Multi-Head Monitoring of Metric Temporal Logic

Martin Raszyk, David Basin, Srđan Krstić, and Dmitriy Traytel

ETH zürich

Organisation of the Talk

Introduction — Monitoring Problem and Metric Temporal Logic

Online vs Multi-Head Monitoring

MTL Multi-Head Monitor

Evaluation

Future Work

Roadmap

Introduction — Monitoring Problem and Metric Temporal Logic

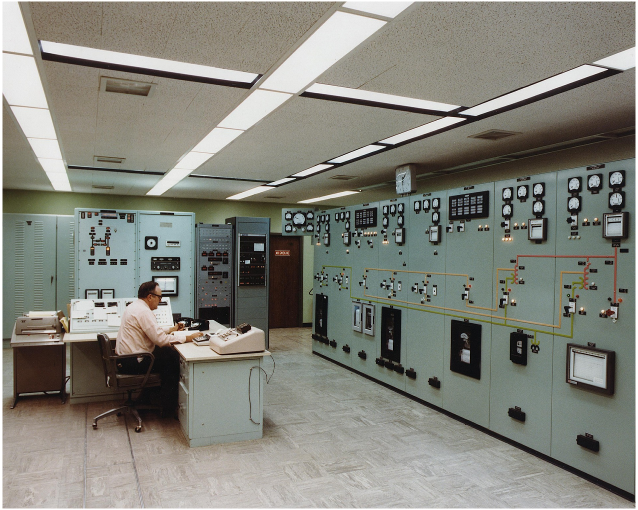
Online vs Multi-Head Monitoring

MTL Multi-Head Monitor

Evaluation

Future Work

Monitoring Problem



Example Property[1]

Every alarm is followed by a shut down event in 10 time units unless all clear is sounded first.

[1] Joël Ouaknine and James Worrell, FORMATS 2008

Linear Temporal Logic (with Past and Future)

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bullet\ \varphi \mid \bigcirc\ \varphi \mid \varphi\ \mathsf{S}\ \varphi \mid \varphi\ \mathsf{U}\ \varphi$$

p: *atomic proposition*

Linear Temporal Logic (with Past and Future)

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bullet\, \varphi \mid \circ\, \varphi \mid \varphi\, \mathsf{S}\, \varphi \mid \varphi\, \mathsf{U}\, \varphi$$

p: atomic proposition

Semantics:

$\bullet\, \varphi$ (previous)

Linear Temporal Logic (with Past and Future)

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bullet\, \varphi \mid \bigcirc\, \varphi \mid \varphi\ \mathsf{S}\ \varphi \mid \varphi\ \mathsf{U}\ \varphi$$

p: *atomic proposition*

Semantics:

$\bullet\, \varphi$ (previous)

$\bigcirc\, \varphi$ (next)

Linear Temporal Logic (with Past and Future)

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bullet\, \varphi \mid \circ\, \varphi \mid \varphi \,\mathsf{S}\, \varphi \mid \varphi \,\mathsf{U}\, \varphi$$

p: *atomic proposition*

Semantics:

$\bullet\, \varphi$ (previous)

$\circ\, \varphi$ (next)

$\varphi \,\mathsf{S}\, \psi$ (since)

Linear Temporal Logic (with Past and Future)

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \bullet\, \varphi \mid \bigcirc\, \varphi \mid \varphi \,\mathsf{S}\, \varphi \mid \varphi \,\mathsf{U}\, \varphi$$

p: atomic proposition

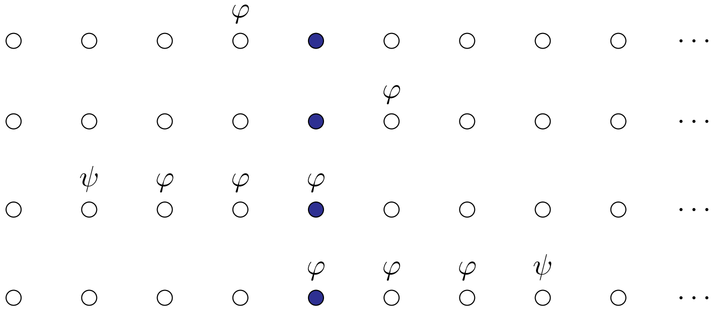
Semantics:

$\bullet\, \varphi$ (previous)

$\bigcirc\, \varphi$ (next)

$\varphi \,\mathsf{S}\, \psi$ (since)

$\varphi \,\mathsf{U}\, \psi$ (until)



Example Property

Every alarm is followed by a shut down event in 10 time units unless all clear is sounded first.

Atomic propositions:

$$P = \{\texttt{alarm}, \texttt{shut\_down}, \texttt{all\_clear}\}$$

Example Property

Every alarm is followed by a shut down event in 10 time units unless all clear is sounded first.

Atomic propositions:

$$P = \{\texttt{alarm}, \texttt{shut\_down}, \texttt{all\_clear}\}$$

Example Property

Every alarm is followed by a shut down event in 10 time units unless all clear is sounded first.

Atomic propositions:

$$P = \{\texttt{alarm}, \texttt{shut\_down}, \texttt{all\_clear}\}$$

Example Property

Every alarm is followed by a shut down event in 10 time units unless all clear is sounded first.

Atomic propositions:

$$P = \{\texttt{alarm}, \texttt{shut\_down}, \texttt{all\_clear}\}$$

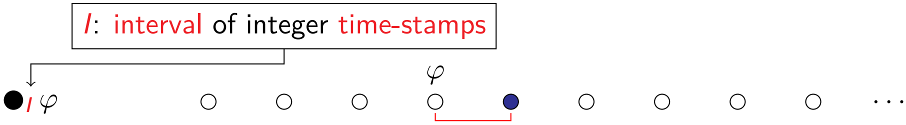
Example Property

Every alarm is followed by a shut down event <u>in 10 time units</u> unless all clear is sounded first.

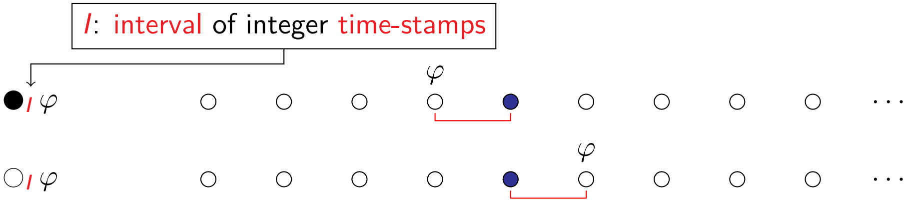
Atomic propositions:

$$P = \{\texttt{alarm}, \texttt{shut\_down}, \texttt{all\_clear}\}$$

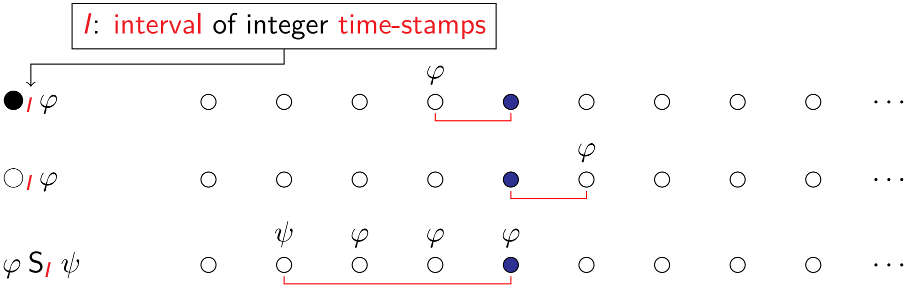
Metric Temporal Logic



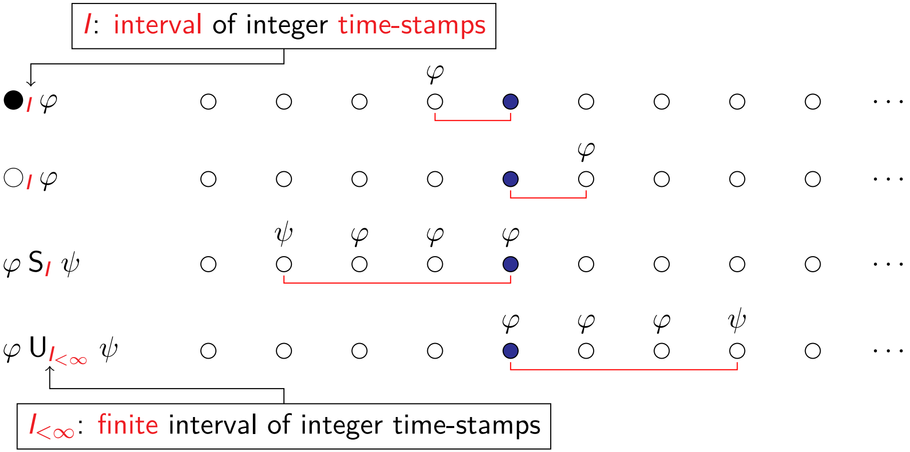
Metric Temporal Logic



Metric Temporal Logic



Metric Temporal Logic



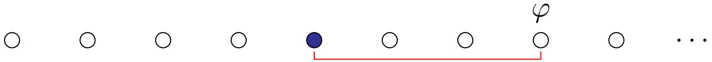
Example Property

Every alarm is followed by a shut down event in 10 time units unless all clear is sounded first.

MTL formula:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\, \texttt{all\_clear} \vee \Diamond_{[10,10]}\, \texttt{shut\_down})$$

$\Diamond_{I_{<\infty}}\, \varphi$



MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\, \texttt{all\_clear} \vee \Diamond_{[10,10]}\, \texttt{shut\_down})$$

Time-stamped stream ρ Boolean output stream

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\, \texttt{all\_clear} \vee \Diamond_{[10,10]}\, \texttt{shut\_down})$$

Time-stamped stream ρ		Boolean output stream
@0	all_clear	

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\, \texttt{all\_clear} \vee \Diamond_{[10,10]}\, \texttt{shut\_down})$$

Time-stamped stream ρ		Boolean output stream	
@0	all_clear	@0	true

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\,\texttt{all\_clear} \vee \Diamond_{[10,10]}\,\texttt{shut\_down})$$

Time-stamped stream ρ		Boolean output stream	
@0	all_clear	@0	true
@10	alarm		

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]} \texttt{all\_clear} \vee \Diamond_{[10,10]} \texttt{shut\_down})$$

Time-stamped stream ρ	
@0	all_clear
@10	alarm
@20	all_clear

Boolean output stream	
@0	true

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\, \texttt{all\_clear} \vee \Diamond_{[10,10]}\, \texttt{shut\_down})$$

Time-stamped stream ρ	
@0	all_clear
@10	alarm
@20	all_clear

Boolean output stream	
@0	true
@10	true
@20	true

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\mathtt{alarm} \rightarrow (\Diamond_{[0,10]}\, \mathtt{all\_clear} \vee \Diamond_{[10,10]}\, \mathtt{shut\_down})$$

Time-stamped stream ρ	
@0	all_clear
@10	alarm
@20	all_clear
@30	alarm

Boolean output stream	
@0	true
@10	true
@20	true

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]} \texttt{all\_clear} \vee \Diamond_{[10,10]} \texttt{shut\_down})$$

Time-stamped stream ρ	
@0	all_clear
@10	alarm
@20	all_clear
@30	alarm
@45	all_clear

Boolean output stream	
@0	true
@10	true
@20	true

MTL Monitoring Problem: Definition

Input MTL formula φ, time-stamped stream of events ρ.

Output *Stream* of Boolean values denoting whether $(\rho, i) \models \varphi$.

Example:

$$\texttt{alarm} \rightarrow (\Diamond_{[0,10]}\, \texttt{all\_clear} \vee \Diamond_{[10,10]}\, \texttt{shut\_down})$$

Time-stamped stream ρ		Boolean output stream	
@0	all_clear	@0	true
@10	alarm	@10	true
@20	all_clear	@20	true
@30	alarm	@30	false
@45	all_clear	@45	true

Roadmap

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]} \mathtt{b}$$

Input stream ρ

Boolean output stream

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Boolean output stream
@0	a	

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \texttt{a} \rightarrow \Diamond_{[0,60]}\ \texttt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0				

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\, \mathtt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \texttt{a} \to \Diamond_{[0,60]}\, \texttt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		
@0	a			

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		
@0	a	@0	a	

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \texttt{a} \rightarrow \Diamond_{[0,60]}\ \texttt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		
@0	a	@0	a	
@0				

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		
@0	a	@0	a	
@0		@0		

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\, \mathtt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		
@0	a	@0	a	
@0		@0		
@10	b			

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events	Boolean output stream	
@0	a		@0	true
@0			@0	true
@0	a		@0	true
@0			@0	true
@10	b		@10	true

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\, \mathtt{b}$$

Input stream ρ		Buffered events		Boolean output stream
@0	a	@0	a	
@0		@0		
@0	a	@0	a	
@0		@0		
@80				

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]} \mathtt{b}$$

Input stream ρ		Buffered events	Boolean output stream	
@0	a		@0	false
@0			@0	true
@0	a		@0	false
@0			@0	true
@80			@80	true

Online Monitoring

Online monitor — algorithm reading event stream *once*, one event at a time.

Example:

$$\varphi = \mathtt{a} \rightarrow \lozenge_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events	Boolean output stream	
@0	a		@0	false
@0			@0	true
@0	a		@0	false
@0			@0	true
@80			@80	true

Online monitor might need to buffer the entire trace in memory.

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ | Buffered events | Output stream

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events	Output stream
@0	a		

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream
@0	a	@0:0	a	

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream
@0	a	@0:0	a	
@0				

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\, \mathtt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0					

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0					
@0	a				

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0				@0:2	= @0:0
@0	a				

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0				@0:2	= @0:0
@0	a				
@0					

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \texttt{a} \rightarrow \Diamond_{[0,60]}\ \texttt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0				@0:2	= @0:0
@0	a			@0:3	true
@0					

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0				@0:2	= @0:0
@0	a			@0:3	true
@0					
@10	b				

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\, \mathtt{b}$$

Input stream ρ		Buffered events	Output stream	
@0	a		@0:1	true
@0			@0:2	= @0:0
@0	a		@0:3	true
@0			@0:0	true
@10	b		@10:0	true

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events		Output stream	
@0	a	@0:0	a	@0:1	true
@0				@0:2	= @0:0
@0	a			@0:3	true
@0					
@80					

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events	Output stream	
@0	a		@0:1	true
@0			@0:2	= @0:0
@0	a		@0:3	true
@0			@0:0	false
@80			@80:0	true

Related Work: AERIAL (TACAS 2017)

Idea

Represent the Boolean output stream implicitly using out-of-order verdicts and equivalence verdicts.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Input stream ρ		Buffered events	Output stream	
@0	a		@0:1	true
@0			@0:2	= @0:0
@0	a		@0:3	true
@0			@0:0	false
@80			@80:0	true

AERIAL only needs to buffer a constant number of events (independent of the trace).

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \texttt{a} \rightarrow \Diamond_{[0,60]}\ \texttt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream
● ●	@0	a	
	@0		
	@0	a	
	@0		
	@10	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads		Time-stamped stream ρ		Boolean output stream
●		@0	a	
	●	@0		
		@0	a	
		@0		
		@10	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream
	@0	a	
	@0		
	@0	a	
	@0		
	@10	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream
● (blue)	@0	a	
	@0		
	@0	a	
● (red)	@0		
	@10	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads		Time-stamped stream ρ		Boolean output stream
●		@0	a	
		@0		
		@0	a	
		@0		
	●	@10	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	true
● (blue)	@0			
	@0	a		
	@0			
● (red)	@10	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	true
	@0		@0	true
● (blue)	@0	a		
	@0			
● (red)	@10	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \texttt{a} \rightarrow \Diamond_{[0,60]}\ \texttt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	true
	@0		@0	true
	@0	a	@0	true
	@0			
	@10	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	true
	@0		@0	true
	@0	a	@0	true
	@0		@0	true
● ●	@10	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\, \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	true
	@0		@0	true
	@0	a	@0	true
	@0		@0	true
● ●	@10	b	@10	true

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \lozenge_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream
●	@0	a	
	@0		
	@0	a	
●	@0		
	@80	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream
●	@0	a	
	@0		
	@0	a	
	@0		
●	@80	b	

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	false
●	@0			
	@0	a		
	@0			
●	@80	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	false
	@0		@0	true
●	@0	a		
	@0			
●	@80	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	false
	@0		@0	true
	@0	a	@0	false
●	@0			
●	@80	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	false
	@0		@0	true
	@0	a	@0	false
	@0		@0	true
● ●	@80	b		

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	false
	@0		@0	true
	@0	a	@0	false
	@0		@0	true
● ●	@80	b	@80	true

Multi-Head Monitoring

Idea

Read multiple events from the event stream at once.

$$\varphi = \mathtt{a} \rightarrow \Diamond_{[0,60]}\ \mathtt{b}$$

Heads	Time-stamped stream ρ		Boolean output stream	
	@0	a	@0	false
	@0		@0	true
	@0	a	@0	false
	@0		@0	true
● ●	@80	b	@80	true

Can produce in-order explicit Boolean verdicts in constant working memory (using one head for each atomic proposition).

Roadmap

Introduction — Monitoring Problem and Metric Temporal Logic

Online vs Multi-Head Monitoring

MTL Multi-Head Monitor

Evaluation

Future Work

Multi-Head Monitor's Structure

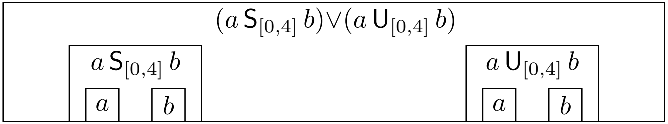
Example MTL formula:

$$(a \, \mathsf{S}_{[0,4]} \, b) \vee (a \, \mathsf{U}_{[0,4]} \, b)$$

Multi-Head Monitor's Structure

Example MTL formula:

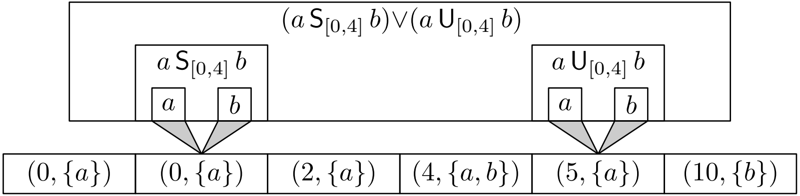
$$(a \, \mathsf{S}_{[0,4]} \; b) \vee (a \, \mathsf{U}_{[0,4]} \; b)$$



Multi-Head Monitor's Structure

Example MTL formula:

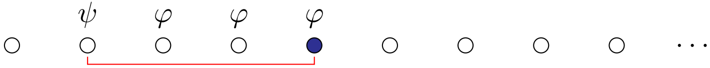
$$(a \,\mathsf{S}_{[0,4]}\, b) \vee (a \,\mathsf{U}_{[0,4]}\, b)$$



Since Operator Evaluation

Semantics:

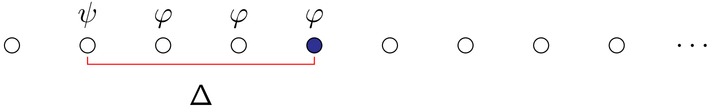
$\varphi \, \mathsf{S}_{[a,b]} \, \psi$



Since Operator Evaluation

Semantics:

$\varphi \, \mathsf{S}_{[a,b]} \, \psi$



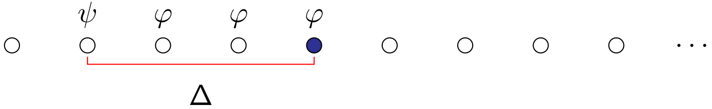
Idea (assuming $b \neq \infty$)

Store all the time-stamp differences $\Delta \in [0, b]$*.*

Since Operator Evaluation

Semantics:

$\varphi \, \mathsf{S}_{[a,b]} \, \psi$



Idea (assuming $b \neq \infty$)

Store all the time-stamp differences $\Delta \in [0, b]$.

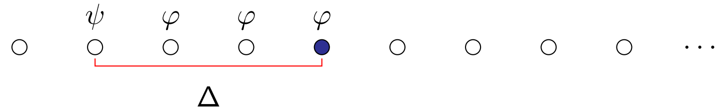
Need to store a set of natural numbers and (efficiently)

- query the maximum element,

Since Operator Evaluation

Semantics:

$\varphi \, \mathsf{S}_{[a,b]} \, \psi$



Idea (assuming $b \neq \infty$)

Store all the time-stamp differences $\Delta \in [0, b]$.

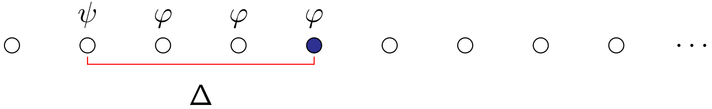
Need to store a set of natural numbers and (efficiently)

- query the maximum element,
- remove the maximum element,

Since Operator Evaluation

Semantics:

$\varphi \, \mathsf{S}_{[a,b]} \, \psi$



Idea (assuming $b \neq \infty$)

Store all the time-stamp differences $\Delta \in [0, b]$.

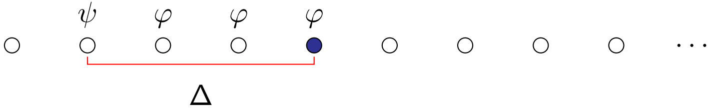
Need to store a set of natural numbers and (efficiently)

- query the maximum element,
- remove the maximum element,
- insert a zero,

Since Operator Evaluation

Semantics:

$\varphi \, \mathsf{S}_{[a,b]} \, \psi$



Idea (assuming $b \neq \infty$)

Store all the time-stamp differences $\Delta \in [0, b]$.

Need to store a set of natural numbers and (efficiently)

- query the maximum element,
- remove the maximum element,
- insert a zero,
- increase all elements by δ.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[] \quad (\Sigma = 0)$$

represents the multiset $\{\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0] \quad (\Sigma = 0)$$

represents the multiset $\{0\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4] \quad (\Sigma = 4)$$

represents the multiset $\{4\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4, 2] \quad (\Sigma = 6)$$

represents the multiset $\{6\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4, 2, 0] \quad (\Sigma = 6)$$

represents the multiset $\{6, 0\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4, 2, {\color{red}0^2}] \quad (\Sigma = 6)$$

represents the multiset $\{6, 0^2\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4, 2, 0^2, 1] \quad (\Sigma = 7)$$

represents the multiset $\{7, 1^2\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4, 2, 0^2, 1, {\color{red}0}] \quad (\Sigma = 7)$$

represents the multiset $\{7, 1^2, 0\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0, 4, 2, 0^2, 1, 0, \color{red}{3}] \quad (\Sigma = 10)$$

represents the multiset $\{10, 4^2, 3\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[\cancel{0}, \cancel{4}, \cancel{2}, 0^2, 1, 0, 3] \quad (\Sigma = 4)$$

represents the multiset $\{\cancel{10}, 4^2, 3\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0^2, 1, 0, 3] \quad (\Sigma = 4)$$

represents the multiset $\{4^2, 3\}$.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0^2, 1, 0, 3] \quad (\Sigma = 4)$$

represents the multiset $\{4^2, 3\}$.

Each operation takes amortized constant time.

Delta Queue

Idea

Just store all inserted zeros and increases in a list and keep track of its sum!

Example:

$$[0^2, 1, 0, 3] \quad (\Sigma = 4)$$

represents the multiset $\{4^2, 3\}$.

Each operation takes amortized constant time and space complexity is linear in Σ.

Until Operator Evaluation

Semantics:

$\varphi \, \mathsf{U}_{[a,b]} \, \psi$

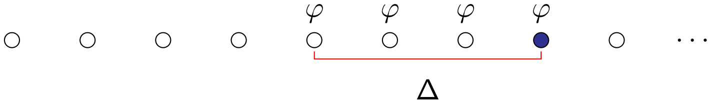
Until Operator Evaluation

Semantics:

$\varphi \, \mathsf{U}_{[a,b]} \, \psi$

Idea

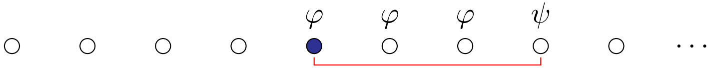
Store all the time-stamp differences $\Delta \in [0, b]$ *such that*



Until Operator Evaluation

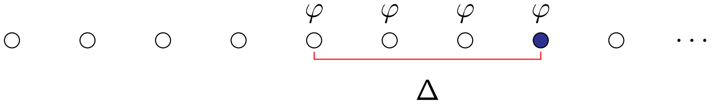
Semantics:

$\varphi \, \mathsf{U}_{[a,b]} \, \psi$



Idea

Store all the time-stamp differences $\Delta \in [0, b]$ *such that*



We can reuse Delta Queue!

Space and Time Complexity

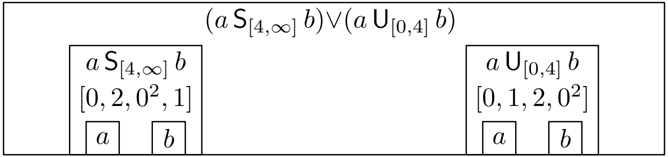
Size	$\lvert\varphi\rvert$	number of MTL operators in φ	$(a \mathsf{S}_{[4,\infty]}\ b) \vee (a \mathsf{U}_{[0,4]}\ b)$
Temporal Size	$\lVert\varphi\rVert$	plus interval bounds	$(a \mathsf{S}_{[4,\infty]}\ b) \vee (a \mathsf{U}_{[0,4]}\ b)$

Space and Time Complexity

Size	$\lvert\varphi\rvert$	number of MTL operators in φ	$(a\,\mathsf{S}_{[4,\infty]}\ b) \vee (a\,\mathsf{U}_{[0,4]}\ b)$
Temporal Size	$\lVert\varphi\rVert$	plus interval bounds	$(a\,\mathsf{S}_{[4,\infty]}\ b) \vee (a\,\mathsf{U}_{[0,4]}\ b)$

Theorem

Multi-head monitor's state for MTL formula φ requires $O(\lVert\varphi\rVert)$ registers storing time-stamps and indices into the trace.

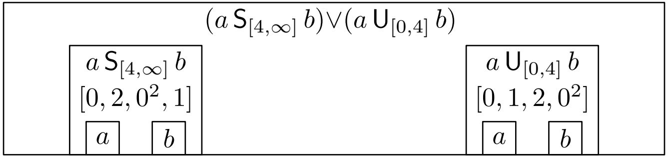


Space and Time Complexity

Size	$\lvert\varphi\rvert$	number of MTL operators in φ	$(a\,\mathsf{S}_{[4,\infty]}\ b) \vee (a\,\mathsf{U}_{[0,4]}\ b)$
Temporal Size	$\lVert\varphi\rVert$	plus interval bounds	$(a\,\mathsf{S}_{[4,\infty]}\ b) \vee (a\,\mathsf{U}_{[0,4]}\ b)$

Theorem

Multi-head monitor's state for MTL formula φ requires $O(\lVert\varphi\rVert)$ registers storing time-stamps and indices into the trace.



Claim

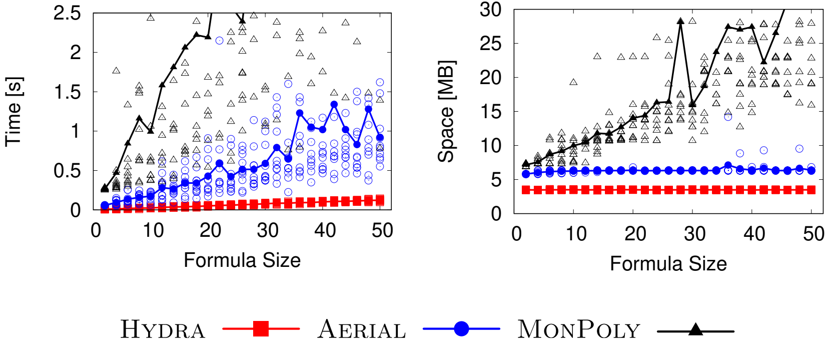
Multi-head monitor for MTL formula φ runs in amortized time $O(\lvert\varphi\rvert)$ per event.

Roadmap

Benchmarking Experiments — Average Case

Formula pseudo-random formula of given size.

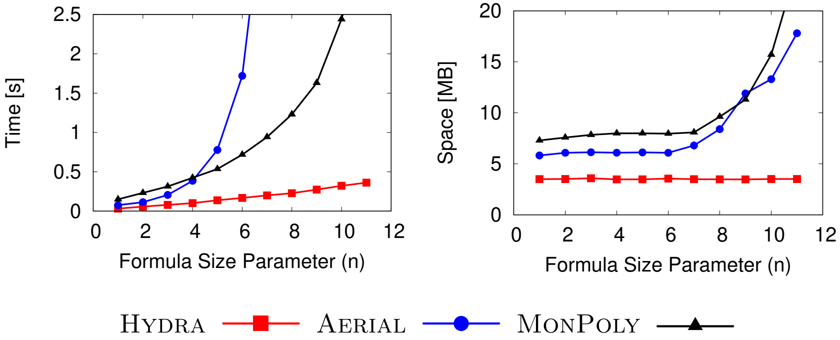
Trace pseudo-random trace of fixed length.



Benchmarking Experiments — Worst Case

Formula MTL formula of size $O(n)$ encoding that a bit string $x \in \{0,1\}^n$ precedes a distinguished labeled bit string $y \in \{0,1\}^n$.

Trace all bit strings $x \in \{0,1\}^n$ followed by a single bit string $y \in \{0,1\}^n$ (pattern repeated to obtain a fixed length trace).



Roadmap

Introduction — Monitoring Problem and Metric Temporal Logic

Online vs Multi-Head Monitoring

MTL Multi-Head Monitor

Evaluation

Future Work

Multi-Head Monitoring of Metric Dynamic Logic

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid |r\rangle_I \mid \langle r|_I \qquad\qquad r = \star \mid \varphi? \mid r + r \mid r \cdot r \mid r^*$$

[2] Martin Raszyk, David A. Basin, Dmitriy Traytel: From Nondeterministic to Multi-Head Deterministic Finite-State Transducers. ICALP 2019.

Multi-Head Monitoring of Metric Dynamic Logic

Syntax:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid |r\rangle_I \mid \langle r|_I \qquad r = \star \mid \varphi? \mid r + r \mid r \cdot r \mid r^*$$

Building up on the following result[2]:

$$\begin{array}{ccccc} \mathcal{L}(\text{2DFT}) & = & \mathcal{L}(f\text{-2NFT}) & & \\ & & \cup\!\!\!\!\!\nmid\, & & \\ \mathcal{L}(\text{1DFT}) & \subsetneq & \mathcal{L}(f\text{-1NFT}) & \subsetneq & \mathcal{L}(\text{MH-1DFT}) \end{array}$$

[2]Martin Raszyk, David A. Basin, Dmitriy Traytel: From Nondeterministic to Multi-Head Deterministic Finite-State Transducers. ICALP 2019.