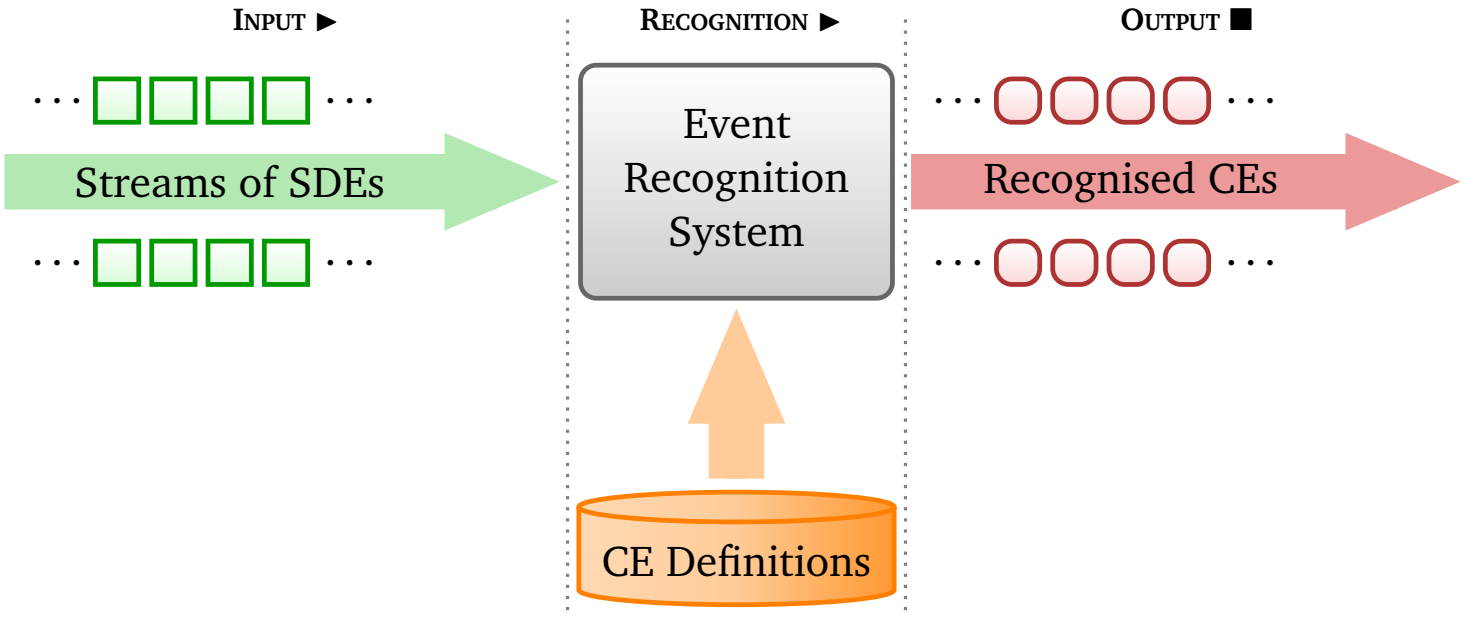
Event Calculus for Event Recognition

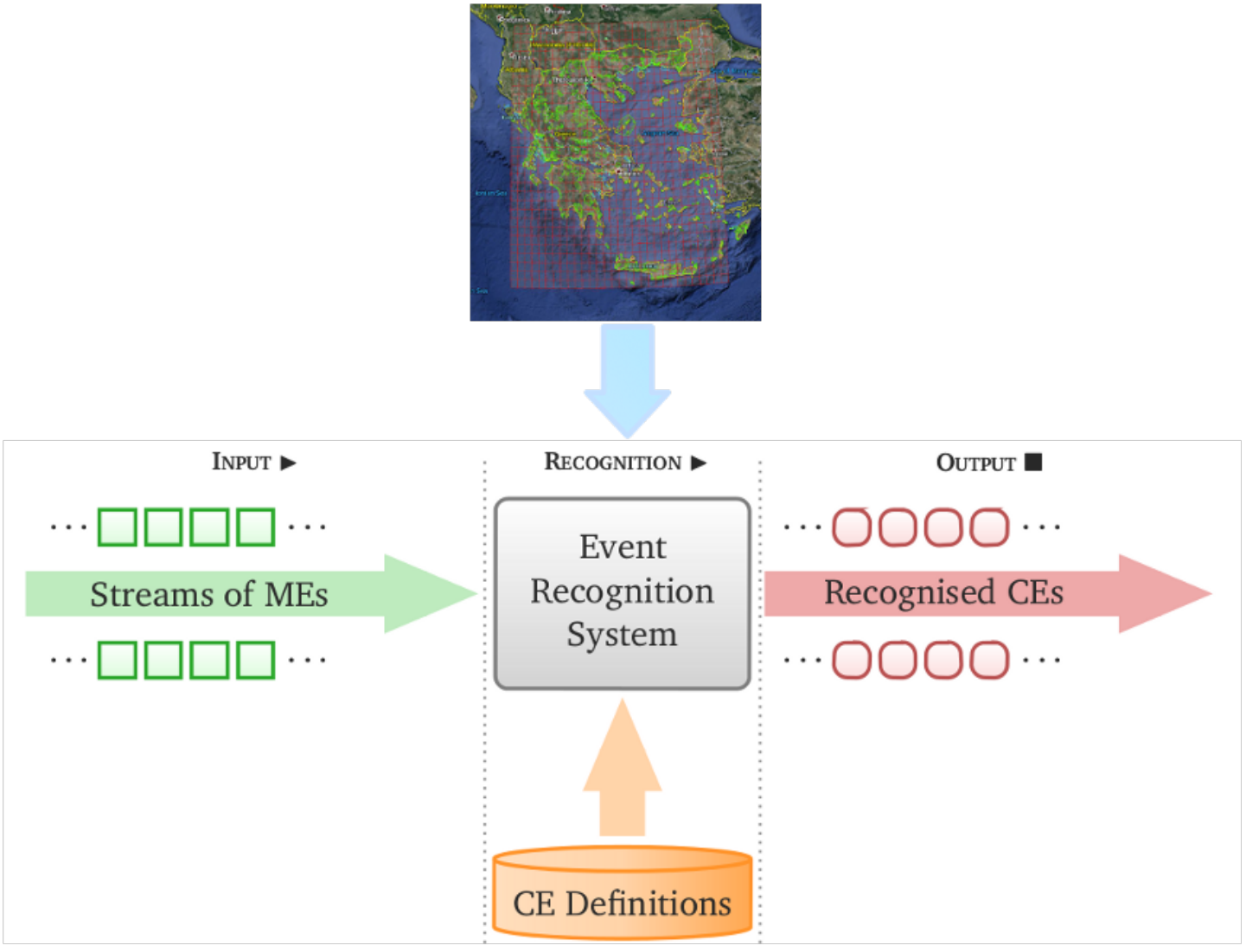
Alexander Artikis
Institute of Informatics & Telecommunications
NCSR Demokritos
Athens, Greece

http://cer.iit.demokritos.gr

Event Recognition



Event Recognition for Maritime Surveillance



Fast Approach

- A vessel is moving at a high speed ...
- towards another vessel ...
- (away from ports).

Possible Rendezvous

- Two vessels are suspiciously delayed ...
- in the same location ...
- at the same time.

Other Applications

- City transport & traffic management.
- Credit card fraud management.
- Unobtrusive assisted living.
- Social media verification.
- Public space surveillance.

Event Recognition

Requirements:

- Efficient reasoning
  - to support real-time decision-making in large-scale, (geographically) distributed applications.
- Reasoning under uncertainty
  - to deal with various types of noise.
- Automated knowledge construction
  - to avoid the time-consuming, error-prone manual CE definition development.

Event Recognition

Requirements:

- Efficient reasoning
  - to support real-time decision-making in large-scale, (geographically) distributed applications.
- Reasoning under uncertainty
  - to deal with various types of noise.
- Automated knowledge construction
  - to avoid the time-consuming, error-prone manual CE definition development.

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.

CE Definitions in the Run-Time Event Calculus: Simple Fluents

CE definition:

$$\textbf{initiatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{In_1},\ T), [\text{conditions}]$$
$$\ldots$$
$$\textbf{initiatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{In_i},\ T), [\text{conditions}]$$

$$\textbf{terminatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{T_1},\ T), [\text{conditions}]$$
$$\ldots$$
$$\textbf{terminatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{T_j},\ T), [\text{conditions}]$$

where

$$\text{conditions:} \quad {}^{0-K}\textbf{happensAt}(E_k,\ T), \ {}^{0-M}\textbf{holdsAt}(F_m,\ T), \ {}^{0-N}\text{atemporal-constraint}_n$$

CE Definitions in the Run-Time Event Calculus: Simple Fluents

CE definition:

initiatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{In_1},\ T)$,
[conditions]

...

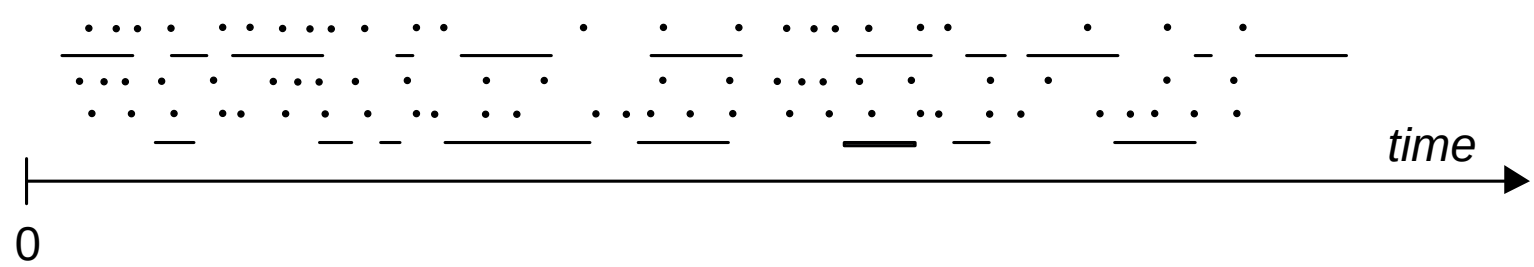
initiatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{In_i},\ T)$,
[conditions]

terminatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{T_1},\ T)$,
[conditions]

...

terminatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{T_j},\ T)$,
[conditions]

CE recognition:



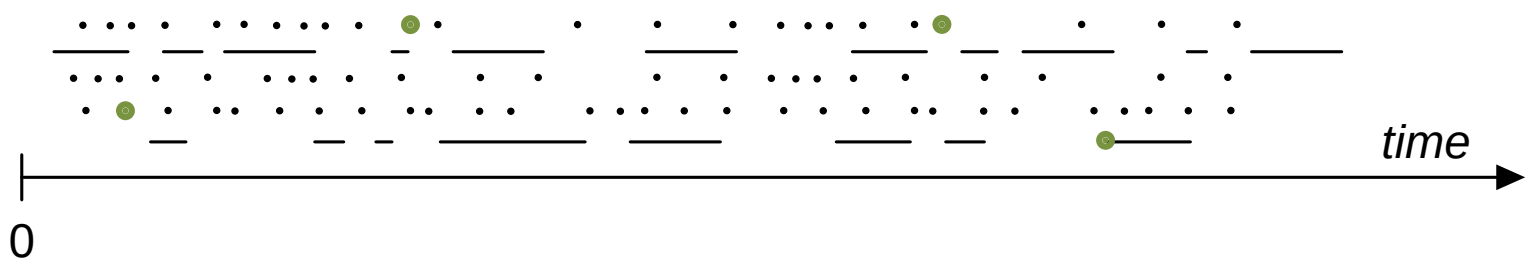
CE Definitions in the Run-Time Event Calculus: Simple Fluents

CE definition:

$$\textbf{initiatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{In_1},\ T), [\text{conditions}]$$
$$\ldots$$
$$\textbf{initiatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{In_i},\ T), [\text{conditions}]$$

$$\textbf{terminatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{T_1},\ T), [\text{conditions}]$$
$$\ldots$$
$$\textbf{terminatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{T_j},\ T), [\text{conditions}]$$

CE recognition:



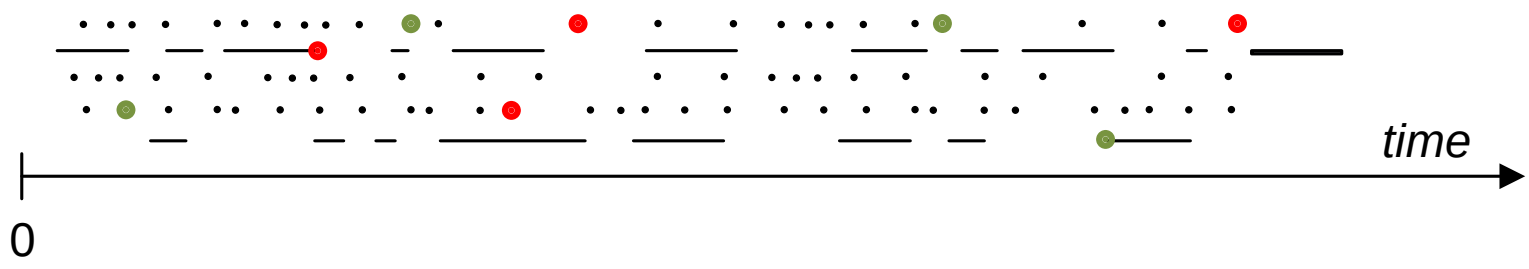
CE Definitions in the Run-Time Event Calculus: Simple Fluents

CE definition:

$$\textbf{initiatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{In_1},\ T), [\text{conditions}]$$
$$\ldots$$
$$\textbf{initiatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{In_i},\ T), [\text{conditions}]$$

$$\textbf{terminatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{T_1},\ T), [\text{conditions}]$$
$$\ldots$$
$$\textbf{terminatedAt}(CE,\ T) \leftarrow \textbf{happensAt}(E_{T_j},\ T), [\text{conditions}]$$

CE recognition:



CE Definitions in the Run-Time Event Calculus: Simple Fluents

CE definition:

initiatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{In_1},\ T)$,
[conditions]

...

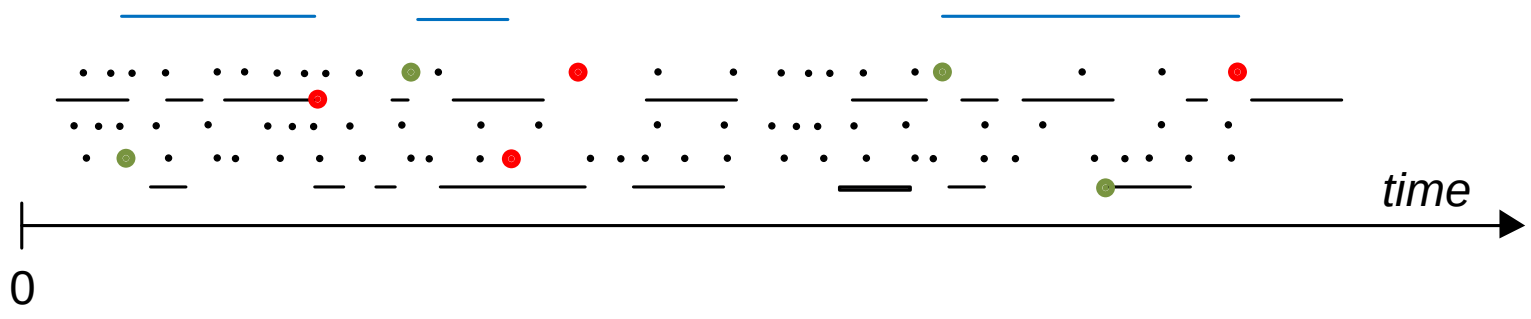
initiatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{In_i},\ T)$,
[conditions]

terminatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{T_1},\ T)$,
[conditions]

...

terminatedAt$(CE,\ T) \leftarrow$
happensAt$(E_{T_j},\ T)$,
[conditions]

CE recognition: **holdsFor**$(CE,\ I)$



CE Definitions in the Run-Time Event Calculus: Simple Fluents

CE definition:

$$\begin{aligned}
&\textbf{initiatedAt}(\mathit{suspicious}(\mathit{Area}) = \text{true},\ T) \leftarrow \\
&\quad \textbf{happensAt}(\textbf{start}(\mathit{stopped}(\mathit{Vessel}) = \text{true}),\ T), \\
&\quad \textbf{holdsAt}(\mathit{coord}(\mathit{Vessel}) = (\mathit{Lon}, \mathit{Lat}),\ T), \\
&\quad \mathit{close}(\mathit{Lon}, \mathit{Lat}, \mathit{Area}), \\
&\quad \textbf{holdsAt}(\mathit{vesselsStoppedIn}(\mathit{Area}) = N,\ T),\ N > 3 \\
&\textbf{terminatedAt}(\mathit{suspicious}(\mathit{Area}) = \text{true},\ T) \leftarrow \\
&\quad \textbf{happensAt}(\textbf{end}(\mathit{stopped}(\mathit{Vessel}) = \text{true}),\ T), \\
&\quad \textbf{holdsAt}(\mathit{coord}(\mathit{Vessel}) = (\mathit{Lon}, \mathit{Lat}),\ T), \\
&\quad \mathit{close}(\mathit{Lon}, \mathit{Lat}, \mathit{Area}), \\
&\quad \textbf{holdsAt}(\mathit{vesselsStoppedIn}(\mathit{Area}) = N,\ T),\ N \leq 3
\end{aligned}$$

CE recognition: $\textbf{holdsFor}(\mathit{suspicious}(\mathit{Area}) = \text{true},\ I)$

CE Definitions in the Run-Time Event Calculus: Statically Determined Fluents

CE definition:

$$\begin{aligned}&\textbf{holdsFor}(CE,\ I)\ \leftarrow\\ &\quad \textbf{holdsFor}(F_1,\ I_{F_1}),\\ &\quad \ldots,\\ &\quad \textbf{holdsFor}(F_f,\ I_{F_f}),\\ &\quad \mathit{interval\_manipulation}_1(I_\alpha, \ldots, I_\omega),\\ &\quad \ldots,\\ &\quad \mathit{interval\_manipulation}_k(I_A, \ldots, I_\Omega)\end{aligned}$$

where

$$\begin{aligned}&\mathit{interval\_manipulation}(I_1, \ldots, I_n):\\ &\quad I_1 \cup \cdots \cup I_n\\ &\quad I_1 \cap \cdots \cap I_n\\ &\quad I_1 \setminus I_2\end{aligned}$$

CE Definitions in the Run-Time Event Calculus: Statically Determined Fluents

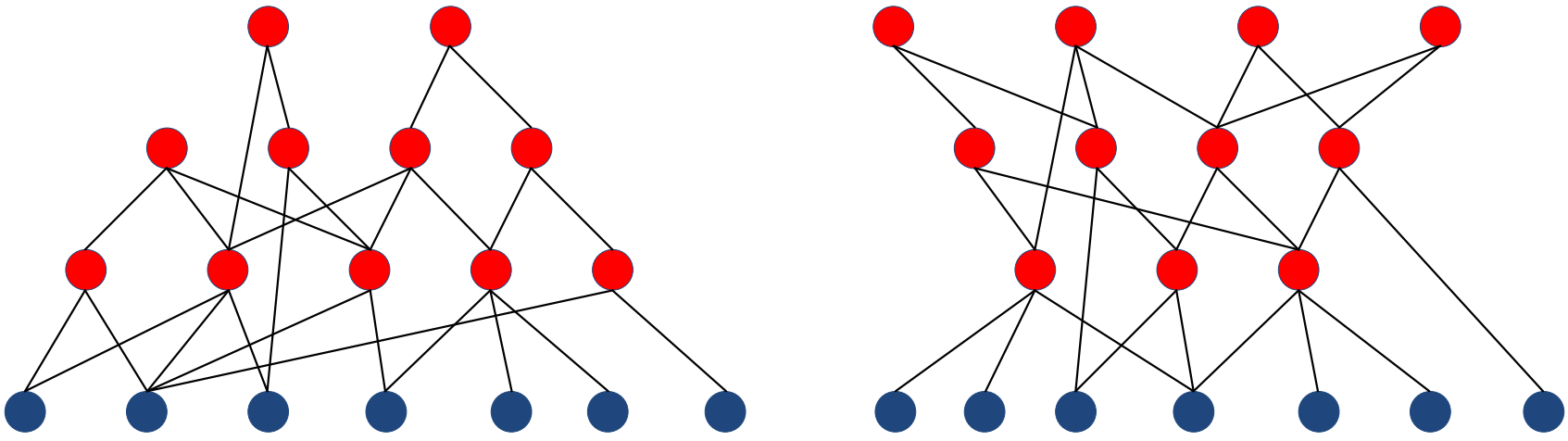
CE definition:

$$
\begin{aligned}
&\mathit{suspicious}(\mathit{Vessel}_1, \mathit{Vessel}_2) \text{ iff} \\
&\quad \mathit{abnormal}(\mathit{Vessel}_1), \\
&\quad \mathit{abnormal}(\mathit{Vessel}_2), \\
&\quad \mathit{close}(\mathit{Vessel}_1, \mathit{Vessel}_2), \\
&\quad \text{not } (\mathit{in}(\mathit{Vessel}_1, \mathit{Area}) \text{ or } \mathit{in}(\mathit{Vessel}_2, \mathit{Area}))
\end{aligned}
$$

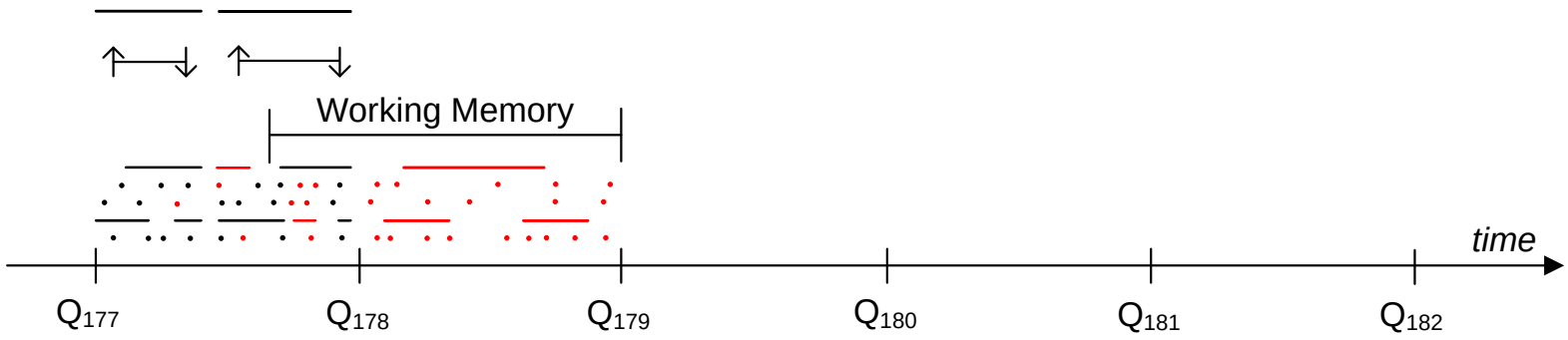
Compiled CE definition:

$$
\begin{aligned}
&\textbf{holdsFor}(\mathit{suspicious}(\mathit{Vessel}_1, \mathit{Vessel}_2) = \text{true}, \ (I_1 \cap I_2 \cap I_3) \setminus (I_4 \cup I_5)) \leftarrow \\
&\quad \textbf{holdsFor}(\mathit{abnormal}(\mathit{Vessel}_1) = \text{true}, \ I_1), \\
&\quad \textbf{holdsFor}(\mathit{abnormal}(\mathit{Vessel}_2) = \text{true}, \ I_2), \\
&\quad \textbf{holdsFor}(\mathit{close}(\mathit{Vessel}_1, \mathit{Vessel}_2) = \text{true}, \ I_3), \\
&\quad \textbf{holdsFor}(\mathit{in}(\mathit{Vessel}_1, \mathit{Area}) = \text{true}, \ I_4), \\
&\quad \textbf{holdsFor}(\mathit{in}(\mathit{Vessel}_2, \mathit{Area}) = \text{true}, \ I_5)
\end{aligned}
$$

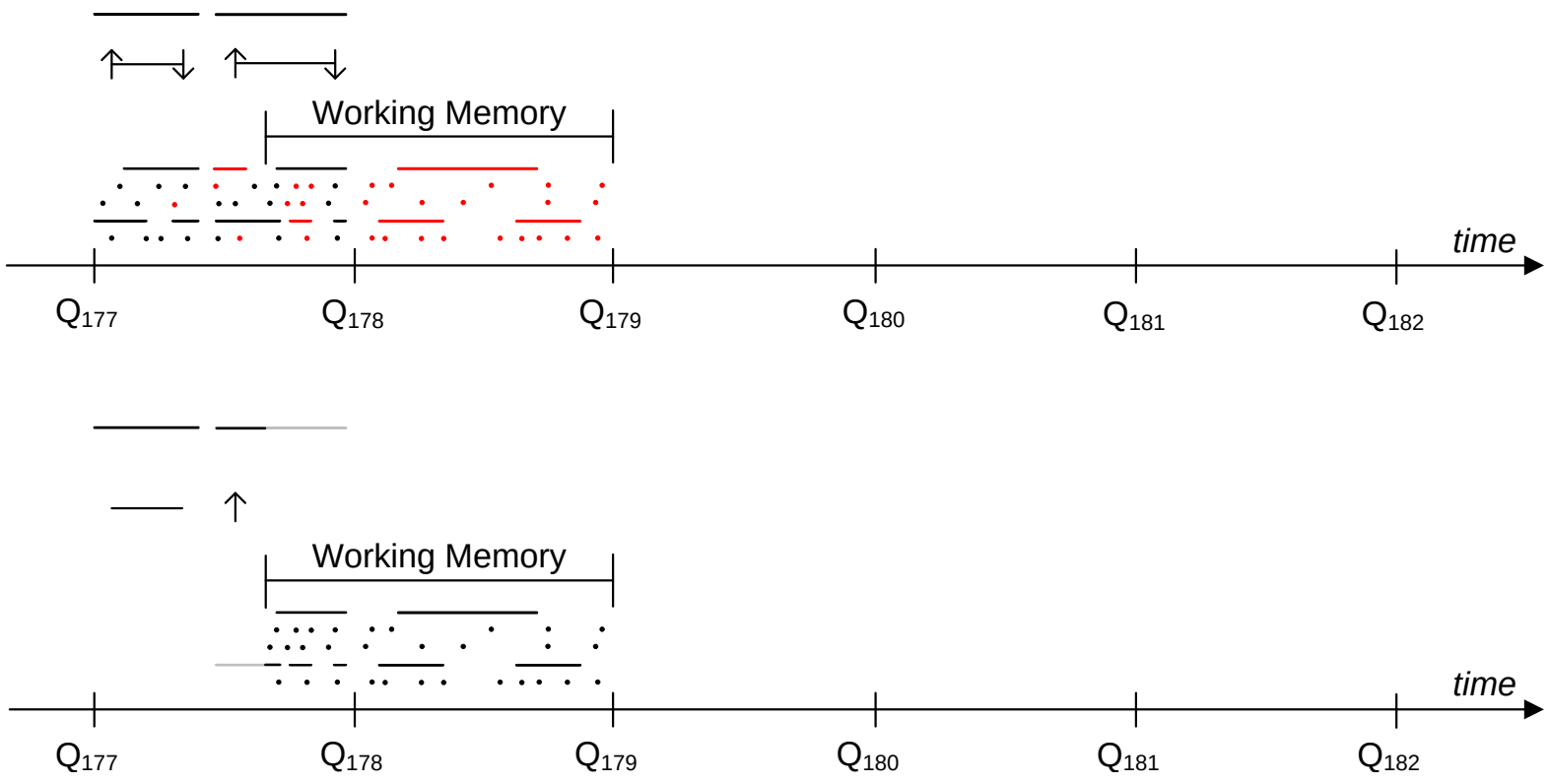
CE Hierarchies



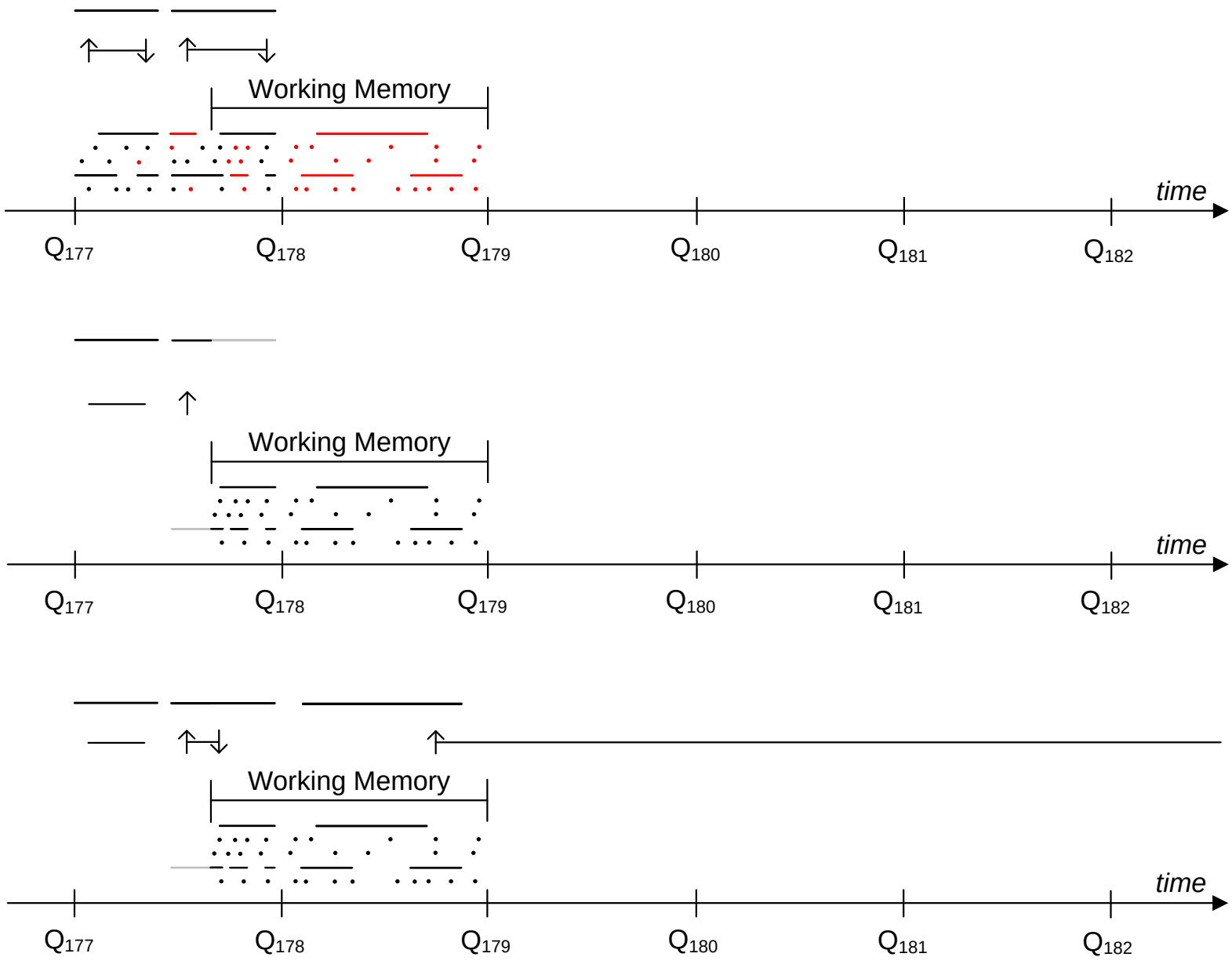
Run-Time Event Calculus



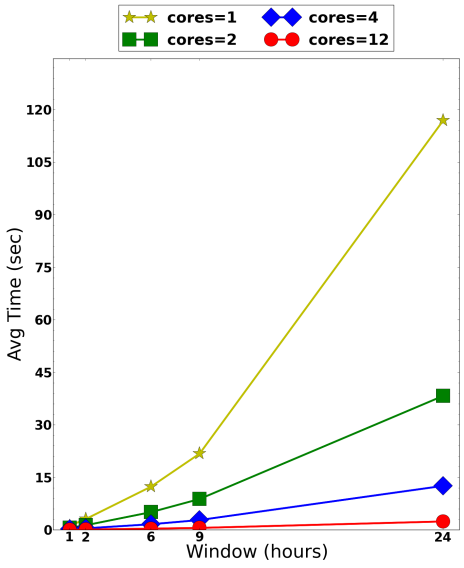
Run-Time Event Calculus

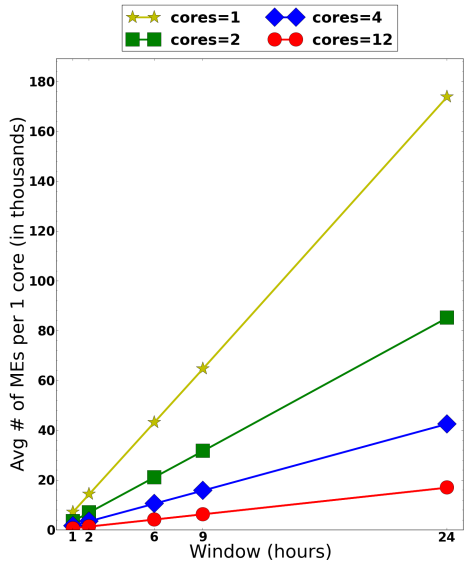


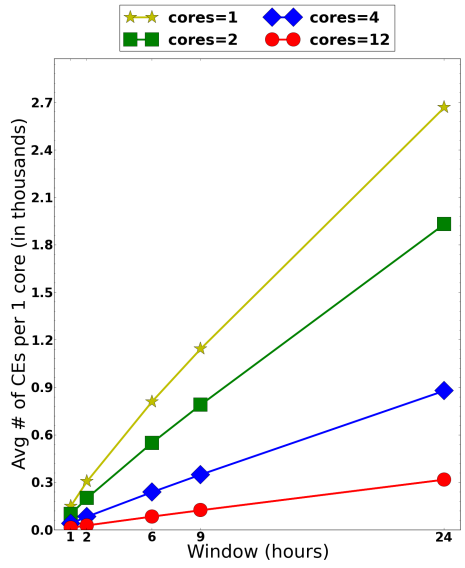
Run-Time Event Calculus



Empirical Analysis: Maritime Surveillance

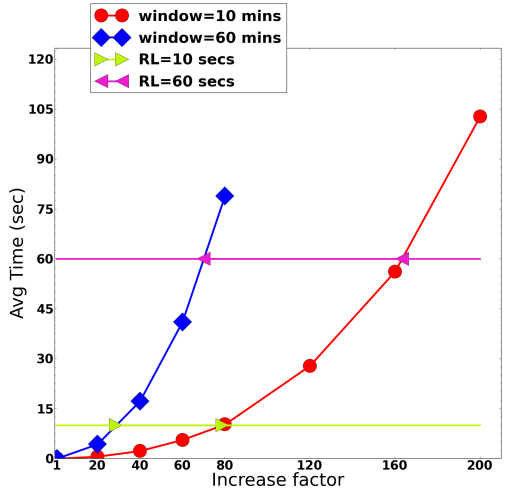


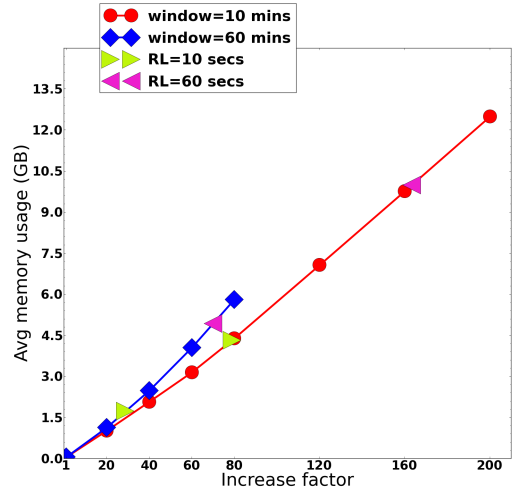


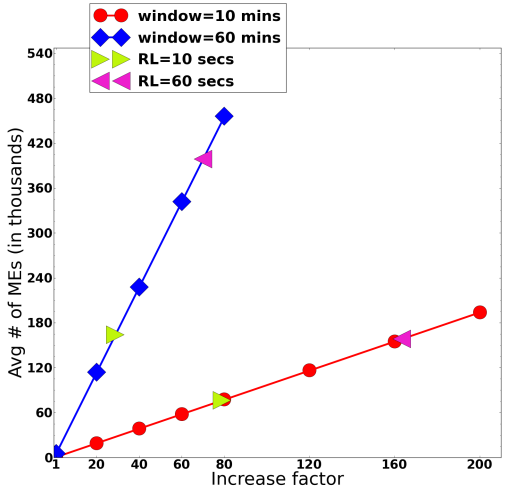


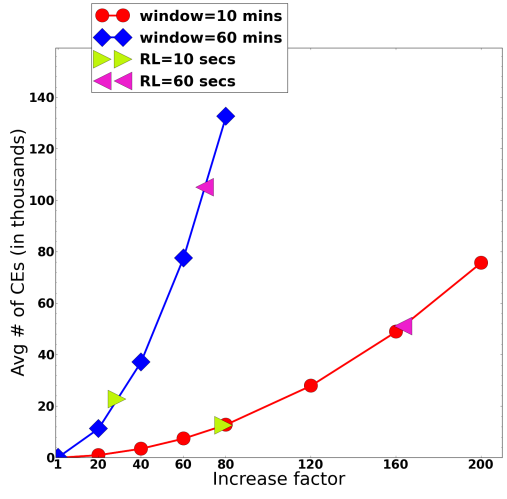
- June–August 2009.
- >6K vessels sailing in the Greek seas.
- >4K areas of interest/polygons (1K sides per polygon).
- >15M streaming critical movement events.
- CE recognition requires complex spatio-temporal reasoning.

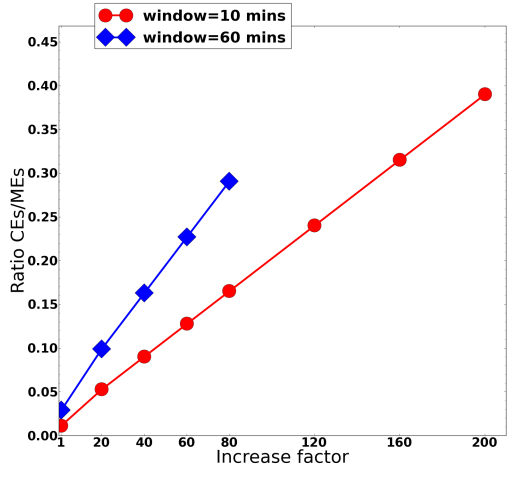
Big Maritime Data: x200 Larger Datasets









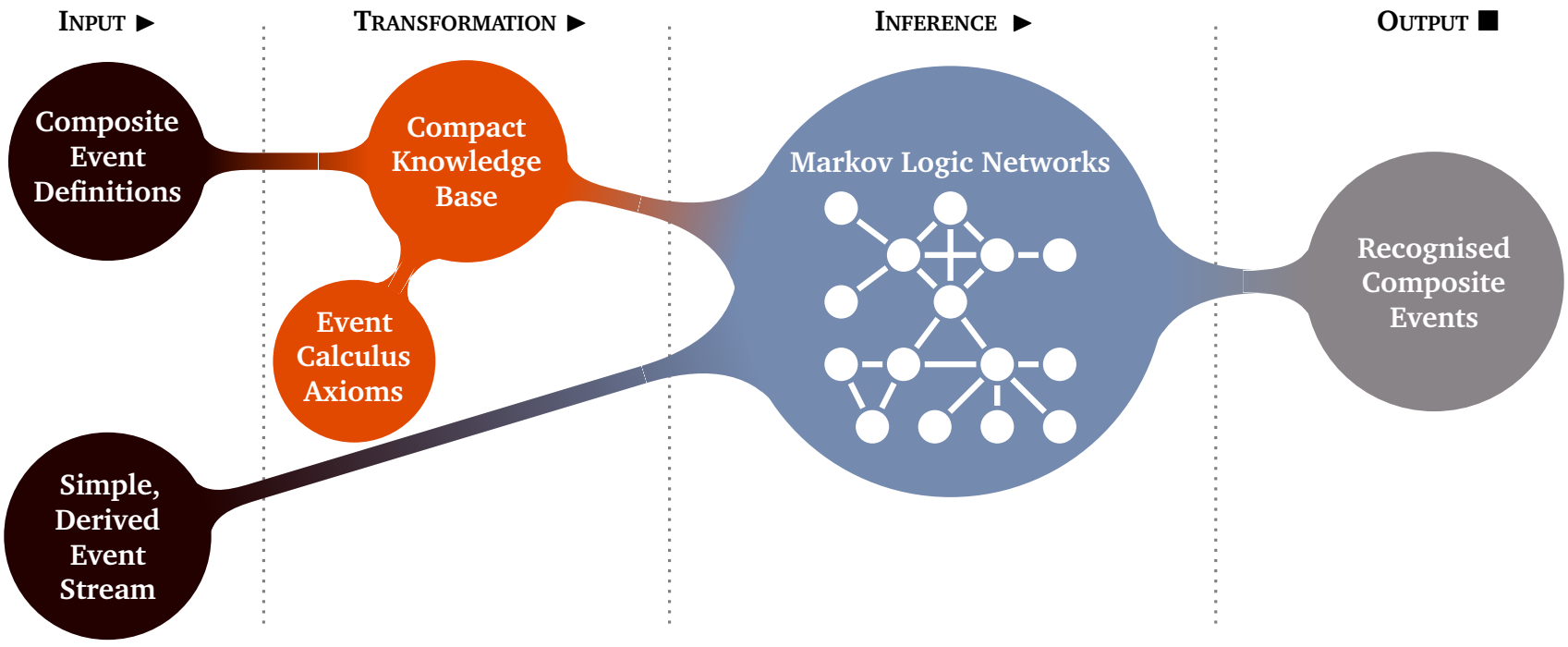


Event Recognition

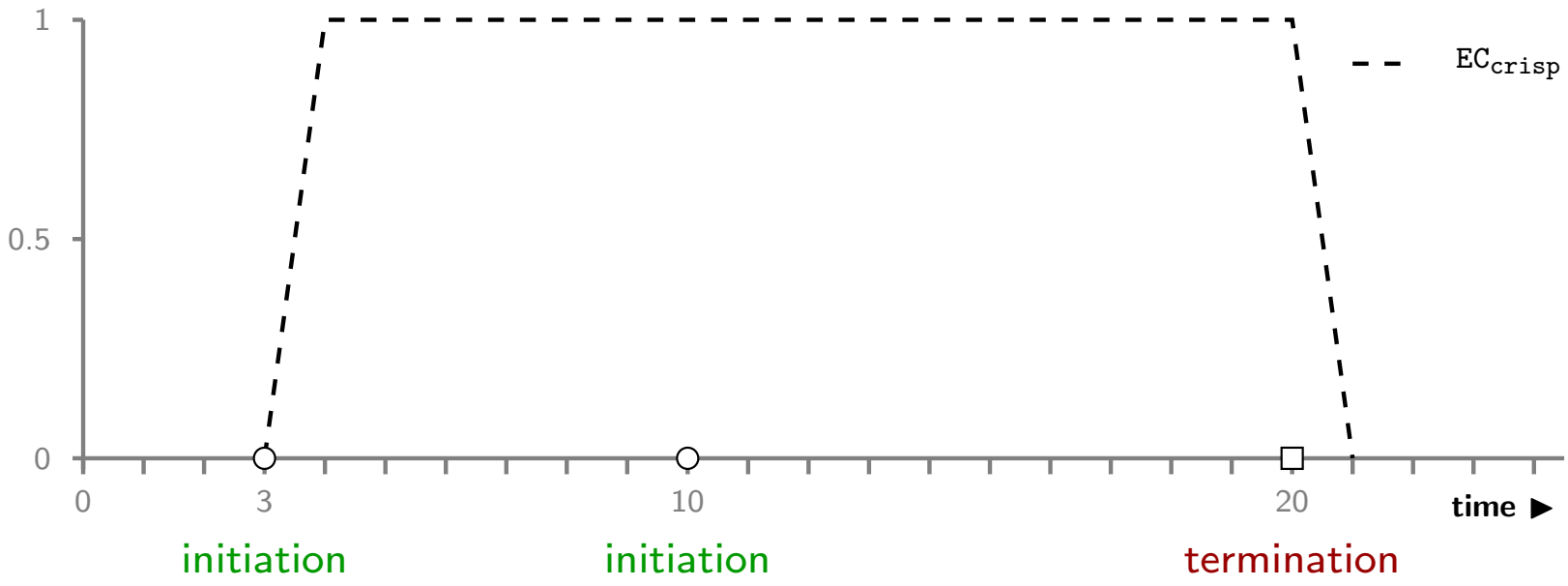
Requirements:

- Efficient reasoning
  - to support real-time decision-making in large-scale, (geographically) distributed applications.
- Reasoning under uncertainty
  - to deal with various types of noise.
- Automated knowledge construction
  - to avoid the time-consuming, error-prone manual CE definition development.

Probabilistic Event Calculus based on Markov Logic Networks (MLN-EC)

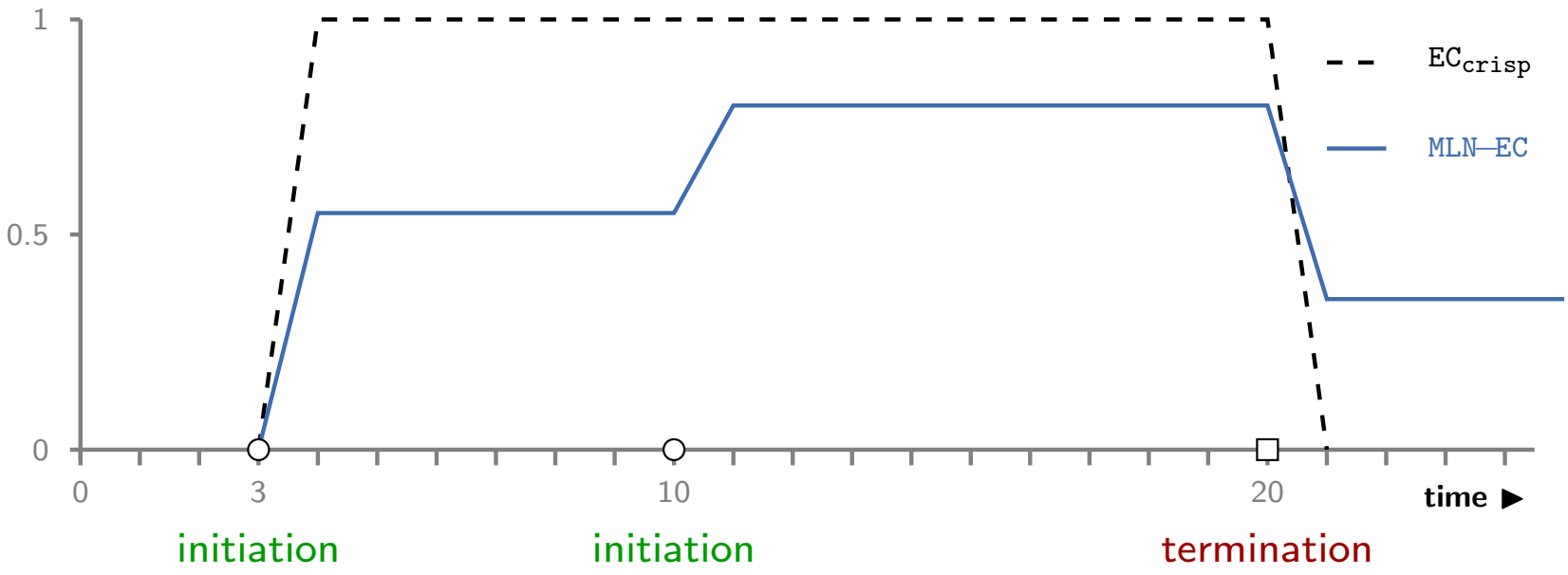


MLN-EC: Inertia



- ∞ holdsAt(CE, T+1)$\Leftarrow$ [Initiation Conditions]
- ∞ $\neg$holdsAt(CE, T+1)$\Leftarrow$ [Termination Conditions]
- ∞ $\neg$holdsAt(CE, T+1)$\Leftarrow$ $\neg$holdsAt(CE), T) $\wedge$ $\neg$[Initiation Conditions]
- ∞ holdsAt(CE, T+1)$\Leftarrow$ holdsAt(CE, T) $\wedge$ $\neg$[Termination Conditions]

MLN-EC: Inertia



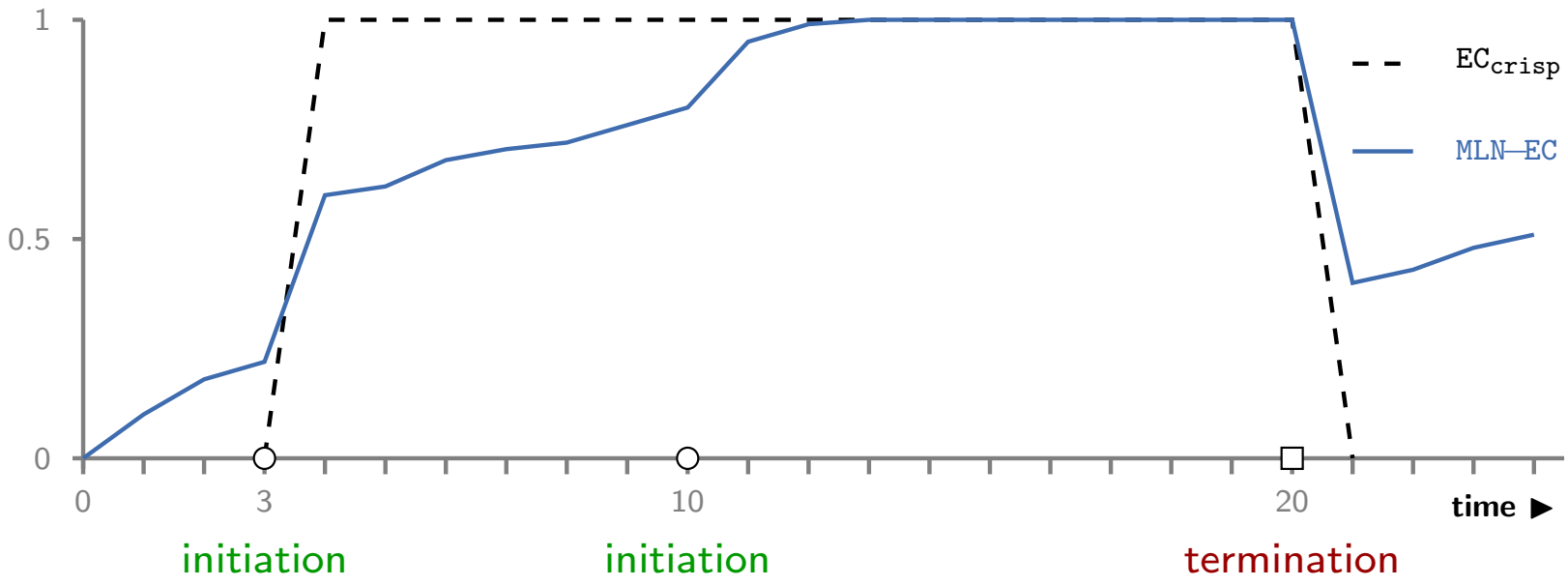
1.2 $\text{holdsAt}(CE,\ T+1) \Leftarrow$
$[\texttt{Initiation Conditions}]$

∞ $\neg\text{holdsAt}(CE,\ T+1) \Leftarrow$
$\neg\text{holdsAt}(CE),\ T) \wedge$
$\neg[\texttt{Initiation Conditions}]$

0.7 $\neg\text{holdsAt}(CE,\ T+1) \Leftarrow$
$[\texttt{Termination Conditions}]$

∞ $\text{holdsAt}(CE,\ T+1) \Leftarrow$
$\text{holdsAt}(CE,\ T) \wedge$
$\neg[\texttt{Termination Conditions}]$

MLN-EC: Inertia



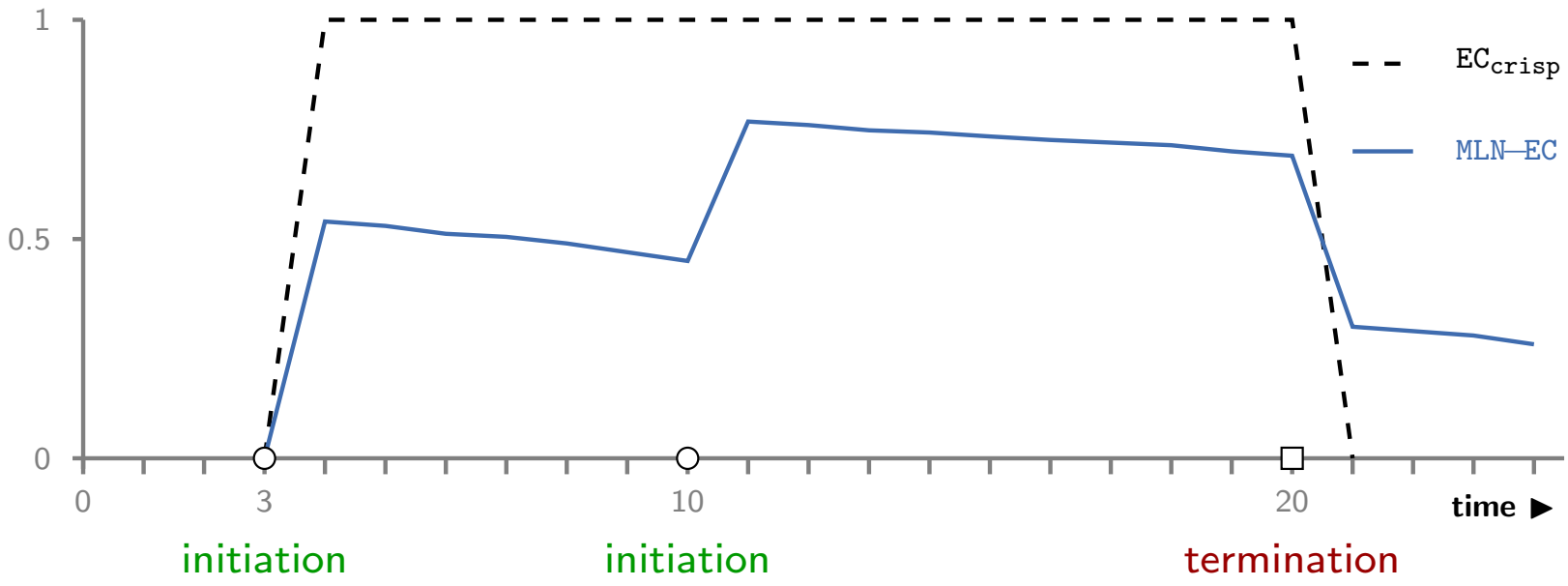
1.2 holdsAt(CE, T+1)⇐
[Initiation Conditions]

2.3 ¬holdsAt(CE, T+1)⇐
¬holdsAt(CE), T) ∧
¬[Initiation Conditions]

0.7 ¬holdsAt(CE, T+1)⇐
[Termination Conditions]

∞ holdsAt(CE, T+1)⇐
holdsAt(CE, T) ∧
¬[Termination Conditions]

MLN-EC: Inertia



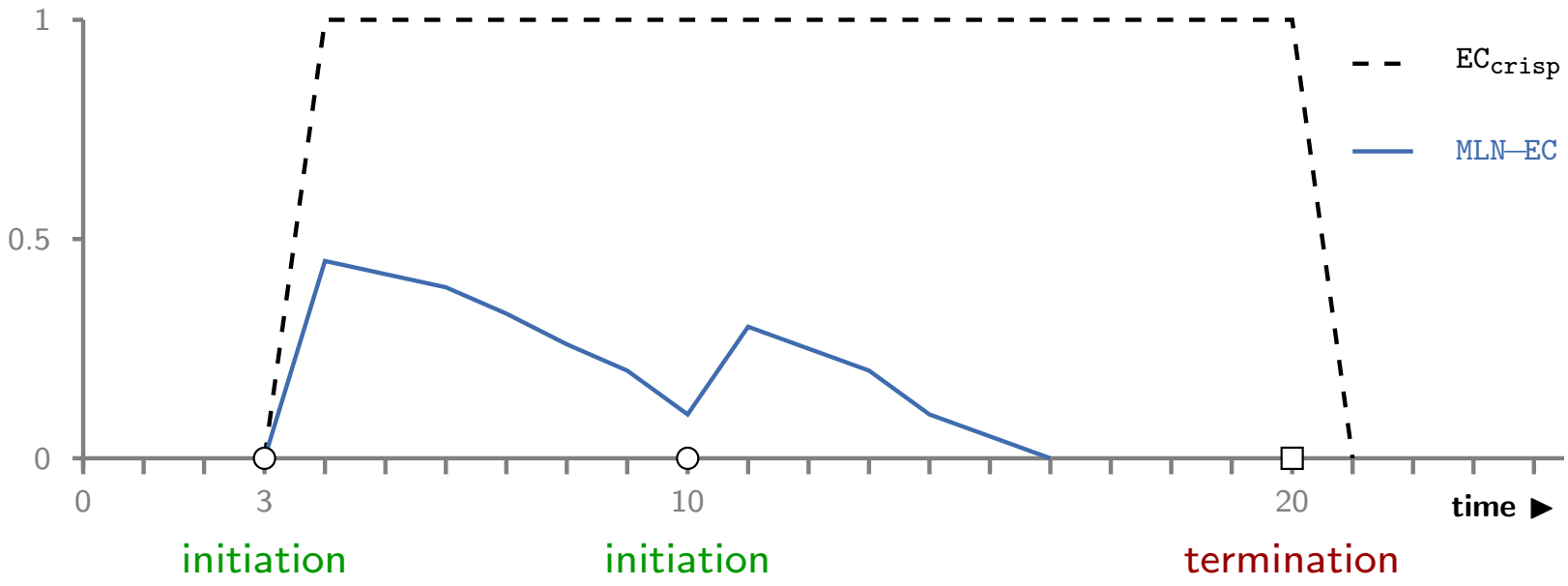
1.2 holdsAt(CE, T+1)⇐
[Initiation Conditions]

∞ ¬holdsAt(CE, T+1)⇐
¬holdsAt(CE), T) ∧
¬[Initiation Conditions]

0.7 ¬holdsAt(CE, T+1)⇐
[Termination Conditions]

2.3 holdsAt(CE, T+1)⇐
holdsAt(CE, T) ∧
¬[Termination Conditions]

MLN-EC: Inertia



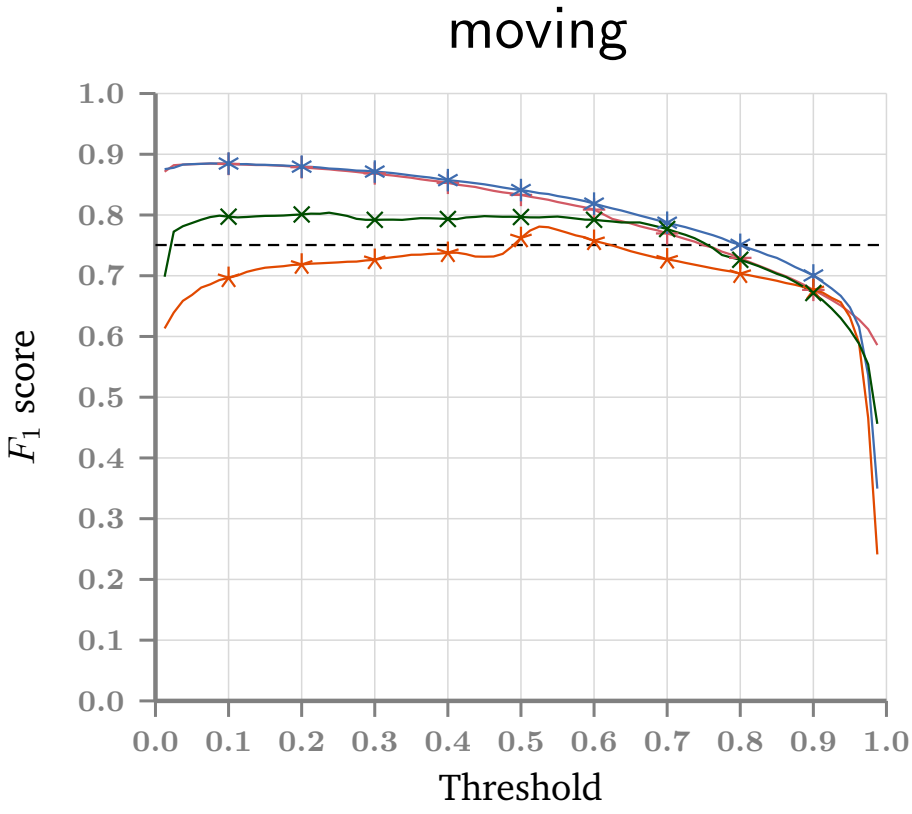
1.2 holdsAt(CE, T+1)⇐
[Initiation Conditions]

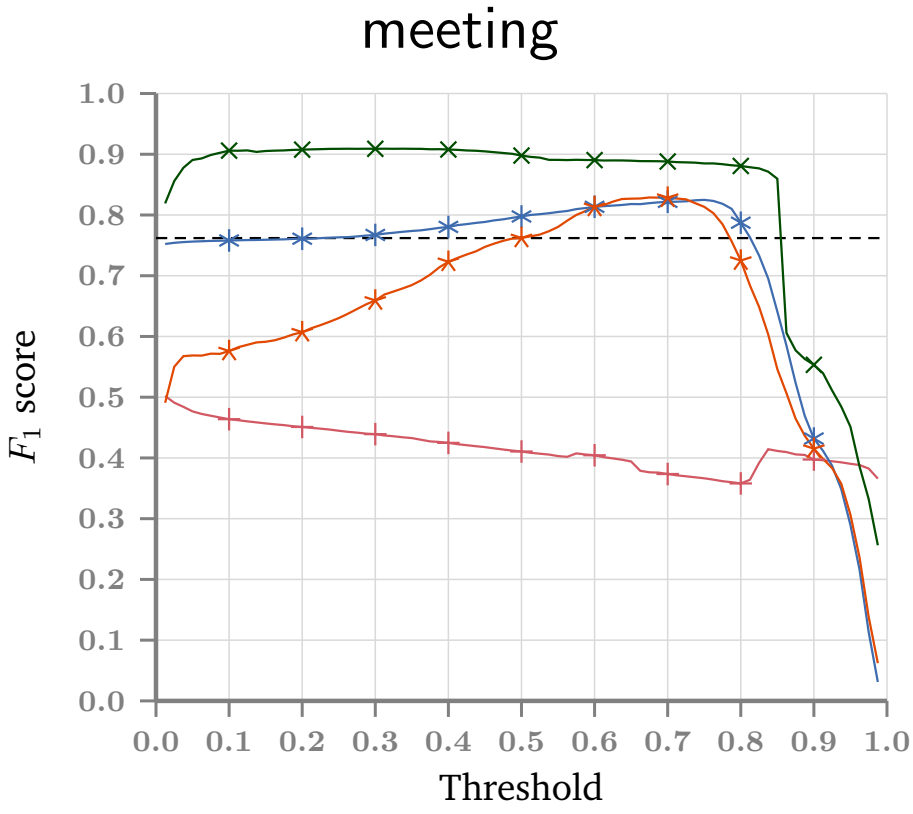
∞ ¬holdsAt(CE, T+1)⇐
¬holdsAt(CE), T) ∧
¬[Initiation Conditions]

0.7 ¬holdsAt(CE, T+1)⇐
[Termination Conditions]

0.6 holdsAt(CE, T+1)⇐
holdsAt(CE, T) ∧
¬[Termination Conditions]

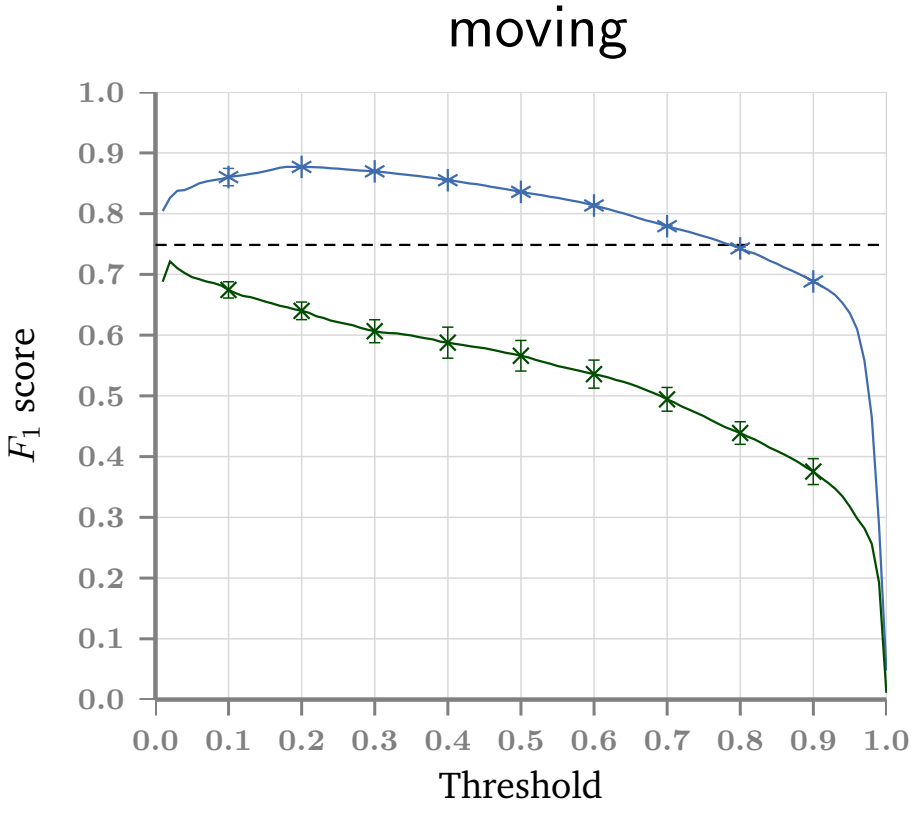
MLN-EC: Complete data (marginal inference)

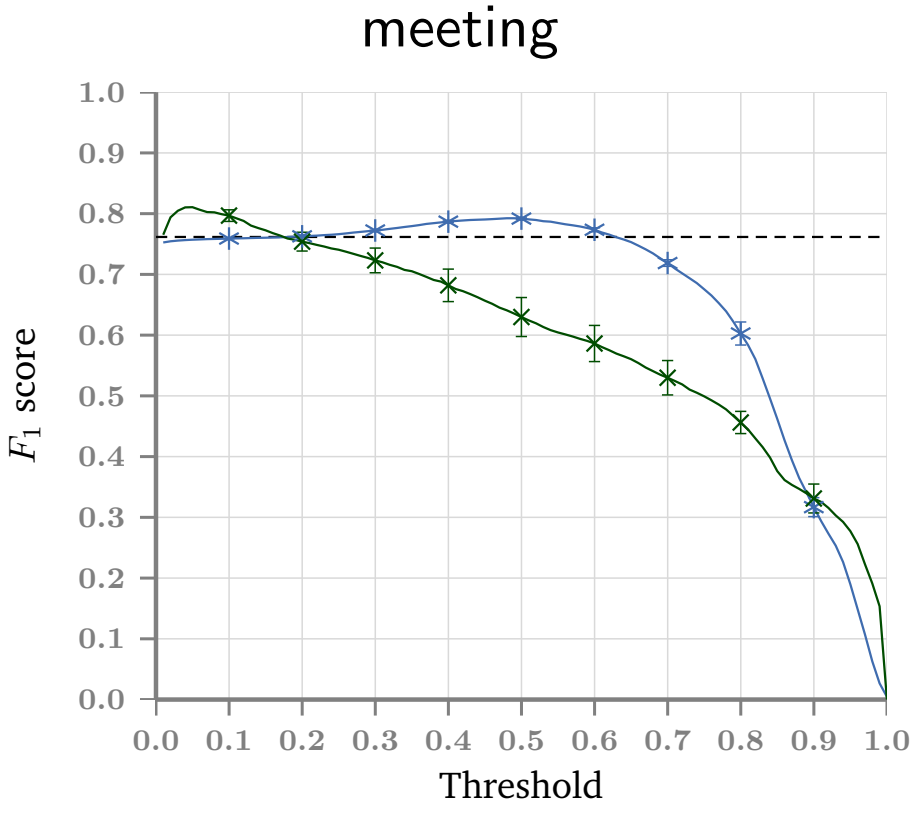




----	$\texttt{EC}_{\texttt{crisp}}$	:	Crisp Event Calculus
—×—	$\texttt{l}-\texttt{CRF}$	:	Linear-chain Conditional Random Field
—+—	$MLN-EC_{HI}$	:	Hard-constrained inertia
—✳—	$MLN-EC_{SIT}$	:	Soft-constrained termination inertial rules
—✳—	$MLN-EC_{SI}$	:	Soft-constrained inertial rules

MLN-EC: Incomplete data (marginal inference)





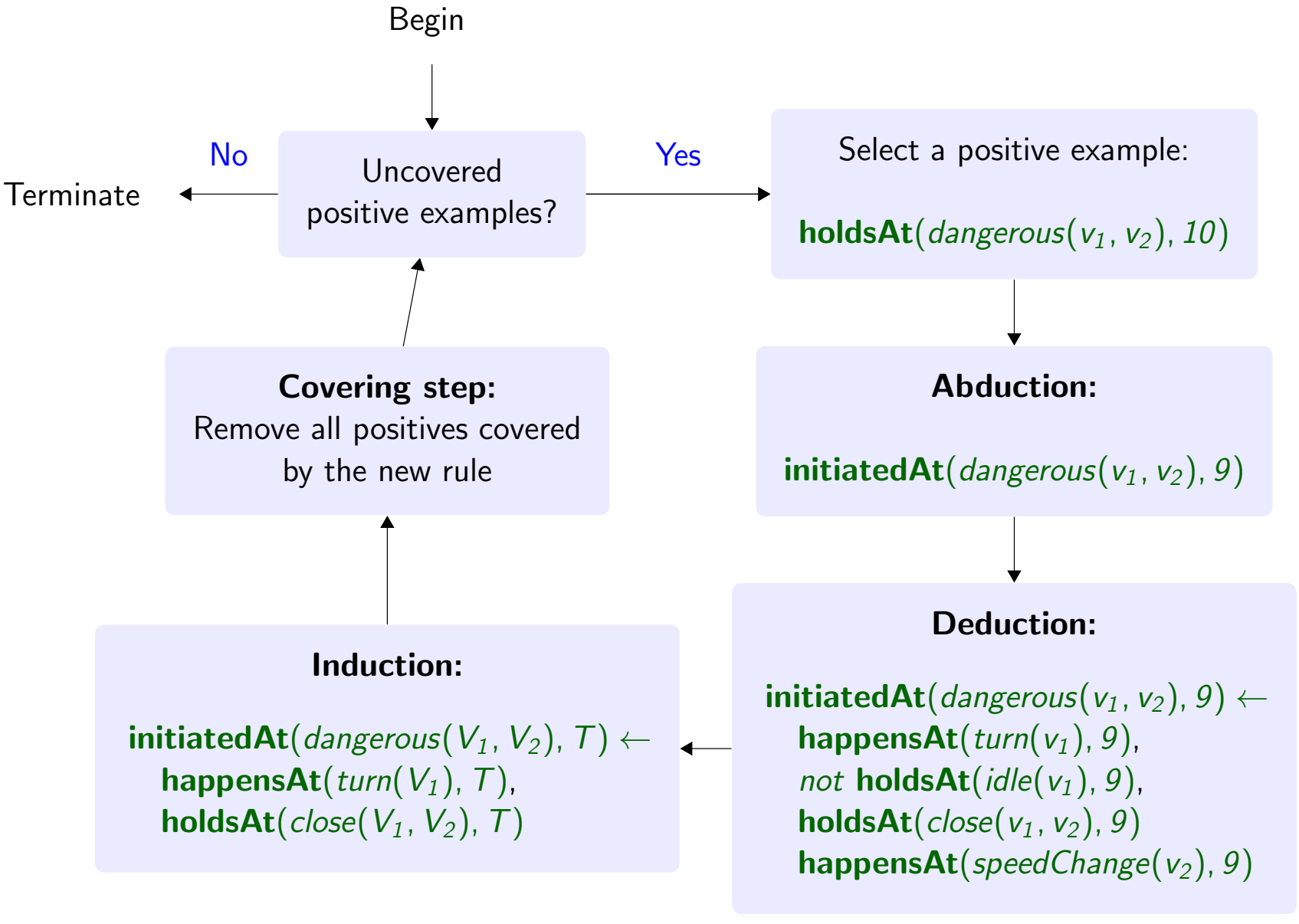
----	$\texttt{EC}_{\texttt{crisp}}$	:	Crisp Event Calculus
—×—	$\texttt{l}-\texttt{CRF}$	:	Linear-chain Conditional Random Field
—✳—	$MLN-EC_{SIT}$	:	Soft-constrained termination inertial rules

Event Recognition

Requirements:

- Efficient reasoning
  - to support real-time decision-making in large-scale, (geographically) distributed applications.
- Reasoning under uncertainty
  - to deal with various types of noise.
- Automated knowledge construction
  - to avoid the time-consuming, error-prone manual CE definition development.

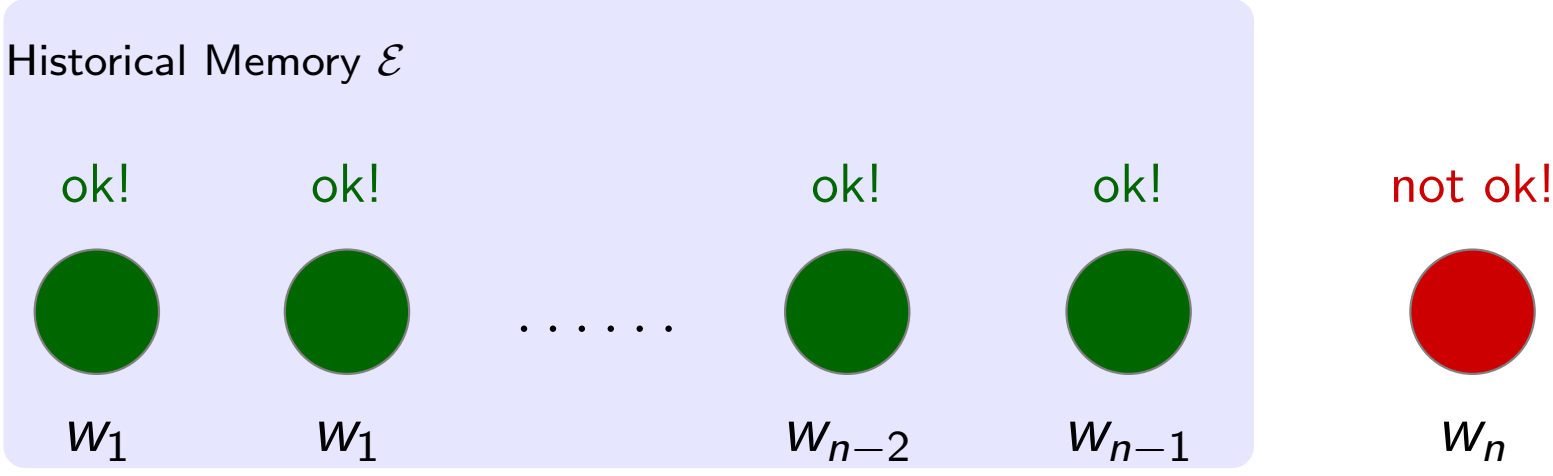
eXtended Hybrid Abductive-Inductive Learning – XHAIL



Incremental Learning

Given:

- A SDE stream $\mathcal{E}$ annotated with CE (historical memory).
- A CE definition H which is correct w.r.t $\mathcal{E}$.
- A new SDE batch in which H is incorrect.



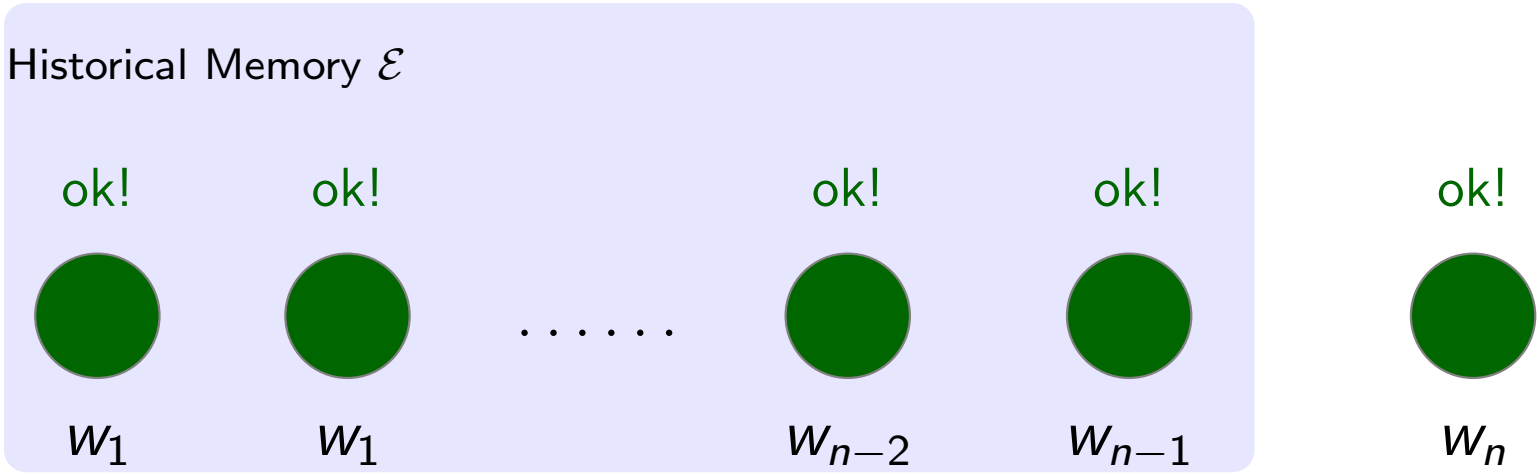
H :

initiatedAt$(dangerous(V_1, V_2), T) \leftarrow$
  happensAt$(turn(V_1), T)$,
  holdsAt$(idle(V_2), T)$

Incremental Learning

Goal:

- Revise H to an H' that is correct w.r.t all examples.



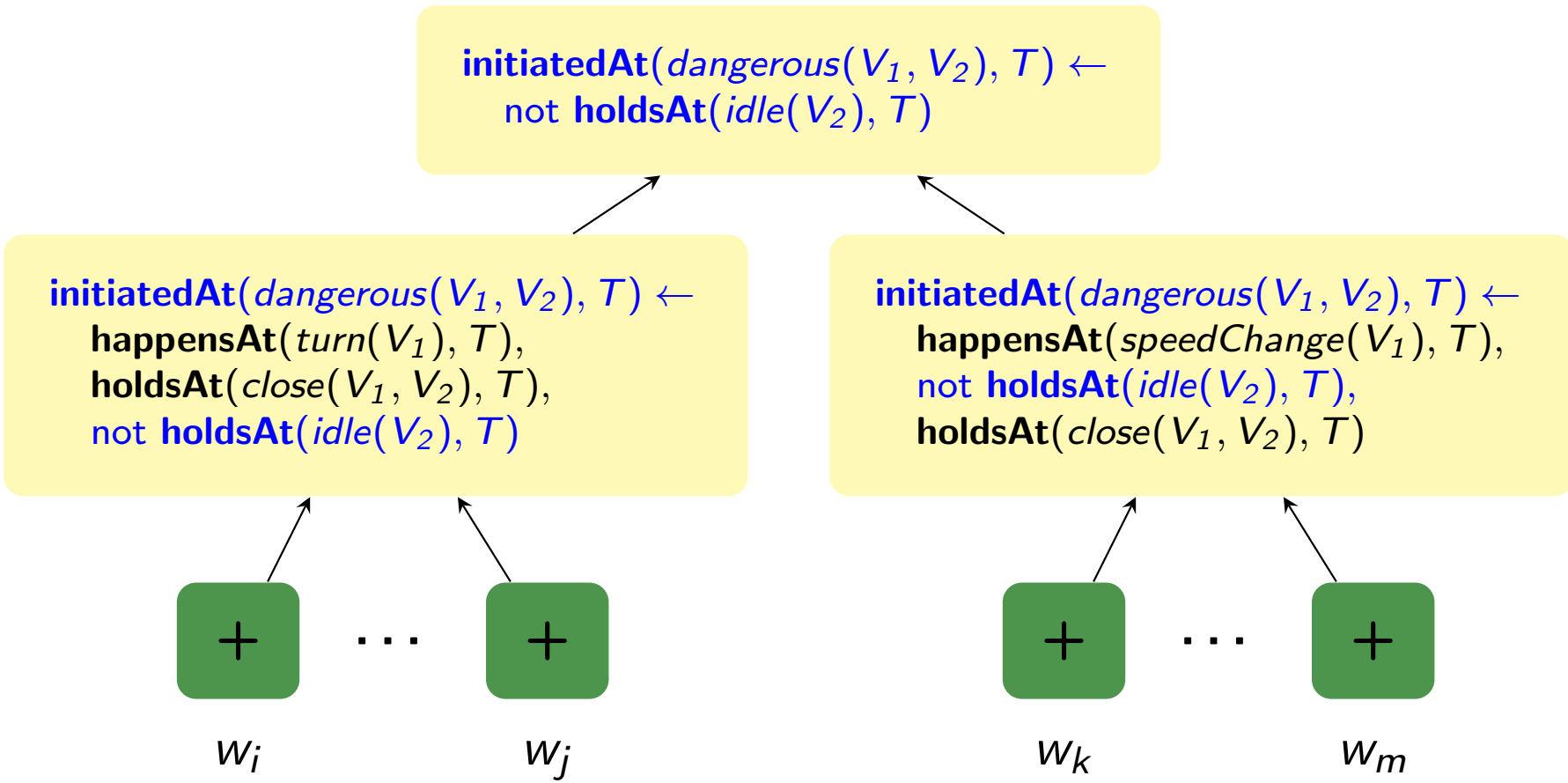
H' :

initiatedAt$(dangerous(V_1, V_2), T) \leftarrow$
  happensAt$(turn(V_1), T)$,
  not **holdsAt**$(idle(V_2), T)$,
  holds$(close(V_1, V_2), T)$

initiatedAt$(dangerous(V_1, V_2), T) \leftarrow$
  happensAt$(speedChange(V_1), T)$,
  not **holdsAt**$(idle(V_2), T)$,
  holdsAt$(close(V_1, V_2), T)$

Efficient Incremental Learning: Support Set

- While constructing a CE definition, summarise the positive examples it covers so far.
- Use this memory for specialisation without having to look back.
  - Reject negatives locally, preserve positives globally.



Evaluation: ILED vs XHAIL

	ILED			XHAIL (1 batch)
Granularity G	$G = 10$	$G = 50$	$G = 100$	$G = 1000$
Training Time (sec)	34.15	23.04	286.74	1560.88
Revisions	11.2	9.1	5.2	–
Hypothesis size	17.82	17.54	17.5	15
Precision	98.713	99.767	99.971	99.973
Recall	99.789	99.845	99.988	99.992

Resources

- Run-Time Event Calculus (RTEC):
  - Alexander Artikis, Marek J. Sergot, Georgios Paliouras: An Event Calculus for Event Recognition. IEEE Transactions on Knowledge & Data Engineering 27(4): 895-908 (2015)
  - **https://github.com/aartikis/RTEC**
- Event Calculus in Markov Login Networks (LoMRF):
  - Anastasios Skarlatidis, Georgios Paliouras, Alexander Artikis, George A. Vouros: Probabilistic Event Calculus for Event Recognition. ACM Transactions on Computational Logic 16(2): 11:1-11:37 (2015)
  - **https://github.com/anskarl/LoMRF**
- Incremental Learning of Event Definitions (ILED):
  - Nikos Katzouris, Alexander Artikis, Georgios Paliouras: Incremental learning of event definitions with Inductive Logic Programming. Machine Learning 100(2-3): 555-585 (2015)
  - **https://github.com/nkatzz/ILED**
- Further info: **http://cer.iit.demokritos.gr**