Max-plus automata and Tropical identities

Laure Daviaud
University of Warwick

Oxford, 01-02-2018
A natural and fundamental question:

\[
\mathcal{A}(u) = \mathcal{A}(v) \quad ?
\]
A natural and fundamental question:

Which pairs of inputs can be distinguished by a given computational model?
Given a class $C$ of computational models:
Given a class $C$ of computational models:

1. For all $u \neq v$, is there $A \in C$ which distinguishes $u$ and $v$?
Given a class $C$ of computational models:

1. For all $u \neq v$, is there $A \in C$ which distinguishes $u$ and $v$?

2. Is there $A \in C$ which distinguishes all pairs $u \neq v$?
Given a class $C$ of computational models:

1. For all $u \neq v$, is there $A \in C$ which distinguishes $u$ and $v$?

2. Is there $A \in C$ which distinguishes all pairs $u \neq v$?

3. Minimal size to distinguish two given input words?
Finite alphabet \( A = \{a, b\} \)
Set of words \( A^* \)
Finite alphabet $A = \{a, b\}$
Set of words $A^*$

Check if a word has at least two $b$'s.
Finite alphabet $A = \{a, b\}$
Set of words $A^*$

Check if a word has at least two $b$’s.
Finite alphabet $A = \{a, b\}$
Set of words $A^*$

Check if a word has at least two $b$’s.

A word is *accepted* by the automaton if there is a path labelled by the word from an initial state to a final state.
Boolean Automata

$[A] : A^* \rightarrow \{\text{Acc, Rej}\}$

For all $u \neq v$, is there $A \in C$ which distinguishes $u$ and $v$?

→ Yes

Is there $A \in C$ which distinguishes all pairs $u \neq v$?

→ No

Minimal size to distinguish two given input words?

→ Some lower bounds [Robson], Profinite theory...
Boolean Automata

\([\mathcal{A}] : A^* \rightarrow \{\text{Acc, Rej}\}\)

1. For all \(u \neq v\), is there \(\mathcal{A} \in C\) which distinguishes \(u\) and \(v\)?
   \(\rightarrow\) Yes
Boolean Automata

\[ [A] : A^* \rightarrow \{ \text{Acc, Rej} \} \]

1. For all \( u \neq v \), is there \( A \in \mathcal{C} \) which distinguishes \( u \) and \( v \)?
   → Yes

2. Is there \( A \in \mathcal{C} \) which distinguishes all pairs \( u \neq v \)?
   → No
Boolean Automata

$[\mathcal{A}] : A^* \rightarrow \{\text{Acc}, \text{Rej}\}$

For all $u \neq v$, is there $\mathcal{A} \in \mathcal{C}$ which distinguishes $u$ and $v$?

→ Yes

Is there $\mathcal{A} \in \mathcal{C}$ which distinguishes all pairs $u \neq v$?

→ No

Minimal size to distinguish two given input words?

→ Some lower bounds [Robson], Profinite theory...
Max-plus automata

Syntax: Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).

Semantic: Weight of a run: sum of the weights of the transitions.

\[ a, b : 0 \quad a : 1 \quad a, b : 0 \]

\[ a_n 0, b = b_{n+1} \quad b : 0 \quad b : 0 \]

\[ a + \rightarrow N \cup \{-\infty\} \quad w \mapsto \rightarrow \text{Max of the weights of the accepting runs labelled by } w \quad (\text{or } -\infty \text{ if no such run}) \]

\[ a_n 0, b = b_{n+1} \quad b : 0 \quad b : 0 \]

\[ \text{Max}(n_1, \ldots, n_k) \]
Max-plus automata

Syntax: Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).
Max-plus automata

Syntax: Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).

Semantic: Weight of a run: sum of the weights of the transitions.

\[ A^+ \rightarrow \mathbb{N} \cup \{-\infty\} \]

\[ w \mapsto \text{Max of the weights of the accepting runs labelled by } w \]

\[ (-\infty \text{ if no such run}) \]
Max-plus automata

Syntax: Non deterministic finite automaton for which each transition is labelled by a non negative integer (weight).

Semantic: Weight of a run: sum of the weights of the transitions.

\[ A^+ \rightarrow \mathbb{N} \cup \{-\infty\} \]

\[ w \mapsto \text{Max of the weights of the accepting runs labelled by } w \]

\[ (-\infty \text{ if no such run}) \]

\[ a^{n_0} ba^{n_1} b \cdots ba^{n_k+1} \mapsto \max(n_1, \ldots, n_k) \]
Weighted automata [Schützenberger]

$\mathcal{A} : A^* \rightarrow S$
Weighted automata \([Schützenberger]\)

\[
[A] : A^* \rightarrow S
\]

Semiring \((S, \oplus, \otimes)\): transitions are weighted by elements of \(S\)
Semiring \((S, \oplus, \otimes)\): transitions are weighted by elements of \(S\)

Paths: \(\otimes\) 
Non-determinism: \(\oplus\)
Weighted automata [Schützenberger]

\[ [\mathcal{A}] : A^* \to S \]

Semiring \((S, \oplus, \otimes)\): transitions are weighted by elements of \(S\)

Paths: \(\otimes\)  \hspace{2cm}  Non-determinism: \(\oplus\)

\[ [\mathcal{A}] : w \mapsto \bigoplus_{\rho \text{ accepting path labelled by } w} (\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_{|w|}) \]
Automata weighted over \((\mathbb{R}, +, \times)\)

\[ [A] : A^* \rightarrow \mathbb{R} \]

An example with \(A = \{0, 1\}\)

\begin{align*}
0, 1 & : 2 \\
0 & : 0 \\
1 & : 1 \\
0, 1 & : 1
\end{align*}
Automata weighted over \((\mathbb{R}, +, \times)\)

\[ [\mathcal{A}] : A^* \rightarrow \mathbb{R} \]

An example with \(A = \{0, 1\}\)

\[
\begin{align*}
0, 1 & : 2 \\
0 & : 0 \\
1 & : 1
\end{align*}
\]

\(100101 \mapsto 2^0 + 0 + 0 + 2^3 + 0 + 2^5\)
Automata weighted over \((\mathbb{R}, +, \times)\)

\[ [\mathcal{A}] : \mathbb{A}^* \rightarrow \mathbb{R} \]

An example with \(\mathbb{A} = \{0, 1\}\)

\[
\begin{align*}
0, 1 : 2 & \quad 0 : 0 \\
0, 1 : 1 & \quad 1 : 1
\end{align*}
\]

100101 \(\mapsto\) \(2^0 + 0 + 0 + 2^3 + 0 + 2^5\)

1. For all \(u \neq v\), is there \(\mathcal{A} \in \mathcal{C}\) which distinguishes \(u\) and \(v\)?
   \(\rightarrow\) Yes
Automata weighted over \((\mathbb{R}, +, \times)\)

\[ [\mathcal{A}] : A^* \to \mathbb{R} \]

An example with \( A = \{0, 1\} \)

\[
\begin{align*}
0, 1 : & 2 \\
0 : & 0 \\
1 : & 1 \\
\end{align*}
\]

\[
100101 \mapsto 2^0 + 0 + 0 + 2^3 + 0 + 2^5
\]

1. For all \( u \neq v \), is there \( \mathcal{A} \in \mathcal{C} \) which distinguishes \( u \) and \( v \)?
   \( \rightarrow \) Yes

2. Is there \( \mathcal{A} \in \mathcal{C} \) which distinguishes all pairs \( u \neq v \)?
   \( \rightarrow \) Yes
Automata weighted over $(\mathbb{R}, +, \times)$

$[\mathcal{A}] : A^* \rightarrow \mathbb{R}$

An example with $A = \{0, 1\}$

$0, 1 : 2$

0 : 0

1 : 1

$0, 1 : 1$

100101 \mapsto 2^0 + 0 + 0 + 2^3 + 0 + 2^5$

1. For all $u \neq v$, is there $\mathcal{A} \in \mathcal{C}$ which distinguishes $u$ and $v$?
   → Yes

2. Is there $\mathcal{A} \in \mathcal{C}$ which distinguishes all pairs $u \neq v$?
   → Yes

3. Minimal size to distinguish two given input words?
   → 1 or 2 states
Max-plus automata

Semiring \((\mathbb{N} \cup \{-\infty\}, \max, +)\)

\([A] : A^* \rightarrow \mathbb{N} \cup \{-\infty\}\)

\([A] : w \mapsto \max_{\rho \text{ accepting path labelled by } w} \left( \rho_1 + \rho_2 + \cdots + \rho_{|w|} \right)\)
Max-plus automata

Semiring \((\mathbb{N} \cup \{-\infty\}, \max, +)\)

\[ [A] : A^* \rightarrow \mathbb{N} \cup \{-\infty\} \]

\[
[A] : w \mapsto \max_{\rho \text{ accepting path labelled by } w} (\rho_1 + \rho_2 + \cdots + \rho_{|w|})
\]

1. For all \(u \neq v\), is there \(A \in C\) which distinguishes \(u\) and \(v\)?
   \(\rightarrow\) Yes
Max-plus automata

Semiring \((\mathbb{N} \cup \{-\infty\}, \max, +)\)

\([A] : A^* \rightarrow \mathbb{N} \cup \{-\infty\}\)

\([A] : w \mapsto \max_{\rho \text{ accepting path labelled by } w} (\rho_1 + \rho_2 + \cdots + \rho_{|w|})\)

1. For all \(u \neq v\), is there \(A \in \mathcal{C}\) which distinguishes \(u\) and \(v\)?
   \(\rightarrow\) Yes

2. Is there \(A \in \mathcal{C}\) which distinguishes all pairs \(u \neq v\)?
   \(\rightarrow\) No
Max-plus automata

Semiring \((\mathbb{N} \cup \{-\infty\}, \max, +)\)

\[[\mathcal{A}]: A^* \rightarrow \mathbb{N} \cup \{-\infty\}\]

\[[\mathcal{A}]: \quad w \mapsto \max_{\rho \text{ accepting path labelled by } w} (\rho_1 + \rho_2 + \cdots + \rho_{|w|})\]

1. For all \(u \neq v\), is there \(\mathcal{A} \in \mathcal{C}\) which distinguishes \(u\) and \(v\)?
   → Yes

2. Is there \(\mathcal{A} \in \mathcal{C}\) which distinguishes all pairs \(u \neq v\)?
   → No

3. Minimal size to distinguish two given input words?
   → ???????
Given a positive integer $n$, are there $u \neq v$ such that for all max-plus automata $\mathcal{A}$ with at most $n$ states:

$$\llbracket \mathcal{A} \rrbracket (u) = \llbracket \mathcal{A} \rrbracket (v)$$
If $n = 1$

$A = \{a, b\}$

Max-plus automata with one state can distinguish words with different contents (in particular different lengths), and only these ones.
If $n = 1$

$A = \{a, b\}$

$w \mapsto \alpha w_a + \beta w_b$

Max-plus automata with one state can distinguish words with different contents (in particular different lengths), and only these ones.
Max-plus automata with one state can distinguish words with different contents (in particular different lengths), and only these ones.
If \( n = 2 \) or \( n = 3 \)

There exist pairs of distinct words with the same values for all automata with at most 3 states...

But we do not know much more.
If $n = 2$ or $n = 3$

There exist pairs of distinct words with the same values for all automata with at most 3 states...

But we do not know much more.

2 states [Izhakian, Margolis] - words of length 20
If $n = 2$ or $n = 3$

There exist pairs of distinct words with the same values for all automata with at most 3 states...

But we do not know much more.

2 states [Izhakian, Margolis] - words of length 20

3 states [Shitov] - words of length 1795308
Theorem [Izhakian]

For all $n$, there exist a pair of distinct words $u \neq v$ such that for all triangular automata $A$ with at most $n$ states, 

$[A](u) = [A](v)$
Theorem [Izhakian]
For all $n$, there exist a pair of distinct words $u \neq v$ such that for all triangular automata $A$ with at most $n$ states,

$$[A](u) = [A](v)$$
Let's go back to automata with 2 states

\[ A = \{a, b\} \]
Let’s go back to automata with 2 states

\[ A = \{a, b\} \]
Let’s go back to automata with 2 states

$A = \{a, b\}$

First attempt: Restrict the class of automata we have to consider

- $\mathbb{R} \rightarrow \mathbb{Q} \rightarrow \mathbb{Z} \rightarrow \mathbb{N}$
Let’s go back to automata with 2 states

\[ A = \{a, b\} \]

First attempt: Restrict the class of automata we have to consider

- \(\mathbb{R} \rightarrow \mathbb{Q} \rightarrow \mathbb{Z} \rightarrow \mathbb{N}\)
- Complete automaton
Let’s go back to automata with 2 states

\[ A = \{a, b\} \]

First attempt: Restrict the class of automata we have to consider

- \( \mathbb{R} \rightarrow \mathbb{Q} \rightarrow \mathbb{Z} \rightarrow \mathbb{N} \)
- Complete automaton
- Only one initial and one final states
Let’s go back to automata with 2 states

\[ A = \{a, b\} \]

First attempt: Restrict the class of automata we have to consider

- \( \mathbb{R} \rightarrow \mathbb{Q} \rightarrow \mathbb{Z} \rightarrow \mathbb{N} \)
- Complete automaton
- Only one initial and one final states
- Reduce the number of parameters
Let’s go back to automata with 2 states

Second attempt: Give a list of criteria which can be checked
Let’s go back to automata with 2 states

Second attempt: Give a list of criteria which can be checked

- Content, length
Let’s go back to automata with 2 states

Second attempt: Give a list of criteria which can be checked

- Content, length
- ...
Let’s go back to automata with 2 states

Second attempt: Give a list of criteria which can be checked

- Content, length
- ...

Theorem [D., Johnson] - counter-example to a conjecture of Izhakian

There are two pairs of distinct words of minimal length which cannot be distinguished by any max-plus automata with two states:

\[ a^2 b^3 a^3 b a b a b^3 a^2 = a^2 b^3 a b a b a^3 b^3 a^2 \quad \text{and} \quad ab^3 a^4 b a b a^2 b^3 a = ab^3 a^2 b a b a b^3 a \]
Let’s go back to automata with 2 states

**Second attempt:** Give a list of criteria which can be checked

- Content, length
- ...

**Theorem [D., Johnson]** - counter-example to a conjecture of Izhakian

There are two pairs of distinct words of minimal length which cannot be distinguished by any max-plus automata with two states:

\[ a^2 b^3 a^3 babab^3 a^2 = a^2 b^3 ababa^3 b^3 a^2 \quad \text{and} \quad ab^3 a^4 baba^2 b^3 a = ab^3 a^2 baba^4 b^3 a \]

\[ \rightarrow \] Eliminate the shortest pairs by using the list of criteria

\[ \rightarrow \] Checking the pairs directly using the restrictions
A closer look at the list of criteria
A closer look at the list of criteria

- First and last blocks
A closer look at the list of criteria

- First and last blocks
- Bloc-permutation

![Diagram with states and transitions]
A closer look at the list of criteria

- First and last blocks
- Bloc-permutation
- “Counting modulo 2” criteria

Number of $a$’s after an even number of $b$’s

![Diagram showing the number of $a$’s after an even number of $b$’s]
A closer look at the list of criteria

- First and last blocks
- Bloc-permutation
- “Counting modulo 2” criteria
- Triangular automata with two states
Matrix representation
Matrix representation

\[
\begin{bmatrix}
0 & -\infty & -\infty \\
-\infty & 1 & -\infty \\
-\infty & -\infty & 0
\end{bmatrix}
\]
Matrix representation

\[ \mu(a) = \begin{pmatrix} 0 & -\infty & -\infty \\ -\infty & 1 & -\infty \\ -\infty & -\infty & 0 \end{pmatrix} \quad \mu(b) = \begin{pmatrix} 0 & 0 & -\infty \\ -\infty & -\infty & 0 \\ -\infty & -\infty & 0 \end{pmatrix} \]
Matrix representation

\[
\begin{align*}
\mu(a) &= \begin{pmatrix} 0 & -\infty & -\infty \\
-\infty & 1 & -\infty \\
-\infty & -\infty & 0 \end{pmatrix} &
\mu(b) &= \begin{pmatrix} 0 & 0 & -\infty \\
-\infty & -\infty & 0 \\
-\infty & -\infty & 0 \end{pmatrix} \\
I &= \begin{pmatrix} 0 & -\infty & -\infty \end{pmatrix} &
F &= \begin{pmatrix} -\infty \\
-\infty \\
0 \end{pmatrix}
\end{align*}
\]
Matrix representation

\[
\begin{align*}
\mu(a) &= \begin{pmatrix}
0 & -\infty & -\infty \\
-\infty & 1 & -\infty \\
-\infty & -\infty & 0
\end{pmatrix} & \mu(b) &= \begin{pmatrix}
0 & 0 & -\infty \\
-\infty & -\infty & 0 \\
-\infty & -\infty & 0
\end{pmatrix} \\
I &= \begin{pmatrix}
0 & -\infty & -\infty \\
-\infty & 0 & -\infty \\
-\infty & -\infty & 0
\end{pmatrix} & F &= \begin{pmatrix}
-\infty \\
-\infty \\
0
\end{pmatrix} \\
\mu(w)_{i,j} &= \text{max of the weights of the runs from } i \text{ to } j \text{ labelled by } w \\
[A](w) &= I\mu(w)F
\end{align*}
\]
Matrix representation

\[
\mu(a) = \begin{pmatrix}
0 & -\infty & -\infty \\
-\infty & 1 & -\infty \\
-\infty & -\infty & 0
\end{pmatrix}
\quad \mu(b) = \begin{pmatrix}
0 & 0 & -\infty \\
-\infty & -\infty & 0 \\
-\infty & -\infty & 0
\end{pmatrix}
\]

\[
I = \begin{pmatrix}
0 & -\infty & -\infty
\end{pmatrix} \quad F = \begin{pmatrix}
-\infty \\
-\infty \\
0
\end{pmatrix}
\]

\[
\mu(w)_{i,j} = \text{max of the weights of the runs from } i \text{ to } j \text{ labelled by } w
\]

\[
[A](w) = I\mu(w)F
\]

Dimension = Number of states
And now what?

- **Ultimate (very far away) goal:** Characterize all the identities holding for the class of max-plus automata with at most $n$ states, for all $n$...

- Is there a strict subset of max-plus automata containing all their computational power?