An Event Calculus for Run-Time Reasoning

Periklis Mantenoglou

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

http://cer.iit.demokritos.gr/
Stream Reasoning

INPUT ▶ RECOGNITION ▶ OUTPUT

Streams of Simple Events

Streams of Complex Events

Stream Reasoning System

Complex Event Definitions

https://cer.iit.demokritos.gr (maritime)
Stream Reasoning

INPUT ▶ RECOGNITION ▶ OUTPUT ▶

Streams of Simple Events

Stream Reasoning System

Complex Event Definitions

Streams of Complex Events

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:

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{I_n_1}, T)[,\text{conditions}].
\]

\[
\vdots
\]

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

\[
\vdots
\]

where conditions:

\[
0^{-K} \text{[not]} \ \text{happensAt}(E_k, T),
\]

\[
0^{-M} \text{[not]} \ \text{holdsAt}(F_m = V_m, T),
\]

\[
0^{-N} \text{atemporal-constraint}_n
\]

Run-Time Event Calculus (RTEC): Fluent Specification

Simple Fluents:

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In_1}, T)[, \text{conditions}].
\]

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

where conditions:

\[
0 - K [\text{not}] \text{happensAt}(E_k, T),
\]

\[
0 - M [\text{not}] \text{holdsAt}(F_m = V_m, T),
\]

\[
0 - N \text{ atemporal-constraint}_n
\]

Statically Determined Fluents:

\[
\text{holdsFor}(F = V, I) \leftarrow \text{holdsFor}(F_1 = V_1, I_1)[, \text{conditions}]. \ldots \text{holdsFor}(F_n = V_n, I_n), \text{intervalOperation}(L_1, I_{n+1}), \ldots \text{intervalOperation}(L_m, I)].
\]

where intervalOperation:

\[
\text{union}_\text{all} \text{ or } \text{intersect}_\text{all} \text{ or } \text{relative}_\text{complement}_\text{all}
\]

Simple Fluent: High Speed Near Coast

\[
\text{initiatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, \ T) \leftarrow \\
\text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}, \_\text{CoG}, \_\text{TrueHeading}), \ T), \\
\text{holdsAt}(\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}, \ T), \\
\text{threshold}(v_{hs}, \ V), \text{Speed} > \ V.
\]
Simple Fluent: High Speed Near Coast

\[\text{initiatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[\text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}, \_\text{CoG}, \_\text{TrueHeading}), T),\]
\[\text{holdsAt}(\text{withinArea}(\text{Vessel, nearCoast}) = \text{true}, T),\]
\[\text{threshold}(v_{hs}, V), \text{Speed} > V.\]

\[\text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[\text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}), T),\]
\[\text{threshold}(v_{hs}, V), \text{Speed} \leq V.\]

\[\text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[\text{happensAt}(\text{Vessel, end}(\text{withinArea}(\text{Vessel, nearCoast}) = \text{true}), T).\]
Simple Fluent: High Speed Near Coast

\[ \text{initiatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[ \text{happensAt}(\text{velocity}(\text{Vessel, Speed, CoG, TrueHeading}), T), \]
\[ \text{holdsAt}(\text{withinArea}(\text{Vessel, nearCoast}) = \text{true}, T), \]
\[ \text{threshold}(\nu_{hs}, V), \text{Speed} > V. \]

\[ \text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[ \text{happensAt}(\text{velocity}(\text{Vessel, Speed}), T), \]
\[ \text{threshold}(\nu_{hs}, V), \text{Speed} \leq V. \]

\[ \text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[ \text{happensAt}(\text{Vessel, end}(\text{withinArea}(\text{Vessel, nearCoast}) = \text{true}), T). \]

Reasoning: \text{holdsFor}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, I)
Simple Fluent: High Speed Near Coast

\[ \text{initiatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[ \text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}, \_\text{CoG}, \_\text{TrueHeading}), T), \]
\[ \text{holdsAt}(\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}, T), \]
\[ \text{threshold}(v_{hs}, V), \text{Speed} > V. \]

\[ \text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[ \text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}), T), \]
\[ \text{threshold}(v_{hs}, V), \text{Speed} \leq V. \]

\[ \text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \]
\[ \text{happensAt}(\text{Vessel}, \text{end}(\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}), T). \]

Reasoning: \[ \text{holdsFor}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, I) \]

\[ \begin{array}{cccccccc}
\text{velocity} & > V & > V & > V & \leq V & > V & > V & \leq V \\
\text{withinArea} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet
\end{array} \]

\[ \text{Time} \]
Simple Fluent: High Speed Near Coast

\begin{align*}
\text{initiatedAt} & (\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt} & (\text{velocity}(\text{Vessel}, \text{Speed}, _{-}\text{CoG}, _{-}\text{TrueHeading}), T), \\
\text{holdsAt} & (\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}, T), \\
& \text{threshold}(v_{hs}, V), \text{Speed} > V.
\end{align*}

\begin{align*}
\text{terminatedAt} & (\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt} & (\text{velocity}(\text{Vessel}, \text{Speed}), T), \\
& \text{threshold}(v_{hs}, V), \text{Speed} \leq V.
\end{align*}

\begin{align*}
\text{terminatedAt} & (\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt} & (\text{Vessel}, \text{end}(\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}), T).
\end{align*}

Reasoning: \text{holdsFor}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, I)
Simple Fluent: High Speed Near Coast

\[
\text{initiatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}, \_\_\_\text{CoG}, \_\_\_\text{TrueHeading}), T), \\
\text{holdsAt}(\text{withinArea}(\text{Vessel, nearCoast}) = \text{true}, T), \\
\text{threshold}(v_{hs}, V), \text{Speed} > V.
\]

\[
\text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt}(\text{velocity}(\text{Vessel, Speed}), T), \\
\text{threshold}(v_{hs}, V), \text{Speed} \leq V.
\]

\[
\text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt}(\text{Vessel, end(\text{withinArea}(\text{Vessel, nearCoast}) = \text{true}), T}).
\]

Reasoning: \text{holdsFor}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, I)

\[
\begin{align*}
\text{highSpeedNC} & \quad > V & > V & > V & \leq V & > V & > V & \leq V \\
\text{velocity} & \quad \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\text{withinArea} & \quad \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\end{align*}
\]
Simple Fluent: High Speed Near Coast

\[
\text{initiatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}, \_\text{CoG}, \_\text{TrueHeading}), T), \\
\text{holdsAt}(\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}, T), \\
\text{threshold}(v_{hs}, V), \text{Speed} > V.
\]

\[
\text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt}(\text{velocity}(\text{Vessel}, \text{Speed}), T), \\
\text{threshold}(v_{hs}, V), \text{Speed} \leq V.
\]

\[
\text{terminatedAt}(\text{highSpeedNC}(\text{Vessel}) = \text{true}, T) \leftarrow \\
\text{happensAt}(\text{Vessel}, \text{end}(\text{withinArea}(\text{Vessel}, \text{nearCoast}) = \text{true}), T).
\]
Statically Determined Fluent: Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ I_{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ I_{wa}), \\
\text{intersect \_all}([I_{sf}, I_{wa}], \ I_{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ I_{sn}), \\
\text{union \_all}([I_{sa}, I_{sn}], \ I).
\]
Statically Determined Fluent: Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ I_{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ I_{wa}), \\
\text{intersect\_all}([I_{sf}, I_{wa}], \ I_{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ I_{sn}), \\
\text{union\_all}([I_{sa}, I_{sn}], \ I). \\
\]
Statically Determined Fluent: Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ I_{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ I_{wa}), \\
\text{intersect\_all}([I_{sf}, I_{wa}], \ I_{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ I_{sn}), \\
\text{union\_all}([I_{sa}, I_{sn}], \ I).
\]
Statically Determined Fluent: Anchored or Moored

\[\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ I_{sf}), \]
\[\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ I_{wa}), \]
\[\text{intersect\_all}([I_{sf}, I_{wa}], \ I_{sa}), \]
\[\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ I_{sn}), \]
\[\text{union\_all}([I_{sa}, I_{sn}], \ I).\]
Statically Determined Fluent: Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, \ I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, \ I_{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, \ I_{wa}), \\
\text{intersect\_all}([I_{sf}, I_{wa}], \ I_{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, \ I_{sn}), \\
\text{union\_all}([I_{sa}, I_{sn}], \ I).
\]
Statically Determined Fluent:
Anchored or Moored

\[
\text{holdsFor}(\text{anchoredOrMoored}(\text{Vessel}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{farFromPorts}, I_{sf}), \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{anchorage}) = \text{true}, I_{wa}), \\
\text{intersect\_all}([I_{sf}, I_{wa}], I_{sa}), \\
\text{holdsFor}(\text{stopped}(\text{Vessel}) = \text{nearPorts}, I_{sn}), \\
\text{union\_all}([I_{sa}, I_{sn}], I).
\]

https://cer.iit.demokritos.gr (maritime)
Maritime Knowledge Base

Maritime Knowledge Base

Maritime Knowledge Base

Semantics

An event description of RTEC is a locally stratified logic program.
Cyclic Dependencies in Temporal Specifications

Semantics

An event description of RTEC with cyclic dependencies is a locally stratified logic program.

Cyclic Dependencies in Temporal Specifications

Semantics

An event description of RTEC with cyclic dependencies is a locally stratified logic program.

Interval Operations & Allen Relations

rel_{\text{ative complement all}}(I_1, [I_2, I_3], I_c)
\text{intersect all}([I_1, I_2, I_3], I_i)
\text{union all}([I_1, I_2, I_3], I_u)

<table>
<thead>
<tr>
<th>Relation</th>
<th>Illustration</th>
</tr>
</thead>
<tbody>
<tr>
<td>before($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
<tr>
<td>meets($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
<tr>
<td>starts($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
<tr>
<td>finishes($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
<tr>
<td>during($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
<tr>
<td>overlaps($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
<tr>
<td>equal($i^s, i^t$)</td>
<td>_ _ $i^s$ _ _ $i^t$</td>
</tr>
</tbody>
</table>

RTEC with Allen Relations

\[\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow\]
\[\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S),\]
\[\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T),\]
\[\text{allen}(\text{meets}, S, T, \text{target}, I).\]
RTEC with Allen Relations

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]

RTEC with Allen Relations

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets, } S, T, \text{target, } I).
\]

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

---

Experimental Setup

Multi-Agent Systems: Voting & NetBill

- Compute, e.g., normative positions of agents.
Experimental Setup

Multi-Agent Systems: Voting & NetBill

- Compute, e.g., normative positions of agents.

Maritime Situational Awareness

- Recognise dangerous, illegal and suspicious vessel activity.
Experimental Setup

Multi-Agent Systems: Voting & NetBill

- Compute, e.g., normative positions of agents.

Maritime Situational Awareness

- Recognise dangerous, illegal and suspicious vessel activity.

Code, Data & Temporal Specifications

https://github.com/aartikis/RTEC
https://github.com/aartikis/RTEC/tree/Allen
Experimental Results

NetBill: monitoring active quotes

![Graph showing reasoning time vs number of events for different systems.](image-url)
Experimental Results

NetBill: monitoring active quotes

Voting: monitoring the status of motions (cycles)
## Experimental Results

### Monitoring maritime activities with Allen relations

<table>
<thead>
<tr>
<th>Window size Intervals</th>
<th>Days</th>
<th>Input Intervals</th>
<th>Reasoning Time (ms)</th>
<th>Output Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RTEC</td>
<td>D^2IA</td>
<td>RTEC</td>
</tr>
<tr>
<td>Days</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>19K</td>
<td>40</td>
<td>410</td>
<td>6K</td>
</tr>
<tr>
<td>2</td>
<td>37K</td>
<td>65</td>
<td>592</td>
<td>9K</td>
</tr>
<tr>
<td>4</td>
<td>74K</td>
<td>99</td>
<td>1.1K</td>
<td>16K</td>
</tr>
<tr>
<td>8</td>
<td>148K</td>
<td>156</td>
<td>1.6K</td>
<td>32K</td>
</tr>
<tr>
<td>16</td>
<td>297K</td>
<td>285</td>
<td>2.7K</td>
<td>77K</td>
</tr>
</tbody>
</table>
Summary & Further Work

RTEC:

- An open-source **stream reasoning** framework.
- **Locally stratified** specifications.
- Efficient treatment of **cyclic dependencies**.
- Support for **Allen relations** in event patterns.
- **Reproducible** empirical evaluation on large data streams.

Further Work:

- Compare expressive power with event sequencing operators.
- Support events with delayed effects.
Summary & Further Work

RTEC:

- An open-source stream reasoning framework.
- Locally stratified specifications.
- Efficient treatment of cyclic dependencies.
- Support for Allen relations in event patterns.
- Reproducible empirical evaluation on large data streams.

Further Work:

- Compare expressive power with event sequencing operators.
- Support events with delayed effects.
Appendix
Run-Time Event Calculus (RTEC)

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

## Run-Time Event Calculus (RTEC)

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

Fluent-Value Pair Computation

Definition:

\[ \text{initiatedAt}(F = V, \ T) \leftarrow \text{happensAt}(E_{ln_1}, \ T), \]
[conditions]

\[ \ldots \]

\[ \text{initiatedAt}(F = V, \ T) \leftarrow \text{happensAt}(E_{ln_i}, \ T), \]
[conditions]

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

\[ \ldots \]

\[ \text{terminatedAt}(F = V, \ T) \leftarrow \text{happensAt}(E_{T_j}, \ T), \]
[conditions]

Reasoning:

\[ 0 \rightarrow \] time
Fluent-Value Pair Computation

Definition:

\[ \text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In_1}, T), \]
\[ \text{conditions} \]

\[ \ldots \]

\[ \text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In_i}, T), \]
\[ \text{conditions} \]

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

\[ \ldots \]

\[ \text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{T_j}, T), \]
\[ \text{conditions} \]

Reasoning:
Fluent-Value Pair Computation

Definition:

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In_1}, T), \quad [\text{conditions}]
\]

\[
\text{initiatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{In_i}, T), \quad [\text{conditions}]
\]

Reasoning:

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

\[
\text{terminatedAt}(F = V, T) \leftarrow \text{happensAt}(E_{T_j}, T), \quad [\text{conditions}]
\]
Fluent-Value Pair Computation

Definition:

\[ \text{initiatedAt}(F = V, \ T) \leftarrow \text{happensAt}(E_{in_i}, \ T), \]  
[conditions]

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

\[ \ldots \]

\[ \text{initiatedAt}(F = V, \ T) \leftarrow \text{happensAt}(E_{in_i}, \ T), \]  
[conditions]

\[ \text{terminatedAt}(F = V, \ T) \leftarrow \text{happensAt}(E_{T_j}, \ T), \]  
[conditions]

Reasoning: \( \text{holdsFor}(F = V, \ I) \)

\[ \text{time} \]

0
RTEC Architecture

Domain-Specific Rules

initiatedAt(rich(X) = true, T) ←
happensAt(win_lottery(X), T).
terminatedAt(rich(X) = true, T) ←
happensAt(lose_wallet(X), T).
initiatedAt(location(X) = Y, T) ←
happensAt(go_to(X, Y), T).
holdsFor(happy(X) = true, I) ←
holdsFor(rich(X) = true, I₁),
holdsFor(location(X) = pub, I₂),
union_all([I₁, I₂], I).

Domain-Specific Declarations

grounding(win_lottery(X)) ← person(X).
grounding(lose_wallet(X)) ← person(X).
grounding(go_to(X, Y)) ← person(X),
place(Y).
grounding(rich(X) = true) ← person(X).
grounding(location(X) = Y) ← person(X),
place(Y).
grounding(happy(X) = true) ← person(X).

Output Entity Stream

holdsFor(location(chris) = home, [(2, 10)])
holdsFor(location(chris) = work, [(10, inf)])
holdsFor(location(chris) = true, [(14, 20)])
holdsFor(location(chris) = work, [(11, 18)])
holdsFor(location(chris) = pub, [(18, inf)])
holdsFor(happy(chris) = true, [(14, inf)])
holdsFor(location(chris) = pub, [(21, 22)])
holdsFor(location(chris) = home, [(22, 30)])
holdsFor(location(chris) = work, [(30, inf)])
holdsFor(happy(chris) = true, [(21, 22)])

Data Stream

\[ \omega_{i-1} \]

| go_to | 11 | chris | home |
| go_to | 9  | chris | work |
| go_to | 9  | chris | pub  |
| go_to | 17 | chris | pub  |
| go_to | 17 | chris | pub  |
| go_to | 19 | chris | pub  |
| go_to | 21 | chris | home |
| go_to | 29 | chris | work |

\[ \omega_i \]

| win_lottery | 13 | chris |
| go_to | 17 | chris | pub  |
| go_to | 17 | chris | pub  |
| go_to | 21 | chris | home |
| go_to | 29 | chris | work |

\[ \omega_{i+1} \]

| go_to | 21 | chris | home |
| go_to | 29 | chris | work |

...
RTEC: Windowing
RTEC: Windowing
RTEC: Windowing
Cyclic Dependencies in Temporal Specifications

\[
\begin{align*}
\text{initiatedAt}(\text{status}(M) = \text{proposed}, T) & \leftarrow \\
\text{happensAt}(\text{propose}(P, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{null}, T). \\
\text{initiatedAt}(\text{status}(M) = \text{voting}, T) & \leftarrow \\
\text{happensAt}(\text{second}(S, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{proposed}, T). \\
\text{initiatedAt}(\text{status}(M) = \text{voted}, T) & \leftarrow \\
\text{happensAt}(\text{close\_ballot}(C, M), T), \\
\text{holdsAt}(\text{status}(M) = \text{voting}, T). \\
\text{initiatedAt}(\text{status}(M) = \text{null}, T) & \leftarrow \\
\text{happensAt}(\text{declare}(C, M, \text{Res}), T), \\
\text{holdsAt}(\text{status}(M) = \text{voted}, T).
\end{align*}
\]
**RTEC\textsubscript{A}: Windowing**

\[\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \text{allen}(\text{meets}, S, T, \text{target}, I).\]

**Query time:** $q_{81}$
RTEC_A: Windowing

holdsFor\(\text{disappearedInArea}(\text{Vessel}, \text{Area Type}) = \text{true}, I) \leftarrow \)
holdsFor\(\text{withinArea}(\text{Vessel}, \text{Area Type}) = \text{true}, S), \)
holdsFor\(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \)
allen(meets, S, T, target, I).

Query time: \(q_{81}\)

\[ w_{81} \]

\[ S \quad \quad \]

\[ T \quad \quad \quad \]

\[ q_{81} \]
RTEC_A: Windowing

\[
\text{holdsFor}(\text{disappearedInArea}(Vessel, AreaType) = \text{true}, I) \leftarrow \text{holdsFor}(\text{withinArea}(Vessel, AreaType) = \text{true}, S), \text{holdsFor}(\text{gap}(Vessel) = \text{farFromPorts}, T), \text{allen}(\text{meets}, S, T, \text{target}, I).
\]

Query time: \(q_{81}\)

\[
\begin{array}{c}
\text{Query time: } q_{81} \\
\begin{array}{c}
 S \\
 i_1^{s, q_{81}} \quad i_1 \\
 i_1^{t, q_{81}} \quad \underline{i_1^{t, q_{81}}} \quad i_2^{t, q_{81}} \quad \underline{i_2^{t, q_{81}}}
\end{array}
\end{array}
\]
RTEC\textsubscript{A}: Windowing

\texttt{holdsFor}(\texttt{disappearedInArea}(\texttt{Vessel}, \texttt{Area Type}) = \text{true}, I) \leftarrow \texttt{holdsFor}(\texttt{withinArea}(\texttt{Vessel}, \texttt{Area Type}) = \text{true}, S), \texttt{holdsFor}(\texttt{gap}(\texttt{Vessel}) = \texttt{farFromPorts}, T), \texttt{allen}(\texttt{meets}, S, T, \text{target}, I).

\textbf{Query time: } q_{81}
RTECₐ: Windowing

\[ \text{holdsFor}(\text{disappearedInArea}(Vessel, \text{Area Type}) = \text{true}, I) \leftarrow \]
\[ \text{holdsFor}(\text{withinArea}(Vessel, \text{Area Type}) = \text{true}, S), \]
\[ \text{holdsFor}(\text{gap}(Vessel) = \text{farFromPorts}, T), \]
\[ \text{allen}(\text{meets}, S, T, \text{target}, I). \]

**Query time:** \( q_{82} \)
\textbf{RTEC}_{A}: Windowing

\begin{align*}
\text{holdsFor}(\text{disappearedInArea}(Vessel, AreaType) = \text{true}, l) & \leftarrow \\
\text{holdsFor}(\text{withinArea}(Vessel, AreaType) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(Vessel) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, l).
\end{align*}

\textbf{Query time: } q_{82}

\begin{tikzpicture}[scale=0.5]
\draw[->] (0,0) -- (5,0) node[midway,right]{$q_{81}$};
\draw[->] (0,-2) -- (5,-2) node[midway,right]{$q_{82}$};
\draw[dashed] (0,-2) -- (0,0);
\draw[blue] (2,-2) -- (2,0);
\draw[red] (0,-2) -- (2,-2);
\draw[red] (3,-2) -- (3,0);
\draw[blue] (4,-2) -- (4,0);
\end{tikzpicture}
RTEC_A: Windowing

\[ \text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{AreaType}) = \text{true}, I) \leftarrow \]
\[ \text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{AreaType}) = \text{true}, S), \]
\[ \text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \]
\[ \text{allen}(\text{meets}, S, T, \text{target}, I). \]

Query time: \( q_{82} \)
\textbf{RTEC}_A: \textit{Windowing}

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

\begin{center}
Query time: \(q_{82}\)
\end{center}
RTEC_A: Windowing

\[
\text{holdsFor}(\text{disappearedInArea}(\text{Vessel}, \text{Area\ Type}) = \text{true}, I) \leftarrow \\
\text{holdsFor}(\text{withinArea}(\text{Vessel}, \text{Area\ Type}) = \text{true}, S), \\
\text{holdsFor}(\text{gap}(\text{Vessel}) = \text{farFromPorts}, T), \\
\text{allen}(\text{meets}, S, T, \text{target}, I).
\]
RTEC\textsubscript{A}: Correctness & Complexity

**Correctness of RTEC\textsubscript{A}**

RTEC\textsubscript{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 $O(n)$, where $n$ is the number of input intervals.
Interval Manipulation: Relative Complement

relative_complement_all
(I₁, [I₂], I)

I₁
I₂

time
\textbf{RTEC}_A: RTEC with Allen Relations

\texttt{holdsFor(suspiciousRendezVous(Vessel}_1, Vessel}_2) = true, I) \leftarrow
\texttt{holdsFor(gap(Vessel}_1) = farFromPorts, I_{g_1}),
\texttt{holdsFor(gap(Vessel}_2) = farFromPorts, I_{g_2}),
\texttt{holdsFor(proximity(Vessel}_1, Vessel}_2) = true, T),
\texttt{union\_all([I_{g_1}, I_{g_2}], S),}
\texttt{allen(during, S, T, target, I)).}
## Experimental Evaluation

### Batch setting.

<table>
<thead>
<tr>
<th>Batch size</th>
<th>Reasoning Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Intervals</td>
<td>RTECA</td>
</tr>
<tr>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>2K</td>
<td>14</td>
</tr>
<tr>
<td>20K</td>
<td>154</td>
</tr>
<tr>
<td>200K</td>
<td>1.8K</td>
</tr>
</tbody>
</table>

### Streaming setting.

<table>
<thead>
<tr>
<th>Window size</th>
<th>Reasoning Time</th>
<th>Output Interval Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Days</td>
<td>Input Intervals</td>
<td>RTECA</td>
</tr>
<tr>
<td>1</td>
<td>125</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>250</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>1K</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>2K</td>
<td>15</td>
</tr>
</tbody>
</table>