First-Order Transformations of Finite Words

Emmanuel Filiot	ULB-FNRS
Krishna Shankara Narayanan	IIT Bombay
A. Trivedi	IIT Bombay

Highlights of Logic, Games and Automata, 2014

Overview

- Σ : finite alphabet

Theorem (Engelfriet, Hoogeboom, 01)

A function $f : \Sigma^* \to \Sigma^*$ is (Courcelle) MSO-definable iff it is definable by a deterministic two-way transducer.

Overview

- Σ : finite alphabet

Theorem (Engelfriet, Hoogeboom, 01)

A function $f : \Sigma^* \to \Sigma^*$ is (Courcelle) MSO-definable iff it is definable by a deterministic two-way transducer.

Theorem (Alur, Cerny, 10)

A function $f : \Sigma^* \to \Sigma^*$ is (Courcelle) MSO-definable iff it is definable by a streaming string transducer (SST).

Overview

- Σ : finite alphabet

Theorem (Engelfriet, Hoogeboom, 01)

A function $f : \Sigma^* \to \Sigma^*$ is (Courcelle) MSO-definable iff it is definable by a deterministic two-way transducer.

Theorem (Alur, Cerny, 10)

A function $f : \Sigma^* \to \Sigma^*$ is (Courcelle) MSO-definable iff it is definable by a streaming string transducer (SST).

Theorem (Main result of this talk)

A function $f : \Sigma^* \to \Sigma^*$ is (Courcelle) **FO**-*definable iff it is definable by an* **aperiodic** *streaming string transducer.*

Examples of Transformations

- f_{del}: delete all 'a' positions

$$abbabaa \mapsto bbb$$

- f_{rev}: reverse the input word

$$stressed \mapsto desserts$$

- f_{halve}: maps all inputs a^n to $a^{\lfloor \frac{n}{2} \rfloor}$.

$$a^5 \mapsto a^2$$

- f_{copy}: copy the input word twice

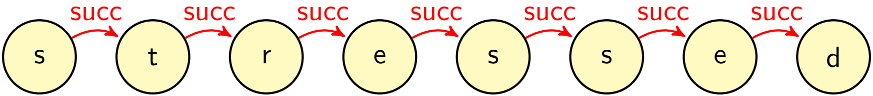
$$ab\# \mapsto ab\#ab\#$$

(Courcelle) MSO Transformations

- words as a structures over $\{succ, (lab_a)_{a \in \Sigma}\}$
- output predicates defined by MSO formulas interpreted over the input structure

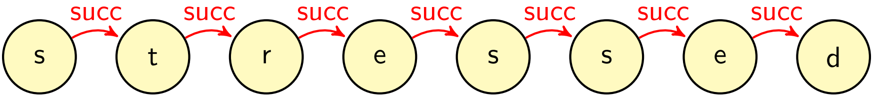
(Courcelle) MSO Transformations

- words as a structures over $\{succ, (lab_a)_{a \in \Sigma}\}$
- output predicates defined by MSO formulas interpreted over the input structure



(Courcelle) MSO Transformations

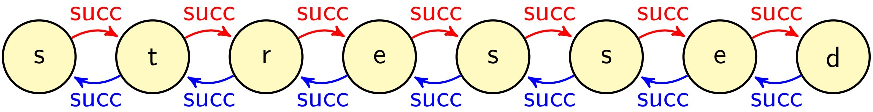
- words as a structures over $\{succ, (lab_a)_{a \in \Sigma}\}$
- output predicates defined by MSO formulas interpreted over the input structure



$$\begin{array}{rcl} \phi_{succ}(x, y) & \equiv & succ(y, x) \\ \phi_{lab_a}(x) & \equiv & lab_a(x) \end{array}$$

(Courcelle) MSO Transformations

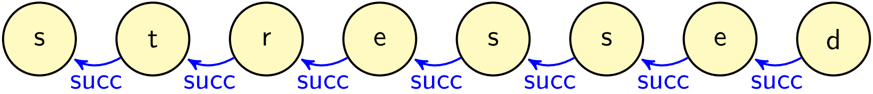
- words as a structures over $\{succ, (lab_a)_{a \in \Sigma}\}$
- output predicates defined by MSO formulas interpreted over the input structure



$$\phi_{succ}(x, y) \equiv succ(y, x)$$
$$\phi_{lab_a}(x) \equiv lab_a(x)$$

(Courcelle) MSO Transformations

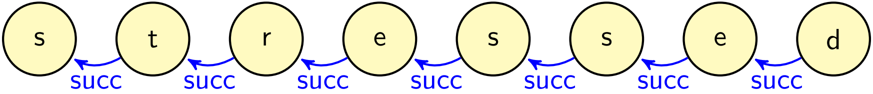
- words as a structures over $\{succ, (lab_a)_{a\in\Sigma}\}$
- output predicates defined by MSO formulas interpreted over the input structure



$$\begin{aligned} \phi_{succ}(x,y) &\equiv succ(y,x) \\ \phi_{lab_a}(x) &\equiv lab_a(x) \end{aligned}$$

(Courcelle) MSO Transformations

- words as a structures over $\{succ, (lab_a)_{a\in\Sigma}\}$
- output predicates defined by MSO formulas interpreted over the input structure



$$
\begin{aligned}
\phi_{succ}(x, y) &\equiv succ(y, x) \\
\phi_{lab_a}(x) &\equiv lab_a(x)
\end{aligned}
$$

- more generally, input structure can be copied a fixed number of times ($w \mapsto ww$)
- **FO-transformations**: MSO replaced by FO over $\{\leq, (lab_a)_{a\in\Sigma}\}$.

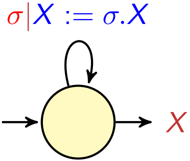
Streaming String Transducers (SST)

- one-way, deterministic model
- extend finite automata with a finite set of word variables $X, Y \dots$
  - appending a word u: $X := Xu$
  - prepending a word: $X := uX$
  - concatenating two variables: $X := YZ$

Streaming String Transducers (SST)

- one-way, deterministic model
- extend finite automata with a finite set of word variables $X, Y \ldots$
  - appending a word u: $X := Xu$
  - prepending a word: $X := uX$
  - concatenating two variables: $X := YZ$

reverse :

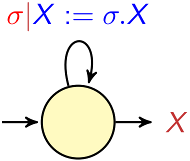


$\sigma \in \Sigma$

Streaming String Transducers (SST)

- one-way, deterministic model
- extend finite automata with a finite set of word variables $X, Y \ldots$
  - appending a word u: $X := Xu$
  - prepending a word: $X := uX$
  - concatenating two variables: $X := YZ$

reverse :



$\sigma \in \Sigma$

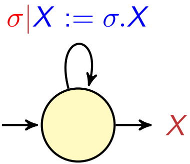
Theorem (Alur, Cerny, 10)

A function $f : \Sigma^* \to \Sigma^*$ is MSO-definable iff it is definable by an SST with copyless variable update.

Streaming String Transducers (SST)

- one-way, deterministic model
- extend finite automata with a finite set of word variables $X, Y \dots$
  - appending a word u: $X := Xu$
  - prepending a word: $X := uX$
  - concatenating two variables: $X := YZ$

reverse :



$\sigma \in \Sigma$

Theorem (Alur, Cerny, 10)

A function $f : \Sigma^* \to \Sigma^*$ is MSO-definable iff it is definable by an SST with copyless variable update.

Question: What restriction to put on SST to capture FO ?

Aperiodic Finite Automata

Among several characterizations of FO languages[1], we use the following:

Theorem

A language $L \subseteq \Sigma^*$ is FO-definable iff it is definable by an aperiodic finite automaton (AFA).

[1] *First-order definable languages*, V. Diekert and P. Gastin. 2007.

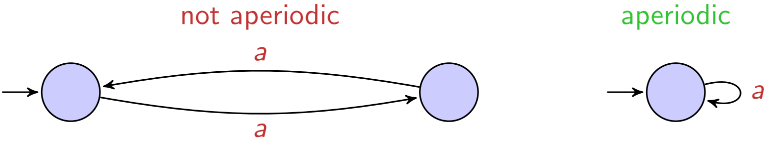
Aperiodic Finite Automata

Among several characterizations of FO languages[1], we use the following:

Theorem

A language $L \subseteq \Sigma^*$ is FO-definable iff it is definable by an aperiodic finite automaton (AFA).

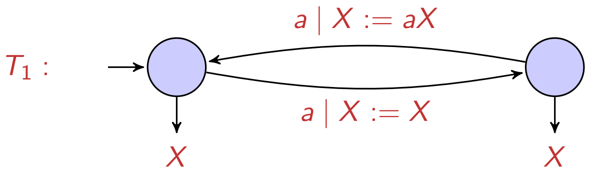
- AFA = finite automaton with aperiodic transition monoid $\mathcal{T}(A)$
- $\mathcal{T}(A) = \{M_w \mid w \in \Sigma^*\}$
- for any two states p, q, $M_w[p][q] = 1$ iff $p \rightsquigarrow^w q$.
- T_A is aperiodic if $\exists m \geq 0$, for all $M \in T_A$, $M^m = M^{m+1}$
- Examples:



[1] *First-order definable languages*, V. Diekert and P. Gastin. 2007.

Towards a restriction: $f_{halve} \;:\; a^n \;\mapsto\; a^{\lfloor \frac{n}{2} \rfloor}$ again

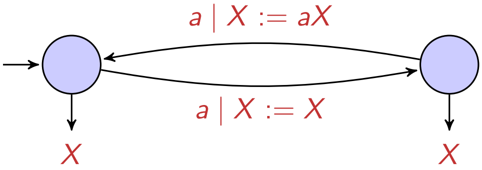
- not FO-definable
- definable by:



Towards a restriction: $f_{halve} \;:\; a^n \;\mapsto\; a^{\lfloor \frac{n}{2} \rfloor}$ again

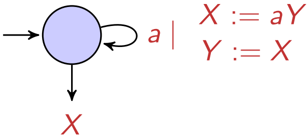
- not FO-definable
- definable by:

T_1 :



- aperiodicity of the underlying input automaton is **not sufficient**:

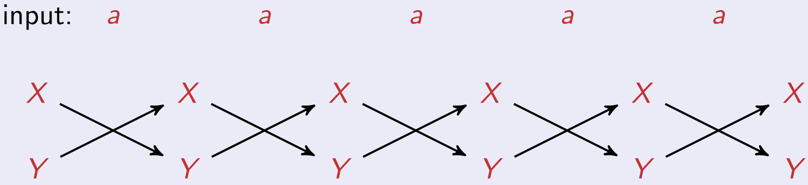
T_0 :



Variable flow

T_0 :

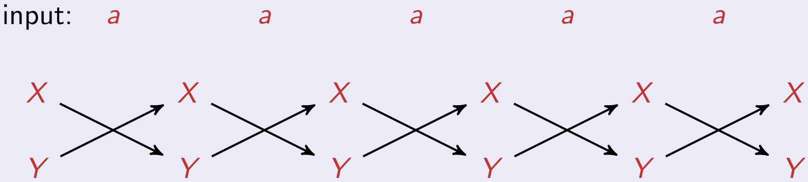
Dependency graph



Variable flow

T_0 :

Dependency graph



$\Rightarrow$ impose aperiodicity of the variable flow !

SST Transition Monoid

- set of Boolean matrices M_w indexed by pairs (q, X)
- coefficients in $\mathbb{N} \cup \{\perp\}$
- $M_w[p, X][q, Y] = \perp$ if there no run from p to q on w
- $M_w[p, X][q, Y] = n \in \mathbb{N}$ if
  - there is a run r from p to q on w
  - on this run, X "flows" n times to Y

SST Transition Monoid

- set of Boolean matrices M_w indexed by pairs (q, X)
- coefficients in $\mathbb{N} \cup \{\perp\}$
- $M_w[p, X][q, Y] = \perp$ if there no run from p to q on w
- $M_w[p, X][q, Y] = n \in \mathbb{N}$ if
  - there is a run r from p to q on w
  - on this run, X "flows" n times to Y
- Example:

$$q_0 \xrightarrow{a \left| \begin{array}{l} X := aXb \\ Y := bY \end{array} \right.} q_1 \xrightarrow{a \left| \begin{array}{l} X := YY \\ Y := \epsilon \end{array} \right.} q_2$$

Then $M_{aa}[q_0, Y][q_2, X] = 2$.

Results and Perspectives

Theorem

- *A function* $f : \Sigma^* \to \Sigma^*$ *is MSO-definable iff it is definable by a SST with* $\underline{\textit{finite transition monoid}}$*.*
- *A function* $f : \Sigma^* \to \Sigma^*$ *is FO-definable iff it is definable by a SST with* $\underline{\textit{finite and aperiodic transition monoid}}$*.*

Results and Perspectives

Theorem

- *A function $f : \Sigma^* \to \Sigma^*$ is MSO-definable iff it is definable by a SST with finite transition monoid.*
- *A function $f : \Sigma^* \to \Sigma^*$ is FO-definable iff it is definable by a SST with finite and aperiodic transition monoid.*

Open question

Give an effective, machine-independent, characterisation of FOT.

Related to M. Bojanczyk's work on a weaker semantics (with origin).