State-based opacity of real-time automata

Kuize Zhang

Control Systems Group
Technical University of Berlin
10587 Berlin, Germany

13 July 2021
AUTOMATA 2021, Marseille, France
kuize.zhang@campus.tu-berlin.de
Content

1. Review of opacity results in the literature

2. Notation in real-time automata

3. Main results
   - The definitions of opacity
   - The notions of observer and reverse observer
   - Sufficient and necessary conditions for opacity

4. Conclusion
Background

- Opacity is a confidentiality property (firstly proposed by (Mazaré, 2004)) used to characterize information flow security, and has been widely used to describe all kinds of scenarios in security/privacy problems.
Opacity is a confidentiality property (firstly proposed by (Mazaré, 2004)) used to characterize information flow security, and has been widely used to describe all kinds of scenarios in security/privacy problems.

It describes whether a labeled (aka partially-observed) system can forbid an external intruder from making sure whether some secrets have been visited,
Background

- Opacity is a confidentiality property (firstly proposed by (Mazaré, 2004)) used to characterize information flow security, and has been widely used to describe all kinds of scenarios in security/privacy problems.

- It describes whether a labeled (aka partially-observed) system can forbid an external intruder from making sure whether some secrets have been visited, given that the intruder knows complete knowledge of the system's structure but can only see outputs generated by the system.
A general framework for opacity

Run-based opacity (Bryans et al., 2008)

\[
q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} q_n \quad (\forall \text{ secret run})
\]

\[
q'_0 \xrightarrow{e'_1} q'_1 \xrightarrow{e'_2} \cdots \xrightarrow{e'_m} q'_m \quad (\exists \text{ non-secret run})
\]

s.t. \( \ell(e_1 \ldots e_n) = \ell(e'_1 \ldots e'_m) \) (the same label seq.)
Two special classes of opacity: I

Language-based (aka trace-based) opacity

\[ e_1 \ldots e_n \quad (\forall \text{ secret trace}) \]

\[ e'_1 \ldots e'_m \quad (\exists \text{ non-secret trace}) \]

s.t. \( \ell(e_1 \ldots e_n) = \ell(e'_1 \ldots e'_m) \) (the same label seq.)
Two special classes of opacity: I

Language-based (aka trace-based) opacity

\[ e_1 \ldots e_n \quad (\forall \text{ secret trace}) \]
\[ e'_1 \ldots e'_m \quad (\exists \text{ non-secret trace}) \]
\[ \text{s.t. } \ell(e_1 \ldots e_n) = \ell(e'_1 \ldots e'_m) \quad (\text{the same label seq.}) \]

Verification results in untimed automata
Two special classes of opacity: I

Language-based (aka trace-based) opacity

\[ e_1 \ldots e_n \quad (\forall \text{ secret trace}) \]
\[ e'_1 \ldots e'_m \quad (\exists \text{ non-secret trace}) \]
\[ \text{s.t. } \ell(e_1 \ldots e_n) = \ell(e'_1 \ldots e'_m) \quad (\text{the same label seq.}) \]

Verification results in untimed automata

- **undecidable** in labeled finite automata (LFAs) with $\epsilon$-labeling functions (Bryans et al., 2008)
Two special classes of opacity: I

Language-based (aka trace-based) opacity

\[ e_1 \ldots e_n \quad (\forall \text{ secret trace}) \]
\[ e'_1 \ldots e'_m \quad (\exists \text{ non-secret trace}) \]
\[ \text{s.t. } \ell(e_1 \ldots e_n) = \ell(e'_1 \ldots e'_m) \quad (\text{the same label seq.}) \]

Verification results in untimed automata

- **undecidable** in labeled finite automata (LFAs) with \( \epsilon \)-labeling functions (Bryans et al., 2008)
- **decidable in EXPTIME** in LFAs when secret languages and non-secrete languages are regular (Lin, 2011)
Two special classes of opacity: II

State-based opacity (specified according to the time when secrets visited)

\[ q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \cdots \xrightarrow{e_i} q_i \xrightarrow{e_{i+1}} \cdots \xrightarrow{e_n} q_n \quad (\forall \text{ secret state}) \]

\[ q'_0 \xrightarrow{e'_1} q'_1 \xrightarrow{e'_2} \cdots \xrightarrow{e'_j} q'_j \xrightarrow{e'_{j+1}} \cdots \xrightarrow{e'_m} q'_m \quad (\exists \text{ non-secret state}) \]

\[ \text{s.t. } \ell(e_1 \ldots e_i) = \ell(e'_1 \ldots e'_j) =: \gamma_1 \]

\[ \ell(e_{i+1} \ldots e_n) = \ell(e'_{j+1} \ldots e'_m) =: \gamma_2 \quad (\text{the same label seq.}) \]
Two special classes of opacity: II

State-based opacity (specified according to the time when secrets visited)

\[ q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \ldots \xrightarrow{e_i} q_i \xrightarrow{e_{i+1}} \ldots \xrightarrow{e_n} q_n \quad (\forall \text{ secret state}) \]

\[ q'_0 \xrightarrow{e'_1} q'_1 \xrightarrow{e'_2} \ldots \xrightarrow{e'_j} q'_j \xrightarrow{e'_{j+1}} \ldots \xrightarrow{e'_{m}} q'_m \quad (\exists \text{ non-secret state}) \]

\[ \text{s.t. } \ell(e_1 \ldots e_i) = \ell(e'_1 \ldots e'_j) =: \gamma_1 \]

\[ \ell(e_{i+1} \ldots e_n) = \ell(e'_{j+1} \ldots e'_m) =: \gamma_2 \quad (\text{the same label seq.}) \]

Verification results in untimed automata (plenty of)

Initial-state opacity (ISO) \((i = j = 0)\), current-state opacity (CSO, \(i = n, j = m\)), infinite-step opacity (InfSO), and \(K\)-step opacity (KSO, \(|\gamma_2| \leq K\)) are \text{PSPACE}-complete in LFAs, and equivalent (Saboori and Hadjicostis, 2013) (Cassez, Dubreil, and Marchand, 2009) (Wu and Lafortune, 2013).
Language-based opacity
Language-based opacity

- **decidable** in (labeled) real-time automata when secret languages and non-secrete languages are those recognized by real-time automata (Wang, Zhan, and An, 2018)
Results in timed automata (rare)

Language-based opacity

- **decidable** in (labeled) real-time automata when secret languages and non-secrete languages are those recognized by real-time automata (Wang, Zhan, and An, 2018)

State-based opacity
Results in timed automata (rare)

Language-based opacity

- **decidable** in (labeled) real-time automata when secret languages and non-secrete languages are those **recognized** by real-time automata (Wang, Zhan, and An, 2018)

State-based opacity

- CSO is **undecidable** in time-deterministic event recording automata (Cassez, 2009).
Results in timed automata (rare)

Language-based opacity

- **decidable** in (labeled) real-time automata when secret languages and non-secret languages are those recognized by real-time automata (Wang, Zhan, and An, 2018)

State-based opacity

- CSO is **undecidable** in time-deterministic event recording automata (Cassez, 2009).
- ISO is **decidable** in real-time automata (Wang, Zhan, and An, 2018).
Results in timed automata (rare)

Language-based opacity
- **decidable** in (labeled) real-time automata when secret languages and non-secrete languages are those recognized by real-time automata (Wang, Zhan, and An, 2018)

State-based opacity
- CSO is **undecidable** in time-deterministic event recording automata (Cassez, 2009).
- ISO is **decidable** in real-time automata (Wang, Zhan, and An, 2018).
- ISO, CSO, KSO, InfSO in real-time automata are decidable with 2-EXPTIME upper bounds (Zhang, 2021)
Content

1. Review of opacity results in the literature

2. Notation in real-time automata

3. Main results
   - The definitions of opacity
   - The notions of observer and reverse observer
   - Sufficient and necessary conditions for opacity

4. Conclusion
A (labeled) real-time automaton (RTA) is a tuple

\[ \mathcal{A} = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell), \]

where
A (labeled) real-time automaton (RTA) is a tuple

\[ \mathcal{A} = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell) \],

where

- \( Q \) is a finite set of states,

Kuize Zhang (TUB)
A (labeled) real-time automaton (RTA) is a tuple

\[ \mathcal{A} = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell) \],

where

- \( Q \) is a finite set of states,
- \( E \) is a finite set of events,
A (labeled) real-time automaton (RTA) is a tuple

$$\mathcal{A} = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell),$$

where

- $Q$ is a finite set of states,
- $E$ is a finite set of events,
- $Q_0 \subseteq Q$ is a set of initial states,
A (labeled) real-time automaton (RTA) is a tuple

\[ A = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell), \]

where

- \( Q \) is a finite set of states,
- \( E \) is a finite set of events,
- \( Q_0 \subseteq Q \) is a set of initial states,
- \( \Delta \subseteq Q \times E \times Q \) is the transition relation (elements of \( \Delta \) are transitions),
A (labeled) real-time automaton (RTA) is a tuple

\[ \mathcal{A} = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell), \]

where

- \( Q \) is a finite set of states,
- \( E \) is a finite set of events,
- \( Q_0 \subset Q \) is a set of initial states,
- \( \Delta \subset Q \times E \times Q \) is the transition relation (elements of \( \Delta \) are transitions),
- \( \mu \) assigns to each transition \((q, e, q') \in \Delta\) (also written as \( q \xrightarrow{e} q' \)) a nonempty interval \( \mu(e)_{qq'} \in \mathbb{R}_{\geq 0} \) with left endpoint and right endpoint being \( a \in \mathbb{Q}_{\geq 0} \) and \( b \in \mathbb{Q}_{\geq 0} \cup \{+\infty\} \), respectively,
A (labeled) real-time automaton (RTA) is a tuple

\[ \mathcal{A} = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell), \]

where

- \( Q \) is a finite set of states,
- \( E \) is a finite set of events,
- \( Q_0 \subset Q \) is a set of initial states,
- \( \Delta \subset Q \times E \times Q \) is the transition relation (elements of \( \Delta \) are transitions),
- \( \mu \) assigns to each transition \( (q, e, q') \in \Delta \) (also written as \( q \xrightarrow{e} q' \)) a nonempty interval \( \mu(e)_{qq'} \) of \( \mathbb{R}_{\geq 0} \) with left endpoint and right endpoint being \( a \in \mathbb{Q}_{\geq 0} \) and \( b \in \mathbb{Q}_{\geq 0} \cup \{+\infty\} \), respectively,
- \( \Sigma \) is a finite set of labels/outputs,
A (labeled) real-time automaton (RTA) is a tuple

\[ A = (Q, E, Q_0, \Delta, \mu, \Sigma, \ell), \]

where

- \( Q \) is a finite set of states,
- \( E \) is a finite set of events,
- \( Q_0 \subset Q \) is a set of initial states,
- \( \Delta \subset Q \times E \times Q \) is the transition relation (elements of \( \Delta \) are transitions),
- \( \mu \) assigns to each transition \( (q, e, q') \in \Delta \) (also written as \( q \xrightarrow{e} q' \)) a nonempty interval \( \mu(e)_{qq'} \) of \( \mathbb{R}_{\geq 0} \) with left endpoint and right endpoint being \( a \in \mathbb{Q}_{\geq 0} \) and \( b \in \mathbb{Q}_{\geq 0} \cup \{+\infty\} \), respectively,
- \( \Sigma \) is a finite set of labels/outputs,
- \( \ell : E \to \Sigma \cup \{\epsilon\} \) is the labeling function.
observable events $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
Notation in real-time automata

- **observable events** $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- **unobservable events** $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- observable events $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- unobservable events $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- observable transition $(q, e, q') \in \Delta$ with $e \in E_o$
- Observable events $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- Unobservable events $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- Observable transition $(q, e, q') \in \Delta$ with $e \in E_o$
- Unobservable transition $(q, e, q') \in \Delta$ with $e \in E_{uo}$
- Observable events $E_o = \{ e \in E \mid \ell(e) \in \Sigma \}$
- Unobservable events $E_{uo} = \{ e \in E \mid \ell(e) = \epsilon \}$
- Observable transition $(q, e, q') \in \Delta$ with $e \in E_o$
- Unobservable transition $(q, e, q') \in \Delta$ with $e \in E_{uo}$

- $\ell$ extended to $E \times \mathbb{R}_{\geq 0}$: $\ell((e, t)) = \begin{cases} (\ell(e), t) & \text{if } e \in E_o, \\ \epsilon & \text{if } e \in E_{uo}. \end{cases}$
- observable events $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- unobservable events $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- observable transition $(q, e, q') \in \Delta$ with $e \in E_o$
- unobservable transition $(q, e, q') \in \Delta$ with $e \in E_{uo}$
- $\ell$ extended to $E \times \mathbb{R}_{\geq 0}$: $\ell((e, t)) = \begin{cases} (\ell(e), t) & \text{if } e \in E_o, \\ \epsilon & \text{if } e \in E_{uo}. \end{cases}$
- $\ell$ recursively extended to $E^*$ and also to $(E \times \mathbb{R}_{\geq 0})^*$ analogously.
Notation in real-time automata

- **observable events** $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- **unobservable events** $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- **observable transition** $(q, e, q') \in \Delta$ with $e \in E_o$
- **unobservable transition** $(q, e, q') \in \Delta$ with $e \in E_{uo}$
- $\ell$ extended to $E \times \mathbb{R}_{\geq 0}$: $\ell((e, t)) = \begin{cases} (\ell(e), t) & \text{if } e \in E_o, \\ \epsilon & \text{if } e \in E_{uo}. \end{cases}$
- $\ell$ recursively extended to $E^*$ and also to $(E \times \mathbb{R}_{\geq 0})^*$ analogously.
- A path is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} q_n$, where $n \in \mathbb{Z}_+$, $(q_{i-1}, e_i, q_i) \in \Delta$ for all $i \in [1, n]$. 

Kuize Zhang (TUB)
• observable events $E_0 = \{ e \in E | \ell(e) \in \Sigma \}$
• unobservable events $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
• observable transition $(q, e, q') \in \Delta$ with $e \in E_0$
• unobservable transition $(q, e, q') \in \Delta$ with $e \in E_{uo}$

$\ell$ extended to $E \times \mathbb{R}_{\geq 0}$: $\ell((e, t)) = \begin{cases} (\ell(e), t) & \text{ if } e \in E_0, \\ \epsilon & \text{ if } e \in E_{uo}. \end{cases}$

$\ell$ recursively extended to $E^*$ and also to $(E \times \mathbb{R}_{\geq 0})^*$ analogously.

A path is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} q_n$, where $n \in \mathbb{Z}_+, (q_{i-1}, e_i, q_i) \in \Delta$ for all $i \in [1, n]$.

A run is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1/t_1} q_1 \xrightarrow{e_2/t_2} \cdots \xrightarrow{e_n/t_n} q_n =: \pi$, where $n \in \mathbb{Z}_+, (q_{i-1}, e_i, q_i) \in \Delta$, $t_i \in \mu(e_i)_{q_{i-1}q_i}$ for all $i \in [1, n]$. 
- observable events $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- unobservable events $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- observable transition $(q, e, q') \in \Delta$ with $e \in E_o$
- unobservable transition $(q, e, q') \in \Delta$ with $e \in E_{uo}$

$\ell$ extended to $E \times \mathbb{R}_{\geq 0}$: $\ell((e, t)) = \begin{cases} (\ell(e), t) & \text{if } e \in E_o, \\ \epsilon & \text{if } e \in E_{uo}. \end{cases}$

$\ell$ recursively extended to $E^*$ and also to $(E \times \mathbb{R}_{\geq 0})^*$ analogously.

A path is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} q_n$, where $n \in \mathbb{Z}_+, (q_{i-1}, e_i, q_i) \in \Delta$ for all $i \in [1, n]$.

A run is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1/t_1} q_1 \xrightarrow{e_2/t_2} \cdots \xrightarrow{e_n/t_n} q_n = \pi$, where $n \in \mathbb{Z}_+, (q_{i-1}, e_i, q_i) \in \Delta$, $t_i \in \mu(e_i)q_{i-1}q_i$ for all $i \in [1, n]$.

The timed word of $\pi$ is defined by $\tau(\pi) = (e_1, t'_1)(e_2, t'_2) \cdots (e_n, t'_n)$, where $t'_i = \sum_{k=1}^{i} t_k$ for all $i \in [1, n]$. 
Notation in real-time automata

- **observable events** $E_o = \{ e \in E | \ell(e) \in \Sigma \}$
- **unobservable events** $E_{uo} = \{ e \in E | \ell(e) = \epsilon \}$
- **observable transition** $(q, e, q') \in \Delta$ with $e \in E_o$
- **unobservable transition** $(q, e, q') \in \Delta$ with $e \in E_{uo}$

- $\ell$ extended to $E \times \mathbb{R}_{\geq 0}$: $\ell((e, t)) = \begin{cases} (\ell(e), t) & \text{if } e \in E_o, \\ \epsilon & \text{if } e \in E_{uo}. \end{cases}$

- $\ell$ recursively extended to $E^*$ and also to $(E \times \mathbb{R}_{\geq 0})^*$ analogously.

- A **path** is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1} q_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} q_n$, where $n \in \mathbb{Z}_+$, $(q_{i-1}, e_i, q_i) \in \Delta$ for all $i \in [1, n]$.

- A **run** is either $\epsilon$ or a sequence $q_0 \xrightarrow{e_1/t_1} q_1 \xrightarrow{e_2/t_2} \cdots \xrightarrow{e_n/t_n} q_n =: \pi$, where $n \in \mathbb{Z}_+$, $(q_{i-1}, e_i, q_i) \in \Delta$, $t_i \in \mu(e_i)_{q_{i-1}q_i}$ for all $i \in [1, n]$.

- The **timed word** of $\pi$ is defined by $\tau(\pi) = (e_1, t'_1)(e_2, t'_2) \cdots (e_n, t'_n)$, where $t'_i = \sum_{k=1}^{i} t_k$ for all $i \in [1, n]$.

- The **weight** $W_{\pi}$ of $\pi$ is defined by $t'_n$. 
Example 1

Consider the RTA $\mathcal{A}_1$:

![Diagram of RTA $\mathcal{A}_1$]

\textbf{Figure 1}: An RTA $\mathcal{A}_1$, $q_0$ is the initial state, $a$ is an observable event, $\ell(a) = a$, $u$ is unobservable, $\ell(u) = \epsilon$. 
Example 1

Consider the RTA $A_1$:

![Diagram of RTA $A_1$]

$\begin{align*}
q_0 &\xrightarrow{a} q_1 \\
q_3 &\xrightarrow{a} q_1 \\
q_0 &\xrightarrow{a} q_2 \\
q_2 &\xrightarrow{a} q_4 \\
q_5 &\xrightarrow{u} q_3 \\
q_3 &\xrightarrow{u} q_5
\end{align*}$

Figure 1: An RTA $A_1$, $q_0$ is the initial state, $a$ is an observable event, $\ell(a) = a$, $u$ is unobservable, $\ell(u) = \epsilon$.

$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_3 \xrightarrow{u} q_5,$

(path)
Example 1

Consider the RTA $A_1$:

![Diagram of RTA $A_1$]

Figure 1: An RTA $A_1$, $q_0$ is the initial state, $a$ is an observable event, $\ell(a) = a$, $u$ is unobservable, $\ell(u) = \epsilon$.

\[
q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_3 \xrightarrow{u} q_5, \quad \text{(path)}
\]

\[
\pi = q_0 \xrightarrow{a/2} q_1 \xrightarrow{a/1} q_3 \xrightarrow{u/1} q_5, \quad \text{(run)}
\]
Example 1

Consider the RTA $A_1$:

$q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_3 \xrightarrow{u} q_5$,  \hspace{1cm} (path)

$\pi = q_0 \xrightarrow{a/2} q_1 \xrightarrow{a/1} q_3 \xrightarrow{u/1} q_5$,  \hspace{1cm} (run)

$\tau(\pi) = (a, 2)(a, 3)(u, 4)$,  \hspace{1cm} (timed word)
Example 1

Consider the RTA $A_1$:

![Diagram of RTA $A_1$]

$\pi = q_0 \xrightarrow{a/2} q_1 \xrightarrow{a/1} q_3 \xrightarrow{u/1} q_5,$

(path)

$\tau(\pi) = (a, 2)(a, 3)(u, 4),$

(timed word)

$WT_\pi = 4,$

(weight)

Figure 1: An RTA $A_1$, $q_0$ is the initial state, $a$ is an observable event, $\ell(a) = a$, $u$ is unobservable, $\ell(u) = \epsilon$. 

Kuize Zhang (TUB)
Example 1

Consider the RTA $A_1$:

![Diagram of RTA $A_1$]

Figure 1: An RTA $A_1$, $q_0$ is the initial state, $a$ is an observable event, $\ell(a) = a$, $u$ is unobservable, $\ell(u) = \epsilon$.

\[
\begin{align*}
q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_3 \xrightarrow{u} q_5, && \text{(path)} \\
\pi = q_0 \xrightarrow{a/2} q_1 \xrightarrow{a/1} q_3 \xrightarrow{u/1} q_5, && \text{(run)} \\
\tau(\pi) = (a, 2)(a, 3)(u, 4), && \text{(timed word)} \\
WT_{\pi} = 4, && \text{(weight)} \\
\ell(\tau(\pi)) = (a, 2)(a, 3). && \text{(timed label seq.)}
\end{align*}
\]
A run $\pi$ is called **instantaneous** if $WT_\pi = 0$, called **noninstantaneous** if $WT_\pi > 0$. 
A run $\pi$ is called **instantaneous** if $WT_\pi = 0$, called **noninstantaneous** if $WT_\pi > 0$.

A run $\pi$ is called **unobservable** if $\ell(e_1 \ldots e_n) = \epsilon$, called **observable** if $\ell(e_1 \ldots e_n) \in \Sigma^+$. 
A run $\pi$ is called **instantaneous** if $WT_\pi = 0$, called **noninstantaneous** if $WT_\pi > 0$.

A run $\pi$ is called **unobservable** if $\ell(e_1 \ldots e_n) = \epsilon$, called **observable** if $\ell(e_1 \ldots e_n) \in \Sigma^+$.

Given $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^*$, $[\gamma]$ denotes the set of runs $\pi$ of $A$ starting from initial states such that $\ell(\tau(\pi)) = \gamma$. $\text{last}([\gamma])$
A run $\pi$ is called **instantaneous** if $WT_\pi = 0$, called **noninstantaneous** if $WT_\pi > 0$.

A run $\pi$ is called **unobservable** if $\ell(e_1 \ldots e_n) = \epsilon$, called **observable** if $\ell(e_1 \ldots e_n) \in \Sigma^+$. 

Given $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^*$, $[\gamma]$ denotes the set of runs $\pi$ of $A$ starting from initial states such that $\ell(\tau(\pi)) = \gamma$. $\text{last}([\gamma])$

$$\text{interm}(\gamma_1, \gamma_2) = \{ q \in Q | (\exists \text{ runs } \pi_1, \pi_2)[(\text{init}(\pi_1) \in Q_0) \land (\text{last}(\pi_1) = \text{init}(\pi_2) = q) \land (\ell(\tau(\pi_1)) = \gamma_1) \land (\ell(\tau(\pi_1 \pi_2)) = \gamma_1 \gamma_2) \land (WT_{\pi_1} = \text{last}_R(\gamma_1)) \land (WT_{\pi_2} = \text{last}_R(\gamma_2) - \text{last}_R(\gamma_1))\}$$: the set of states $A$ can be in when $A$ has just generated timed label seq. $\gamma_1$, given that the current timed label seq. is $\gamma_1 \gamma_2 \in (\Sigma \times \mathbb{R}_{\geq 0})^*$.
Example 2 (cont. $A_1$)
Example 2 (cont. $A_1$)

$$\text{last}([(a, 2)]) = \{q_1, q_2\},$$
Example 2 (cont. $A_1$)

$$\text{last}([(a, 2)]) = \{q_1, q_2\},$$

$$\text{last}([(a, 2)(a, 3)]) = \{q_3, q_4, q_5\},$$
Example 2 (cont. $A_1$)

\[
\begin{align*}
&\text{last}([(a, 2)]) = \{q_1, q_2\}, \\
&\text{last}([(a, 2)(a, 3)]) = \{q_3, q_4, q_5\}, \\
&\text{interm}(A_1, (a, 2), (a, 3)) = \{q_1, q_2\}.
\end{align*}
\]
Notation in real-time automata

Current-state estimate

For $\mathcal{A}$, $x \subset Q$, and $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^*$, the current-state estimate is

$$M(\mathcal{A}, \gamma|x) := \{ q \in Q | (\exists q_0 \in x)(\exists n \in \mathbb{N})(\exists m \in \mathbb{N})$$

$$\left( \exists \text{ a run } \pi = q_0 \xrightarrow{e_1/t_1} \cdots \xrightarrow{e_n/t_n} q_n \xrightarrow{e_{n+1}/0} \cdots \xrightarrow{e_{n+m}/0} q \right)$$

$$[(e_n \in E_o) \land (e_{n+1} \ldots e_{n+m} \in (E_{uo})^*) \land \ell(\tau(\pi)) = \gamma] \}.$$
For $\mathcal{A}$, $x \subset Q$, and $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^*$, the current-state estimate is

$$M(\mathcal{A}, \gamma|x) := \{q \in Q | (\exists q_0 \in x)(\exists n \in \mathbb{N})(\exists m \in \mathbb{N})$$

$$\left( \exists \text{ a run } \pi = q_0 \xrightarrow{e_1/t_1} \cdots \xrightarrow{e_n/t_n} q_n \xrightarrow{e_{n+1}/0} \cdots \xrightarrow{e_{n+m}/0} q \right)$$

$$[(e_n \in E_o) \land (e_{n+1} \cdots e_{n+m} \in (E_{uo})^*) \land \ell(\tau(\pi)) = \gamma] \}.$$ 

$M(\mathcal{A}, \gamma)$ denotes the set of states $\mathcal{A}$ can be in when $\gamma$ has been generated.
For $\mathcal{A}$, $x \subset Q$, and $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^*$, the current-state estimate is

$$
\mathcal{M}(\mathcal{A}, \gamma|x) := \{ q \in Q | (\exists q_0 \in x)(\exists n \in \mathbb{N})(\exists m \in \mathbb{N})
\left(\exists \text{ a run } \pi = q_0 \xrightarrow{e_1/\tau_1} \cdots \xrightarrow{e_n/\tau_n} q_n \xrightarrow{e_{n+1}/0} \cdots \xrightarrow{e_{n+m}/0} q \right)
$$

$$
[(e_n \in E_o) \land (e_{n+1} \cdots e_{n+m} \in (E_{uo})^*) \land \ell(\tau(\pi)) = \gamma] \}.
$$

$\mathcal{M}(\mathcal{A}, \gamma)$ denotes the set of states $\mathcal{A}$ can be in when $\gamma$ has been generated.

$\mathcal{M}(\mathcal{A}, \gamma|Q_0) =: \mathcal{M}(\mathcal{A}, \gamma) \subset \text{last}([\gamma])$ for $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^*$. 


Example 3 (cont. $A_1$)

Notation in real-time automata

$A_1$
Example 3 (cont. $A_1$)

$$\text{last}([(a, 2)]) = \{q_1, q_2\},$$
Example 3 (cont. $\mathcal{A}_1$)

\[
\text{last}([\langle a, 2 \rangle]) = \{q_1, q_2\}, \\
\mathcal{M}(\mathcal{A}_1, (a, 2)) = \{q_1, q_2\},
\]
Example 3 (cont. $A_1$)

$\text{last}([\{(a, 2)\}]) = \{q_1, q_2\},$

$\mathcal{M}(A_1, (a, 2)) = \{q_1, q_2\},$

$\text{last}([\{(a, 2)(a, 3)\}]) = \{q_3, q_4, q_5\},$
Example 3 (cont. $A_1$)

\[
\begin{array}{ccccccc}
q_5 & \xleftarrow{u/\{1, 2\}} & q_3 & \xrightarrow{a/\{1\}} & q_1 & \xleftarrow{a/\{1, 3\}} & q_0 & \xrightarrow{a/\{1, 2\}} & q_2 & \xrightarrow{a/\{1, 2\}} & q_4 \\
\end{array}
\]

\[
\text{last}([(a, 2)]) = \{q_1, q_2\},
\]
\[
\mathcal{M}(A_1, (a, 2)) = \{q_1, q_2\},
\]
\[
\text{last}([(a, 2)(a, 3)]) = \{q_3, q_4, q_5\},
\]
\[
\mathcal{M}(A_1, (a, 2)(a, 3)) = \{q_3, q_4\},
\]
Example 3 (cont. $A_1$)

\[
\begin{align*}
q_5 & \quad u/[1, 2] \quad q_3 \\
& \quad a/\{1\} \\
& \quad a/[1, 3] \\
& \quad a/[1, 2] \\
& \quad a/[1, 2] \\
q_0 & \quad \downarrow \\
q_2 & \quad \downarrow \\
q_4 & \quad \downarrow \\
q_5 & \quad q_3 \\
& \quad q_1 \\
& \quad q_0 \\
& \quad q_2 \\
& \quad q_4
\end{align*}
\]

\[
\begin{align*}
\text{last}([((a, 2)]) &= \{q_1, q_2\}, \\
\mathcal{M}(A_1, (a, 2)) &= \{q_1, q_2\}, \\
\text{last}([((a, 2)(a, 3)]) &= \{q_3, q_4, q_5\}, \\
\mathcal{M}(A_1, (a, 2)(a, 3)) &= \{q_3, q_4\}, \\
\mathcal{M}(A_1, (a, 2)(a, 3)) &\nsubseteq \text{last}([((a, 2)(a, 3)]).
\end{align*}
\]
Content

1. Review of opacity results in the literature

2. Notation in real-time automata

3. Main results
   - The definitions of opacity
   - The notions of observer and reverse observer
   - Sufficient and necessary conditions for opacity

4. Conclusion
Language generated by RTA $\mathcal{A}$: $L(\mathcal{A}) = \{ \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^* | M(\mathcal{A}, \gamma) \neq \emptyset \}$
Language generated by RTA $\mathcal{A}$: $\mathcal{L}(\mathcal{A}) = \{ \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^* | \mathcal{M}(\mathcal{A}, \gamma) \neq \emptyset \}$

Specify a subset $Q_S \subset Q$ of secret states.
Main results  The definitions of opacity

**Language generated by RTA \( \mathcal{A} \):** \( \mathcal{L}(\mathcal{A}) = \{ \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^* | \mathcal{M}(\mathcal{A}, \gamma) \neq \emptyset \} \)

Specify a subset \( Q_S \subset Q \) of secret states.

**Definition 4 (ISO)**

An RTA \( \mathcal{A} \) is called **initial-state opaque (ISO)** w.r.t. \( Q_S \) if for every \( \gamma \in \mathcal{L}(\mathcal{A}) \), \( \text{init}([\gamma]) \notin Q_S \).
Language generated by RTA $\mathcal{A}$: $\mathcal{L}(\mathcal{A}) = \{ \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^* | \mathcal{M}(\mathcal{A}, \gamma) \neq \emptyset \}$

Specify a subset $Q_S \subset Q$ of secret states.

Definition 4 (ISO)

An RTA $\mathcal{A}$ is called initial-state opaque (ISO) w.r.t. $Q_S$ if for every $\gamma \in \mathcal{L}(\mathcal{A})$, $\text{init}([\gamma]) \not\subset Q_S$.

ISO means that when observing a timed label sequence $\gamma \in \mathcal{L}(\mathcal{A})$, not all possible initial states are secret, so that one cannot make sure whether the initial state is secret.
Language generated by RTA $A$: $\mathcal{L}(A) = \{ \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^* | M(A, \gamma) \neq \emptyset \}$

Specify a subset $Q_S \subset Q$ of secret states.

Definition 4 (ISO)

An RTA $A$ is called initial-state opaque (ISO) w.r.t. $Q_S$ if for every $\gamma \in \mathcal{L}(A)$, $\text{init}([\gamma]) \not\subset Q_S$.

ISO means that when observing a timed label sequence $\gamma \in \mathcal{L}(A)$, not all possible initial states are secret, so that one cannot make sure whether the initial state is secret.

Definition 5 (CSO)

An RTA $A$ is called current-state opaque (CSO) w.r.t. $Q_S$ if for every $\gamma \in \mathcal{L}(A)$, in $M(A, \gamma)$ there exists at least one non-eventually-secret state.
Definition 6

A state $q$ of an RTA $A$ is called *eventually secret* if either (1) $q$ is secret or (2) there is an unobservable path starting from $q$ and along each of such paths at least one secret state will be visited.
Definition 6

A state $q$ of an RTA $\mathcal{A}$ is called **eventually secret** if either (1) $q$ is secret or (2) there is an unobservable path starting from $q$ and along each of such paths at least one secret state will be visited.

Proposition 1

A state $q$ is not eventually secret iff (1) $q \notin Q_S$ and (2) either there is no unobservable path from $q$ or there is an unobservable path from $q$ without any secret state that either ends at a **dead** state or contains a cycle.
Definition 6

A state $q$ of an RTA $A$ is called **eventually secret** if either (1) $q$ is secret or (2) there is an unobservable path starting from $q$ and along each of such paths at least one secret state will be visited.

Proposition 1

A state $q$ is not eventually secret iff (1) $q \notin Q_S$ and (2) either there is no unobservable path from $q$ or there is an unobservable path from $q$ without any secret state that either ends at a **dead** state or contains a cycle.

Example 7 (cont. $A_1$)

Let $Q_S = \{q_5\}$. Then $q_3$ is eventually secret because of the unique unobservable path $q_3 \xrightarrow{u} q_5$ (with $q_5$ dead, i.e., no transition starts at $q_5$).
Example 8 (cont. $A_1$)

Let $Q_S = \{ q_5 \}$. 

```
Example 8 (cont. \( A_1 \))

Let \( Q_S = \{q_5\} \). In Example 7, we have shown \( q_3 \) is eventually secret.
Example 8 (cont. \( A_1 \))

Let \( Q_S = \{ q_5 \} \). In Example 7, we have shown \( q_3 \) is eventually secret. In addition, none of \( q_1, q_0, q_2, q_4 \) is eventually secret.
Example 8 (cont. $\mathcal{A}_1$)

Let $Q_S = \{q_5\}$. In Example 7, we have shown $q_3$ is eventually secret. In addition, none of $q_1$, $q_0$, $q_2$, $q_4$ is eventually secret. We have $\mathcal{M}(\mathcal{A}_1, (a, 3)(a, 4)) = \{q_3\}$ which does not contain any non-eventually-secret state.
Example 8 (cont. $A_1$)

Let $Q_5 = \{q_5\}$. In Example 7, we have shown $q_3$ is eventually secret. In addition, none of $q_1, q_0, q_2, q_4$ is eventually secret. We have $M(A_1, (a, 3)(a, 4)) = \{q_3\}$ which does not contain any non-eventually-secret state. Hence $A_1$ is not CSO w.r.t. $\{q_5\}$. 
Definition 9 (InfSO and KSO)

An RTA $\mathcal{A}$ is called infinite-step opaque (InfSO) w.r.t. $Q_S$ if for all $\gamma_1, \gamma_2 \in \mathcal{L}(\mathcal{A})$ such that $1 \leq |\gamma_2|$, $\text{interm}(\gamma_1, \gamma_2)$ contains at least one non-secret state $q$.

\begin{align*}
q_0 \xrightarrow{E^* E_o} q' \quad \gamma_1 & \xrightarrow{E^* E_o} q \quad \gamma_1 \gamma_2 \quad \gamma_1 \gamma_2 \\
q' \quad \text{inst. unobs.} & \xrightarrow{E^* E_o} q'' \quad \text{inst. unobs.}
\end{align*}

(*)
Definition 9 (InfSO and KSO)

An RTA $A$ is called **infinite-step opaque (InfSO)** w.r.t. $Q_S$ if for all $\gamma_1, \gamma_2 \in \mathcal{L}(A)$ such that $1 \leq |\gamma_2|$, $\text{interm}(\gamma_1, \gamma_2)$ contains at least one non-secret state $q$.

When observing $\gamma_1\gamma_2$ with $1 \leq |\gamma_2|$, one cannot make sure whether the state when $\gamma_1$ has just been generated is secret.
Definition 9 (InfSO and KSO)

An RTA $A$ is called infinite-step opaque (InfSO) ($K$-step opaque (KSO)) w.r.t. $Q_S$ if for all $\gamma_1 \gamma_2 \in \mathcal{L}(A)$ such that $1 \leq |\gamma_2| (\leq K)$, interm$(\gamma_1, \gamma_2)$ contains at least one non-secret state $q$.

When observing $\gamma_1 \gamma_2$ with $1 \leq |\gamma_2| (\leq K)$, one cannot make sure whether the state when $\gamma_1$ has just been generated is secret.
Example 10 (cont. $A_1$)

$$
\begin{array}{c}
\text{q}_5 \quad \text{u}/[1, 2] \quad \text{q}_3 \quad \text{a}/\{1\} \quad \text{q}_1 \quad \text{a}/[1, 3] \quad \text{q}_0 \quad \text{a}/[1, 2] \quad \text{q}_2 \quad \text{a}/[1, 2] \quad \text{q}_4
\end{array}
$$
Example 10 (cont. \(A_1\))

For \((a, 3)(a, 4)\), we only have

\[
\gamma_1 \gamma_2 = (a, 3)(a, 4)
\]

\[
\begin{align*}
q_0 &\xrightarrow{a/3} q_1 \\
q_1 &\xrightarrow{\epsilon} q_1 \\
q_1 &\xrightarrow{a/1} q_3
\end{align*}
\]

\[
\text{interm}(A_1, (a, 3), (a, 4)) = \{q_1\}
\]

which violates InfSO, i.e., \(A_1\) is not InfSO w.r.t. \(\{q_1\}\).
Definition 11

For an RTA $A$, we define its pre-observer as a deterministic automaton

$$A_{\text{pre obs}}^{\text{pre}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta_{\text{pre obs}}^{\text{pre}}), \quad (1)$$

where $X \subset 2^Q \setminus \{\emptyset\}$ is the state set, $\Sigma \times \mathbb{R}_{\geq 0}$ the (infinite) alphabet, $x_0 = \mathcal{M}(A, \epsilon) \in X$ the unique initial state, $\delta_{\text{obs}}^{\text{pre}} \subset X \times (\Sigma \times \mathbb{R}_{\geq 0}) \times X$ the transition relation: for all $x, x' \in X$ and $(\sigma, t) \in \Sigma \times \mathbb{R}_{\geq 0}$, $(x, (\sigma, t), x') \in \delta_{\text{obs}}^{\text{pre}}$ iff $x' = \mathcal{M}(A, (\sigma, t)|x)$. For all $x \in Q$ different from $x_0$, $x \in X$ iff there is $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^+$ such that $x = \mathcal{M}(A, \gamma)$. 
Definition 11

For an RTA \( \mathcal{A} \), we define its **pre-observer** as a deterministic automaton

\[
\mathcal{A}_\text{pre}^{\text{obs}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta^{\text{pre}}_\text{obs}),
\]

(1)

where \( X \subset 2^Q \setminus \{\emptyset\} \) is the state set, \( \Sigma \times \mathbb{R}_{\geq 0} \) the (infinite) alphabet, \( x_0 = \mathcal{M}(\mathcal{A}, \epsilon) \in X \) the unique initial state, \( \delta^{\text{pre}}_\text{obs} \subset X \times (\Sigma \times \mathbb{R}_{\geq 0}) \times X \) the transition relation: for all \( x, x' \in X \) and \( (\sigma, t) \in \Sigma \times \mathbb{R}_{\geq 0} \),

\( (x, (\sigma, t), x') \in \delta^{\text{pre}}_\text{obs} \) iff \( x' = \mathcal{M}(\mathcal{A}, (\sigma, t)|x) \). For all \( x \in Q \) different from \( x_0 \), \( x \in X \) iff there is \( \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^+ \) such that \( x = \mathcal{M}(\mathcal{A}, \gamma) \).
Main results

The notions of observer and reverse observer

**Definition 11**

For an RTA \( A \), we define its **pre-observer** as a deterministic automaton

\[
A_{\text{pre}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta_{\text{pre}}), \tag{1}
\]

where \( X \subset 2^Q \setminus \{\emptyset\} \) is the state set, \( \Sigma \times \mathbb{R}_{\geq 0} \) the (infinite) alphabet, \( x_0 = M(A, \epsilon) \in X \) the unique initial state, \( \delta_{\text{pre}} \subset X \times (\Sigma \times \mathbb{R}_{\geq 0}) \times X \) the transition relation: for all \( x, x' \in X \) and \( (\sigma, t) \in \Sigma \times \mathbb{R}_{\geq 0} \),

\[ (x, (\sigma, t), x') \in \delta_{\text{pre}} \text{ iff } x' = M(A, (\sigma, t)|x). \]

For all \( x \in Q \) different from \( x_0 \), \( x \in X \) iff there is \( \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^+ \) such that \( x = M(A, \gamma) \).

- After \( \delta_{\text{pre}} \) is recursively extended to \( \delta_{\text{obs}} \subset X \times (\Sigma \times \mathbb{R}_{\geq 0})^* \times X \), one has for all \( x \in X \) and \( (\sigma_1, t_1) \ldots (\sigma_n, t_n) =: \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^+ \),

\[ (x_0, \gamma, x) \in \delta_{\text{obs}} \text{ iff } M(A, \tau(\gamma)) = x, \text{ where } \]

\[ \tau(\gamma) = (\sigma_1, t_1)(\sigma_1, t_1 + t_2) \ldots (\sigma_n, t_1 + \cdots + t_n). \]
Definition 11

For an RTA $\mathcal{A}$, we define its pre-observer as a deterministic automaton

$$\mathcal{A}_{\text{obs}}^{\text{pre}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta_{\text{obs}}^{\text{pre}}),$$

where $X \subset 2^Q \setminus \{\emptyset\}$ is the state set, $\Sigma \times \mathbb{R}_{\geq 0}$ the (infinite) alphabet, $x_0 = \mathcal{M}(\mathcal{A}, \epsilon) \in X$ the unique initial state, $\delta_{\text{obs}}^{\text{pre}} \subset X \times (\Sigma \times \mathbb{R}_{\geq 0}) \times X$ the transition relation: for all $x, x' \in X$ and $(\sigma, t) \in \Sigma \times \mathbb{R}_{\geq 0}$,

$$(x, (\sigma, t), x') \in \delta_{\text{obs}}^{\text{pre}} \text{ iff } x' = \mathcal{M}(\mathcal{A}, (\sigma, t)|x).$$

For all $x \in Q$ different from $x_0$, $x \in X$ iff there is $\gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^+$ such that $x = \mathcal{M}(\mathcal{A}, \gamma)$.

- After $\delta_{\text{obs}}^{\text{pre}}$ is recursively extended to $\delta_{\text{obs}}^{\text{pre}} \subset X \times (\Sigma \times \mathbb{R}_{\geq 0})^* \times X$, one has for all $x \in X$ and $(\sigma_1, t_1) \ldots (\sigma_n, t_n) =: \gamma \in (\Sigma \times \mathbb{R}_{\geq 0})^+$,

$$(x_0, \gamma, x) \in \delta_{\text{obs}}^{\text{pre}} \text{ iff } \mathcal{M}(\mathcal{A}, \tau(\gamma)) = x,$$

where

$$\tau(\gamma) = (\sigma_1, t_1)(\sigma_1, t_1 + t_2)\ldots(\sigma_n, t_1 + \cdots + t_n).$$

- Alphabet $\Sigma \times \mathbb{R}_{\geq 0}$ is not finite, one cannot compute the whole $\mathcal{A}_{\text{obs}}^{\text{pre}}$. Next, we define observer $\mathcal{A}_{\text{obs}}$ as a finite sub-automaton of $\mathcal{A}_{\text{obs}}^{\text{pre}}$. 
Definition 12

For an RTA $\mathcal{A}$, consider its pre-observer $\mathcal{A}_{\text{obs}}^{\text{pre}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta_{\text{pre}}^{\text{obs}})$, we define its observer as a finite automaton

$$\mathcal{A}_{\text{obs}} = (X, \Sigma_{\text{obs}}, x_0, \delta_{\text{obs}}),$$

(2)

where $\Sigma_{\text{obs}}$ (resp., $\delta_{\text{obs}}$) is a finite subset of $\Sigma \times \mathbb{Q}_{\geq 0}$ (resp., $\delta_{\text{pre}}^{\text{obs}}$), such that if there exists a transition from $x \in X$ to $x' \in X$ in $\delta_{\text{pre}}^{\text{obs}}$ then at least one such transition belongs to $\delta_{\text{obs}}$. 

Remark 1

For an RTA $\mathcal{A}$, it may have more than one observer, because $\Sigma_{\text{obs}}$ may not be unique; but $X$ and $x_0$ must be unique.

For a labeled finite automaton, it has a unique observer, which is actually the powerset construction used for determinizing the automaton.
Definition 12

For an RTA $A$, consider its pre-observer $A_{\text{pre}}^{\text{obs}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta_{\text{pre}}^{\text{obs}})$, we define its observer as a finite automaton

$$A_{\text{obs}} = (X, \Sigma_{\text{obs}}, x_0, \delta_{\text{obs}}),$$

(2)

where $\Sigma_{\text{obs}}$ (resp., $\delta_{\text{obs}}$) is a finite subset of $\Sigma \times \mathbb{Q}_{\geq 0}$ (resp., $\delta_{\text{pre}}^{\text{obs}}$), such that if there exists a transition from $x \in X$ to $x' \in X$ in $\delta_{\text{pre}}^{\text{obs}}$ then at least one such transition belongs to $\delta_{\text{obs}}$.

Remark 1

For an RTA $A$, it may have more than one observer, because $\Sigma_{\text{obs}}$ may not be unique; but $X$ and $x_0$ must be unique. For a labeled finite automaton, it has a unique observer, which is actually the powerset construction used for determinizing the automaton.
Definition 12

For an RTA $\mathcal{A}$, consider its pre-observer $\mathcal{A}_{\text{pre}}^{\text{obs}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta_{\text{pre}}^{\text{obs}})$, we define its observer as a finite automaton

$$\mathcal{A}_{\text{obs}} = (X, \Sigma_{\text{obs}}, x_0, \delta_{\text{obs}}),$$

(2)

where $\Sigma_{\text{obs}}$ (resp., $\delta_{\text{obs}}$) is a finite subset of $\Sigma \times \mathbb{Q}_{\geq 0}$ (resp., $\delta_{\text{pre}}^{\text{obs}}$), such that if there exists a transition from $x \in X$ to $x' \in X$ in $\delta_{\text{pre}}^{\text{obs}}$ then at least one such transition belongs to $\delta_{\text{obs}}$.

Remark 1

- For an RTA $\mathcal{A}$, it may have more than one observer, because $\Sigma_{\text{obs}}$ may not be unique; but $X$ and $x_0$ must be unique.
Main results

The notions of observer and reverse observer

### Definition 12

For an RTA \( A \), consider its pre-observer \( A^{\text{pre}}_{\text{obs}} = (X, \Sigma \times \mathbb{R}_{\geq 0}, x_0, \delta^{\text{pre}}_{\text{obs}}) \), we define its observer as a finite automaton

\[
A_{\text{obs}} = (X, \Sigma_{\text{obs}}, x_0, \delta_{\text{obs}}),
\]

(2)

where \( \Sigma_{\text{obs}} \) (resp., \( \delta_{\text{obs}} \)) is a finite subset of \( \Sigma \times \mathbb{Q}_{\geq 0} \) (resp., \( \delta^{\text{pre}}_{\text{obs}} \)), such that if there exists a transition from \( x \in X \) to \( x' \in X \) in \( \delta^{\text{pre}}_{\text{obs}} \) then at least one such transition belongs to \( \delta_{\text{obs}} \).

### Remark 1

- **For an RTA \( A \), it may have more than one observer, because \( \Sigma_{\text{obs}} \) may not be unique; but \( X \) and \( x_0 \) must be unique.**
- **For a labeled finite automaton, it has a unique observer, which is actually the powerset construction used for determinizing the automaton.**
Example 13 (cont. $A_1$)

One of its observers is $q_0 \xrightarrow{a/[1,2]} q_1 \xrightarrow{a/[1,3]} q_0 \xrightarrow{a/[1,2]} q_2 \xrightarrow{a/[1,2]} q_4$.

Figure 2: $A_1 \text{obs}$.
Example 13 (cont. $A_1$)

One of its observers is

Figure 2: $A_{1\text{obs}}$. 
Theorem 14

An RTA $\mathcal{A}$ is CSO w.r.t. $Q_S$ iff in observer $\mathcal{A}_{\text{obs}}$, every reachable state $x$ contains at least one non-eventually-secret state of $\mathcal{A}$.
Theorem 14

An RTA $\mathcal{A}$ is CSO w.r.t. $Q_S$ iff in observer $\mathcal{A}_{obs}$, every reachable state $x$ contains at least one non-eventually-secret state of $\mathcal{A}$.

Example 15 (cont. $\mathcal{A}_1$)

Let $Q_S = \{q_5\}$, so the eventually secret states are $q_3$ and $q_5$. 
Theorem 14

An RTA $\mathcal{A}$ is CSO w.r.t. $Q_S$ iff in observer $\mathcal{A}_{\text{obs}}$, every reachable state $x$ contains at least one non-eventually-secret state of $\mathcal{A}$.

Example 15 (cont. $\mathcal{A}_1$)

Let $Q_S = \{q_5\}$, so the eventually secret states are $q_3$ and $q_5$. In $\mathcal{A}_{1\text{obs}}$, there is a reachable state $\{q_3\}$ which only contains eventually secret states, then $\mathcal{A}_1$ is not CSO w.r.t. $\{q_5\}$.
Main results
Sufficient and necessary conditions for opacity

**Theorem 14**

An RTA $\mathcal{A}$ is **CSO** w.r.t. $Q_S$ iff in observer $\mathcal{A}_{obs}$, every reachable state $x$ contains at least one non-eventually-secret state of $\mathcal{A}$.

**Example 15 (cont. $\mathcal{A}_1$)**

Let $Q_S = \{q_5\}$, so the eventually secret states are $q_3$ and $q_5$. In $\mathcal{A}_{1 \text{ obs}}$, there is a reachable state $\{q_3\}$ which only contains eventually secret states, then $\mathcal{A}_1$ is **not CSO** w.r.t. $\{q_5\}$.

For an RTA $\mathcal{A}$, its observer $\mathcal{A}_{obs}$ can be computed in **2-EXPTIME**.
Content

1. Review of opacity results in the literature

2. Notation in real-time automata

3. Main results
   - The definitions of opacity
   - The notions of observer and reverse observer
   - Sufficient and necessary conditions for opacity

4. Conclusion
Results
Results

- Notions of state-based opacity in real-time automata (RTAs)
Results

- Notions of state-based opacity in real-time automata (RTAs)
- Notions of observer and reverse observer of RTAs
Results

- Notions of state-based opacity in real-time automata (RTAs)
- Notions of observer and reverse observer of RTAs
- Verification of state-based opacity with complexity upper bounds
Thank you for your attention!
References


