Learning Realtime One-Counter Automata

Highlights of Logic, Games, and Automata

Véronique Bruyère   Guillermo A. Pérez   Gaëtan Staquet

Theoretical computer science
Computer Science Department
Science Faculty
University of Mons

Formal Techniques in Software Engineering
Computer Science Department
Science Faculty
University of Antwerp

September 17, 2021

UMONS Université de Mons   fnrs LA LIBERTÉ DE CHERCHER   University of Antwerp

Definition 1

A *realtime one-counter automaton* (ROCA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta_{=0}, \delta_{>0}, q_0, F)$ with

- Q is the set of states;
- Σ is the alphabet;
- $\delta_{=0} : Q \times \Sigma \to Q \times \{0, +1\}$ and $\delta_{>0} : Q \times \Sigma \to Q \times \{-1, 0, +1\}$ are the transition functions;
- $q_0 \in Q$ is the initial state; and
- $F \subseteq Q$ is the set of accepting states.

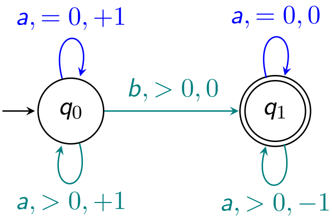

Figure 1: An ROCA $\mathcal{A}$ accepting $\{a^n b a^m \mid 0 < n \leq m\}$.

From any ROCA $\mathcal{A}$, one can construct the *behavior graph of* $\mathcal{A}$, which is an (infinite) automaton accepting $\mathcal{L}(\mathcal{A})$.

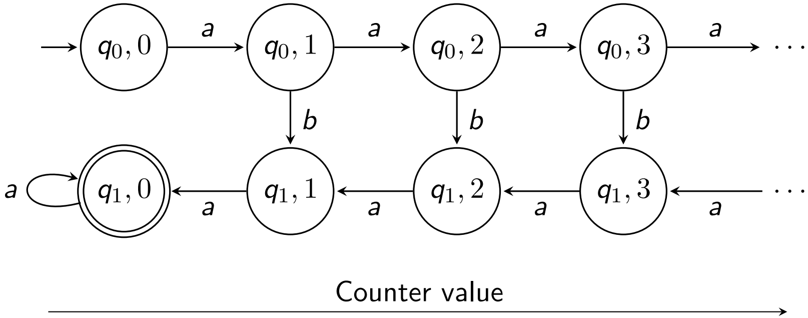

Figure 2: The behavior graph of $\mathcal{A}$.

From any ROCA $\mathcal{A}$, one can construct the *behavior graph of* $\mathcal{A}$, which is an (infinite) automaton accepting $\mathcal{L}(\mathcal{A})$.

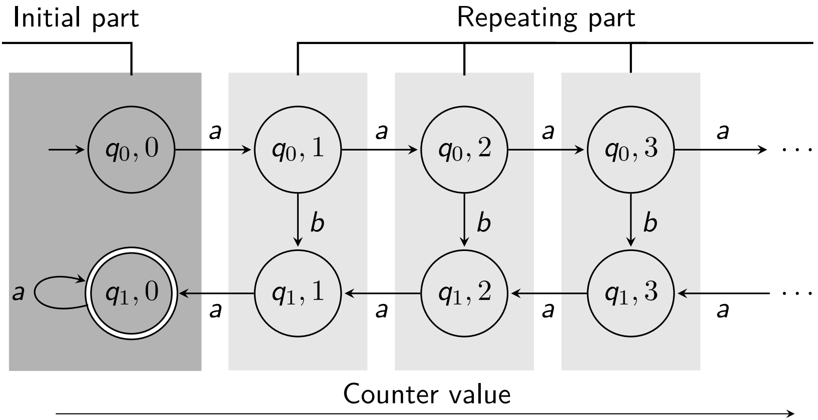

Figure 2: The behavior graph of $\mathcal{A}$.

From any ROCA $\mathcal{A}$, one can construct the *behavior graph of* $\mathcal{A}$, which is an (infinite) automaton accepting $\mathcal{L}(\mathcal{A})$.

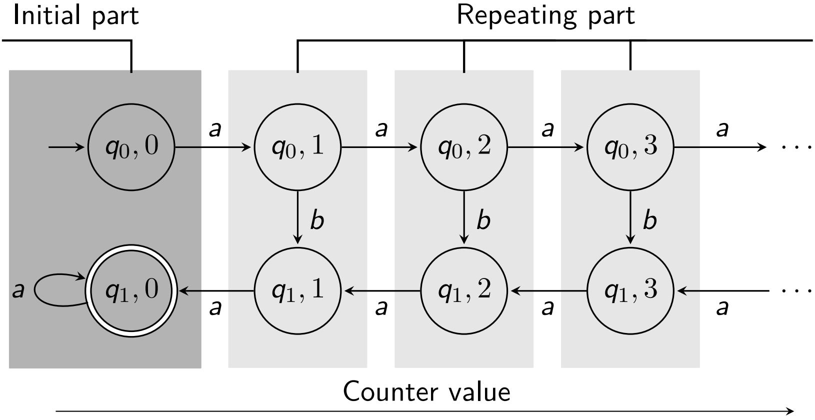

Figure 2: The behavior graph of $\mathcal{A}$.

This behavior graph is *ultimately periodic*.

Theorem 2

The behavior graph of any ROCA is ultimately periodic.

[1] Neider and Löding, *Learning visibly one-counter automata in polynomial time*, 2010.

Theorem 2

The behavior graph of any ROCA is ultimately periodic.

Let $\mathcal{A}$ be an ROCA. The main idea of the learning algorithm (based on Neider and Löding's work[1]) is as follows:

1. We learn $\mathcal{L}(BG(\mathcal{A}))$ up to a fixed counter value ℓ, noted L_ℓ.
2. We compute every possible periodic description of the resulting DFA.
3. We construct an ROCA for each description.
4. If one of the ROCAs accepts $\mathcal{L}(\mathcal{A})$, we are done. Otherwise, we increase ℓ and repeat the process.

[1] Neider and Löding, *Learning visibly one-counter automata in polynomial time*, 2010.

Figure 3: Adaptation of Angluin's framework[2] for ROCAs.

[2] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

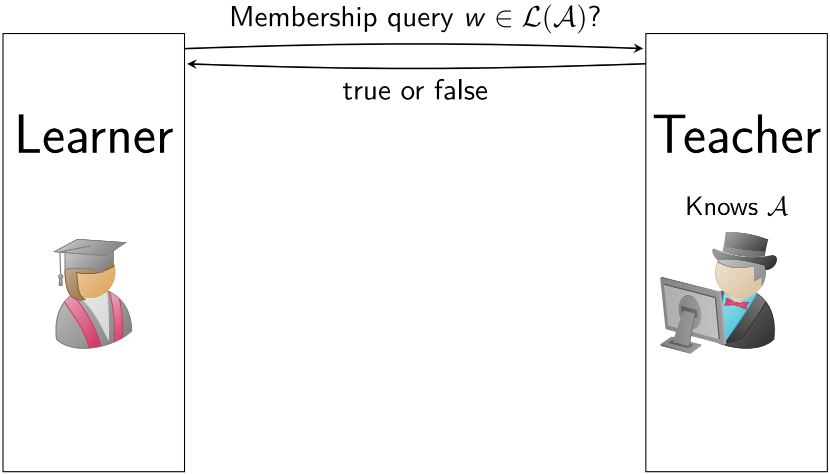

Figure 3: Adaptation of Angluin's framework[2] for ROCAs.

[2] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

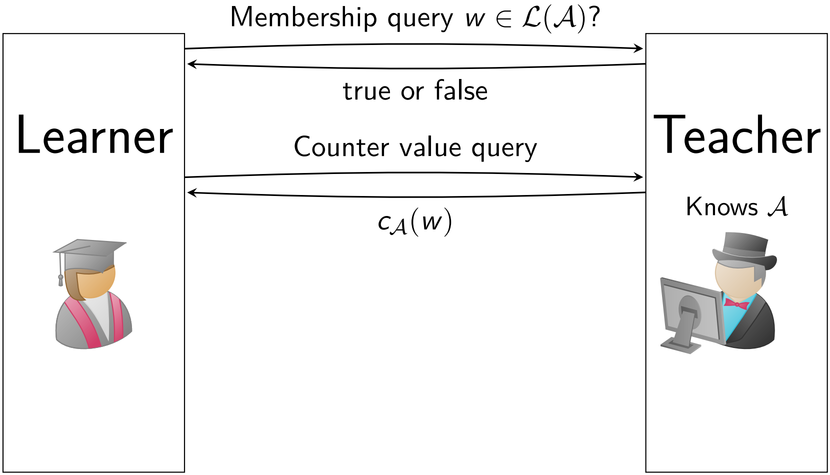

Figure 3: Adaptation of Angluin's framework[2] for ROCAs.

[2] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

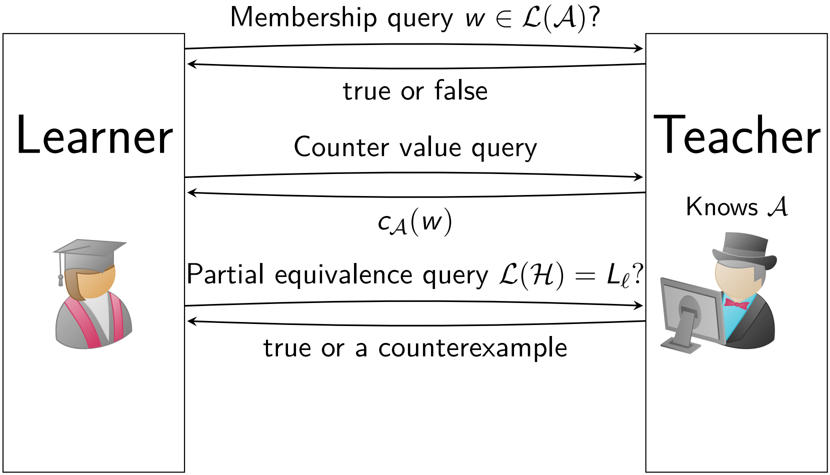

Figure 3: Adaptation of Angluin's framework[2] for ROCAs.

[2]Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

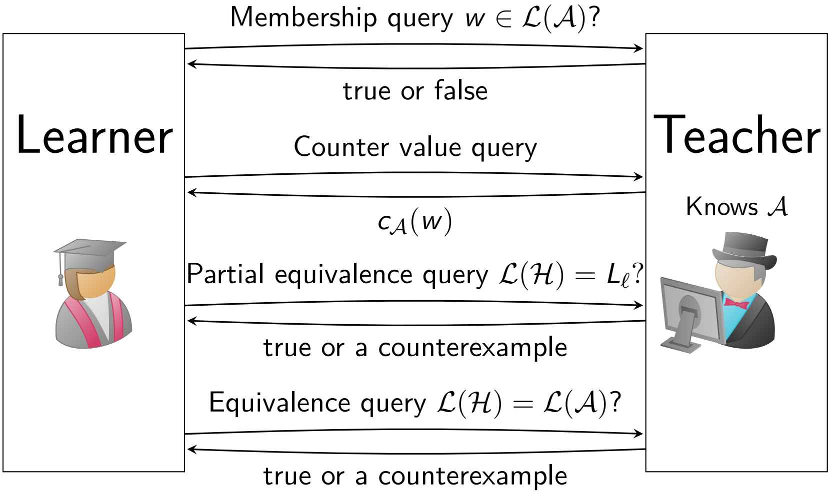

Figure 3: Adaptation of Angluin's framework[2] for ROCAs.

[2] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

Theorem 3

The learning algorithm has an exponential time and space complexity in the size of $\mathcal{A}$ and the length of the longest counterexample provided by the teacher.

Theorem 3

The learning algorithm has an exponential time and space complexity in the size of $\mathcal{A}$ and the length of the longest counterexample provided by the teacher.

Applications: verification of JSON, XML, and so on.

Theorem 3

The learning algorithm has an exponential time and space complexity in the size of $\mathcal{A}$ and the length of the longest counterexample provided by the teacher.

Applications: verification of JSON, XML, and so on.

A prototype was implemented (by modifying LearnLib and AutomataLib) and benchmarks will be presented in the paper.

References I

Angluin, Dana. "Learning Regular Sets from Queries and Counterexamples". In: *Inf. Comput.* 75.2 (1987), pp. 87–106. DOI: 10.1016/0890-5401(87)90052-6. URL: https://doi.org/10.1016/0890-5401(87)90052-6.

Neider, Daniel and Christof Löding. *Learning visibly one-counter automata in polynomial time*. Tech. rep. Technical Report AIB-2010-02, RWTH Aachen (January 2010), 2010.