Learning Nondeterministic Real-Time Automata

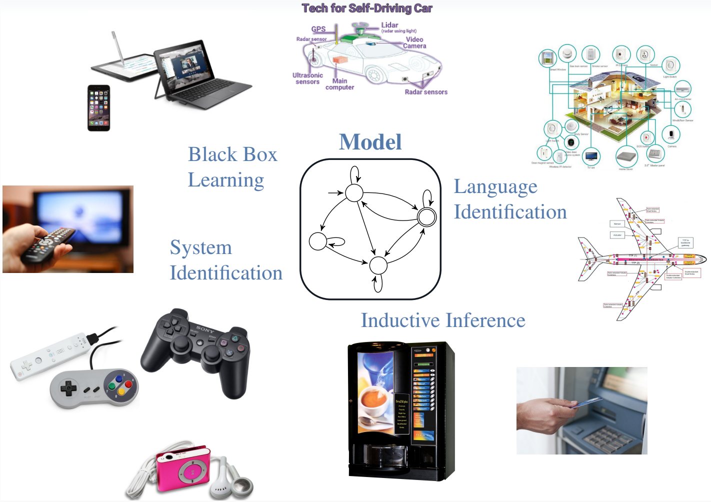
Jie An[1], Bohua Zhan[2], Naijun Zhan[2], Miaomiao Zhang[3]

1. Max Planck Institute for Software Systems, Kaiserslautern, Germany
2. SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing, China
3. School of Software Engineering, Tongji University, Shanghai, China

jiean@mpi-sws.org

EMSOFT'21 · Oct. 12

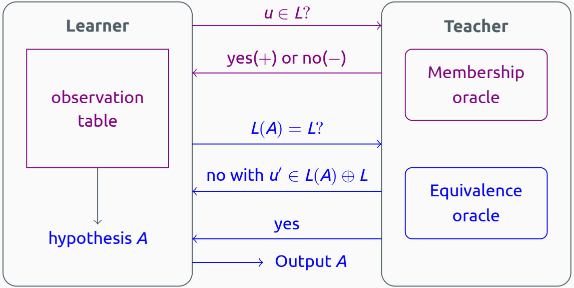
Model/Automaton learning



© The figure comes from Irini-Eleftheria Mens.

Minimally adequate teacher (MAT)

- Dana Angluin proposed an online, active, and exact learning framework L^* for Deterministic Finite Automata (DFA) in 1987.
- Two kinds of queries : membership query and equivalence query.
- Table conditions : **closed** and **consistent**.



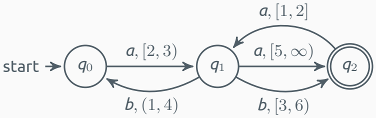
Real-time automata

☺ RTAs yield simple models while preserving adequate expressiveness.

Definition (Real-time automata)

A real-time automaton (RTA) is a tuple $\mathcal{A} = (Q, \Sigma, \Delta, Q_0, F)$ where

- Q is a finite set of locations;
- Σ is a finite alphabet;
- $\Delta \subseteq Q \times \Sigma \times 2^{\mathbb{R}_{\geq 0}} \times Q$ is a transition relation with $|\Delta| < \infty$, where $2^{\mathbb{R}_{\geq 0}}$ represents the set of intervals whose endpoints are in $\mathbb{N} \cup \{\infty\}$;
- $Q_0 \subseteq Q$ is a finite set of initial locations;
- $F \subseteq Q$ is a finite set of accepting locations.



- E.g. $\omega = (a, 2.1)(b, 3)$ is accepted.
- $\rho_1 = q_0 \xrightarrow[2.1]{a} q_1 \xrightarrow[3]{b} q_0$ and
  $\rho_2 = q_0 \xrightarrow[2.1]{a} q_1 \xrightarrow[3]{b} q_2$.

- Timed words $\omega \in (\Sigma \times \mathbb{R}_{\geq 0})^*$.
- Timed language : a set of timed words.
- **real-time language** : $\mathcal{L}$ can be recognized by an RTA.
- Querying timed words to learn real-time languages (automata) in the MAT framework.

Motivation and basic ideas

- Our previous work proposed a method to learn **deterministic** RTAs, solving the problem of infinite timed actions.
- ☕ How to learn **nondeterministic** RTAs directly?
  - DRTA can be exponentially bigger than an equivalent NRTA (w.r.t. # location).
  - NRTAs as more succinct models may be more useful, especially for some applications in verification.
  - ☹ No unique minimal NRTA for a real-time language $\Longrightarrow$ **It is not clear which target automaton should be learned**.

Motivation and basic ideas

- Our previous work proposed a method to learn **deterministic** RTAs, solving the problem of infinite timed actions.
- How to learn **nondeterministic** RTAs directly?
  - DRTA can be exponentially bigger than an equivalent NRTA (w.r.t. # location).
  - NRTAs as more succinct models may be more useful, especially for some applications in verification.
  - ☹ No unique minimal NRTA for a real-time language $\Longrightarrow$ **It is not clear which target automaton should be learned**.
- ☺ Define residual real-time automata, and prove that there is a unique minimal RRTA for a real-time language.
- ☺ Transform the learning problem of NRTAs to the learning problem of RRTAs.
- ☺ Propose two efficient learning algorithms for RRTAs.

Residual real-time automata

- **Residual real-time language** : $\omega^{-1}\mathcal{L} = \{\omega' \in (\Sigma \times \mathbb{R}_{\geq 0})^* \mid \omega\omega' \in \mathcal{L}\}$, given a real-time language $\mathcal{L}$ and a time word ω
  - finite number of residuals,
  - **Prime** residuals : $\omega^{-1}\mathcal{L}$ is called *prime* if $\bigcup \{\omega'^{-1}\mathcal{L} \mid \omega'^{-1}\mathcal{L} \subsetneq \omega^{-1}\mathcal{L}\} \subsetneq \omega^{-1}\mathcal{L}$,
  - **Composed** residuals : otherwise.

Residual real-time automata

- **Residual real-time language** : $\omega^{-1}\mathcal{L} = \{\omega' \in (\Sigma \times \mathbb{R}_{\geq 0})^* \mid \omega\omega' \in \mathcal{L}\}$, given a real-time language $\mathcal{L}$ and a time word ω
  - finite number of residuals,
  - **Prime** residuals : $\omega^{-1}\mathcal{L}$ is called *prime* if $\bigcup \{\omega'^{-1}\mathcal{L} \mid \omega'^{-1}\mathcal{L} \subsetneq \omega^{-1}\mathcal{L}\} \subsetneq \omega^{-1}\mathcal{L}$,
  - **Composed** residuals : otherwise.
- **Residual real-time automaton** is an NRTA $\mathcal{A} = (Q, \Sigma, \Delta, Q_0, F)$ such that $\forall q \in Q, \exists \omega \in (\Sigma \times \mathbb{R}_{\geq 0})^* : \mathcal{L}_q = \omega^{-1}\mathcal{L}(\mathcal{A})$.
- **Canonical residual real-time automaton** : $Q = \{\omega^{-1}\mathcal{L} \mid \omega^{-1}\mathcal{L} \text{ is prime}\}$

Theorem (Minimal RRTA)

The canonical residual real-time automaton $\mathcal{A}$ of a real-time language $\mathcal{L}$ is the minimal (w.r.t. the number of locations) RRTA which recognizes $\mathcal{L}$.

Residual real-time automata

- **Residual real-time language** : $\omega^{-1}\mathcal{L} = \{\omega' \in (\Sigma \times \mathbb{R}_{\geq 0})^* \mid \omega\omega' \in \mathcal{L}\}$, given a real-time language $\mathcal{L}$ and a time word ω
  - finite number of residuals,
  - **Prime** residuals : $\omega^{-1}\mathcal{L}$ is called *prime* if $\bigcup\{\omega'^{-1}\mathcal{L} \mid \omega'^{-1}\mathcal{L} \subsetneq \omega^{-1}\mathcal{L}\} \subsetneq \omega^{-1}\mathcal{L}$,
  - **Composed** residuals : otherwise.
- **Residual real-time automaton** is an NRTA $\mathcal{A} = (Q, \Sigma, \Delta, Q_0, F)$ such that $\forall q \in Q, \exists \omega \in (\Sigma \times \mathbb{R}_{\geq 0})^* : \mathcal{L}_q = \omega^{-1}\mathcal{L}(\mathcal{A})$.
- **Canonical residual real-time automaton** : $Q = \{\omega^{-1}\mathcal{L} \mid \omega^{-1}\mathcal{L} \text{ is prime}\}$

Theorem (Minimal RRTA)

The canonical residual real-time automaton $\mathcal{A}$ of a real-time language $\mathcal{L}$ is the minimal (w.r.t. the number of locations) RRTA which recognizes $\mathcal{L}$.

- Learning $\mathcal{L}$ (i.e. learning NRTA) $\Rightarrow$ Learning CRRTA $\Rightarrow$ finding all prime residuals.

NRTALearning

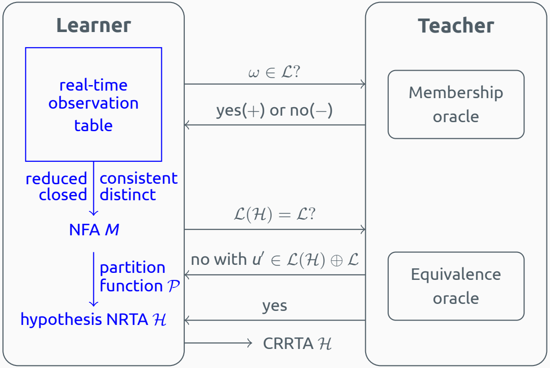


Figure – NRTALearning framework

NRTALearning

$\mathcal{T}$	ϵ	$(a, 8.1)$	$(a, 8.1)(a, 8.1)$	$(a, 15)$	
ϵ	$-$	$-$	$+$	$-$	$\checkmark$
$(a, 5.1)$	$-$	$+$	$-$	$-$	$\checkmark$
$(a, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
$(a, 5.1)(a, 8.1)$	$+$	$-$	$-$	$-$	$\checkmark$
$(b, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
$(a, 5.1)(a, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
$(a, 5.1)(b, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
$\cdots$			$\cdots$		$\cdots$
$(a, 7)$	$+$	$+$	$-$	$-$	$\times$
$\cdots$			$\cdots$		$\cdots$

NRTALearning

$\mathcal{T}$	ϵ	$(a, 8.1)$	$(a, 8.1)(a, 8.1)$	$(a, 15)$	
ϵ	$-$	$-$	$+$	$-$	✓
$(a, 5.1)$	$-$	$+$	$-$	$-$	✓
$(a, 0)$	$-$	$-$	$-$	$-$	✓
$(a, 5.1)(a, 8.1)$	$+$	$-$	$-$	$-$	✓
$(b, 0)$	$-$	$-$	$-$	$-$	✓
$(a, 5.1)(a, 0)$	$-$	$-$	$-$	$-$	✓
$(a, 5.1)(b, 0)$	$-$	$-$	$-$	$-$	✓
$\cdots$			$\cdots$		$\cdots$
$(a, 7)$	$+$	$+$	$-$	$-$	$\times$
$\cdots$			$\cdots$		$\cdots$

- $\omega^{-1}\mathcal{L} = \{\omega' \in (\Sigma \times \mathbb{R}_{\geq 0})^* \mid \omega\omega' \in \mathcal{L}\}$, finding the prime prefixes ω;
- Prefixes $S \cup R$; Suffixes E
- Prime prefixes : ✓ ; Composed prefixes : $\times$

NRTALearning

E

	$\mathcal{T}$	ϵ	$(a, 8.1)$	$(a, 8.1)(a, 8.1)$	$(a, 15)$	
S	ϵ	$-$	$-$	$+$	$-$	$\checkmark$
	$(a, 5.1)$	$-$	$+$	$-$	$-$	$\checkmark$
	$(a, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$(a, 5.1)(a, 8.1)$	$+$	$-$	$-$	$-$	$\checkmark$
R	$(b, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$(a, 5.1)(a, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$(a, 5.1)(b, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$\dots$			$\dots$		$\dots$
	$(a, 7)$	$+$	$+$	$-$	$-$	$\times$
	$\dots$			$\dots$		$\dots$

- $\omega^{-1}\mathcal{L} = \{\omega' \in (\Sigma \times \mathbb{R}_{\geq 0})^* \mid \omega\omega' \in \mathcal{L}\}$, finding the prime prefixes ω;
- Prefixes $S \cup R$; Suffixes E
- Prime prefixes : $\checkmark$; Composed prefixes : $\times$
- Function $val : S \cup R \to (E \to \{+, -\})$, e.g., $val(\epsilon) = - - + -$
- Function $row : S \cup R \to 2^E$ by $row(\omega) = \{e \in E \mid f(\omega \cdot e) = +\}$ for each $\omega \in S \cup R$, e.g.,
  $row((a, 7)) = \{\epsilon, (a, 8.1)\}$,
  $row((a, 5.1)) = \{(a, 8.1)\}$,
  $row((a, 5.1)(a, 8.1)) = \{\epsilon\}$;
  $row((a, 7)) = row((a, 5.1)) \cup row((a, 5.1)(a, 8.1))$.

NRTALearning

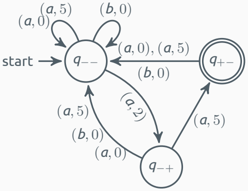
	$\mathcal{T}$	ϵ	$(a, 8.1)$	$(a, 8.1)(a, 8.1)$	$(a, 15)$	
S	ϵ	$-$	$-$	$+$	$-$	$\checkmark$
	$(a, 5.1)$	$-$	$+$	$-$	$-$	$\checkmark$
	$(a, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$(a, 5.1)(a, 8.1)$	$+$	$-$	$-$	$-$	$\checkmark$
R	$(b, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$(a, 5.1)(a, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$(a, 5.1)(b, 0)$	$-$	$-$	$-$	$-$	$\checkmark$
	$\cdots$			$\cdots$		$\cdots$
	$(a, 7)$	$+$	$+$	$-$	$-$	$\times$
	$\cdots$			$\cdots$		$\cdots$

E = $\{\epsilon, (a, 8.1), (a, 8.1)(a, 8.1), (a, 15)\}$

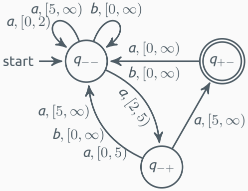
- $\omega^{-1}\mathcal{L} = \{\omega' \in (\Sigma \times \mathbb{R}_{\geq 0})^* \mid \omega\omega' \in \mathcal{L}\}$, finding the prime prefixes ω;
- Prefixes $S \cup R$; Suffixes E
- Prime prefixes : $\checkmark$; Composed prefixes : $\times$
- Function $val : S \cup R \to (E \to \{+, -\})$, e.g., $val(\epsilon) = - - + -$
- Function $row : S \cup R \to 2^E$ by $row(\omega) = \{e \in E \mid f(\omega \cdot e) = +\}$ for each $\omega \in S \cup R$, e.g., $row((a, 7)) = \{\epsilon, (a, 8.1)\}$, $row((a, 5.1)) = \{(a, 8.1)\}$, $row((a, 5.1)(a, 8.1)) = \{\epsilon\}$; $row((a, 7)) = row((a, 5.1)) \cup row((a, 5.1)(a, 8.1))$.

- Table conditions
  - **Reduced** : $\forall s \in S : s$ is prime, and $\forall s, s' \in S : s \neq s' \Rightarrow val(s) \neq val(s')$.
  - **Closed** : $\forall r \in R : row(r) = \bigcup \{row(s) \mid s \in S \wedge row(s) \subseteq row(r)\}$.
  - **Consistent** : $\forall \omega, \omega' \in S \cup R : row(\omega) \subseteq row(\omega') \Rightarrow row(\omega \cdot \boldsymbol{\sigma}) \subseteq row(\omega' \cdot \boldsymbol{\sigma})$ if $\omega \cdot \boldsymbol{\sigma}, \omega' \cdot \boldsymbol{\sigma} \in S \cup R$, where $\boldsymbol{\sigma} \in \Sigma \times \mathbb{R}_{\geq 0}$.
  - **Distinct** : $\forall \omega \in S \cup R, \boldsymbol{\sigma} \in \Sigma \times \mathbb{R}_{\geq 0} : \omega \cdot \boldsymbol{\sigma} \in S \cup R \Rightarrow s_i \cdot \boldsymbol{\sigma} \in S \cup R$, where $s_i \in \{s \in S \mid row(s) \subseteq row(\omega)\}$.

$\mathcal{T}$	ϵ	$(a,5)$
ϵ	$-$	$-$
$(a,2)$	$-$	$+$
$(a,2)(a,5)$	$+$	$-$
$(a,0)$	$-$	$-$
$(b,0)$	$-$	$-$
$(a,2)(a,0)$	$-$	$-$
$(a,2)(b,0)$	$-$	$-$
$(a,5)$	$-$	$-$
$(a,2)(a,5)(a,0)$	$-$	$-$
$(a,2)(a,5)(b,0)$	$-$	$-$
$(a,2)(a,5)(a,5)$	$-$	$-$



M



$\mathcal{H}$

Figure – A prepared table $\mathcal{T}$, the corresponding NFA M and the hypothesis NRTA $\mathcal{H}$.

$\mathcal{T}$	ϵ	$(a,5)$
ϵ	$-$	$-$
$(a,2)$	$-$	$+$
$(a,2)(a,5)$	$+$	$-$
$(a,0)$	$-$	$-$
$(b,0)$	$-$	$-$
$(a,2)(a,0)$	$-$	$-$
$(a,2)(b,0)$	$-$	$-$
$(a,5)$	$-$	$-$
$(a,2)(a,5)(a,0)$	$-$	$-$
$(a,2)(a,5)(b,0)$	$-$	$-$
$(a,2)(a,5)(a,5)$	$-$	$-$

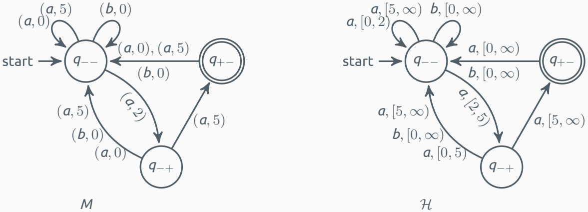


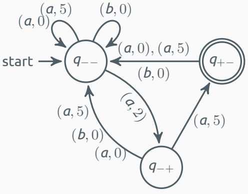
Figure – A prepared table $\mathcal{T}$, the corresponding NFA M and the hypothesis NRTA $\mathcal{H}$.

- $\mathcal{T}$ to M
  - $Q_M = \{q_{val(s)} \mid s \in S\}$; e.g., 3 locations : q_{--}, q_{-+} and q_{+-}.
  - $\Delta_M = \{(q_{val(\omega)}, \boldsymbol{\sigma}, q_{val(s')}) \mid \omega \cdot \boldsymbol{\sigma} \in S \cup R \wedge row(\omega) \in \{row(s) \mid s \in S\} \wedge row(s') \subseteq row(\omega \cdot \boldsymbol{\sigma}) \wedge s' \in S\}$;

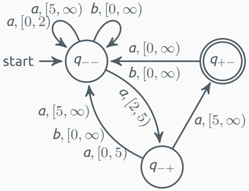
    e.g., Consider $(a,2)$ and $(a,2)(a,5)$, two transitions $q_{-+} \xrightarrow{(a,5)} q_{q_{--}}$ and $q_{-+} \xrightarrow{(a,5)} q_{q_{+-}}$, since $row(\epsilon) \subseteq row((a,2)(a,5))$.

NRTALearning

$\mathcal{T}$	ϵ	$(a,5)$
ϵ	$-$	$-$
$(a,2)$	$-$	$+$
$(a,2)(a,5)$	$+$	$-$
$(a,0)$	$-$	$-$
$(b,0)$	$-$	$-$
$(a,2)(a,0)$	$-$	$-$
$(a,2)(b,0)$	$-$	$-$
$(a,5)$	$-$	$-$
$(a,2)(a,5)(a,0)$	$-$	$-$
$(a,2)(a,5)(b,0)$	$-$	$-$
$(a,2)(a,5)(a,5)$	$-$	$-$



M



$\mathcal{H}$

Figure – A prepared table $\mathcal{T}$, the corresponding NFA M and the hypothesis NRTA $\mathcal{H}$.

- $\mathcal{T}$ to M
  - $Q_M = \{q_{val(s)} \mid s \in S\}$; e.g., 3 locations : q_{--}, q_{-+} and q_{+-}.
  - $\Delta_M = \{(q_{val(\omega)}, \boldsymbol{\sigma}, q_{val(s')}) \mid \omega \cdot \boldsymbol{\sigma} \in S \cup R \wedge row(\omega) \in \{row(s) \mid s \in S\} \wedge row(s') \subseteq row(\omega \cdot \boldsymbol{\sigma}) \wedge s' \in S\}$;

    e.g., Consider $(a,2)$ and $(a,2)(a,5)$, two transitions $q_{-+} \xrightarrow{(a,5)} q_{q_{--}}$ and $q_{-+} \xrightarrow{(a,5)} q_{q_{+-}}$, since $row(\epsilon) \subseteq row((a,2)(a,5))$.
- M to $\mathcal{H}$
  - Partition function maps a list of clock valuations $\ell = \tau_0, \tau_1, \cdots, \tau_n$ with $\lfloor \tau_i \rfloor \neq \lfloor \tau_j \rfloor$ to $\{I_0, I_1, \ldots, I_n\}$ with $\bigcup I_i = \mathbb{R}_{\geq 0}$,

$$I_i = \begin{cases} [\tau_i, \tau_{i+1}), & \text{if } \tau_i \in \mathbb{N} \wedge \tau_{i+1} \in \mathbb{N}; \\ (\lfloor \tau_i \rfloor, \tau_{i+1}), & \text{if } \tau_i \in \mathbb{R}_{\geq 0} \setminus \mathbb{N} \wedge \tau_{i+1} \in \mathbb{N}; \\ [\tau_i, \lfloor \tau_{i+1} \rfloor], & \text{if } \tau_i \in \mathbb{N} \wedge \tau_{i+1} \in \mathbb{R}_{\geq 0} \setminus \mathbb{N}; \\ (\lfloor \tau_i \rfloor, \lfloor \tau_{i+1} \rfloor], & \text{if } \tau_i \in \mathbb{R}_{\geq 0} \setminus \mathbb{N} \wedge \tau_{i+1} \in \mathbb{R}_{\geq 0} \setminus \mathbb{N}. \end{cases}$$

  - e.g., $\Psi_{q_{--}, a} = \{0, 2, 5\}$ and then get the intervals $[0,2)$,$[2,5)$ and $[5,\infty)$.

NRTALearning

Theorem (Correctness and Complexity)

*The Algorithm **NRTALearning** always terminates and takes $\mathcal{O}((n + nh(n^2 + m\kappa n)) \cdot h(n^2 + m\kappa n)))$ membership queries and $\mathcal{O}(n^2 + m\kappa n)$ equivalence queries to learn the CRRTA recognizing $\mathcal{L}$.*

Extended-NL^*

- NL^* is a learning algorithm for residual finite-state automata in the MAT framework [BolligHKL09].
- **Region** : since only one clock c, given a clock valuation $\nu \in \mathbb{R}_{\geq 0}$, the region

$$[\![\nu]\!] = \begin{cases} [\nu, \nu], & \text{if } \nu \in \mathbb{N} \text{ and } \nu \leq \kappa; \\ (\lfloor \nu \rfloor, \lfloor \nu \rfloor + 1), & \text{if } \nu \notin \mathbb{N} \text{ and } \nu \leq \kappa; \\ (\kappa, \infty), & \text{otherwise}. \end{cases}$$

  κ is the maximum integer appearing in the RTA.
- Suppose κ is known, then we can extend NL^* by enumerating all regions.
  1. Enumerate all regions $[0, 0]$,$(0, 1)$,$\ldots$,$[\kappa, \kappa]$,(κ, ∞) offline, denoted as $Reg_0, Reg_1, \ldots, Reg_{2\kappa+1}$.
  2. Construct an abstract alphabet $\Sigma_{M'} = \{(\sigma, \tau_i')|\sigma \in \Sigma \wedge \tau_i' \in Reg_i \wedge 0 \leq i \leq 2\kappa + 1\}$.
  3. Follow NL^* to learn NFA and then use partition function to construct hypothesis NRTA for the equivalence query.
- One solution is to guess an initial value of κ, then increase it whenever a counterexample involves a larger time value.

Theorem (Extended-NL^*)

*The Algorithm **Extended-NL^*** terminates and returns the CRRTA which recognizes the target real-time language $\mathcal{L}$ after performing $\mathcal{O}(h\kappa mn^3)$ membership queries and $\mathcal{O}(n^2)$ equivalence queries.*

Experiments

Group ID	$\lvert\Delta\rvert$	n_{DRTA}	Method	#Membership N_{min}	#Membership N_{mean}	#Membership N_{max}	#Equivalence N_{min}	#Equivalence N_{mean}	#Equivalence N_{max}	$\lvert Q_{\mathcal{H}}\rvert$	$t\,(s)$
10_2_20	23.5	66.5	NRTA-L	3520	5665.3	9660	31	36.4	40	11.1	1.7
			E-NL^*	15725	26571.5	35310	3	4.7	6		1.1
			DRTA-L	38461	223977.9	670380	6	9.1	13	66.5	19.3
10_4_20	32.3	62.7	NRTA-L	4975	9664.4	15192	33	49.2	62	11.1	3.8
			E-NL^*	31433	55362.6	75618	4	5.9	9		2.3
			DRTA-L	68561	347454.2	781231	7	8.7	11	62.7	24.6
10_6_20	43.7	79.0	NRTA-L	8876	17346.3	26910	52	62.8	76	11.0	10.6
			E-NL^*	68817	79628.1	96792	3	5.7	8		3.4
			DRTA-L	171886	577082.6	1234835	7	8.9	11	79.0	44.1
10_8_20	56.5	101.7	NRTA-L	14756	20309.8	29886	65	83.1	97	11.0	13.5
			E-NL^*	78642	109356.2	191550	4	5.9	7		4.6
			DRTA-L	354175	1259531.4	2848321	7	9.7	13	101.7	122.3
10_10_20	64.1	109.2	NRTA-L	12078	26558.5	56840	67	92.0	118	11.0	24.4
			E-NL^*	100816	138915.4	174746	4	6.0	8		6.5
			DRTA-L	807271	1632323.2	2763231	8	10.5	13	109.2	176.0
12_4_20	42.1	127.1	NRTA-L	8652	14968.3	21096	50	66.0	80	13.0	7.8
			E-NL^*	47899	74348.9	111755	4	6.0	8		3.1
			DRTA-L	88567	1071517.5	1958611	10	12.5	16	127.1	117.8
14_4_20	49.5	130.3	NRTA-L	10290	21305.2	38372	58	73.6	85	15.1	13.6
			E-NL^*	59997	101304.0	145188	4	6.7	9		4.3
			DRTA-L	422722	1014350.4	3067064	11	11.8	14	130.3	97.1
16_4_20	56.2	325.9	NRTA-L	18258	41004.4	132928	73	85.5	103	17.1	51.0
			E-NL^*	99819	128872.7	182818	3	6.2	9		5.8
			DRTA-L	408270	3883803.1	13490655	10	15.0	20	325.9	909.8
18_4_20	61.8	330.7	NRTA-L	14256	35404.0	61464	73	97.5	128	19.4	34.3
			E-NL^*	140486	178679.7	254072	7	8.6	11		7.9
			DRTA-L	613914	4989360.0	11884279	11	16.6	23	330.7	899.5
20_4_20	68.7	422.7	NRTA-L	26255	56817.1	135675	86	104.0	117	21.0	84.1
			E-NL^*	154600	211740.0	260107	6	7.6	10		9.9
			DRTA-L	2921594	6720561.6	13160576	17	18.6	22	422.7	1392.3

Figure – Experimental results on the randomly generated NRTAs.

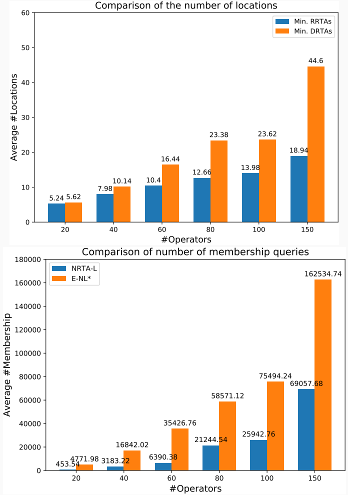


Figure – Comparison results on randomly generated rational regular expressions.

Thanks a lot!