Equivalence Testing of Weighted Automata over Partially Commutative Monoids

V. Arvind[1] Abhranil Chatterjee[1] Rajit Datta[2]
Partha Mukhopadhyay[2]

[1]Institute of Mathematical Sciences(HBNI), India
[2]Chennai Mathematical Institute, India

Highlights of Logic Automata and Games 2020

Alphabet: $\Sigma = \{a, b\}$

A has 5 states. Black states are final states.

Series Recognized:

$$\mathcal{S}(A) = \sum_{i=0}^{\infty} (6ab)^i - (6ba)^i$$

Coefficient of the word *baba* in $\mathcal{S}(A)$ is -36.

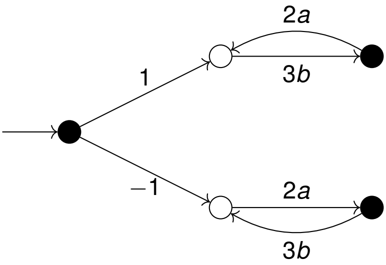

Figure: Weighted Automaton A

Two weighted Automata A, B are said to be *equivalent* if $\mathcal{S}(A) = \mathcal{S}(B)$.

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

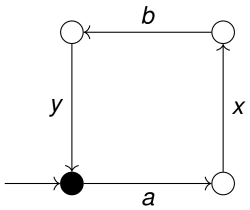

Figure: Multi-tape Automaton A

Input Tape:

$\overset{\downarrow}{a}\ b\ a\ b$

$\overset{\downarrow}{x}\ y\ x\ y$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

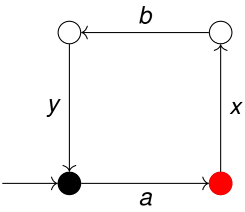

Figure: Multi-tape Automaton A

Input Tape:

$$a\ \overset{\downarrow}{b}\ a\ b$$

$$\overset{\downarrow}{x}\ y\ x\ y$$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

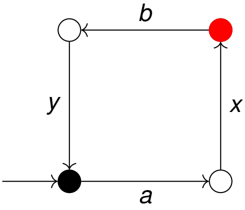

Figure: Multi-tape Automaton A

Input Tape:

$$a \overset{\downarrow}{b} a b$$

$$x \overset{\downarrow}{y} x y$$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

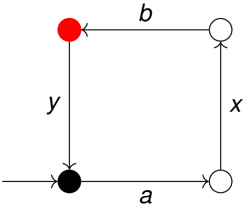

Figure: Multi-tape Automaton *A*

Input Tape:

$$a\ b\ \overset{\downarrow}{a}\ b$$

$$x\ \overset{\downarrow}{y}\ x\ y$$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

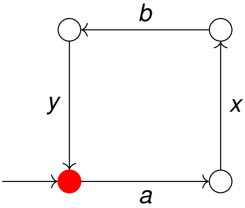

Figure: Multi-tape Automaton A

Input Tape:

$$a\ b\ \overset{\downarrow}{a}\ b$$

$$x\ y\ \overset{\downarrow}{x}\ y$$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

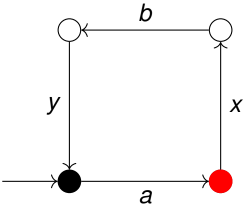

Figure: Multi-tape Automaton A

Input Tape:

$a\ b\ a\ \overset{\downarrow}{b}$

$x\ y\ \overset{\downarrow}{x}\ y$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

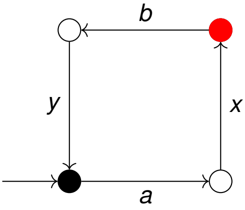

Figure: Multi-tape Automaton A

Input Tape:

$$a\ b\ a\ \overset{\downarrow}{b}$$

$$x\ y\ x\ \overset{\downarrow}{y}$$

Multi-tape Automaton

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

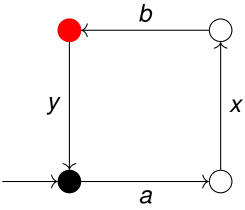

Figure: Multi-tape Automaton A

Input Tape:

$$a\ b\ a\ b$$

$$x\ y\ x\ \overset{\downarrow}{y}$$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

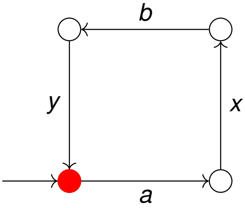

Figure: Multi-tape Automaton A

Input Tape:

$a\ b\ a\ b$

$x\ y\ x\ y$

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

An accepting run looks like:

$$a\ x\ b\ y\ a\ x\ b\ y$$

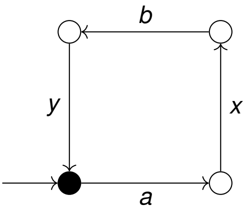

Figure: Multi-tape Automaton A

Input Tape:

$$a\ b\ a\ b$$

$$x\ y\ x\ y$$

Multi-tape Automaton

Alphabets:

$$\Sigma_1 = \{a, b\}$$

$$\Sigma_2 = \{x, y\}$$

A has 4 states. Black states are final states.

An accepting run looks like:

$$a\ x\ b\ y\ a\ x\ b\ y$$

***k*-tape Language Accepted:**

$$\mathcal{L}^2 \subseteq \Sigma_1^* \times \Sigma_2^*$$

For the automaton in the figure we have

$$\mathcal{L}^2(A) = \left\{ \left((ab)^i, (xy)^i \right) \right\}_{i=0}^{\infty}$$

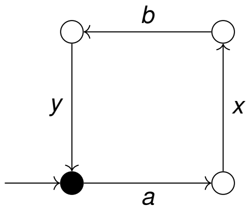

Figure: Multi-tape Automaton *A*

Input Tape:

$a\ b\ a\ b$

$x\ y\ x\ y$

Equivalence of Weighted Multi-tape Automata

Alphabets:

$$\Sigma_1 = \{a\} \qquad \Sigma_2 = \{b\}$$

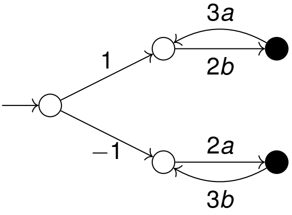

Figure: 2-tape Automaton A

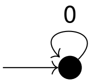

Figure: 2-tape Automaton B

Two weighted k-tape automata A, B are said to be *equivalent* if they recognize the same series. In this case

$$\begin{aligned} \mathcal{S}^2(A) &= \sum_{i=1}^{\infty}((2a)^i, (3b)^i) - ((3a)^i, (2b)^i) \\ &= \sum_{i=1}^{\infty} 6^i(a^i, b^i) - 6^i(a^i, b^i) \\ &= 0 = \mathcal{S}^2(B) \end{aligned} \qquad (1)$$

History of Multi-tape Automata

[RS59]**Rabin & Scott 1959:** Introduced the concept of multi-tape automata.

[Gri68]**Griffiths 1968:** Equivalence of multi-tape NFA is undecidable.

[Bir73, Val74]**Bird 1973, Valiant 1974:** Equivalence of 2-tape DFA is decidable.

[Bee76]**Beeri 1976:** Exponential time algorithm for Equivalence 2-tape DFA.

[FG82]**Friedman & Greibach 1982:** Polynomial time algorithm for equivalence of 2-tape DFA. The authors also conjectured the same for k-tape automaton for fixed k.

[HK91]**Harju & Karhumäki 1991:** Equivalence of weighted multi-tape NFA is decidable.

[Wor13]**Worrell 2013:** Randomized Polynomial time algorithm for Equivalence of weighted k-tape NFA for fixed k.

This Work 2020: *Deterministic* Quasi-Polynomial time algorithm for Equivalence of weighted k-tape NFA (and more).

Partially Commutative Monoids

Alphabet:

$$\Sigma = \{x_1, x_2, \dots, x_n\}$$

Relations I can be extended to Σ^*.

$$x_1 x_5 x_3 x_4 x_2 \sim_I x_5 x_1 x_2 x_3 x_4$$

Quotenting by I we obtain a **partially commutative monoid**

$$M = \Sigma^* / I$$

In case of k-tape Automata we have

$$\Sigma = \Sigma_1 \dot{\cup} \Sigma_2 \dot{\cup} \cdots \dot{\cup} \Sigma_k$$

$$I = \cup_{i=1}^{k} \Sigma_i \times \Sigma_i$$

G_M is a disjoint union of k many cliques.

Symmetric non-commutation relations

$$I \subseteq \Sigma \times \Sigma$$

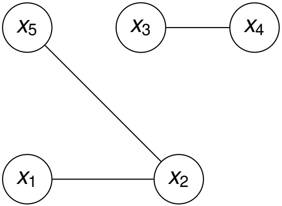

Figure: Example of a non-commutation graph G_M

Structure of non-commutation graph and Complexity of Equivalence testing.

Let A and B be $\mathbb{F}$-weighted automata of total size s over a pc monoid M.

Theorem

If the non-commutation graph G_M has a clique cover of size k. Then the equivalence of A and B can be decided in deterministic $(nks)^{O(k^2 \log ns)}$ time. Here n is the size of the alphabet of M and the clique edge-cover is given as part of the input.

Theorem

If the non-commutation graph G_M has a clique and star edge-cover of size k. Then the equivalence of A and B can be decided in randomized $(ns)^{O(k)}$ time. Here n is the size of the alphabet of M and the clique and star cover is given as part of the input.

Open Problems

1. Deterministic Polynomial time algorithm?
2. Efficient algorithm for other types of coverings of G_M?
3. Hardness over general G_M?
   a) We show that the hardest case is when G_M is a matching.

Open Problems

1. Deterministic Polynomial time algorithm?
2. Efficient algorithm for other types of coverings of G_M?
3. Hardness over general G_M?
   a) We show that the hardest case is when G_M is a matching.

Thank You!

References I

C. Beeri.
An improvement on valiant's decision procedure for equivalence of deterministic finite turn pushdown machines.
Theoretical Computer Science, 3(3):305 – 320, 1976.

Malcolm Bird.
The equivalence problem for deterministic two-tape automata.
J. Comput. Syst. Sci., 7(2):218–236, 1973.

Emily P. Friedman and Sheila A. Greibach.
A polynomial time algorithm for deciding the equivalence problem for 2-tape deterministic finite state acceptors.
SIAM J. Comput., 11:166–183, 1982.

T. V. Griffiths.
The unsolvability of the equivalence problem for nondeterministic generalized machines.
J. ACM, 15(3):409–413, July 1968.

Tero Harju and Juhani Karhumäki.
The equivalence problem of multitape finite automata.
Theor. Comput. Sci., 78(2):347–355, 1991.

M. Rabin and D. Scott.
Finite automata and their decision problems.
IBM J. Res. Dev., 3:114–125, 1959.

Leslie G. Valiant.
The equivalence problem for deterministic finite-turn pushdown automata.
Information and Control, 25(2):123 – 133, 1974.

James Worrell.
Revisiting the equivalence problem for finite multitape automata.
In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, pages 422–433, 2013.