Stream Reasoning with Cycles*

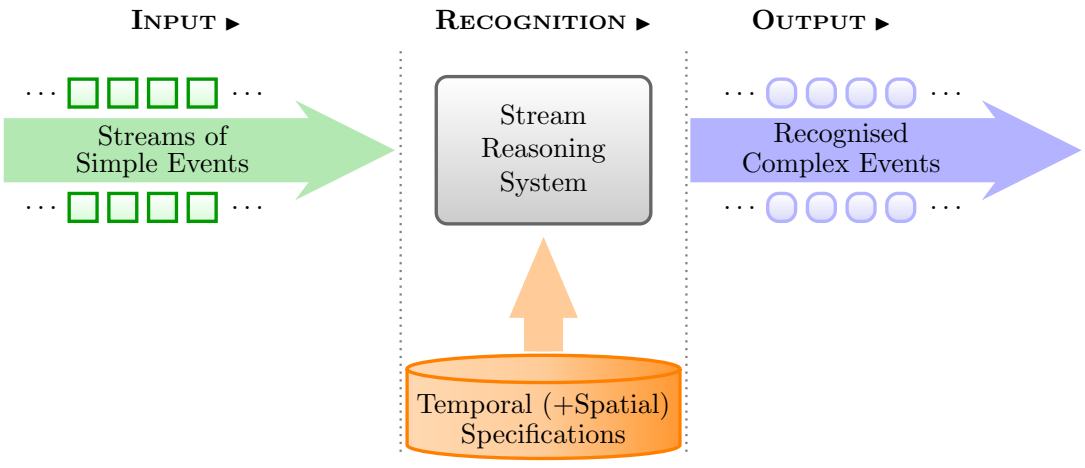
Periklis Mantenoglou[2,1] Manolis Pitsikalis[4]
Alexander Artikis[3,1]

[1]NCSR Demokritos, Greece
[2]National and Kapodistrian University of Athens, Greece
[3]University of Piraeus, Greece
[4]University of Liverpool, UK

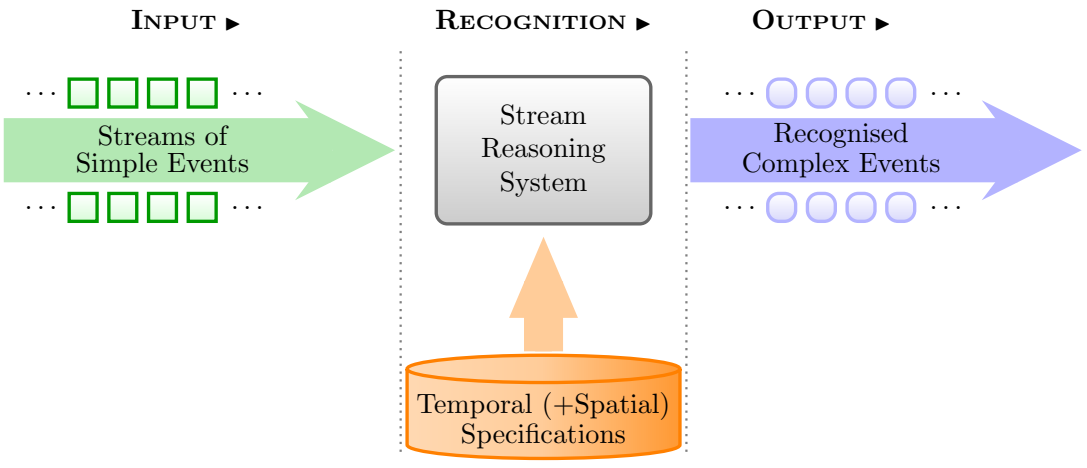
`http://cer.iit.demokritos.gr/`

*Originally presented at KR 2022.

Stream Reasoning



Stream Reasoning



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.

Robert A. Kowalski, Marek J. Sergot: A Logic-based Calculus of Events. New Gener. Comput. 4(1): 67-95 (1986)

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.

Robert A. Kowalski, Marek J. Sergot: A Logic-based Calculus of Events. New Gener. Comput. 4(1): 67-95 (1986)

Run-Time Event Calculus

Time in RTEC is a **linear** and **discrete** structure.

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

Artikis A., Sergot M. and Paliouras G., An Event Calculus for Event Recognition. In IEEE Transactions on Knowledge and Data Engineering (TKDE), 27(4), 895–908, 2015.

Fluent-Value Pair Specification

Definition:

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

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

where

$$\text{conditions:} \quad \begin{array}{l} {}^{0-K}\ [\text{not}]\ \textbf{happensAt}(E_k,\ T), \\ {}^{0-M}\ [\text{not}]\ \textbf{holdsAt}(F_m = V_m,\ T), \\ {}^{0-N}\ \text{atemporal-constraint}_n \end{array}$$

Fluent-Value Pair Computation

Definition:

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

...

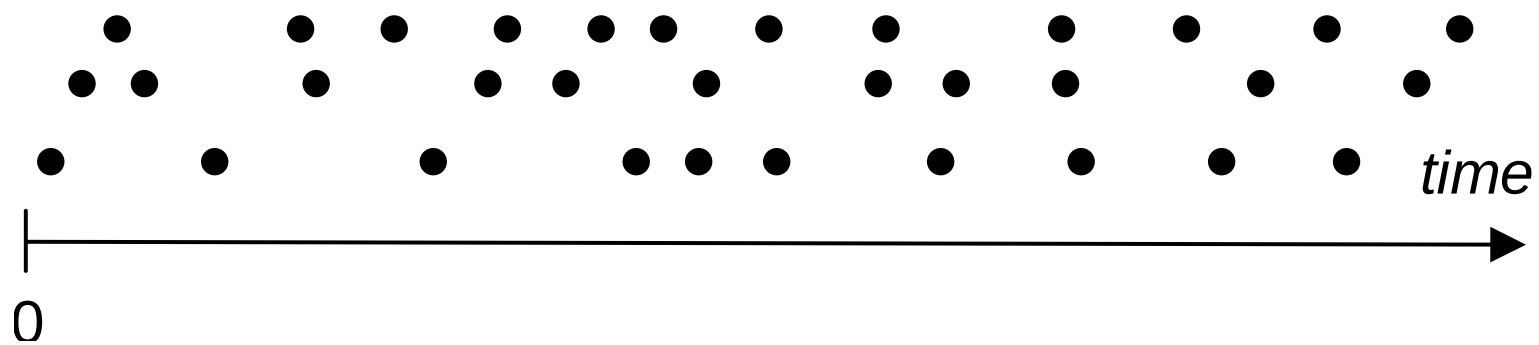
initiatedAt$(F = V,\ T) \leftarrow$
  happensAt$(E_{In_i},\ T)$,
  [conditions]

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

...

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

Reasoning:



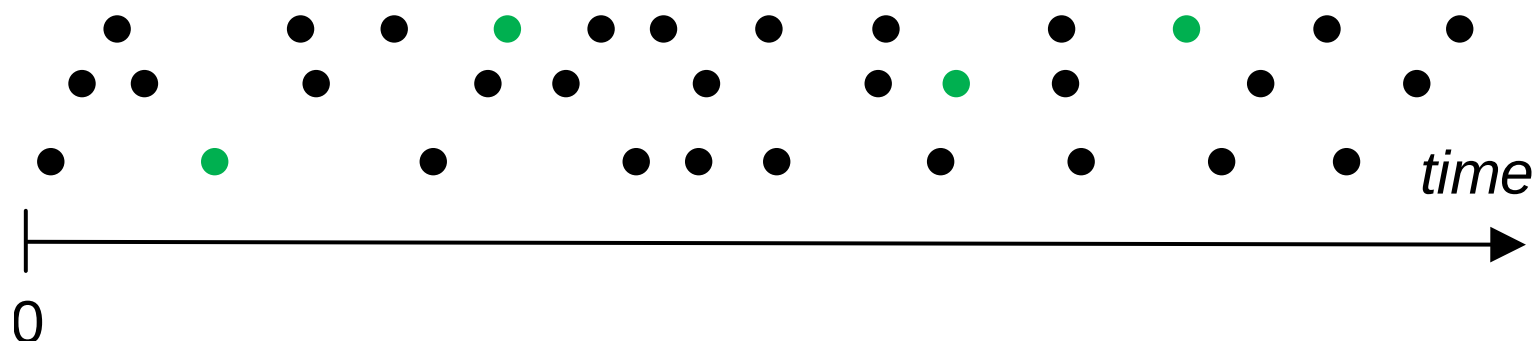
Fluent-Value Pair Computation

Definition:

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

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

Reasoning:



Fluent-Value Pair Computation

Definition:

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

$$\ldots$$

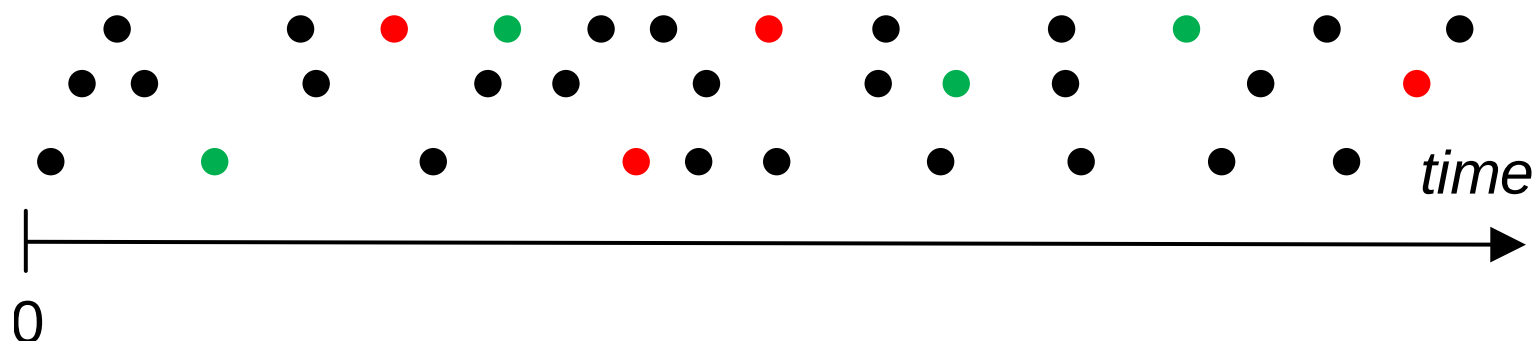
$$\textbf{initiatedAt}(F = V,\ T) \leftarrow \textbf{happensAt}(E_{In_i},\ T), [\text{conditions}]$$

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

$$\ldots$$

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

Reasoning:



Fluent-Value Pair Computation

Definition:

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

$$\ldots$$

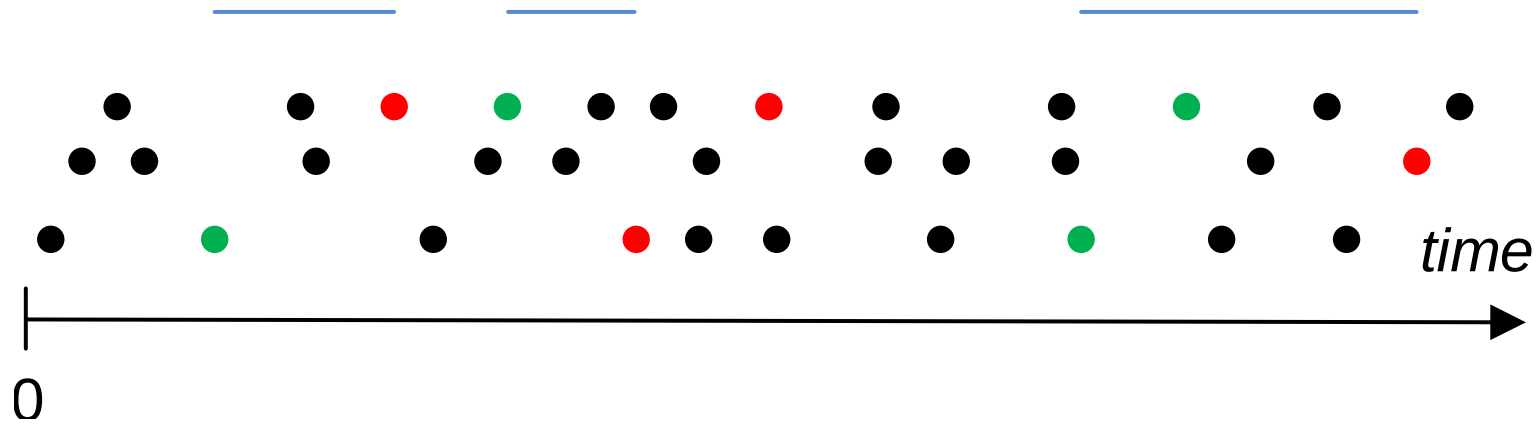
$$\textbf{initiatedAt}(F = V,\ T) \leftarrow \textbf{happensAt}(E_{In_i},\ T), [\text{conditions}]$$

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

$$\ldots$$

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

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



Stratification

Definition

A logic program P is called stratified when it is made up of several distinct strata P_i i.e.,

$$P = P_0 \cup P_1 \cdots P_n$$

where:

- P_0 contains only definite clauses and,
- for $i > 0$ any clauses with negative literals in P_i may only refer to predicates of P_j where $j < i$.

Stratification

Definition

A logic program P is called stratified when it is made up of several distinct strata P_i i.e.,

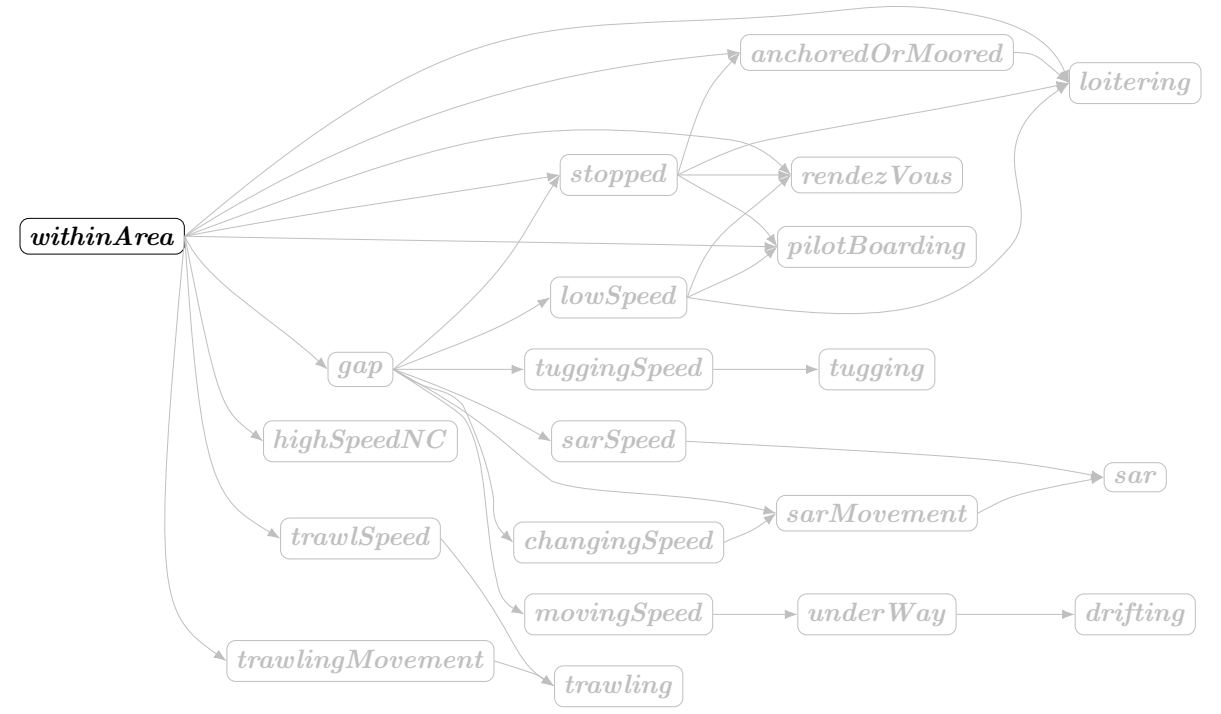
$$P = P_0 \cup P_1 \cdots P_n$$

where:

- P_0 contains only definite clauses and,
- for $i > 0$ any clauses with negative literals in P_i may only refer to predicates of P_j where $j < i$.

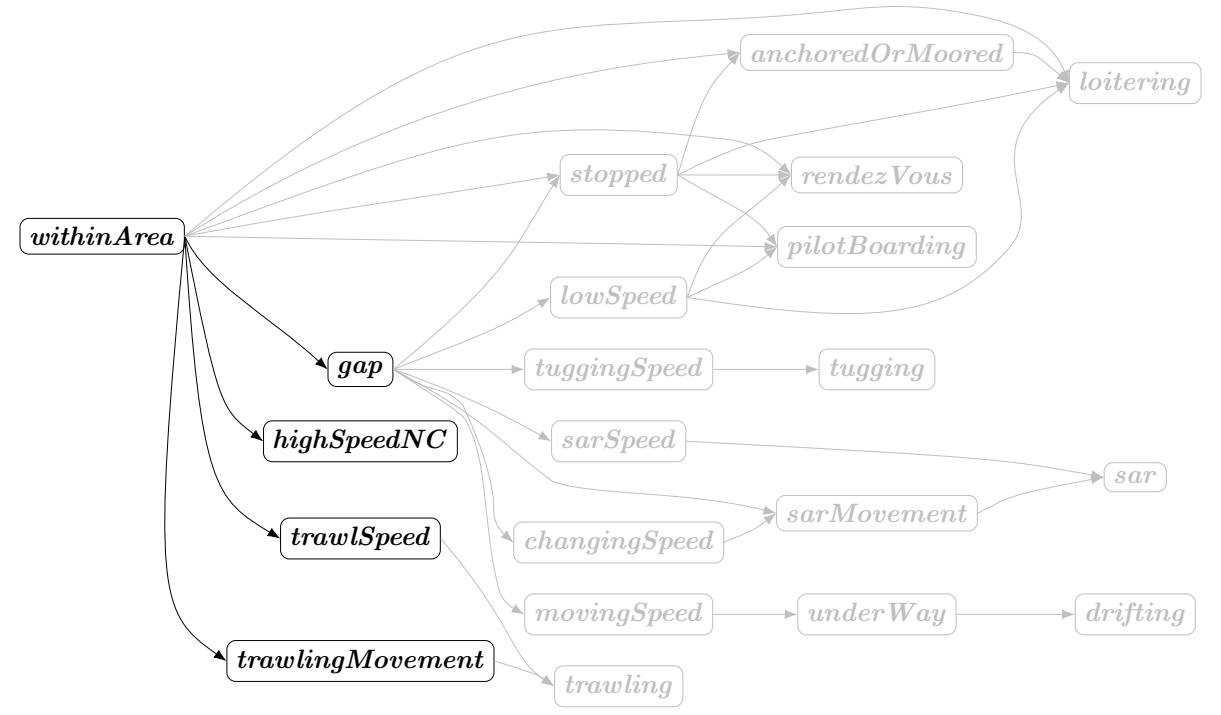
An acyclic event description in RTEC is a locally stratified logic program.

Stratification in RTEC

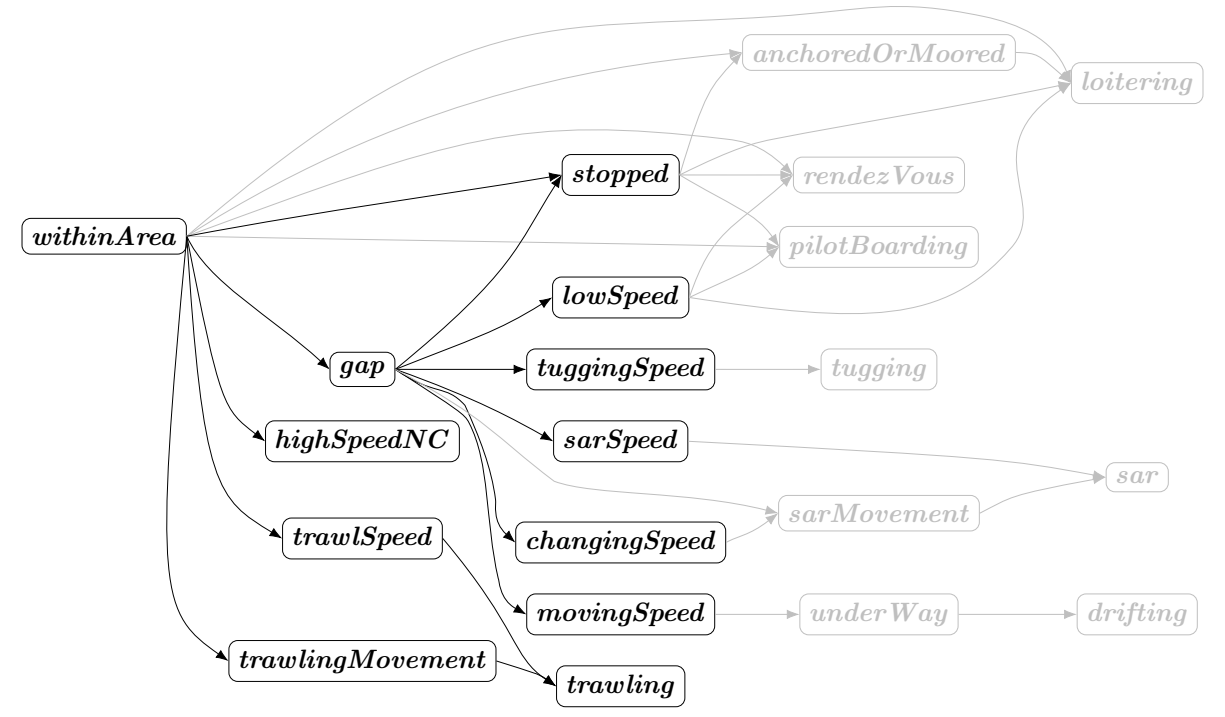


P_1 P_2 P_3 P_4 P_5

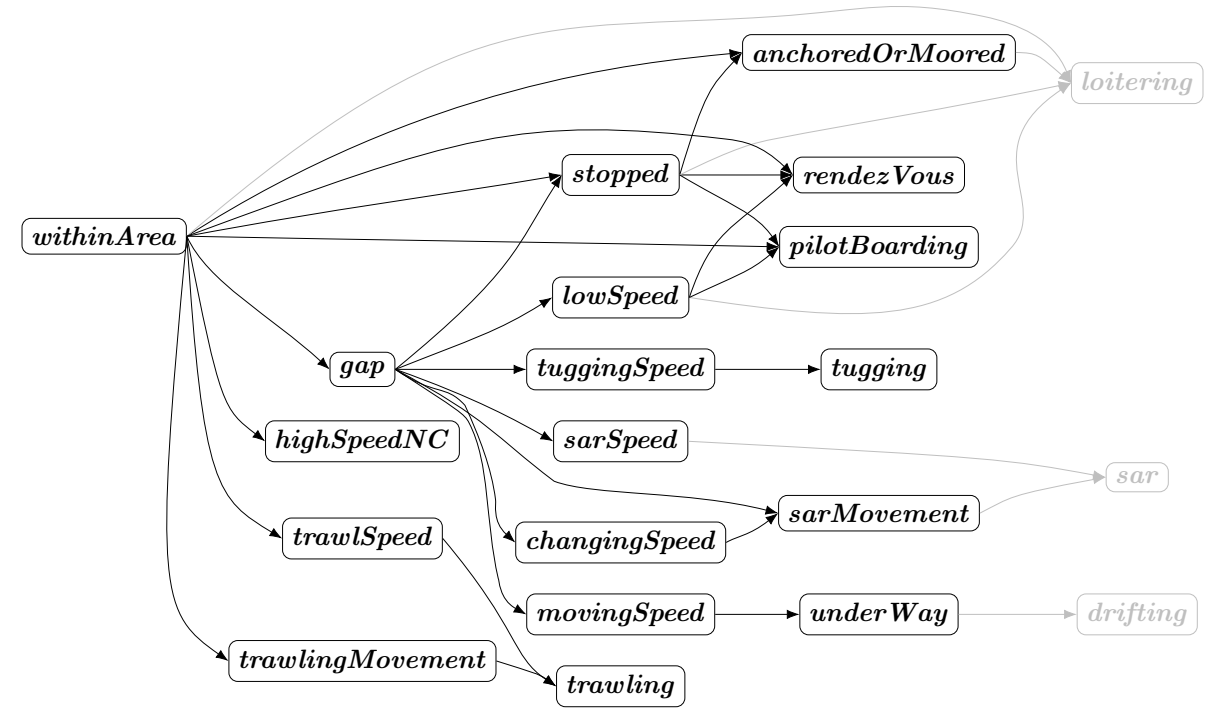
Stratification in RTEC



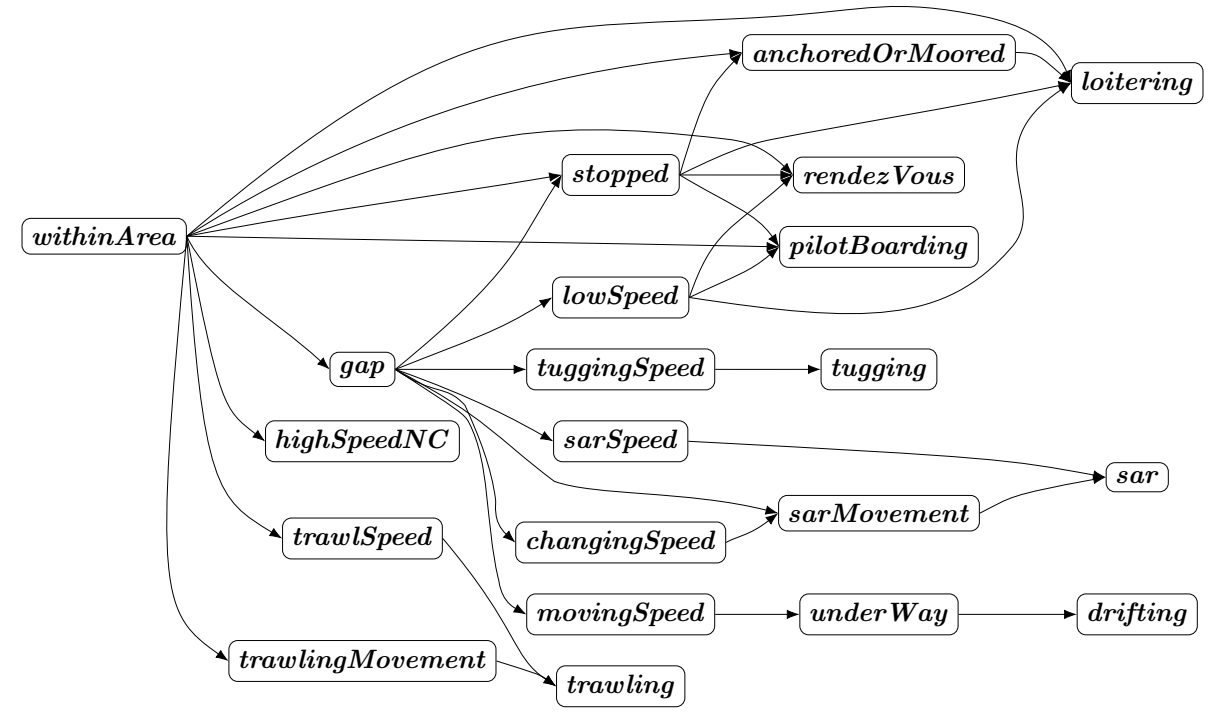
Stratification in RTEC



Stratification in RTEC



Stratification in RTEC



P_1 P_2 P_3 P_4 P_5

Cyclic Dependencies in Temporal Specifications

$$\textbf{initiatedAt}(\mathit{status}(M) = \mathit{proposed}, T) \leftarrow$$
$$\quad \textbf{happensAt}(\mathit{propose}(P, M), T),$$
$$\quad \textbf{holdsAt}(\mathit{status}(M) = \mathit{null}, T).$$

Cyclic Dependencies in Temporal Specifications

$$\textbf{initiatedAt}(\mathit{status}(M) = \mathit{proposed}, T) \leftarrow$$
$$\quad \textbf{happensAt}(\mathit{propose}(P, M), T),$$
$$\quad \textbf{holdsAt}(\mathit{status}(M) = \mathit{null}, T).$$
$$\textbf{initiatedAt}(\mathit{status}(M) = \mathit{voting}, T) \leftarrow$$
$$\quad \textbf{happensAt}(\mathit{second}(S, M), T),$$
$$\quad \textbf{holdsAt}(\mathit{status}(M) = \mathit{proposed}, T).$$

Cyclic Dependencies in Temporal Specifications

$$\textbf{initiatedAt}(\mathit{status}(M) = \mathit{proposed}, T) \leftarrow$$
$$\quad \textbf{happensAt}(\mathit{propose}(P, M), T),$$
$$\quad \textbf{holdsAt}(\mathit{status}(M) = \mathit{null}, T).$$
$$\textbf{initiatedAt}(\mathit{status}(M) = \mathit{voting}, T) \leftarrow$$
$$\quad \textbf{happensAt}(\mathit{second}(S, M), T),$$
$$\quad \textbf{holdsAt}(\mathit{status}(M) = \mathit{proposed}, T).$$
$$\textbf{initiatedAt}(\mathit{status}(M) = \mathit{voted}, T) \leftarrow$$
$$\quad \textbf{happensAt}(\mathit{close\_ballot}(C, M), T),$$
$$\quad \textbf{holdsAt}(\mathit{status}(M) = \mathit{voting}, T).$$

Cyclic Dependencies in Temporal Specifications

$$\textbf{initiatedAt}(status(M) = proposed, T) \leftarrow$$
$$\quad \textbf{happensAt}(propose(P, M), T),$$
$$\quad \textbf{holdsAt}(status(M) = null, T).$$
$$\textbf{initiatedAt}(status(M) = voting, T) \leftarrow$$
$$\quad \textbf{happensAt}(second(S, M), T),$$
$$\quad \textbf{holdsAt}(status(M) = proposed, T).$$
$$\textbf{initiatedAt}(status(M) = voted, T) \leftarrow$$
$$\quad \textbf{happensAt}(close\_ballot(C, M), T),$$
$$\quad \textbf{holdsAt}(status(M) = voting, T).$$
$$\textbf{initiatedAt}(status(M) = null, T) \leftarrow$$
$$\quad \textbf{happensAt}(declare(C, M, Res), T),$$
$$\quad \textbf{holdsAt}(status(M) = voted, T).$$

Cyclic Dependencies in Temporal Specifications

initiatedAt$(status(M) = proposed, T) \leftarrow$
  happensAt$(propose(P, M), T)$,
  holdsAt$(status(M) = null, T)$.

initiatedAt$(status(M) = voting, T) \leftarrow$
  happensAt$(second(S, M), T)$,
  holdsAt$(status(M) = proposed, T)$.

initiatedAt$(status(M) = voted, T) \leftarrow$
  happensAt$(close\_ballot(C, M), T)$,
  holdsAt$(status(M) = voting, T)$.

initiatedAt$(status(M) = null, T) \leftarrow$
  happensAt$(declare(C, M, Res), T)$,
  holdsAt$(status(M) = voted, T)$.

Cyclic Dependencies in Temporal Specifications

initiatedAt$(status(M) = proposed, T) \leftarrow$
  happensAt$(propose(P, M), T)$,
  holdsAt$(status(M) = null, T)$.

initiatedAt$(status(M) = voting, T) \leftarrow$
  happensAt$(second(S, M), T)$,
  holdsAt$(status(M) = proposed, T)$.

initiatedAt$(status(M) = voted, T) \leftarrow$
  happensAt$(close\_ballot(C, M), T)$,
  holdsAt$(status(M) = voting, T)$.

initiatedAt$(status(M) = null, T) \leftarrow$
  happensAt$(declare(C, M, Res), T)$,
  holdsAt$(status(M) = voted, T)$.

Cyclic Dependencies in Temporal Specifications

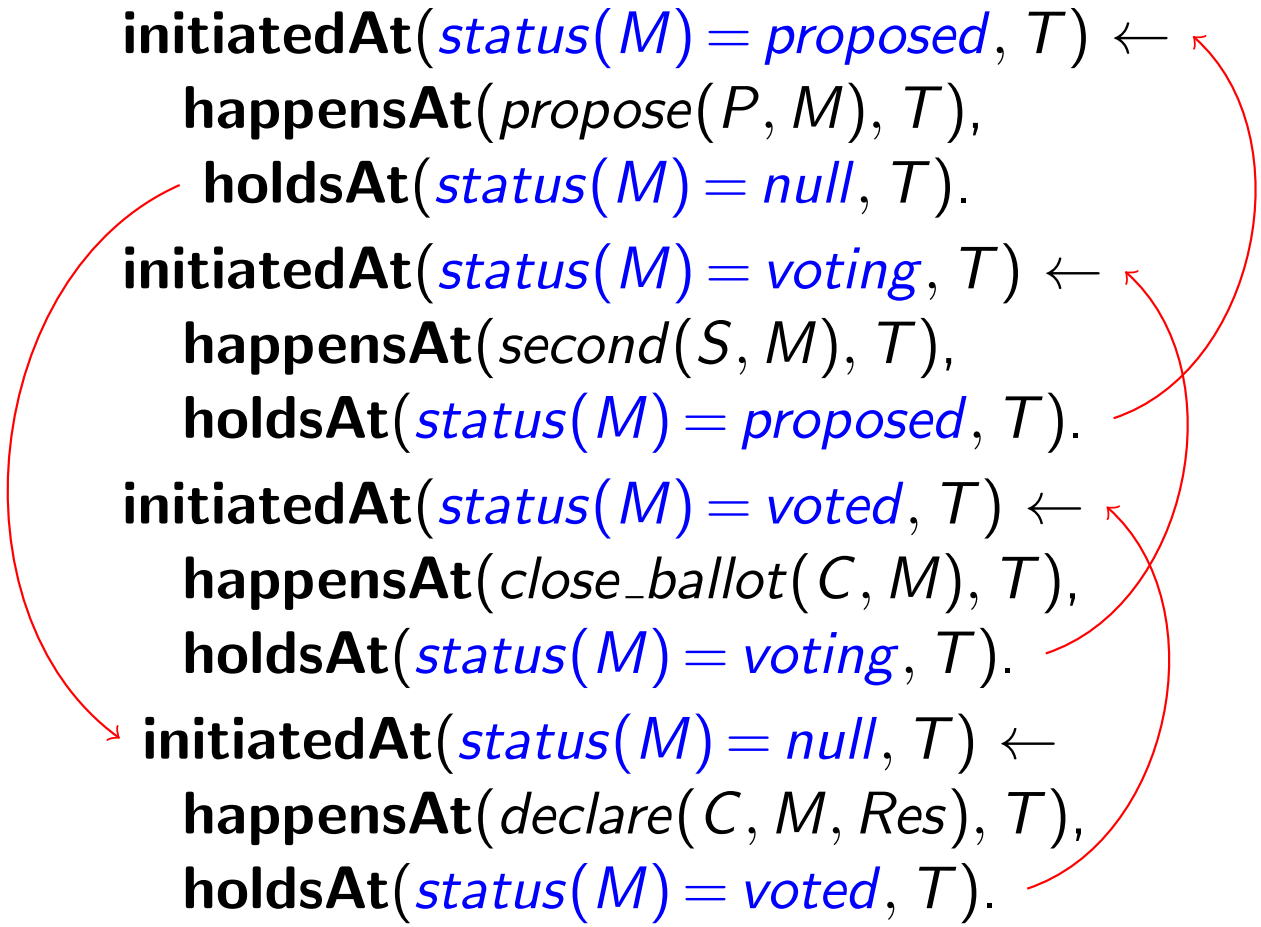
initiatedAt$(status(M) = proposed, T) \leftarrow$
  happensAt$(propose(P, M), T)$,
  holdsAt$(status(M) = null, T)$.

initiatedAt$(status(M) = voting, T) \leftarrow$
  happensAt$(second(S, M), T)$,
  holdsAt$(status(M) = proposed, T)$.

initiatedAt$(status(M) = voted, T) \leftarrow$
  happensAt$(close\_ballot(C, M), T)$,
  holdsAt$(status(M) = voting, T)$.

initiatedAt$(status(M) = null, T) \leftarrow$
  happensAt$(declare(C, M, Res), T)$,
  holdsAt$(status(M) = voted, T)$.

Cyclic Dependencies in Temporal Specifications

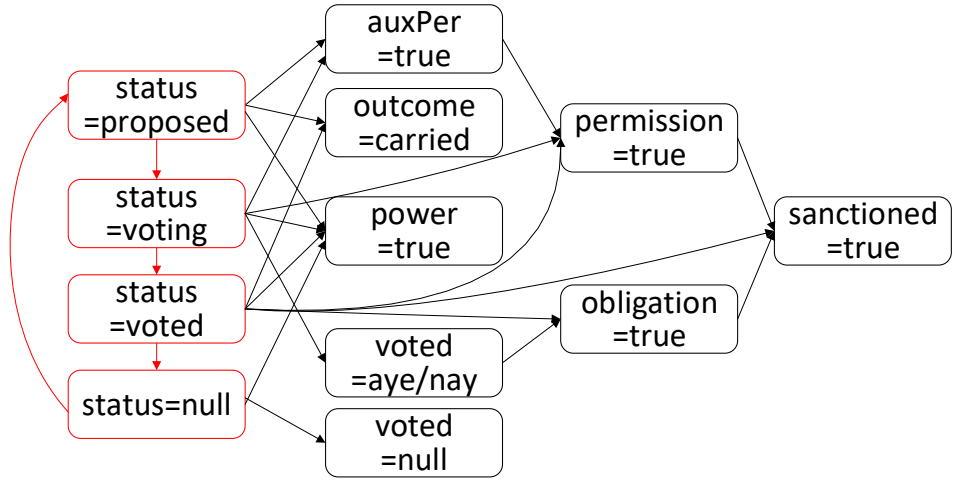


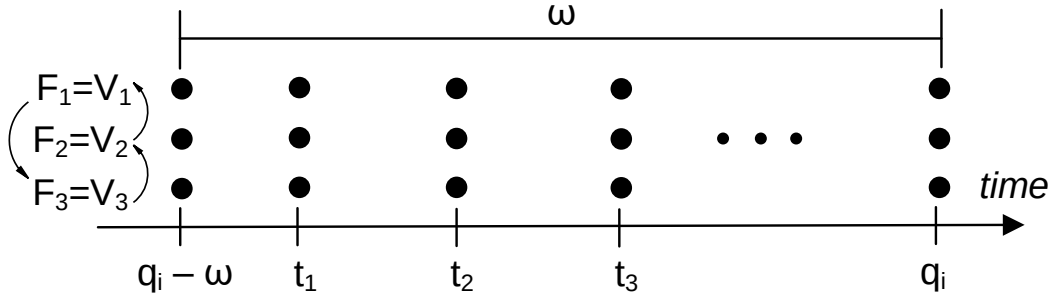
$\text{RTEC}_\circ$: Run-Time Event Calculus for Cyclic Dependencies

$\text{RTEC}_\circ$: an extension of RTEC for efficient reasoning over specifications with cyclic dependencies

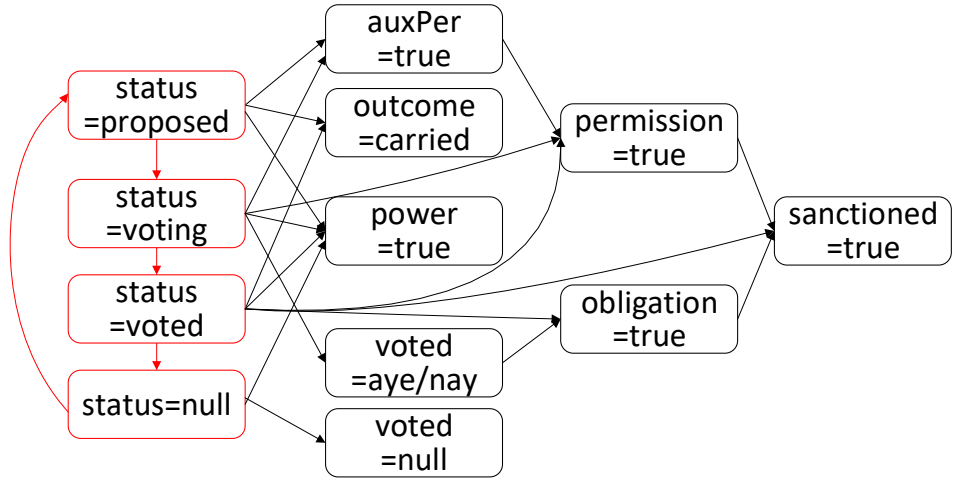
- Formal & open-source computational framework
- Locally stratified specifications
- Incremental caching of intermediate computations
- Scalable in large, real-world data streams

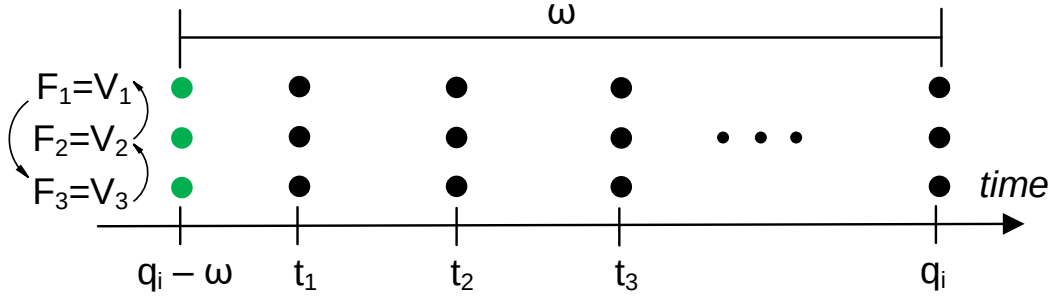
Stratification in RTEC$_\circ$



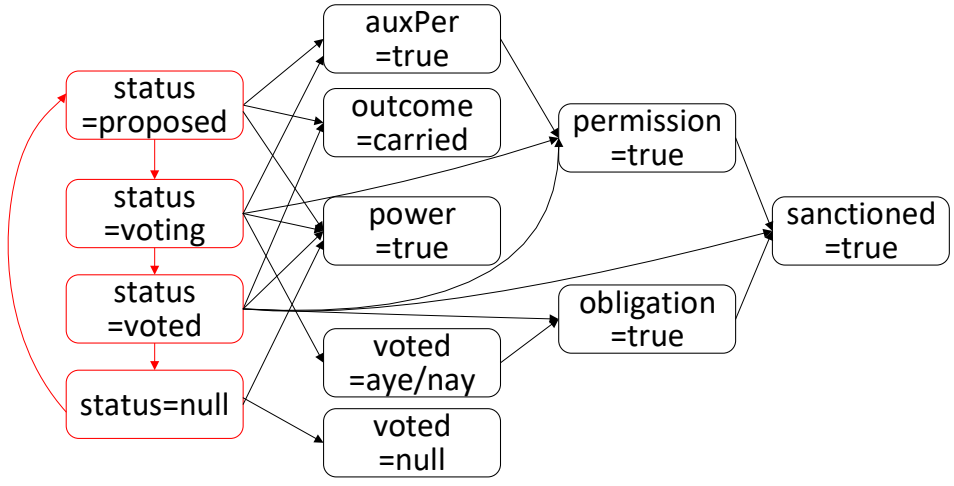


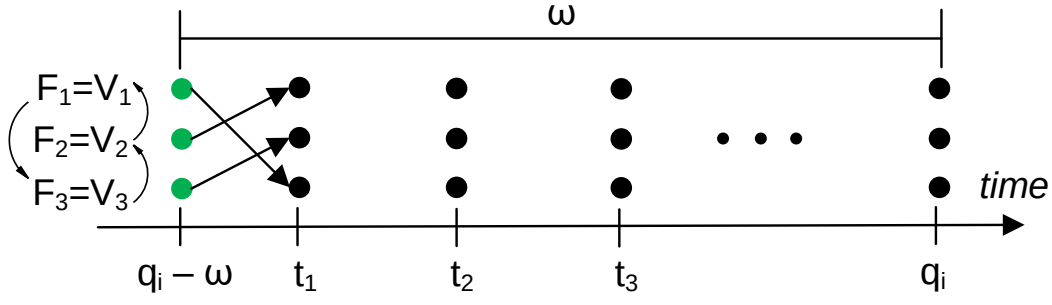
Stratification in RTEC$_\circ$



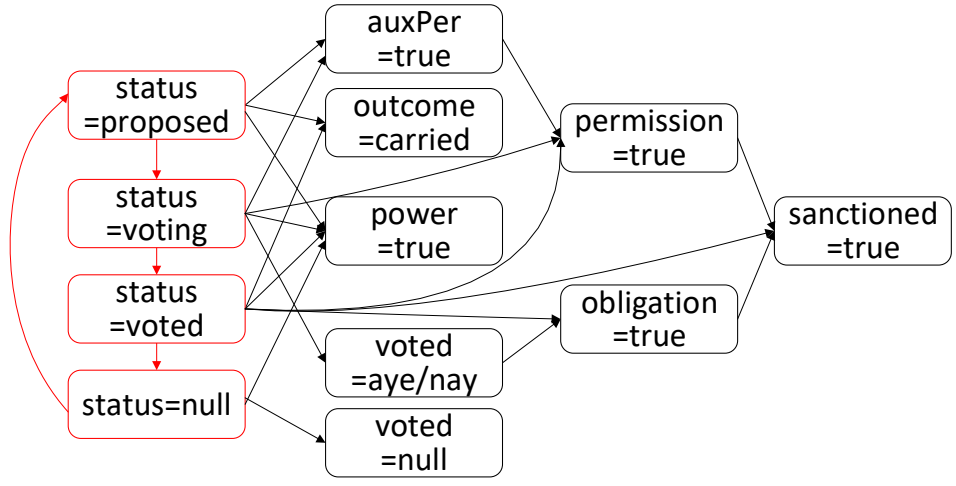


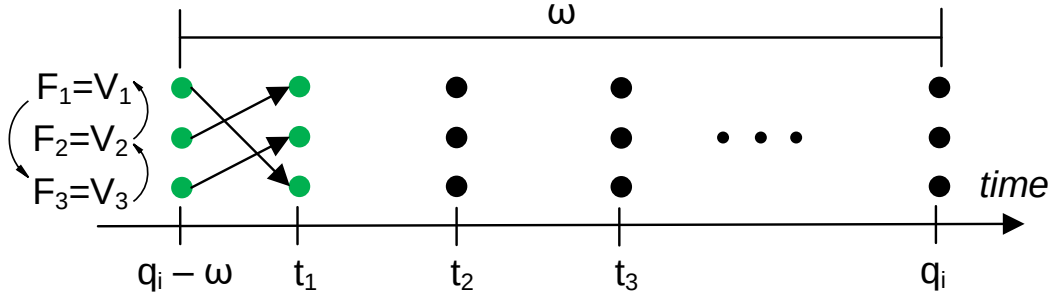
Stratification in RTEC$_\circ$



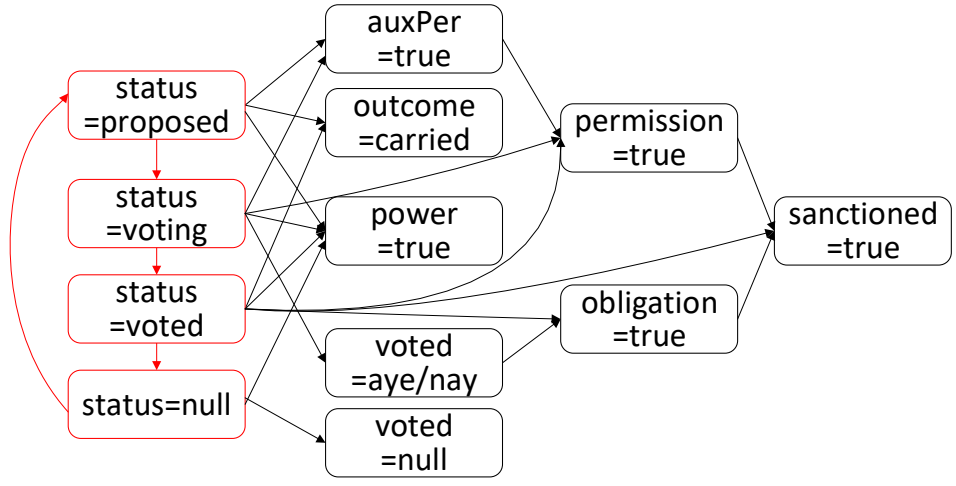


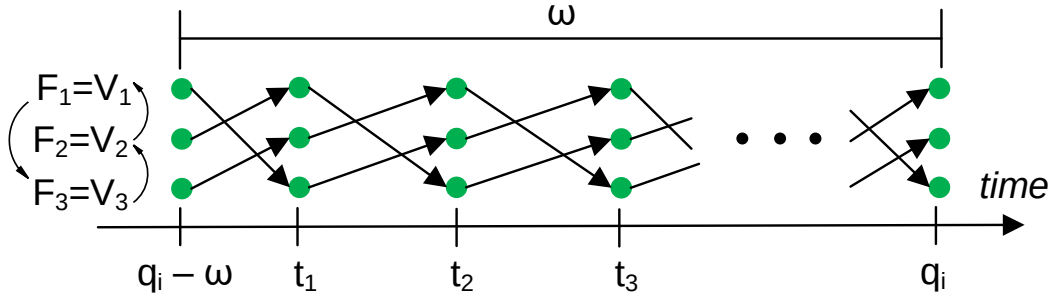
Stratification in RTEC$_\circ$



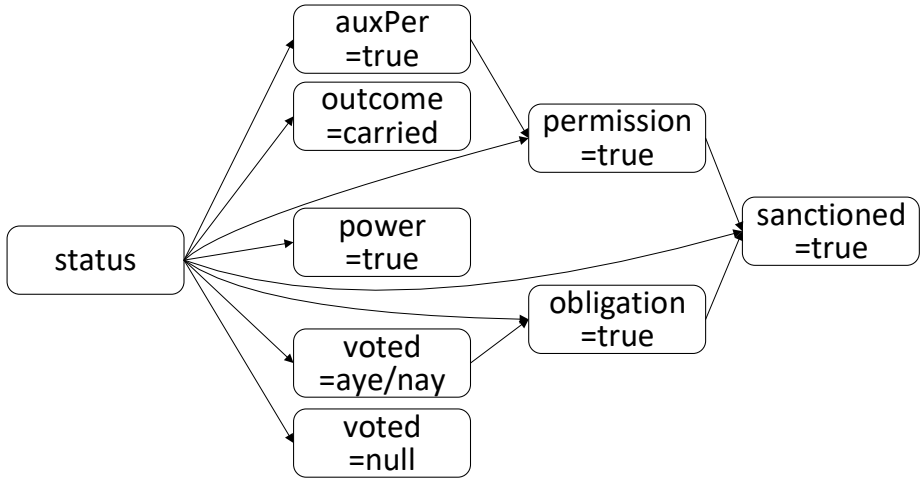


Stratification in RTEC$_\circ$





Stratification in RTEC$_\circ$

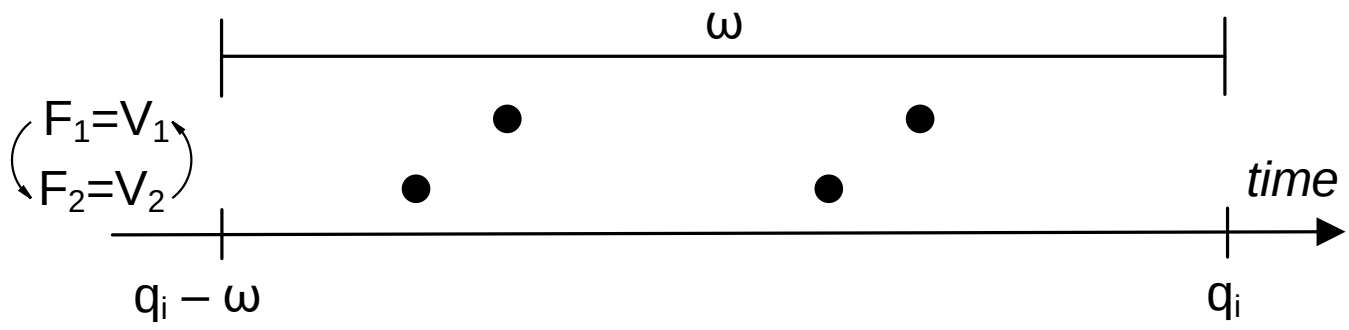


$P_1 \qquad P_2 \qquad P_3 \qquad P_4$

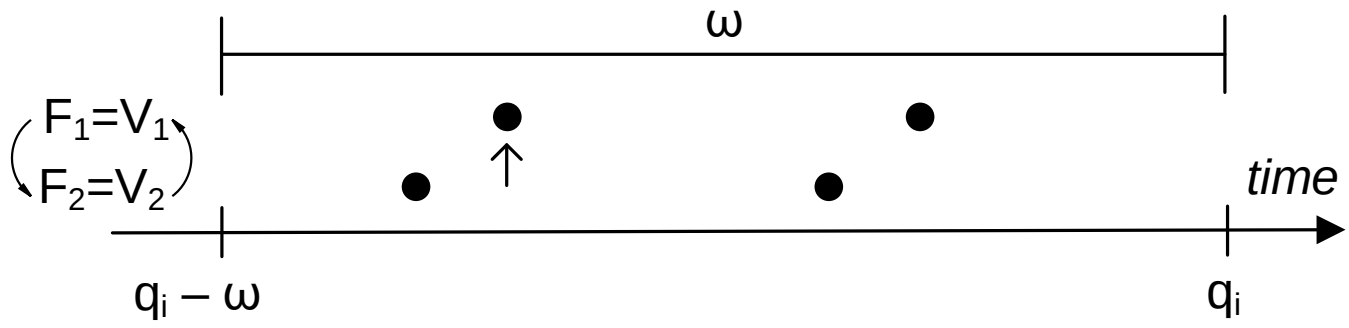
$$P_1 = P_{q_i-\omega} \cdots P_{q_i}$$

An event description of RTEC$_\circ$ is a locally stratified logic program.

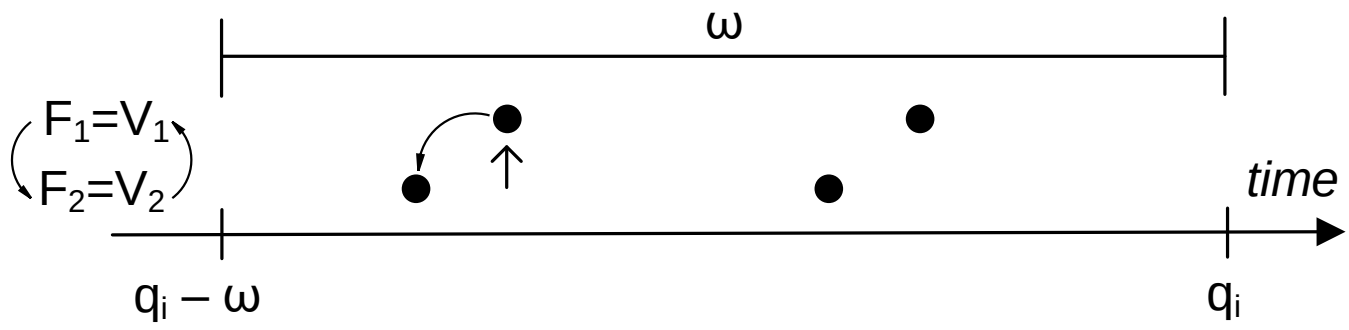
Handling Cyclic Dependencies



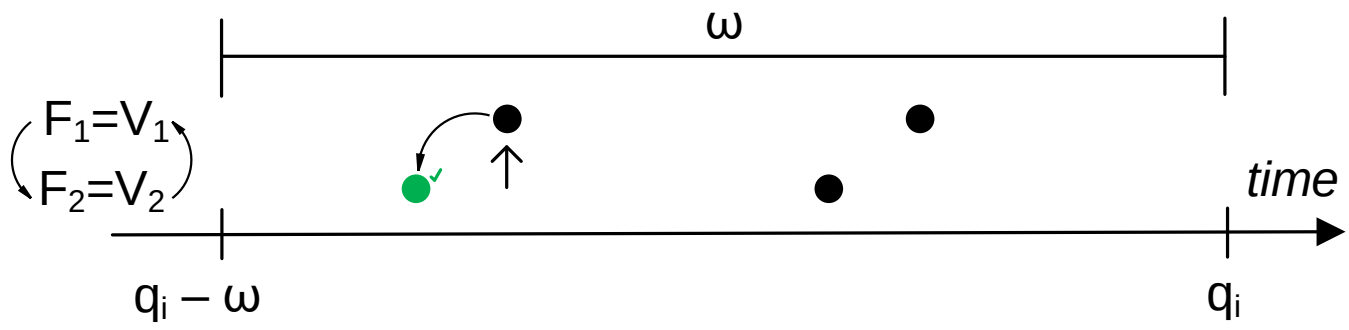
Handling Cyclic Dependencies



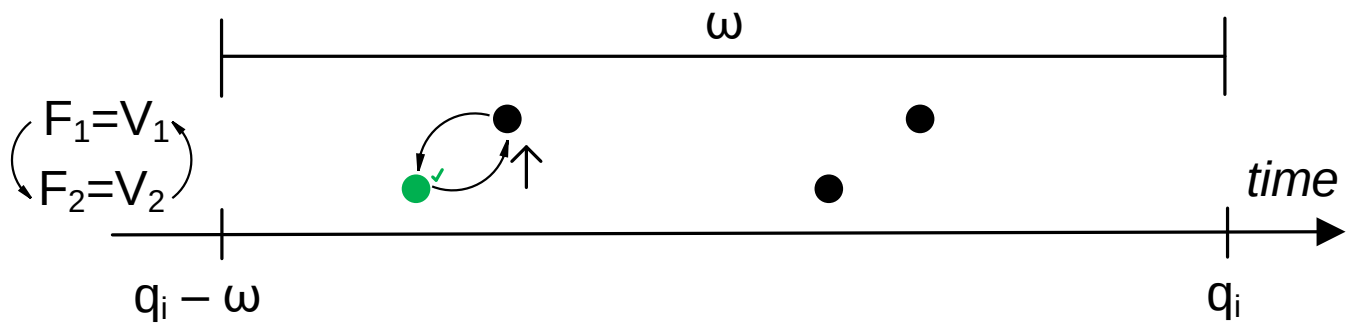
Handling Cyclic Dependencies



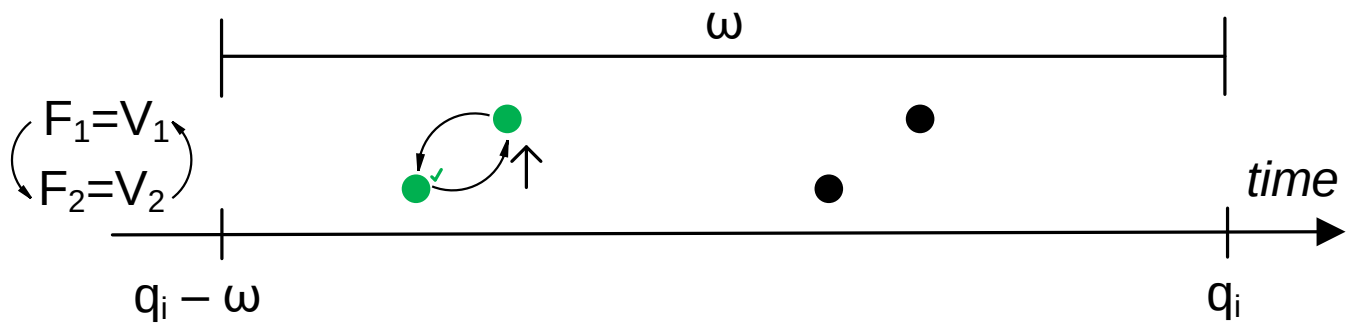
Handling Cyclic Dependencies



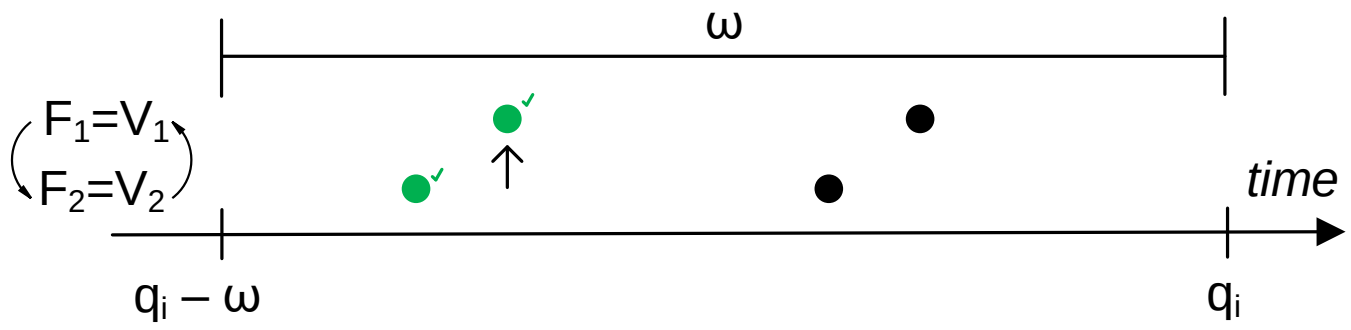
Handling Cyclic Dependencies



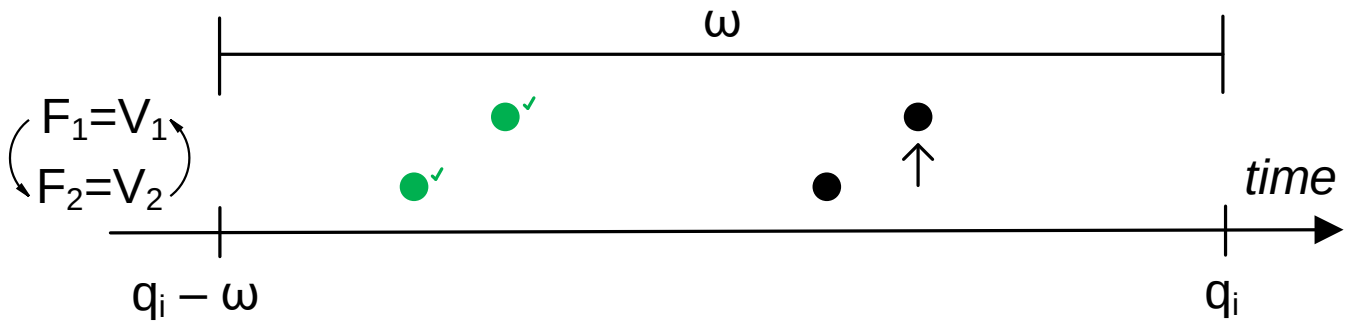
Handling Cyclic Dependencies



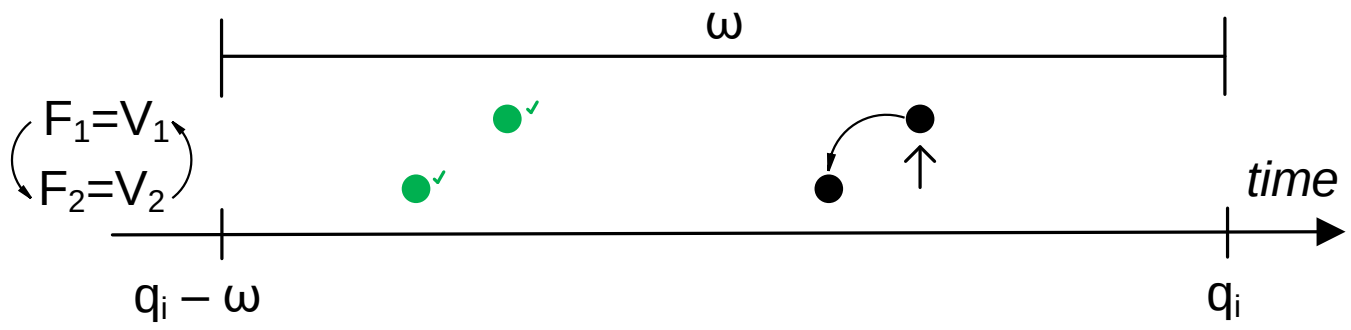
Handling Cyclic Dependencies



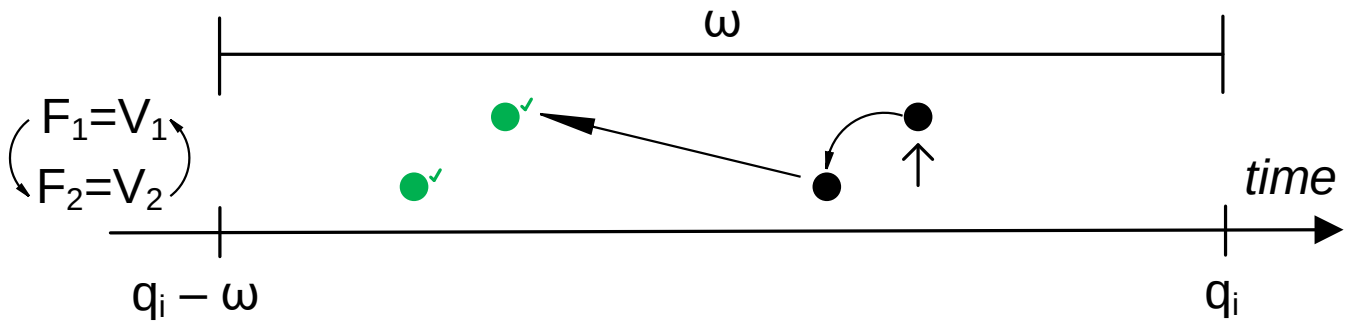
Handling Cyclic Dependencies



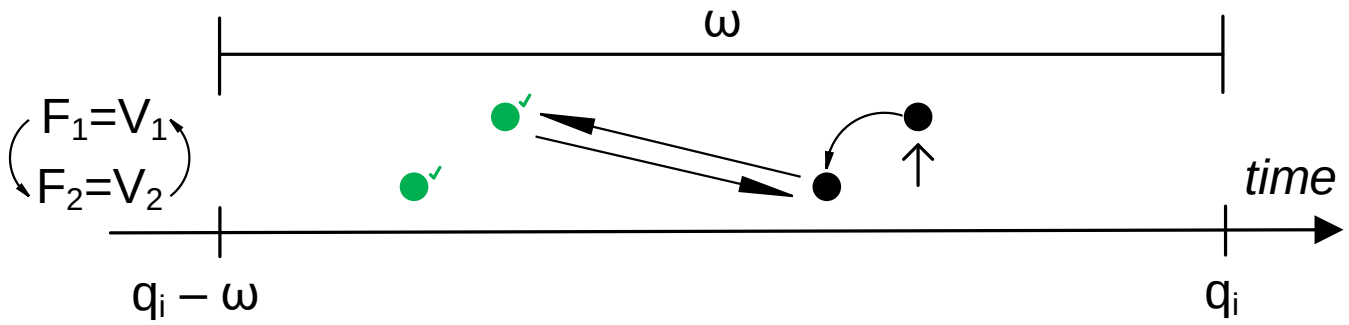
Handling Cyclic Dependencies



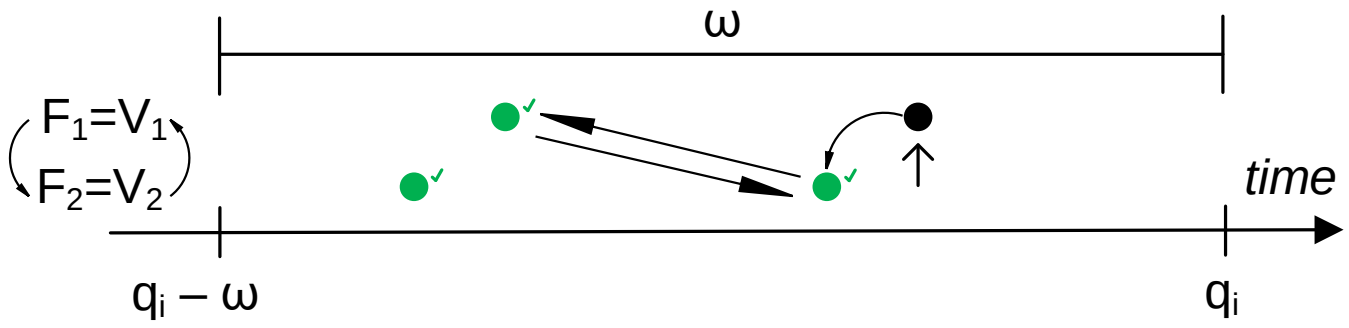
Handling Cyclic Dependencies



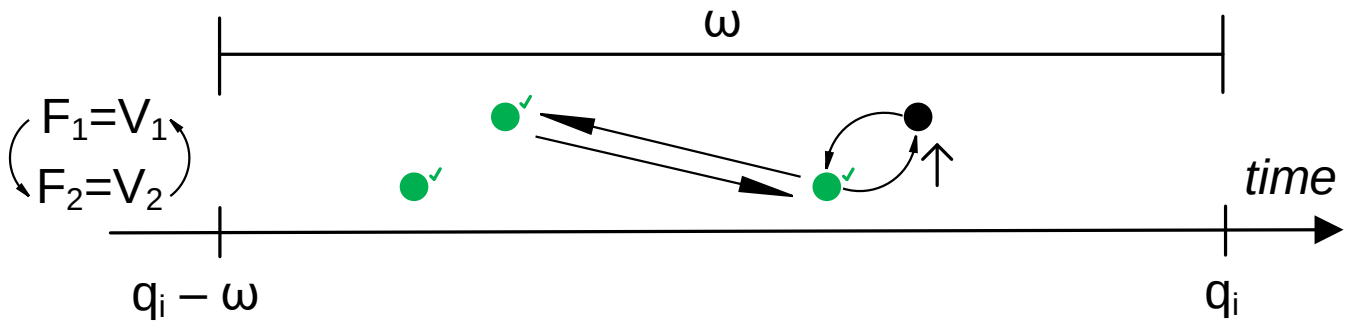
Handling Cyclic Dependencies



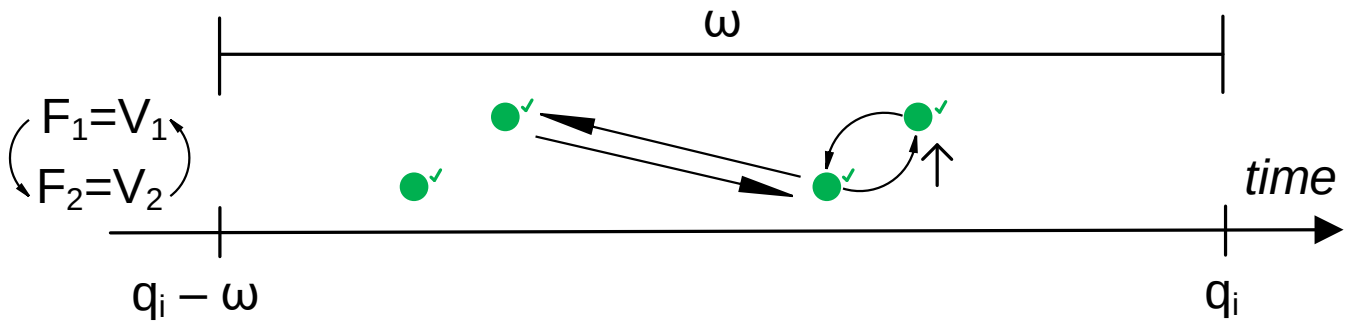
Handling Cyclic Dependencies



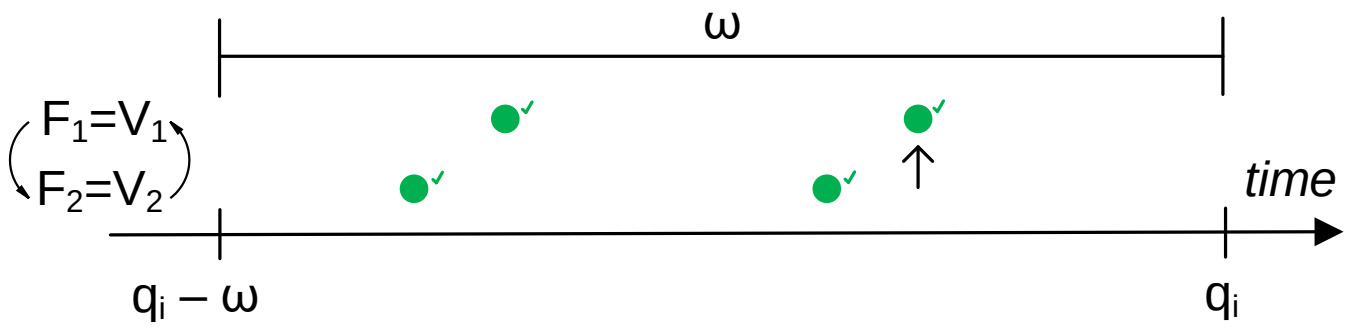
Handling Cyclic Dependencies



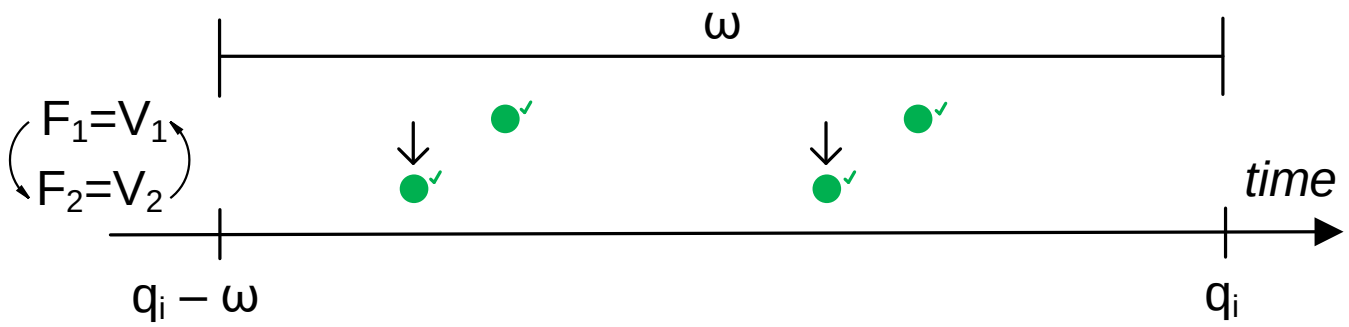
Handling Cyclic Dependencies



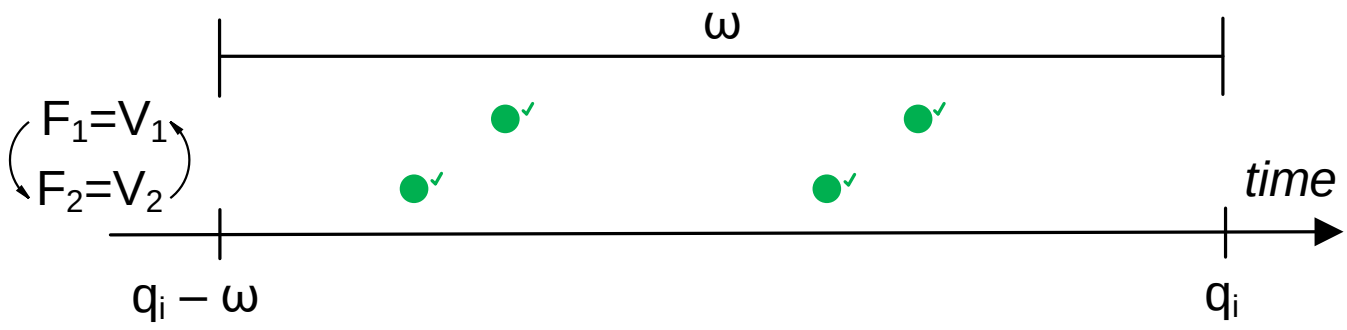
Handling Cyclic Dependencies



Handling Cyclic Dependencies

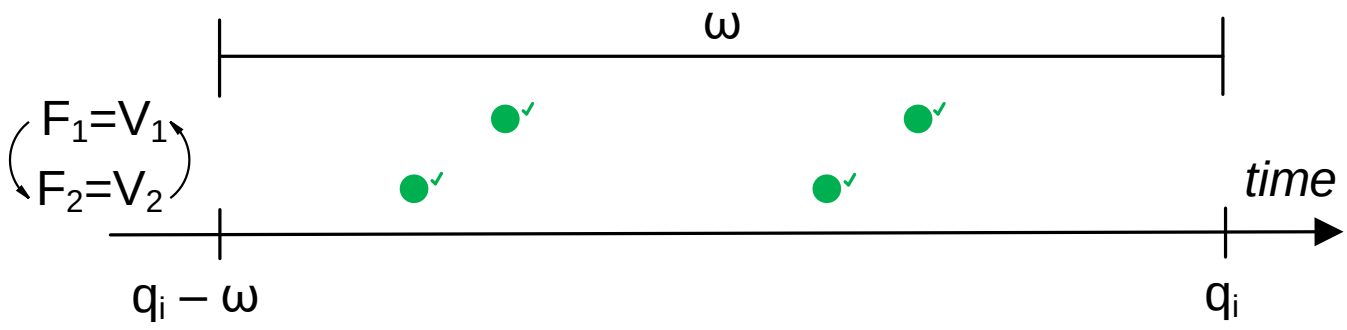


Handling Cyclic Dependencies

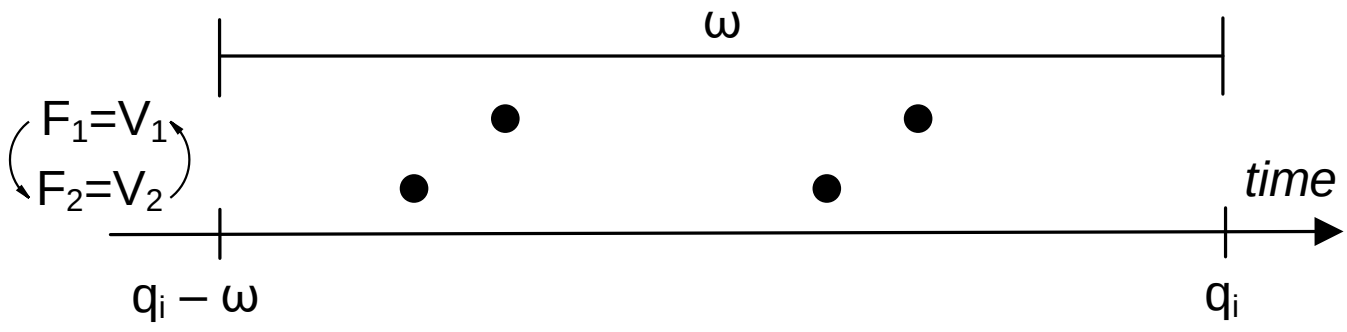


Handling Cyclic Dependencies

RTEC$_\circ$

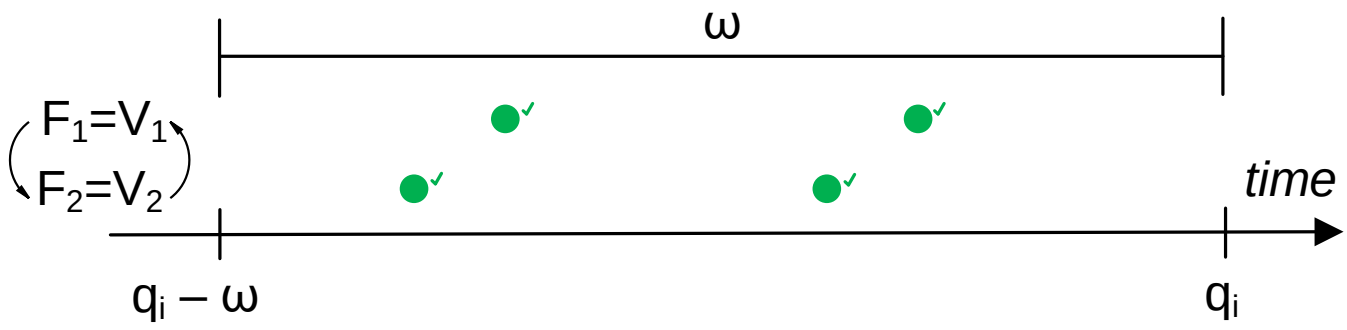


Event Calculus

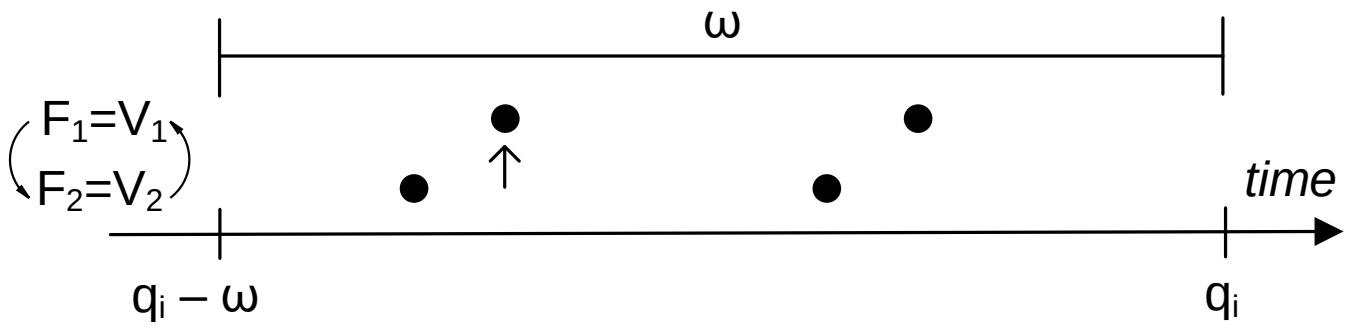


Handling Cyclic Dependencies

RTEC$_\circ$

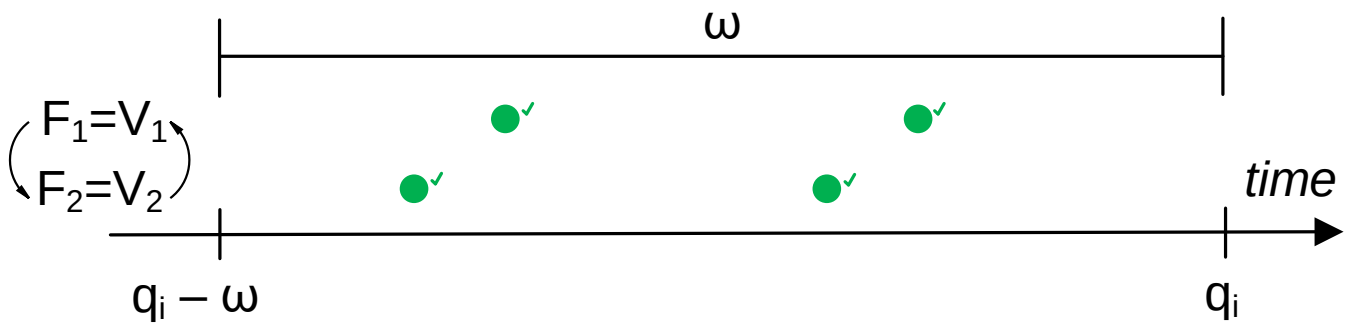


Event Calculus

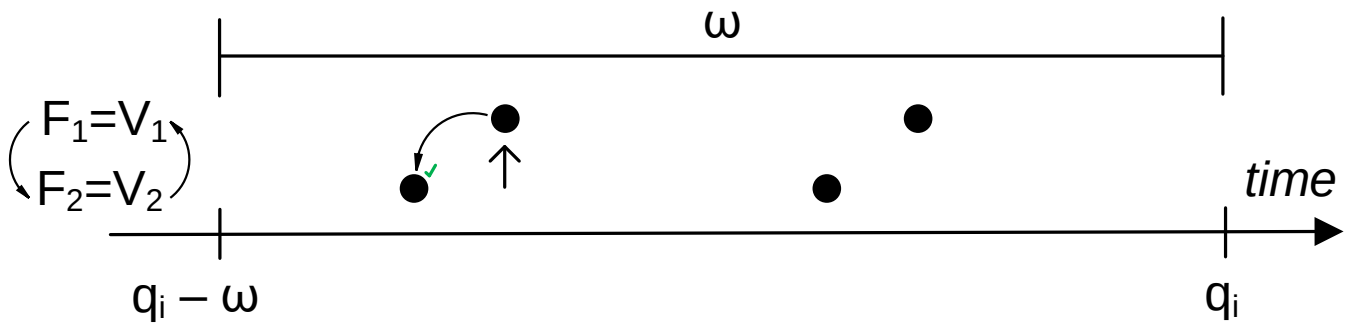


Handling Cyclic Dependencies

RTEC$_\circ$

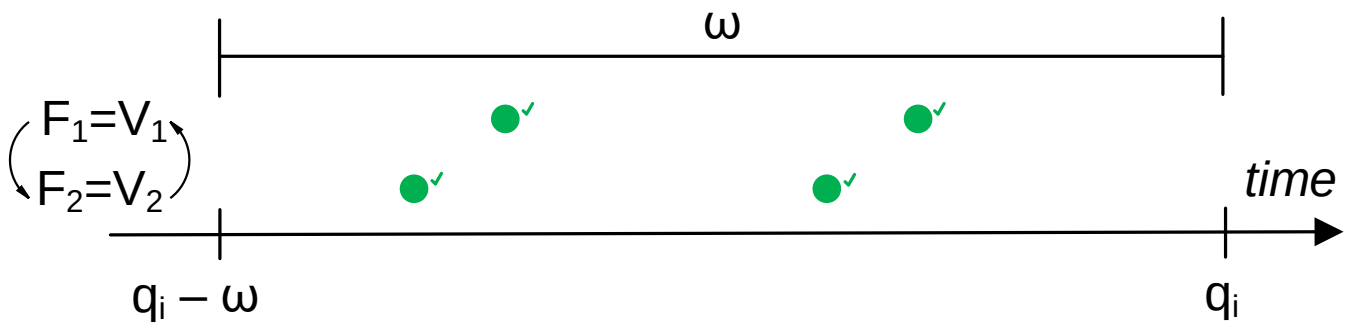


Event Calculus

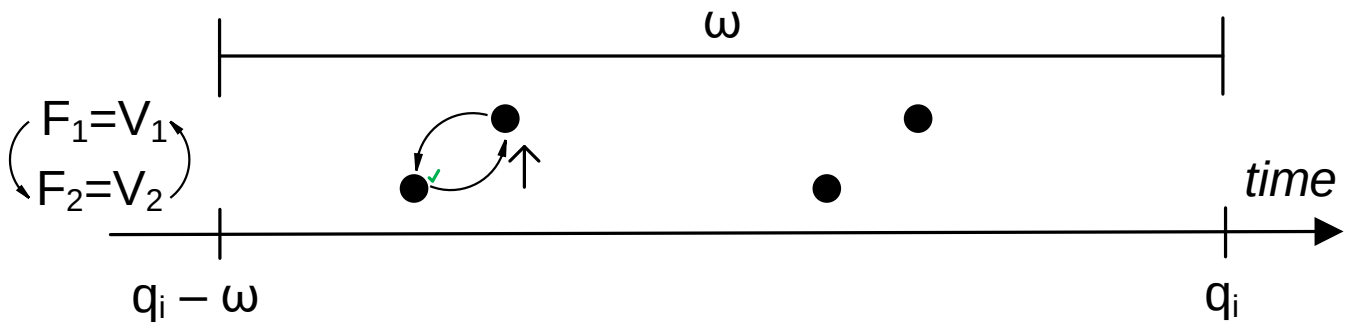


Handling Cyclic Dependencies

RTEC$_\circ$

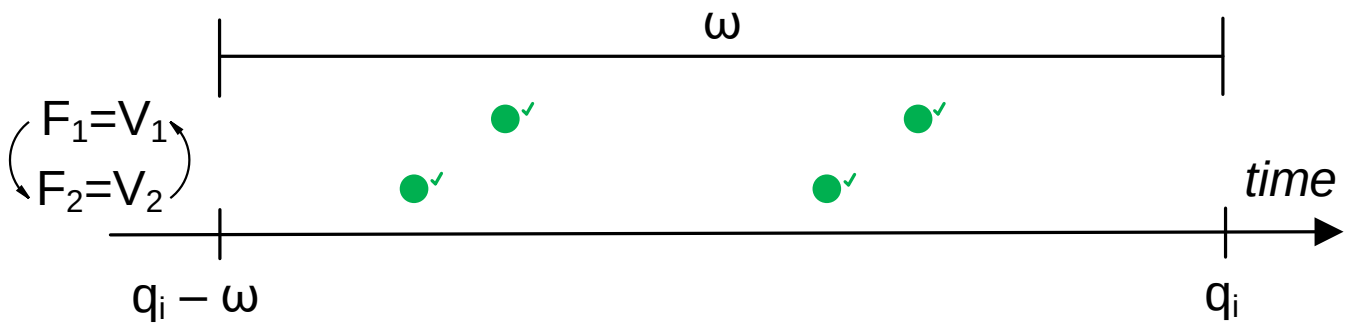


Event Calculus

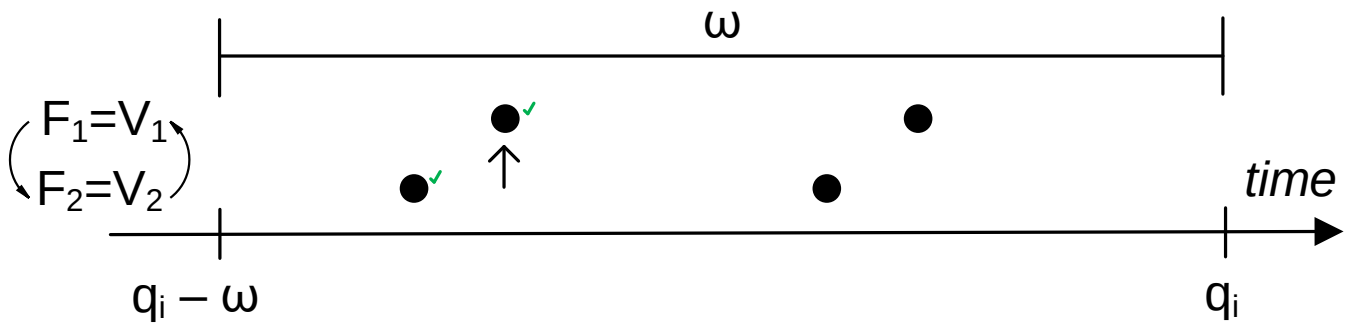


Handling Cyclic Dependencies

RTEC$_\circ$

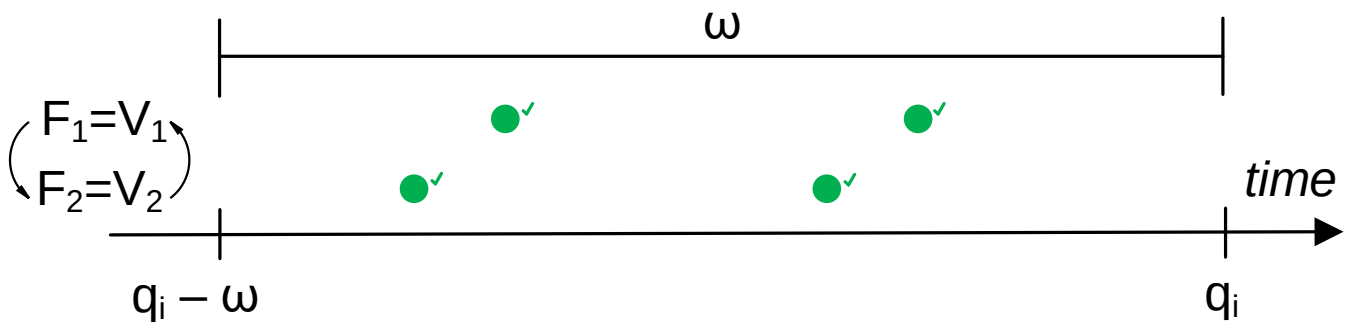


Event Calculus

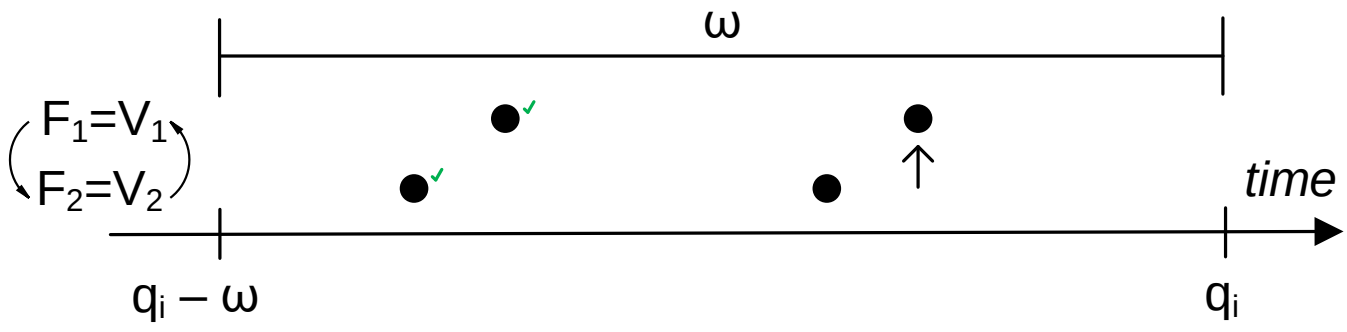


Handling Cyclic Dependencies

RTEC$_\circ$

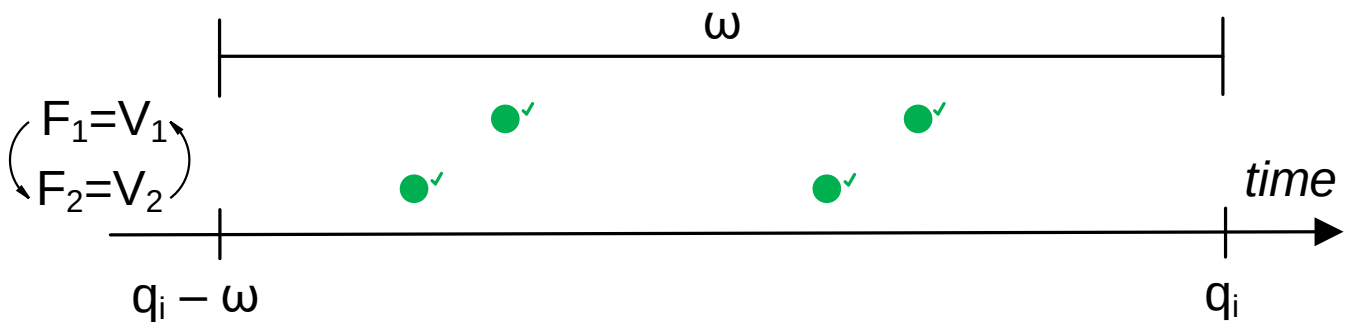


Event Calculus

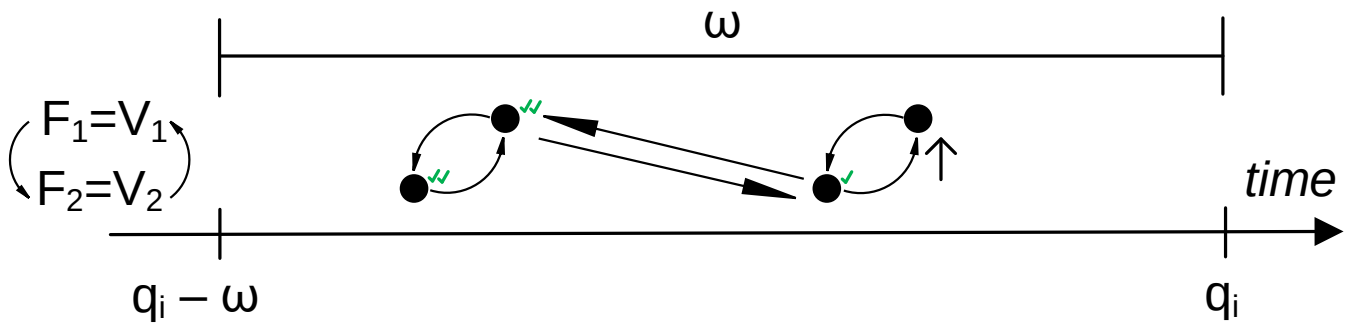


Handling Cyclic Dependencies

RTEC$_\circ$

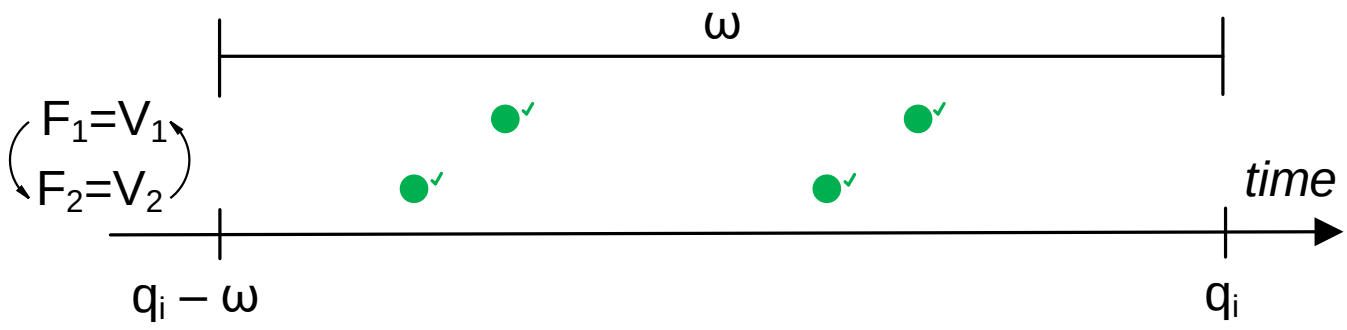


Event Calculus

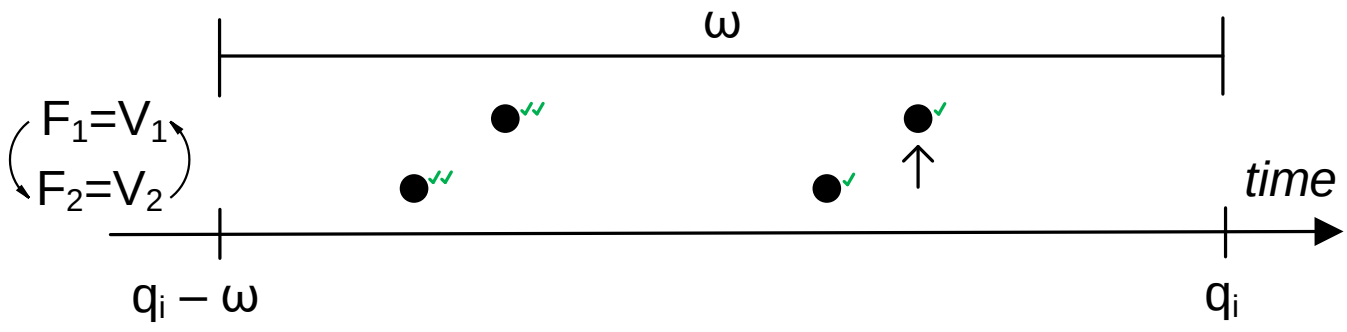


Handling Cyclic Dependencies

RTEC$_\circ$

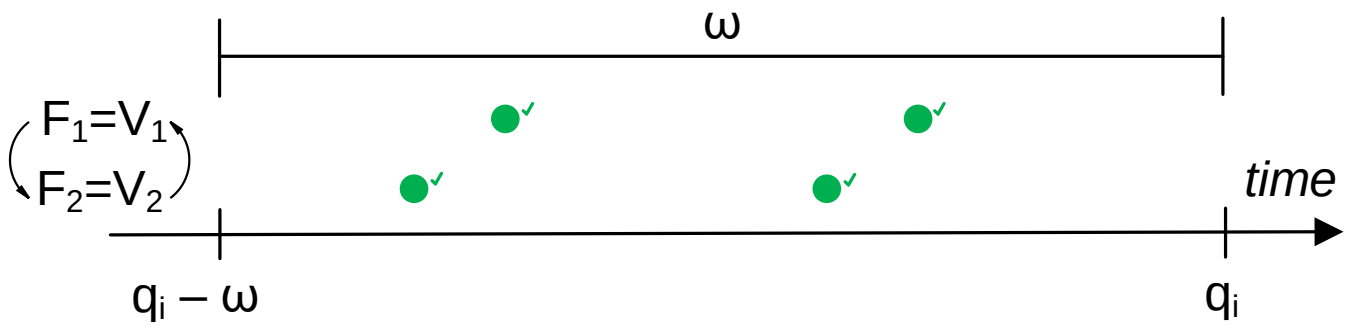


Event Calculus

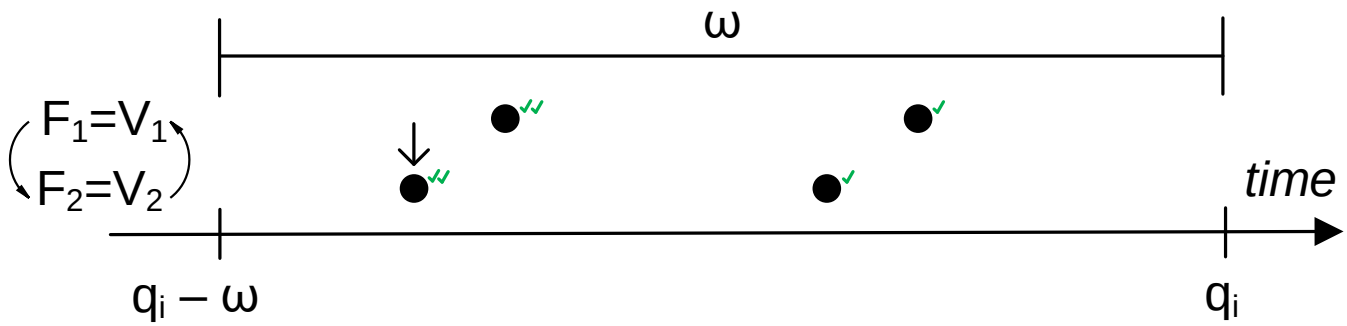


Handling Cyclic Dependencies

RTEC$_\circ$

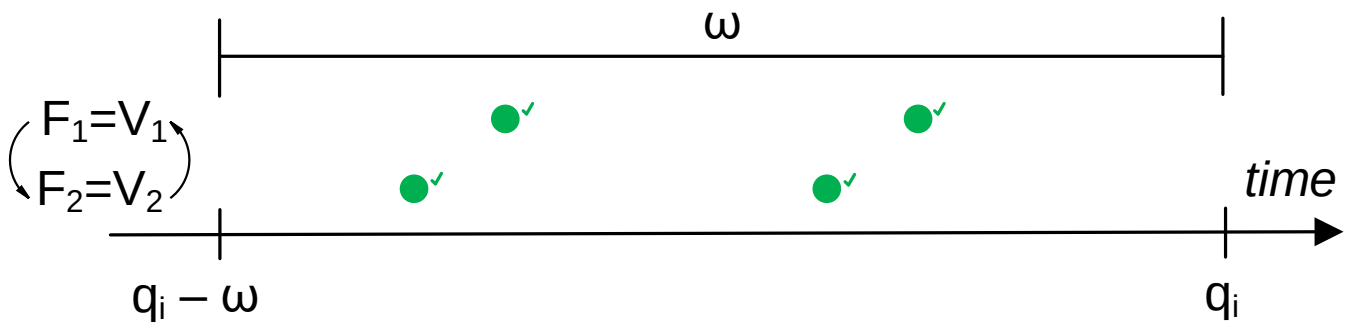


Event Calculus

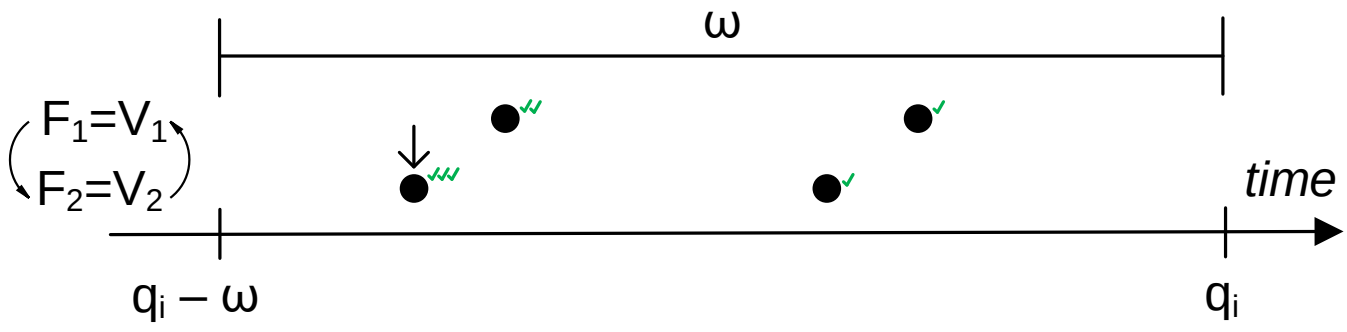


Handling Cyclic Dependencies

RTEC$_\circ$

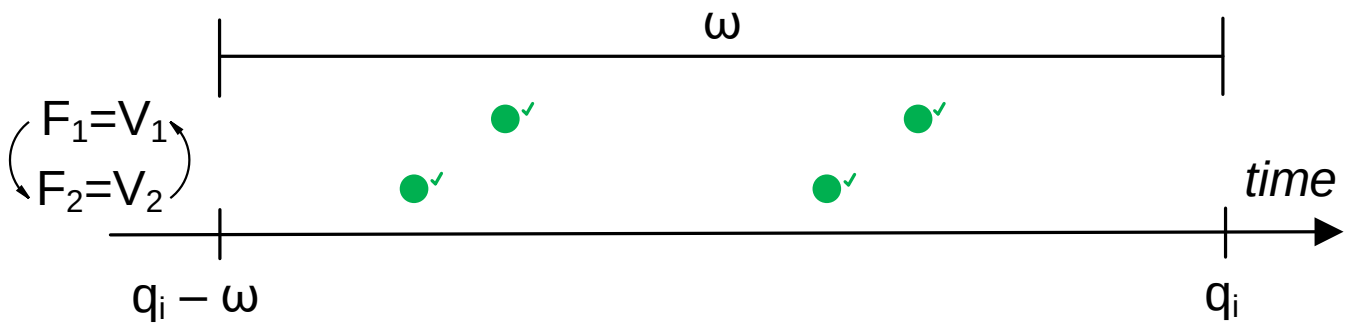


Event Calculus

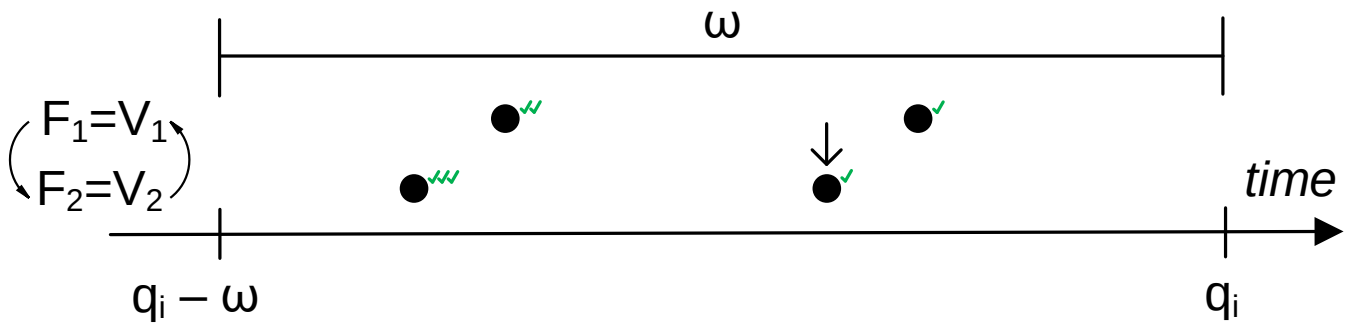


Handling Cyclic Dependencies

RTEC$_\circ$

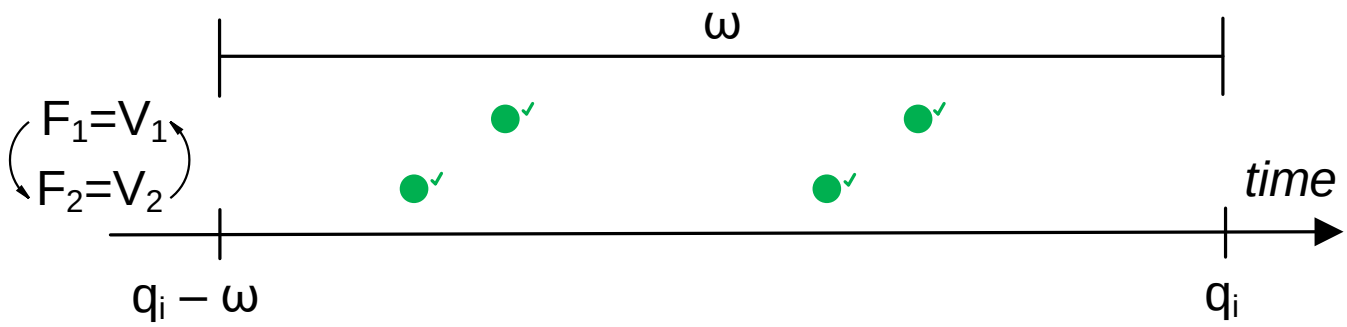


Event Calculus

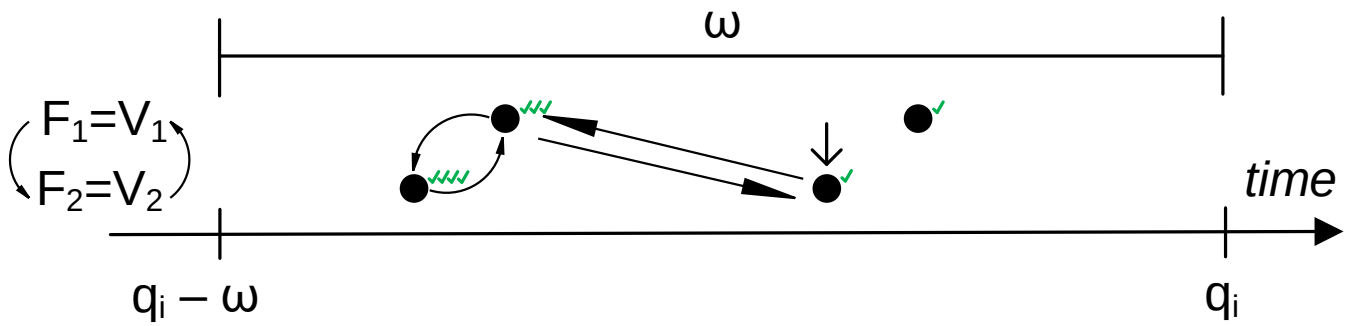


Handling Cyclic Dependencies

RTEC$_\circ$

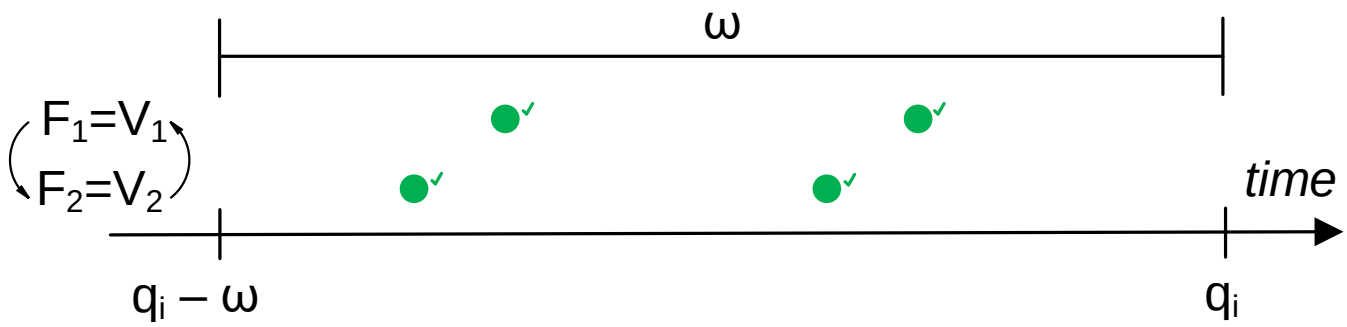


Event Calculus

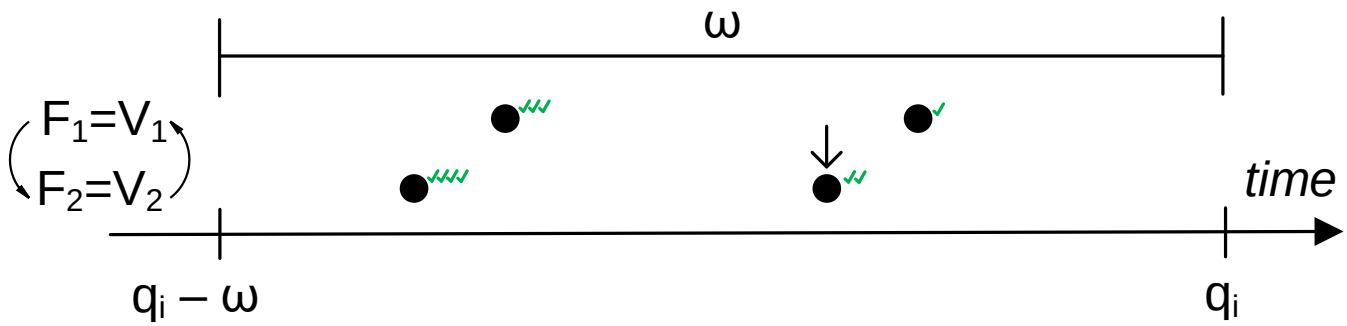


Handling Cyclic Dependencies

RTEC$_\circ$

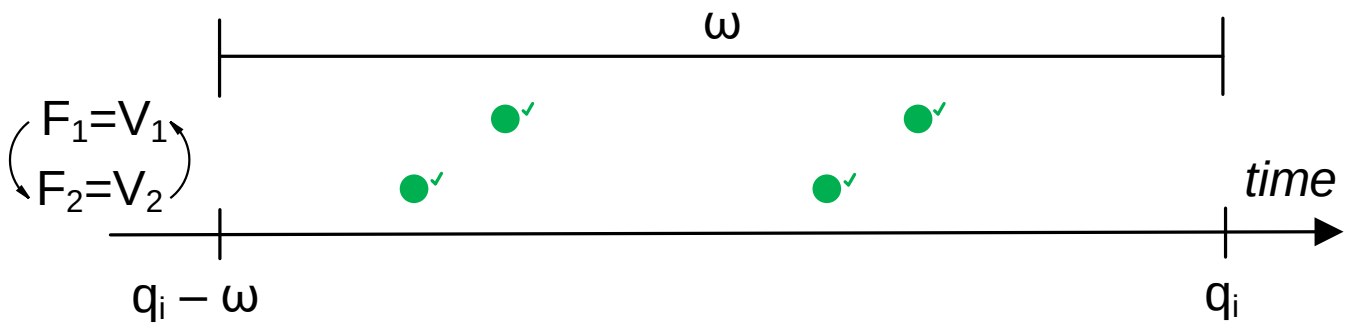


Event Calculus

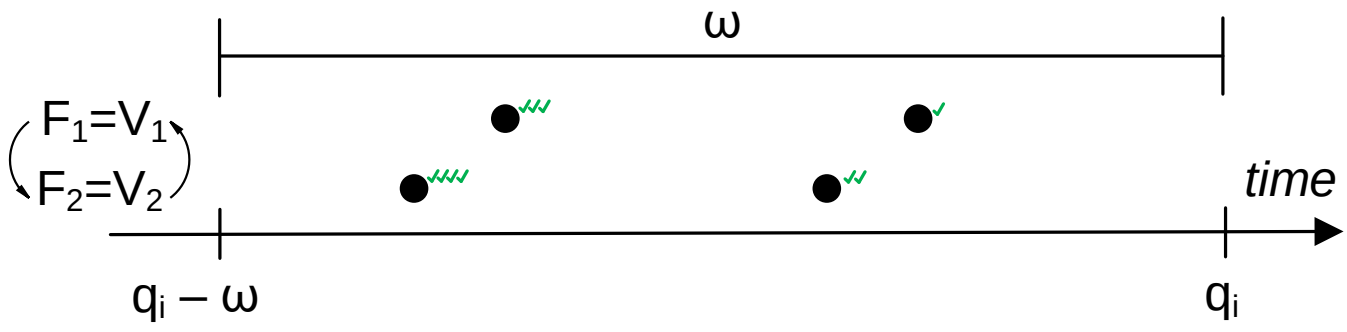


Handling Cyclic Dependencies

RTEC$_\circ$

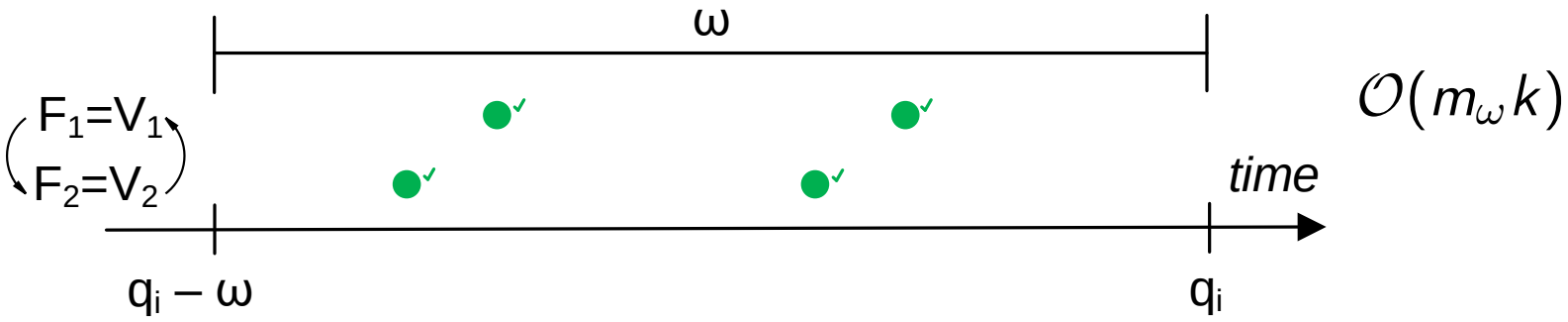


Event Calculus

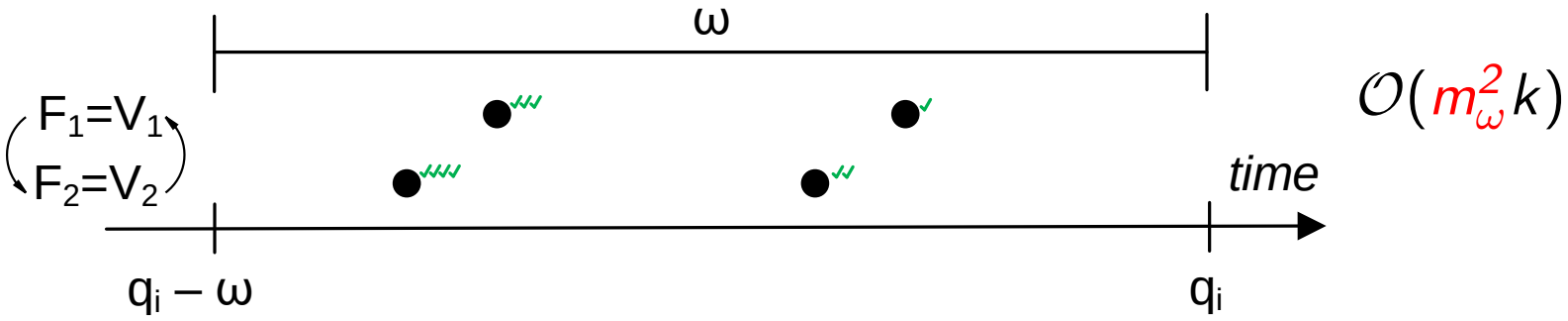


Handling Cyclic Dependencies

RTEC$_\circ$



Event Calculus



Experimental Setup

Multi-Agent Systems: Voting & NetBill

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

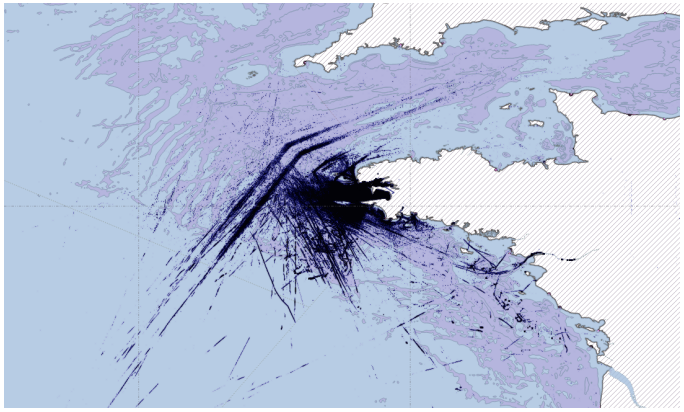
Experimental Setup

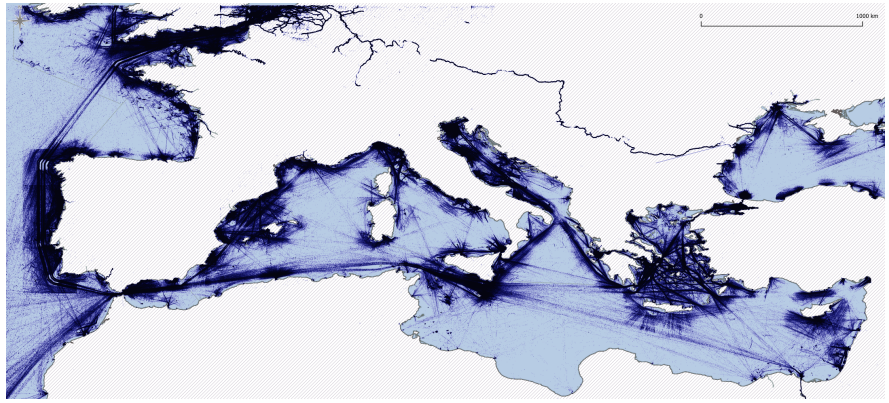
Multi-Agent Systems: Voting & NetBill

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

Maritime Situational Awareness

- Recognise dangerous, illegal and suspicious vessel activity.





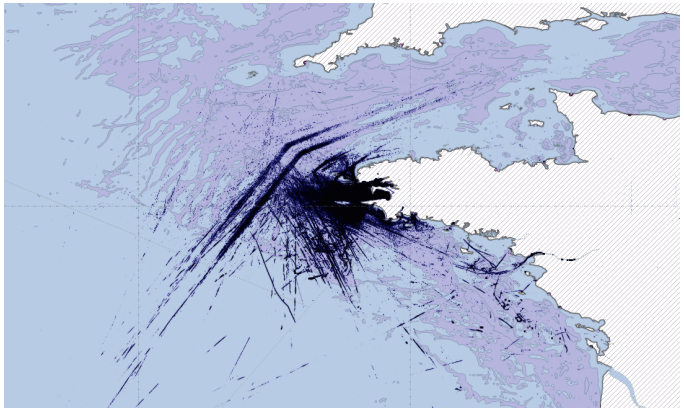
Experimental Setup

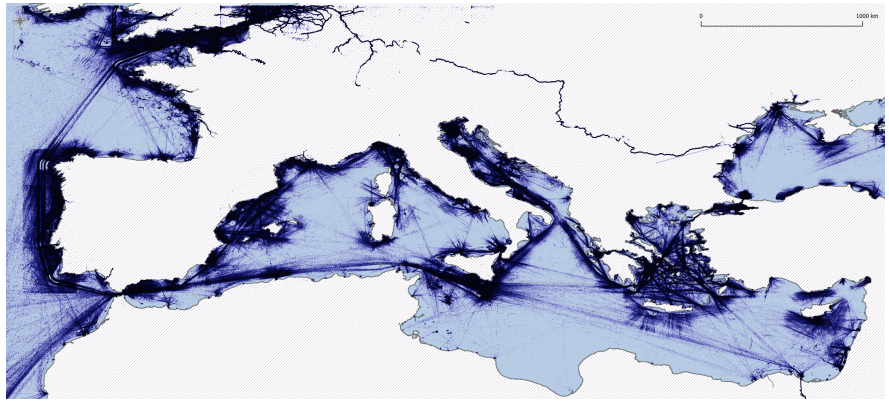
Multi-Agent Systems: Voting & NetBill

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

Maritime Situational Awareness

- Recognise dangerous, illegal and suspicious vessel activity.



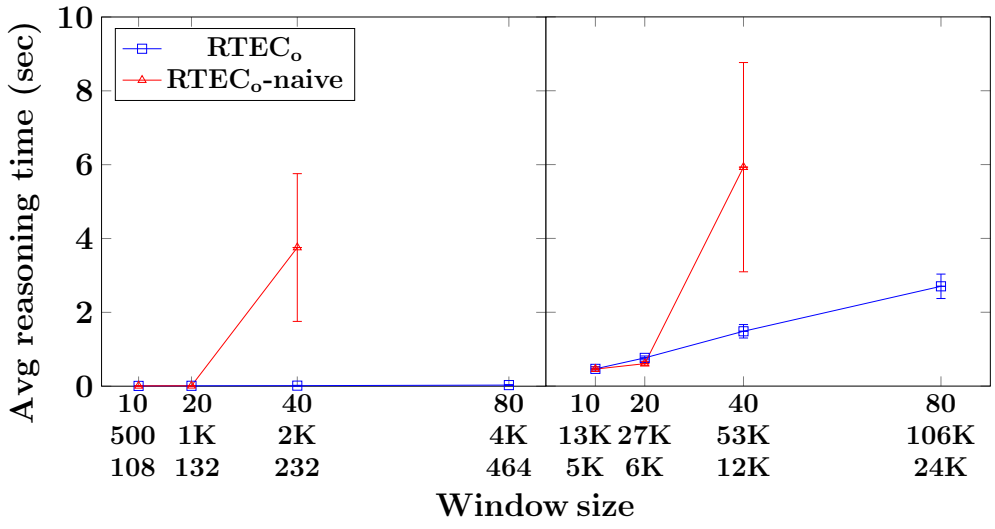


Code, Data & Temporal Specifications

`https://github.com/aartikis/RTEC`

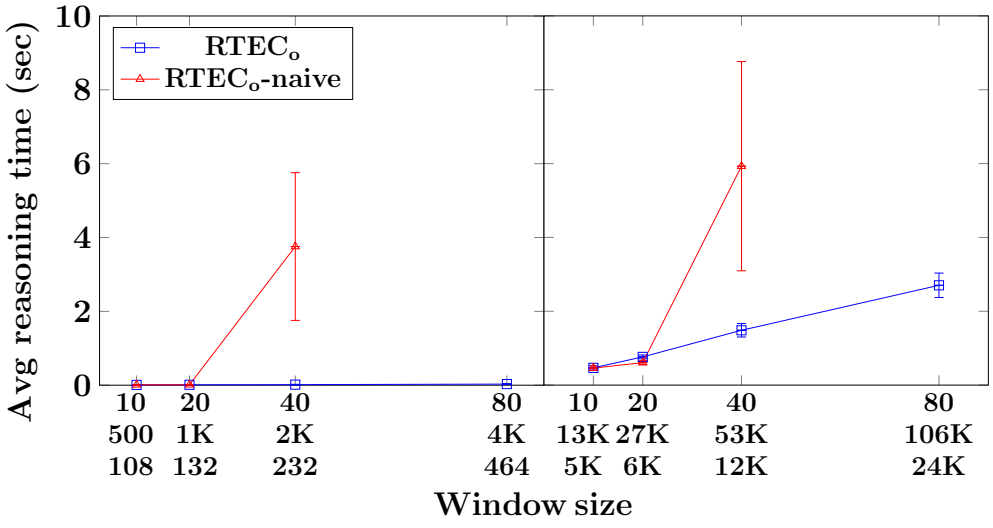
Experimental Results

Voting & NetBill

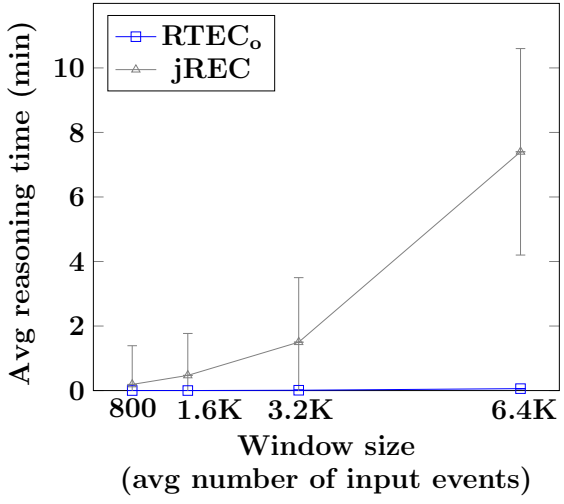


Experimental Results

Voting & NetBill

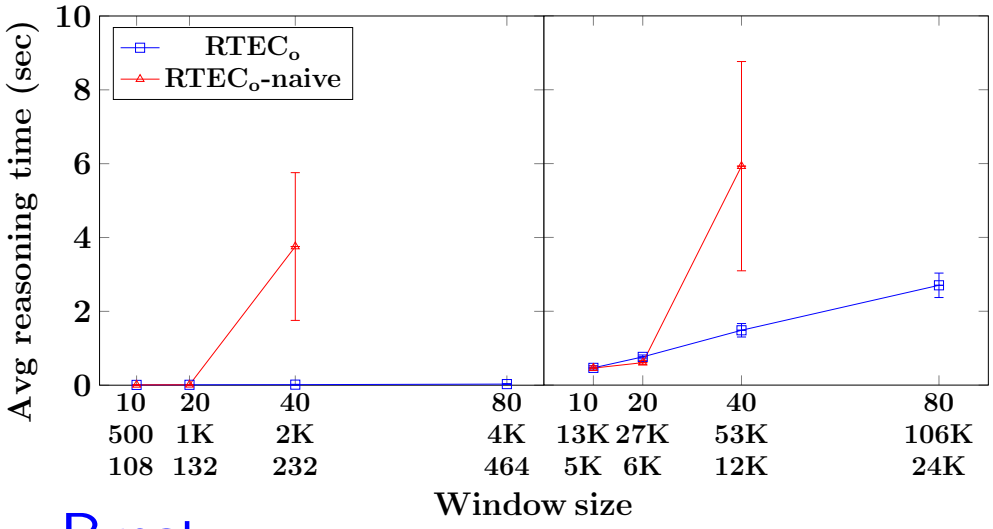


Voting: *status* fluent

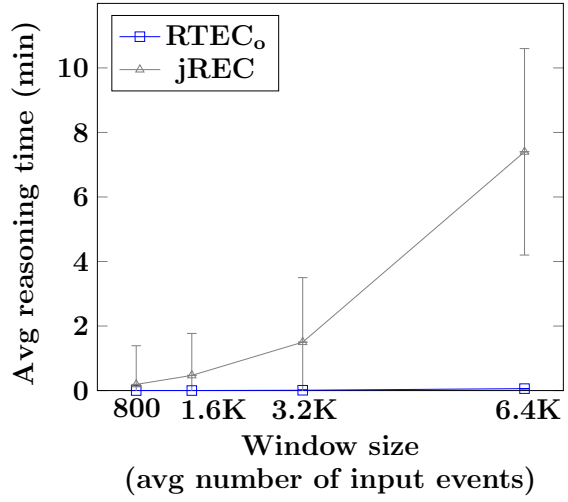


Experimental Results

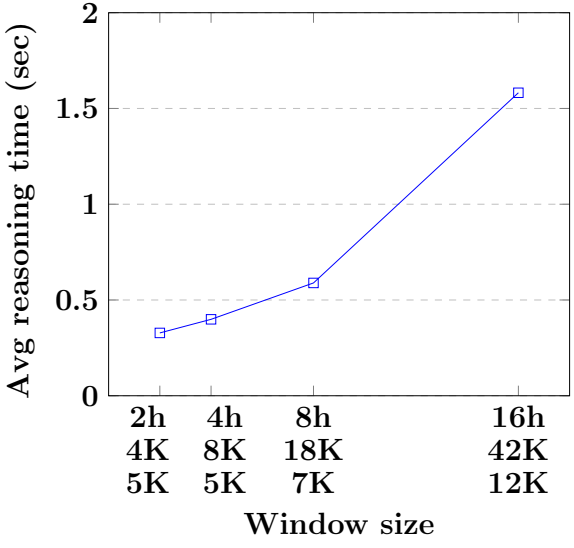
Voting & NetBill



Voting: *status* fluent

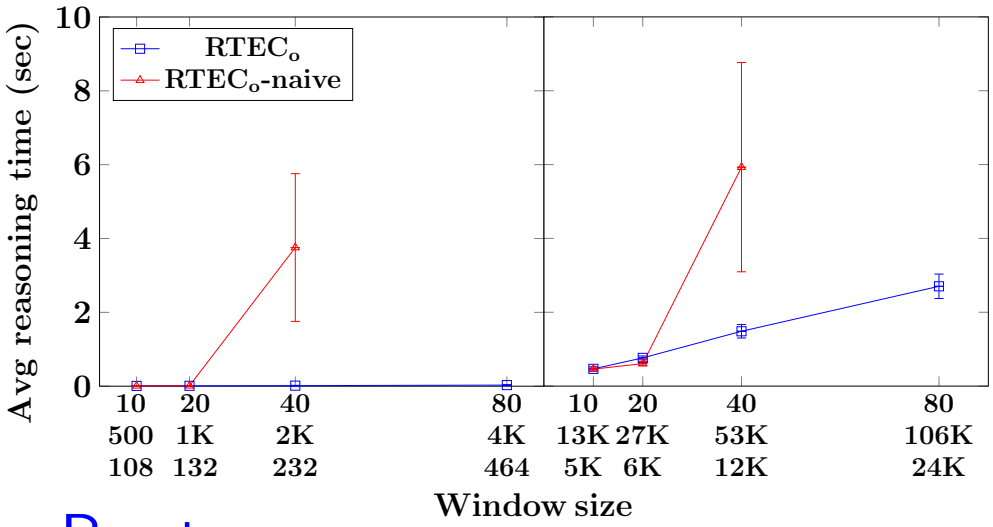


Brest

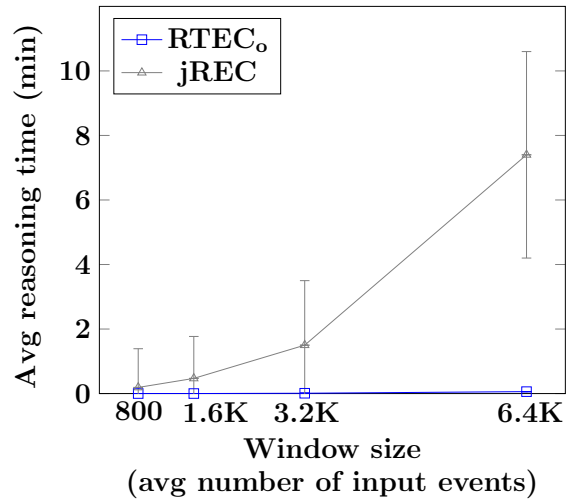


Experimental Results

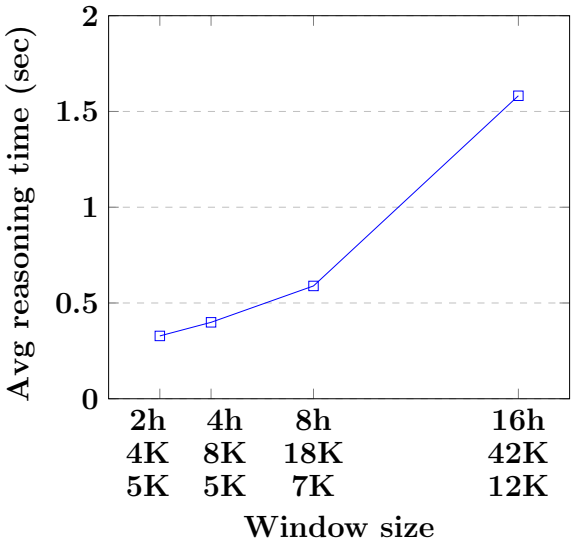
Voting & NetBill



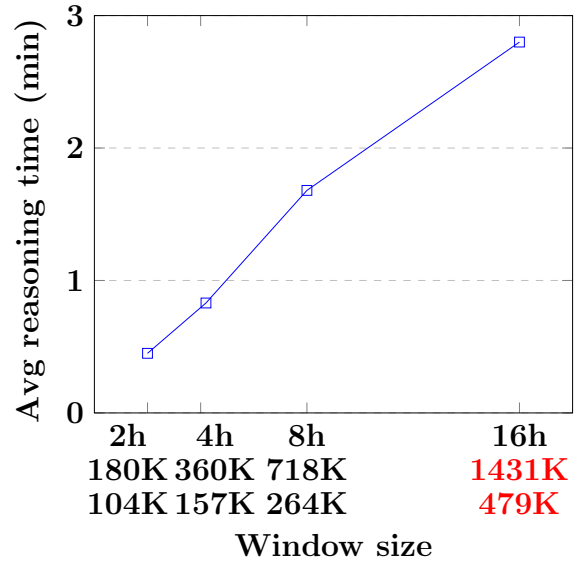
Voting: *status* fluent



Brest



European seas



Summary & Further Work

RTEC$_\circ$

- A formal, open-source stream reasoning system
- Efficient treatment of cyclic dependencies
- Locally stratified specifications
- Reproducible empirical evaluation on large data streams

Summary & Further Work

RTEC$_\circ$

- A formal, open-source stream reasoning system
- Efficient treatment of cyclic dependencies
- Locally stratified specifications
- Reproducible empirical evaluation on large data streams

Further Work

- Efficient treatment of deadlines
- Integrating RTEC$_\circ$ in neuro-symbolic frameworks
- Empirical comparison with other stream reasoning systems
- RTEC at the edge

Summary & Further Work

RTEC$_\circ$

- A formal, open-source stream reasoning system
- Efficient treatment of cyclic dependencies
- Locally stratified specifications
- Reproducible empirical evaluation on large data streams

Further Work

- Efficient treatment of deadlines
- Integrating RTEC$_\circ$ in neuro-symbolic frameworks
- Empirical comparison with other stream reasoning systems
- RTEC at the edge

Resources

`https://cer.iit.demokritos.gr/`

Thank you!