

Complex Event Recognition with Allen Relations

Periklis Mantenoglou^{2,1} Dimitrios Kelesis^{3,1}
Alexander Artikis^{4,1}

¹NCSR Demokritos, Greece

²National and Kapodistrian University of Athens, Greece

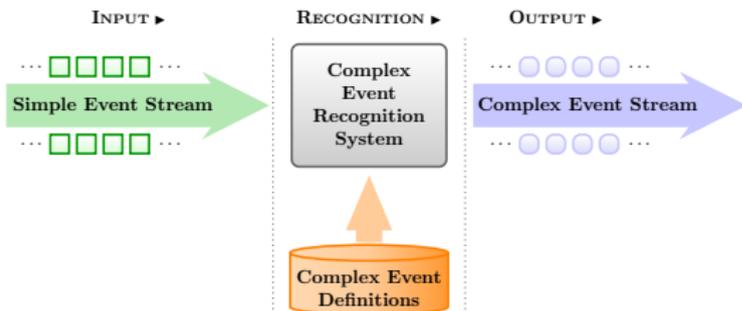
³National Technical University of Athens, Greece

⁴University of Piraeus, Greece

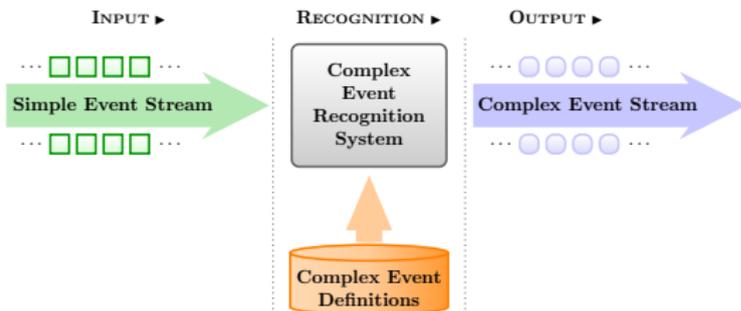
<http://cer.iit.demokritos.gr/>



Complex Event Recognition



Complex Event Recognition



[https://cer.iit.demokritos.gr \(maritime\)](https://cer.iit.demokritos.gr (maritime))

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.

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.

Run-Time Event Calculus (RTEC): Fluent Specification

Simple Fluents:

initiatedAt($F = V, T$) \leftarrow
happensAt(E_{In_I}, T)[,
conditions].

\vdots

terminatedAt($F = V, T$) \leftarrow
happensAt(E_{T_I}, T)[,
conditions].

\vdots

where conditions:

$0-K$ [not] **happensAt**(E_k, T),

$0-M$ [not] **holdsAt**($F_m = V_m, T$),

$0-N$ atemporal-constraint_n

Run-Time Event Calculus (RTEC): Fluent Specification

Simple Fluents:

initiatedAt($F = V, T$) \leftarrow
happensAt($E_{I_{n_1}}, T$)[,
conditions].

\vdots

terminatedAt($F = V, T$) \leftarrow
happensAt(E_{T_1}, T)[,
conditions].

\vdots

where conditions:

$0-K$ [not] **happensAt**(E_k, T),

$0-M$ [not] **holdsAt**($F_m = V_m, T$),

$0-N$ atemporal-constraint_n

Statically Determined Fluents:

holdsFor($F = V, I$) \leftarrow
holdsFor($F_1 = V_1, I_1$)[,
holdsFor($F_2 = V_2, I_2$), ...
holdsFor($F_n = V_n, I_n$),
intervalConstruct(L_1, I_{n+1}), ...
intervalConstruct(L_m, I)].

where intervalConstruct:

union_all or

intersect_all or

relative_complement_all

Statically Determined Fluent: Anchored or Moored

holdsFor(*anchoredOrMoored*(*Vessel*) = true, *I*) ←
 holdsFor(*stopped*(*Vessel*) = *farFromPorts*, *I_{sf}*),
 holdsFor(*withinArea*(*Vessel*, *anchorage*) = true, *I_{wa}*),
 intersect_all([*I_{sf}*, *I_{wa}*], *I_{sa}*),
 holdsFor(*stopped*(*Vessel*) = *nearPorts*, *I_{sn}*),
 union_all([*I_{sa}*, *I_{sn}*], *I*).

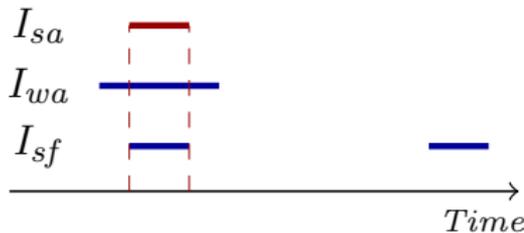
Statically Determined Fluent: Anchored or Moored

holdsFor(*anchoredOrMoored*(*Vessel*) = true, *I*) \leftarrow
holdsFor(*stopped*(*Vessel*) = *farFromPorts*, *I_{sf}*),
holdsFor(*withinArea*(*Vessel*, *anchorage*) = true, *I_{wa}*),
intersect_all([*I_{sf}*, *I_{wa}*], *I_{sa}*),
holdsFor(*stopped*(*Vessel*) = *nearPorts*, *I_{sn}*),
union_all([*I_{sa}*, *I_{sn}*], *I*).



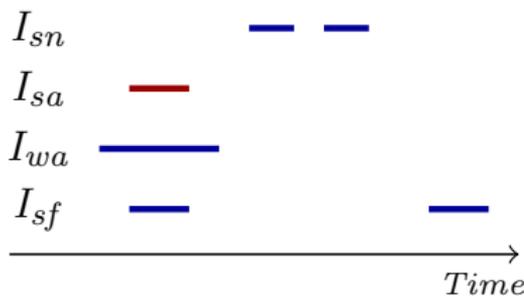
Statically Determined Fluent: Anchored or Moored

holdsFor(*anchoredOrMoored*(*Vessel*) = true, *I*) \leftarrow
holdsFor(*stopped*(*Vessel*) = *farFromPorts*, *I_{sf}*),
holdsFor(*withinArea*(*Vessel*, *anchorage*) = true, *I_{wa}*),
intersect_all([*I_{sf}*, *I_{wa}*], *I_{sa}*),
holdsFor(*stopped*(*Vessel*) = *nearPorts*, *I_{sn}*),
union_all([*I_{sa}*, *I_{sn}*], *I*).



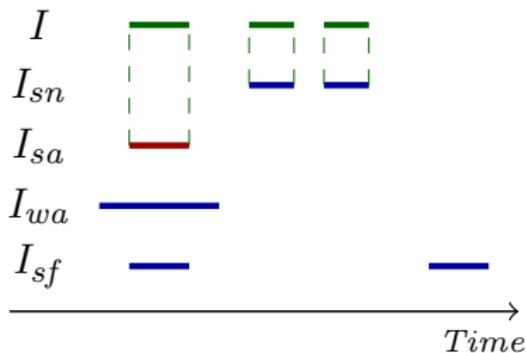
Statically Determined Fluent: Anchored or Moored

holdsFor(*anchoredOrMoored*(*Vessel*) = true, *I*) \leftarrow
holdsFor(*stopped*(*Vessel*) = *farFromPorts*, *I_{sf}*),
holdsFor(*withinArea*(*Vessel*, *anchorage*) = true, *I_{wa}*),
intersect_all([*I_{sf}*, *I_{wa}*], *I_{sa}*),
holdsFor(*stopped*(*Vessel*) = *nearPorts*, *I_{sn}*),
union_all([*I_{sa}*, *I_{sn}*], *I*).



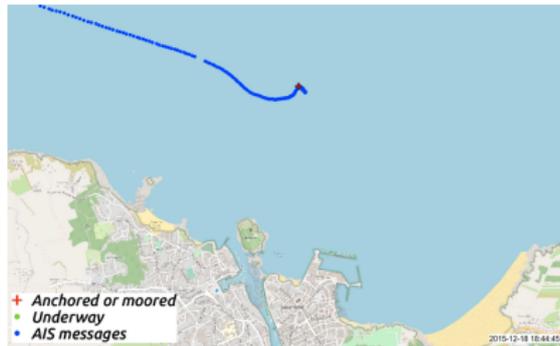
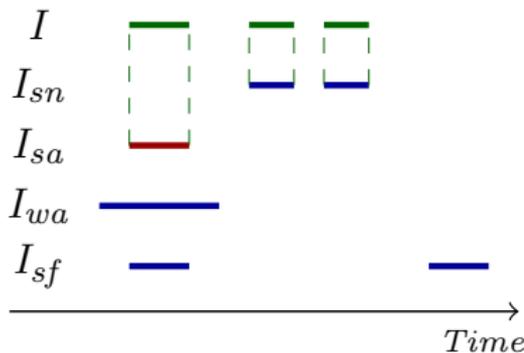
Statically Determined Fluent: Anchored or Moored

holdsFor(*anchoredOrMoored*(*Vessel*) = true, *I*) \leftarrow
holdsFor(*stopped*(*Vessel*) = *farFromPorts*, *I_{sf}*),
holdsFor(*withinArea*(*Vessel*, *anchorage*) = true, *I_{wa}*),
intersect_all([*I_{sf}*, *I_{wa}*], *I_{sa}*),
holdsFor(*stopped*(*Vessel*) = *nearPorts*, *I_{sn}*),
union_all([*I_{sa}*, *I_{sn}*], *I*).



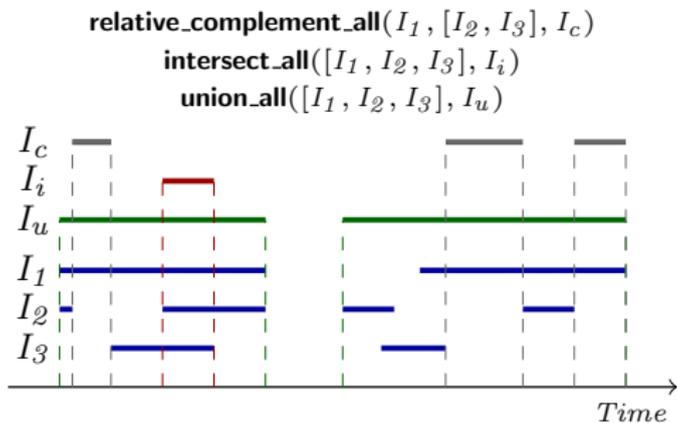
Statically Determined Fluent: Anchored or Moored

holdsFor(*anchoredOrMoored*(Vessel) = true, *I*) \leftarrow
holdsFor(*stopped*(Vessel) = *farFromPorts*, *I_{sf}*),
holdsFor(*withinArea*(Vessel, *anchorage*) = true, *I_{wa}*),
intersect_all([*I_{sf}*, *I_{wa}*], *I_{sa}*),
holdsFor(*stopped*(Vessel) = *nearPorts*, *I_{sn}*),
union_all([*I_{sa}*, *I_{sn}*], *I*).



[https://cer.iit.demokritos.gr \(maritime\)](https://cer.iit.demokritos.gr (maritime))

Interval Constructs & Allen Relations



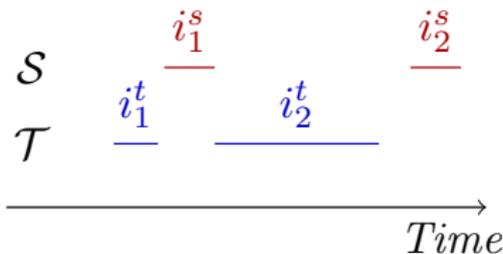
Relation	Illustration
$\text{before}(i^s, i^t)$	
$\text{meets}(i^s, i^t)$	
$\text{starts}(i^s, i^t)$	
$\text{finishes}(i^s, i^t)$	
$\text{during}(i^s, i^t)$	
$\text{overlaps}(i^s, i^t)$	
$\text{equal}(i^s, i^t)$	

RTEC_A: RTEC with Allen Relations

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).

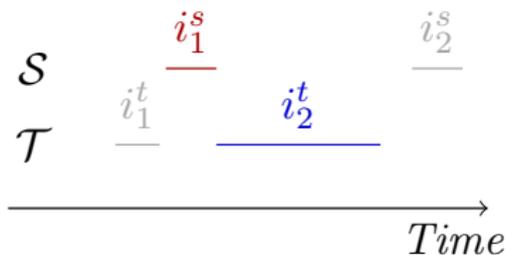
RTEC_A: RTEC with Allen Relations

holdsFor(*disappearedInArea*(Vessel, AreaType) = true, I) ←
holdsFor(*withinArea*(Vessel, AreaType) = true, S),
holdsFor(*gap*(Vessel) = *farFromPorts*, T),
allen(meets, S, T, target, I).



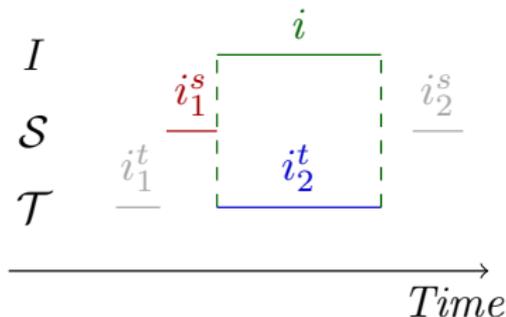
RTEC_A: RTEC with Allen Relations

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, \mathcal{S}),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, \mathcal{T}),
allen(meets, \mathcal{S} , \mathcal{T} , target, *I*).



RTEC_A: RTEC with Allen Relations

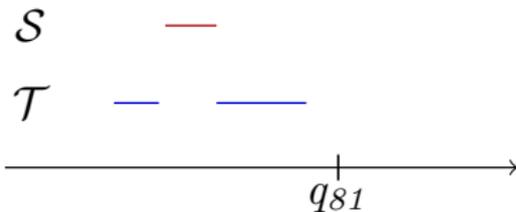
holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).



RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, \mathcal{S}),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, \mathcal{T}),
allen(meets, \mathcal{S} , \mathcal{T} , target, *I*).

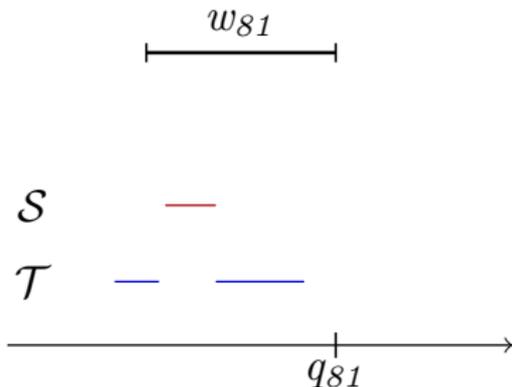
Query time: q_{81}



RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).

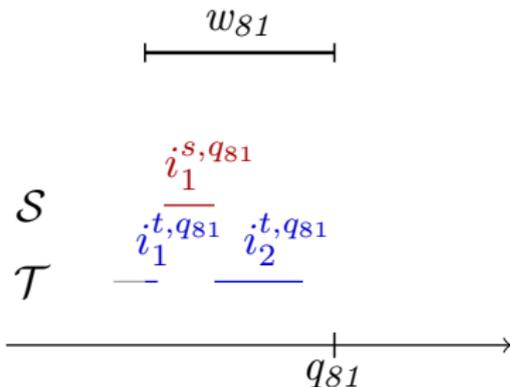
Query time: q_{81}



RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).

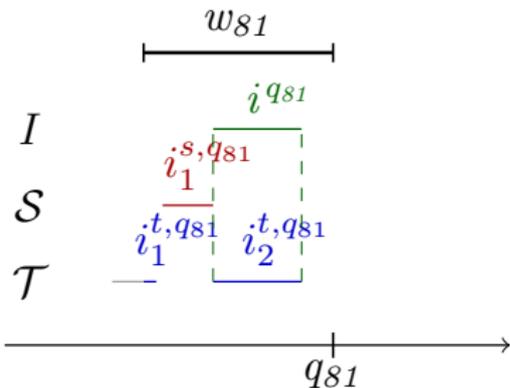
Query time: q_{81}



RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(*meets*, *S*, *T*, *target*, *I*).

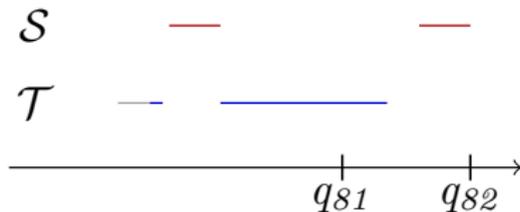
Query time: q_{81}



RTEC_A: Windowing

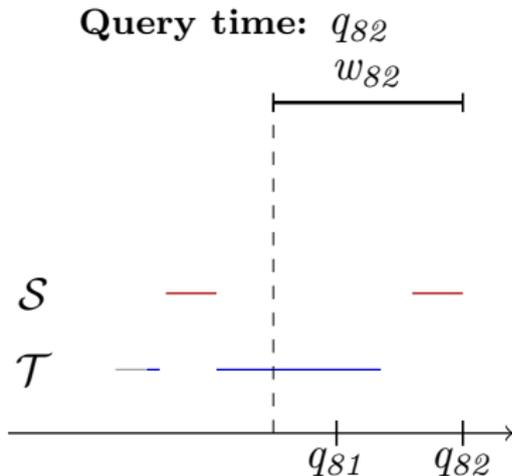
holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).

Query time: q_{82}



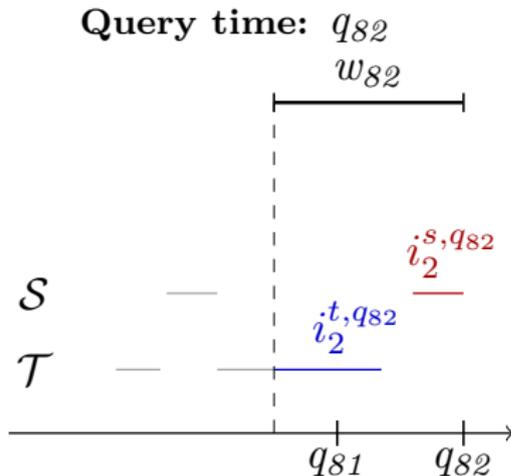
RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).



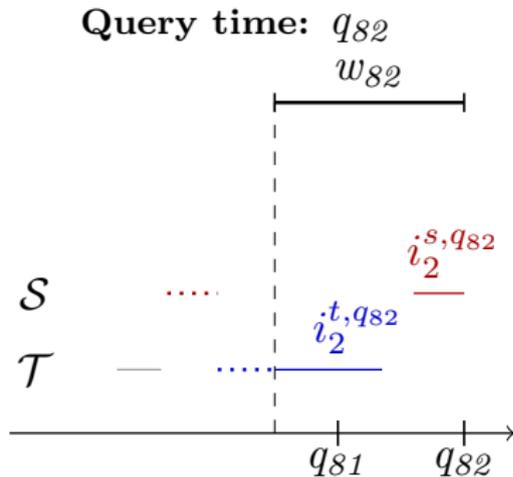
RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).



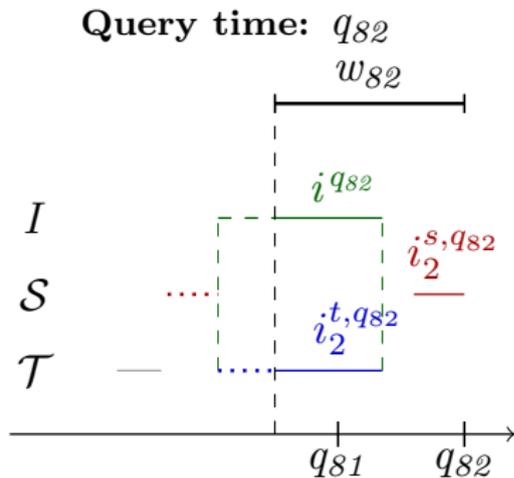
RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).



RTEC_A: Windowing

holdsFor(*disappearedInArea*(*Vessel*, *AreaType*) = true, *I*) ←
holdsFor(*withinArea*(*Vessel*, *AreaType*) = true, *S*),
holdsFor(*gap*(*Vessel*) = *farFromPorts*, *T*),
allen(meets, *S*, *T*, target, *I*).



RTEC_A: Correctness & Complexity

Correctness of RTEC_A

RTEC_A computes all maximal intervals of a fluent defined in terms of an Allen relation, and no other interval.

RTEC_A: Correctness & Complexity

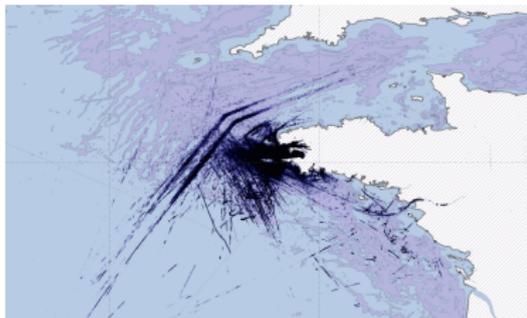
Correctness of RTEC_A

RTEC_A computes all maximal intervals of a fluent defined in terms of an Allen relation, and no other interval.

Complexity of RTEC_A

The cost of computing the maximal intervals of a fluent defined in terms of an Allen relation is $\mathcal{O}(n)$, where n is the number of input intervals.

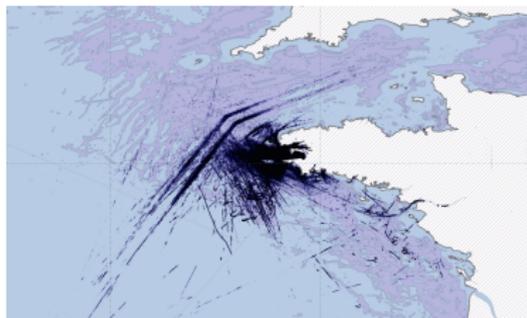
Experimental Evaluation



Code, Data & Temporal Specifications:

- <https://github.com/aartikis/RTEC/tree/allen>

Experimental Evaluation



Window size		Reasoning Time (ms)		Output Intervals	
Days	Input Intervals	RTECA	D ² IA	RTECA	D ² IA
1	19K	40	410	6K	6K
2	37K	65	592	9K	9K
4	74K	99	1.1K	16K	16K
8	148K	156	1.6K	32K	31K
16	297K	285	2.7K	77K	76K

Code, Data & Temporal Specifications:

- <https://github.com/aartikis/RTEC/tree/allen>

Summary & Further Work

RTEC_A:

- An open-source **complex event recognition** framework.
- Support for **Allen relations** in event patterns.
- Correct Allen relation computation with **windowing**.
- **Linear time complexity**.
- **Reproducible** empirical evaluation on large, real data streams.

Summary & Further Work

RTEC_A:

- An open-source **complex event recognition** framework.
- Support for **Allen relations** in event patterns.
- Correct Allen relation computation with **windowing**.
- **Linear time complexity**.
- **Reproducible** empirical evaluation on large, real data streams.

Further Work:

- Support **approximate Allen relations**.
- Contrast Allen relation with **event sequencing operators**.
- Support **events with delayed effects**.

Appendix

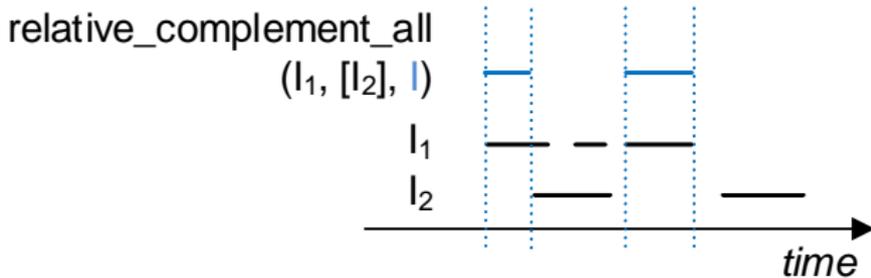
Run-Time Event Calculus (RTEC)

Predicate	Meaning
happensAt (E, T)	Event E occurs at time T
initiatedAt ($F = V, T$)	At time T a period of time for which $F = V$ is initiated
terminatedAt ($F = V, T$)	At time T a period of time for which $F = V$ is terminated
holdsFor ($F = V, I$)	I is the list of the maximal intervals for which $F = V$ holds continuously
holdsAt ($F = V, T$)	The value of fluent F is V at time T
union_all ($[J_1, \dots, J_n], I$)	$I = (J_1 \cup \dots \cup J_n)$
intersect_all ($[J_1, \dots, J_n], I$)	$I = (J_1 \cap \dots \cap J_n)$
relative_complement_all ($I', [J_1, \dots, J_n], I$)	$I = I' \setminus (J_1 \cup \dots \cup J_n)$

Run-Time Event Calculus (RTEC)

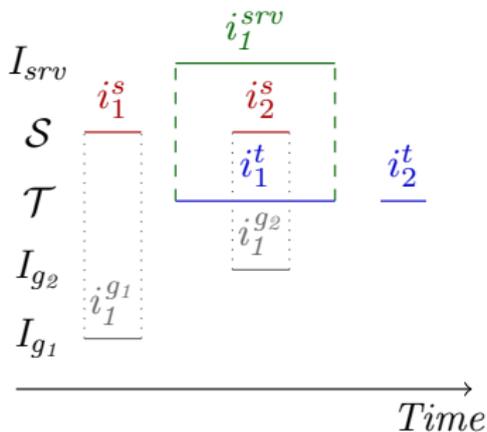
Predicate	Meaning
happensAt (E, T)	Event E occurs at time T
initiatedAt ($F = V, T$)	At time T a period of time for which $F = V$ is initiated
terminatedAt ($F = V, T$)	At time T a period of time for which $F = V$ is terminated
holdsFor ($F = V, I$)	I is the list of the maximal intervals for which $F = V$ holds continuously
holdsAt ($F = V, T$)	The value of fluent F is V at time T
union_all ($[J_1, \dots, J_n], I$)	$I = (J_1 \cup \dots \cup J_n)$
intersect_all ($[J_1, \dots, J_n], I$)	$I = (J_1 \cap \dots \cap J_n)$
relative_complement_all ($I', [J_1, \dots, J_n], I$)	$I = I' \setminus (J_1 \cup \dots \cup J_n)$

Interval Manipulation: Relative Complement



RTEC_A: RTEC with Allen Relations

$\text{holdsFor}(\text{suspiciousRendezVous}(\text{Vessel}_1, \text{Vessel}_2) = \text{true}, I) \leftarrow$
 $\text{holdsFor}(\text{gap}(\text{Vessel}_1) = \text{farFromPorts}, I_{g_1}),$
 $\text{holdsFor}(\text{gap}(\text{Vessel}_2) = \text{farFromPorts}, I_{g_2}),$
 $\text{holdsFor}(\text{proximity}(\text{Vessel}_1, \text{Vessel}_2) = \text{true}, \mathcal{T}),$
 $\text{union_all}([I_{g_1}, I_{g_2}], \mathcal{S}),$
 $\text{allen}(\text{during}, \mathcal{S}, \mathcal{T}, \text{target}, I).$



Experimental Evaluation

Batch setting.

Batch size	Reasoning Time		
	RTEC _A	AEGLE	D ² IA
Input Intervals			
200	1	980	2K
2K	14	4K	6K
20K	154	71.5K	395K
200K	1.8K	MEM	>3.6M

Streaming setting.

Window size		Reasoning Time		Output Interval Pairs	
Days	Input Intervals	RTEC _A	D ² IA	RTEC _A	D ² IA
1	125	1	48	5K	5K
2	250	2	164	19K	18K
4	500	4	568	72K	71K
8	1K	8	1.7K	237K	236K
16	2K	15	7.8K	878K	874K