
  - There exists a **regular language** $L \subseteq \Omega_k^*$ such that
    
    $$g \in G \iff g = \alpha_k(w) \text{ for some } w \in L$$

  - From an automaton recognising $L$, we build a **NSST** with $\varepsilon$-transitions realising $G$
    - If bounded degree $\implies$ elimination of $\varepsilon$-transition
    - If functional $\implies$ disambiguation
Corollary: The following is decidable:

**Input**
- Two NSST, $A$ and $B$.

**Question**
- Whether they have the same origin semantics.
Corollary: The following is decidable:

Input
- Two NSST, $A$ and $B$.

Question
- Whether they have the same origin semantics.

Proof: We show that we can check whether $A \cap \overline{B}$ is empty.
- The origin semantics of $B$ is MSO-definable by a formula $\phi$,
- We can check whether $\neg\phi$ is true in some origin graph in the origin semantics of $A$. 
Conclusion

bounded crossing

ε NSST

NSST = MSOT

f SST = 2f FT = f MSOT

REG

MSO

bounded degree

functional
variants of

\[
\epsilon_{\text{NSST}}
\]

\[
f_{\text{ST}} = f_{\text{MSOT}}
\]

\[
\text{with common guess}
\]

\[
\text{REG}
\]

\[
\text{with common guess}
\]

\[
\text{with common guess}
\]

\[
\epsilon_{\text{NSST}}
\]

\[
f_{\text{FT}}
\]

\[
\text{against}\n\]

\[
\text{as a consequence of}
\]

\[
\text{of w, } n \in \mathbb{N}
\]

\[
\text{and}
\]

\[
\text{in the case of}
\]

\[
\text{of w, } n \in \mathbb{N}
\]

\[
\text{of w, } n \in \mathbb{N}
\]
Conclusion

\[ \varepsilon NSST = NSST = MSOT \]

\[ fSST = 2ft = fMSOT \]

\[ fSST = 2ft = fMSOT \]

\[ \text{REG} \]

\[ fSST = 2ft = fMSOT \]

\[ = \]

\[ \{ (w, w) \mid w \in \Sigma^*, n \in \mathbb{N} \} \]

\[ \{ (a, an) \mid n \in \mathbb{N} \} \]

\[ \{ (w, v) \mid v \text{ is a subword of } w \} \]

\[ \{ (w, v^2) \mid v \text{ is a subword of } w \} \]

\[ \cup \{ (a, an) \mid n \in \mathbb{N} \} \]

\[ \{ (w, vn^2) \mid v \text{ is a subword of } w, n \in \mathbb{N} \} \]

Thank you for your attention.
Conclusion

...
Conclusion

bounded crossing

\[
\varepsilon_{NSST} = MSOT
\]

variants of

\[
\begin{align*}
\{ (w, w_1) &| w \in \Sigma^*, n \in \mathbb{N} \} \\
\{ (a, an) &| n \in \mathbb{N} \} \\
\{ (w, v) &| v \text{ is a subword of } w \} \\
\{ (w, w_2) &| v \text{ is a subword of } w \} \\
\cup \{ (a, an) &| n \in \mathbb{N} \} \\
\{ (w, v^n) &| v \text{ is a subword of } w, n \in \mathbb{N} \}
\end{align*}
\]

Thank you for your attention.
\[
\begin{align*}
\varepsilon \text{NSST} &= \text{MSOT} \\
2f \text{FT} &= f \text{MSOT} \\
2N \text{FT} &= 2N \text{FT with common guess} \\
\text{REG} &= \text{MSOT} \\
\text{variants of} &= \text{MSOT} \\
\end{align*}
\]
Conclusion

\[
\begin{align*}
\text{bounded crossing} & \quad \varepsilon \text{NSST} \\
\text{MSOT} & \quad \varepsilon \text{NSST}
\end{align*}
\]

\[
\begin{align*}
f_{\text{FT}} &= f_{\text{MSOT}}
\end{align*}
\]

\[
\begin{align*}
\text{REG} & \quad \varepsilon \text{NSST}
\end{align*}
\]

\[
\begin{align*}
\text{variations of MSOT}
\end{align*}
\]
\[ \begin{array}{c}
\{ (w, w^2) \mid w \in \Sigma^* \} \\
\{ (a, an) \mid n \in \mathbb{N} \} \\
\{ (w, v^2) \mid v \text{ is a subword of } w \} \\
\{ (w, v^2) \mid v \text{ is a subword of } w, n \in \mathbb{N} \} \\
\{ (w, v^2) \mid v \text{ is a subword of } w \} \cup \{ (a, an) \mid n \in \mathbb{N} \}
\end{array} \]
Conclusion
Conclusion

Thank you for your attention.