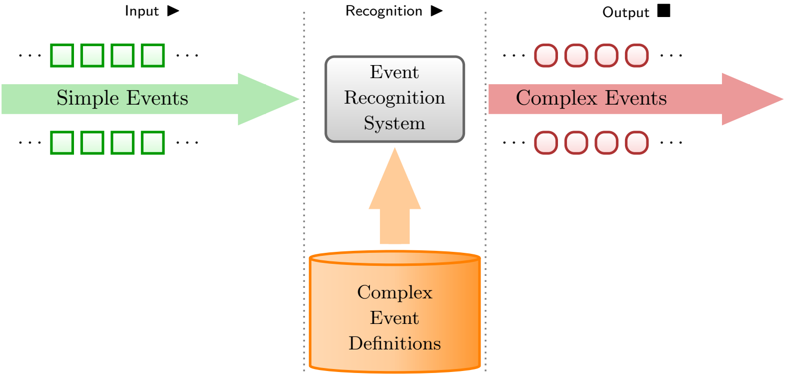
# Scalable Complex Event Recognition

Efthimis Tsilionis

National and Kapodistrian University of Athens, Greece
NCSR Demokritos, Greece

# Complex Event Recognition



# Motivation

Logic-based approaches:

# Motivation

Logic-based approaches:

- Formal, declarative semantics.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

Temporal frameworks:

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

Temporal frameworks:

- Event Calculus: Reasoning about events and their effects.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

Temporal frameworks:

- Event Calculus: Reasoning about events and their effects.
- Event Calculus for Run-Time Reasoning (RTEC): EC with optimization techniques for CER.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

- The *what* (pattern) is to be computed.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

- The *what* (pattern) is to be computed.
- Not necessarily the *how* it is to be computed.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

- The *what* (pattern) is to be computed.
- Not necessarily the *how* it is to be computed.

Motivation: Modify the *what* and *how* to achieve scalability.

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

- The *what* (pattern) is to be computed.
- Not necessarily the *how* it is to be computed.

Motivation: Modify the *what* and *how* to achieve scalability.
Contributions:

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

- The *what* (pattern) is to be computed.
- Not necessarily the *how* it is to be computed.

Motivation: Modify the *what* and *how* to achieve scalability.
Contributions:

- $RTEC_{inc}$: Incremental RTEC (the *what*).

# Motivation

Logic-based approaches:

- Formal, declarative semantics.
- Hierarchical definitions.
- Relational patterns.
- Background Knowledge.
- Efficient reasoning.

The language specifies:

- The *what* (pattern) is to be computed.
- Not necessarily the *how* it is to be computed.

Motivation: Modify the *what* and *how* to achieve scalability.
Contributions:

- $RTEC_{inc}$: Incremental RTEC (the *what*).
- tensor-EC: Tensor-based formalization of EC (the *how*).

# Publications

Journal Publications:

- Tsilionis E., Artikis A., Paliouras G., Incremental Event Calculus for Run-Time Reasoning. In *Journal of Artificial Intelligence Research (JAIR)*, 73, pp. 967—1023, 2022.
- Tsilionis E., Koutroumanis N., Nikitopoulos P., Doulkeridis C. and Artikis A., Online Event Recognition from Moving Vehicles. In *Theory and Practice of Logic Programming (TPLP)*, 19(5-6), pp. 841—856, 2019.

Conference Publications:

- Tsilionis E., Artikis A., Paliouras G., A Tensor-Based Formalization of the Event Calculus. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 3584–3592, 2024.
- Tsilionis E., Artikis A., Paliouras G., Incremental Event Calculus for Run-Time Reasoning (Extended Abstract). In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI–23*, pp. 6974-6978, 2023.
- Tsilionis E., Artikis A. and Paliouras G., Incremental Event Calculus for Run-Time Reasoning. In *13th International Conference on Distributed and Event-Based Systems (DEBS)*, pp. 79–90, 2019.

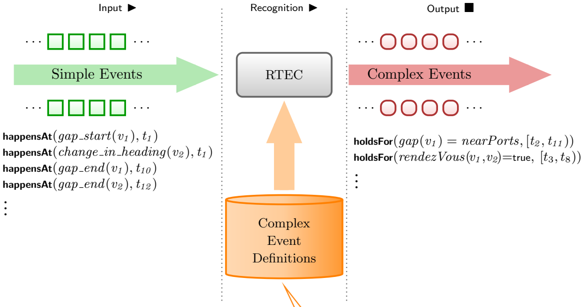
# Background

## Event Calculus

- A logic programming language for representing and reasoning about events and their effects.
- Key components:
  - event (typically instantaneous).
  - fluent: a property that may have different values at different points in time.
- Built-in representation of inertia:
  - $F = V$ holds at a particular time-point if $F = V$ has been *initiated* by an event at some earlier time-point, and not *terminated* by another event in the meantime.

# Background

## Run-Time Event Calculus (RTEC)



$\textsf{happensAt}(\mathit{gap\_start}(v_1), t_1)$
$\textsf{happensAt}(\mathit{change\_in\_heading}(v_2), t_1)$
$\textsf{happensAt}(\mathit{gap\_end}(v_1), t_{10})$
$\textsf{happensAt}(\mathit{gap\_end}(v_2), t_{12})$
$\vdots$

$\textsf{holdsFor}(\mathit{gap}(v_1) = \mathit{nearPorts}, [t_2, t_{11}])$
$\textsf{holdsFor}(\mathit{rendezVous}(v_1, v_2){=}\textsf{true},\ [t_3, t_8])$
$\vdots$

**Simple Fluents**

$$
\begin{aligned}
&\textsf{initiatedAt}(\mathit{gap}(\mathit{Vessel}) = \mathit{nearPorts},\ T) \leftarrow\\
&\quad \textsf{happensAt}(\mathit{gap\_start}(\mathit{Vessel}),\ T),\\
&\quad \textsf{holdsAt}(\mathit{withinArea}(\mathit{Vessel}, \mathit{nearPorts}) = \textsf{true},\ T).
\end{aligned}
$$

$$
\begin{aligned}
&\textsf{terminatedAt}(\mathit{gap}(\mathit{Vessel}) = \_\mathit{Value},\ T) \leftarrow\\
&\quad \textsf{happensAt}(\mathit{gap\_end}(\mathit{Vessel}),\ T).
\end{aligned}
$$

**Statically Determined Fluents**

$$
\begin{aligned}
&\textsf{holdsFor}(\mathit{rendezVous}(\mathit{Vessel}_1, \mathit{Vessel}_2){=}\textsf{true},\ I) \leftarrow\\
&\quad \textsf{holdsFor}(\mathit{proximity}(\mathit{Vessel}_1, \mathit{Vessel}_2){=}\textsf{true},\ I_p),\\
&\quad \textsf{holdsFor}(\mathit{lowSpeed}(\mathit{Vessel}_1){=}\textsf{true},\ I_{l1}),\\
&\quad \textsf{holdsFor}(\mathit{lowSpeed}(\mathit{Vessel}_2){=}\textsf{true},\ I_{l2}),\\
&\quad \textsf{holdsFor}(\mathit{stopped}(\mathit{Vessel}_1){=}\mathit{farFromPorts},\ I_{s1}),\\
&\quad \textsf{holdsFor}(\mathit{stopped}(\mathit{Vessel}_2){=}\mathit{farFromPorts},\ I_{s2}),\\
&\quad \textsf{union\_all}([I_{l1}, I_{s1}],\ I_1),\\
&\quad \textsf{union\_all}([I_{l2}, I_{s2}],\ I_2),\\
&\quad \textsf{intersect\_all}([I_1, I_2, I_p],\ I).
\end{aligned}
$$

# Simple Fluent: Communication Gap Far From Ports

$$
\begin{aligned}
&\textbf{initiatedAt}(gap(Vessel){=}farFromPorts,\ T) \leftarrow\\
&\quad \textbf{happensAt}(gap\_start(Vessel),\ T),\\
&\quad \textbf{not}\ \ \textbf{holdsAt}(withinArea(Vessel, nearPorts){=}\mathsf{true},\ T).
\end{aligned}
$$

# Simple Fluent: Communication Gap Far From Ports

$$
\begin{aligned}
&\textbf{initiatedAt}(gap(Vessel)\!=\!farFromPorts,\ T) \leftarrow\\
&\quad \textbf{happensAt}(gap\_start(Vessel),\ T),\\
&\quad \textbf{not}\ \ \textbf{holdsAt}(withinArea(Vessel, nearPorts)\!=\!\mathsf{true},\ T).\\
&\textbf{terminatedAt}(gap(Vessel)\!=\!farFromPorts,\ T) \leftarrow\\
&\quad \textbf{happensAt}(gap\_end(Vessel),\ T).
\end{aligned}
$$

# Simple Fluent: Communication Gap Far From Ports

**initiatedAt**$(gap(Vessel){=}farFromPorts,\ T) \leftarrow$

  **happensAt**$(gap\_start(Vessel),\ T)$,

  **not holdsAt**$(withinArea(Vessel, nearPorts){=}$true$,\ T)$.

**terminatedAt**$(gap(Vessel){=}farFromPorts,\ T) \leftarrow$

  **happensAt**$(gap\_end(Vessel),\ T)$.

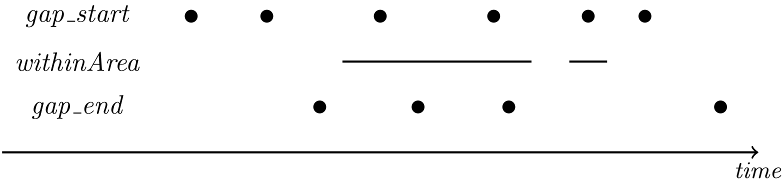
Reasoning: **holdsFor**$(gap(Vessel){=}farFromPorts,\ I)$

# Simple Fluent: Communication Gap Far From Ports

**initiatedAt**$(gap(Vessel){=}farFromPorts,\ T) \leftarrow$
  **happensAt**$(gap\_start(Vessel),\ T)$,
  **not holdsAt**$(withinArea(Vessel, nearPorts){=}\mathsf{true},\ T)$.

**terminatedAt**$(gap(Vessel){=}farFromPorts,\ T) \leftarrow$
  **happensAt**$(gap\_end(Vessel),\ T)$.

Reasoning: **holdsFor**$(gap(Vessel){=}farFromPorts,\ I)$



# Simple Fluent: Communication Gap Far From Ports

**initiatedAt**$(gap(Vessel)=farFromPorts,\ T) \leftarrow$

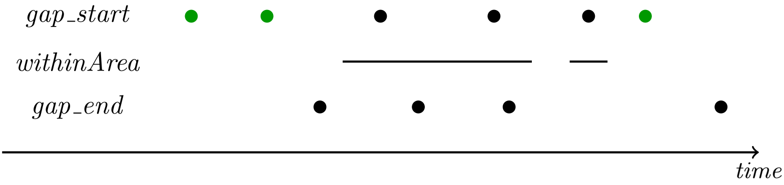
**happensAt**$(gap\_start(Vessel),\ T)$,

**not holdsAt**$(withinArea(Vessel, nearPorts)=$true$,\ T)$.

**terminatedAt**$(gap(Vessel)=farFromPorts,\ T) \leftarrow$

**happensAt**$(gap\_end(Vessel),\ T)$.

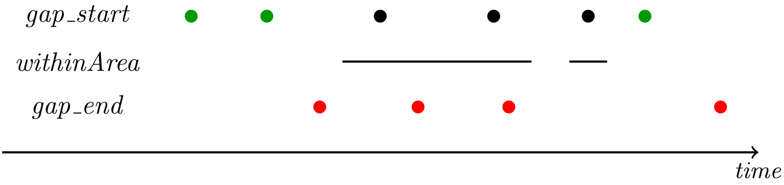
Reasoning: **holdsFor**$(gap(Vessel)=farFromPorts,\ I)$



# Simple Fluent: Communication Gap Far From Ports

$$
\begin{aligned}
&\textbf{initiatedAt}(gap(Vessel){=}farFromPorts,\ T) \leftarrow \\
&\quad \textbf{happensAt}(gap\_start(Vessel),\ T), \\
&\quad \textbf{not}\ \ \textbf{holdsAt}(withinArea(Vessel, nearPorts){=}\mathsf{true},\ T). \\
&\textbf{terminatedAt}(gap(Vessel){=}farFromPorts,\ T) \leftarrow \\
&\quad \textbf{happensAt}(gap\_end(Vessel),\ T).
\end{aligned}
$$

Reasoning: $\textbf{holdsFor}(gap(Vessel){=}farFromPorts,\ I)$



# Simple Fluent: Communication Gap Far From Ports

**initiatedAt**$(gap(Vessel){=}farFromPorts,\ T) \leftarrow$

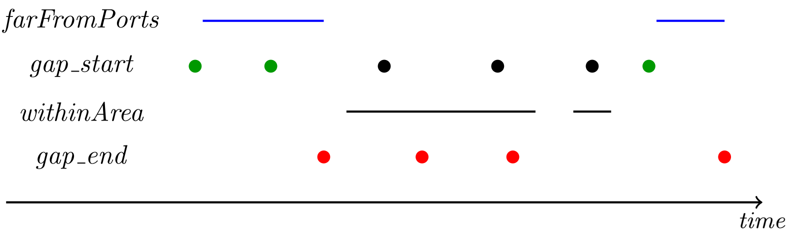
  **happensAt**$(gap\_start(Vessel),\ T),$

  **not holdsAt**$(withinArea(Vessel, nearPorts){=}$true$,\ T).$

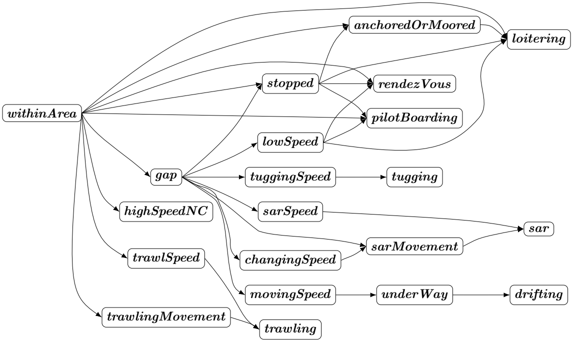
**terminatedAt**$(gap(Vessel){=}farFromPorts,\ T) \leftarrow$

  **happensAt**$(gap\_end(Vessel),\ T).$

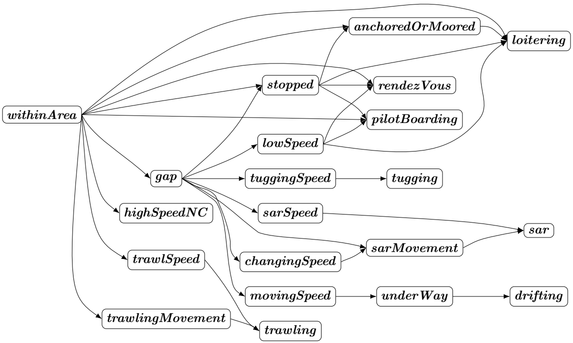
Reasoning: **holdsFor**$(gap(Vessel){=}farFromPorts,\ I)$



# Hierarchical Event Descriptions



# Hierarchical Event Descriptions

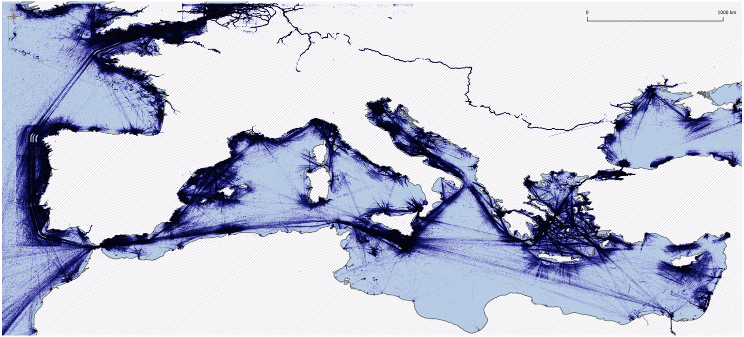


## Semantics

An event description is a locally stratified logic program, i.e., it has a **unique perfect** model.

# Part 1: Modify the *what*

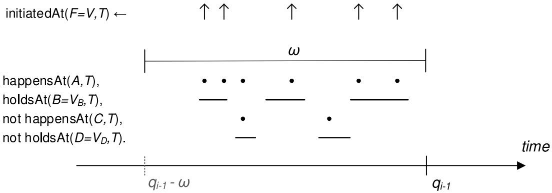
## Incremental CER



- Delayed events (e.g., satellite GPS messages)
- Overlapping temporal windows

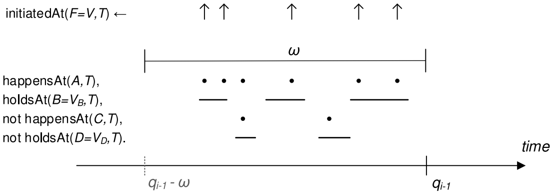
# Problem Statement

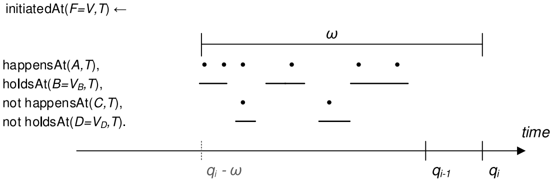
## RTEC



# Problem Statement

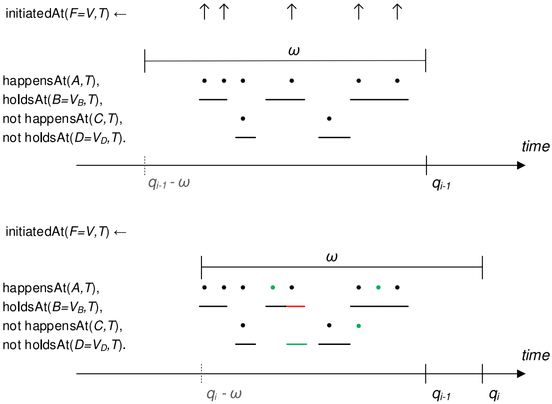
## RTEC





# Problem Statement

## RTEC



initiatedAt($F$=$V$,$T$) ←

$\omega$

happensAt($A$,$T$),

holdsAt($B$=$V_B$,$T$),

not happensAt($C$,$T$),

not holdsAt($D$=$V_D$,$T$).

*time*

$q_{i\text{-}1}$ - $\omega$

$q_{i\text{-}1}$

initiatedAt($F$=$V$,$T$) ←

$\omega$

happensAt($A$,$T$),

holdsAt($B$=$V_B$,$T$),

not happensAt($C$,$T$),

not holdsAt($D$=$V_D$,$T$).

*time*

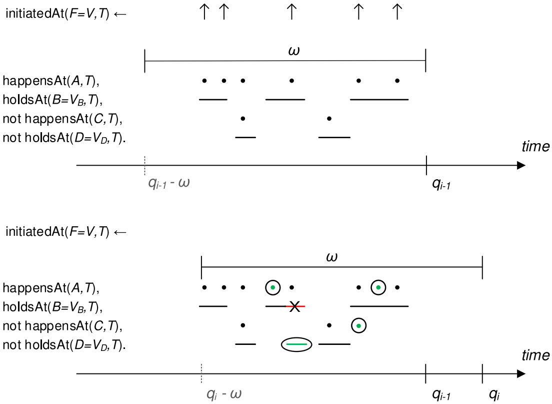
$q_i$ - $\omega$

$q_{i\text{-}1}$

$q_i$

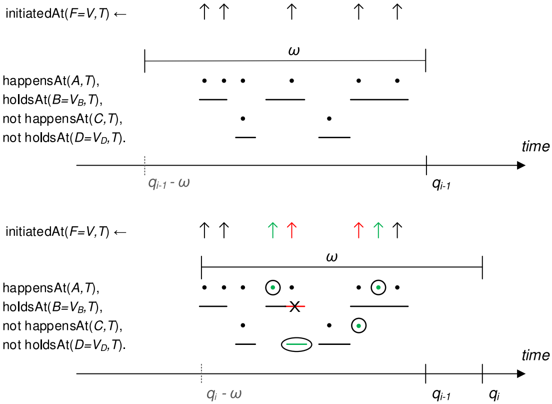
# Problem Statement

## RTEC


initiatedAt(*F*=*V*,*T*) ←
ω
happensAt(*A*,*T*),
holdsAt(*B*=*V*$_B$,*T*),
not happensAt(*C*,*T*),
not holdsAt(*D*=*V*$_D$,*T*).
time
$q_{i-1}$ - ω
$q_{i-1}$
initiatedAt(*F*=*V*,*T*) ←
ω
happensAt(*A*,*T*),
holdsAt(*B*=*V*$_B$,*T*),
not happensAt(*C*,*T*),
not holdsAt(*D*=*V*$_D$,*T*).
time
$q_i$ - ω
$q_{i-1}$
$q_i$

# Problem Statement

## RTEC

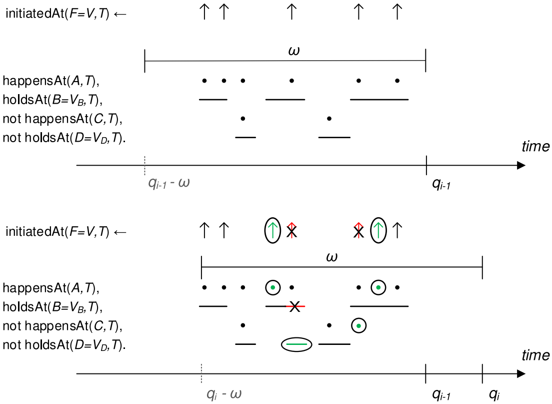


initiatedAt($F$=$V$,$T$) ←
happensAt($A$,$T$),
holdsAt($B$=$V_B$,$T$),
not happensAt($C$,$T$),
not holdsAt($D$=$V_D$,$T$).

initiatedAt($F$=$V$,$T$) ←
happensAt($A$,$T$),
holdsAt($B$=$V_B$,$T$),
not happensAt($C$,$T$),
not holdsAt($D$=$V_D$,$T$).

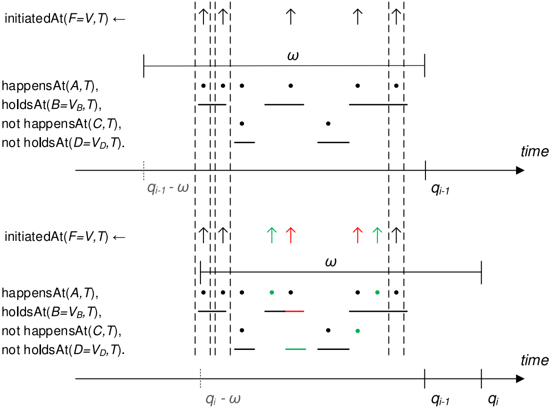
# Problem Statement

## RTEC


initiatedAt(*F*=*V*,*T*) ←
ω
happensAt(*A*,*T*),
holdsAt(*B*=*V*<sub>B</sub>,*T*),
not happensAt(*C*,*T*),
not holdsAt(*D*=*V*<sub>D</sub>,*T*).
time
$q_{i-1}$ - ω
$q_{i-1}$
initiatedAt(*F*=*V*,*T*) ←
ω
happensAt(*A*,*T*),
holdsAt(*B*=*V*<sub>B</sub>,*T*),
not happensAt(*C*,*T*),
not holdsAt(*D*=*V*<sub>D</sub>,*T*).
time
$q_i$ - ω
$q_{i-1}$
$q_i$

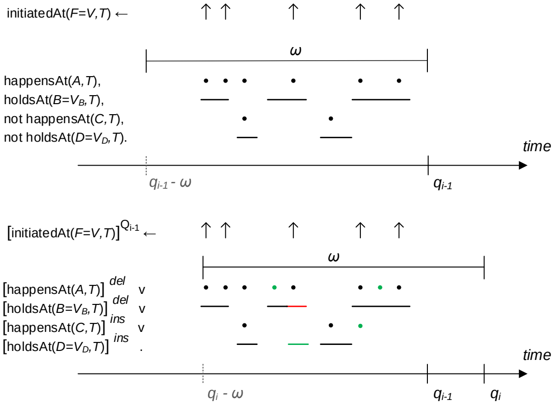
# Problem Statement

## RTEC



# $RTEC_{inc}$: Deletion Phase

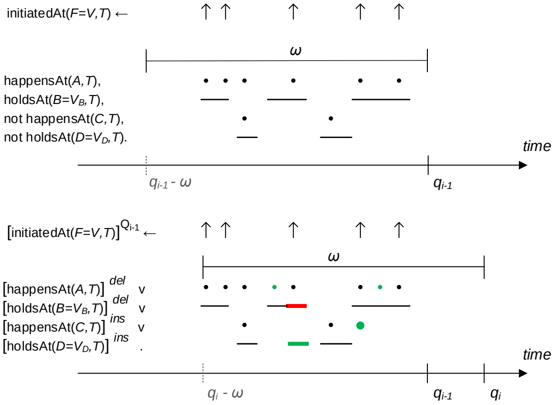
# $RTEC_{inc}$: Deletion Phase



initiatedAt($F$=$V$,$T$) ←
happensAt($A$,$T$),
holdsAt($B$=$V_B$,$T$),
not happensAt($C$,$T$),
not holdsAt($D$=$V_D$,$T$).

$[\text{initiatedAt}(F{=}V,T)]^{Q_{i-1}}$ ←
$[\text{happensAt}(A,T)]^{del}$ v
$[\text{holdsAt}(B{=}V_B,T)]^{del}$ v
$[\text{happensAt}(C,T)]^{ins}$ v
$[\text{holdsAt}(D{=}V_D,T)]^{ins}$ .

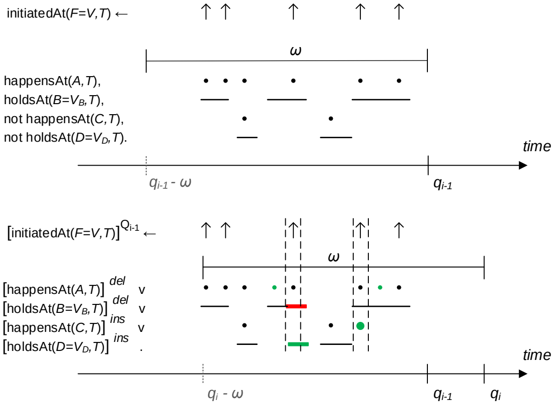
# $RTEC_{inc}$: Deletion Phase



initiatedAt($F$=$V$,$T$) ←
happensAt($A$,$T$),
holdsAt($B$=$V_B$,$T$),
not happensAt($C$,$T$),
not holdsAt($D$=$V_D$,$T$).

[initiatedAt($F$=$V$,$T$)]$^{Q_{i-1}}$ ←
[happensAt($A$,$T$)]$^{del}$ v
[holdsAt($B$=$V_B$,$T$)]$^{del}$ v
[happensAt($C$,$T$)]$^{ins}$ v
[holdsAt($D$=$V_D$,$T$)]$^{ins}$ .

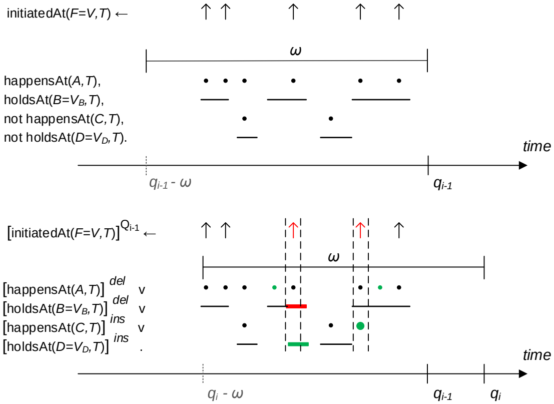
# $RTEC_{inc}$: Deletion Phase



initiatedAt($F$=$V$,$T$) ←
happensAt($A$,$T$),
holdsAt($B$=$V_B$,$T$),
not happensAt($C$,$T$),
not holdsAt($D$=$V_D$,$T$).

$[\text{initiatedAt}(F{=}V,T)]^{Q_{i-1}}$ ←
$[\text{happensAt}(A,T)]^{del}$ v
$[\text{holdsAt}(B{=}V_B,T)]^{del}$ v
$[\text{happensAt}(C,T)]^{ins}$ v
$[\text{holdsAt}(D{=}V_D,T)]^{ins}$ .

# $RTEC_{inc}$: Deletion Phase



initiatedAt($F$=$V$,$T$) ←
happensAt($A$,$T$),
holdsAt($B$=$V_B$,$T$),
not happensAt($C$,$T$),
not holdsAt($D$=$V_D$,$T$).

$[\text{initiatedAt}(F{=}V,T)]^{Q_{i-1}}$ ←
$[\text{happensAt}(A,T)]^{del}$ v
$[\text{holdsAt}(B{=}V_B,T)]^{del}$ v
$[\text{happensAt}(C,T)]^{ins}$ v
$[\text{holdsAt}(D{=}V_D,T)]^{ins}$ .

# $RTEC_{inc}$: Addition Phase

$$\begin{aligned}
&\mathsf{initiatedAt}(F{=}V,\ T) \leftarrow \\
&\quad \Big[\mathsf{happensAt}(A,\ T)\Big]^{ins}, \\
&\quad \Big[\mathsf{holdsAt}(B{=}V_B,\ T)\Big]^{\mathsf{Q_i}}, \\
&\quad \mathsf{not}\ \Big[\mathsf{happensAt}(C,\ T)\Big]^{\mathsf{Q_i}}, \\
&\quad \mathsf{not}\ \Big[\mathsf{holdsAt}(D{=}V_D,\ T)\Big]^{\mathsf{Q_i}}.
\end{aligned} \qquad \text{(a)}$$

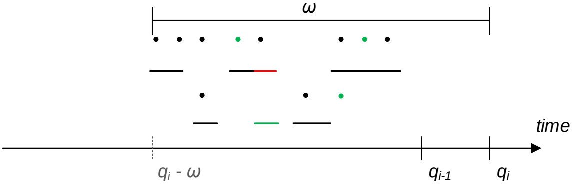
$$\begin{aligned}
&\mathsf{initiatedAt}(F{=}V,\ T) \leftarrow \\
&\quad \Big[\mathsf{happensAt}(A,\ T)\Big]^{\mathsf{Q_i}\setminus ins}, \\
&\quad \Big[\mathsf{holdsAt}(B{=}V_B,\ T)\Big]^{ins}, \\
&\quad \mathsf{not}\ \Big[\mathsf{happensAt}(C,\ T)\Big]^{\mathsf{Q_i}}, \\
&\quad \mathsf{not}\ \Big[\mathsf{holdsAt}(D{=}V_D,\ T)\Big]^{\mathsf{Q_i}}.
\end{aligned} \qquad \text{(b)}$$

$$\begin{aligned}
&\mathsf{initiatedAt}(F{=}V,\ T) \leftarrow \\
&\quad \Big[\mathsf{happensAt}(C,\ T)\Big]^{del}, \\
&\quad \Big[\mathsf{happensAt}(A,\ T)\Big]^{\mathsf{Q_i}\setminus ins}, \\
&\quad \Big[\mathsf{holdsAt}(B{=}V_B,\ T)\Big]^{\mathsf{Q_i}\setminus ins}, \\
&\quad \mathsf{not}\ \Big[\mathsf{holdsAt}(D{=}V_D,\ T)\Big]^{\mathsf{Q_i}}.
\end{aligned} \qquad \text{(c)}$$

$$\begin{aligned}
&\mathsf{initiatedAt}(F{=}V,\ T) \leftarrow \\
&\quad \Big[\mathsf{happensAt}(A,\ T)\Big]^{\mathsf{Q_i}\setminus ins}, \\
&\quad \Big[\mathsf{holdsAt}(D{=}V_D,\ T)\Big]^{del}, \\
&\quad \Big[\mathsf{holdsAt}(B{=}V_B,\ T)\Big]^{\mathsf{Q_i}\setminus ins}, \\
&\quad \mathsf{not}\ \Big[\mathsf{happensAt}(C,\ T)\Big]^{\mathsf{Q_i}\cup del}.
\end{aligned} \qquad \text{(d)}$$

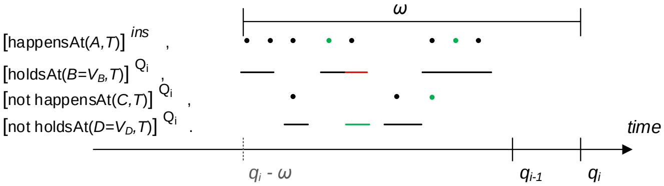
# $RTEC_{inc}$: Addition Phase

## *Delta* rule (a)



# $RTEC_{inc}$: Addition Phase

## *Delta* rule (a)



$[\text{happensAt}(A,T)]^{ins}$ ,

$[\text{holdsAt}(B=V_B,T)]^{Q_i}$ ,

$[\text{not happensAt}(C,T)]^{Q_i}$ ,

$[\text{not holdsAt}(D=V_D,T)]^{Q_i}$ .

# $RTEC_{inc}$: Addition Phase

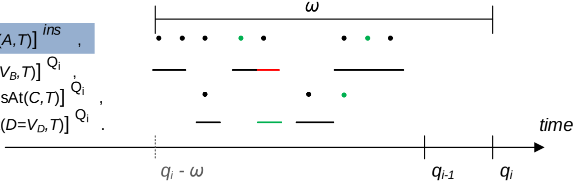
## *Delta* rule (a)

$[\text{happensAt}(A,T)]^{ins}$ ,

$[\text{holdsAt}(B{=}V_B,T)]^{Q_i}$ ,

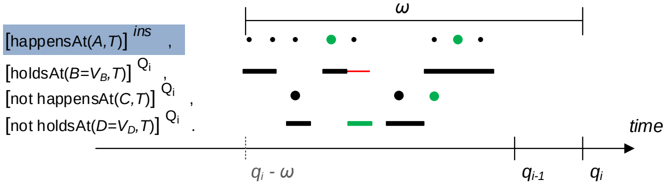
$[\text{not happensAt}(C,T)]^{Q_i}$ ,

$[\text{not holdsAt}(D{=}V_D,T)]^{Q_i}$ .



# $RTEC_{inc}$: Addition Phase

## *Delta* rule (a)



$[\text{happensAt}(A,T)]^{ins}$ ,

$[\text{holdsAt}(B{=}V_B,T)]^{Q_i}$ ,

$[\text{not happensAt}(C,T)]^{Q_i}$ ,

$[\text{not holdsAt}(D{=}V_D,T)]^{Q_i}$ .

# $RTEC_{inc}$: Addition Phase

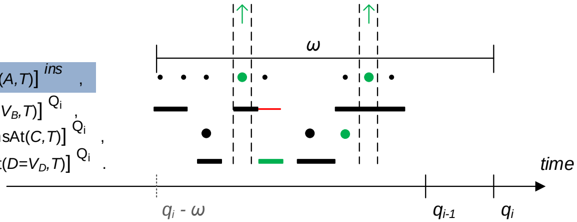
## *Delta* rule (a)

$[\text{happensAt}(A,T)]^{ins}$ ,

$[\text{holdsAt}(B{=}V_B,T)]^{Q_i}$ ,

$[\text{not happensAt}(C,T)]^{Q_i}$ ,

$[\text{not holdsAt}(D{=}V_D,T)]^{Q_i}$ .



# $RTEC_{inc}$: Formal properties

**Correctness**

$RTEC_{inc}$ computes exactly the same intervals of the fluents of an event description as RTEC, and no other interval.

# $RTEC_{inc}$: Formal properties

### Correctness

$RTEC_{inc}$ computes exactly the same intervals of the fluents of an event description as RTEC, and no other interval.

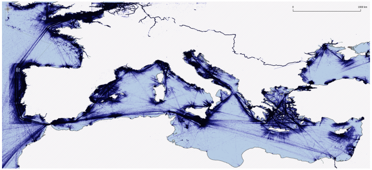
### Complexity

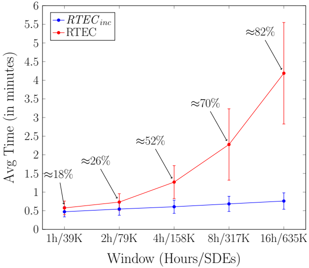
The most important factor for performance improvement is the ratio of delayed insertions/retractions to the degree of overlap:

$$\frac{n \times e}{m_{\mathsf{ov}}} < 1$$

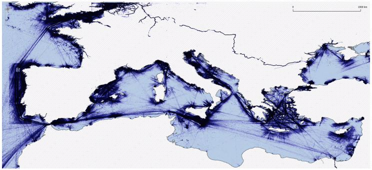
# $RTEC_{inc}$: Empirical Evaluation

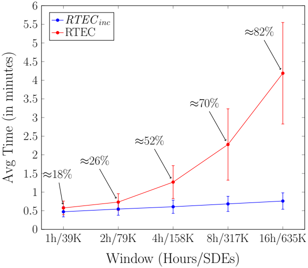
# $RTEC_{inc}$: Empirical Evaluation

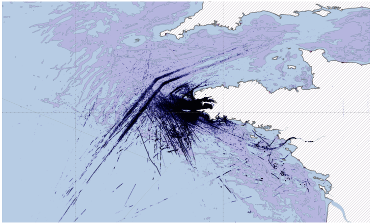


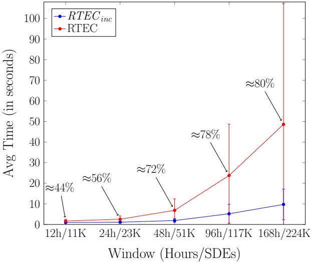


# $RTEC_{inc}$: Empirical Evaluation









# Part 2: Modify the *how*

## Logical Inference in Tensor Spaces

# Part 2: Modify the *how*

## Logical Inference in Tensor Spaces

- Numerical computation has potential to cope with Web scale data.

# Part 2: Modify the *how*

## Logical Inference in Tensor Spaces

- Numerical computation has potential to cope with Web scale data.
- Logical reasoning through algebraic operations is a step towards neuro-symbolic integration.

# Part 2: Modify the *how*

## Logical Inference in Tensor Spaces

- Numerical computation has potential to cope with Web scale data.
- Logical reasoning through algebraic operations is a step towards neuro-symbolic integration.
- Use of efficient (parallel) algorithms and great computing resources (GPUs).

# Tensor-EC: Encoding

# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

$\mathsf{happensAt}(e(X, Y),\ T)$

# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

happensAt$(e(X, Y),\ T)$

# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

$$\text{happensAt}(e(X, Y),\ T)$$

# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

$\mathsf{happensAt}(e(X, Y), T)$

# Tensor-EC: Encoding

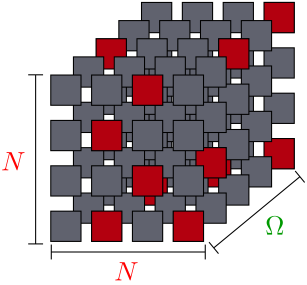
- Domain entities: $N$
- Window time-points: $\Omega$

$$\underbrace{\boxed{\mathsf{happensAt}(e(X, Y),\ T)}}_{r}$$

# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

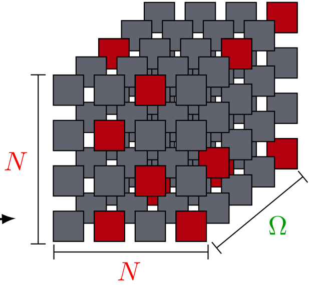
$$\underbrace{\mathsf{happensAt}(e(X, Y),\ T)}_{r}$$



# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

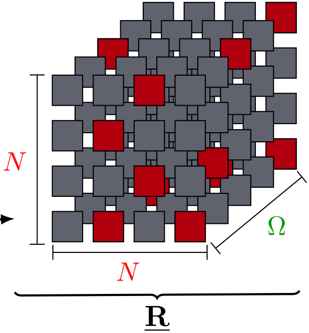
$$\underbrace{\text{happensAt}(e(X, Y),\ T)}_{r} \longrightarrow$$



# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

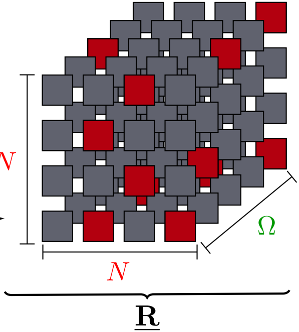
$$\underbrace{\text{happensAt}(e(X, Y),\ T)}_{r} \longrightarrow$$



# Tensor-EC: Encoding

- Domain entities: $N$
- Window time-points: $\Omega$

$$\underbrace{\text{happensAt}(e(X,Y),\ T)}_{r} \longrightarrow$$



$$\underline{\mathbf{R}}_{\,i,j,k} = \begin{cases} 1, & \text{if } \mathbf{M}_P \models r, \ \text{ for } c_i, c_j, t_k \quad 🟥 \\ 0, & \text{o.w} \quad ⬛ \end{cases}$$

$$\forall\, 1 \leq i,j \leq N, 1 \leq k \leq \Omega \ .$$

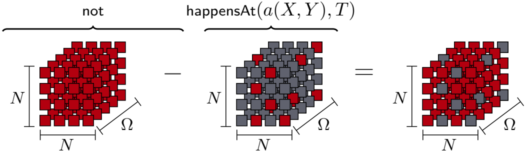
# Tensor-EC: Reasoning

# Tensor-EC: Reasoning

**Negation**:

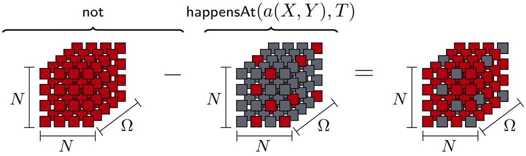
# Tensor-EC: Reasoning

**Negation**:



# Tensor-EC: Reasoning

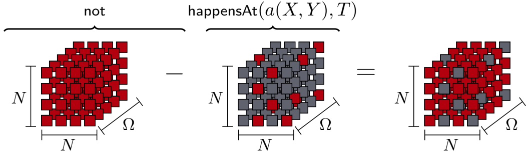
**Negation**:



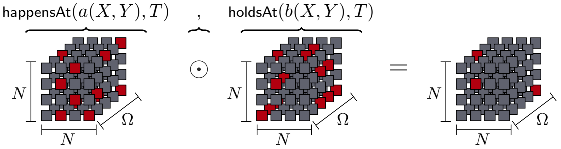
**Conjunction**:

# Tensor-EC: Reasoning

**Negation**:

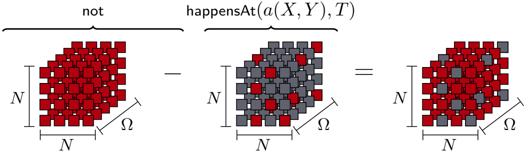


**Conjunction**:

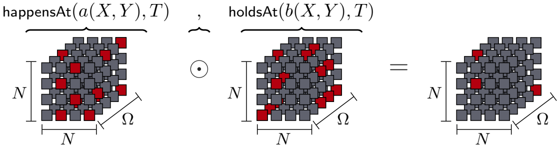


# Tensor-EC: Reasoning

**Negation**:



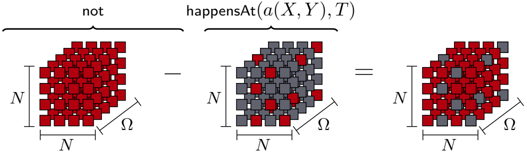
**Conjunction**:



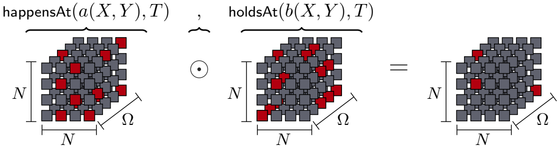
**Disjunction**:

# Tensor-EC: Reasoning

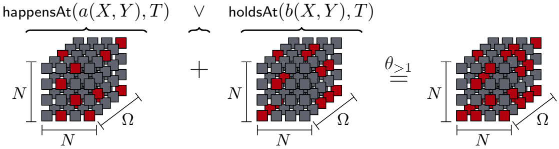
**Negation**:



**Conjunction**:



**Disjunction**:



# Tensor-EC: Reasoning

## Rule Evaluation

$$
\begin{aligned}
&\mathsf{initiatedAt}\big(\mathit{fl}(X,Y)=v,\ T\big)\ \leftarrow\\
&\quad \mathsf{happensAt}\big(e(X,Y),\ T\big)\ \ ,\\
&\quad \mathsf{holdsAt}\big(d(X,Y)=v_d,\ T\big)\ \ .
\end{aligned}
$$

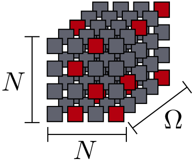
# Tensor-EC: Reasoning

## Rule Evaluation

$$\text{initiatedAt}(\mathit{fl}(X, Y){=}v,\ T) \leftarrow$$

$$\text{happensAt}(e(X, Y),\ T)\ ,$$

$$\text{holdsAt}(d(X, Y){=}v_d,\ T)\ .$$



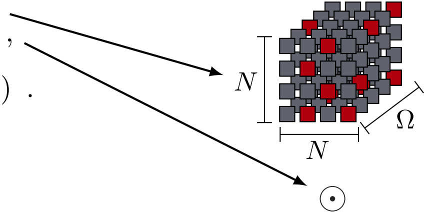
# Tensor-EC: Reasoning

## Rule Evaluation

initiatedAt$(\mathit{fl}(X,Y){=}v,\ T)$ $\leftarrow$

happensAt$(e(X,Y),\ T)$ ,

holdsAt$(d(X,Y){=}v_d,\ T)$ .



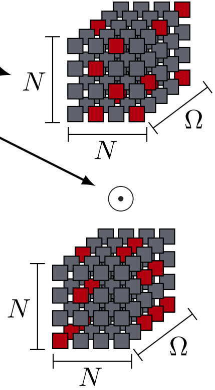
# Tensor-EC: Reasoning

## Rule Evaluation

initiatedAt$(\mathit{fl}(X,Y){=}v,\ T)\ \leftarrow$

happensAt$(e(X,Y),\ T)$ ,

holdsAt$(d(X,Y){=}v_d,\ T)$ .



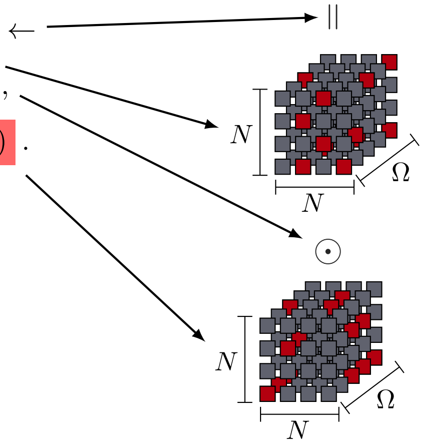
# Tensor-EC: Reasoning

## Rule Evaluation

initiatedAt$(\mathit{fl}(X,Y){=}v,\ T)$ ←

happensAt$(e(X,Y),\ T)$ ,

holdsAt$(d(X,Y){=}v_d,\ T)$ .



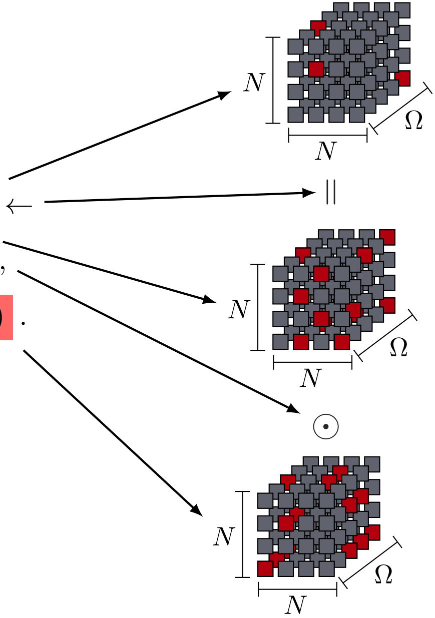
# Tensor-EC: Reasoning

## Rule Evaluation

initiatedAt$(\mathit{fl}(X,Y){=}v,\ T)$ $\leftarrow$

happensAt$(e(X,Y),\ T)$ ,

holdsAt$(d(X,Y){=}v_d,\ T)$ .



# Tensor-EC: Computing a Model

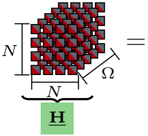
$$
\begin{array}{l}
\mathsf{holdsAt}(\mathit{fl}(X,Y){=}v,\ T) \leftarrow \\
\quad \mathsf{initiatedAt}(\mathit{fl}(X,Y){=}v,\ T_{prev}), \\
\quad \mathsf{not\ \ terminatedAt}(\mathit{fl}(X,Y){=}v,\ T_{prev}), \\
\quad \mathit{next}(T_{prev}, T).
\end{array}
$$

$$
\begin{array}{l}
\mathsf{holdsAt}(\mathit{fl}(X,Y){=}v,\ T) \leftarrow \\
\quad \mathsf{holdsAt}(\mathit{fl}(X,Y){=}v,\ T_{prev}), \\
\quad \mathsf{not\ \ terminatedAt}(\mathit{fl}(X,Y){=}v,\ T_{prev}), \\
\quad \mathit{next}(T_{prev}, T).
\end{array}
$$

# Tensor-EC: Computing a Model

holdsAt($\mathit{fl}(X,Y){=}v,\ T$) $\leftarrow$
initiatedAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
not terminatedAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
$\mathit{next}(T_{prev},\ T)$.

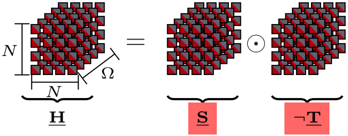
holdsAt($\mathit{fl}(X,Y){=}v,\ T$) $\leftarrow$
holdsAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
not terminatedAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
$\mathit{next}(T_{prev},\ T)$.



# Tensor-EC: Computing a Model

$$\begin{aligned}&\mathsf{holdsAt}(\mathit{fl}(X,Y)=v,\ T) \leftarrow\\ &\quad \mathsf{initiatedAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),\\ &\quad \mathsf{not}\ \ \mathsf{terminatedAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),\\ &\quad \mathit{next}(T_{prev},\ T).\end{aligned}$$

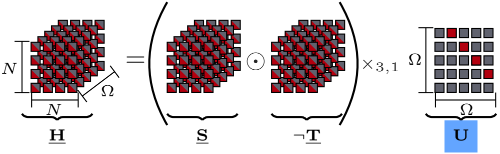
$$\begin{aligned}&\mathsf{holdsAt}(\mathit{fl}(X,Y)=v,\ T) \leftarrow\\ &\quad \mathsf{holdsAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),\\ &\quad \mathsf{not}\ \ \mathsf{terminatedAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),\\ &\quad \mathit{next}(T_{prev},\ T).\end{aligned}$$



# Tensor-EC: Computing a Model

$\mathsf{holdsAt}(\mathit{fl}(X,Y)=v,\ T) \leftarrow$
$\quad \mathsf{initiatedAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),$
$\quad \mathsf{not\ \ terminatedAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),$
$\quad \mathit{next}(T_{prev},\ T).$

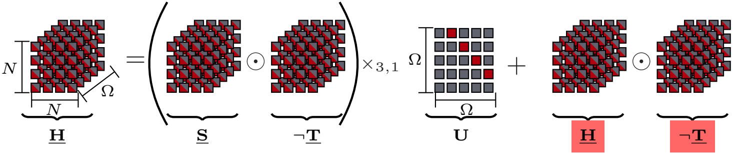
$\mathsf{holdsAt}(\mathit{fl}(X,Y)=v,\ T) \leftarrow$
$\quad \mathsf{holdsAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),$
$\quad \mathsf{not\ \ terminatedAt}(\mathit{fl}(X,Y)=v,\ T_{prev}),$
$\quad \mathit{next}(T_{prev},\ T).$



# Tensor-EC: Computing a Model

holdsAt($fl(X, Y){=}v,\ T$) ←<br>
  initiatedAt($fl(X, Y){=}v,\ T_{prev}$),<br>
  not terminatedAt($fl(X, Y){=}v,\ T_{prev}$),<br>
  $next(T_{prev}, T)$.

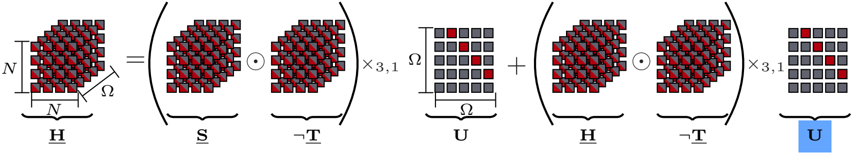
holdsAt($fl(X, Y){=}v,\ T$) ←<br>
  holdsAt($fl(X, Y){=}v,\ T_{prev}$),<br>
  not terminatedAt($fl(X, Y){=}v,\ T_{prev}$),<br>
  $next(T_{prev}, T)$.



# Tensor-EC: Computing a Model

holdsAt($\mathit{fl}(X, Y){=}v,\ T$) ←
initiatedAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
not terminatedAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
$\mathit{next}(T_{prev},\ T)$.

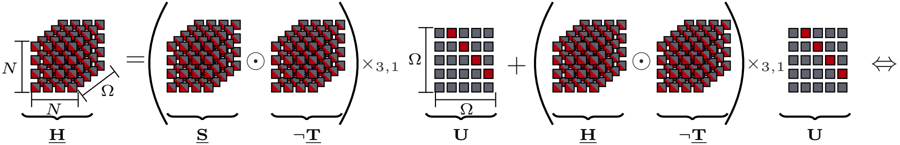
holdsAt($\mathit{fl}(X, Y){=}v,\ T$) ←
holdsAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
not terminatedAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
$\mathit{next}(T_{prev},\ T)$.



# Tensor-EC: Computing a Model

holdsAt($\mathit{fl}(X, Y){=}v,\ T$) $\leftarrow$
  initiatedAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
  not terminatedAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
  $\mathit{next}(T_{prev}, T)$.

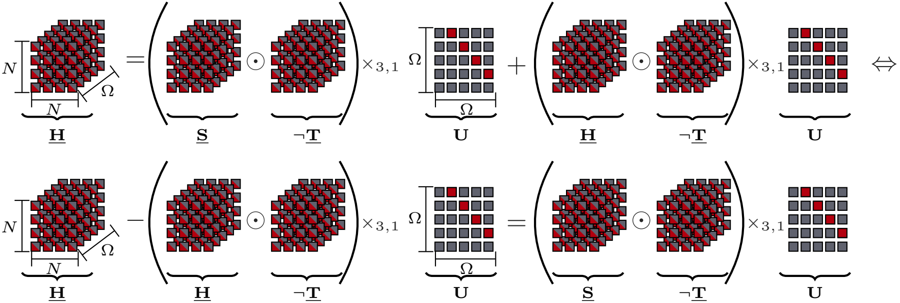
holdsAt($\mathit{fl}(X, Y){=}v,\ T$) $\leftarrow$
  holdsAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
  not terminatedAt($\mathit{fl}(X, Y){=}v,\ T_{prev}$),
  $\mathit{next}(T_{prev}, T)$.



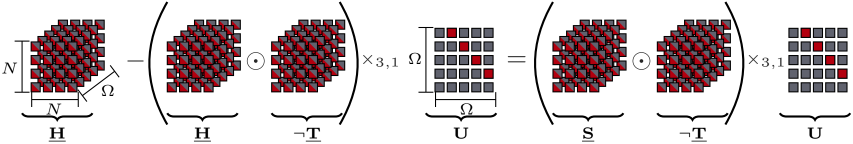
# Tensor-EC: Computing a Model

holdsAt($\mathit{fl}(X,Y){=}v,\ T$) $\leftarrow$
initiatedAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
not terminatedAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
$\mathit{next}(T_{prev},\ T)$.

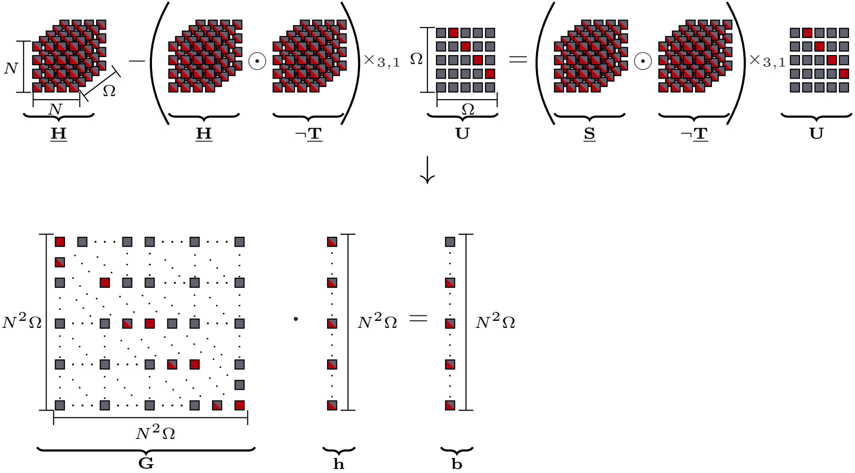
holdsAt($\mathit{fl}(X,Y){=}v,\ T$) $\leftarrow$
holdsAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
not terminatedAt($\mathit{fl}(X,Y){=}v,\ T_{prev}$),
$\mathit{next}(T_{prev},\ T)$.



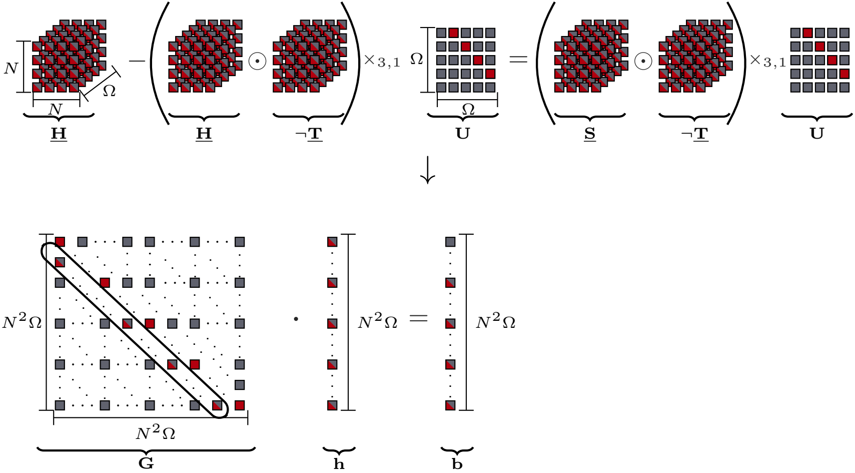
# Tensor-EC: Computing a Model



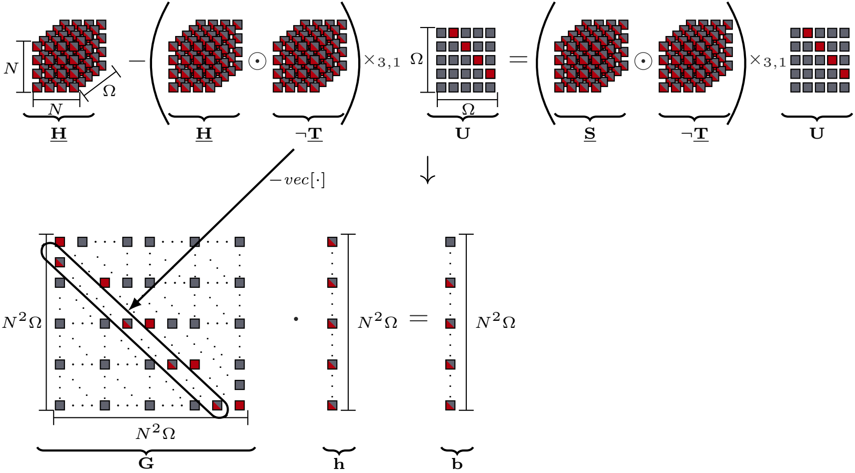
# Tensor-EC: Computing a Model



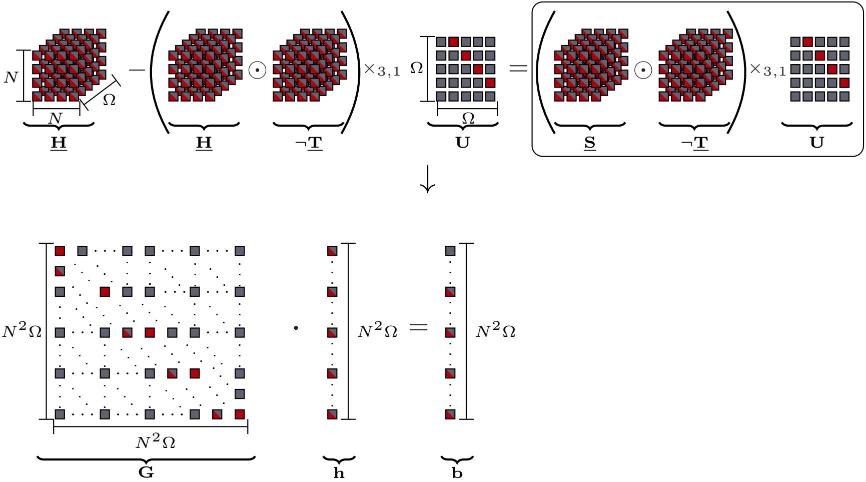
# Tensor-EC: Computing a Model



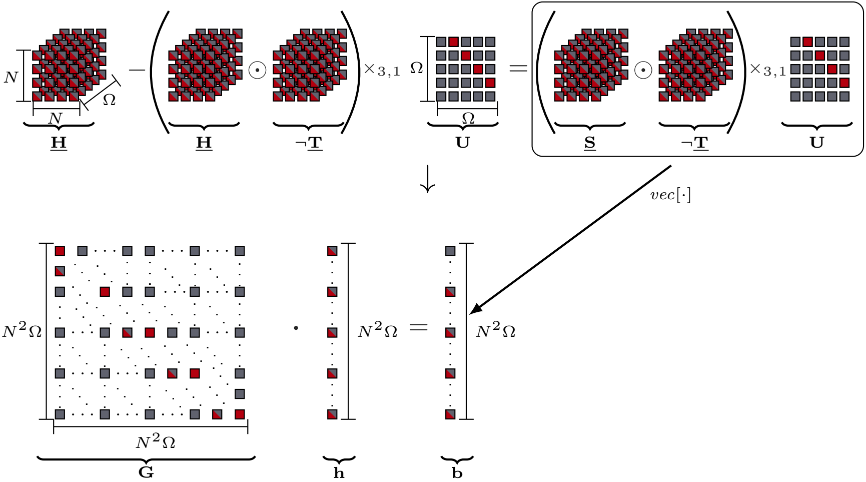
# Tensor-EC: Computing a Model



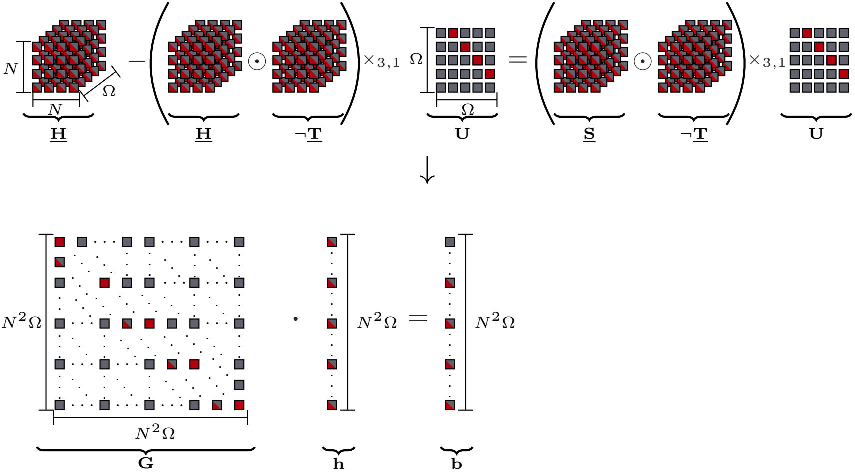
# Tensor-EC: Computing a Model



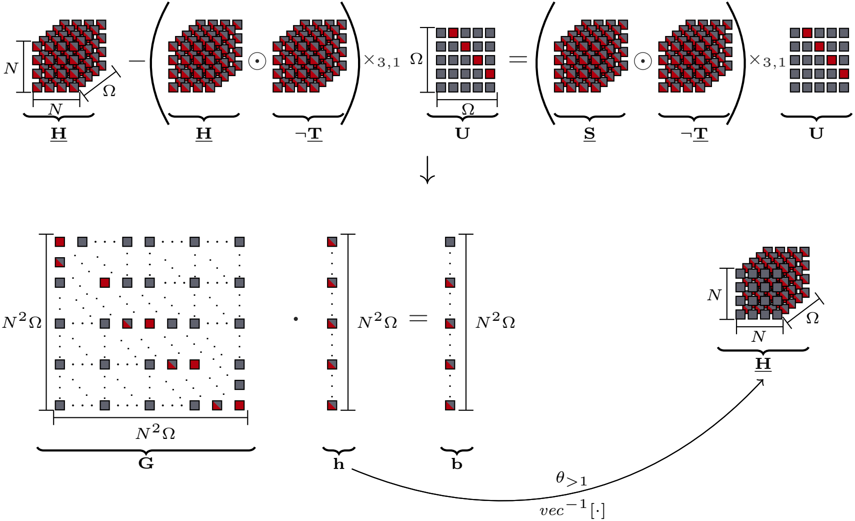
# Tensor-EC: Computing a Model



# Tensor-EC: Computing a Model



# Tensor-EC: Computing a Model



# Tensor-EC: Formal Properties

### Correctness

The **unique solution** of the equation coincides with the time-points at which a fluent-value pair holds, as expressed by the **perfect model** of the corresponding logic program.

# Tensor-EC: Formal Properties

## Correctness

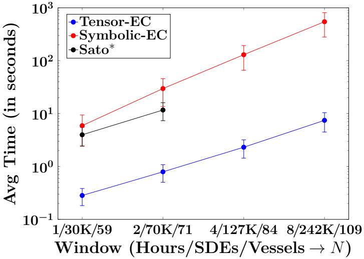
The **unique solution** of the equation coincides with the time-points at which a fluent-value pair holds, as expressed by the **perfect model** of the corresponding logic program.

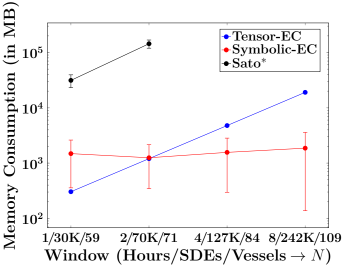
## Complexity

The time complexity of solving the equation is $\mathcal{O}(N^{p-1}\Omega)$ for order-$p$ tensors.

# Tensor-EC: Experimental Evaluation

## Brest

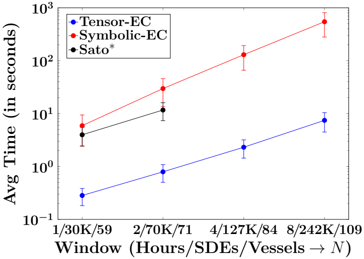


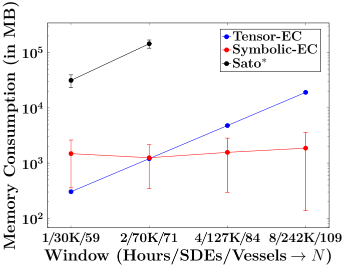


[*] Sato, T. A linear algebraic approach to datalog evaluation. *Theory and Practice of Logic Programming*, 17(3):244–265, 2017.

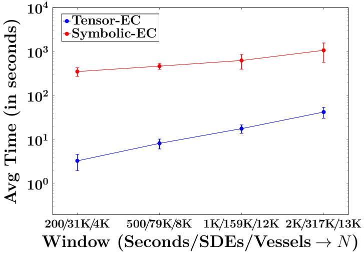
# Tensor-EC: Experimental Evaluation

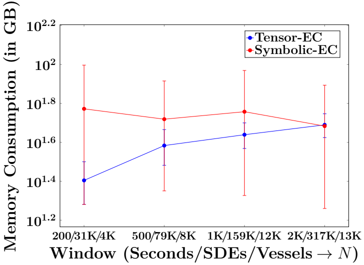
## Brest





## European seas





[*] Sato, T. A linear algebraic approach to datalog evaluation. *Theory and Practice of Logic Programming*, 17(3):244–265, 2017.

# Summary

## Scalable Complex Event Recognition

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:

[†] `https://github.com/eftsilio/Incremental_RTEC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.

[†] `https://github.com/eftsilio/Incremental_RTEC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.

[†] `https://github.com/eftsilio/Incremental_RTEC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.

[†] `https://github.com/eftsilio/Incremental_RTEC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.
  - Significant performance gains compared to re-computation from scratch.

[†] `https://github.com/eftsilio/Incremental_RTEC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.
  - Significant performance gains compared to re-computation from scratch.
- Modify the *how* — tensor-EC[‡]:

[†] https://github.com/eftsilio/Incremental_RTEC

[‡] https://github.com/eftsilio/Tensor-EC

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.
  - Significant performance gains compared to re-computation from scratch.
- Modify the *how* — tensor-EC[‡]:
  - Transforms symbolic representations into vector spaces.

[†] `https://github.com/eftsilio/Incremental_RTEC`

[‡] `https://github.com/eftsilio/Tensor-EC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.
  - Significant performance gains compared to re-computation from scratch.
- Modify the *how* — tensor-EC[‡]:
  - Transforms symbolic representations into vector spaces.
  - Reasoning through algebraic computation.

[†] https://github.com/eftsilio/Incremental_RTEC

[‡] https://github.com/eftsilio/Tensor-EC

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.
  - Significant performance gains compared to re-computation from scratch.
- Modify the *how* — tensor-EC[‡]:
  - Transforms symbolic representations into vector spaces.
  - Reasoning through algebraic computation.
  - Formal Properties.

[†] `https://github.com/eftsilio/Incremental_RTEC`

[‡] `https://github.com/eftsilio/Tensor-EC`

# Summary

## Scalable Complex Event Recognition

- Modify the *what* is to be computed — $RTEC_{inc}$[†]:
  - Handles delays/revisions in the input.
  - Optimal rule rewriting and evaluation of small sets early.
  - Formal Properties.
  - Significant performance gains compared to re-computation from scratch.
- Modify the *how* — tensor-EC[‡]:
  - Transforms symbolic representations into vector spaces.
  - Reasoning through algebraic computation.
  - Formal Properties.
  - Performance improvement wrt state-of-the-art.

[†] `https://github.com/eftsilio/Incremental_RTEC`

[‡] `https://github.com/eftsilio/Tensor-EC`

# Future Work

# Future Work

- Decide on the fly which execution strategy to adopt.

# Future Work

- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.

# Future Work

- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.
- Apply distribution and parallelization techniques.

# Future Work

- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.
- Apply distribution and parallelization techniques.
- Handle more expressive temporal specifications.

# Future Work

- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.
- Apply distribution and parallelization techniques.
- Handle more expressive temporal specifications.
- Formalize in tensor spaces $RTEC_{inc}$.

# Future Work

- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.
- Apply distribution and parallelization techniques.
- Handle more expressive temporal specifications.
- Formalize in tensor spaces $RTEC_{inc}$.
- Encode in tensors the time-intervals of fluents.

# Future Work

- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.
- Apply distribution and parallelization techniques.
- Handle more expressive temporal specifications.
- Formalize in tensor spaces $RTEC_{inc}$.
- Encode in tensors the time-intervals of fluents.
- Probabilistic CER in tensor spaces.

# Future Work

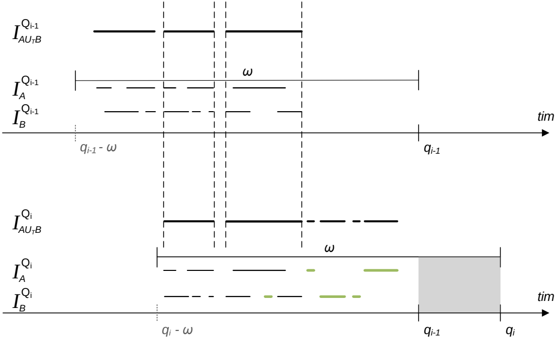
- Decide on the fly which execution strategy to adopt.
- Comparison with automata-based methods.
- Apply distribution and parallelization techniques.
- Handle more expressive temporal specifications.
- Formalize in tensor spaces $RTEC_{inc}$.
- Encode in tensors the time-intervals of fluents.
- Probabilistic CER in tensor spaces.
- Neuro-symbolic CER.

# Appendix

# $RTEC_{inc}$: Statically Determined Fluents

## Union

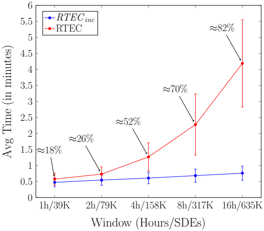
$$I_F^{\mathsf{Q_i}} = \left[ (I_{A \cup_T B}^{\mathsf{Q_{i-1}}}) \setminus_T \left[ \left[ (I_A^{del} \cup_T I_B^{del}) \setminus_T (I_{A \cap_T B}^{\mathsf{Q_{i-1}}}) \right] \cup_T (I_A^{del} \cap_T I_B^{del}) \right] \right] \cup_T (I_A^{ins} \cup_T I_B^{ins})$$



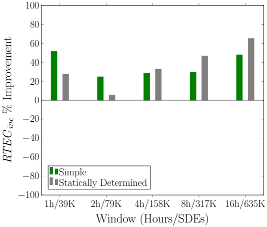
# $RTEC_{inc}$: Evaluation (Natural Delays)

## European seas

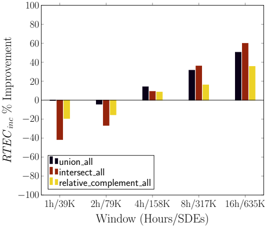
- Delays up to 16 hours
- $17M$ position signals, $34K$ vessels
- January 2016



(a)



(b)

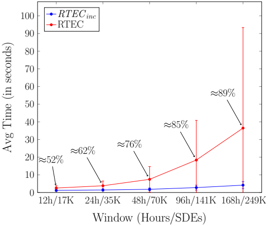


(c)

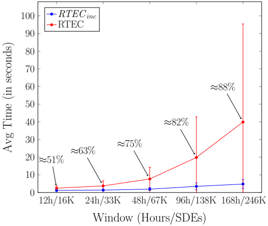
# $RTEC_{inc}$: Evaluation (Synthetic Delays)

## Brest

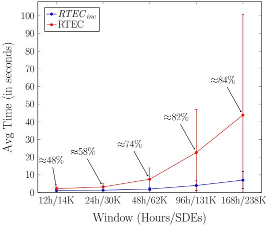
- $5M$ position signals, $5K$ vessels
- October 2015 — March 2016



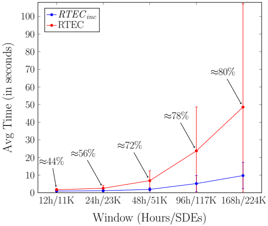
(a) 5%



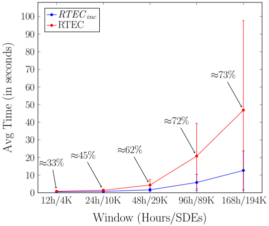
(b) 10%



(c) 20%



(d) 40%

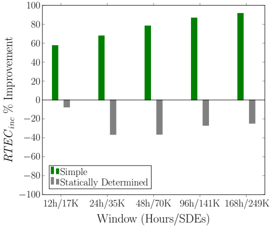


(e) 80%

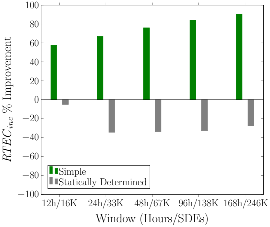
# $RTEC_{inc}$: Evaluation (Synthetic Delays)

## Brest

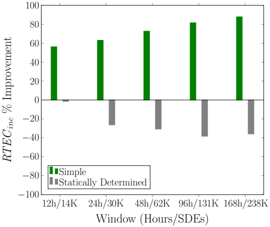
- $5M$ position signals, $5K$ vessels
- October 2015 — March 2016



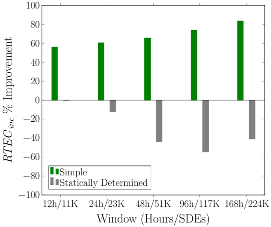
(a) 5%



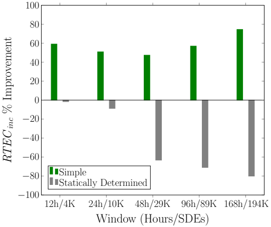
(b) 10%



(c) 20%



(d) 40%

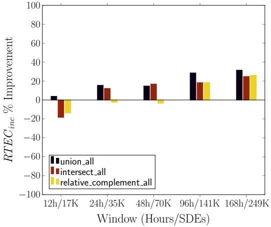


(e) 80%

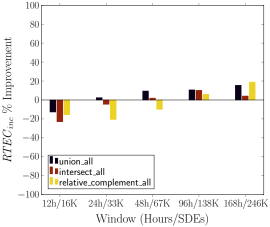
# $RTEC_{inc}$: Evaluation (Synthetic Delays)

## Brest

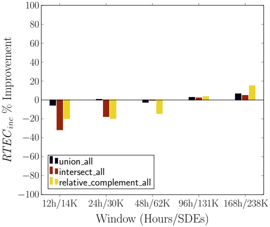
- $5M$ position signals, $5K$ vessels
- October 2015 — March 2016



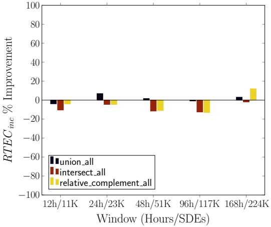
(a) 5%



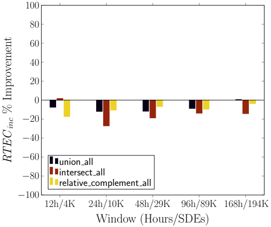
(b) 10%



(c) 20%



(d) 40%

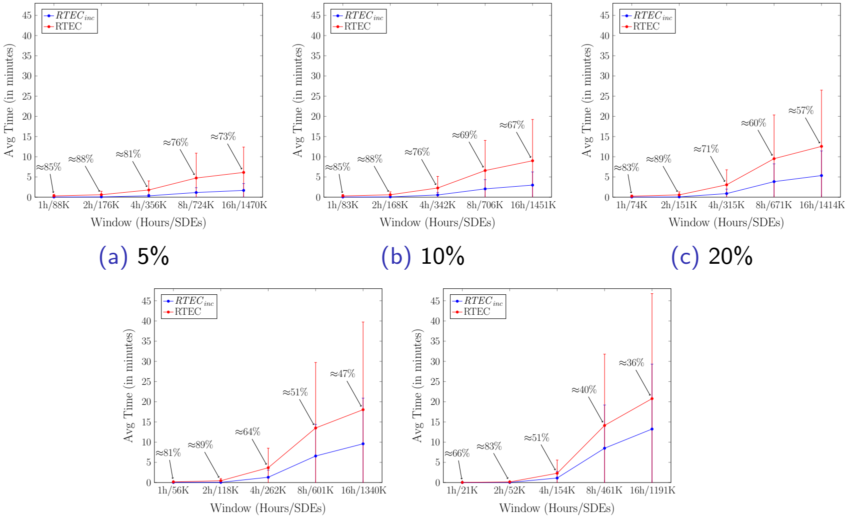


(e) 80%

# $RTEC_{inc}$: Evaluation (Synthetic Delays)

## Fleet Management

- $70M$ position signals, $6K$ vessels
- June 2018 — August 2018



(a) 5%

(b) 10%

(c) 20%

(d) 40%

(e) 80%

# Tensor-EC: Evaluation

## Simple Event Description

- One fluent and three input events.
- The fluent is defined by one initiatedAt and one terminatedAt rule, plus the *inertia* axiom.
- Temporal window of 20 time-points.

| **Method** | Reasoning time (msec) | Memory (MB) |
|---|---|---|
| tensor-EC | 1 | 0.01 |
| symbolic-EC | 0 | 5 |
| Sato[§] | 1 | 0.01 |
| Sakama et al.[¶] | 9205 | 475.3 |

[§] Sato, T. A linear algebraic approach to datalog evaluation. *Theory and Practice of Logic Programming*, 17(3):244–265, 2017.

[¶] Sakama et al. Logic programming in tensor spaces. *Annals of Mathematics and Artificial Intelligence*, 89, 12 2021.