Learning Deterministic One-Counter Automata

Prince Mathew, *prince@iitgoa.ac.in*
Indian Institute of Technology Goa, India

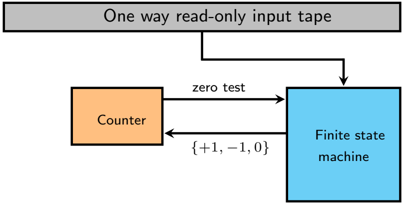
Vincent Penelle, *vincent.penelle@u-bordeaux.fr*
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, France

A.V. Sreejith, *sreejithav@iitgoa.ac.in*
Indian Institute of Technology Goa, India

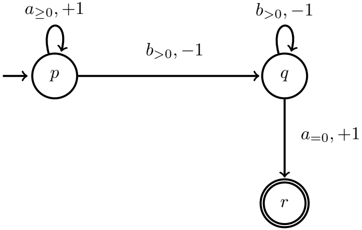
2 October, 2025

RP 2025

One-counter automata

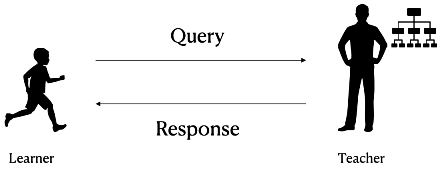


Counter: Can be incremented, decremented or tested for zero.



OCA accepting $\{a^n b^n a \mid n \geq 0\}$.

Active learning framework



- Teacher knows the language of a DOCA $\mathcal{T}$.
- Learner's aim: Learn a DOCA $\mathcal{L}$ equivalent to $\mathcal{T}$.
- Learner can query Teacher.
- Learner uses Teacher's response to learn.

Queries

- The Teacher knows the language of a DOCA $\mathcal{T}$.
- Learner's aim: Learn a DOCA $\mathcal{L}$ equivalent to $\mathcal{T}$.

Membership query

Learner: Is w in the language of $\mathcal{T}$?

Teacher: Yes or No.

Minimal-equivalence query

Learner: Is DOCA $\mathcal{L}$ equivalent to $\mathcal{T}$?

Teacher: Yes or "No and a minimal word w that distinguishes $\mathcal{L}$ and $\mathcal{T}$".

Literature review: Active learning of DOCA

Angluin, 1987	DFA – Membership, Equivalence P
Fahmy & Roos, 1995	DROCA[1] – Membership, Min. Equivalence EXPTIME
Neider & Löding, 2010	VOCA[2] – Membership, Min. Equivalence EXPTIME
Bruyère, Pérez & Staquet, 2022	DROCA[1] – Membership, Min. Equivalence, Counter Value EXPTIME
M., Penelle & Sreejith, 2025	DROCA[1] – Membership, Min. Equivalence, Counter Value EXPTIME, Polynomial queries
M., Penelle & Sreejith, 2025	DOCA – Membership, Min. Equivalence P

[1] Realtime DOCA: strict subclass of DOCA

[2] VOCA: Visibly OCA

Literature review: Active learning of DOCA

Angluin, 1987 → DFA – Membership, Equivalence
P

Fahmy & Roos, 1995 → DROCA[1] – Membership, Min. Equivalence
EXPTIME

Neider & Löding, 2010 → VOCA[2] – Membership, Min. Equivalence
EXPTIME

Bruyère, Pérez & Staquet, 2022 → DROCA[1] – Membership, Min. Equivalence, Counter Value
EXPTIME

M., Penelle & Sreejith, 2025 → DROCA[1] – Membership, Min. Equivalence, Counter Value
EXPTIME, Polynomial queries

M., Penelle & Sreejith, 2025 → DOCA – Membership, Min. Equivalence
P

[1] Realtime DOCA: strict subclass of DOCA

[2] VOCA: Visibly OCA

The OL^* Algorithm

OL^* - Active learning of DOCA

Theorem (OL^* is in P)

- *Let Teacher knows a language recognised by a DOCA $\mathcal{T}$.*
- *The OL^* algorithm learns a DOCA $\mathcal{L}$ that is equivalent to $\mathcal{T}$ in time polynomial in $|\mathcal{T}|$, using membership and minimal-equivalence queries.*

Assumptions on DOCA

- The Teacher knows a language accepted by a DOCA $\mathcal{T}$.
  - We denote by $n = |\mathcal{T}|$, the number of states.
- To make the presentation simpler, we assume the following about $\mathcal{T}$:
  - There are no ε-transitions.
  - In a transition, the counter is incremented or decremented at most by one.
- Learner wants to learn a DOCA $\mathcal{L}$ equivalent to $\mathcal{T}$.

OL^*: Step 1. Learning k-behaviour DFA

- A DFA $\mathcal{A}$ is a k-**behaviour DFA** of $\mathcal{T}$ if $\mathcal{A}$ is k-equivalent to $\mathcal{T}$. That is,

$$w \text{ is accepted by } \mathcal{A} \quad \text{iff} \quad w \text{ is accepted by } \mathcal{T}, \quad \text{for all } |w| \leq k.$$

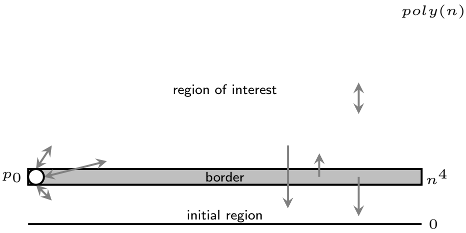
- Angluin's L^* algorithm can learn a k-behaviour DFA in time polynomial in k and n.
  - This is done using membership and minimal-equivalence queries.

- Step 1. of learner is to **learn a** $poly(n)$**-behaviour DFA**.

OL^*: Step 2. Partitioning the behaviour DFA

The DFA $\mathcal{A}$ is partitioned into:

- **Initial region**: States reachable by words of length $< n^4$.
- **Border region**: States reachable by words of length n^4 but not less.
- **Region of interest**: Remaining states.

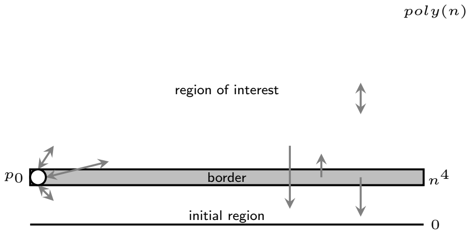


- What next?
  - Learner constructs a partial OCA for all border states.
  - Combine all these partial OCA to get the final DOCA $\mathcal{A}$.

OL^*: Step 2. Partitioning the behaviour DFA

The DFA $\mathcal{A}$ is partitioned into:

- **Initial region**: States reachable by words of length $< n^4$.
- **Border region**: States reachable by words of length n^4 but not less.
- **Region of interest**: Remaining states.



- What next?
  - Learner constructs a partial OCA for all border states.
  - Combine all these partial OCA to get the final DOCA $\mathcal{A}$.

OL^*: Step 3. Finding a winning sequence

Definition

We say $w_0, w_1, w_2, \ldots, w_K$ is a **winning sequence** for a state p if the run of these words on $\mathcal{T}$ reach configurations

$$(p, i),\ (p, i+d),\ (p, i+2d), \ldots, (p, i+Kd)$$

respectively, for some $d \leq n^2$ and $i > n^3$.

Lemma (Winning sequence lemma)

For any state p_0 in behaviour DFA $\mathcal{A}$, one can enumerate polynomially many sequences of the form $w_0, w_1, w_2, \ldots, w_K$ in polynomial time such that one of them is a winning sequence for p_0.

OL^*: Step 3. Finding a winning sequence

Definition

We say $w_0, w_1, w_2, \ldots, w_K$ is a **winning sequence** for a state p if the run of these words on $\mathcal{T}$ reach configurations

$$(p, i),\ (p, i+d),\ (p, i+2d), \ldots, (p, i+Kd)$$

respectively, for some $d \leq n^2$ and $i > n^3$.

Lemma (Winning sequence lemma)

For any state p_0 in behaviour DFA $\mathcal{A}$, one can enumerate polynomially many sequences of the form $w_0, w_1, w_2, \ldots, w_K$ in polynomial time such that one of them is a winning sequence for p_0.

OL^*: Step 4. Parallel BFS on $\mathcal{A}$

- Consider a *winning sequence*

$$w_0, w_1, w_2, \quad \ldots, \quad w_K.$$

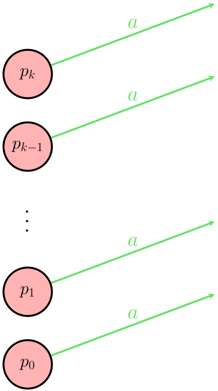
- Run these words on the behaviour DFA. We reach the state sequence

$$p_0, p_1, p_2, \quad \ldots, \quad p_K.$$

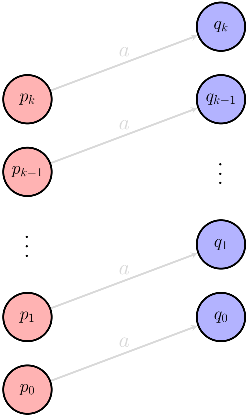
- Run parallel BFS (depth at most n^3) from this sequence.
  - All distinct sequences identified.
  - At most n^3 distinct sequences.
  - These sequences are the states of DOCA $\mathcal{L}_{p_0}$.

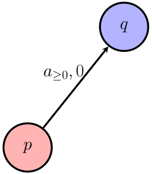
Parallel BFS

Parallel BFS

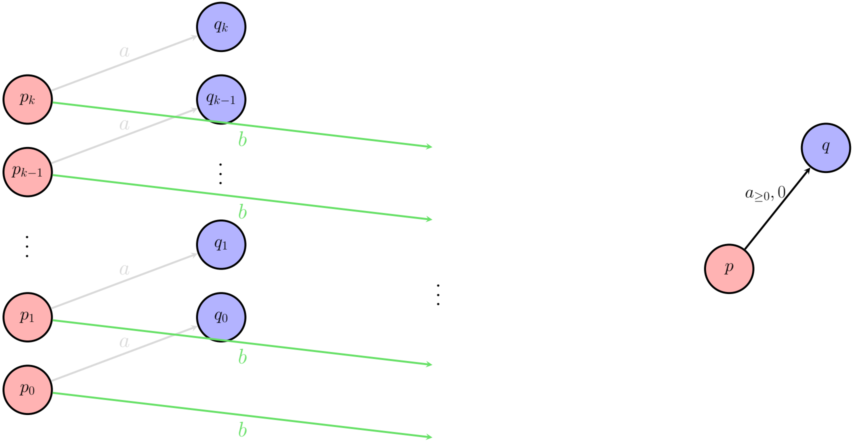


Parallel BFS

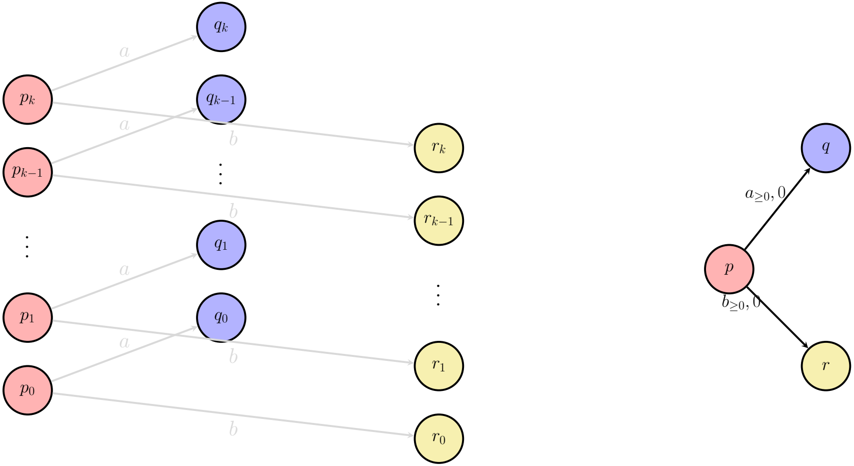




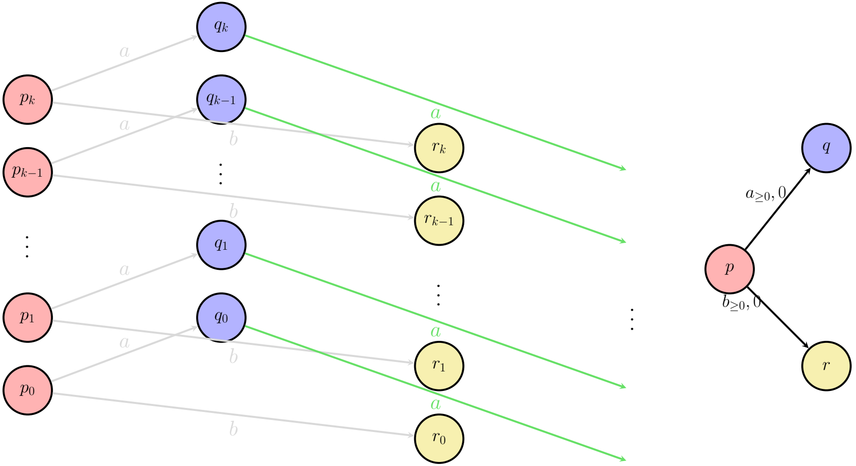
Parallel BFS



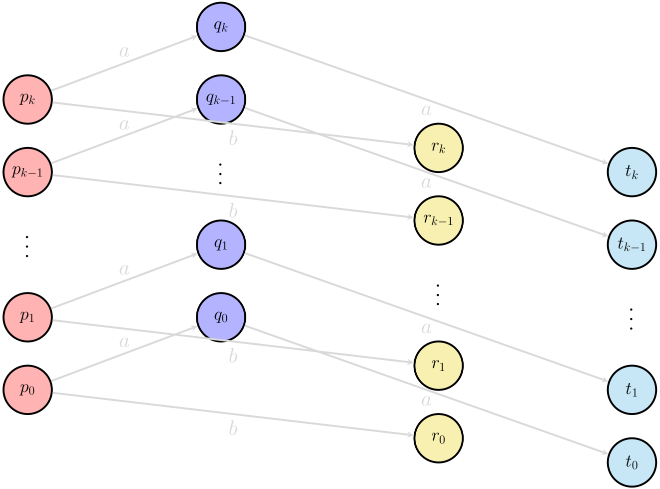
Parallel BFS

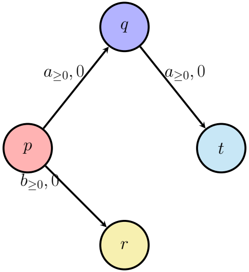


Parallel BFS

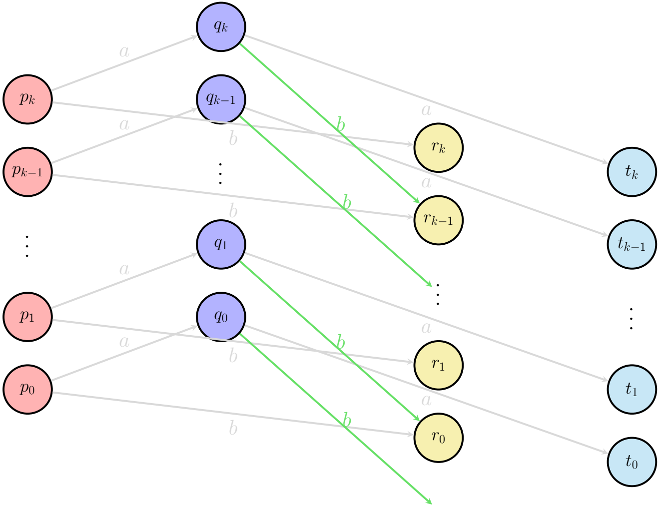


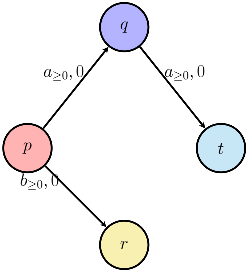
Parallel BFS



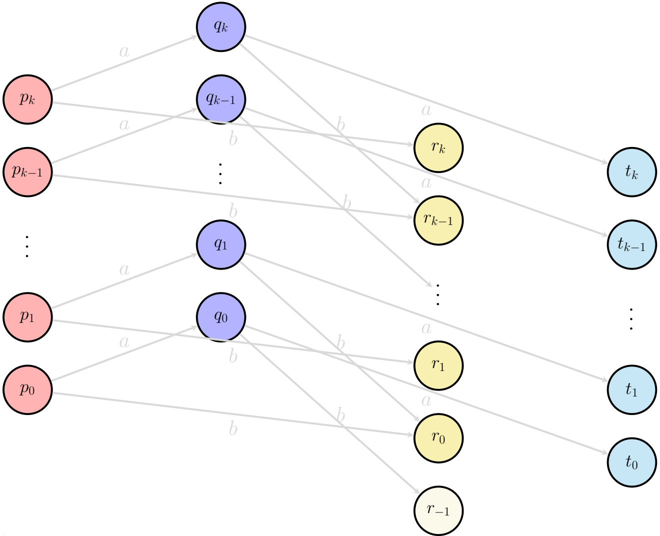


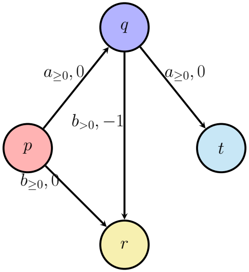
Parallel BFS



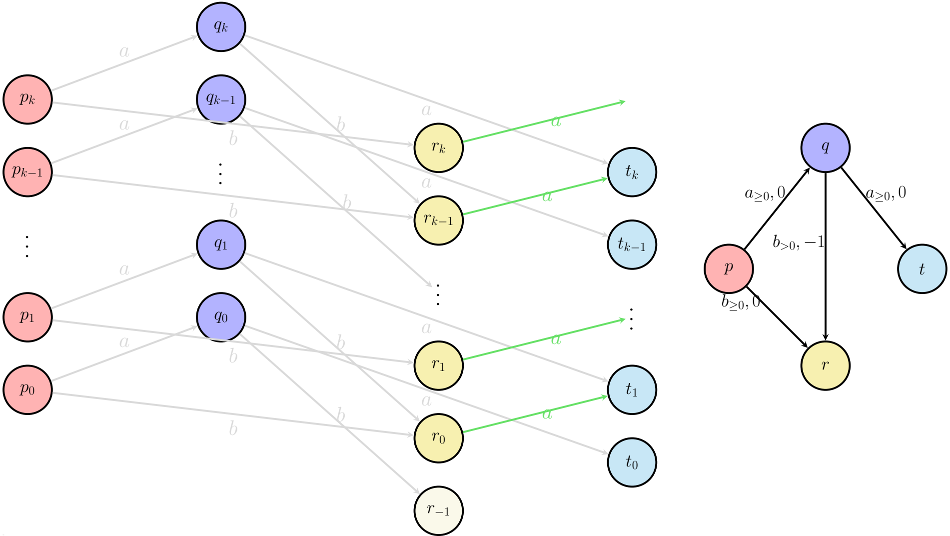


Parallel BFS

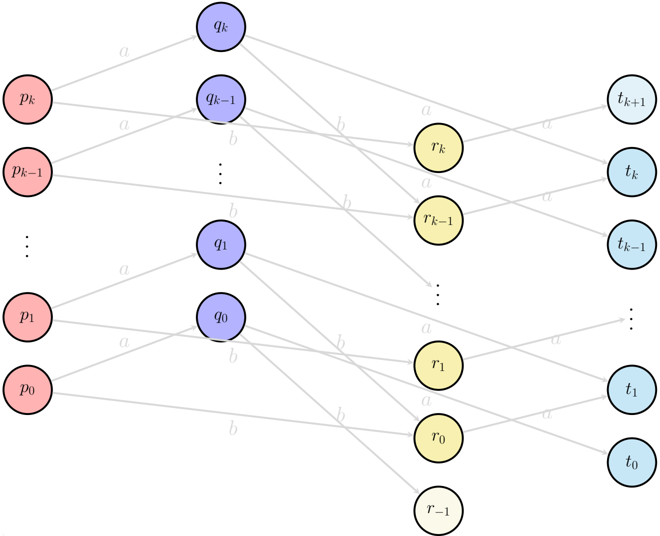


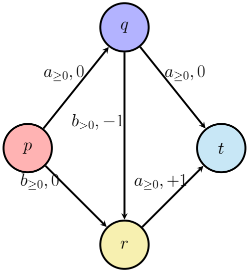


Parallel BFS

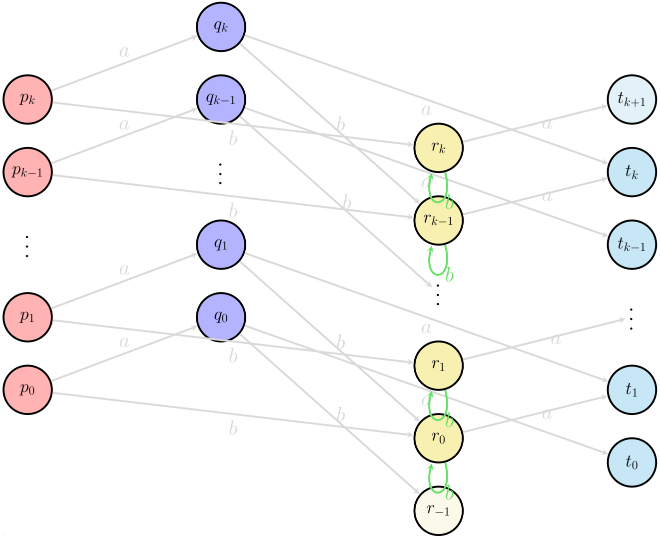


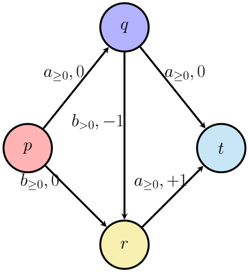
Parallel BFS



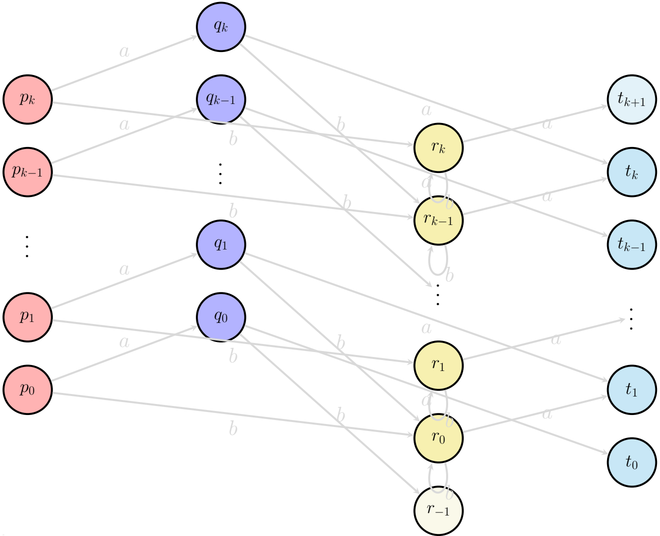


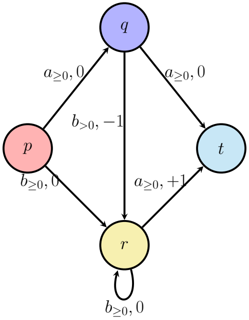
Parallel BFS



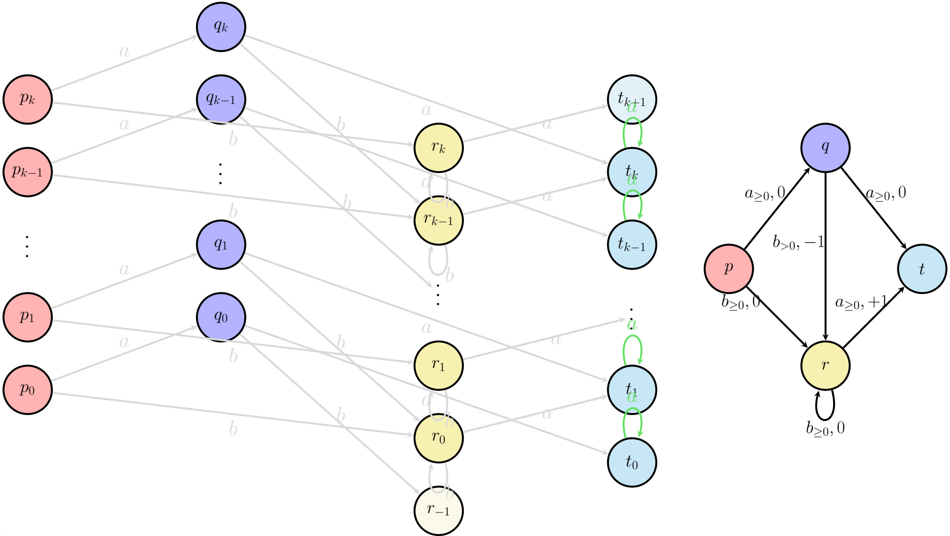


Parallel BFS

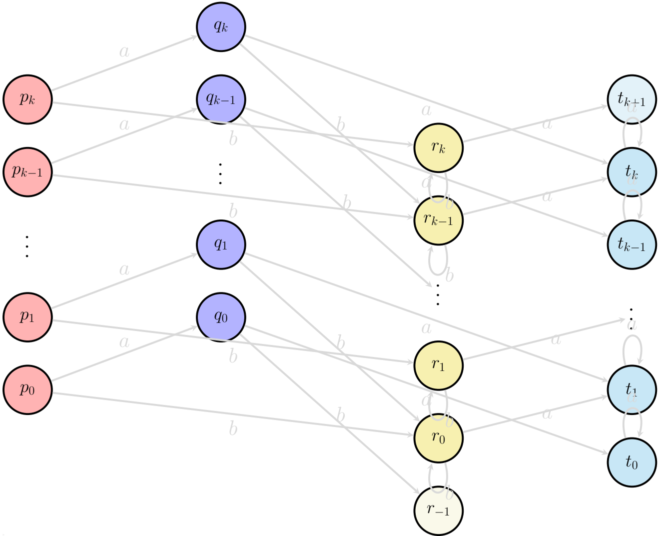


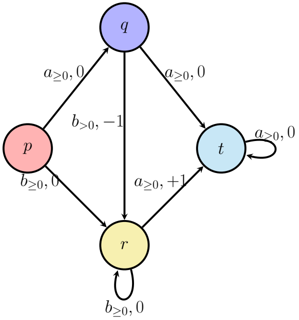


Parallel BFS

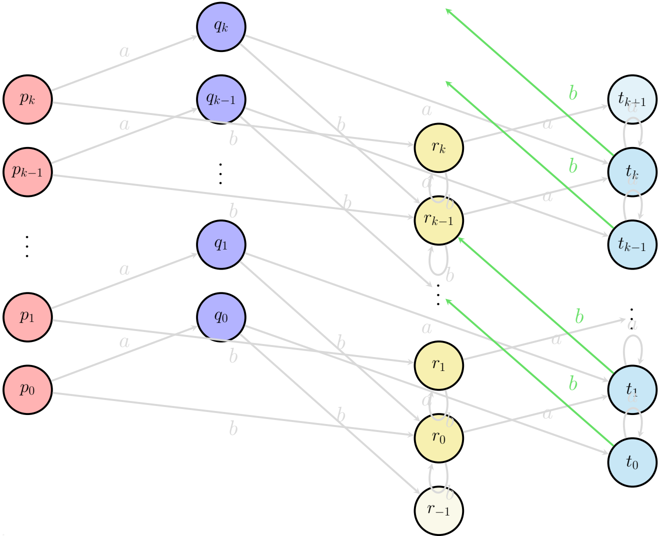


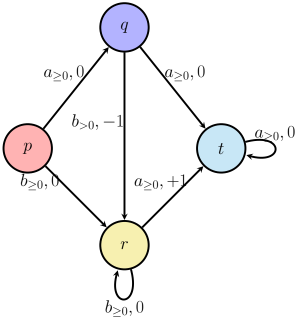
Parallel BFS



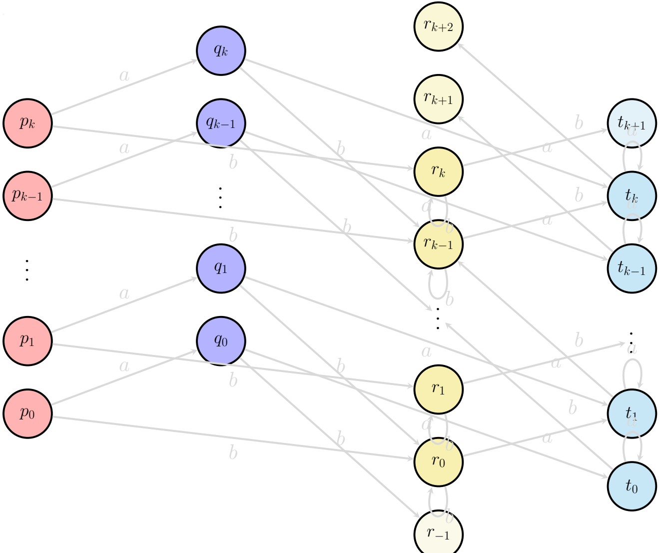


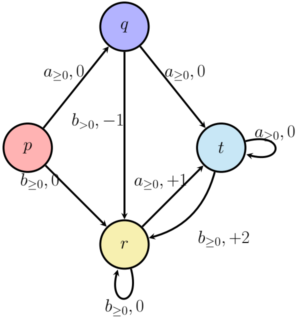
Parallel BFS





Parallel BFS

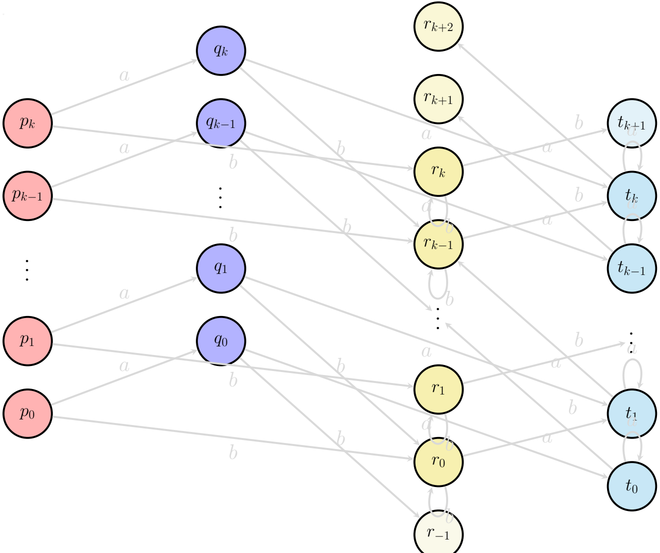


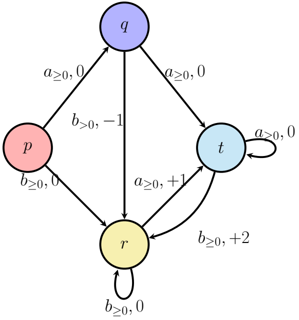


OL*: Step 5. Constructing $\mathcal{L}_{p_0}$

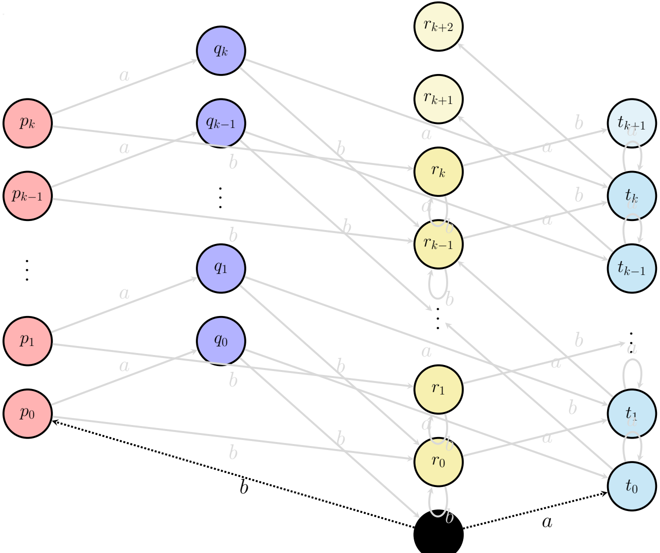
- The parallel BFS colors “most” of the reachable states from border state p.
- However, upto n^3 number of states are not colored.
  - eg. the state r_{-1} in the example, and some states reachable from r_{-1}.
- These are added to the partial OCA with zero counter value.
- The initial region is also added to the partial OCA with zero counter value.

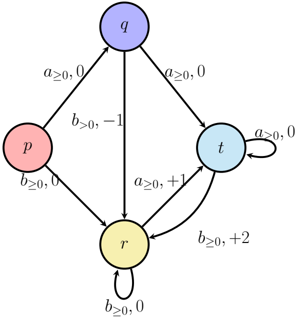
OL*: Step 5. Constructing $\mathcal{L}_{p_0}$



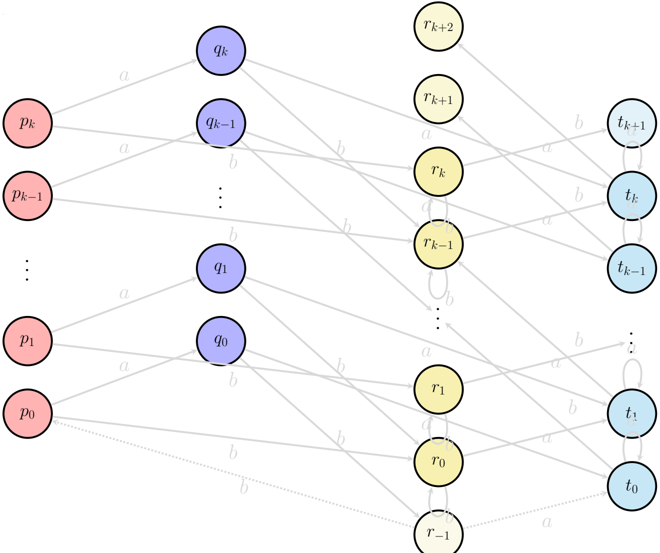


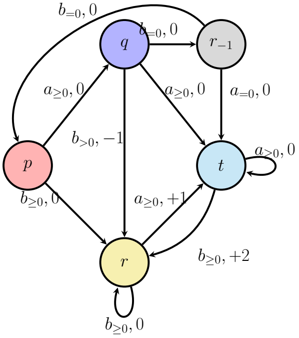
OL*: Step 5. Constructing $\mathcal{L}_{p_0}$





OL*: Step 5. Constructing $\mathcal{L}_{p_0}$





OL^*: Step 6. Constructing $\mathcal{L}$

- The final OCA $\mathcal{L}$ is the union of partial OCA corresponding to all border states and the DFA corresponding to the initial region.
- Size of $\mathcal{L}$ is $\mathcal{O}(n^8)$.
- There exists a $poly(n)$ such that if $\mathcal{L}$ is $poly(n)$ equivalent to $\mathcal{T}$, then $\mathcal{L}$ and $\mathcal{T}$ are equivalent (Böhm, Göller & Jančar, 2013).
- Correctness:

$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{A}$	(Step 6. construction)
$\mathcal{A}$ is $poly(n)$-equivalent to $\mathcal{T}$	(Step 1. construction)
$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{T}$	(from above)
$\mathcal{L}$ is equivalent to $\mathcal{T}$	(Böhm, Göller & Jančar, 2013)

OL^*: Step 6. Constructing $\mathcal{L}$

- The final OCA $\mathcal{L}$ is the union of partial OCA corresponding to all border states and the DFA corresponding to the initial region.
- Size of $\mathcal{L}$ is $\mathcal{O}(n^8)$.
- There exists a $poly(n)$ such that if $\mathcal{L}$ is $poly(n)$ equivalent to $\mathcal{T}$, then $\mathcal{L}$ and $\mathcal{T}$ are equivalent (Böhm, Göller & Jančar, 2013).
- Correctness:

$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{A}$	(Step 6. construction)
$\mathcal{A}$ is $poly(n)$-equivalent to $\mathcal{T}$	(Step 1. construction)
$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{T}$	(from above)
$\mathcal{L}$ is equivalent to $\mathcal{T}$	(Böhm, Göller & Jančar, 2013)

OL^*: Step 6. Constructing $\mathcal{L}$

- The final OCA $\mathcal{L}$ is the union of partial OCA corresponding to all border states and the DFA corresponding to the initial region.
- Size of $\mathcal{L}$ is $\mathcal{O}(n^8)$.
- There exists a $poly(n)$ such that if $\mathcal{L}$ is $poly(n)$ equivalent to $\mathcal{T}$, then $\mathcal{L}$ and $\mathcal{T}$ are equivalent (Böhm, Göller & Jančar, 2013).
- Correctness:

$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{A}$	(Step 6. construction)
$\mathcal{A}$ is $poly(n)$-equivalent to $\mathcal{T}$	(Step 1. construction)
$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{T}$	(from above)
$\mathcal{L}$ is equivalent to $\mathcal{T}$	(Böhm, Göller & Jančar, 2013)

OL^*: Step 6. Constructing $\mathcal{L}$

- The final OCA $\mathcal{L}$ is the union of partial OCA corresponding to all border states and the DFA corresponding to the initial region.
- Size of $\mathcal{L}$ is $\mathcal{O}(n^8)$.
- There exists a $poly(n)$ such that if $\mathcal{L}$ is $poly(n)$ equivalent to $\mathcal{T}$, then $\mathcal{L}$ and $\mathcal{T}$ are equivalent (Böhm, Göller & Jančar, 2013).
- Correctness:

$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{A}$	(Step 6. construction)
$\mathcal{A}$ is $poly(n)$-equivalent to $\mathcal{T}$	(Step 1. construction)
$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{T}$	(from above)
$\mathcal{L}$ is equivalent to $\mathcal{T}$	(Böhm, Göller & Jančar, 2013)

OL^*: Step 6. Constructing $\mathcal{L}$

- The final OCA $\mathcal{L}$ is the union of partial OCA corresponding to all border states and the DFA corresponding to the initial region.
- Size of $\mathcal{L}$ is $\mathcal{O}(n^8)$.
- There exists a $poly(n)$ such that if $\mathcal{L}$ is $poly(n)$ equivalent to $\mathcal{T}$, then $\mathcal{L}$ and $\mathcal{T}$ are equivalent (Böhm, Göller & Jančar, 2013).
- Correctness:

$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{A}$	(Step 6. construction)
$\mathcal{A}$ is $poly(n)$-equivalent to $\mathcal{T}$	(Step 1. construction)
$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{T}$	(from above)
$\mathcal{L}$ is equivalent to $\mathcal{T}$	(Böhm, Göller & Jančar, 2013)

OL^*: Step 6. Constructing $\mathcal{L}$

- The final OCA $\mathcal{L}$ is the union of partial OCA corresponding to all border states and the DFA corresponding to the initial region.
- Size of $\mathcal{L}$ is $\mathcal{O}(n^8)$.
- There exists a $poly(n)$ such that if $\mathcal{L}$ is $poly(n)$ equivalent to $\mathcal{T}$, then $\mathcal{L}$ and $\mathcal{T}$ are equivalent (Böhm, Göller & Jančar, 2013).
- Correctness:

$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{A}$	(Step 6. construction)
$\mathcal{A}$ is $poly(n)$-equivalent to $\mathcal{T}$	(Step 1. construction)
$\mathcal{L}$ is $poly(n)$-equivalent to $\mathcal{T}$	(from above)
$\mathcal{L}$ is equivalent to $\mathcal{T}$	(Böhm, Göller & Jančar, 2013)

Summary of OL^*

- Construct $poly(n)$-behaviour DFA using L^* algorithm.
- Partition the behaviour DFA into *initial region*, *border*, and *region of interest*.
- For each border state:
  - Generate a winning sequence of words: $w_0, w_1, \ldots, w_K$.
  - Run these words on the DFA to get the sequence of states: $p_0, p_1, \ldots, p_K$.
  - Run parallel BFS from this sequence.
  - All reachable sequences of parallel BFS form states of partial OCA.
  - Counter values are incremented / decremented based on the shift in the sequence.
  - Add the missed states and initial region to get partial OCA.
- Construct the final OCA by combining the partial OCA.

Conclusion

Theorem

OL^* learns a DOCA equivalent to the Teacher's DOCA using membership and minimal-equivalence queries, and in time polynomial in the size of a smallest DOCA recognising the language.

Corollary

Polynomial approximation for minimisation of DOCA.

Future work

- Replacing minimal-equivalence with equivalence query.
- Practical OL^* algorithm.
- Improving running time of equivalence.
- Learning weighted models (like weighted visibly OCA).

Thank You!