

A Formally Verified, Optimized Monitor for Metric First-Order Dynamic Logic

David Basin, Thibault Dardinier, Lukas Heimes,
Srđan Krstić, Martin Raszyk, Joshua Schneider and Dmitriy Traytel

ETH zürich

Department of Computer Science



Dmitriy

Rocket engineer



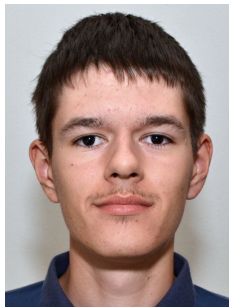
Joshua

Monitoring researcher



Srđan

Working formalizer



Martin

Quality assurer

All characters and events mentioned in this presentation are entirely fictitious.

The paper is real.

Act I: Going to Space



Dmitriy

Rocket engineer



Joshua

Monitoring researcher



Where: WASA Cafeteria

When: One week before the IJCAR deadline

Monitoring



Background image: NASA/JPL-Caltech/MSSS

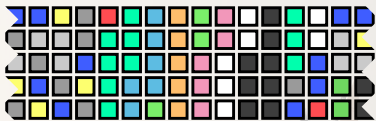
Monitoring

Correct behavior?



Background image: NASA/JPL-Caltech/MSSS

Monitoring

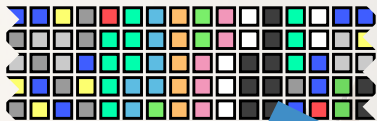


Observations
at runtime

Correct behavior?



Monitoring



Observations
at runtime



Correct behavior?



Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)
with aggregations

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi \mathcal{S}_I \varphi \mid \varphi \mathcal{U}_I \varphi$

$\mid x \leftarrow \Omega x; \vec{x}. \varphi \mid \dots$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

$\mid x \leftarrow \Omega x; \vec{x}. \varphi \mid \dots$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$

Examples:

Published reports must have been approved in the past seven days.

$\text{publish}(r) \rightarrow \blacklozenge_{[0,7d]} \text{approve}(r)$

(where $\blacklozenge_I \varphi = \text{true } S_I \varphi$)

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

$\mid x \leftarrow \Omega x; \vec{x}. \varphi \mid \dots$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$

Examples:

Published reports must have been approved in the past seven days.

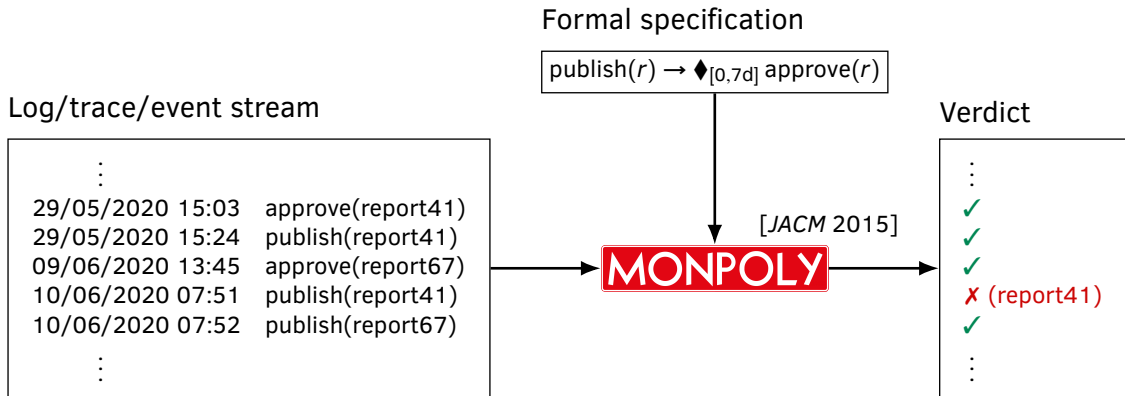
$\text{publish}(r) \rightarrow \blacklozenge_{[0,7d]} \text{approve}(r)$

(where $\blacklozenge_I \varphi = \text{true } S_I \varphi$)

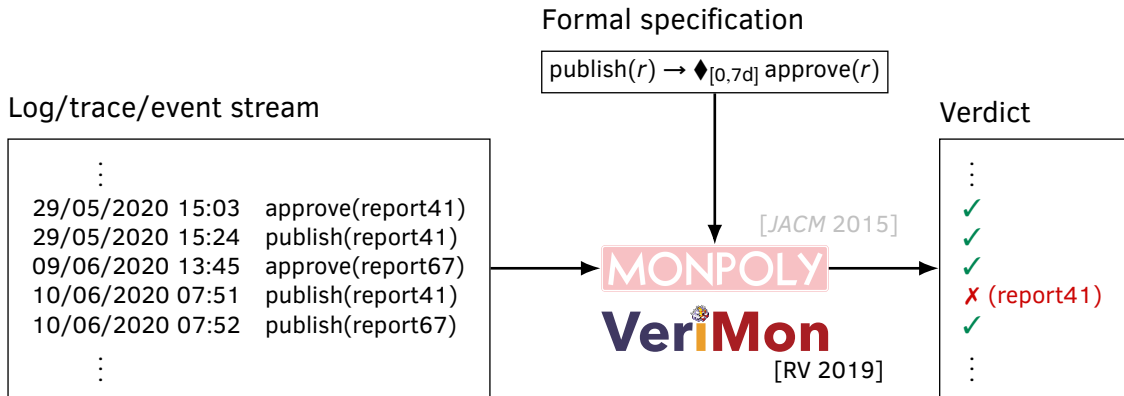
Maximum radiation must not exceed 3 Roentgen.

$(m \leftarrow \text{MAX } x. \text{rad}(x)) \rightarrow m \leq 3$

Monitors



Monitors



Let's try ...

Spec 1: The robot must not start to move if three or more transmissions of the same data failed within the last ten minutes without a successful transmission in between.





Let's try ...

Spec 1: The robot must not start to move if three or more transmissions of the same data failed within the last ten minutes without a successful transmission in between.

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.



MIND THE GAP

MONPOLY

MFOTL

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi \mathbf{S}_I \varphi \mid \varphi \mathbf{U}_I \varphi$

$\mid x \leftarrow \Omega t; \vec{x}. \varphi$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$

VeriMon

subset of MFOTL

no aggregations

$t ::= x \mid c$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t$

$\mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi \mathbf{S}_I \varphi \mid \varphi \mathbf{U}_I \varphi$

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$



MIND THE GAP

MONPOLY

MFOTL

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

$\mid x \leftarrow \Omega t; \vec{x}. \varphi$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$

VeriMon

subset of MFOTL

no aggregations

$t ::= x \mid c$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

cannot express Spec 2!

$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$

Spec 1 in MFOTL:

$$\left(\bigvee_{\substack{x \in \mathbf{N}^6, \\ \sum_i x_i = 600}} \begin{array}{l} \bullet_{[0,x_1]}(\neg \text{com_ok}(d) S_{[0,x_2]}(\text{com_fail}(d) \wedge \\ \bullet_{[0,x_3]}(\neg \text{com_ok}(d) S_{[0,x_4]}(\text{com_fail}(d) \wedge \\ \bullet_{[0,x_5]}(\neg \text{com_ok}(d) S_{[0,x_6]} \text{com_fail}(d)))))) \end{array} \right) \rightarrow \neg \text{move}$$

VeriMon
subset of MFOTL
no aggregations

C
 t_1, \dots, t_n
 t
 $\varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$
 $\varphi \mid \bigcirc_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

$$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$$

$$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$$

Spec 1 in MFOTL:

$$\begin{aligned}
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,600]} \\
 & \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,1]}(\neg \text{com_ok}(d) S_{[0,599]} \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,2]}(\neg \text{com_ok}(d) S_{[0,598]} \text{com_fail}(d)))))) \vee \\
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,3]}(\neg \text{com_ok}(d) S_{[0,597]} \\
 & \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,4]}(\neg \text{com_ok}(d) S_{[0,596]} \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,5]}(\neg \text{com_ok}(d) S_{[0,595]} \text{com_fail}(d)))))) \vee \\
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,6]}(\neg \text{com_ok}(d) S_{[0,594]} \\
 & \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,7]}(\neg \text{com_ok}(d) S_{[0,593]} \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,8]}(\neg \text{com_ok}(d) S_{[0,592]} \text{com_fail}(d)))))) \vee \\
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,9]}(\neg \text{com_ok}(d) S_{[0,591]} \\
 & \text{com_fail}(d)))))) \vee \dots
 \end{aligned}$$

664 353 676 371 disjuncts, assuming discrete time in seconds

VeriMon
subset of MFOTL
no aggregations

C
 $1, \dots, t_n$
 t
 $\varphi \wedge \varphi \mid \exists x. \varphi$
 $\varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

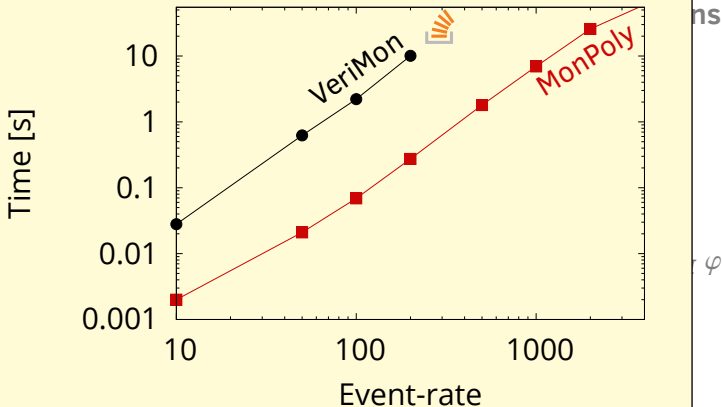
$$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$$

$$I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$$

Spec 1 in MFOTL:

$$\begin{aligned}
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \\
 & \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,1]}(\neg \text{com_ok}(d) S_{[0,599]} \text{com} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fa} \\
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \\
 & \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,4]}(\neg \text{com_ok}(d) S_{[0,596]} \text{com} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fa} \\
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \\
 & \text{com_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,7]}(\neg \text{com_ok}(d) S_{[0,593]} \text{com} \\
 & (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fa} \\
 & (\bullet_{[0,0]}(\neg \text{com_ok}(d) S_{[0,0]} (\text{com_fail}(d) \wedge \bullet_{[0,0]}(\neg \\
 & \text{com_fail}(d)))))) \vee \dots
 \end{aligned}$$

664 353 676 371 disjuncts, as

 $I ::= [\mathbf{N}, \mathbf{N} \cup \{\infty\}]$ 

Act II: Groundwork on Expressiveness



Joshua

Monitoring researcher



Srđan

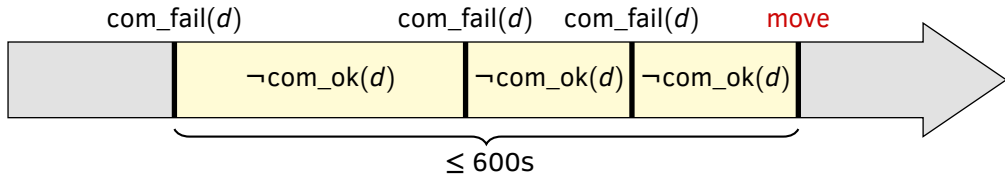
Working formalizer

Where: VeriMon Headquarters

When: 5 days before the IJCAR deadline

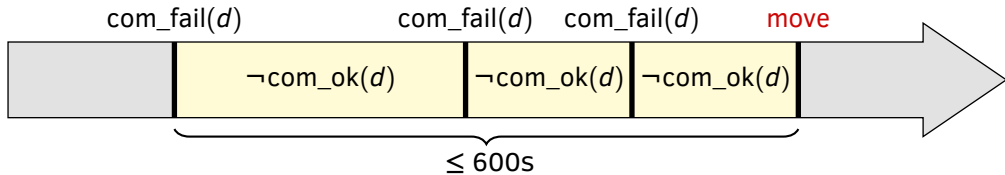
Regular expressions

Recall Spec 1: The robot must not start to move if three or more transmissions of the same data failed in a row within the last ten minutes.



Regular expressions

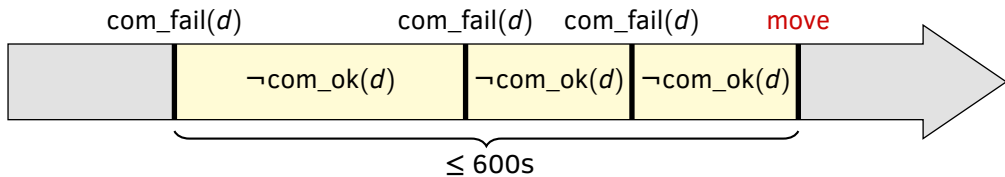
Recall Spec 1: The robot must not start to move if three or more transmissions of the same data failed in a row within the last ten minutes.



In MFOTL: $\left(\bigvee_{\substack{x \in \mathbb{N}^6, \\ \sum_i x_i = 600}} \bullet_{[0, x_1]} (\neg com_ok(d) S_{[0, x_2]} (com_fail(d) \wedge \bullet_{[0, x_3]} (\neg com_ok(d) S_{[0, x_4]} (com_fail(d) \wedge \bullet_{[0, x_5]} (\neg com_ok(d) S_{[0, x_6]} com_fail(d)))))) \right) \rightarrow \neg move$

Regular expressions

Recall Spec 1: The robot must not start to move if three or more transmissions of the same data failed in a row within the last ten minutes.

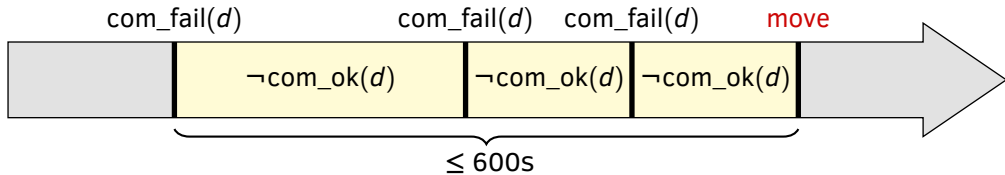


The new operator: $\blacktriangleleft_{\text{Interval}} \text{Pattern}$

where Pattern matches from position i to some past position $j \leq i$ and their time difference is in Interval .

Regular expressions

Recall Spec 1: The robot must not start to move if three or more transmissions of the same data failed in a row within the last ten minutes.



In MFODL: $(\blacktriangleleft_{[0,600]} com_fail(d) \cdot (\neg com_ok(d))^* \cdot$
 $com_fail(d) \cdot (\neg com_ok(d))^* \cdot$
 $com_fail(d) \cdot (\neg com_ok(d))^*) \rightarrow \neg move$

Metric First-Order *Dynamic* Logic

Formulas:

$\varphi ::= \dots$ all existing constructs

| $\blacktriangleleft_I e$ past match

| $\blacktriangleright_I e$ future match

Metric First-Order *Dynamic* Logic

Formulas:

$\varphi ::= \dots$ all existing constructs

| $\blacktriangleleft_I e$ past match

| $\blacktriangleright_I e$ future match

Regular expressions:

$e ::= _$ wildcard

| $\varphi?$ test formula

| $e + e$ alternation

| $e \cdot e$ concatenation

| e^* Kleene star

Metric First-Order *Dynamic* Logic

Formulas:

$\varphi ::= \dots$ all existing constructs

| $\blacktriangleleft_I e$ past match

| $\blacktriangleright_I e$ future match

$\blacktriangleleft_I \text{com_fail}(d) \cdot (\neg \text{com_ok}(d))^*$

abbreviates

$\blacktriangleleft_I (\text{com_fail}(d)? \cdot _) \cdot ((\neg \text{com_ok}(d))?) \cdot _)^*$

Regular expressions:

$e ::= _$ wildcard

| $\varphi?$ test formula

| $e + e$ alternation

| $e \cdot e$ concatenation

| e^* Kleene star

Metric First-Order *Dynamic* Logic

Formulas:

$\varphi ::= \dots$ all existing constructs

| $\blacktriangleleft_I e$ past match

| $\blacktriangleright_I e$ future match

Regular expressions:

$e ::= _$ wildcard

| $\varphi?$ test formula

| $e + e$ alternation

| $e \cdot e$ concatenation

| e^* Kleene star

$\blacktriangleleft_I \text{com_fail}(d) \cdot (\neg \text{com_ok}(d))^*$

abbreviates

$\blacktriangleleft_I (\text{com_fail}(d)? \cdot _) \cdot ((\neg \text{com_ok}(d))?) \cdot _)^*$

Note:

$\bullet_I \alpha \equiv \blacktriangleleft_I (\alpha? \cdot _)$

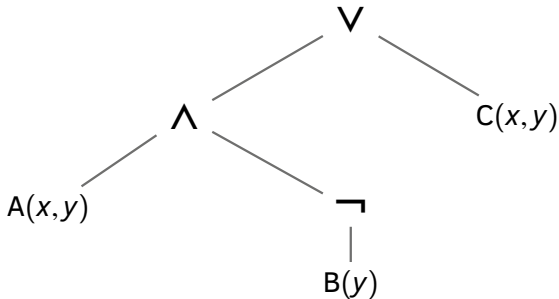
$\alpha \mathbf{S}_I \beta \equiv \blacktriangleleft_I (\beta? \cdot (_ \cdot \alpha?))^*$

$\circ_I \alpha \equiv \blacktriangleright_I (_ \cdot \alpha?)$

$\alpha \mathbf{U}_I \beta \equiv \blacktriangleright_I ((\alpha? \cdot _)^* \cdot \beta?)$

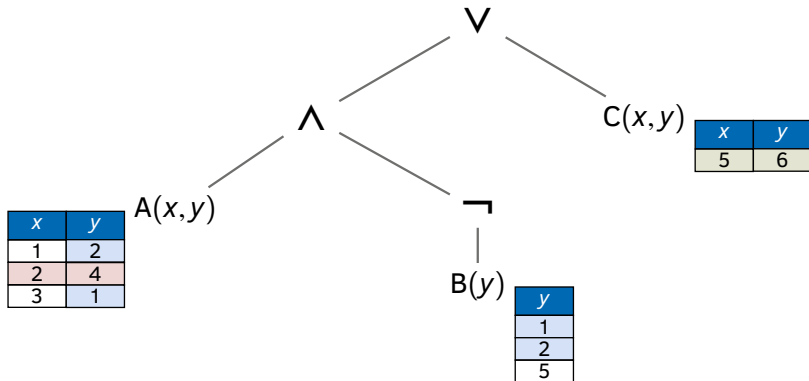
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



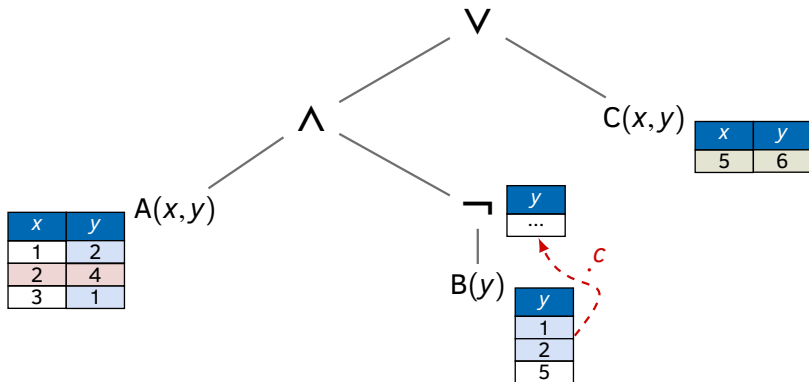
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



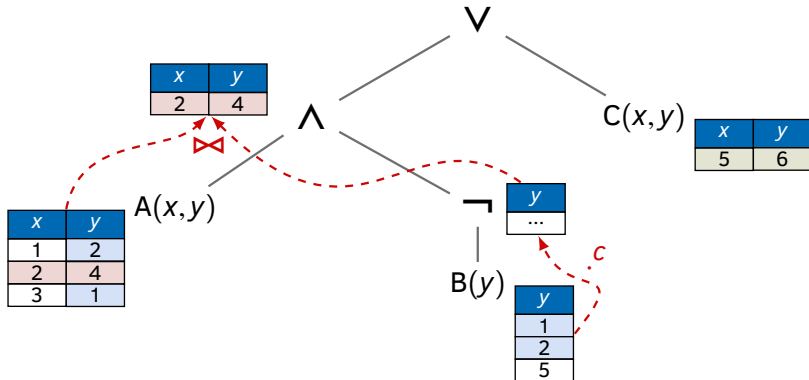
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



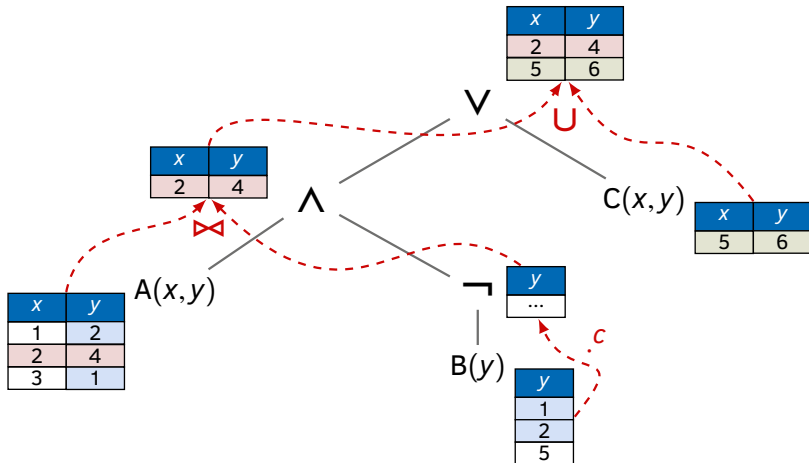
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



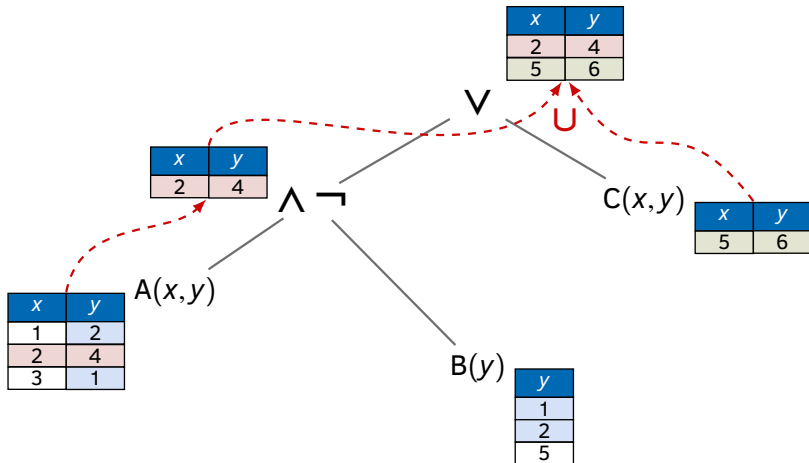
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



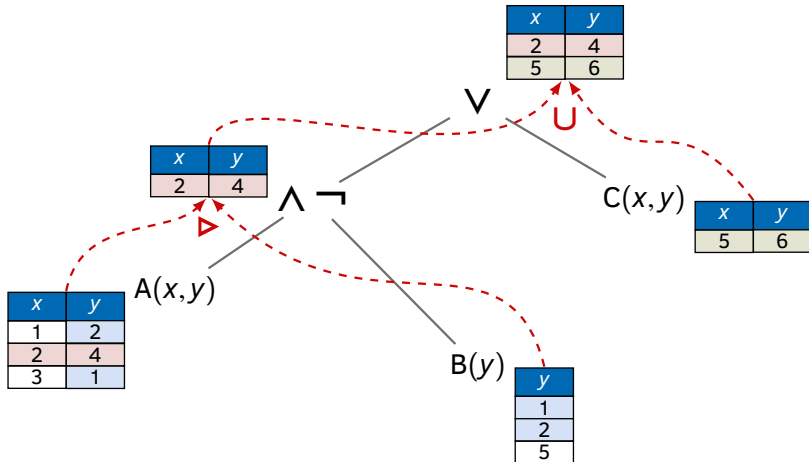
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



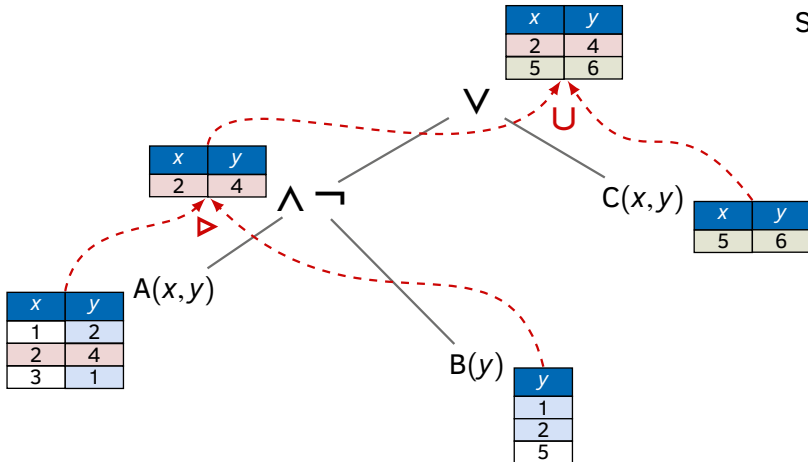
Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



Recap: Evaluation in VeriMon

Evaluating $(A(x,y) \wedge \neg B(y)) \vee C(x,y)$ with finite predicates, using only finite tables:



Syntactic constraints:

$\alpha \wedge \beta$ (join)

no constraint

$\alpha \wedge \neg \beta$ (anti-join)

$FV(\beta) \subseteq FV(\alpha)$

$\alpha \vee \beta$ (union)

$FV(\alpha) = FV(\beta)$

Finitely Evaluable Regular Expressions

Kleene star

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

Finitely Evaluable Regular Expressions

Kleene star

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

since r^* can match from i to i , v can be any valuation of $FV(r)$

Finitely Evaluable Regular Expressions

Kleene star must be guarded

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

since r^* can match from i to i , v can be any valuation of $FV(r)$

Finitely Evaluable Regular Expressions

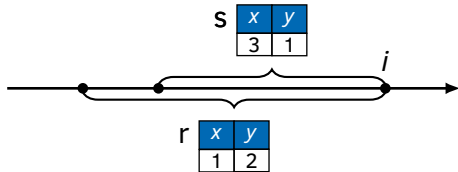
Kleene star

must be guarded

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

Alternation

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r + s$



$$v \in \begin{array}{|c|c|} \hline x & y \\ \hline 1 & 2 \\ \hline \end{array} \cup \begin{array}{|c|c|} \hline x & y \\ \hline 3 & 1 \\ \hline \end{array}$$

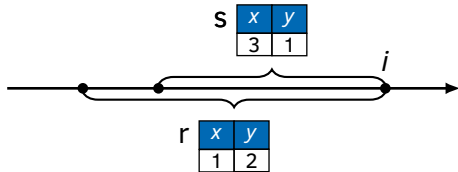
Finitely Evaluable Regular Expressions

Kleene star must be guarded

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

Alternation $FV(r) = FV(s)$

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r + s$



$$v \in \begin{array}{|c|c|} \hline x & y \\ \hline 1 & 2 \\ \hline \end{array} \cup \begin{array}{|c|c|} \hline x & y \\ \hline 3 & 1 \\ \hline \end{array}$$

Finitely Evaluable Regular Expressions

Kleene star must be guarded

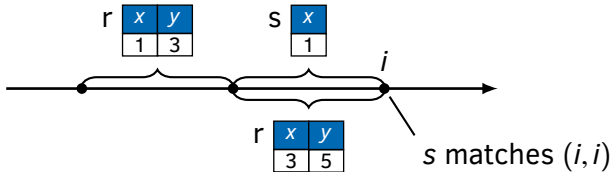
Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

Alternation $FV(r) = FV(s)$

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r + s$

Concatenation

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r \cdot s$



$$v \in \left(\begin{array}{|c|c|} \hline x & y \\ \hline 1 & 3 \\ \hline \end{array} \bowtie \begin{array}{|c|} \hline x \\ \hline 1 \\ \hline \end{array} \right) \cup \begin{array}{|c|c|} \hline x & y \\ \hline 3 & 5 \\ \hline \end{array}$$

Finitely Evaluable Regular Expressions

Kleene star must be guarded

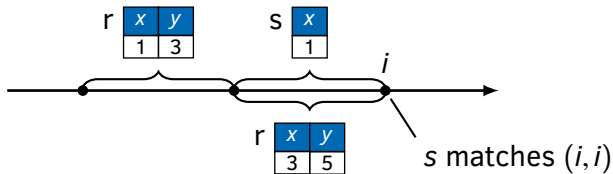
Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r^*$

Alternation $FV(r) = FV(s)$

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r + s$

Concatenation $FV(r) \supseteq FV(s)$

Example: $(v, i) \models \blacktriangleleft_{[0, \infty)} r \cdot s$



$$v \in \left(\begin{array}{|c|c|} \hline x & y \\ \hline 1 & 3 \\ \hline \end{array} \bowtie \begin{array}{|c|} \hline x \\ \hline 1 \\ \hline \end{array} \right) \cup \begin{array}{|c|c|} \hline x & y \\ \hline 3 & 5 \\ \hline \end{array}$$

Finitely Evaluable Regular Expressions

Kleene star $\boxed{\text{must be guarded}}$

Example: $(v, i) \models \triangleright_{[0, \infty)} r^*$

Alternation $\boxed{FV(r) = FV(s)}$

Example: $(v, i) \models \triangleright_{[0, \infty)} r + s$

Concatenation $\boxed{FV(r) \subseteq FV(s)}$

Example: $(v, i) \models \triangleright_{[0, \infty)} r \cdot s$

MONPOLY

MFOTL

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi \mathbf{S}_I \varphi \mid \varphi \mathbf{U}_I \varphi$

$\mid x \leftarrow \Omega t; \vec{x}. \varphi \mid \dots$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$



VeriMon⁺

MFODL

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi \mathbf{S}_I \varphi \mid \varphi \mathbf{U}_I \varphi$

$\mid \blacktriangleleft_I e \mid \blacktriangleright_I e$

$\mid x \leftarrow \Omega x; \vec{x}. \varphi \mid \dots$

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$e ::= _ \mid \varphi? \mid e + e \mid e \cdot e \mid e^*$

MONPOLY

MFOTL

with aggregations



VeriMon⁺

MFODL

with aggregations

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

$\mid x \leftarrow \Omega t \rightarrow t$

can easily express Specs 1 & 2

$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$t ::= x \mid c \mid t + t \mid t \times t \mid \dots$

$\varphi ::= p(t_1, \dots, t_n)$

$\mid t = t \mid t < t \mid t \leq t$

$\mid \neg \varphi \mid \varphi \wedge \varphi \mid \exists x. \varphi$

$\mid \bullet_I \varphi \mid \circ_I \varphi \mid \varphi S_I \varphi \mid \varphi U_I \varphi$

$\mid \blacktriangleleft_I e \mid \blacktriangleright_I e$

$\mid x \leftarrow \Omega x; \vec{x}. \varphi \mid \dots$

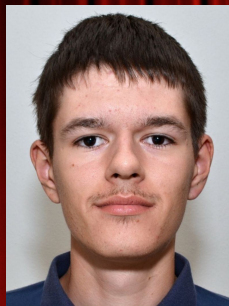
$\Omega ::= \text{MAX} \mid \text{MIN} \mid \text{CNT} \mid \text{SUM} \mid \text{AVG}$

$e ::= _ \mid \varphi? \mid e + e \mid e \cdot e \mid e^*$

Act III: Let's Make It Fly



Srđan
Working formalizer



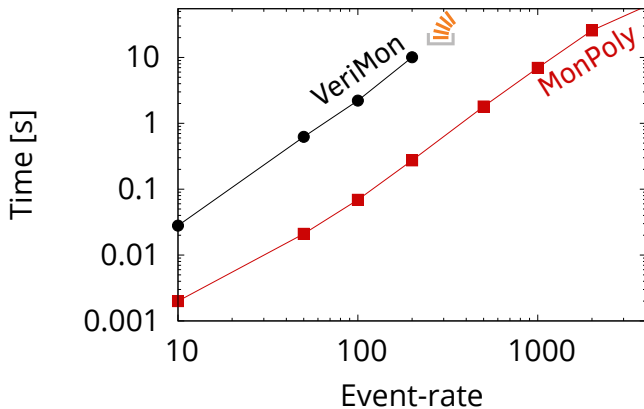
Martin
Quality assurer

Where: Implementer's Den

When: 3 days before the IJCAR deadline

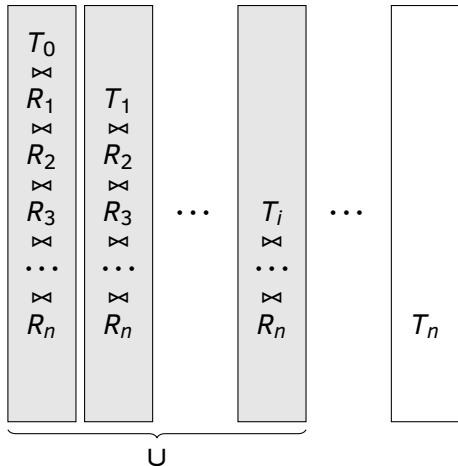
VeriMon Benchmark

$$((\blacklozenge_{[0,30]} P(x,y)) \wedge Q(x,z)) \wedge (\blacklozenge_{[0,30]} R(x,w))$$



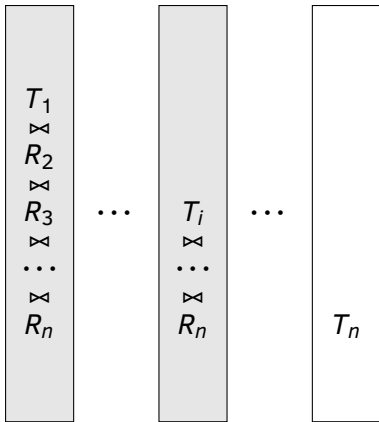
Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon



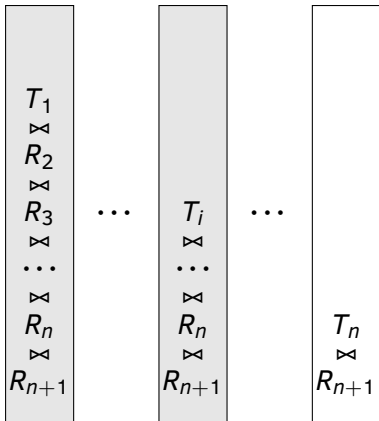
Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon



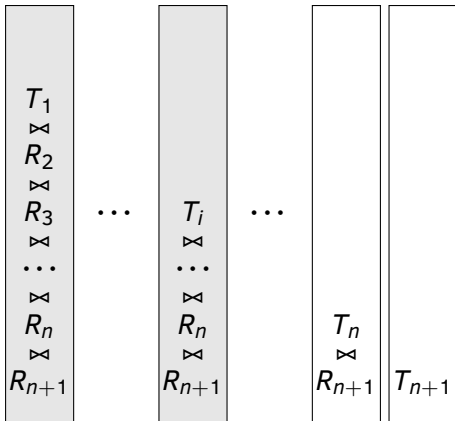
Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon



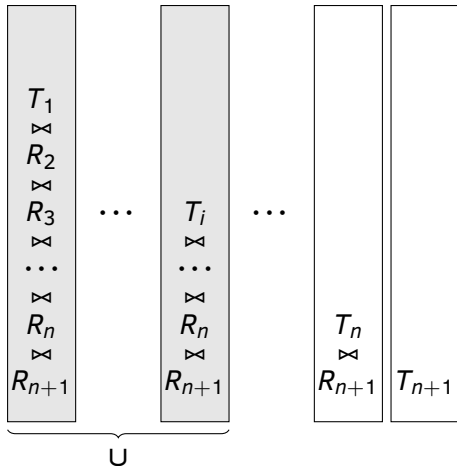
Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon



Sliding Window: $R(x) S_{[a,b]} T(x)$

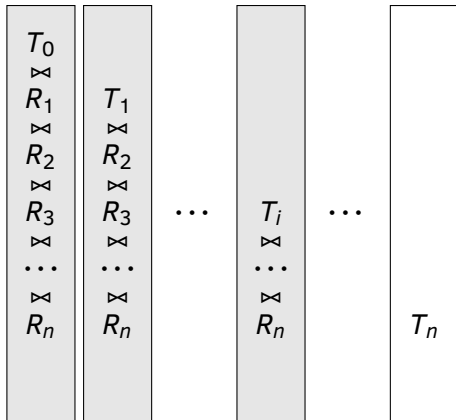
VeriMon



Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon

VeriMon⁺

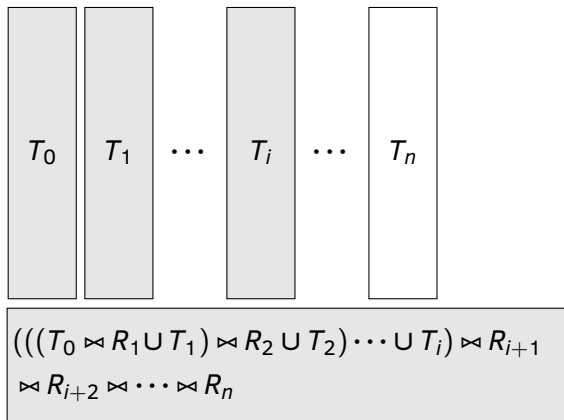
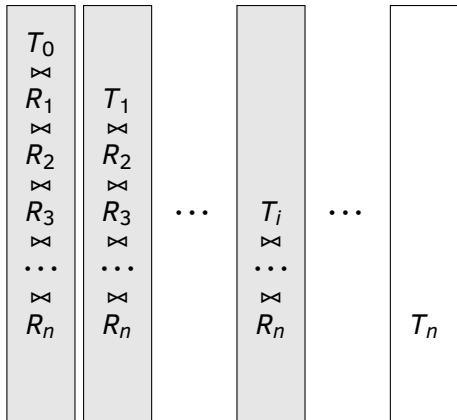


$$(((T_0 \bowtie R_1 \cup T_1) \bowtie R_2 \cup T_2) \cdots \cup T_i) \bowtie R_{i+1} \bowtie R_{i+2} \bowtie \cdots \bowtie R_n$$

Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon

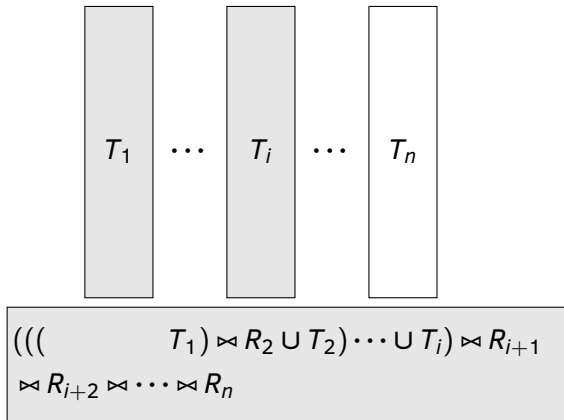
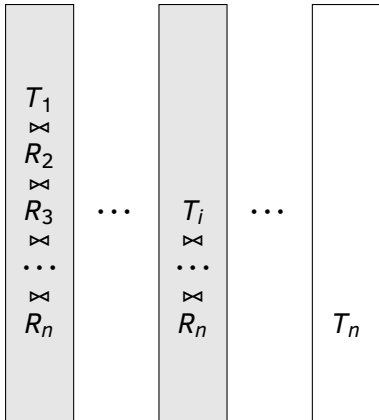
VeriMon⁺



Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon

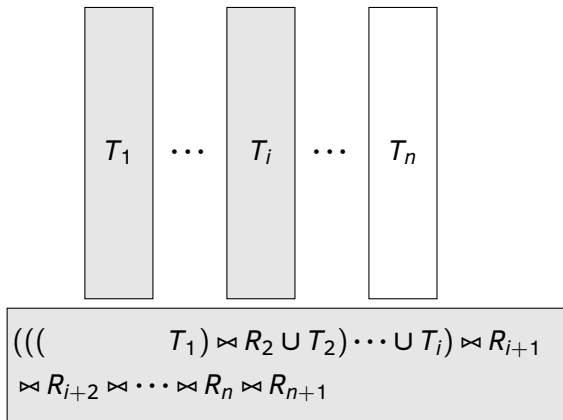
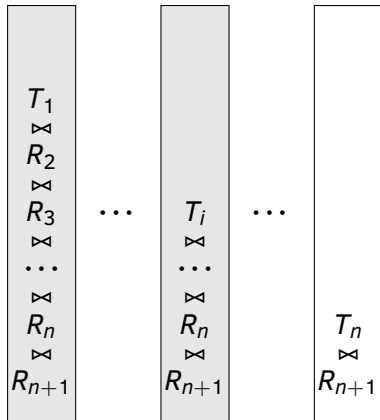
VeriMon⁺



Sliding Window: $R(x) S_{[a,b]} T(x)$

VeriMon

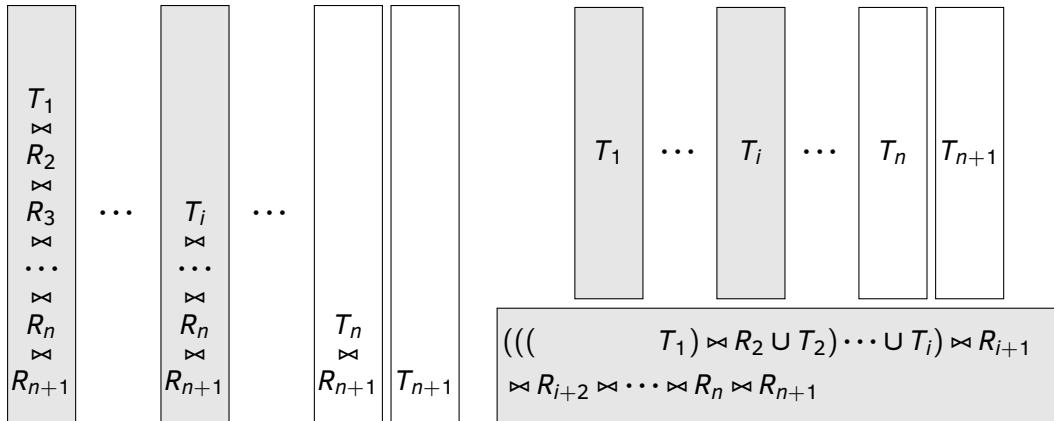
VeriMon⁺



Sliding Window: $R(x) S_{[a,b]} T(x)$

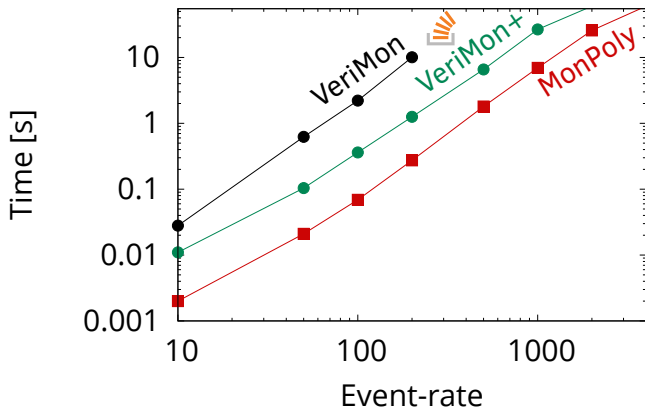
VeriMon

VeriMon⁺



VeriMon⁺ Benchmark

$$((\blacklozenge_{[0,30]} P(x,y)) \wedge Q(x,z)) \wedge (\blacklozenge_{[0,30]} R(x,w))$$

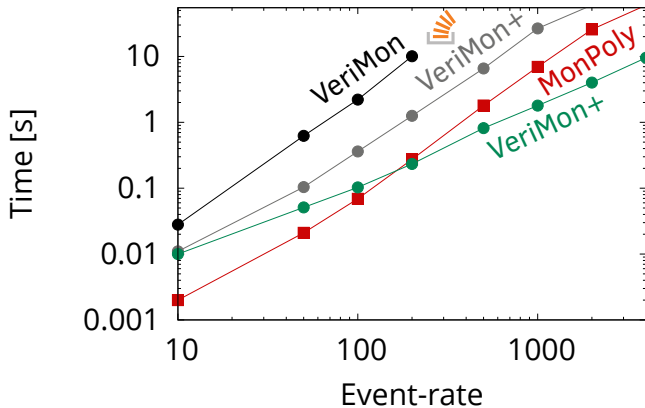


Multi-Way Join

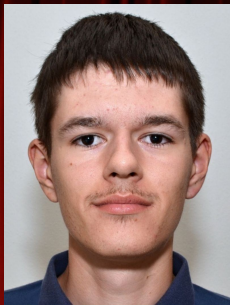
$$((\blacklozenge_{[0,30]} P(x, y)) \wedge Q(x, z)) \wedge (\blacklozenge_{[0,30]} R(x, w))$$

Multi-Way Join

$$(\diamond_{[0,30]} P(x, y)) \wedge Q(x, z) \wedge (\diamond_{[0,30]} R(x, w))$$



Act IV: Ready for Takeoff



Martin
Quality assurer



Dmitriy
Rocket engineer

Where: WASA Exhibition Room

When: 1 day before the IJCAR deadline

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

$$(e \leftarrow \text{MAX } e; (\blacklozenge_{[60,120]} E(e, id))) \wedge (\blacklozenge_{[60,120]} E(e, id))$$

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

$$(e \leftarrow \text{MAX } e; (\blacklozenge_{[60,120]} E(e, id))) \wedge (\blacklozenge_{[60,120]} E(e, id))$$

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

$$(e \leftarrow \text{MAX } e; (\blacklozenge_{[60,120]} E(e, id))) \wedge (\color{red}\blacklozenge_{[60,120]} E(e, id))$$

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

$$(e \leftarrow \text{MAX } e; (\blacklozenge_{[60,120]} E(e, id))) \wedge (\blacklozenge_{[60,120]} E(e, id))$$

$$(e \leftarrow \text{MAX } e; (\text{ONCE}[60,120] E(e, id))) \text{ AND } (\text{ONCE}[60,120] E(e, id))$$

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

$$(e \leftarrow \text{MAX } e; (\blacklozenge_{[60,120]} E(e, id))) \wedge (\blacklozenge_{[60,120]} E(e, id))$$

$$(e \leftarrow \text{MAX } e; (\text{ONCE}_{[60,120]} E(e, id))) \text{ AND } (\text{ONCE}_{[60,120]} E(e, id))$$

Log:

```
@70 E(30,"gps") E(25,"wifi")
@100 E(20,"gps")
@170 E(20,"wifi") E(20,"bluetooth")
@230 E(30,"wifi")
@300 E(10,"wifi")
```

Spec 2: The module with the highest energy consumption within the second to last minute must be reported to ground control.

$$(e \leftarrow \text{MAX } e; (\blacklozenge_{[60,120]} E(e, id))) \wedge (\blacklozenge_{[60,120]} E(e, id))$$

$(e \leftarrow \text{MAX } e; (\text{ONCE}_{[60,120]} E(e, id))) \text{ AND } (\text{ONCE}_{[60,120]} E(e, id))$

Log:

```
@70 E(30,"gps") E(25,"wifi")
@100 E(20,"gps")
@170 E(20,"wifi") E(20,"bluetooth")
@230 E(30,"wifi")
@300 E(10,"wifi")
```

Signature:

$E(\text{int}, \text{string})$

VeriMon⁺


✓ Correct

⚙ Expressive

⚡ Efficient

VeriMon⁺

✓ Correct


- 15 000 lines of 
- extraction to OCaml

 Expressive

 Efficient

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml


Expressive

- aggregations
- regular expressions

Efficient

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml

Expressive


- aggregations
- regular expressions

Efficient

- sliding window (S, U)
- multi-way join

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml

⚙ Expressive

- aggregations
- regular expressions

⚡ Efficient


- sliding window (S, U)
- multi-way join

VeriMon⁺⁺

- verified parsing
- extraction to LLVM

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml

⚙ Expressive

- aggregations
- regular expressions

⚡ Efficient

- sliding window (S, U)
- multi-way join

VeriMon⁺⁺

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

VeriMon⁺

✓ Correct

- 15 000
- extract



Expressive

- aggregations
- regular expressions



Efficient

- sliding window (S, U)
- multi-way join

$P(x) \vee \exists y. R(y)$

VeriMon⁺⁺

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

VeriMon⁺

✓ Correct

- 15
- ex

```
let T(x,y) = P(x,y) ∨  
      (∃z. T(x,z) ∧ P(z,y))  
in ...
```



Expressive

- negations
- ar expressions



Efficient

- sliding window (S, U)
- multi-way join


VeriMon⁺⁺

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml

⚙ Expressive

- aggregations
- regular expressions

⚡ Efficient

- sliding window (S, U)
- multi-way join

VeriMon⁺⁺

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

- sliding window (◀, ▶)
- moving aggregations

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml

⚙ Expressive

- aggr
- regul

⚡ Efficient

- sliding window (S, U)
- multi-way join

◀_[a,b] $\psi?$ · (_ · $\varphi?$)*

VeriMon⁺⁺

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

- sliding window (◀, ▶)
- moving aggregations

VeriMon⁺

✓ Correct

- 15 000 lines of 
- extraction to OCaml

⚙ Expressive

-
-

$x \leftarrow \text{AVG } a; u. \blacklozenge_{[0,3600]} P(a, u)$

⚡ Efficient

■ sliding window (S, U)
■ 4-way join

VeriMon⁺⁺

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

- sliding window ($\blacktriangleleft, \blacktriangleright$)
- moving aggregations



A Formally Verified, Optimized Monitor for Metric First-Order Dynamic Logic

David Basin, Thibault Dardinier, Lukas Heimes,
Srđan Krstić, Martin Raszyk, Joshua Schneider and **Dmitriy Traytel**

ETH zürich

Department of Computer Science



merci!
questions?