Approximate Distributed Monitoring under Partial Synchrony

Balancing Speed & Accuracy

Borzoo Bonakdarpour, Anik Momtaz, Dejan Ničković, N. Ege Saraç

MICHIGAN STATE UNIVERSITY

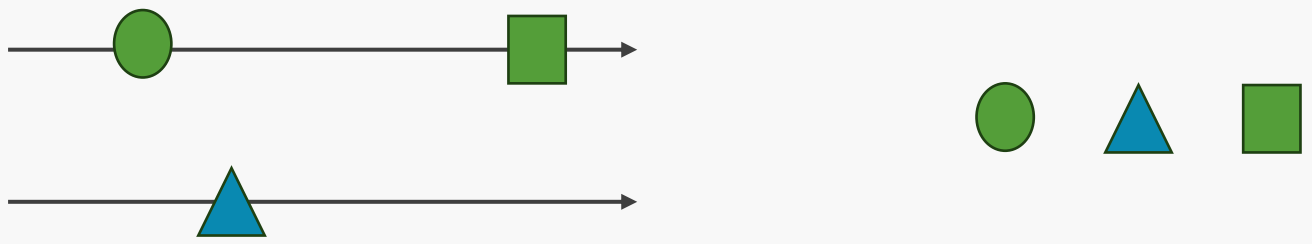
AUSTRIAN INSTITUTE OF TECHNOLOGY

Institute of Science and Technology Austria

Motivation

Monitoring distributed systems.

- Synchronous: too strong
- Asynchronous: combinatorial blow-up + too weak
- Partially synchronous: still prone to combinatorial blow-up
  + more synchrony = more cost

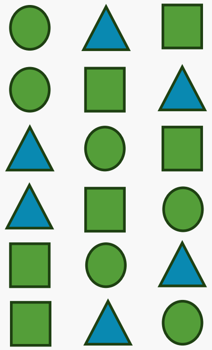


Motivation

Monitoring distributed systems.

- Synchronous: too strong
- Asynchronous: combinatorial blow-up + too weak
- Partially synchronous: still prone to combinatorial blow-up

  + more synchrony = more cost

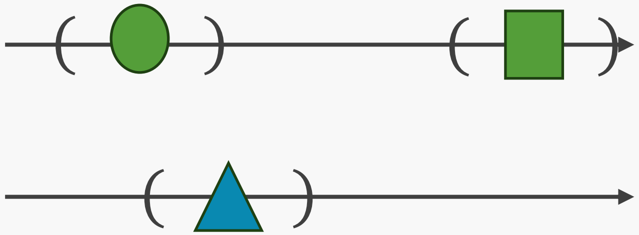


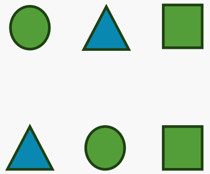
Motivation

Monitoring distributed systems.

- Synchronous: too strong
- Asynchronous: combinatorial blow-up + too weak
- Partially synchronous: still prone to combinatorial blow-up

  + more synchrony = more cost





Motivation

Monitoring distributed systems.

- Synchronous: too strong
- Asynchronous: combinatorial blow-up + too weak
- Partially synchronous: still prone to combinatorial blow-up
  + more synchrony = more cost

Contribution.
A modular, approximate monitoring algorithm for STL specifications on partially synchronous systems, which performs up to $x10^5$ faster.

Signal Temporal Logic (STL)

Signals. $x\colon [0, d) \to \mathbb{R}$ (right-continuous, non-Zeno, piecewise-constant)

STL syntax. $\psi \coloneqq p \mid \neg\psi \mid \psi \wedge \psi \mid \psi\, U_I \psi$

STL semantics.

$$
\begin{aligned}
(w,t) &\models p && \Leftrightarrow f_p\big(x_1(t), \dots, x_n(t)\big) > 0 \\
(w,t) &\models \neg\psi_1 && \Leftrightarrow \overline{(w,t) \models \psi_1} \\
(w,t) &\models \psi_1 \wedge \psi_2 && \Leftrightarrow (w,t) \models \psi_1 \wedge (w,t) \models \psi_2 \\
(w,t) &\models \psi_1 U_I \psi_2 && \Leftrightarrow \exists t' \in (t \oplus I) \cap [0,d) : \\
& && \qquad (w,t') \models \psi_2 \ \wedge\ \forall t'' \in (t,t') : (w,t'') \models \psi_1
\end{aligned}
$$

Distributed Semantics of STL

Partially synchrony.

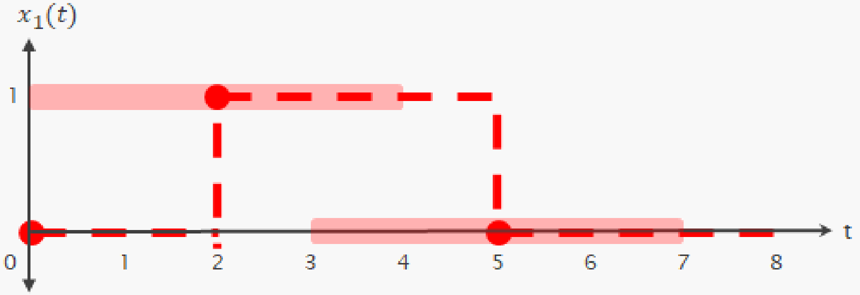
- $|c_i(T) - c_j(T)| <$ ε for all $1 \leq i, j \leq n$ and all $T \in \mathbb{R}_{\geq 0}$
    - Local clocks $c_1, \dots, c_n : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$
    - Maximum clock skew $\varepsilon > 0$

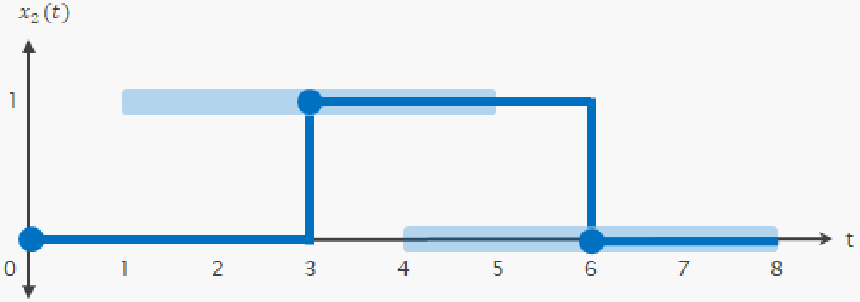
Distributed signals.

- $(S, \rightsquigarrow)$
    - A set of signals $S = \{x_1, \dots, x_n\}$
    - Happened-before relation $\rightsquigarrow$

$\uparrow_{x_1} \rightsquigarrow \downarrow_{x_1}$ and $\uparrow_{x_2} \rightsquigarrow \downarrow_{x_2}$ and $\uparrow_{x_1} \rightsquigarrow \downarrow_{x_2}$

Others are concurrent





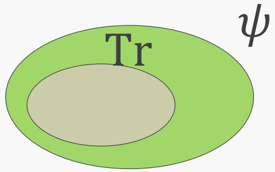
Distributed Semantics of STL

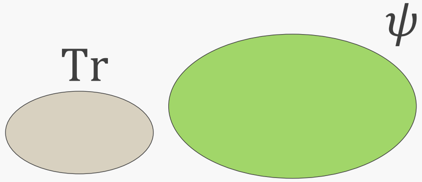
Traces of distributed signals.

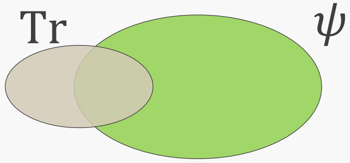
- $\mathrm{Tr}(S, \rightsquigarrow)$ = set of (synchronous) signals that are "consistent" with $\rightsquigarrow$

STL distributed semantics.

$$[(S, \rightsquigarrow) \vDash \psi] = \begin{cases} \top & \text{if } \forall w \in \mathrm{Tr}(S, \rightsquigarrow) : (w, t) \vDash \psi \\ \bot & \text{if } \forall w \in \mathrm{Tr}(S, \rightsquigarrow) : (w, t) \vDash \neg\psi \\ ? & \text{otherwise} \end{cases}$$







Our Approach

$$[(S, \rightsquigarrow) \vDash \psi]_+ = \begin{cases} \top & \text{if } \forall w \in \mathrm{Tr}^+(S, \rightsquigarrow) : (w, t) \vDash \psi \\ \bot & \text{if } \forall w \in \mathrm{Tr}^+(S, \rightsquigarrow) : (w, t) \vDash \neg\psi \\ ? & \text{otherwise} \end{cases}$$

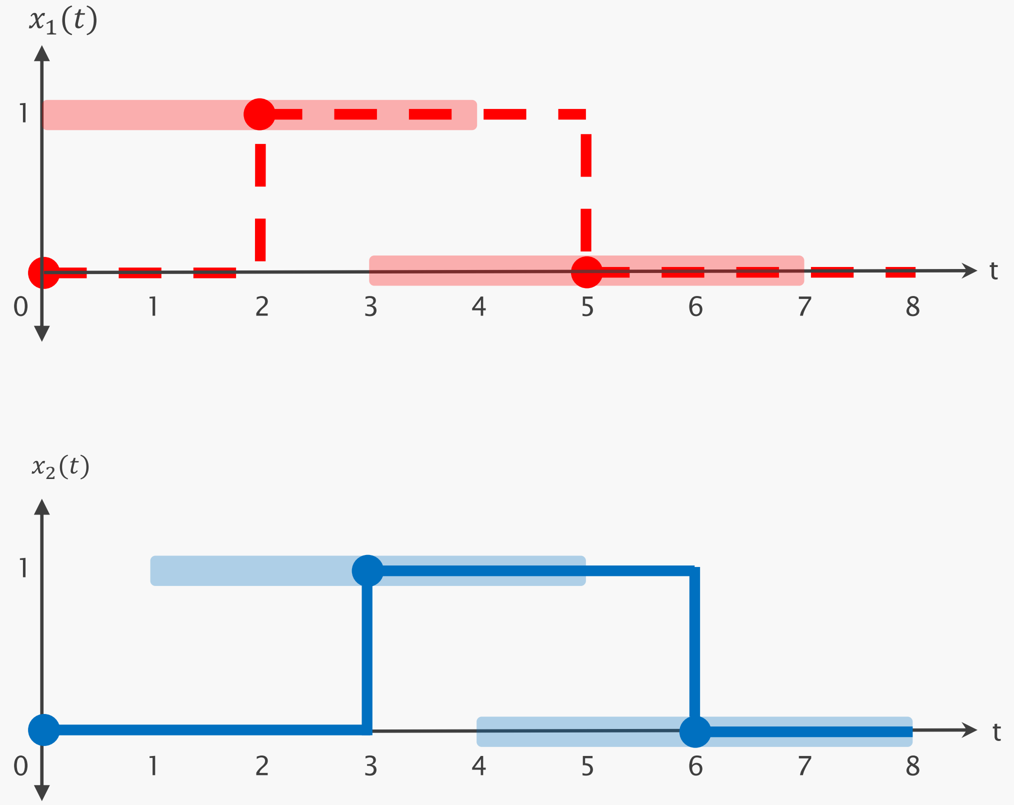
1. Define an overapproximation Tr^+ of Tr
2. Compute $[(S, \rightsquigarrow) \vDash \psi]_+$ fast

Defining Tr^+

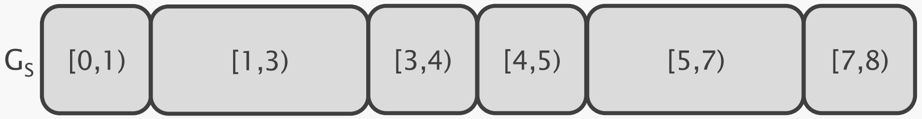
Idea.

- Partition $[0, d)$ into asynchronous segments
  - "canonical segmentation"
- Encode potential signal behaviors for each segment independently
  - "value expressions"
- Tr^+ = set of signals "consistent" with this encoding
  - Ignores most dependencies between segments

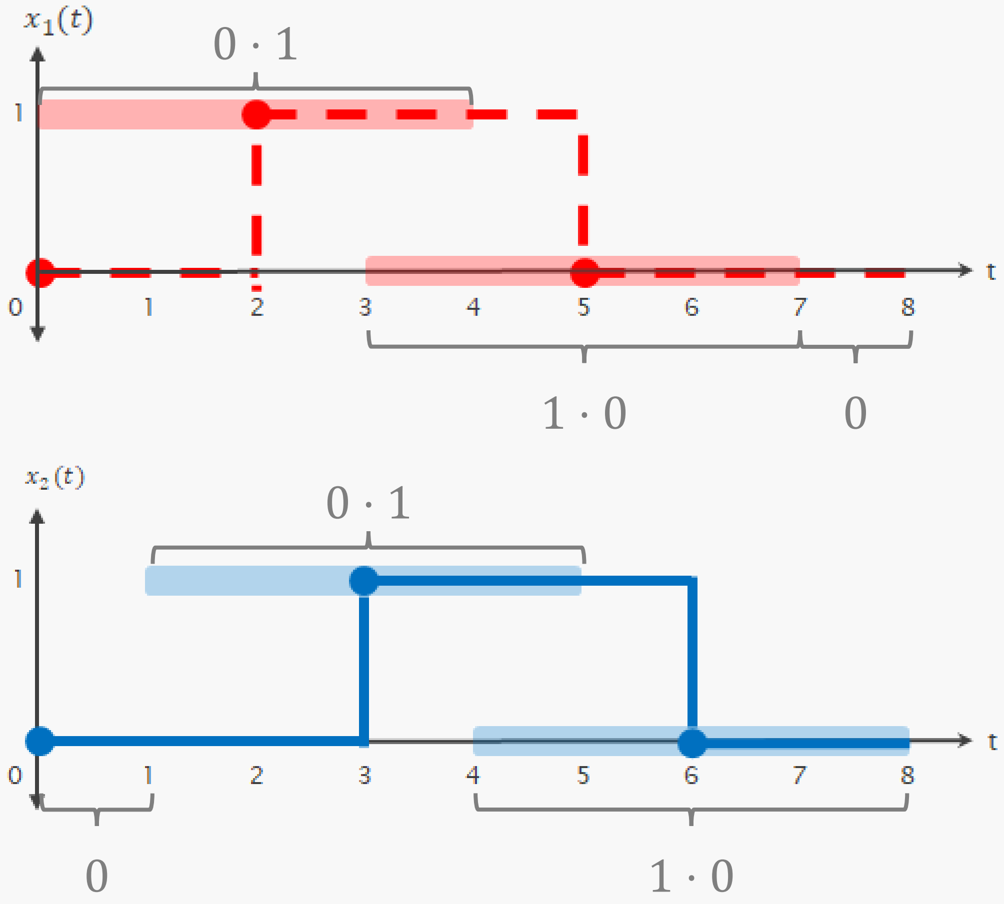
Defining Tr^+: Canonical Segmentation



<u>Canonical segmentation G_S:</u>

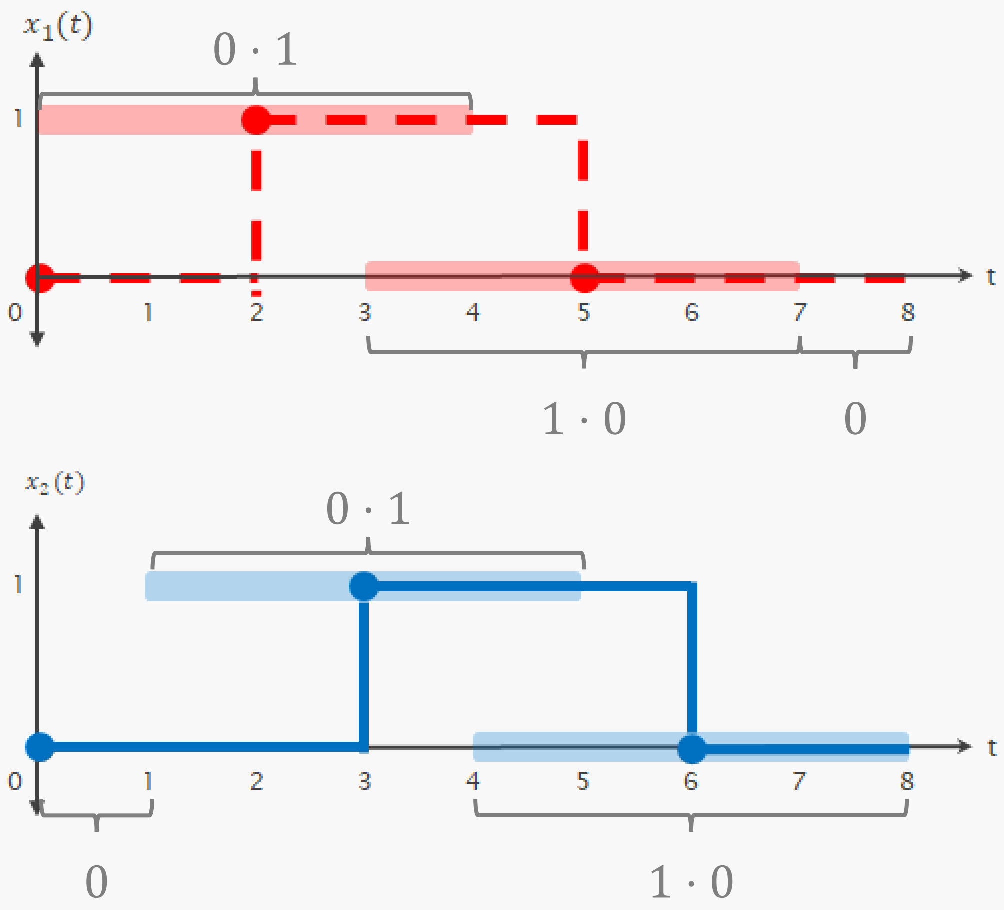


Defining Tr^+: Value Expressions

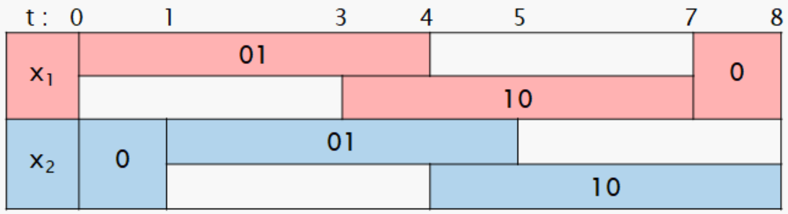


<u>Value expressions:</u> Finite words over the finite set of signal values

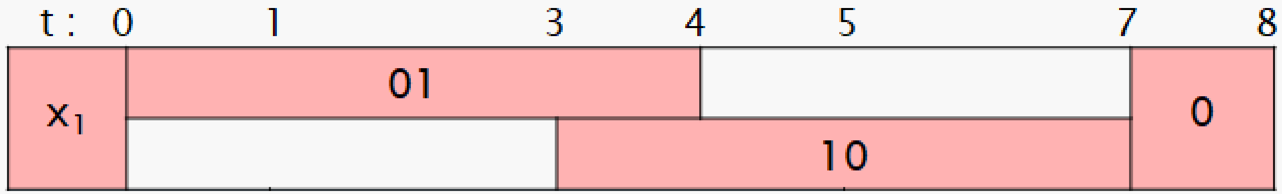
Defining Tr^+: Value Expressions

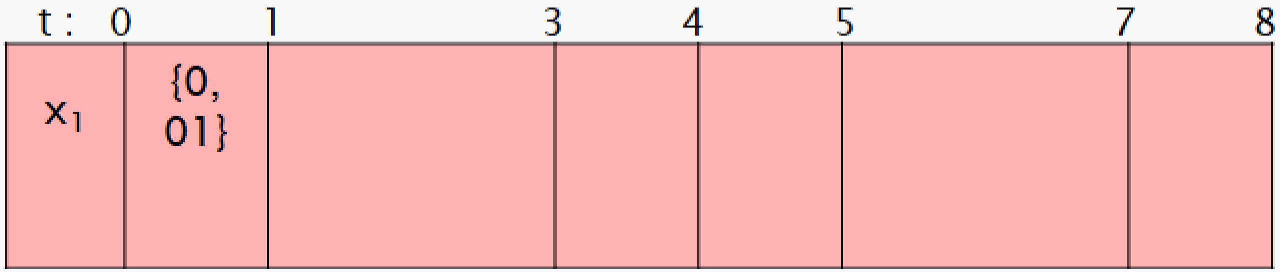


Value expressions: Finite words over the finite set of signal values



Defining Tr^+: Value Expressions

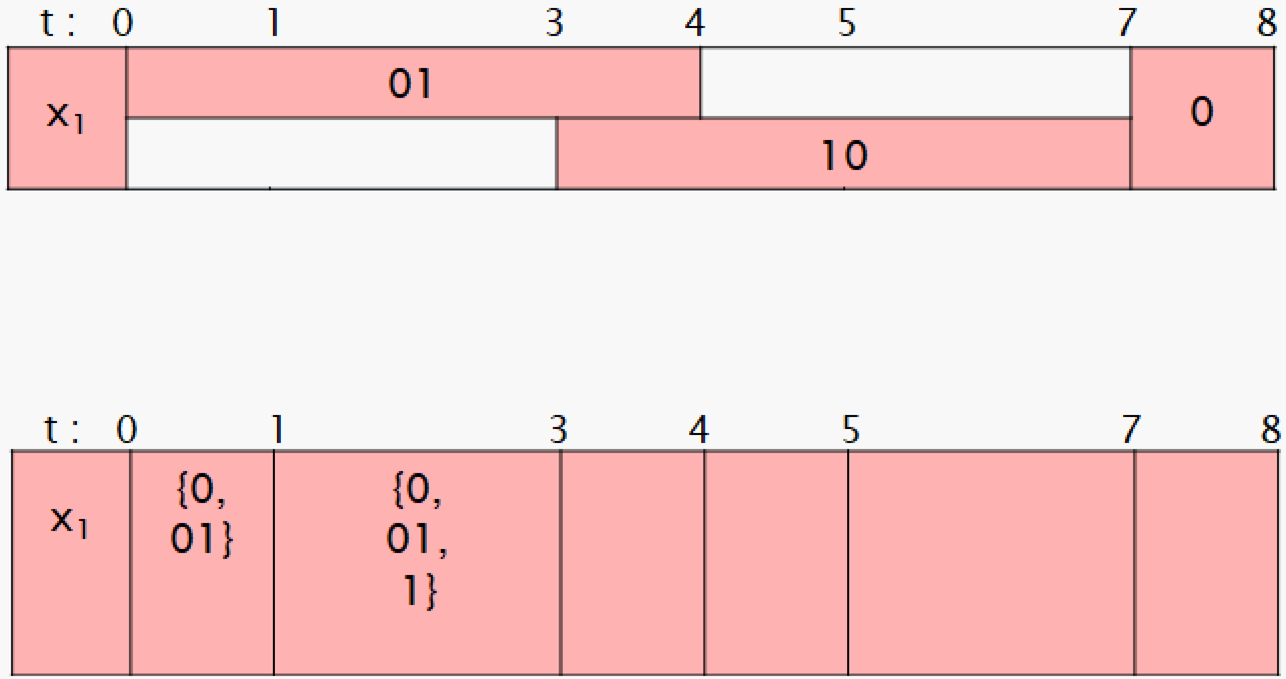




pref(01)

<u>Value expressions on G_S</u>: Obtained by combining initial value expression (concatenation, prefix, infix, suffix)

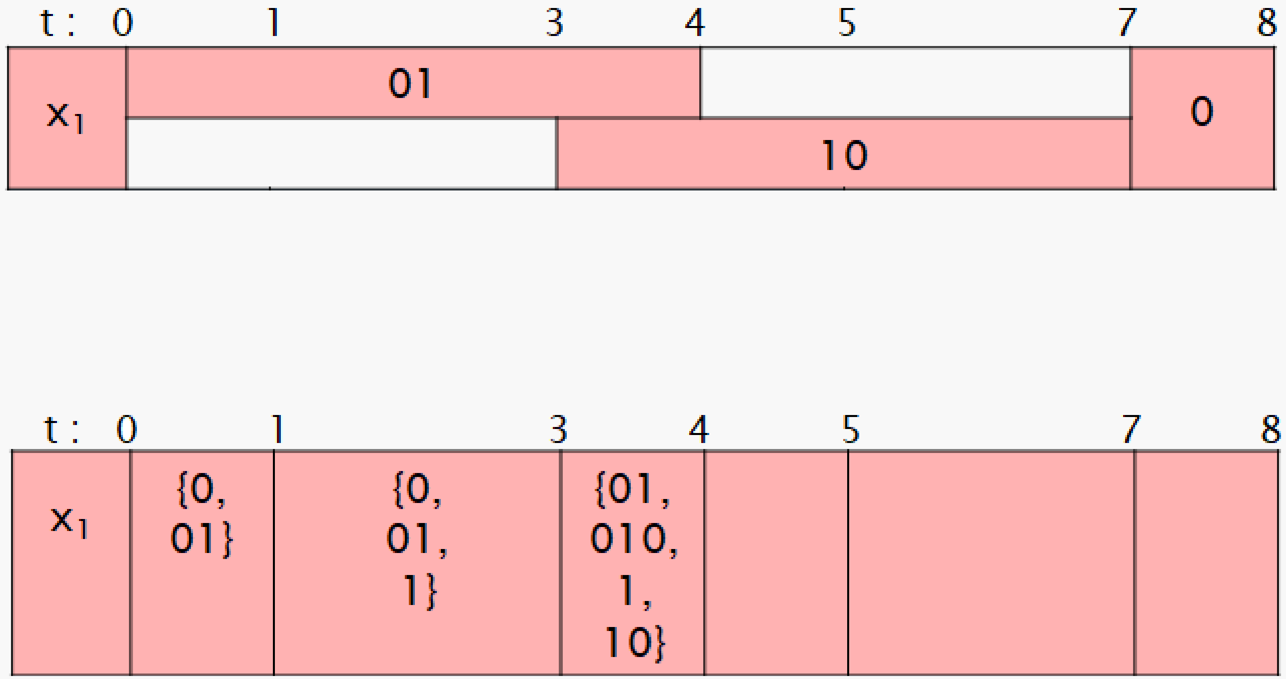
Defining Tr^+: Value Expressions



inf(01)

<u>Value expressions on G_S</u>: Obtained by combining initial value expression (concatenation, prefix, infix, suffix)

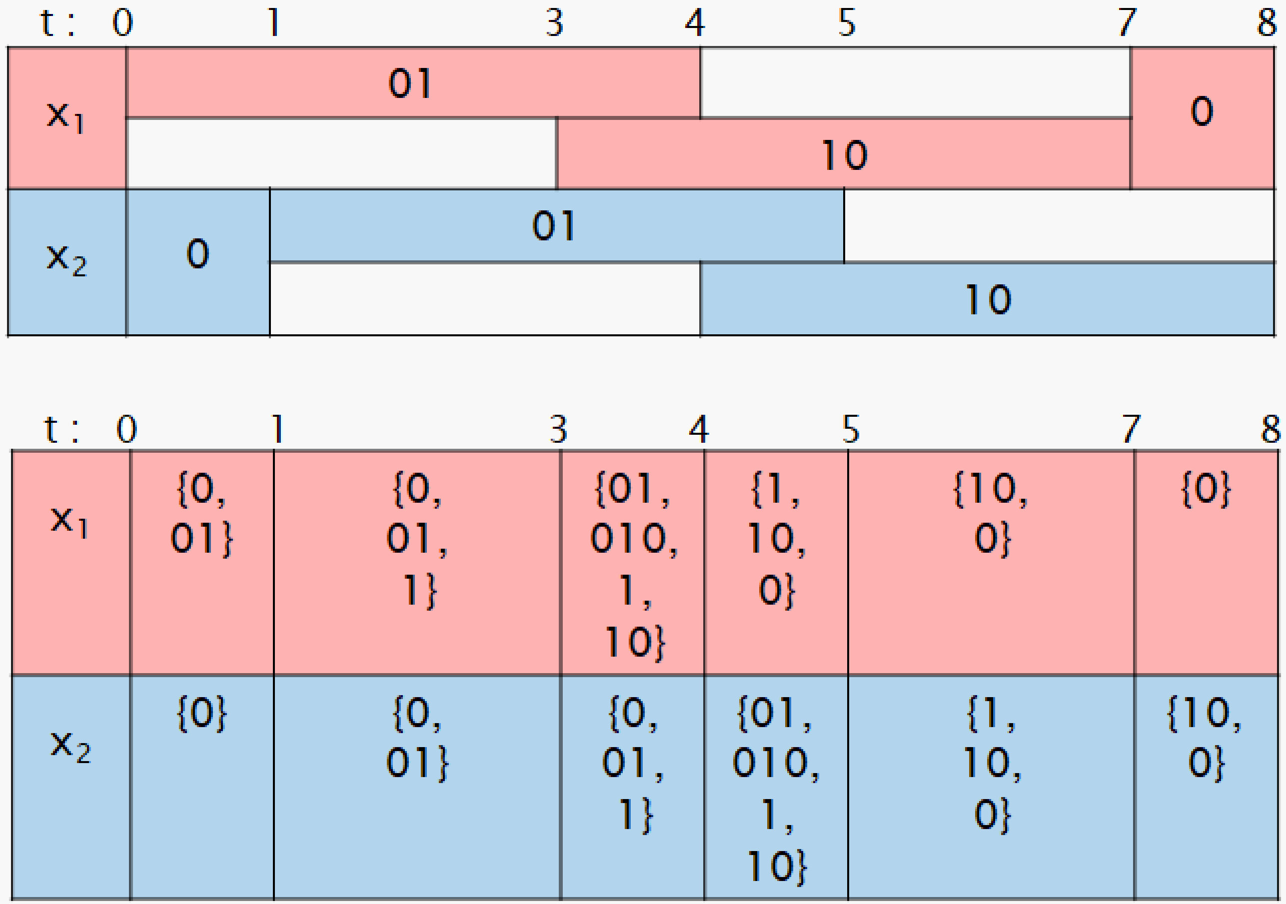
Defining Tr^{+}: Value Expressions



$\mathrm{suf}(01) \cdot \mathrm{pre}(10)$

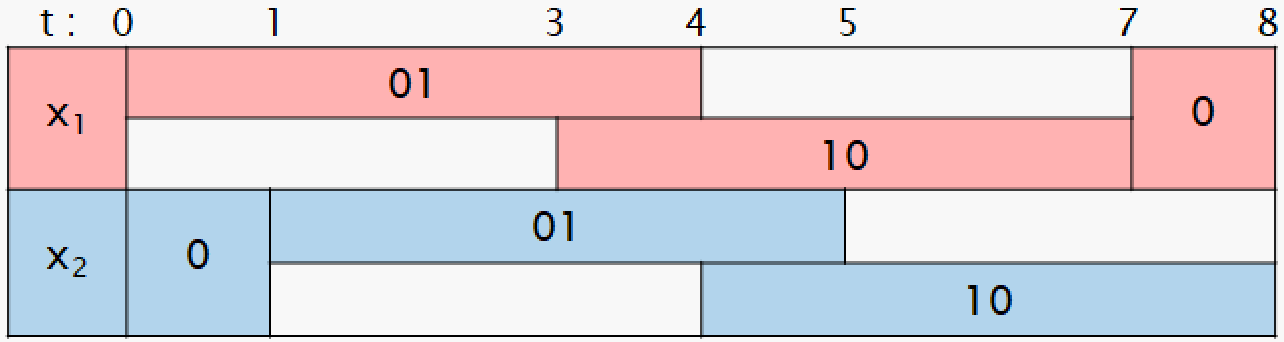
Value expressions on G_S: Obtained by combining initial value expression (concatenation, prefix, infix, suffix)

Defining Tr^+: Value Expressions



<u>Value expressions on G_S</u>: Obtained by combining initial value expression (concatenation, prefix, infix, suffix)

Defining Tr^{+}: Value Expressions



t:	0–1	1–3	3–4	4–5	5–7	7–8
x_1	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}	{0}
x_2	{0}	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}

<u>Value expressions on G_S</u>: Obtained by combining initial value expression (concatenation, prefix, infix, suffix)

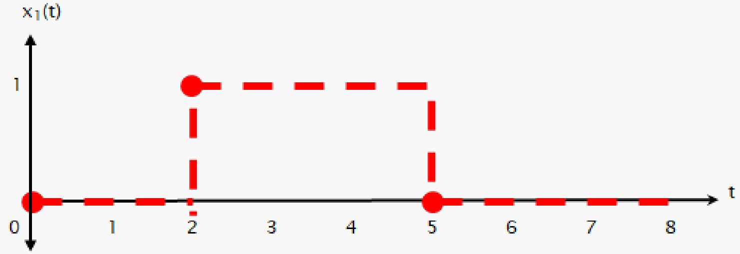
$\gamma(x, I)$ ≔ the set of value expressions of signal $x \in S$ in segment $I \in G_S$

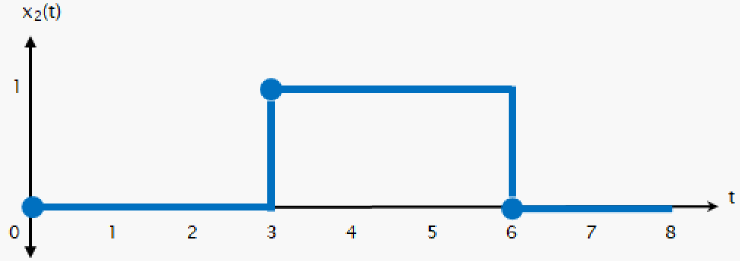
<u>Tabular representation of γ</u>

Defining Tr^+

$\mathrm{Tr}^+(S, \rightsquigarrow)$:= the set of synchronous signals that are "consistent" with γ

t: 0	1	3	4	5	7	8
x_1	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}	{0}
x_2	{0}	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}



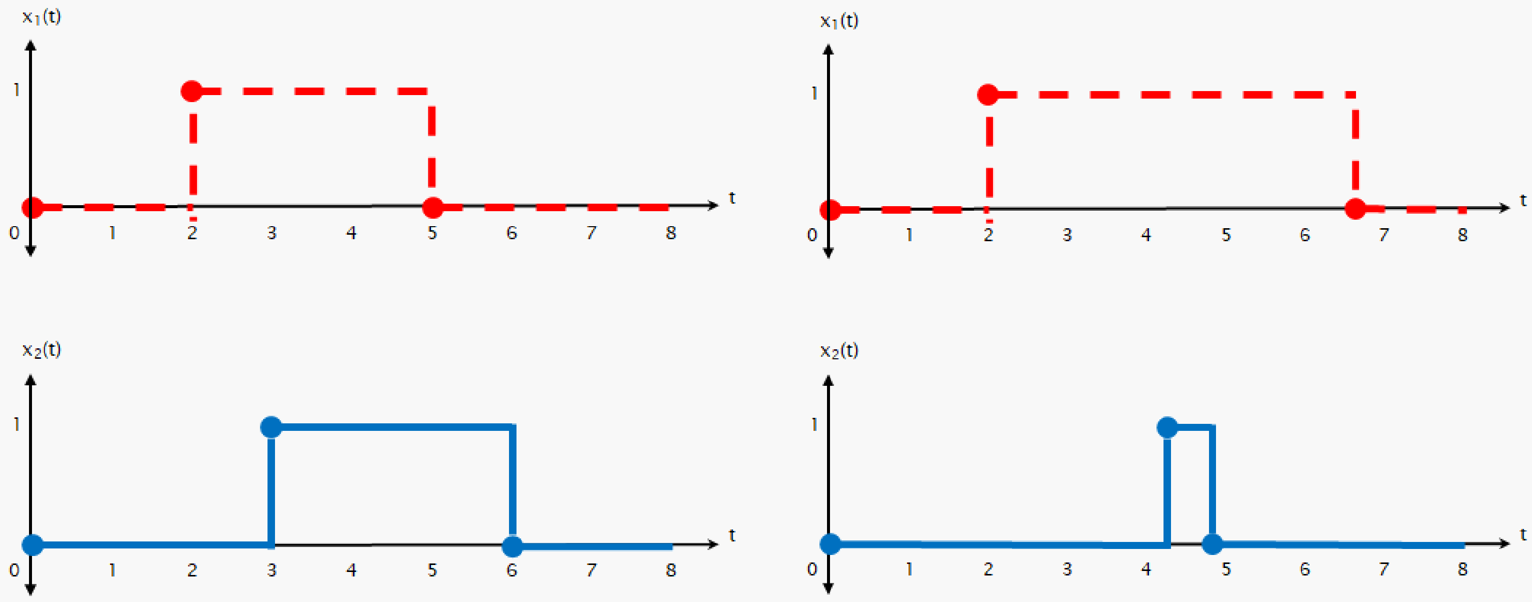


Defining Tr^+

$\mathrm{Tr}^+(S, \rightsquigarrow) :=$ the set of synchronous signals that are "consistent" with γ

t: 0 1 3 4 5 7 8

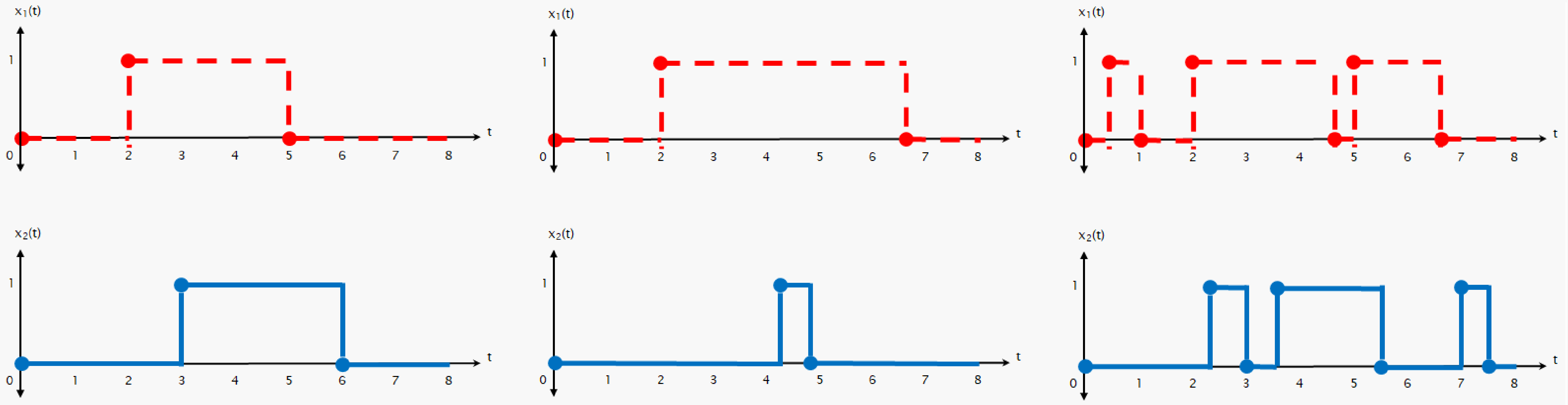
x_1	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}	{0}
x_2	{0}	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}



Defining Tr^+

$\mathrm{Tr}^+(S, \rightsquigarrow) :=$ the set of synchronous signals that are "consistent" with γ

t :	0	1	3	4	5	7 8
x_1	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}	{0}
x_2	{0}	{0, 01}	{0, 01, 1}	{01, 010, 1, 10}	{1, 10, 0}	{10, 0}



Monitoring Algorithm

Idea.

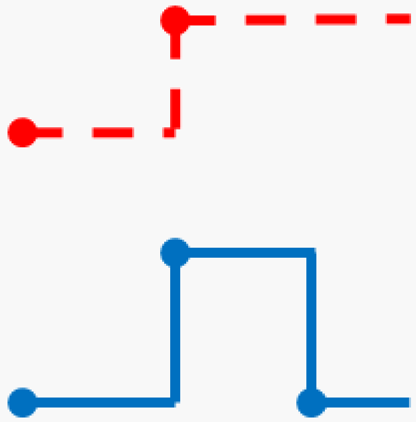
- Compute γ (already done)
- For each subformula ψ and each segment I
  compute the value expressions for satisfaction signal of ψ in I

Asynchronous Product

- $01 \bigotimes 010 =$

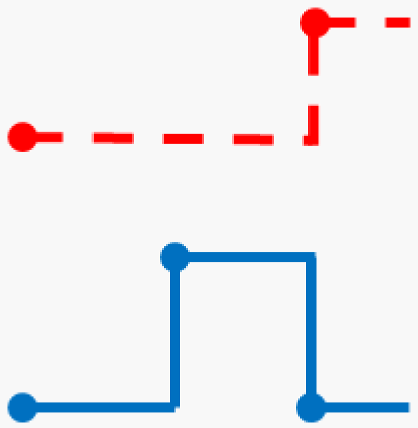
Asynchronous Product

- $01 \otimes 010 = \{(011, 010),$



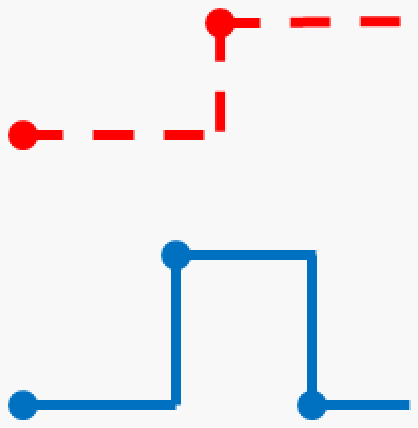
Asynchronous Product

- $01 \otimes 010 = \{(011,010),$
  $(001,010),$



Asynchronous Product

- $01 \otimes 010 = \{(011,010),$
  $(001,010),$
  $(0011,0110),$
  $\ldots\}$



Asynchronous Product

- $$\begin{aligned} 01 \otimes 010 = \{&(011{,}010),\\ &(001{,}010),\\ &(0011{,}0110),\\ &\ldots\} \end{aligned}$$

- $V_1 \otimes V_2 = \{v_1 \otimes v_2 \mid v_1 \in V_1 \text{ and } v_2 \in V_2\}$

Untimed Operations

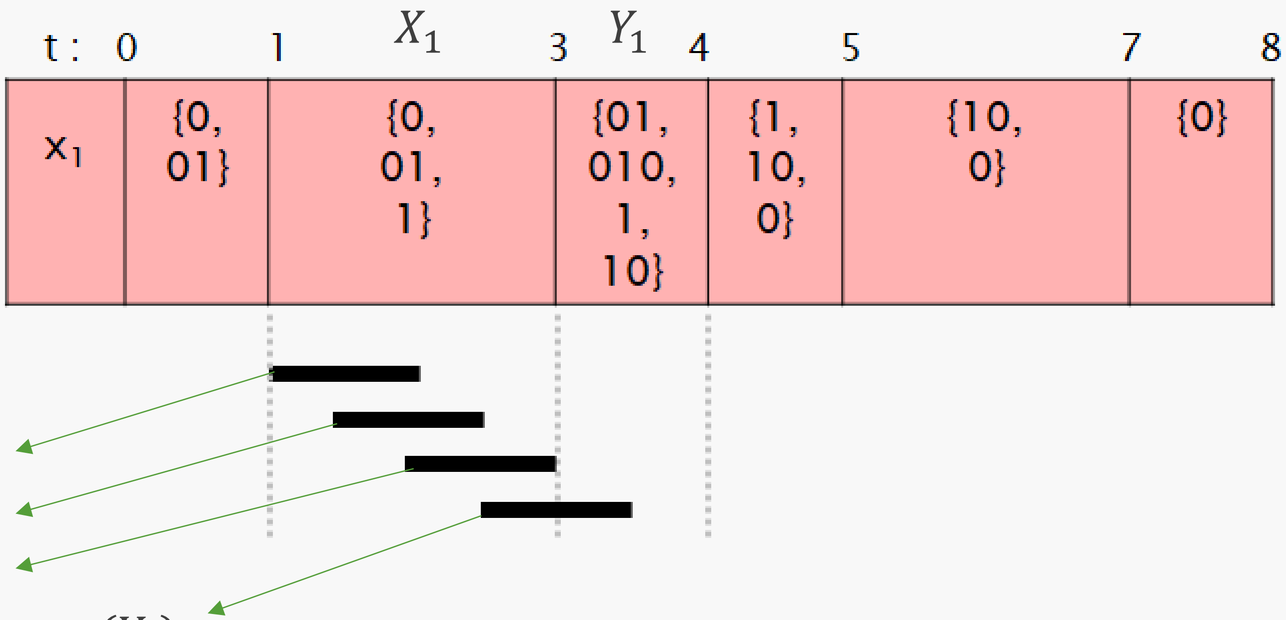
$\mathrm{V}(S, \rightsquigarrow, I, \psi) :=$ the set of value expressions for the satisfaction of ψ by $(S, \rightsquigarrow)$ in segment $I \in G_S$

$$\mathrm{V}(S, \rightsquigarrow, I, p) = \gamma(x_p, I)$$
$$\mathrm{V}(S, \rightsquigarrow, I, \neg\psi) = \{\sim u \mid u \in \mathrm{V}(S, \rightsquigarrow, I, \psi)\}$$
$$\mathrm{V}(S, \rightsquigarrow, I, \psi_1 \wedge \psi_2) = \{u_1 \& u_2 \mid (u_1, u_2) \in \mathrm{V}(S, \rightsquigarrow, I, \psi_1) \otimes \mathrm{V}(S, \rightsquigarrow, I, \psi_2)\}$$
$$\mathrm{V}(S, \rightsquigarrow, I, \psi_1 U \psi_2) = \{u_1 \mathrm{U}^x u_2 \mid (u_1, u_2) \in \mathrm{V}(S, \rightsquigarrow, I, \psi_1) \otimes \mathrm{V}(S, \rightsquigarrow, I, \psi_2)$$
$$x \in \mathrm{firstBit}\big(\mathrm{V}(S, \rightsquigarrow, J, \psi_1 U \psi_2)\big),\ J \text{ is the segment after } I\}$$

U^0: strict until
U^1: weak until

Timed Operations

$$\mathrm{V}\left(S, \rightsquigarrow, [1,3), \mathrm{F}_{[0,1)} x_1\right)$$



$P_1 = \mathrm{pre}(X_1)$
$P_2 = \mathrm{inf}(X_1)$
$P_3 = \mathrm{suf}(X_1)$
$P_4 = \mathrm{suf}(X_1) \cdot \mathrm{pre}(Y_1)$

Timed Operations

$$\mathrm{V}\left(S, \rightsquigarrow, [1,3), \mathrm{F}_{[0,1)} x_1\right)$$

$$= \{1\mathrm{U}^0 p | \, p \in P_1\} \cdot \ldots \cdot \{1\mathrm{U}^0 p \mid p \in P_4\}$$

Correctness

$$[(S, \rightsquigarrow) \vDash \psi]_+ = \top \text{ (resp. } \bot, ?)$$

iff

$$\mathrm{firstBit}\big(\mathrm{V}(S, \rightsquigarrow, \psi)\big) = \{1\} \text{ (resp. } \{0\}, \{0,1\})$$

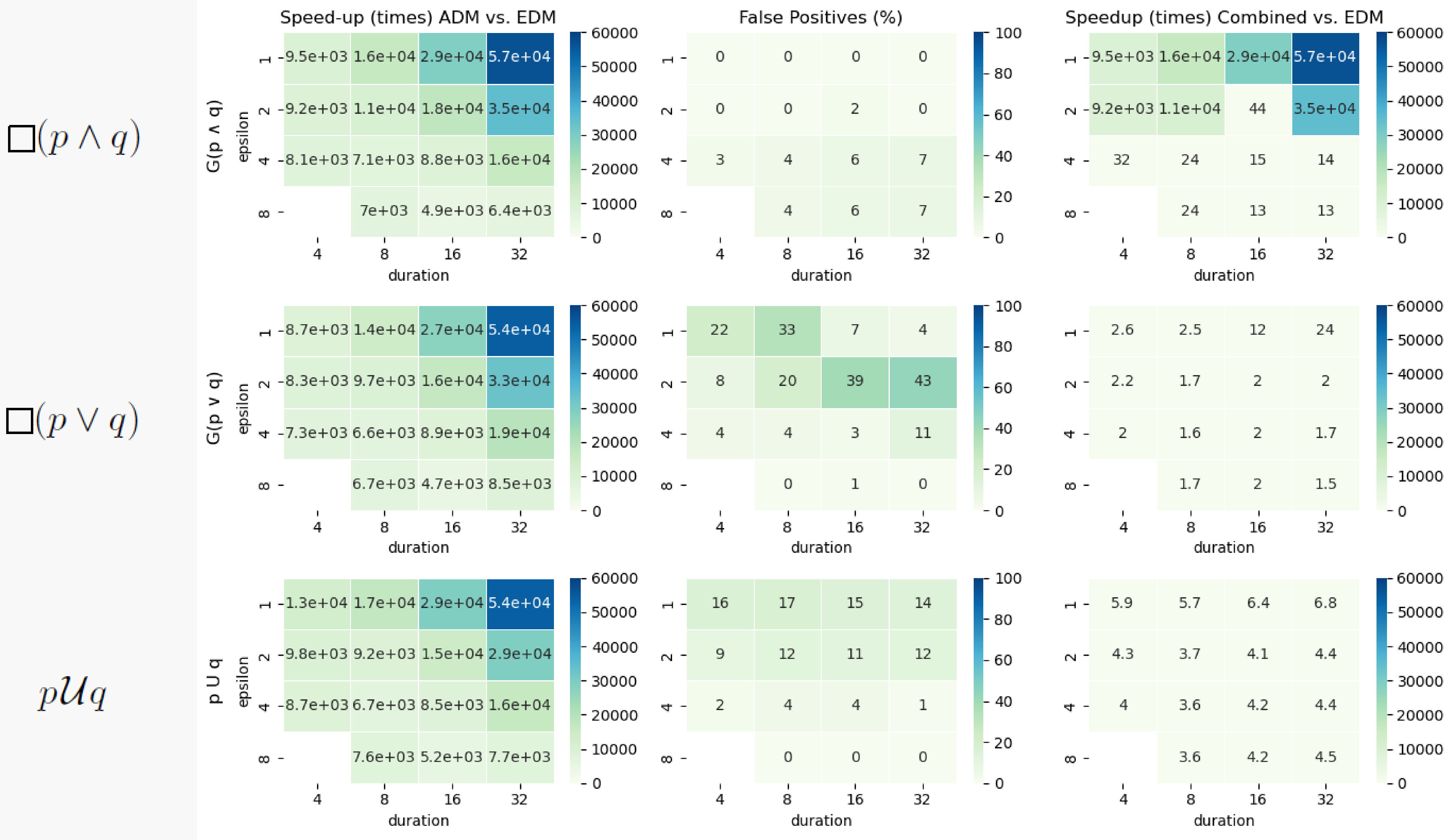
A Few Remarks

- Efficient implementation of $V(S, \rightsquigarrow, I, \psi)$
  - use bitsets + consider only longest value expressions
- Generalization to real-valued signals
  - finite-duration + piecewise-constant + non-Zeno → finitely many values

Experimental Evaluation

- Random generator (boolean)

- Water tanks $\varphi_{\mathrm{WT}} = \square \left(\sum_{i=1}^{n} x_i > c \right)$

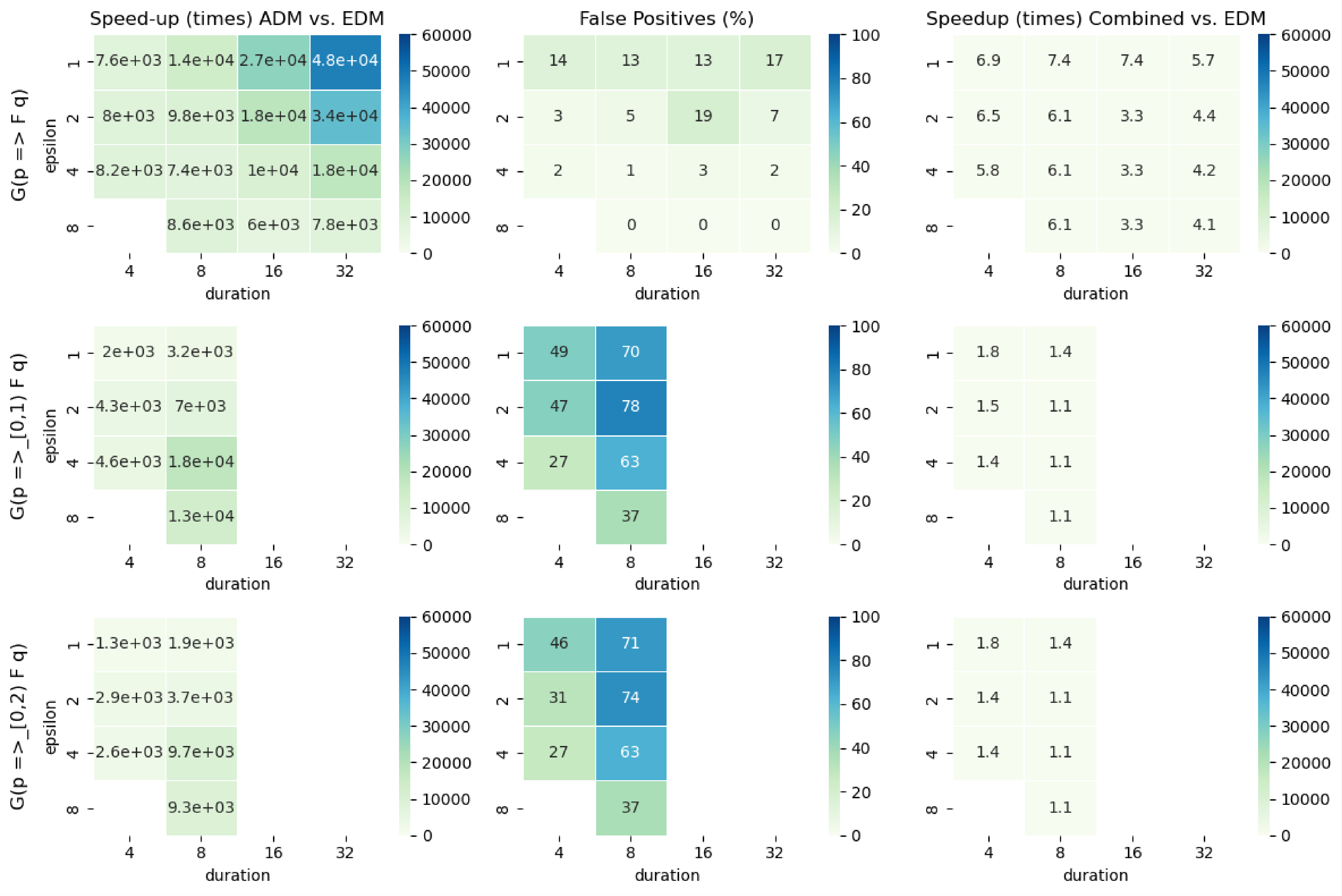


Speed-up (times) ADM vs. EDM
False Positives (%)
Speedup (times) Combined vs. EDM
□(p ∧ q)
□(p ∨ q)
p𝒰q
G(p ∧ q)
G(p v q)
p U q
epsilon
duration

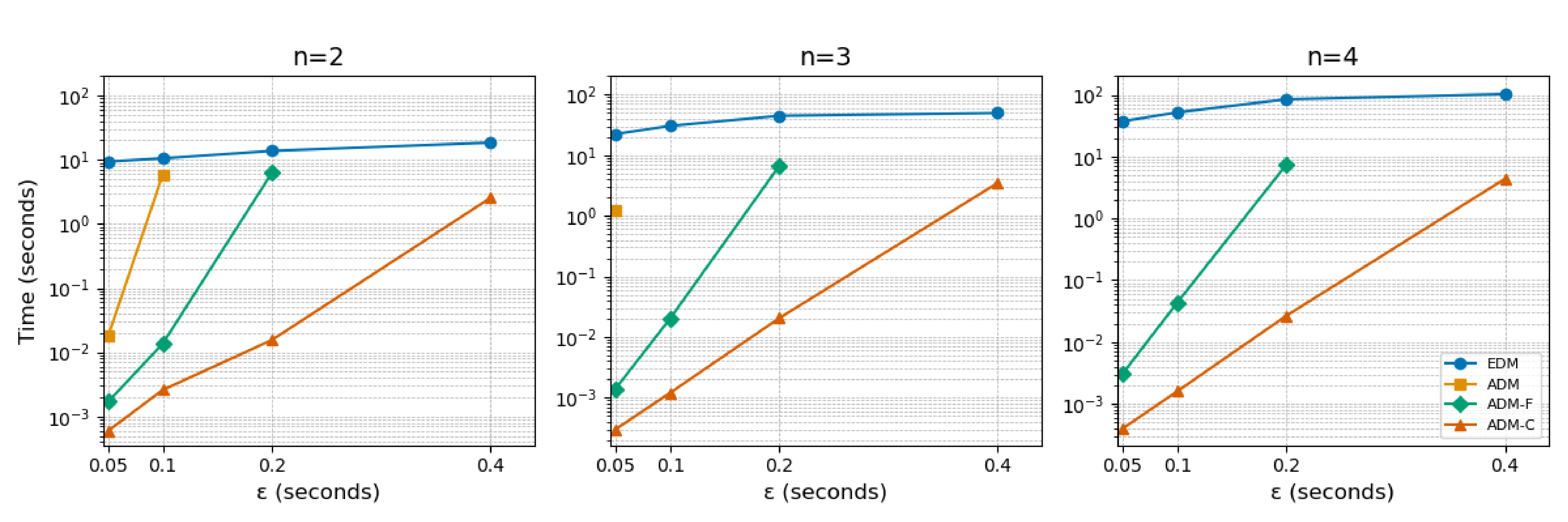
$$\Box(p \Rightarrow \Diamond\, q)$$

$$\Box(p \Rightarrow \Diamond_{[0,1)}\, q)$$

$$\Box(p \Rightarrow \Diamond_{[0,2)}\, q)$$



$$\varphi_{\mathrm{WT}} = \Box\left(\sum_{i=1}^{n} x_i > c\right)$$



Trace length = 1 second

Experimental Evaluation: Summary

Efficiency vs. Accuracy.

- ADM speedup over EDM: up to $\mathrm{x}10^5$
- Tradeoff depends on
  - Type of specification
  - Signal duration
  - Maximum clock skew
- Arithmetic and timed operations → prone to low precision
- Untimed operations → good tradeoff

Approximate (ADM) vs. Exact (EDM) vs. Combined

- Combining ADM and EDM always leads to speedup

Conclusion

Contribution.
A modular, approximate monitoring algorithm for STL specifications on partially synchronous systems, which performs up to $x10^5$ faster.

Future work.

- Symbolic algorithm
- Online monitoring
- Fine-grained control over precision

Thank you!