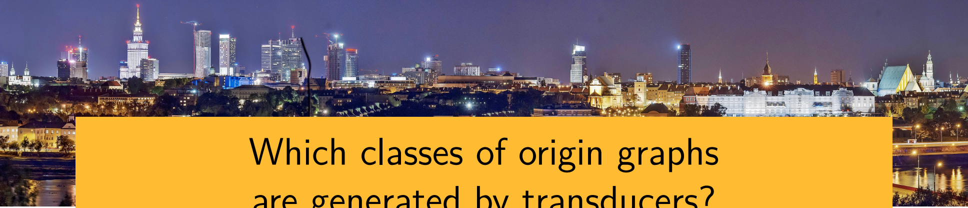


Which classes of origin graphs are generated by transducers?

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).*

Word transductions

Definition

A transduction is a binary relation on words:

a subset of $\Sigma^* \times \Gamma^*$.

Word transductions

Definition

A **transduction** is a binary relation on words:

a subset of $\Sigma^* \times \Gamma^*$.

input alphabet

output alphabet

compute

aababbca → *aaaabbb*

aababbca → *bbbaaaa*

Word transductions

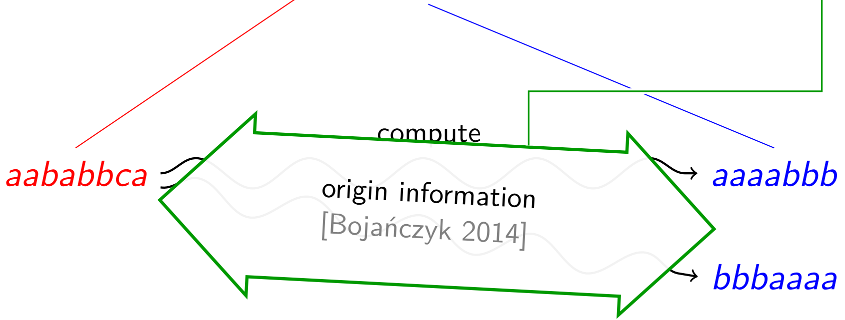
Definition

A **transduction** is a binary relation on words:

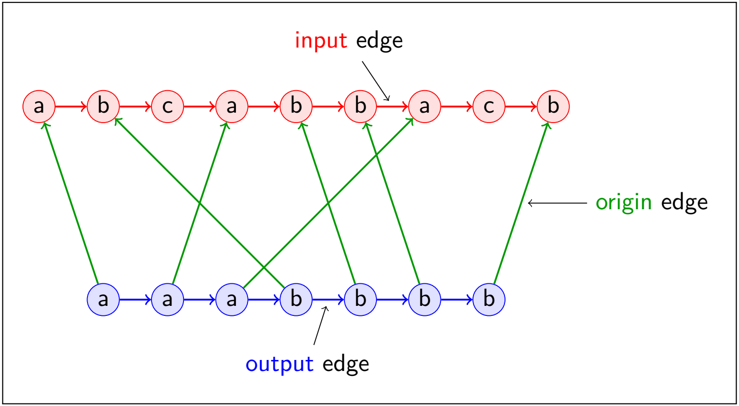
a subset of $\Sigma^* \times \Gamma^* \times$???.

input alphabet

output alphabet



Origin graphs



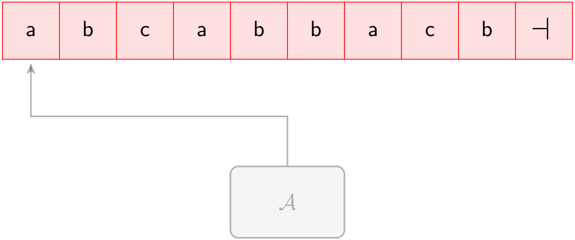
origin: a mapping from output positions into input positions

What is the origin semantic of a transducer?

featuring SST and MSOT

Streaming String Transducers (SST)

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



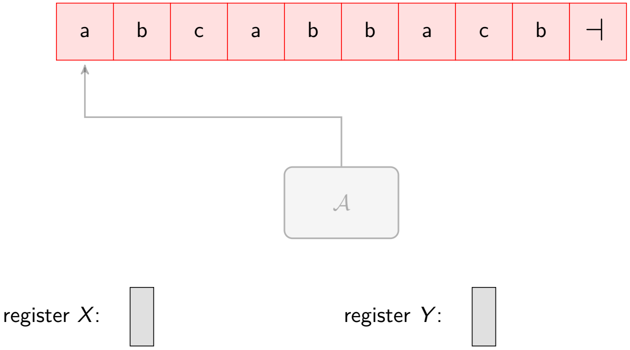
Streaming String Transducers (SST)

- 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

$$\textit{e.g.},\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



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., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.

Definition (Origin semantics for streaming string transducers)

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

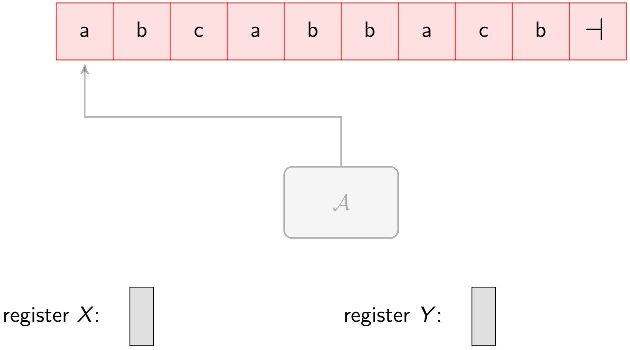
Streaming String Transducers (SST)

- 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

$$e.g.,\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

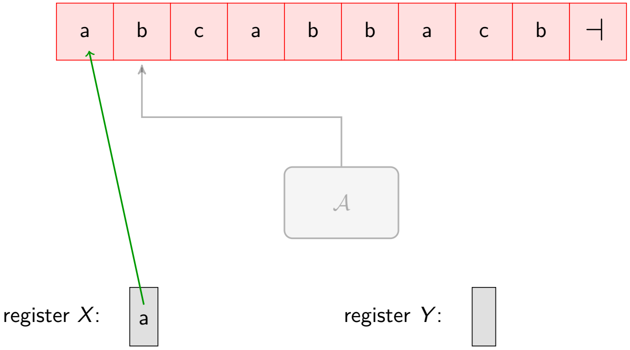
Streaming String Transducers (SST)

- 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., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.



- Example:

  $w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

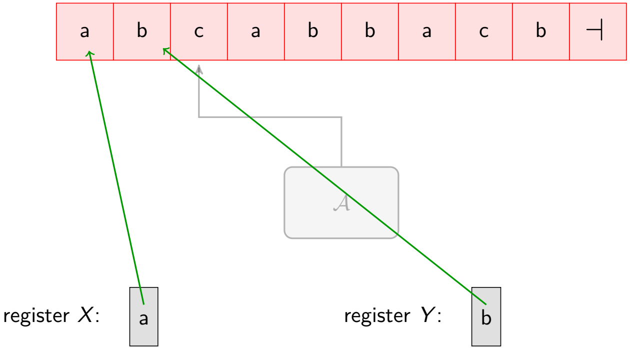
Streaming String Transducers (SST)

- 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

$$e.g.,\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

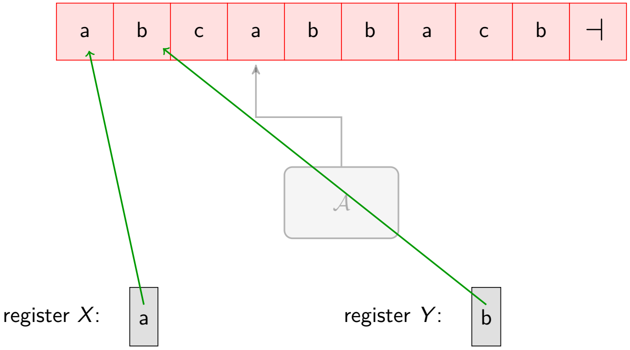
Streaming String Transducers (SST)

- 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., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

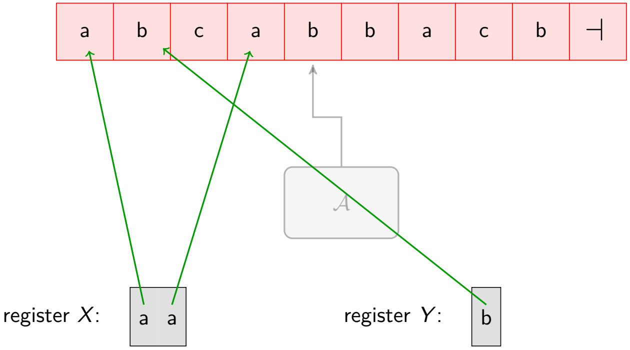
Streaming String Transducers (SST)

- 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

$$e.g.,\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

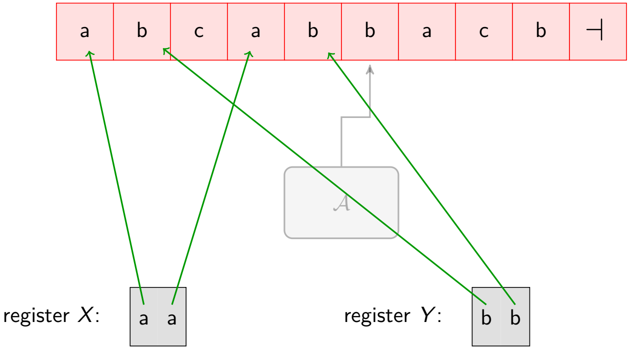
Streaming String Transducers (SST)

- 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

$$e.g., \quad \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}, \quad \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}, \quad \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}, \quad \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

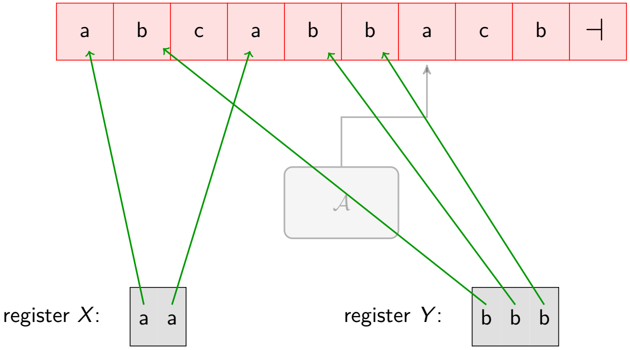
Streaming String Transducers (SST)

- 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

$$e.g., \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}, \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}, \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}, \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

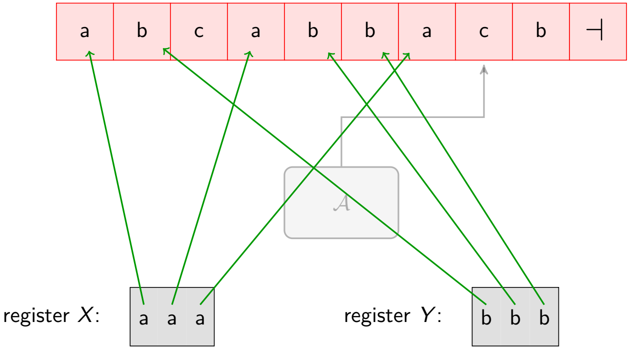
Streaming String Transducers (SST)

- 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

$$e.g.,\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



- Example:

$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

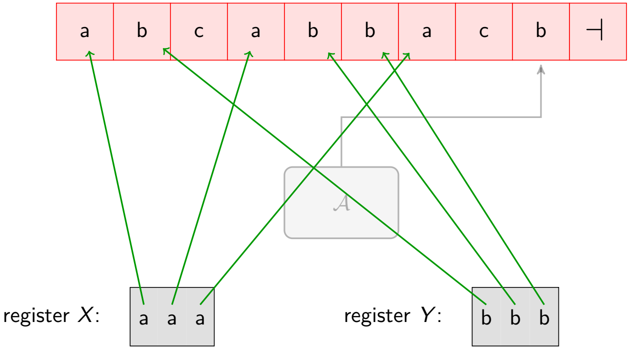
Streaming String Transducers (SST)

- 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., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

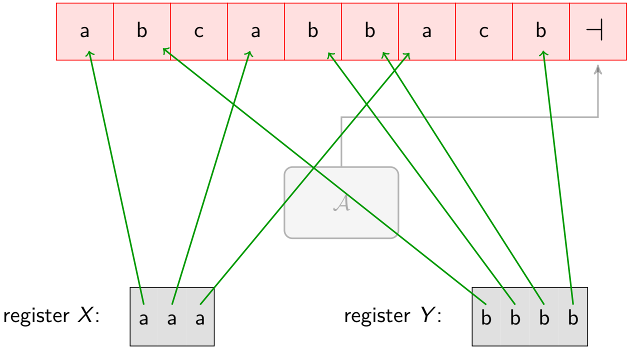
Streaming String Transducers (SST)

- 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

$$e.g.,\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

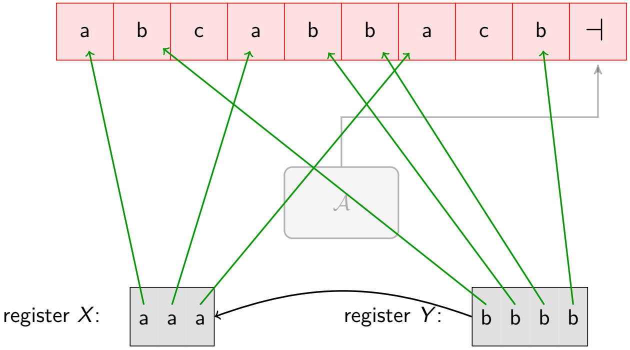
Streaming String Transducers (SST)

- 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., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

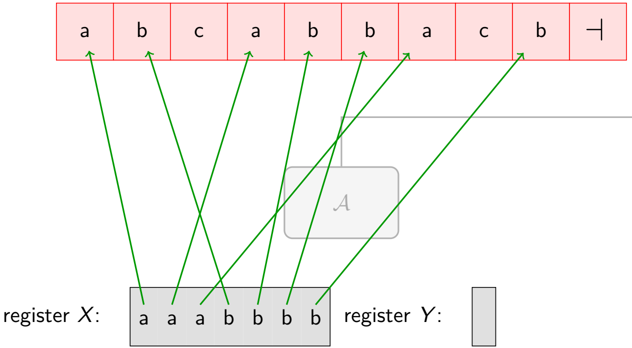
Streaming String Transducers (SST)

- 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

$$e.g.,\ \begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases},\ \begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases},\ \begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases},\ \cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}.$$



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

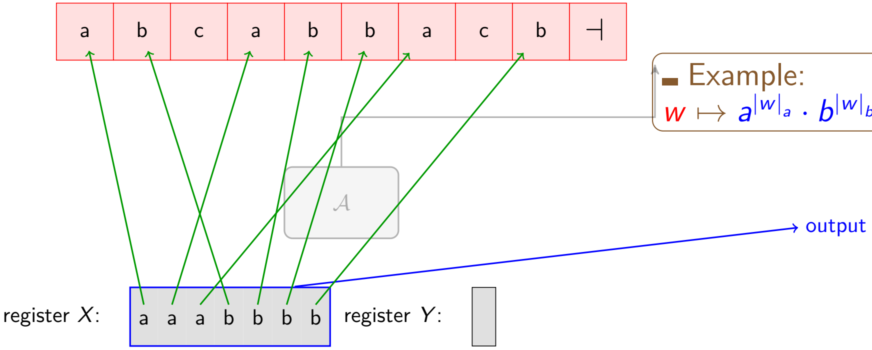
Streaming String Transducers (SST)

- 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., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.



Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

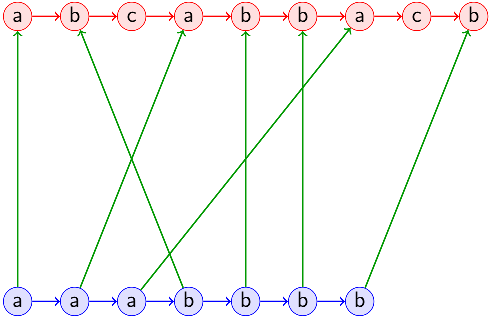
Streaming String Transducers (SST)

- a 1-way automaton $\mathcal{A}$
- a finite set R of registers including a distinguished output register

e.g., $R = \{X, Y\}$

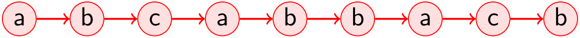
- a labelling of transitions by copyless register updates

e.g., $\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow \varepsilon \end{cases}$, $\begin{cases} X \leftarrow a \cdot Y \\ Y \leftarrow b \cdot a \cdot X \end{cases}$, $\begin{cases} X \leftarrow b \\ Y \leftarrow X \cdot a \cdot Y \end{cases}$, $\cancel{\begin{cases} X \leftarrow X \cdot a \\ Y \leftarrow X \cdot Y \end{cases}}$.



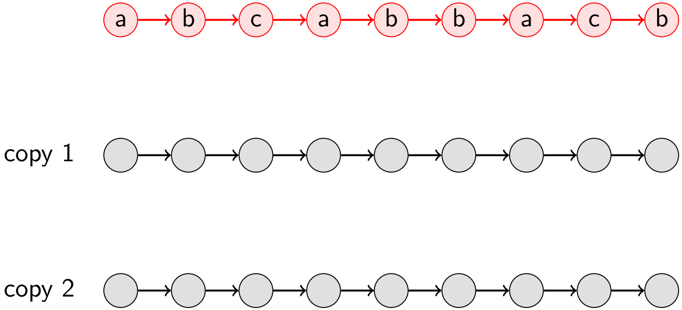
Example:
$w \mapsto a^{|w|_a} \cdot b^{|w|_b}$

string-to-string MSO-transduction [Courcelle 1991]



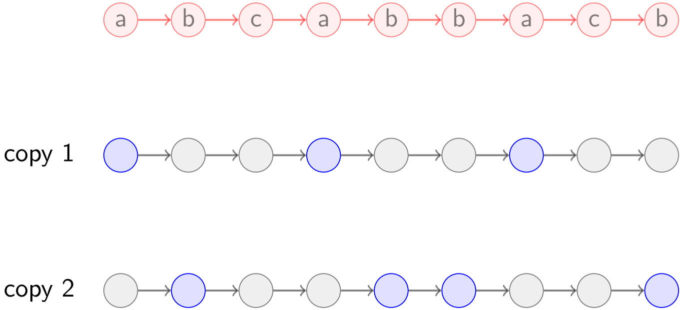
string-to-string MSO-transduction [Courcelle 1991]

- Nondeterministic MSO-colouring (nondeterministic case);
- Copy (finitely many copies of the input);



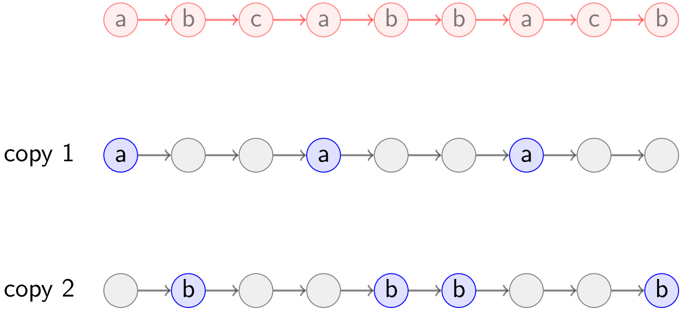
string-to-string MSO-transduction [Courcelle 1991]

- 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.



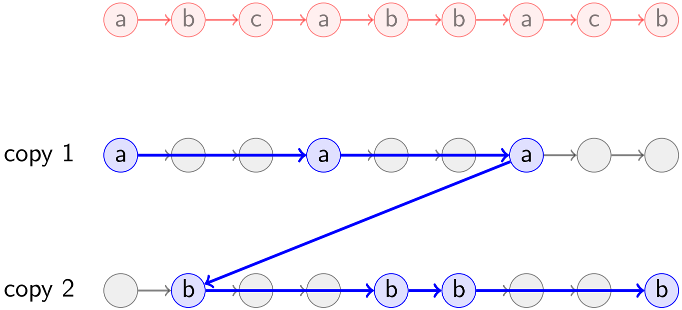
string-to-string MSO-transduction [Courcelle 1991]

- 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.



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.



string-to-string MSO-transduction [Courcelle 1991]

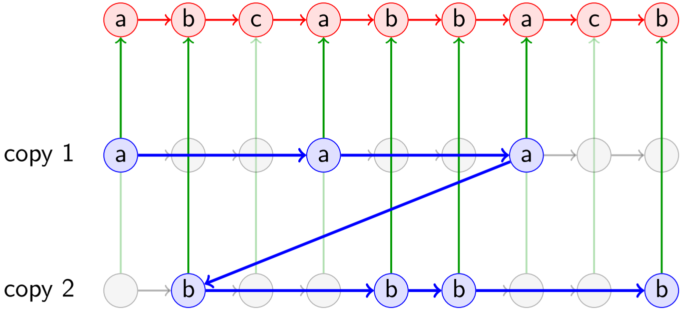
- Nondeterministic MSO-colouring (nondeterministic case);
- Copy (finitely many copies of the input);
- MSO-Interpretation
  - a formula for restricting the universe;
  - a formula for each predicate of the output vocabulary.

Definition (origin semantics of MSO-transduction)

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

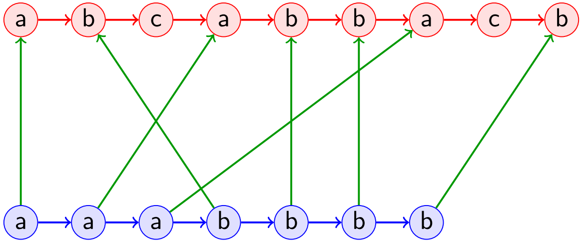
string-to-origin graph MSO-transduction

- 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.



string-to-origin graph MSO-transduction

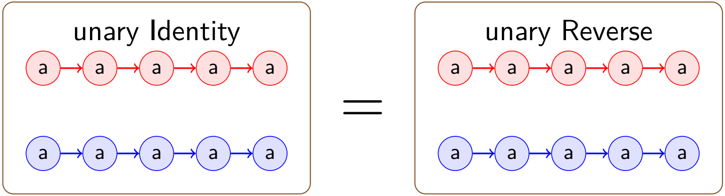
- Nondeterministic MSO-colouring (nondeterministic case);
- Copy (finitely many copies of the input);
- MSO-Interpretation
  - a formula for restricting the universe;
  - a formula for each predicate of the output vocabulary.



What do we get from origin information?

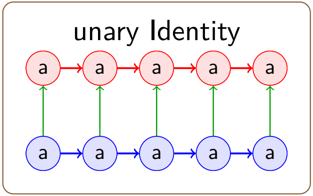
Origin semantics is thinner grained

Examples

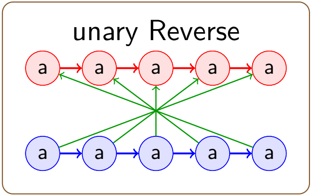


Origin semantics is thinner grained

Examples



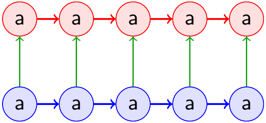
$\neq$



Origin semantics is thinner grained

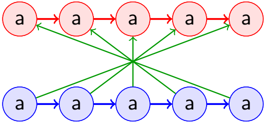
Examples

unary Identity

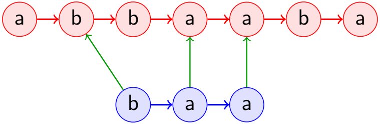


$\neq$

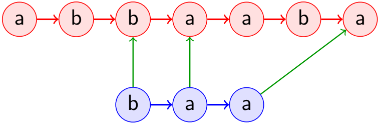
unary Reverse



Subword



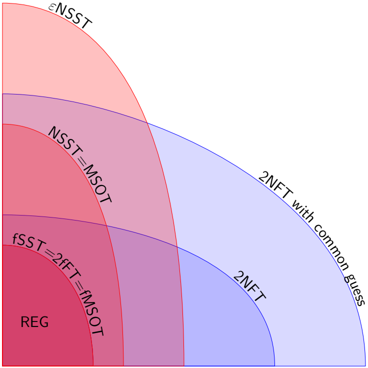
$\neq$



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

MSO satisfiability on origin graphs

Theorem: The following is **decidable** :

Input

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

Question

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

origin vocabulary: binary predicates $\longrightarrow$, $\longrightarrow$, $\longrightarrow$ and labelling in $\Sigma \cup \Gamma$;

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."

$\textsc{mso}$ satisfiability on origin graphs

Theorem: The following is decidable :

Input

- an $\textsc{nsst}$ $\mathcal{A}$
- an $\textsc{mso}$ formula ϕ over the corresponding origin vocabulary

Question

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

Proof: Let $\mathcal{A}$ and ϕ be fixed.

- there is a string-to-origin graph $\textsc{mso}$-transduction ρ equivalent to $\mathcal{A}$

MSO satisfiability on origin graphs

Theorem: The following is **decidable**:

Input

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

Question

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

Proof: Let $\mathcal{A}$ and ϕ be fixed.

- there is a string-to-origin graph MSO-transduction ρ equivalent to $\mathcal{A}$
- we consider $\mathcal{G} = \{G$ origin graph $\mid \phi$ is true over $G\}$

MSO satisfiability on origin graphs

Theorem: The following is decidable :

Input

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

Question

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

Proof: Let $\mathcal{A}$ and ϕ be fixed.

- there is a string-to-origin graph MSO-transduction ρ 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. $\square$

Which properties of origin graphs

characterise

regular sets of origin graphs?

Which sets of origin graphs are generated by transducers?

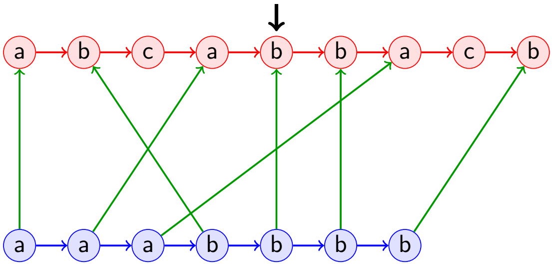
Theorem:

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

- **mso-definable**:
  an MSO sentence using $\longrightarrow$, $\longrightarrow$, $\longrightarrow$ 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**: NEXT SLIDE.

Crossing

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

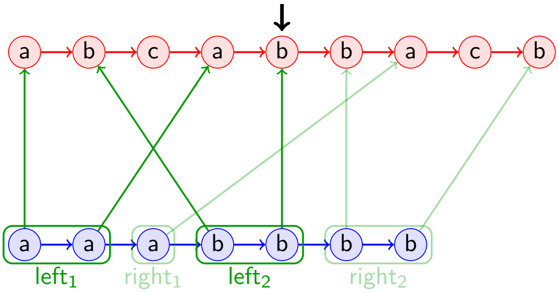


Crossing

- crossing of an input position

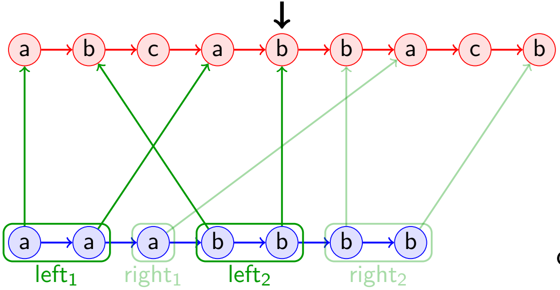
  number of maximal factors of the output

  that originate in the input prefix ended by the position



Crossing

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



crossing: 2

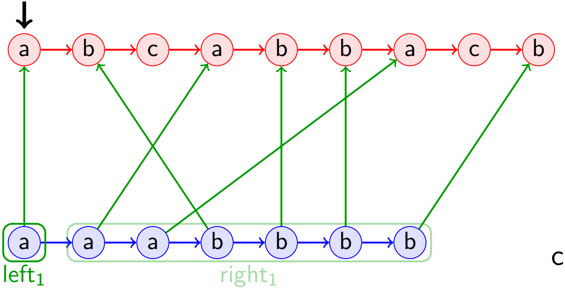
Crossing

- 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: 1

(≡ particular path decomposition of bounded width)

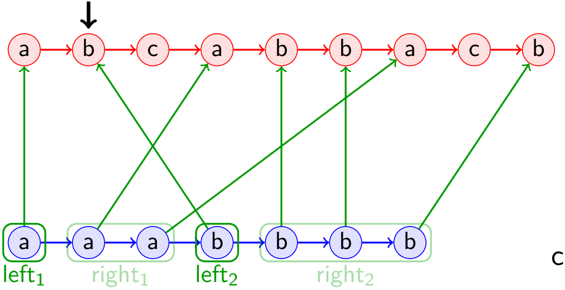
Crossing

- 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)

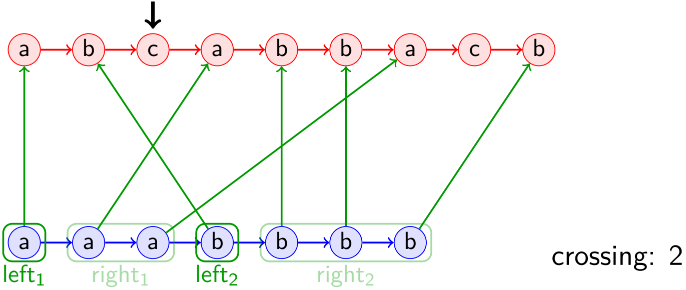
Crossing

- crossing of an input position

  number of maximal factors of the output

  that originate in the input prefix ended by the position

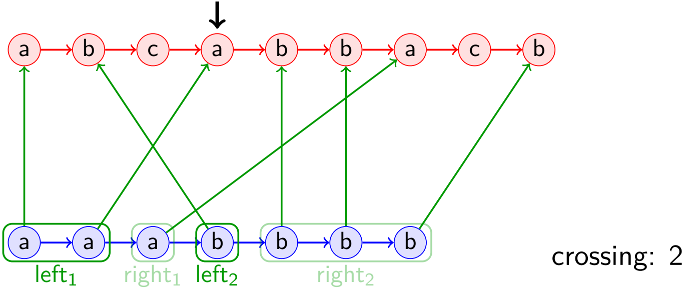
- crossing of an origin graph: max of the crossings



(≡ 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



(≡ particular path decomposition of bounded width)

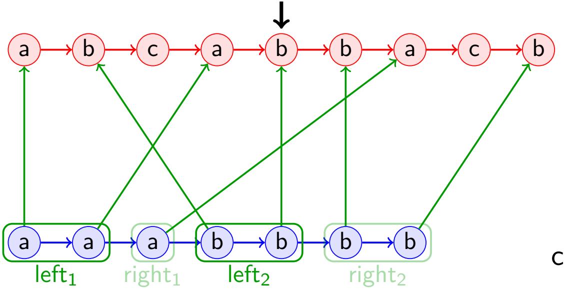
Crossing

- 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

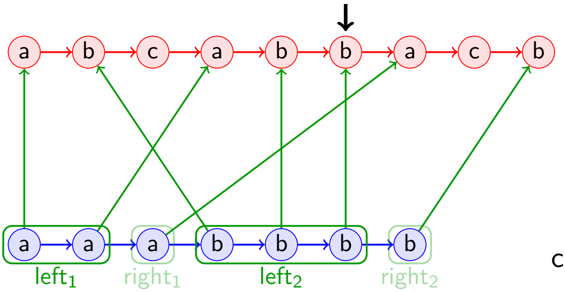


crossing: 2

(≡ particular path decomposition of bounded width)

Crossing

- 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

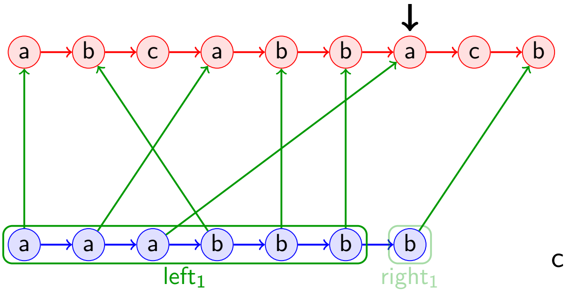


crossing: 2

(≡ particular path decomposition of bounded width)

Crossing

- 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

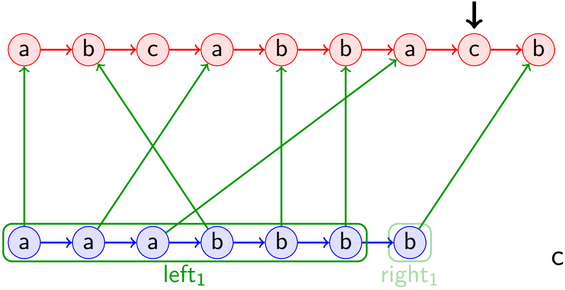


crossing: 1

($\equiv$ 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



crossing: 1

(≡ particular path decomposition of bounded width)

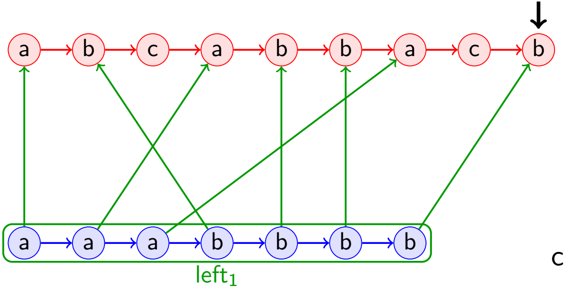
Crossing

- 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 : 2



crossing: 1

(≡ particular path decomposition of bounded width)

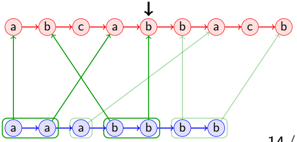
Which sets of origin graphs are generated by transducers?

Theorem:

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

- **mso-definable**:
  an MSO sentence using $\longrightarrow$, $\longrightarrow$, $\longrightarrow$ 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**:

PREVIOUS SLIDE



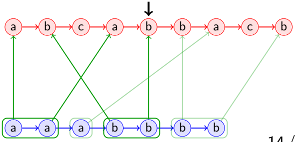
Which sets of origin graphs are generated by transducers?

Theorem:

A set of origin graphs is realised by an unambiguous k-registers SST if and only if it is

- **mso-definable**: an MSO sentence using $\color{red}{\longrightarrow}$, $\color{blue}{\longrightarrow}$, $\color{green}{\longrightarrow}$ and labelling in $\color{red}{\Sigma} \cup \color{blue}{\Gamma}$;
- **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**:

PREVIOUS SLIDE



Which sets of origin graphs are generated by transducers?

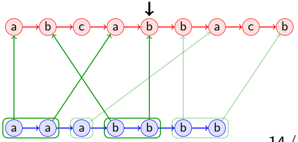
Theorem:

A set of origin graphs is realised by an k-registers SST if and only if it is

- **mso-definable**:
  an MSO sentence using $\color{red}{\longrightarrow}$, $\color{blue}{\longrightarrow}$, $\color{green}{\longrightarrow}$ and labelling in $\color{red}{\Sigma} \cup \color{blue}{\Gamma}$;

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

PREVIOUS SLIDE

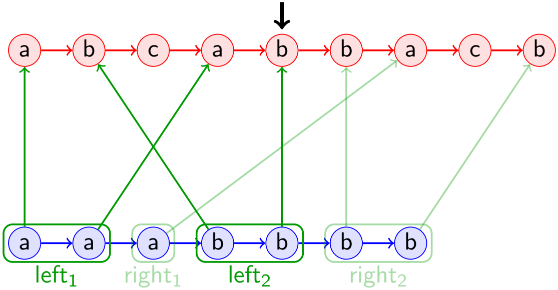


Sketch of the proof $\Longrightarrow$

- unambiguous $\Longrightarrow$ functional
- NSST $\Longrightarrow$ bounded degree

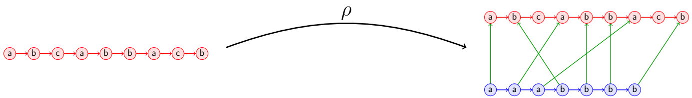
Sketch of the proof $\Longrightarrow$

- unambiguous $\Longrightarrow$ functional
- NSST $\Longrightarrow$ bounded degree
- k-register $\Longrightarrow$ crossing bounded by k



Sketch of the proof $\Longrightarrow$

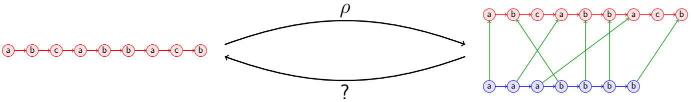
- unambiguous $\Longrightarrow$ functional
- NSST $\Longrightarrow$ bounded degree
- k-register $\Longrightarrow$ crossing bounded by k
- NSST $\Longrightarrow$ string-to-origin graph MSO-transduction



Sketch of the proof $\Longrightarrow$

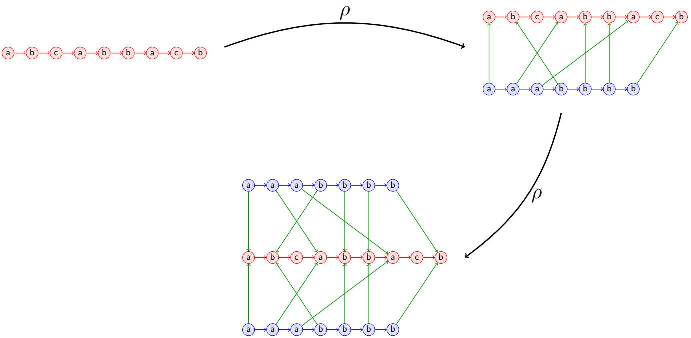
- unambiguous $\Longrightarrow$ functional
- NSST $\Longrightarrow$ bounded degree
- k-register $\Longrightarrow$ crossing bounded by k
- NSST $\Longrightarrow$ string-to-origin graph MSO-transduction

  Proposition: we can inverse this MSO-transduction



Sketch of the proof $\Longrightarrow$

- unambiguous $\Longrightarrow$ functional
- NSST $\Longrightarrow$ bounded degree
- k-register $\Longrightarrow$ crossing bounded by k
- NSST $\Longrightarrow$ string-to-origin graph MSO-transduction
  Proposition: we can inverse this MSO-transduction

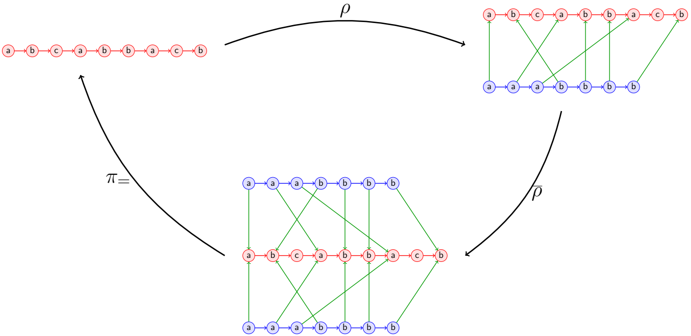


Sketch of the proof $\Longrightarrow$

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

  Proposition: we can inverse this MSO-transduction

  $\Longrightarrow$ MSO-definable



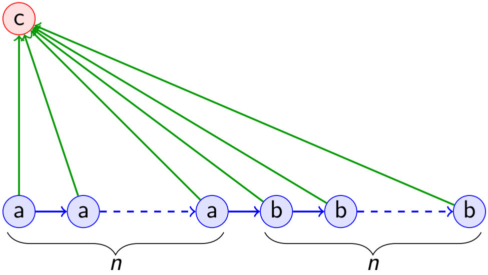
Sketch of the proof $\Longrightarrow$

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

  Proposition: we can inverse this MSO-transduction

  $\Longrightarrow$ MSO-definable

Note: False when ε-transitions are allowed.



Sketch of the proof $\Longleftarrow$

- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

Sketch of the proof $\Longleftarrow$

- 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.

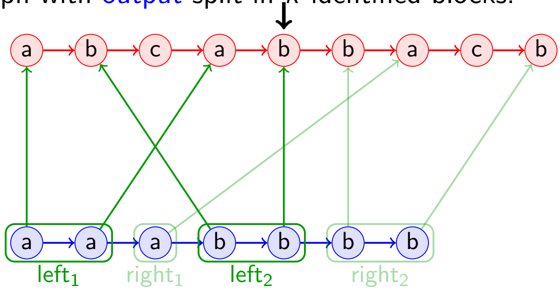
Sketch of the proof $\Longleftarrow$

- 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.



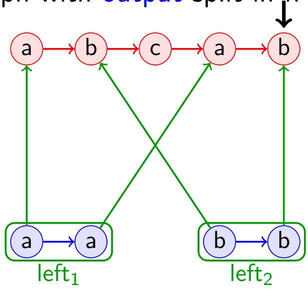
Sketch of the proof $\Longleftarrow$

- 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 $\Longleftarrow$

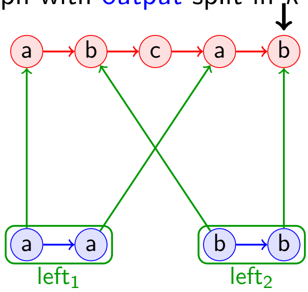
- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_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 $\Longleftarrow$

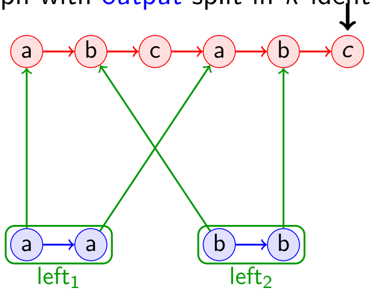
- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_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 $\Longleftarrow$

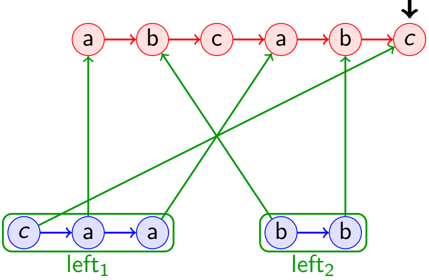
- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_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 $\Longleftarrow$

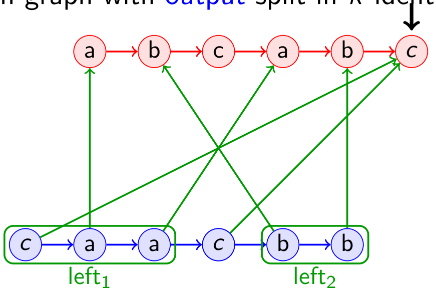
- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_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 $\Longleftarrow$

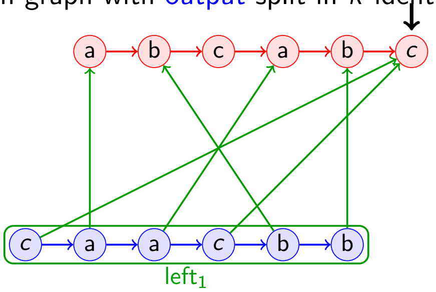
- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_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 $\Longleftarrow$

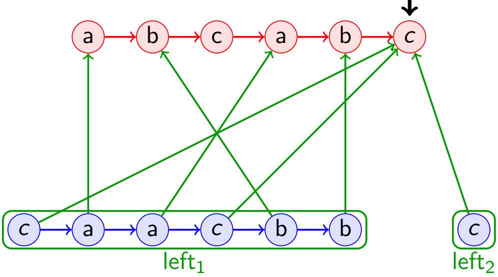
- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_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 $\Longleftarrow$

- Start with an MSO-**definable** set of origin graphs G

  with **crossing bounded by k**

- we define a finite set of (partial) operations Ω_k on k-BLOGs

- the **folding** of a word w over Ω_k^* is the k-BLOG $\alpha_k(w)$

  obtained from the empty graph by applying the operations.

  α_k can be realised by an MSO-transduction.

Sketch of the proof $\Longleftarrow$

- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_k on k-BLOGs

- the folding of a word w over Ω_k^* is the k-BLOG $\alpha_k(w)$

  obtained from the empty graph by applying the operations.

  α_k can be realised by an MSO-transduction.

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

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

Sketch of the proof $\Longleftarrow$

- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_k on k-BLOGs

- the folding of a word w over Ω_k^* is the k-BLOG $\alpha_k(w)$

  obtained from the empty graph by applying the operations.

  α_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 ε-transitions realising G

Sketch of the proof $\Longleftarrow$

- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_k on k-BLOGs

- the folding of a word w over Ω_k^* is the k-BLOG $\alpha_k(w)$

  obtained from the empty graph by applying the operations.

  α_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 ε-transitions realising G

  - if bounded degree $\Longrightarrow$ elimination of ε-transition

Sketch of the proof $\Longleftarrow$

- Start with an MSO-definable set of origin graphs G

  with crossing bounded by k

- we define a finite set of (partial) operations Ω_k on k-BLOGs

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

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

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

- from an automaton recognising L,
  we build a NSST with ε-transitions realising G
  - if bounded degree $\Longrightarrow$ elimination of ε-transition
  - if functional $\Longrightarrow$ disambiguation

Equivalence

Corollary: The following is decidable:

Input

- Two NSST, $\mathcal{A}$ and $\mathcal{B}$.

Question

- Whether they have the same origin semantics.

Equivalence

Corollary: The following is decidable:

Input

- Two NSST, $\mathcal{A}$ and $\mathcal{B}$.

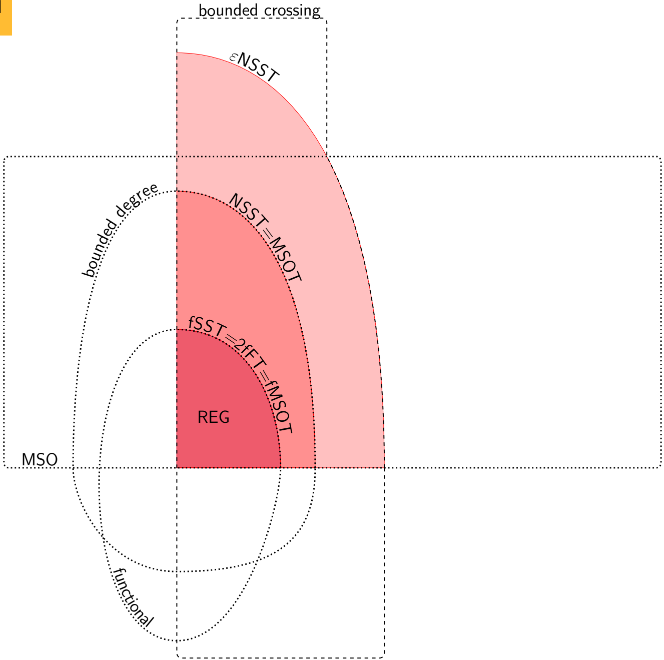
Question

- Whether they have the same origin semantics.

Proof: We show that we can check whether $\mathcal{A} \cap \bar{\mathcal{B}}$ is empty.

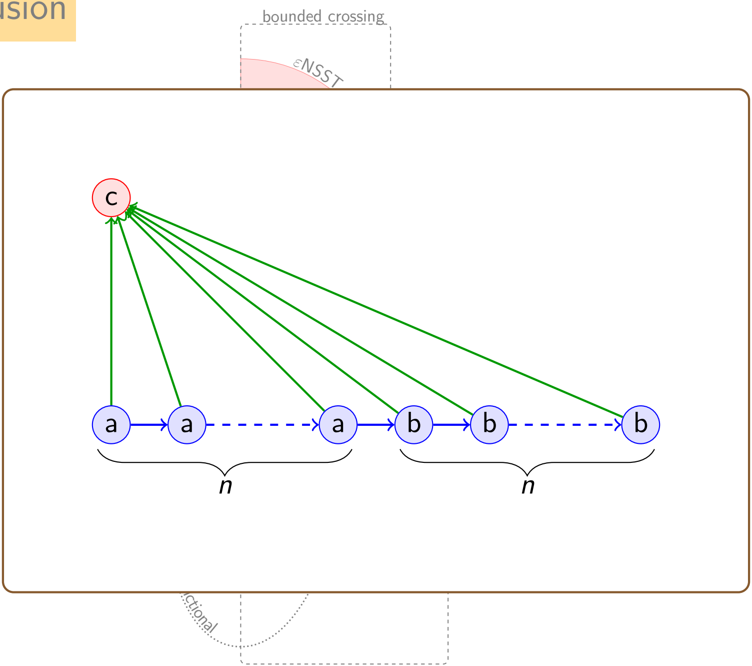
- The origin semantics of $\mathcal{B}$ is MSO-definable by a formula ϕ,
- We can check whether $\neg\phi$ is true in some origin graph in the origin semantics of $\mathcal{A}$.

Conclusion

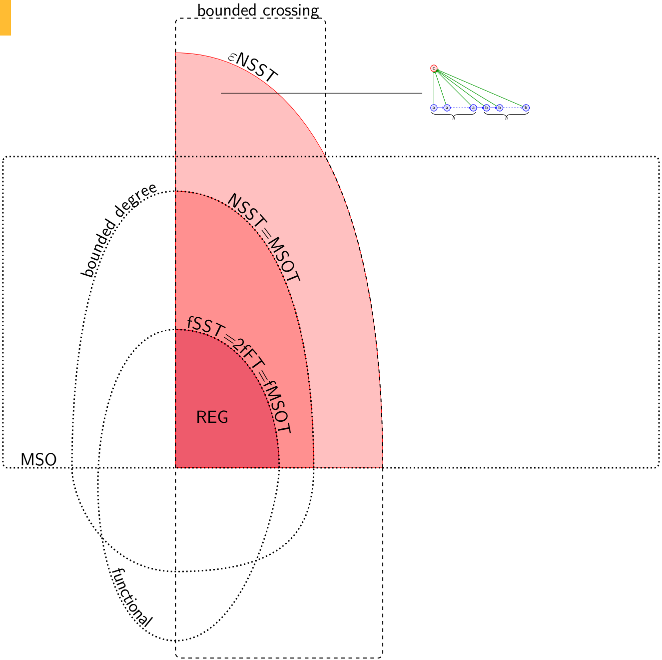


bounded crossing
εNSST
bounded degree
NSST=MSOT
fSST=2fFT=fMSOT
REG
MSO
functional

Conclusion

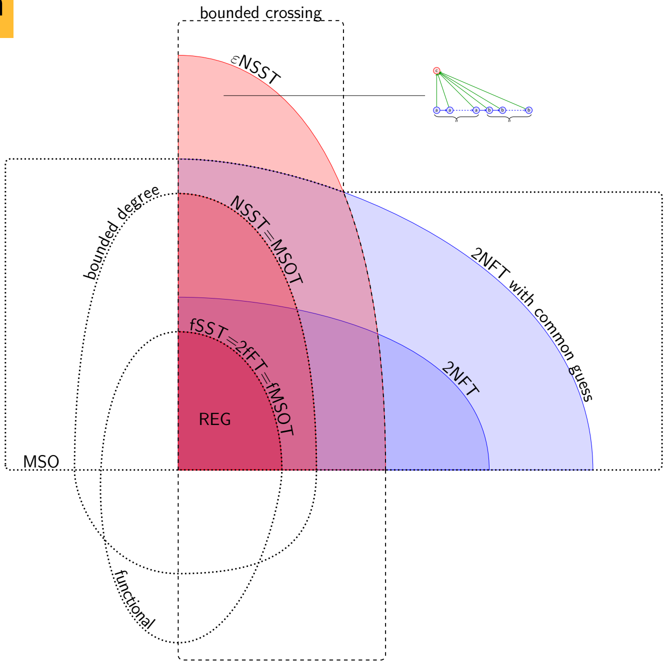


Conclusion



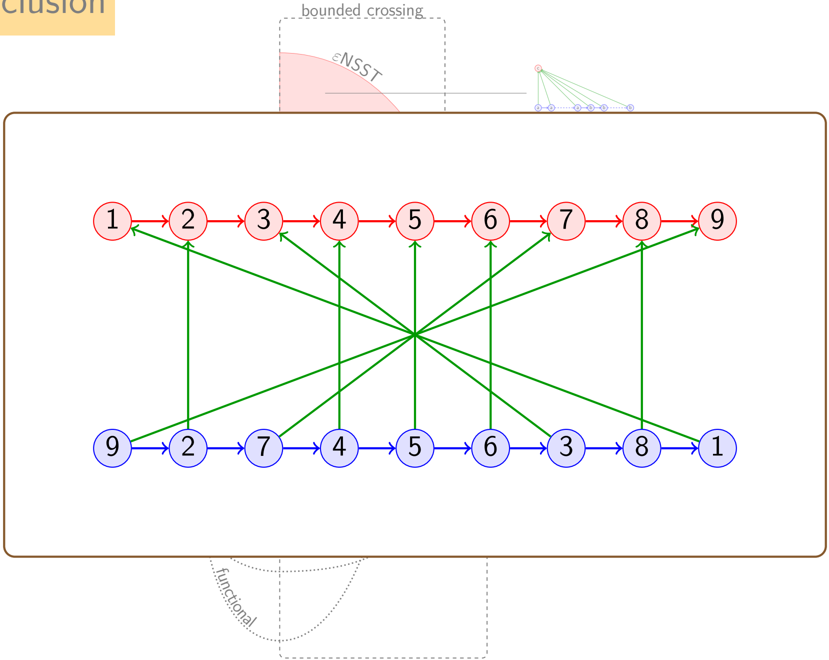
bounded crossing
εNSST
MSO
bounded degree
NSST=MSOT
fSST=2fFT=fMSOT
REG
functional

Conclusion

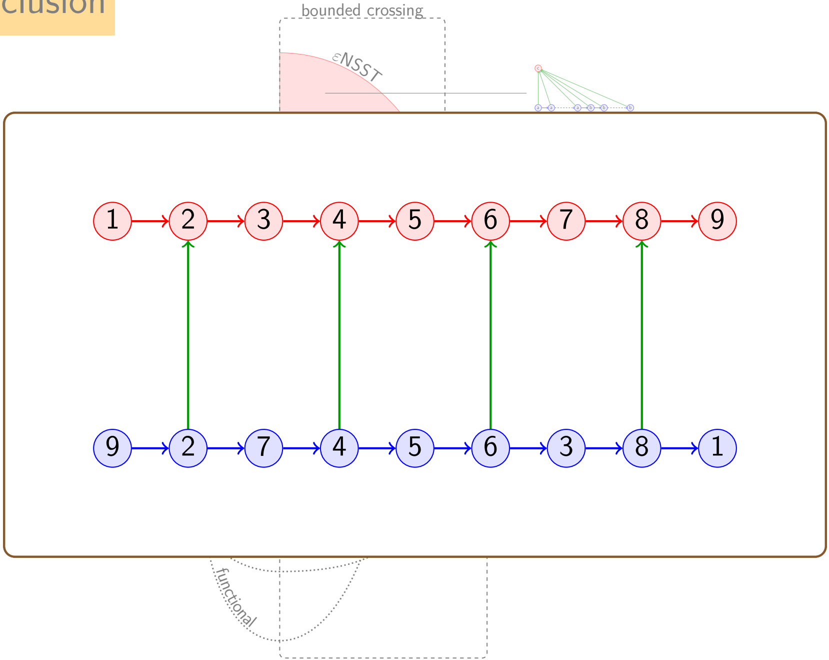


bounded crossing
εNSST
bounded degree
NSST=MSOT
2NFT with common guess
fSST=2fFT=fMSOT
2NFT
REG
MSO
functional

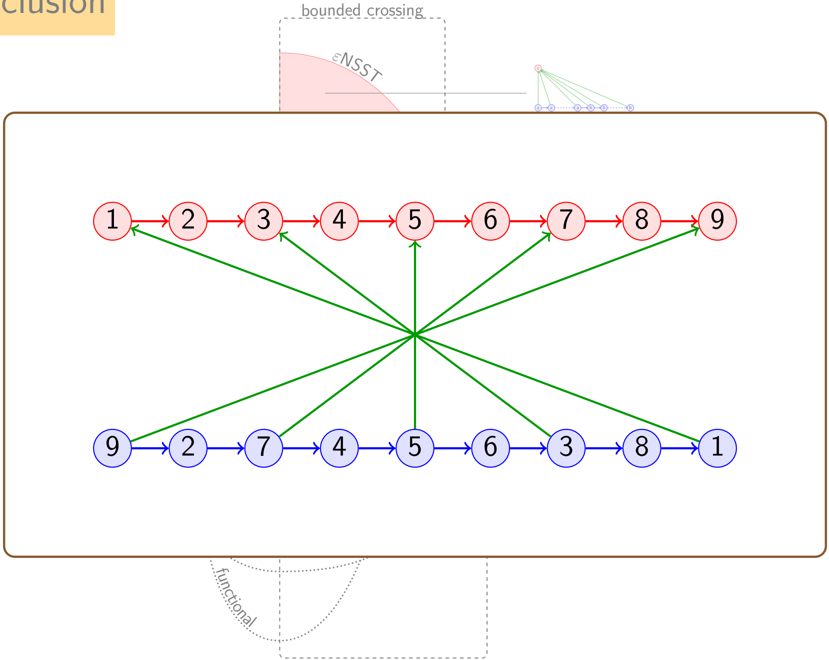
Conclusion



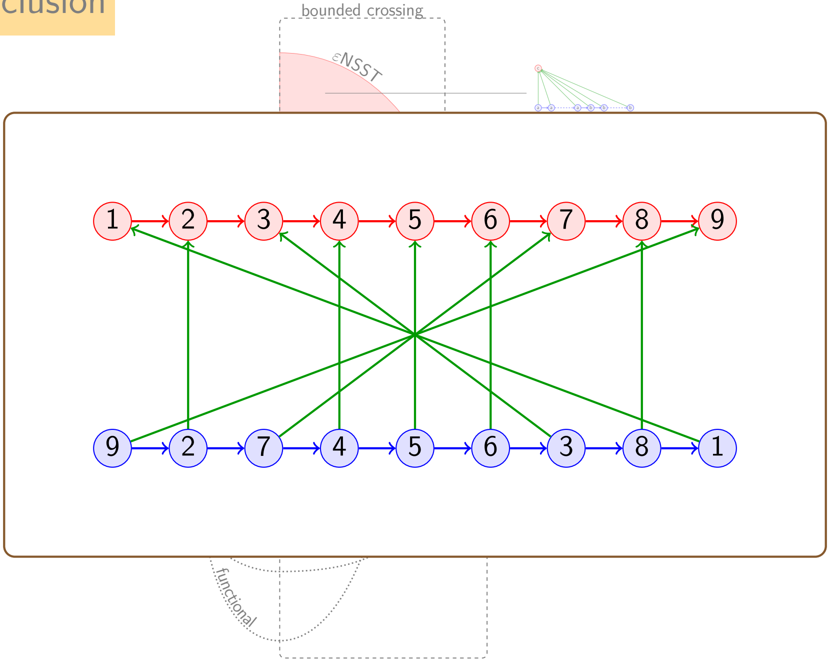
Conclusion



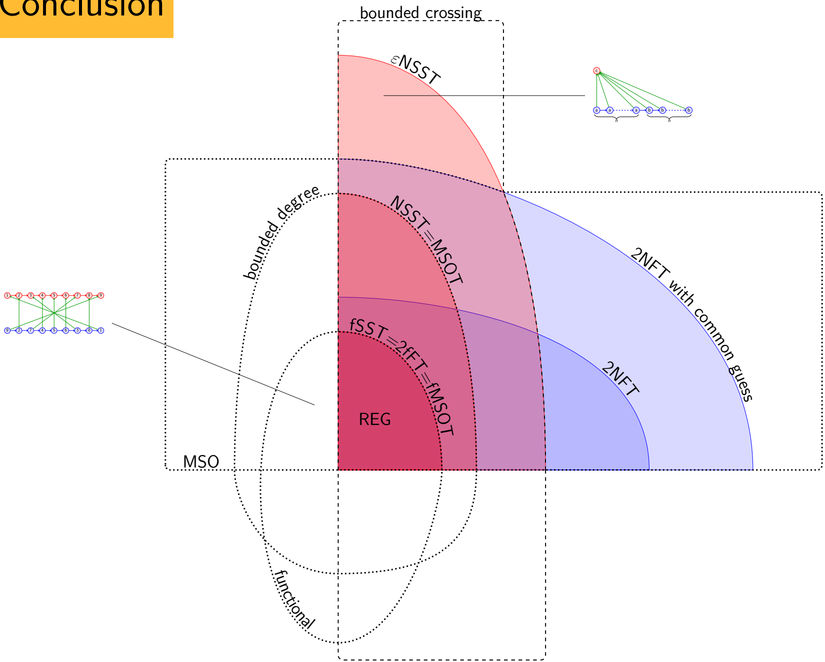
Conclusion



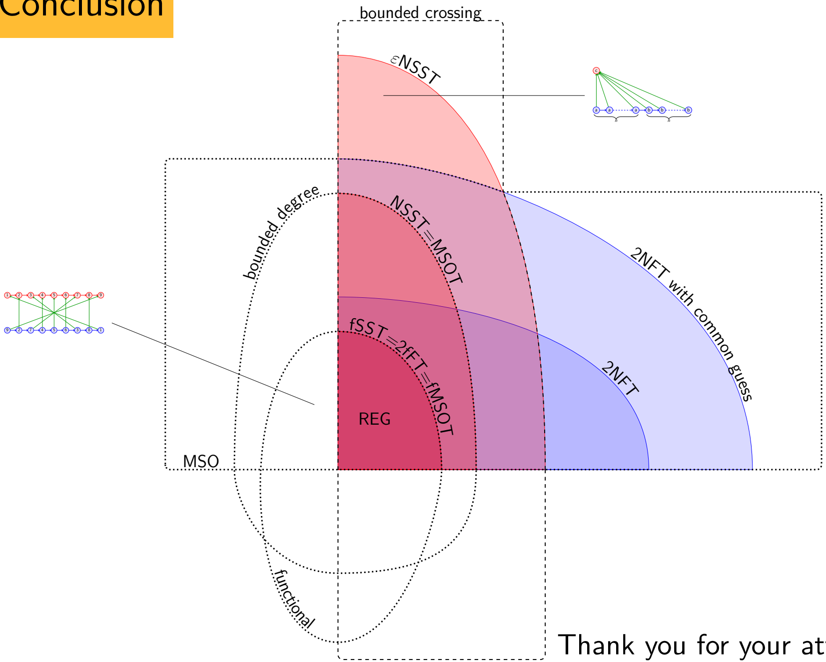
Conclusion



Conclusion


bounded crossing
εNSST
NSST=MSOT
fSST=2fFT=fMSOT
REG
2NFT
2NFT with common guess
bounded degree
MSO
functional

Conclusion



Thank you for your attention.