Scalable Learning of One-Counter Automata via State-Merging Algorithms

Prince Mathew

Postdoctoral Researcher

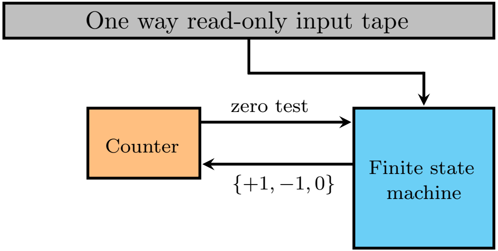
Université Libre de Bruxelles, Belgium

prince.mathew@ulb.be

Joint work with Shibashis Guha (TIFR), Anirban Majumdar (TIFR) and Sreejith A.V (IIT Palakkad)

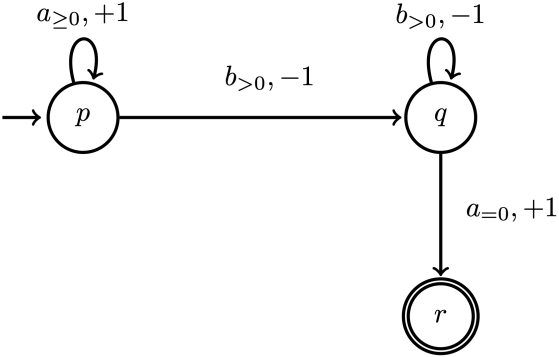
Highlights 2026

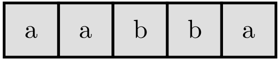
Deterministic real-time one-counter automata



- Write to counter: Increment (+1), No change (0), Decrement (-1).
- Read from counter: zero (0) or positive (+).
- Counter-value is always non-negative.
- Transitions of the finite-state machine are deterministic.
- There are no ϵ- transitions.

Example: DROCA





Input tape

$$\mathscr{L} = \{a^n b^n a \mid n > 0\}$$

Short history: Learning one-counter automata

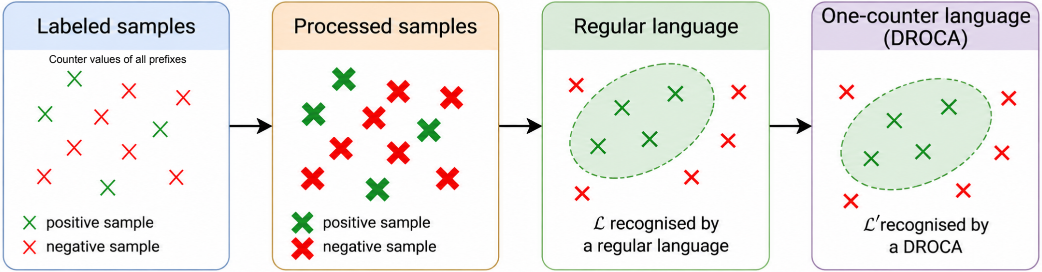
- (Gold, 1978) → Passive learning minimal DFA is NP-complete.
- (Angluin, 1987) → Active learning of DFA in polynomial time.
- (Fahmy & Roos, 1995) → Efficient learning of real-time OCA.
- (Neider & Löding, 2010) → Learning visibly OCA in EXPTIME.
- (Bruyère et. al, 2022) → Learning DROCA with an additional counter-value query in EXPTIME.
- (Mathew et. al, 2025a) → Learning DROCA using polynomially many queries.
- (Mathew et. al, 2025b) → Learning DOCA in polynomial time.

This talk

- `OPNI` : A passive learning algorithm for DROCA.
- `OCA-L*` : A semi-algorithm for learning DROCA that uses `OPNI`.
- The proposed active learning method outperforms existing techniques on randomly generated samples.

OPNI : Passive learning of DROCA

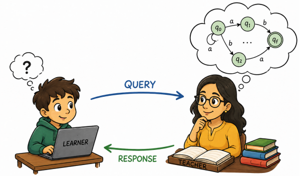
- Input:
  - Set of accepted words S^+ and set of rejected words S^-.
  - Counter values for all prefixes in S^+ and S^-.



Encode counter values　　RPNI (Oncina, Garcia, 1992)　　Decode counter values

- The output DROCA accepts all words in S^+ and reject all words in S^-.
- For all prefixes in S^+ and S^- the counter value reached by the DROCA is same as that of the input.

Active learning framework



Membership queries:

Learner: "Is w in $\mathscr{L}(\mathscr{A})$?"
Teacher : "yes" or "no"

Counter-value queries:

Learner: "What is the counter-value reached on reading w?"
Teacher: "counter-value reached on w"

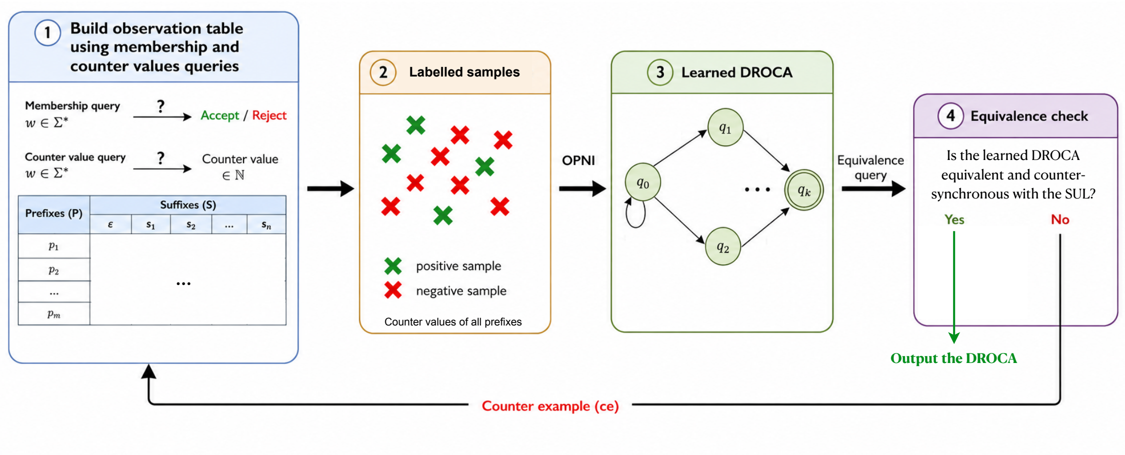
Minimal-equivalence queries:

Learner: "Does $\mathscr{B}$ and $\mathscr{A}$ recognise the same language?"
Teacher : "yes" or "no & the smallest counter-example"

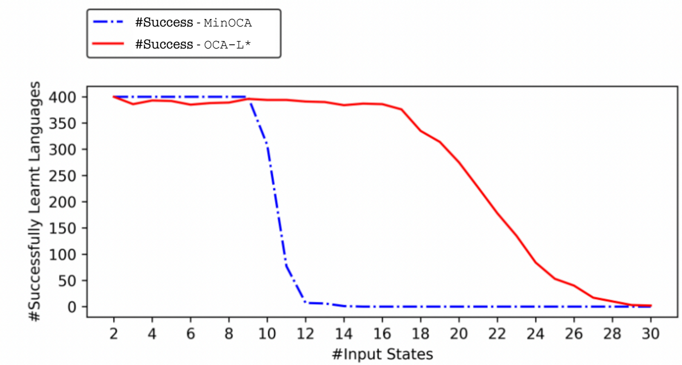
Membership different

Counter-value different

OCA-L* : Active learning of DROCA



Experimental evaluation



Number of successfully learnt DROCAs by `MinOCA` and `OCA-L*`

- Works in practice but **no** termination guarantees.

Conclusion and future work

- Passive learning algorithm (`OPNI`) for DROCA.
- Semi-algorithm (`OCA-L*`) for active learning of DROCA that works well in practice.
- The existing algorithm for learning DROCA in polynomial time is not suitable for practical use.
- Is there a faster polynomial time algorithm for learning DROCA that works in practice?

References I

Dana Angluin.
"Learning regular sets from queries and counter examples"
Information and Computation, pages 87-106, 1987.

Fahmy and Roos.
"Efficient learning of real time one-counter automata."
In ALT: International Workshop on Algorithmic Learning Theory, 1995.

Daniel Neider and Christof Loding.
"Learning visibly one-counter automata in polynomial time."
Technical Report, RWTH Aachen, AIB-2010-02, 2010.

Véronique Bruyère, Guillermo A. Pérez, and Gaëtan Staquet.
"Learning realtime one-counter automata"
Tools and Algorithms for the Construction and Analysis of Systems, pages 244–262, 2022.

References II

Jose Oncina and Pedro García.
"Inferring regular languages in polynomial update time"
World Scientific, 01, 1992.

E Mark Gold.
"Complexity of automaton identification from given data"
Information and Control, pages 302-320, 1978.

Prince Mathew, Vincent Penelle and A.V. Sreejith.
"Learning real-time one-counter automata using polynomially many queries"
Tools and Algorithms for the Construction and Analysis of Systems, pages 276-294, 2025.

Prince Mathew, Vincent Penelle and A.V. Sreejith.
"Learning deterministic one-counter automata in polynomial time"
ACM/IEEE Symposium on Logic in Computer Science, pages 444-457, 2025.

Thank You !