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**

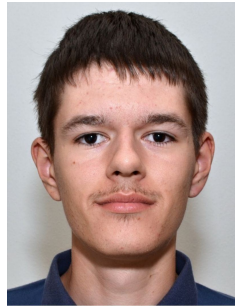
ETH zürich

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.

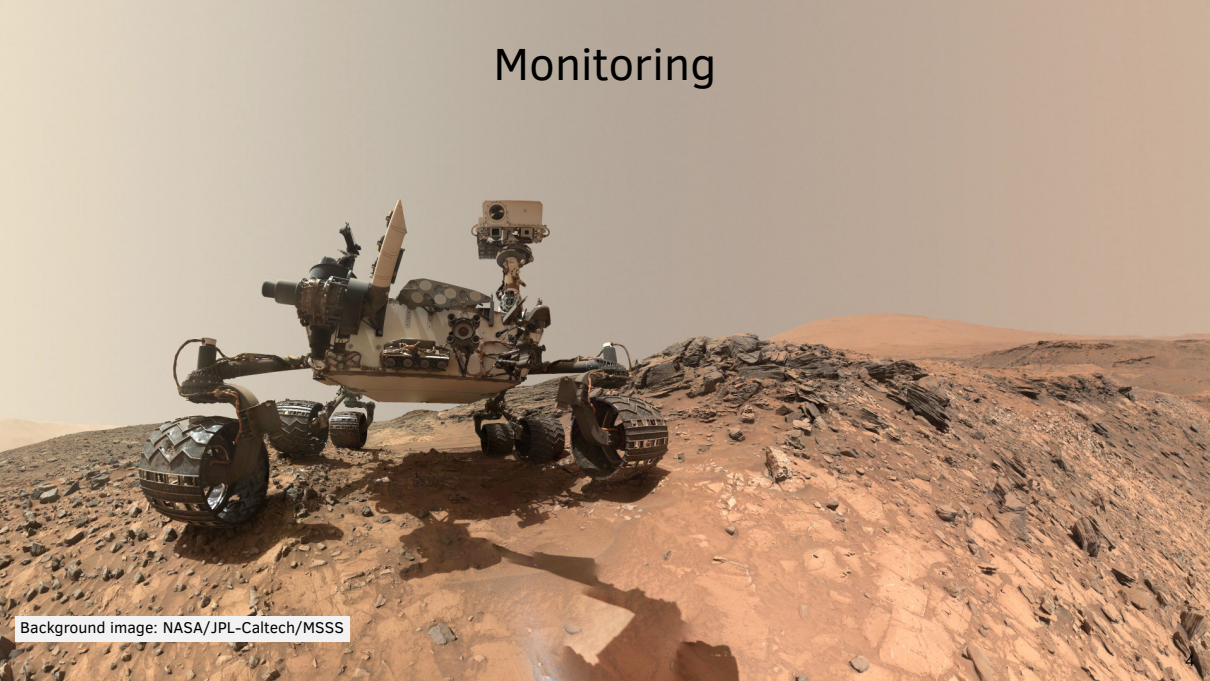
Act I: Going to Space

Dmitriy
Rocket engineer

Joshua
Monitoring researcher

Where: WASA Cafeteria
When: One week before the IJCAR deadline

Monitoring



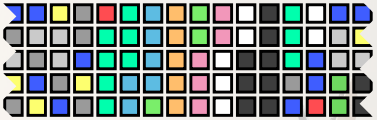
Background image: NASA/JPL-Caltech/MSSS

Monitoring

Correct behavior?

Background image: NASA/JPL-Caltech/MSSS

Monitoring

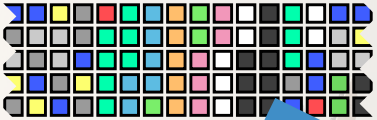


Observations
at runtime

Correct behavior?

Background image: NASA/JPL-Caltech/MSSS

Monitoring



Observations
at runtime

Correct behavior?

Background image: NASA/JPL-Caltech/MSSS

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)
with aggregations

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \dots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \,\mathsf{S}_I\, \varphi \mid \varphi \,\mathsf{U}_I\, \varphi \\ & \mid x \leftarrow \Omega\, x; \vec{x}.\, \varphi \mid \dots\end{aligned}$$

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

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

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \dots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \,\mathsf{S}_I\, \varphi \mid \varphi \,\mathsf{U}_I\, \varphi \\ & \mid x \leftarrow \Omega\, x; \vec{x}.\, \varphi \mid \dots\end{aligned}$$

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

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

Examples:

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

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

(where $\blacklozenge_I \varphi = \mathit{true}\, \mathsf{S}_I\, \varphi$)

Specifications

MONPOLY

Metric First-Order Temporal Logic (MFOTL)
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \dots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \,\mathsf{S}_I\, \varphi \mid \varphi \,\mathsf{U}_I\, \varphi \\ & \mid x \leftarrow \Omega\, x; \vec{x}.\, \varphi \mid \dots\end{aligned}$$

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

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

Examples:

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

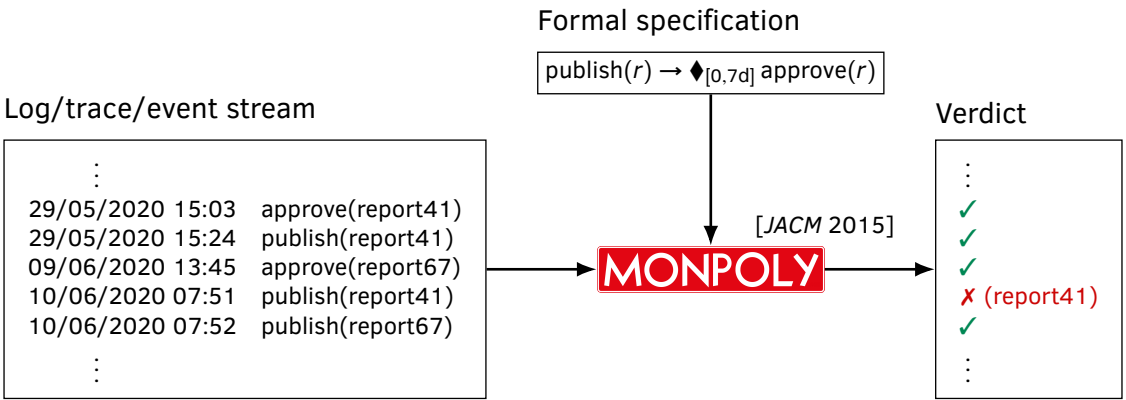
$$\mathsf{publish}(r) \rightarrow \blacklozenge_{[0,7\mathrm{d}]}\, \mathsf{approve}(r)$$

(where $\blacklozenge_I \varphi = \mathit{true}\, \mathsf{S}_I\, \varphi$)

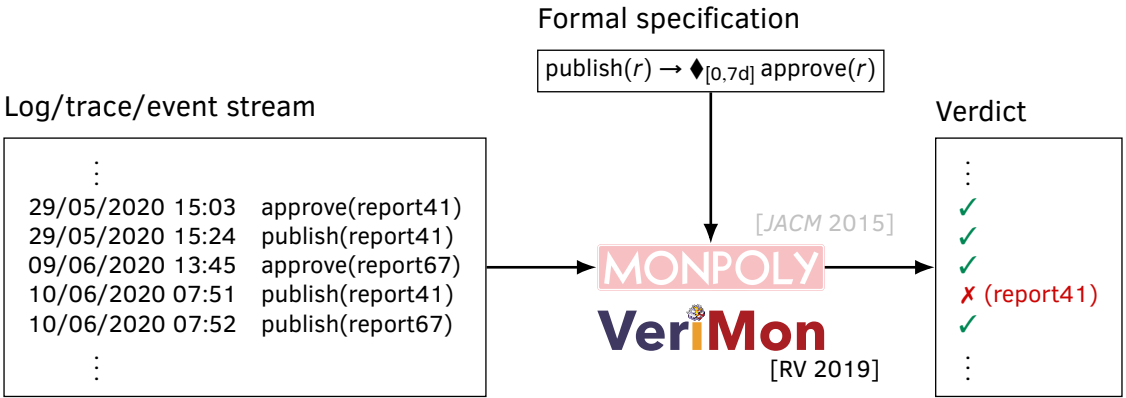
Maximum radiation must not exceed 3 Roentgen.

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

Monitors

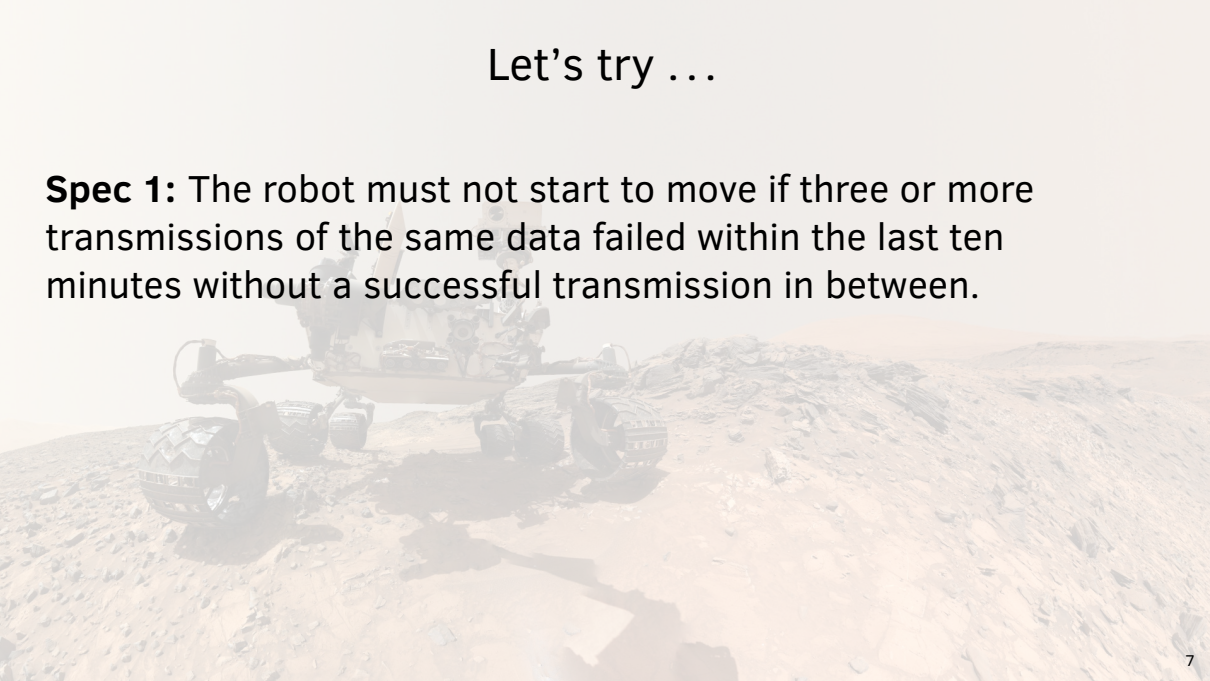

Formal specification
publish(r) → ♦[0,7d] approve(r)
Log/trace/event stream
29/05/2020 15:03 approve(report41)
29/05/2020 15:24 publish(report41)
09/06/2020 13:45 approve(report67)
10/06/2020 07:51 publish(report41)
10/06/2020 07:52 publish(report67)
[JACM 2015]
MONPOLY
Verdict
✓
✓
✓
✗ (report41)
✓

Monitors


Formal specification
publish(r) → ♦[0,7d] approve(r)
Log/trace/event stream
29/05/2020 15:03 approve(report41)
29/05/2020 15:24 publish(report41)
09/06/2020 13:45 approve(report67)
10/06/2020 07:51 publish(report41)
10/06/2020 07:52 publish(report67)
[JACM 2015]
MONPOLY
VeriMon
[RV 2019]
Verdict
✓
✓
✓
✗ (report41)
✓

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.

MONPOLY

MFOTL
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \dots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \, \mathsf{S}_I \, \varphi \mid \varphi \, \mathsf{U}_I \, \varphi \\ & \mid x \leftarrow \Omega \, t; \vec{x}.\, \varphi\end{aligned}$$

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

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

VeriMon

subset of MFOTL
no aggregations

$$t ::= x \mid c$$

$$\begin{aligned}\varphi ::= {} & p(t_1, \dots, t_n) \\ & \mid t = t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \, \mathsf{S}_I \, \varphi \mid \varphi \, \mathsf{U}_I \, \varphi\end{aligned}$$

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

MONPOLY

MFOTL
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \ldots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \, \mathsf{S}_I \, \varphi \mid \varphi \, \mathsf{U}_I \, \varphi \\ & \mid x \leftarrow \Omega \, t; \vec{x}.\, \varphi\end{aligned}$$

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

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

VeriMon
subset of MFOTL
no aggregations

$$t ::= x \mid c$$

$$\begin{aligned}\varphi ::= {} & p(t_1, \ldots, t_n) \\ & \mid t = t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \, \mathsf{S}_I \, \varphi \mid \varphi \, \mathsf{U}_I \, \varphi\end{aligned}$$

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

cannot express Spec 2!

Spec 1 in MFOTL:

$$\Big(\bigvee_{\substack{x \in \mathbf{N}^6, \\ \sum_i x_i = 600}} \bullet_{[0,x_1]}(\neg \text{com\_ok}(d)\ \mathsf{S}_{[0,x_2]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,x_3]}(\neg \text{com\_ok}(d)\ \mathsf{S}_{[0,x_4]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,x_5]}(\neg \text{com\_ok}(d)\ \mathsf{S}_{[0,x_6]}\ \text{com\_fail}(d)))))) \Big) \rightarrow \neg \text{move}$$

VeriMon
subset of MFOTL
no aggregations

c

$_1, \ldots, t_n)$

t

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

$\varphi \mid \bigcirc_I \varphi \mid \varphi\ \mathsf{S}_I\ \varphi \mid \varphi\ \mathsf{U}_I\ \varphi$

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

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

Spec 1 in MFOTL:

$(\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 \ldots$

664 353 676 371 disjuncts, assuming discrete time in seconds

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

VeriMon
subset of MFOTL
no aggregations

c

$1, \ldots, t_n)$

t

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

$\varphi \mid \bigcirc_I \varphi \mid \varphi\ S_I\ \varphi \mid \varphi\ U_I\ \varphi$

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

MIND THE GAP

VeriMon

subset of MFOTL

ns

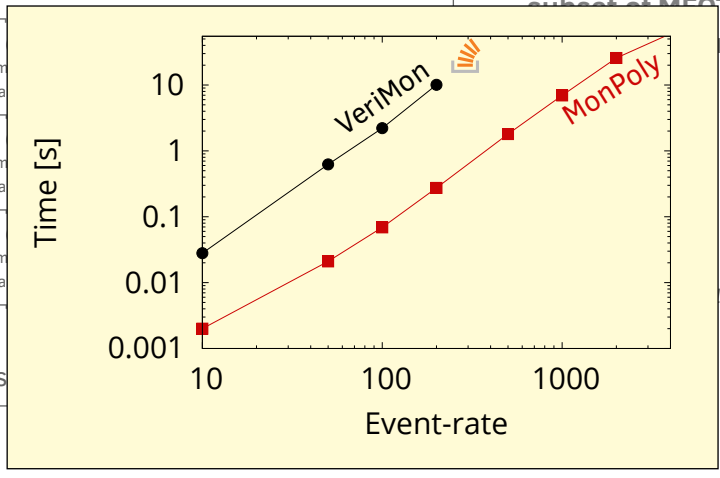
Spec 1 in MFOTL:

$(\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg$
$\text{com\_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}$
$(\text{com\_fail}(d) \wedge \bullet_{[0,1]}(\neg \text{com\_ok}(d)\ S_{[0,599]}\ \text{com}$
$(\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fa}$
$(\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg$
$\text{com\_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}$
$(\text{com\_fail}(d) \wedge \bullet_{[0,4]}(\neg \text{com\_ok}(d)\ S_{[0,596]}\ \text{com}$
$(\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fa}$
$(\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg$
$\text{com\_fail}(d)))))) \vee (\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}$
$(\text{com\_fail}(d) \wedge \bullet_{[0,7]}(\neg \text{com\_ok}(d)\ S_{[0,593]}\ \text{com}$
$(\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fa}$
$(\bullet_{[0,0]}(\neg \text{com\_ok}(d)\ S_{[0,0]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,0]}(\neg$
$\text{com\_fail}(d)))))) \vee \ldots$

664 353 676 371 disjuncts, as

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

φ



Act II: Groundwork on Expressiveness

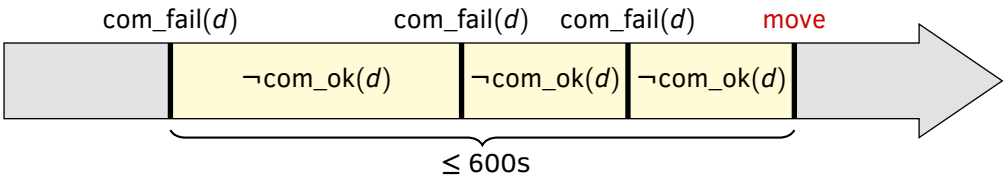
Joshua
Monitoring researcher

Srđan
Working formalizer

Where: VeriMon Headquaters
When: 5 days before the IJCAR deadline

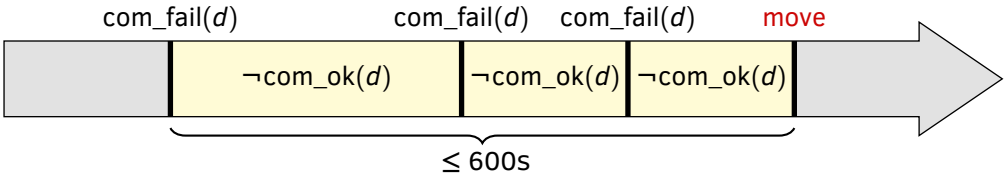
Regular expressions

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.

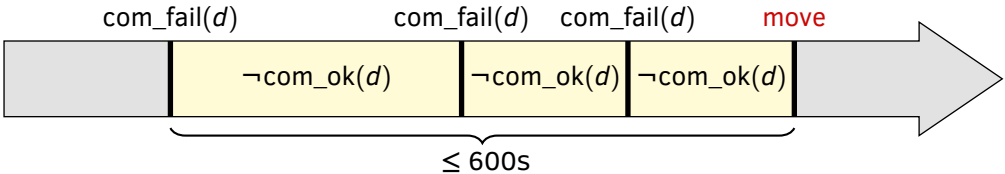


In MFOTL:

$$\Big(\bigvee_{\substack{x\in\mathbf{N}^6,\\ \sum_i x_i=600}} \bullet_{[0,x_1]}(\neg\text{com\_ok}(d)\ S_{[0,x_2]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,x_3]}(\neg\text{com\_ok}(d)\ S_{[0,x_4]}\ (\text{com\_fail}(d) \wedge \bullet_{[0,x_5]}(\neg\text{com\_ok}(d)\ S_{[0,x_6]}\ \text{com\_fail}(d))))))\Big) \rightarrow \neg\text{move}$$

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.

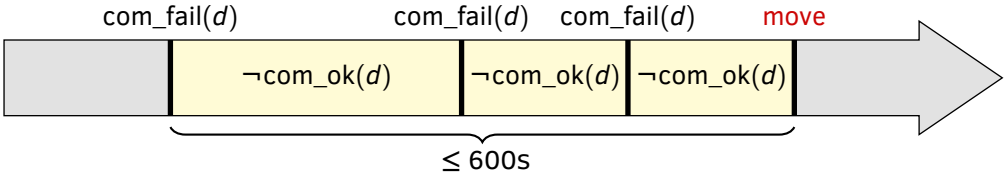


The new operator: $\blacktriangleleft_{Interval}$ *Pattern*

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 MFO***D***L: $\big(\blacktriangleleft_{[0,600]}\ \mathrm{com\_fail}(d) \cdot (\neg \mathrm{com\_ok}(d))^* \cdot \mathrm{com\_fail}(d) \cdot (\neg \mathrm{com\_ok}(d))^* \cdot \mathrm{com\_fail}(d) \cdot (\neg \mathrm{com\_ok}(d))^*\ \big) \rightarrow \neg \mathrm{move}$

DL stands for Dynamic Logic, as in LDL [De Giacomo & Vardi 2013]

Metric First-Order *Dynamic* Logic

Formulas:

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

$\mid \blacktriangleleft_I e$ past match

$\mid \triangleright_I e$ future match

Metric First-Order *Dynamic* Logic

Formulas:

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

$\mid \blacktriangleleft_I e$ past match

$\mid \triangleright_I e$ future match

Regular expressions:

$e ::= \_$ wildcard

$\mid \varphi?$ test formula

$\mid e + e$ alternation

$\mid e \cdot e$ concatenation

$\mid e^*$ Kleene star

Metric First-Order *Dynamic* Logic

Formulas:

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

$\mid \blacktriangleleft_I e$ past match

$\mid \triangleright_I e$ future match

Regular expressions:

$e ::= \_$ wildcard

$\mid \varphi?$ test formula

$\mid e + e$ alternation

$\mid e \cdot e$ concatenation

$\mid e^*$ Kleene star

$\blacktriangleleft_I \mathrm{com\_fail}(d) \cdot (\neg \mathrm{com\_ok}(d))^*$

abbreviates

$\blacktriangleleft_I (\mathrm{com\_fail}(d)? \cdot \_) \cdot ((\neg \mathrm{com\_ok}(d))? \cdot \_)^*$

Metric First-Order *Dynamic* Logic

Formulas:

$\varphi ::= \ldots$	all existing constructs
$\mid \blacktriangleleft_I e$	past match
$\mid \triangleright_I e$	future match

Regular expressions:

$e ::= \_$	wildcard
$\mid \varphi?$	test formula
$\mid e + e$	alternation
$\mid e \cdot e$	concatenation
$\mid e^*$	Kleene star

$\blacktriangleleft_I \mathrm{com\_fail}(d) \cdot (\neg \mathrm{com\_ok}(d))^*$

abbreviates

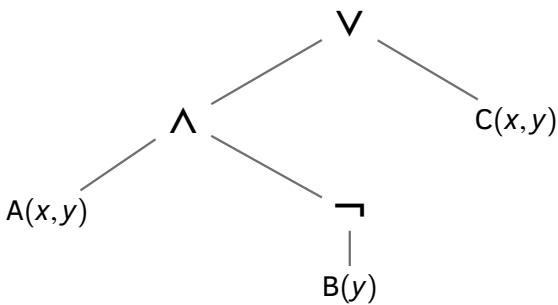
$\blacktriangleleft_I (\mathrm{com\_fail}(d)? \cdot \_) \cdot ((\neg \mathrm{com\_ok}(d))? \cdot \_)^*$

Note:

$$\bullet_I \alpha \equiv \blacktriangleleft_I (\alpha? \cdot \_)$$
$$\alpha \, \mathsf{S}_I \, \beta \equiv \blacktriangleleft_I (\beta? \cdot (\_ \cdot \alpha?)^*)$$
$$\circ_I \alpha \equiv \triangleright_I (\_ \cdot \alpha?)$$
$$\alpha \, \mathsf{U}_I \, \beta \equiv \triangleright_I ((\alpha? \cdot \_)^* \cdot \beta?)$$

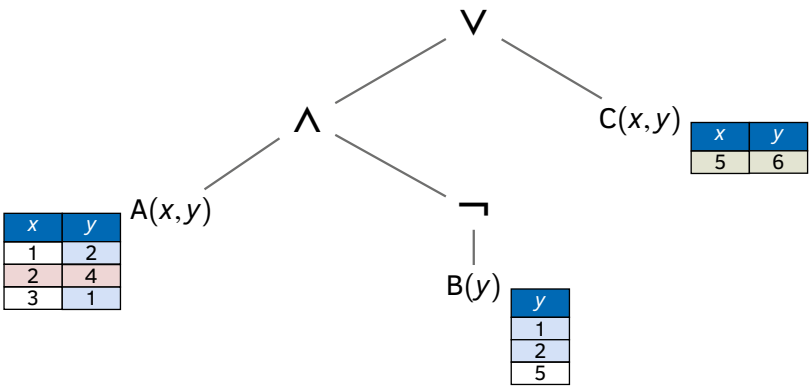
Recap: Evaluation in VeriMon

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



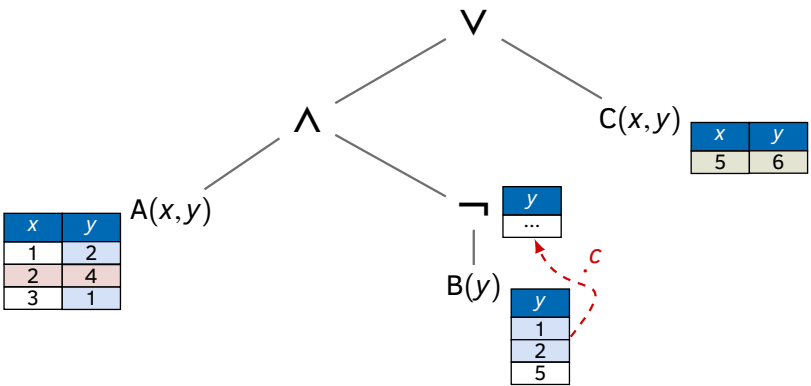
Recap: Evaluation in VeriMon

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



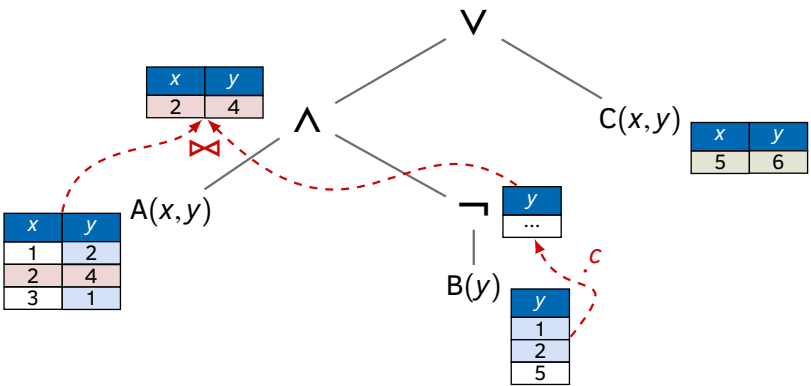
Recap: Evaluation in VeriMon

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



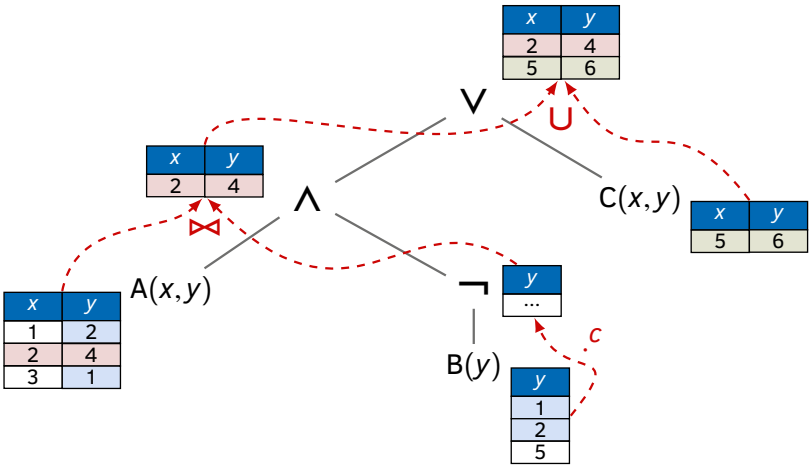
Recap: Evaluation in VeriMon

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



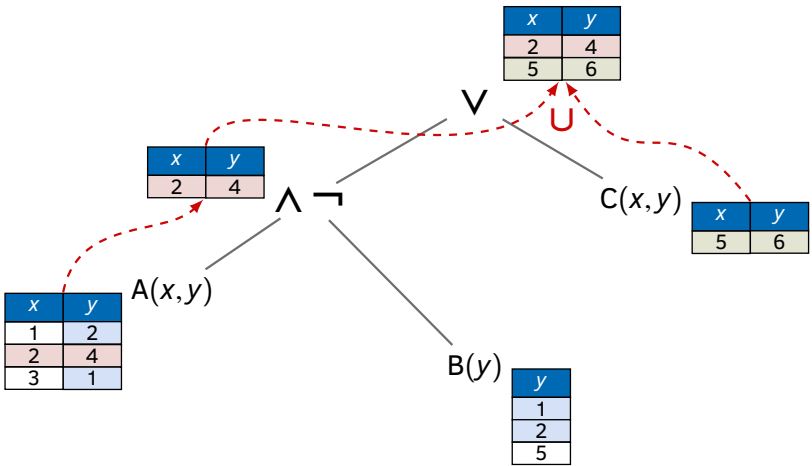
Recap: Evaluation in VeriMon

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



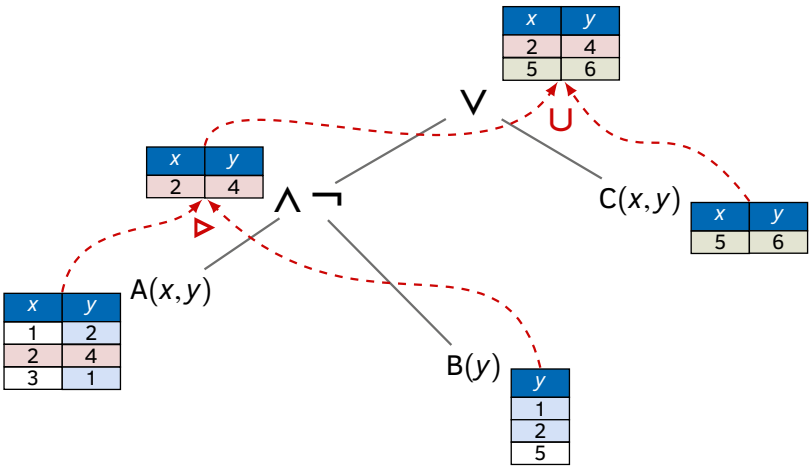
Recap: Evaluation in VeriMon

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



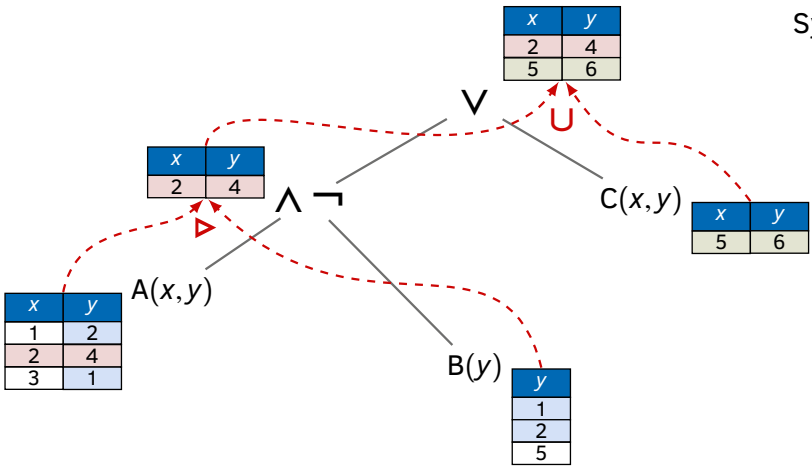
Recap: Evaluation in VeriMon

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 $\mathrm{FV}(r)$

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 $\mathrm{FV}(r)$

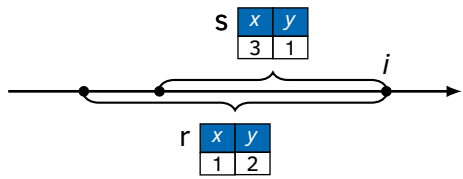
Finitely Evaluable Regular Expressions

Kleene star must be guarded

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

Alternation

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



$v \in$ | x | y | 1 | 2 | $\cup$ | x | y | 3 | 1 |

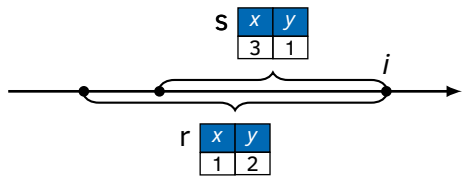
Finitely Evaluable Regular Expressions

Kleene star must be guarded

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

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

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



Finitely Evaluable Regular Expressions

Kleene star must be guarded

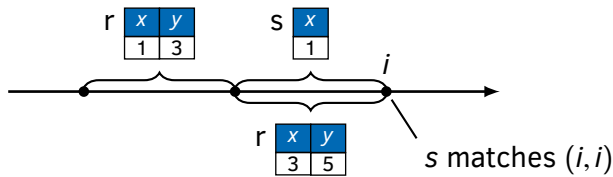
Example: $(v,i) \models \blacktriangleleft_{[0,\infty)} r^*$

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

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

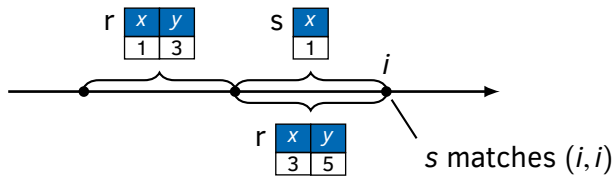
Example: $(v,i) \models \blacktriangleleft_{[0,\infty)} r^*$

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

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

Concatenation $\mathrm{FV}(r) \supseteq \mathrm{FV}(s)$

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 must be guarded

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

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

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

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

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

MONPOLY

MFOTL
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \ldots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \,\mathsf{S}_I\, \varphi \mid \varphi \,\mathsf{U}_I\, \varphi \\ & \mid x \leftarrow \Omega\, t; \vec{x}.\, \varphi \mid \ldots\end{aligned}$$

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

VeriMon+

MFO**D**L
with aggregations

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

$$\begin{aligned}\varphi ::= {} & p(t_1, \ldots, t_n) \\ & \mid t = t \mid t < t \mid t \leq t \\ & \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi \\ & \mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi \,\mathsf{S}_I\, \varphi \mid \varphi \,\mathsf{U}_I\, \varphi \\ & \mid \blacktriangleleft_I e \mid \triangleright_I e \\ & \mid x \leftarrow \Omega\, x; \vec{x}.\, \varphi \mid \ldots\end{aligned}$$

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

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

MONPOLY

MFOTL

with aggregations

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

$$\varphi ::= p(t_1, \ldots, t_n)$$
$$\mid t = t \mid t < t \mid t \leq t$$
$$\mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi$$
$$\mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi\, \mathsf{S}_I\, \varphi \mid \varphi\, \mathsf{U}_I\, \varphi$$
$$\mid x \leftarrow \Omega\, t\, \vec{\ldots}$$

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

can easily express Specs 1 & 2

VeriMon+

MFO**D**L

with aggregations

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

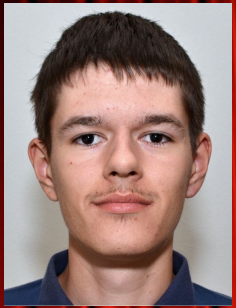
$$\varphi ::= p(t_1, \ldots, t_n)$$
$$\mid t = t \mid t < t \mid t \leq t$$
$$\mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\, \varphi$$
$$\mid \bullet_I \varphi \mid \bigcirc_I \varphi \mid \varphi\, \mathsf{S}_I\, \varphi \mid \varphi\, \mathsf{U}_I\, \varphi$$
$$\mid \blacktriangleleft_I e \mid \triangleright_I e$$
$$\mid x \leftarrow \Omega\, x; \vec{x}.\, \varphi \mid \ldots$$

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

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

Act III: Let's Make It Fly

Srđan
Working formalizer

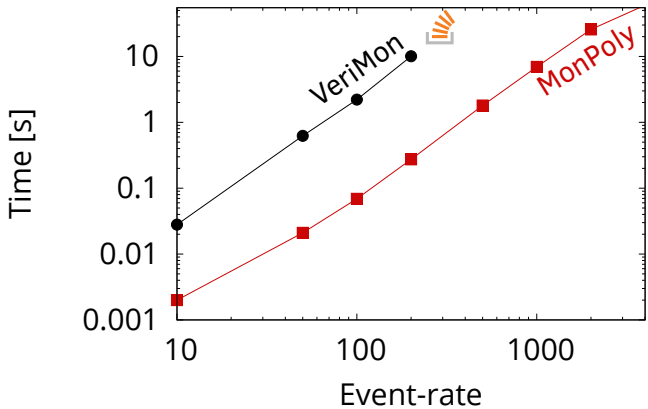


Martin
Quality assurer

Where: Implementer's Den
When: 3 days before the IJCAR deadline

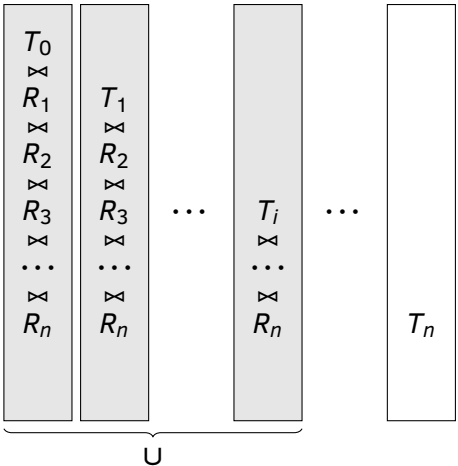
VeriMon Benchmark

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



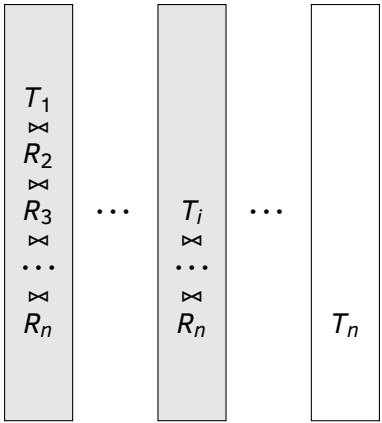
Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon



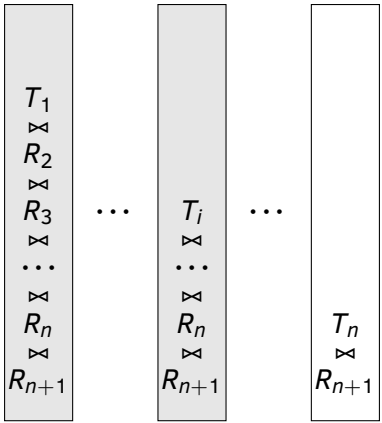
Sliding Window: $R(x)\ \mathrm{S}_{[a,b]}\ T(x)$

VeriMon



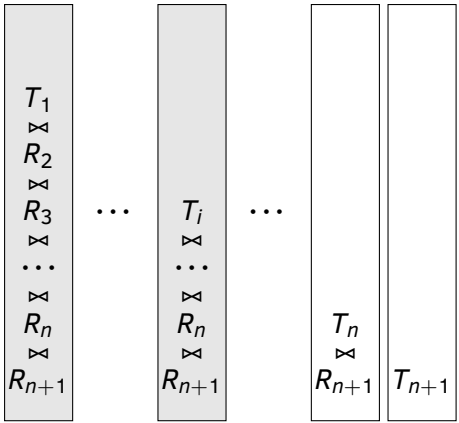
Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon



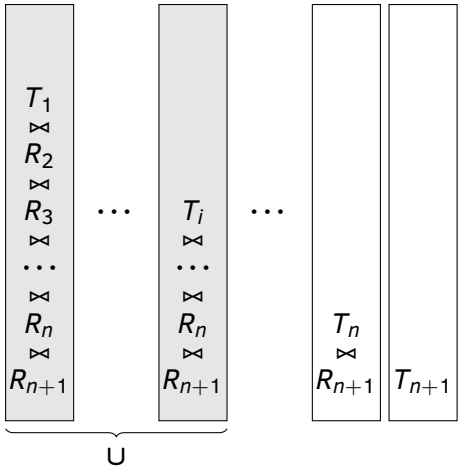
Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon



Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

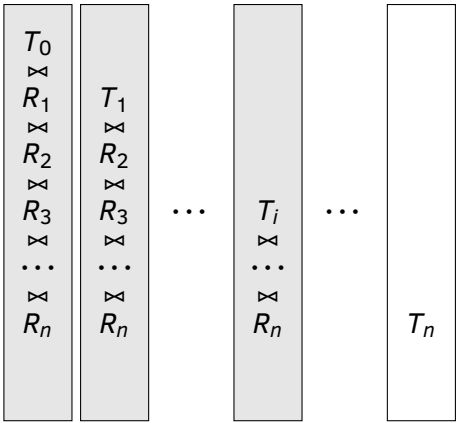
VeriMon



Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon

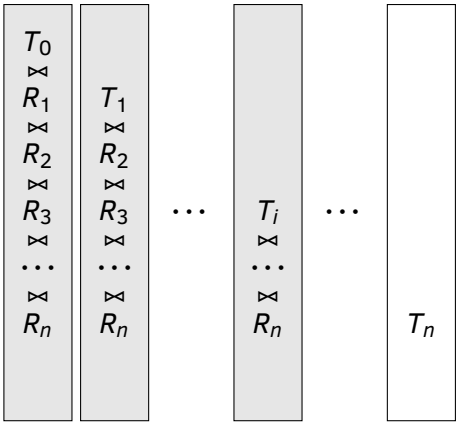
VeriMon+



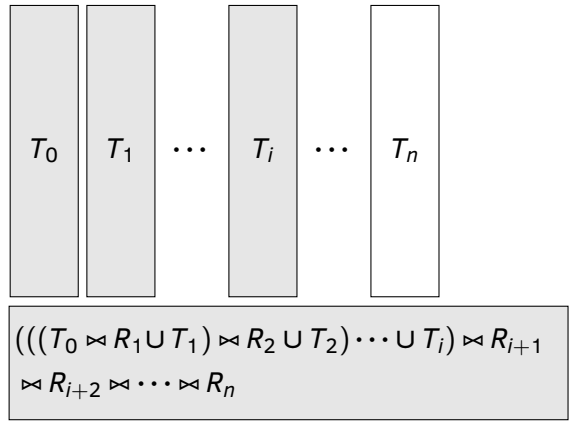
$$(((T_0 \bowtie R_1 \cup T_1) \bowtie R_2 \cup T_2) \cdots \cup T_i) \bowtie R_{i+1} \bowtie R_{i+2} \bowtie \cdots \bowtie R_n$$

Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon

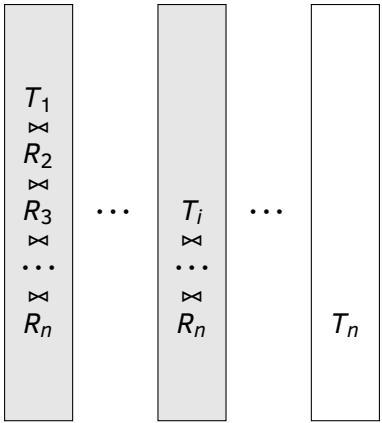


VeriMon+

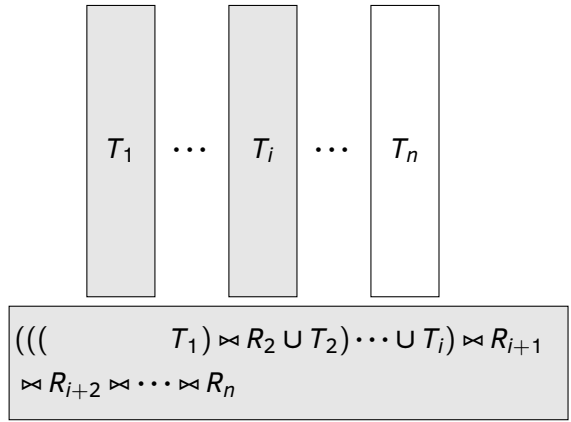


Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon

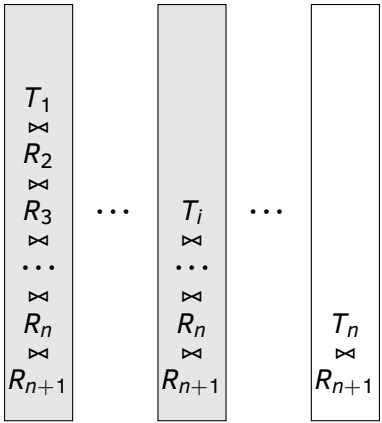


VeriMon+

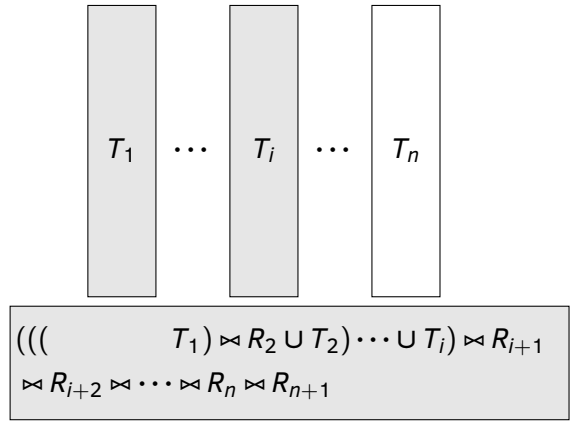


Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon

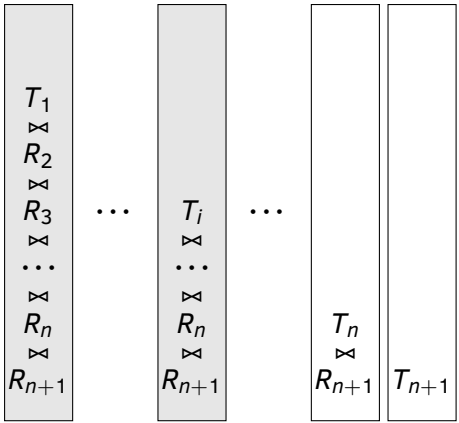


VeriMon+

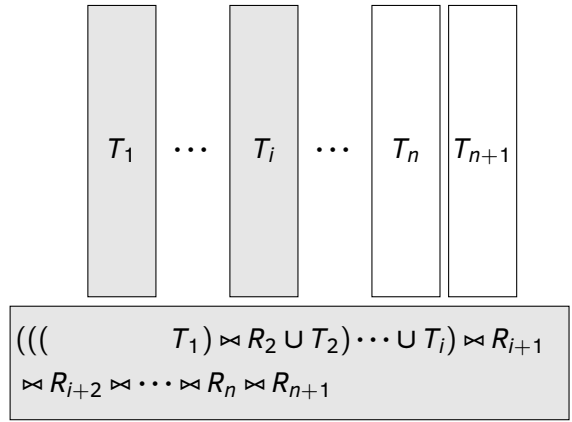


Sliding Window: $R(x)\ \mathsf{S}_{[a,b]}\ T(x)$

VeriMon

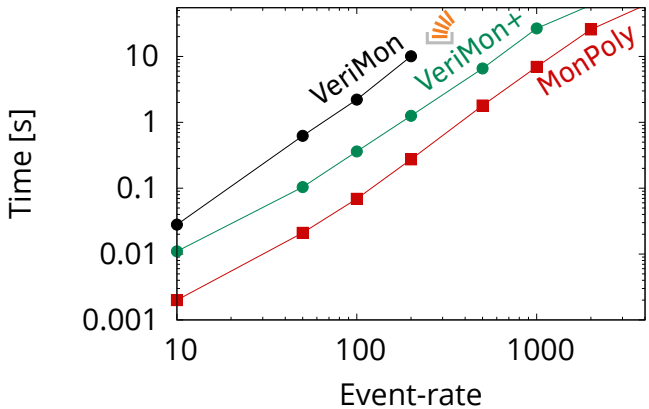


VeriMon+



VeriMon+ Benchmark

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

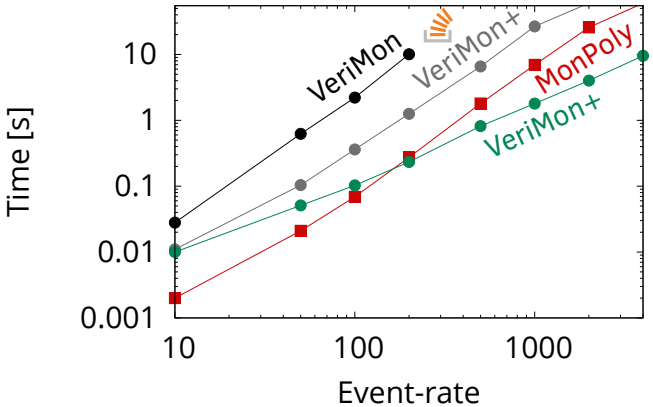


Multi-Way Join

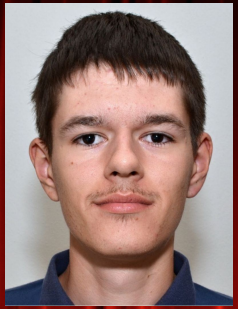
$$((\blacklozenge_{[0,30]}\, P(x,y)) \wedge Q(x,z)) \wedge (\Diamond_{[0,30]}\, R(x,w))$$

Multi-Way Join

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



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.

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

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

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

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

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

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

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

```
(e <- MAX e; (ONCE[60,120] E(e,id))) AND (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.

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

```
(e <- MAX e; (ONCE[60,120] E(e,id))) AND (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.

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

```
(e <- MAX e; (ONCE[60,120] E(e,id))) AND (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(int,string)
```

✔ 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

Correct

- 15 000
- extract

Expressive

- aggregations
- regular expressions

Efficient

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

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

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

VeriMon+

Correct

- 15
- ex

Expressive

- egations
- ar expressions

Efficient

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

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

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
- regu

Efficient

- ding window (S, U)
- ulti-way join

$$\blacktriangleleft_{[a,b]}\ \psi? \cdot (\_ \cdot \varphi?)^*$$

VeriMon++

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

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

VeriMon+

Correct

- 15 000 lines of
- extraction to OCaml

Expressive

-
-

Efficient

- window (S, U)
- way join

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

VeriMon++

- verified parsing
- extraction to LLVM

- unsafe formulas
- recursive let

- sliding window (◀, ▷)
- 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