A Generalised Twinning Property for Minimisation of Cost Register Automata

Laure Daviaud
LIP, ENS Lyon

Joint work with P-A.Reynier and J-M.Talbot
LIF, Aix-Marseille Université

LABRI, February 2016
A first example

Given a word \( w \in \{ a, b \}^* \), compute \( \{|w|_a, |w|_b\} \).
A first example

Given a word $w \in \{a, b\}^*$, compute $\{|w|_a, |w|_b\}$.
A first example

Given a word $w \in \{a, b\}^*$, compute $\{|w|_a, |w|_b\}$.
A first example

Given a word $w \in \{a, b\}^*$, compute $\{|w|_a, |w|_b\}$.
Given a word $w \in \{a, b\}^*$, compute $\{|w|_a, |w|_b\}$.

A first example

<table>
<thead>
<tr>
<th>Transition</th>
<th>$a$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transition</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>Transition</td>
<td>0</td>
<td>+0</td>
</tr>
<tr>
<td>Transition</td>
<td>+1</td>
<td>+1</td>
</tr>
</tbody>
</table>

$\text{union}$

$a: \{X_a := X_a + 1, X_b := X_b\}$

$b: \{X_a := X_a, X_b := X_b + 1\}$
Given a word $w \in \{a, b\}^*$, compute $\{|w|_a, |w|_b\}$.

Question: How many values do we need to keep in memory?
Weighted automata (in a restricted case)

Non deterministic finite automaton whose transitions are weighted by elements of a semiring \((\mathcal{P}_f(G), \cup, \cdot)\), with \((G, \cdot)\) a group

Weight of a run \(\rho\): \(\omega(\rho) = \) product of the weights of the transitions

Function: \(w \mapsto \{\omega(\rho) \mid \rho \text{ accepting run labelled by } w\}\)

\[
\begin{array}{c}
\text{Weighted automata (in a restricted case)} \\
\end{array}
\]
Weighted automata (in a restricted case)

Non deterministic finite automaton whose transitions are weighted by elements of a semiring \((\mathcal{P}_f(G), \cup, \cdot)\), with \((G, \cdot)\) a group + an function \(t\) from the final states to \(G\).

*Weight of a run \(\rho\): \(\omega(\rho) = \) product of the weights of the transitions

*Function: \(w \mapsto \{\omega(\rho)t(q) \mid \rho \text{ accepting run labelled by } w \text{ ending in } q\}\)

![Automaton Diagram]

\(W(w) = \{\|w\| \mid \text{if } w \text{ ends with an } a\}\)

\(W(w) = \{\|w\| \|b\| \mid \text{if } w \text{ ends with a } b\}\)

\((\mathbb{Z}, +)\)
Weighted automata (in a restricted case)

Non deterministic finite automaton whose transitions are weighted by elements of a semiring \( \left( \mathcal{P}_f(G), \cup, \cdot \right) \), with \((G, \cdot)\) a group + an function \( t \) from the final states to \( G \).

**Weight of a run** \( \rho \): \( \omega(\rho) = \) product of the weights of the transitions

**Function**: \( w \mapsto \{ \omega(\rho) t(q) \mid \rho \text{ accepting run labelled by } w \text{ ending in } q \} \)

\[
\begin{align*}
\mathbb{W}(w) &= \{|w|_a, |w|_a + 1\} \\
&\quad \text{if } w \text{ ends with an } a \\
\mathbb{W}(w) &= \{|w|_b\} \\
&\quad \text{if } w \text{ ends with a } b
\end{align*}
\]
Deterministic finite state machine with registers + an output function

Register updates: $X := Y \alpha$ with $\alpha \in G$.  

![Diagram of a deterministic finite state machine with registers](image)
A group $G$ is said to be **infinitary** if for all $\alpha, \beta, \gamma \in G$ such that $\alpha \beta \gamma \neq \beta$, the set $\{\alpha^n \beta \gamma^n \mid n \in \mathbb{N}\}$ is infinite.

- $(\mathbb{Z}, +)$, $(\mathbb{R}, \times)$ are infinitary.
- The free group generated by a finite set is infinitary.
Infinitary group

Definition
A group $G$ is said to be infinitary if for all $\alpha, \beta, \gamma \in G$ such that $\alpha \beta \gamma \neq \beta$, the set $\{\alpha^n \beta \gamma^n \mid n \in \mathbb{N}\}$ is infinite.

- $(\mathbb{Z}, +)$, $(\mathbb{R}, \times)$ are infinitary.
- The free group generated by a finite set is infinitary.

Why infinitary group ?... See later!
Valuedness (of a function)

Ambiguity (of a weighted automaton)

Register complexity (of a function)
Several notions...

Valuedness (of a function)

- $\ell$-valued: for all words $w$, $|f(w)| \leq \ell$
- we only consider finite-valued functions

Ambiguity (of a weighted automaton)

Register complexity (of a function)
Several notions...

**Valuedness (of a function)**

- $\ell$-valued: for all words $w$, $|f(w)| \leq \ell$
- we only consider finite-valued functions

**Ambiguity (of a weighted automaton)**

- number of accepting paths labelled by a word in a weighted automaton
- $\ell$-valued $= \ell$-ambiguous [Filiot, Gentilini, Raskin]

**Register complexity (of a function)**

PSPACE-complete for one-valued additive register cost functions (unambiguous WA over $(\mathbb{Z}, +)$) [Alur, Raghothaman]
Several notions...

Valuedness (of a function)

- $\ell$-valued: for all words $w$, $|f(w)| \leq \ell$
- we only consider finite-valued functions

Ambiguity (of a weighted automaton)

- number of accepting paths labelled by a word in a weighted automaton
- $\ell$-valued $= \ell$-ambiguous [Filiot, Gentilini, Raskin]

Register complexity (of a function)

- minimal number of registers needed to compute a given function by a cost register automaton
- PSPACE-complete for one-valued additive register cost functions (unambiguous WA over $(\mathbb{Z}, +)$) [Alur, Raghothaman]
Over an infinitary group, characterise (effectively) the register complexity of a function computed by a finite-valued weighted automaton.
A very very simple example

\[ w \mapsto \{|w|_a, |w|_a + 1\} \text{ in } (\mathbb{Z}, +) \]
Given $\alpha, \beta \in G$, the **delay** between $\alpha$ and $\beta$ is $\alpha^{-1}\beta$. It is denoted by $\text{delay}(\alpha, \beta)$.

A very very simple example

\[ w \mapsto \{|w|_a, |w|_a + 1\} \text{ in } \mathbb{Z}, + \]
Twinning property [Choffrut]

Definition

A weighted automaton satisfies the **twinning property** if for all initial states $p, p'$ and co-accessible states $q, q'$, for all words $u, v$ such that:

$p \xrightarrow{u: \alpha} q \xrightarrow{v: \beta} q$

$p' \xrightarrow{u: \alpha'} q' \xrightarrow{v: \beta'} q'$

then $\text{delay}(\alpha, \alpha') = \text{delay}(\alpha\beta, \alpha'\beta')$

One register (cost register automata)  
= Deterministic (weighted automata)  
= TP (weighted automata)
Let $\mathcal{W}$ be a finite-valued weighted automaton over an infinitary group, and $k$ be a positive integer. The following assertions are equivalent:

- $\mathcal{W}$ satisfies the twinning property of order $k$,
- $\llbracket \mathcal{W} \rrbracket$ has register complexity $k$,
- $\llbracket \mathcal{W} \rrbracket$ satisfies the $k$-bounded variation property,

And everything is effective...
Twinning Property of order $k$

The weighted automaton satisfies the **twinning property of order** $k$ if for all $q_{0,j}$ initial and $q_{k,j}$ co-accessible such that:

there are $j \neq j'$ such that for all $i \in \{1, \ldots, k\}$,

$$\text{delay}(\alpha_{1,j} \cdots \alpha_{i,j}, \alpha_{1,j'} \cdots \alpha_{i,j'}) = \text{delay}(\alpha_{1,j} \cdots \alpha_{i,j} \beta_{i,j}, \alpha_{1,j'} \cdots \alpha_{i,j'} \beta_{i,j'})$$
Twinning property of order $k$

- Commutative case Vs non commutative case
- Decidability
- Infinitary here !!!
Twinning property of order $k$

If the twinning property is not satisfied $\rightarrow$ construction of a sequence of words that have $k + 1$ diverging behaviours [infinitarity + pumping the loops the right number of times]
Let $\mathcal{W}$ be a finite-valued weighted automaton over an infinitary group, and $k$ be a positive integer. The following assertions are equivalent:

- $\mathcal{W}$ satisfies the twinning property of order $k$,
- $\llbracket \mathcal{W} \rrbracket$ has register complexity $k$,
- $\llbracket \mathcal{W} \rrbracket$ satisfies the $k$-bounded variation property,

And everything is effective...
Bounded variation property

Notions of distance:

- On words: \( \text{dist}(u, v) = |u| + |v| - 2 \times |lcp(u, v)| \) where \( lcp(u, v) \) is the longest common prefix of the two words \( u \) and \( v \).

- On a finitely generated group \( G \) with a finite set of generators \( \Gamma \), \( d(\alpha, \beta) \) is the minimal length of a path linking \( \alpha \) and \( \beta \) in the undirected right Cayley graph of \( (G, \Gamma) \).
Bounded variation property

Notions of distance:

- On words: \( \text{dist}(u, v) = |u| + |v| - 2 \times |\text{lcp}(u, v)| \) where \( \text{lcp}(u, v) \) is the longest common prefix of the two words \( u \) and \( v \).

- On a finitely generated group \( G \) with a finite set of generators \( \Gamma \), \( d(\alpha, \beta) \) is the minimal length of a path linking \( \alpha \) and \( \beta \) in the undirected right Cayley graph of \( (G, \Gamma) \).

**Definition**

A function \( f : A^* \rightarrow \mathcal{P}_f(G) \) satisfies the \( k \)-**bounded variation property** if for all \( n \), there is \( N \) such that for all words \( w_0, \ldots, w_k \in A^* \) and all \( \alpha_0 \in f(w_0), \ldots, \alpha_k \in f(w_k) \), if for all \( 0 \leq i, j \leq k, \text{dist}(w_i, w_j) \leq n \) then there are \( 0 \leq i < j \leq k \) such that \( d(\alpha_i, \alpha_j) \leq N \).
Notions of distance:

- On words: \( \text{dist}(u, v) = |u| + |v| - 2 \times |lcp(u, v)| \) where \( lcp(u, v) \) is the longest common prefix of the two words \( u \) and \( v \).

- On a finitely generated group \( G \) with a finite set of generators \( \Gamma \), \( d(\alpha, \beta) \) is the minimal length of a path linking \( \alpha \) and \( \beta \) in the undirected right Cayley graph of \((G, \Gamma)\).

Definition

A function \( f : A^* \rightarrow P_f(G) \) satisfies the **\( k \)-bounded variation property** if for all \( n \), there is \( N \) such that for all words \( w_0, \ldots, w_k \in A^* \) and all \( \alpha_0 \in f(w_0), \ldots, \alpha_k \in f(w_k) \), if for all \( 0 \leq i, j \leq k \), \( \text{dist}(w_i, w_j) \leq n \) then there are \( 0 \leq i < j \leq k \) such that \( d(\alpha_i, \alpha_j) \leq N \).

- Generalisation of the bounded variation property of Choffrut.
- A machine-independent characterisation of TP\( k \).
- Finitely generated case Vs non finitely generated case.
### Hierarchy

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DET</td>
<td>Functional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 register</td>
<td>WA(TP(k)) RA((k)-reg)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\ell)</td>
<td>...</td>
<td>WA(TP(k), (\ell)-val) CRA((k)-reg, (\ell)-out)</td>
<td>WA((\ell)-val) CRA((\ell)-out)</td>
<td></td>
</tr>
</tbody>
</table>
Let $\mathcal{T}$ be an $\ell$-valued transducer from $A^*$ to $B^*$, and $k$ be a positive integer. The following assertions are equivalent:

- $\mathcal{T}$ satisfies the twinning property of order $k$,
- $\llbracket \mathcal{T} \rrbracket$ satisfies the $k$-bounded variation property,
- $\llbracket \mathcal{T} \rrbracket$ is computed by a cost register automaton over $B^*$ with $k$ registers and $\ell$ outputs.
Sketch of the proof

\( \mathcal{W} \) finite-valued weighted automaton over an infinitary group.

**First step:** \( \mathcal{W} \) satisfies the twinning property of order \( k \)
\[ \iff \lbrack \mathcal{W} \rbrack \text{ satisfies the } k\text{-bounded variation property} \]

**Second step:** \( \mathcal{W} \) satisfies the twinning property of order \( k \)
\[ \iff \lbrack \mathcal{W} \rbrack \text{ has register complexity } k \]

- Register complexity \( k \implies \text{Twinning property of order } k \)
- Twinning property of order \( k \implies \text{Register complexity } k \)
Conclusion and open questions

- $\text{TP}_k \iff \text{BV}_k \iff \text{Register complexity } k$
  for infinitary groups and transducers... (at least)
- Minimisation of cost register automata
  - Generalise the case of transducers
  - Generalisation to visibly pushdown