What & Why

“Towards Ideal Semantics for Analyzing Stream Reasoning”

Stream Reasoning
What & Why

“Towards Ideal Semantics for Analyzing Stream Reasoning”

- **Stream Reasoning**: Logical reasoning on streaming data
“Towards Ideal Semantics for Analyzing Stream Reasoning”

Stream Reasoning: Logical reasoning on streaming data

- Streams = **tuples** (atoms) with **timestamps**
- Essential aspect: **window** functions
What & Why

“Towards Ideal Semantics for Analyzing Stream Reasoning”

- Stream Reasoning: Logical reasoning on streaming data
  - Streams = **tuples** (atoms) with **timestamps**
  - Essential aspect: **window** functions

- Semantics
“Towards Ideal **Semantics** for Analyzing Stream Reasoning”

- **Stream Reasoning**: Logical reasoning on streaming data
  - Streams = **tuples** (atoms) with **timestamps**
  - Essential aspect: **window** functions

- **Semantics**: Lack of theory
“Towards Ideal Semantics for Analyzing Stream Reasoning”

- Stream Reasoning: Logical reasoning on streaming data
  - Streams = **tuples** (atoms) with **timestamps**
  - Essential aspect: **window** functions

- Semantics: Lack of theory

- Analysis
What & Why

“Towards Ideal Semantics for Analyzing Stream Reasoning”

Stream Reasoning: Logical reasoning on streaming data
  - Streams = **tuples** (atoms) with **timestamps**
  - Essential aspect: **window** functions

Semantics: Lack of theory

**Analysis**: Hard to predict, hard to compare
What & Why

“Towards Ideal Semantics for Analyzing Stream Reasoning”

► Stream Reasoning: Logical reasoning on streaming data
  ► Streams = *tuples* (atoms) with *timestamps*
  ► Essential aspect: *window* functions

► Semantics: Lack of theory

► Analysis: Hard to predict, hard to compare

► Ideal
“Towards Ideal Semantics for Analyzing Stream Reasoning”

Stream Reasoning: Logical reasoning on streaming data
- Streams = **tuples** (atoms) with **timestamps**
- Essential aspect: **window** functions

Semantics: Lack of theory

Analysis: Hard to predict, hard to compare

**Ideal**
- Idealization: Abstract from practical (operational) issues
- Generalization: Uniform representation
Example: Trams and buses

Arrival times at different stations $p_i$
Example: Trams and buses

Arrival times at different stations $p_i$
Example: Trams and buses

Arrival times at different stations $p_i$

$bus(i_2, p_1)$
$tram(i_1, p_1)$
Example: Trams and buses

Arrival times at different stations $p_i$

\[
\text{bus}(i_2, p_1) \\
\text{tram}(i_1, p_1)
\]
Example: Trams and buses

Arrival times at different stations $p_i$

$bus(i_2, p_1)$
$tram(i_1, p_1)$
$tram(i_3, p_2)$

0 2 8
Example: Trams and buses

Arrival times at different stations $p_i$

$bus(i_2, p_1)$
$tram(i_1, p_1)$
$tram(i_3, p_2)$
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$
$\text{tram}(i_3, p_2)$
$\text{bus}(i_4, p_2)$

0 2 8 11
Example: Trams and buses

Arrival times at different stations $p_i$

- $bus(i_2, p_1)$
- $tram(i_1, p_1)$
- $tram(i_3, p_2)$
- $bus(i_4, p_2)$

- Normal DB: Query for
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$
$\text{tram}(i_3, p_2)$
$\text{bus}(i_4, p_2)$

- Normal DB: Query for trams and buses arriving at same station $P$
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$  
$\text{tram}(i_1, p_1)$  
$\text{tram}(i_3, p_2)$  
$\text{bus}(i_4, p_2)$

Normal DB: Query for trams and buses arriving at same station $P$
Answer: $i_1, i_2, p_1$
Example: Trams and buses

Arrival times at different stations $p_i$

- $bus(i_2, p_1)$
- $tram(i_1, p_1)$
- $tram(i_3, p_2)$
- $bus(i_4, p_2)$

Normal DB: Query for trams and buses arriving at same station $P$

Answer: $i_1, i_2, p_1$ and $i_3, i_4, p_2$
Example: Trams and buses

Arrival times at different stations $p_i$

$bus(i_2, p_1)$  
$tram(i_1, p_1)$  
$tram(i_3, p_2)$ $bus(i_4, p_2)$

- Normal DB: Query for trams and buses arriving at same station $P$
  Answer: $i_1, i_2, p_1$ and $i_3, i_4, p_2$

- SQL

```
SELECT * FROM tram, bus  
WHERE tram.P = bus.P
```
Example: Trams and buses

Arrival times at different stations $p_i$

- $bus(i_2, p_1)$
- $tram(i_1, p_1)$
- $tram(i_3, p_2)$
- $bus(i_4, p_2)$

- Normal DB: Query for trams and buses arriving at same station $P$
  Answer: $i_1, i_2, p_1$ and $i_3, i_4, p_2$

- SQL

```
SELECT * FROM tram, bus
WHERE tram.P = bus.P
```

H. Beck (TU Vienna) Towards Ideal Semantics for Analyzing Stream Reasoning ReactKnow’14 2 / 15
Example: Trams and buses

Arrival times at different stations $p_i$

\[
\text{bus}(i_2, p_1) \\
\text{tram}(i_1, p_1) \\
\text{tram}(i_3, p_2) \\
\text{bus}(i_4, p_2)
\]

- Normal DB: Query for trams and buses arriving at same station $P$
  Answer: $i_1, i_2, p_1$ and $i_3, i_4, p_2$

- SQL

```
SELECT * FROM tram, bus
WHERE tram.P = bus.P
```
Example: Trams and buses

Arrival times at different stations \( p_i \)

\[
\begin{align*}
\text{bus}(i_2, p_1) & \quad \text{tram}(i_1, p_1) \\
\text{tram}(i_3, p_2) & \quad \text{bus}(i_4, p_2)
\end{align*}
\]

- Normal DB: Query for trams and buses arriving at same station \( P \)
  Answer: \( i_1, i_2, p_1 \) and \( i_3, i_4, p_2 \)

- SQL

```sql
SELECT * FROM tram, bus
WHERE tram.P = bus.P
```
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$
$\text{tram}(i_3, p_2)$
$\text{bus}(i_4, p_2)$

- Stream setting, at time 13: Query for
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$
$\text{tram}(i_3, p_2)$
$\text{bus}(i_4, p_2)$

Stream setting, at time 13: Query for Trams and buses arriving at same station $P$
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$

$\text{tram}(i_3, p_2)$  $\text{bus}(i_4, p_2)$

Stream setting, at time 13: Query for

- Trams and buses arriving at same station $P$ within the last 5 min
Example: Trams and buses

Arrival times at different stations $p_i$

$bus(i_2, p_1)$  
$tram(i_1, p_1)$  
$tram(i_3, p_2)$  
$bus(i_4, p_2)$

- Stream setting, at time 13: Query for
- Trams and buses arriving at same station $P$ within the last 5 min
Answer: $i_3, i_4, p_2$
Example: Trams and buses

Arrival times at different stations \( p_i \)

- \( tram(i_1, p_1) \)
- \( bus(i_2, p_1) \)
- \( tram(i_3, p_2) \)
- \( bus(i_4, p_2) \)

Stream setting, at time 13: Query for

- Trams and buses arriving at same station \( P \) within the last 5 min

Answer: \( i_3, i_4, p_2 \)

CQL

```
SELECT * FROM tram [RANGE 5], bus [RANGE 5]
WHERE tram.P = bus.P
```
Example: Trams and buses

Arrival times at different stations $p_i$

$bus(i_2, p_1)$  
$tram(i_1, p_1)$  
$tram(i_3, p_2)$  
$bus(i_4, p_2)$

- Trams and buses arriving at same station $P$ within the last 5 min at the same time

$\textit{Answer}:

$i_1, i_2, p_1 \text{ for query times } 2, ..., 7$

$\textit{CQL: Not expressible in single query (Snapshot semantics)}$

$\text{SELECT * AS tram, bus FROM tram \[NOW\], bus \[NOW\]}$

$\text{WHERE tram.P = bus.P}$

$\text{SELECT * FROM tram, bus \[RANGE 5\]}$
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$
$\text{tram}(i_3, p_2)$
$\text{bus}(i_4, p_2)$

Trams and buses arriving at same station $P$ within the last 5 min at the same time

Answer: –
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$  $\text{tram}(i_1, p_1)$  $\text{tram}(i_3, p_2)$  $\text{bus}(i_4, p_2)$

Trams and buses arriving at same station $P$ within the last 5 min at the same time

Answer: $i_1, i_2, p_1$ for query times 2, \ldots, 7
Example: Trams and buses

Arrival times at different stations $p_i$

- $bus(i_2, p_1)$
- $tram(i_1, p_1)$
- $tram(i_3, p_2)$
- $bus(i_4, p_2)$

Trams and buses arriving at the same station $P$ within the last 5 min at the same time

Answer: $i_1$, $i_2$, $p_1$ for query times 2, ..., 7

- **CQL**: Not expressible in single query (Snapshot semantics)

```
SELECT * AS tram_bus FROM tram [NOW], bus [NOW]
WHERE tram.P = bus.P
```
Example: Trams and buses

Arrival times at different stations $p_i$

- $bus(i_2, p_1)$
- $tram(i_1, p_1)$

- Trams and buses arriving at same station $P$ within the last 5 min at the same time
  Answer: $i_1, i_2, p_1$ for query times 2, ..., 7

- **CQL**: Not expressible in single query (Snapshot semantics)

  ```sql
  SELECT * AS tram_bus FROM tram [NOW], bus [NOW] WHERE tram.P = bus.P
  ```
Example: Trams and buses

Arrival times at different stations $p_i$

- $\text{bus}(i_2, p_1)$
- $\text{tram}(i_1, p_1)$

Trams and buses arriving at same station $P$ within the last 5 min at the same time

Answer: $i_1, i_2, p_1$ for query times 2, ..., 7

CQL: Not expressible in single query (Snapshot semantics)

```sql
SELECT * AS tram_bus FROM tram [NOW], bus [NOW]
WHERE tram.P = bus.P
```

```sql
SELECT * FROM tram_bus [RANGE 5]
```
Example: Trams and buses

Arrival times at different stations $p_i$

$\text{bus}(i_2, p_1)$
$\text{tram}(i_1, p_1)$

Trams and buses arriving at same station $P$ within the last 5 min at the same time
Answer: $i_1, i_2, p_1$ for query times 2, $\ldots$, 7

CQL: Not expressible in single query (Snapshot semantics)

```sql
SELECT * AS tram_bus FROM tram [NOW], bus [NOW]
WHERE tram.P = bus.P
```

```sql
SELECT * FROM tram_bus [RANGE 5]
```
Window Types

- Time-based

Tuple-based:
- Not necessarily unique. E.g.: Last 3 tuples
- Partition-based
  - Apply tuple-based window on substreams
Window Types

- Time-based
- Tuple-based
Window Types

- **Time-based**
- **Tuple-based**
  - Not necessarily unique. E.g.: Last 3 tuples
Window Types

- Time-based
- Tuple-based
  - Not necessarily unique. E.g.: Last 3 tuples
Window Types

- Time-based
- Tuple-based
  - Not necessarily unique. E.g.: Last 3 tuples
- Partition-based
Window Types

- Time-based
- Tuple-based
  - Not necessarily unique. E.g.: Last 3 tuples
- Partition-based
  - Apply tuple-based window on substreams
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”

- Allow for **nesting**: windows within windows
  - As formal counterpart to repeated runs of continuous queries
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”
- Allow for nesting: windows within windows
  - As formal counterpart to repeated runs of continuous queries
- Allow for looking into the future

\[ w(S, t) \rightarrow S' \]

Stream \( S \), time point \( t \in \mathbb{N} \), new stream \( S' \)
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”

- Allow for **nesting**: windows within windows
  - As formal counterpart to repeated runs of continuous queries

- Allow for looking into the **future**

- View window operators as **first class citizens**
  - Do not separate window application (first) from logic (then)
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”
- Allow for nesting: windows within windows
  - As formal counterpart to repeated runs of continuous queries
- Allow for looking into the future
- View window operators as first class citizens
  - Do not separate window application (first) from logic (then)
- Leave open specific underlying window functions
Ideas for Windows

- Example: “In the last hour, did a bus always arrive within 5 min?”
- Allow for nesting: windows within windows
  - As formal counterpart to repeated runs of continuous queries
- Allow for looking into the future
- View window operators as first class citizens
  - Do not separate window application (first) from logic (then)
- Leave open specific underlying window functions
  - \( w(S, t) \mapsto S' \)
  - Stream \( S \), time point \( t \in \mathbb{N} \), new stream \( S' \)
Ideas for Time Reference

- **Atoms** \( a \) appearing in the stream at time points 1, 2, 5
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$.

[Diagram showing a time axis with time points 0 to 6 and atoms at positions 1, 2, and 5, and a query at time 4]
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold . . . now, i.e., exactly at $t$? . . . at time point 2? . . . at some time point $t'$? . . . at all time points $t'$?
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold...
Ideas for Time Reference

▶ Atoms $a$ appearing in the stream at time points $1, 2, 5$
▶ Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold...
...now, i.e., exactly at $t$?
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold...

...now, i.e., exactly at $t$?

$a$
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval [1, 4]

- Example queries: In this window, does $a$ hold... 
  ...now, i.e., exactly at $t$?  
  $a$  
  no
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval [1, 4]

![Diagram showing a window on a timeline with time points 0 to 6 and a window encompassing time points 1 to 4, with an atom $a$ appearing at time points 1, 2, and 5.]

- Example queries: In this window, does $a$ hold...
  - ...now, i.e., exactly at $t$? $a$ no
  - ...at time point 2?
Ideas for Time Reference

- Atoms \( a \) appearing in the stream at time points 1, 2, 5
- Query time \( t = 4 \). Window on interval \([1, 4]\)

Example queries: In this window, does \( a \) hold...
- ...now, i.e., exactly at \( t \)? \( a \) no
- ...at time point 2? \( @_2 a \)
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

![Diagram showing atoms at time points 1, 2, 5 within the window $[1, 4]$]

- Example queries: In this window, does $a$ hold...
  - ...now, i.e., exactly at $t$? $a$ no
  - ...at time point 2? $\@_2 a$ yes
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold...

...now, i.e., exactly at $t$? $a$ no

...at time point 2? $\text{@}_2 a$ yes

...at some time point $t'$?
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

![Diagram showing time points and atoms]

- Example queries: In this window, does $a$ hold...
  - ...now, i.e., exactly at $t$? $a$ no
  - ...at time point 2? $\@_2 a$ yes
  - ...at some time point $t'$? $\Diamond a$
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold...

- now, i.e., exactly at $t$? $a$ no
- at time point 2? $@_2 a$ yes
- at some time point $t'$? $\checkmark a$ yes
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

![Diagram showing time points and window on interval [1, 4] with an atom appearing at time points 1, 2, and 5, and queries regarding time point 4, 2, and all time points up to 5.]

Example queries: In this window, does $a$ hold...

...now, i.e., exactly at $t$? $a$ no

...at time point 2? $\text{@}_2 a$ yes

...at some time point $t'$? $\text{◊} a$ yes

...at all time points $t'$?
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

![Diagram showing Atoms $a$ appearing at time points 1, 2, 5, with query time $t = 4$.]

- Example queries: In this window, does $a$ hold...
  - ...now, i.e., exactly at $t$? $a$ no
  - ...at time point 2? $\mathbb{@}_2 a$ yes
  - ...at some time point $t'$? $\mathbb{\Diamond} a$ yes
  - ...at all time points $t'$? $\mathbb{\Box} a$
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

Example queries: In this window, does $a$ hold...

- ...now, i.e., exactly at $t$? $a$ no
- ...at time point 2? $@_2 a$ yes
- ...at some time point $t'$? $\diamond a$ yes
- ...at all time points $t'$? $\Box a$ no
Ideas for Time Reference

- Atoms $a$ appearing in the stream at time points 1, 2, 5
- Query time $t = 4$. Window on interval $[1, 4]$

```
0 1 2 3 4 5 6
```

- Example queries: In this window, does $a$ hold...
  - ...now, i.e., exactly at $t$? $a$ no
  - ...at time point 2? $a_2$ yes
  - ...at some time point $t'$? $a$ yes
  - ...at all time points $t'$? $a$ no
Streams

Stream $S = (T, \nu)$, where
Streams

Stream $S = (T, \nu)$, where

- $T$: interval in $\mathbb{N}$
Streams

- Stream $S = (T, \nu)$, where
  - $T$: interval in $\mathbb{N}$
  - $\nu : T \rightarrow 2^{\mathcal{G}}$ (interpretation of ground atoms $\mathcal{G}$)
Streams

Stream $S = (T, \nu)$, where

- $T$: interval in $\mathbb{N}$
- $\nu: T \rightarrow 2^G$ (interpretation of ground atoms $G$)

Example
Streams

Stream $S = (T, \nu)$, where

- $T$: interval in $\mathbb{N}$
- $\nu : T \rightarrow 2^G$ (interpretation of ground atoms $G$)

Example

- $T = [0, 13]$
Streams

\[ \text{Stream } S = (T, \nu), \text{ where} \]

\[ T: \text{ interval in } \mathbb{N} \]

\[ \nu: T \rightarrow 2^G \] (interpretation of ground atoms \( G \))

\[ \text{Example} \]

\[ T = [0, 13] \]

\[ \nu = \left\{ 2 \mapsto \{ \text{tram}(i_1, p_1), \text{bus}(i_2, p_1) \} \right\} \]
Streams

Stream $S = (T, \nu)$, where

- $T$: interval in $\mathbb{N}$
- $\nu: T \rightarrow 2^G$ (interpretation of ground atoms $G$)

Example

- $T = [0, 13]$
- $\nu = \left\{ 2 \mapsto \{\text{tram}(i_1, p_1), \text{bus}(i_2, p_1)\}, \ 8 \mapsto \{\text{tram}(i_3, p_2)\} \right\}$
Streams

Stream $S = (T, \nu)$, where

- $T$: interval in $\mathbb{N}$
- $\nu : T \rightarrow 2^G$ (interpretation of ground atoms $G$)

Example

- $T = [0, 13]$
- $\nu = \begin{cases} 
2 \mapsto \{ \text{tram}(i_1, p_1), \text{bus}(i_2, p_1) \}, \\
8 \mapsto \{ \text{tram}(i_3, p_2) \}, \\
11 \mapsto \{ \text{bus}(i_4, p_2) \}
\end{cases}$
Streams

Stream \( S = (T, \nu) \), where

- \( T \): interval in \( \mathbb{N} \)
- \( \nu : T \rightarrow 2^G \) (interpretation of ground atoms \( G \))

Example

- \( T = [0, 13] \)
- \( \nu = \begin{cases} 2 \mapsto \{tram(i_1, p_1), bus(i_2, p_1)\}, & 8 \mapsto \{tram(i_3, p_2)\}, \\ 11 \mapsto \{bus(i_4, p_2)\}, & i \mapsto \emptyset \quad \text{else} \end{cases} \)
Formulas

- Formulas defined by the grammar (atom $a$, $t \in \mathbb{N}$ timepoint)

$$\alpha ::=$$
Formulas

Formulas defined by the grammar \((\text{atom } a, t \in \mathbb{N} \text{ timepoint})\)

\[ \alpha ::= a | \neg \alpha | \alpha \land \alpha | \alpha \lor \alpha | \alpha \rightarrow \alpha \]
Formulas

Formulas defined by the grammar (atom $a$, $t \in \mathbb{N}$ timepoint)

$$\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \diamond \alpha \mid \Box \alpha \mid @_t \alpha$$
Formulas

- Formulas defined by the grammar (atom $a, t \in \mathbb{N}$ timepoint)

$$\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \Diamond \alpha \mid \Box \alpha \mid \mathtt{@}_t \alpha \mid \mathtt{⊞}_i \alpha$$

- $\mathtt{⊞}_i$ window operator: change view on stream
Formulas

- **Formulas defined by the grammar** (atom $a, t \in \mathbb{N}$ timepoint)

\[ \alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \diamond \alpha \mid \square \alpha \mid @_i \alpha \mid \mathbb{I}_i \alpha \]

- **$\mathbb{I}_i$ window operator**: change view on stream
  - Utilizing window function with identifier $i$
Formulas

- Formulas defined by the grammar (atom $a, t \in \mathbb{N}$ timepoint)

$$\alpha ::= a | \neg \alpha | \alpha \land \alpha | \alpha \lor \alpha | \alpha \rightarrow \alpha | \diamond \alpha | \square \alpha | \oplus_i \alpha$$

- $\oplus_i$ window operator: change view on stream
  - Utilizing window function with identifier $i$
  - Change considered substream based on current time point, and
    - current window, or
    - original stream
Formulas

- Formulas defined by the grammar (atom $a$, $t \in \mathbb{N}$ timepoint)

$$\alpha ::= a | \neg \alpha | \alpha \land \alpha | \alpha \lor \alpha | \alpha \rightarrow \alpha | \Diamond \alpha | \square \alpha | @t \alpha | \boxdot_i \alpha$$

- $\boxdot_i$ window operator: change view on stream
  - Utilizing window function with identifier $i$
  - Change considered substream based on current time point, and
    - current window, or
    - original stream
  - Window operator = window function + stream choice function
Formulas

- Formulas defined by the grammar (atom \( a, t \in \mathbb{N} \) timepoint)

\[
\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \lozenge \alpha \mid \square \alpha \mid \Diamond t \alpha \mid \Box i \alpha
\]

- **\( \Box i \)** window operator: change view on stream
  - Utilizing window function with identifier \( i \)
  - Change considered substream based on current time point, and
    - current window, or
    - original stream
  - Window operator = window function + stream choice function
  - Why keep the original stream?
Nested Windows and Stream Choice

▶ “For the last two trams, did a bus always appear within 5 min?”

<table>
<thead>
<tr>
<th>bus</th>
<th>tram</th>
<th>tram</th>
<th>bus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>
Nested Windows and Stream Choice

▶ “For the last two trams, did a bus always appear within 5 min?”

▶ Partition-based window
Nested Windows and Stream Choice

▶ “For the last two trams, did a bus always appear within 5 min?”

Partition-based window
▶ Partition stream into substreams: trams vs. buses
Nested Windows and Stream Choice

▶ “For the last two trams, did a bus always appear within 5 min?”

tram tram

2 8 13

▶ Partition-based window
  ▶ Partition stream into substreams: trams vs. buses
  ▶ Apply tuple-based windows on substreams: 2 trams, 0 buses
Nested Windows and Stream Choice

▶ “For the last two trams, did a bus always appear within 5 min?”

▶ Partition-based window
  ▶ Partition stream into substreams: trams vs. buses
  ▶ Apply tuple-based windows on substreams: 2 trams, 0 buses

▶ In the new view, buses are invisible
Nested Windows and Stream Choice

- “For the last two trams, did a bus always appear within 5 min?”

- **Partition-based window**
  - Partition stream into substreams: trams vs. buses
  - Apply tuple-based windows on substreams: 2 trams, 0 buses

- In the new view, buses are invisible

- $\Rightarrow$ For “within 5 min” window: use data of original stream again
Nested Windows and Stream Choice

“For the last two trams, did a bus always appear within 5 min?"

Partition-based window
- Partition stream into substreams: trams vs. buses
- Apply tuple-based windows on substreams: 2 trams, 0 buses

In the new view, buses are invisible

⇒ For “within 5 min” window: use data of original stream again
Semantics: Structure

- Structure $M = \langle T, v, \hat{W} \rangle$, where
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
  - $\hat{W}$ mapping from identifiers (in $\mathbb{N}$) to \textit{extended} window functions
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
  - $\hat{W}$ mapping from identifiers (in $\mathbb{N}$) to \textit{extended} window functions
  - choice function $ch(S_1, S_2) \mapsto S'$

$$\hat{w}(S_1, S_2, t) = w(ch(S_1, S_2), t)$$
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
  - $\hat{W}$ mapping from identifiers (in $\mathbb{N}$) to \textit{extended} window functions
  - choice function $ch(S_1, S_2) \mapsto S'$

  $$\hat{w}(S_1, S_2, t) = w(ch(S_1, S_2), t)$$

- Example
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
  - $\hat{W}$ mapping from identifiers (in $\mathbb{N}$) to extended window functions
  - choice function $ch(S_1, S_2) \mapsto S'$
    
    $$\hat{w}(S_1, S_2, t) = w(ch(S_1, S_2), t)$$

- Example
  - $w^5$ time-based window for last 5 minutes
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
  - $\hat{W}$ mapping from identifiers (in $\mathbb{N}$) to extended window functions
  - choice function $ch(S_1, S_2) \mapsto S'$
    \[
    \hat{w}(S_1, S_2, t) = w(ch(S_1, S_2), t)
    \]

- Example
  - $w^5$ time-based window for last 5 minutes
  - $ch_2$ choice that selects the second stream ($ch_2(S_1, S_2) = S_2$)
Semantics: Structure

- Structure $M = \langle T, \nu, \hat{W} \rangle$, where
  - $(T, \nu)$ original stream
  - $\hat{W}$ mapping from identifiers (in $\mathbb{N}$) to extended window functions
  - choice function $ch(S_1, S_2) \mapsto S'$
    
    $$\hat{w}(S_1, S_2, t) = w(ch(S_1, S_2), t)$$

- Example
  - $w^5$ time-based window for last 5 minutes
  - $ch_2$ choice that selects the second stream ($ch_2(S_1, S_2) = S_2$)
  - $\hat{W}(1) = \hat{w}^5$, where $\hat{w}^5(S_1, S_2, t) = w^5(S_2, t)$
Semantics: Entailment

- Structure \( M = \langle T, \nu, \hat{W} \rangle \) with original stream \( S_M = (T, \nu) \)
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
- Entailment between $M, S, t$ and formulas $\alpha, \beta$
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
- Entailment between $M, S, t$ and formulas $\alpha, \beta$

$$M, S, t \models a \iff a \in \nu_S(t),$$
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
- Entailment between $M, S, t$ and formulas $\alpha, \beta$

$M, S, t \models a$ iff $a \in \nu_S(t)$,

$M, S, t \models \neg \alpha$ iff $M, S, t \not\models \alpha$,

$M, S, t \models \alpha \land \beta$ iff $M, S, t \models \alpha$ and $M, S, t \models \beta$,

$M, S, t \models \alpha \lor \beta$ iff $M, S, t \models \alpha$ or $M, S, t \models \beta$,

$M, S, t \models \alpha \rightarrow \beta$ iff $M, S, t \not\models \alpha$ or $M, S, t \models \beta$. 
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
- Entailment between $M, S, t$ and formulas $\alpha, \beta$

$M, S, t \models a$ \quad iff \quad a \in \nu_S(t),$

$M, S, t \models \neg \alpha$ \quad iff \quad $M, S, t \not\models \alpha,$

$M, S, t \models \alpha \land \beta$ \quad iff \quad $M, S, t \models \alpha$ and $M, S, t \models \beta,$

$M, S, t \models \alpha \lor \beta$ \quad iff \quad $M, S, t \models \alpha$ or $M, S, t \models \beta,$

$M, S, t \models \alpha \rightarrow \beta$ \quad iff \quad $M, S, t \not\models \alpha$ or $M, S, t \models \beta,$

$M, S, t \models \diamond \alpha$ \quad iff \quad $M, S, t' \models \alpha$ for some $t' \in T_S,$
Semantics: Entailment

- Structure \( M = \langle T, \nu, \hat{W} \rangle \) with original stream \( S_M = (T, \nu) \)
- Substream \( S = (T_S, \nu_S) \) of \( S_M \): currently considered window
- Time point \( t \in T_S \) (query time)
- Entailment between \( M, S, t \) and formulas \( \alpha, \beta \)
  
  \[ M, S, t \models \alpha \quad \text{iff} \quad \alpha \in \nu_S(t), \]
  \[ M, S, t \models \neg \alpha \quad \text{iff} \quad M, S, t \notmodels \alpha, \]
  \[ M, S, t \models \alpha \land \beta \quad \text{iff} \quad M, S, t \models \alpha \text{ and } M, S, t \models \beta, \]
  \[ M, S, t \models \alpha \lor \beta \quad \text{iff} \quad M, S, t \models \alpha \text{ or } M, S, t \models \beta, \]
  \[ M, S, t \models \alpha \rightarrow \beta \quad \text{iff} \quad M, S, t \notmodels \alpha \text{ or } M, S, t \models \beta, \]
  \[ M, S, t \models \diamond \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ for some } t' \in T_S, \]
  \[ M, S, t \models \square \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ for all } t' \in T_S, \]
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
- Entailment between $M, S, t$ and formulas $\alpha, \beta$

$$
\begin{align*}
M, S, t &\notmodels a \quad \text{iff} \quad a \notin \nu_S(t), \\
M, S, t &\notmodels \neg \alpha \quad \text{iff} \quad M, S, t \notmodels \alpha, \\
M, S, t &\models \alpha \land \beta \quad \text{iff} \quad M, S, t \models \alpha \text{ and } M, S, t \models \beta, \\
M, S, t &\models \alpha \lor \beta \quad \text{iff} \quad M, S, t \models \alpha \text{ or } M, S, t \models \beta, \\
M, S, t &\models \alpha \rightarrow \beta \quad \text{iff} \quad M, S, t \notmodels \alpha \text{ or } M, S, t \models \beta, \\
M, S, t &\notmodels \diamond \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ for some } t' \in T_S, \\
M, S, t &\notmodels \Box \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ for all } t' \in T_S, \\
M, S, t &\models @_t \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ and } t' \in T_S,
\end{align*}
$$
Semantics: Entailment

- Structure $M = \langle T, \nu, \hat{W} \rangle$ with original stream $S_M = (T, \nu)$
- Substream $S = (T_S, \nu_S)$ of $S_M$: currently considered window
- Time point $t \in T_S$ (query time)
- Entailment between $M, S, t$ and formulas $\alpha, \beta$

$M, S, t \vdash \alpha \quad \text{iff} \quad \alpha \in \nu_S(t)$,
$M, S, t \vdash \neg \alpha \quad \text{iff} \quad M, S, t \not\models \alpha,$
$M, S, t \vdash \alpha \land \beta \quad \text{iff} \quad M, S, t \vdash \alpha \quad \text{and} \quad M, S, t \vdash \beta,$
$M, S, t \vdash \alpha \lor \beta \quad \text{iff} \quad M, S, t \not\models \alpha \quad \text{or} \quad M, S, t \models \beta,$
$M, S, t \vdash \alpha \rightarrow \beta \quad \text{iff} \quad M, S, t \not\models \alpha \quad \text{or} \quad M, S, t \models \beta,$
$M, S, t \vdash \Diamond \alpha \quad \text{iff} \quad M, S, t' \models \alpha \quad \text{for some} \quad t' \in T_S,$
$M, S, t \vdash \Box \alpha \quad \text{iff} \quad M, S, t' \models \alpha \quad \text{for all} \quad t' \in T_S,$
$M, S, t \vdash @_{t'} \alpha \quad \text{iff} \quad M, S, t' \models \alpha \quad \text{and} \quad t' \in T_S,$
$M, S, t \vdash \Box_i \alpha \quad \text{iff} \quad M, S', t \models \alpha \quad \text{where} \quad S' = \hat{w}_i(S_M, S, t).$
Queries

- Query $\alpha[t]: \{ M, S_M, t \} \models \alpha$?
Queries

- Query $\alpha[t]$: "$M, S_M, t \models \alpha$"?

```
tram(i_1, p_1)  bus(i_2, p_1)  tram(i_3, p_2)  bus(i_4, p_2)
```

<table>
<thead>
<tr>
<th>0</th>
<th>2</th>
<th>8</th>
<th>11</th>
<th>13</th>
</tr>
</thead>
</table>

Queries

- Query $\alpha[t]$: "$M, S_M, t \vdash \alpha$"?

$$
\begin{array}{c}
bus(i_2, p_1) \\
tram(i_1, p_1) \\
tram(i_3, p_2) \\
bus(i_4, p_2)
\end{array}
$$

$M, S_M, 13 \vdash bus(i_2, p_1)$?
Queries

- Query $\alpha[t]$: "$M, S_M, t \vdash \alpha$"?

$M, S_M, 13 \not\models bus(i_2, p_1)$, since $bus(i_2, p_1) \not\in \nu(13)$
Queries

- Query $\alpha[t]$: “$M, S_M, t \models \alpha$”? 

$$
\begin{aligned}
\text{bus}(i_2, p_1) \\
\text{tram}(i_1, p_1) \\
\text{tram}(i_3, p_2) \quad \text{bus}(i_4, p_2)
\end{aligned}
$$

$M, S_M, 13 \models \Diamond \text{bus}(i_2, p_1)$?
Queries

- Query $\alpha[t]: "M, S_M, t \models \alpha"$?

$$
\begin{align*}
\text{bus}(i_2, p_1) \\
\text{tram}(i_1, p_1) \\
\text{tram}(i_3, p_2) \quad \text{bus}(i_4, p_2)
\end{align*}
$$

$M, S_M, 13 \models \Diamond \text{bus}(i_2, p_1)$, since $\exists t' \in T_{S_M}$ s.t. $\text{bus}(i_2, p_1) \in \nu(t')$
Queries

- Query $\alpha[t]$: “$M, S_M, t \models \alpha$"? $\square_1$: last 5 min

\[
\begin{align*}
\text{bus}(i_2, p_1) & \quad \text{tram}(i_1, p_1) \\
\text{tram}(i_3, p_2) & \quad \text{bus}(i_4, p_2)
\end{align*}
\]

$M, S_M, 13 \models \square_1 \Diamond \text{bus}(i_2, p_1)$?
Queries

- Query $\alpha[t]$: “$M, S_M, t \models \alpha$”?
  - $\square_1$: last 5 min

$M, S_M, 13 \not\models \square_1 \diamond bus(i_2, p_1)$
Queries

- Query $\alpha[t]: \text{"}M, S_M, t \models \alpha\text{"}$?  
  $\mathcal{D}_1$: last 5 min

$M, S_M, 13 \models \mathcal{D}_1 \diamond \text{bus}(i_4, p_2)$
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[
\begin{align*}
bus(i_2, p_1) \\
tram(i_1, p_1) \\
tram(i_3, p_2) \\
bus(i_4, p_2)
\end{align*}
\]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[ \text{bus}(i_2, p_1) \]
\[ \text{tram}(i_1, p_1) \]
\[ \text{tram}(i_3, p_2) \]
\[ \text{bus}(i_4, p_2) \]

\[ M, S_M, 13 \models \mathbb{D}_1 \Diamond \text{bus}(X, P) ? \]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[\begin{align*}
bus(i_2, p_1) \\
tram(i_1, p_1) \\
tram(i_3, p_2) \\
bus(i_4, p_2)
\end{align*}\]

\[M, S_M, 13 \models \blacklozenge_1 \lozenge bus(X, P)？\]

\[X \mapsto i_4, \ P \mapsto p_2\]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[ \begin{align*}
  &bus(i_2, p_1) \\
  &tram(i_1, p_1) \\
  &tram(i_3, p_2) & bus(i_4, p_2) \\
  \end{align*} \]

\[ M, S, U \models \square_1 \Diamond bus(i_2, p_1) ? \]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[
\text{bus}(i_2, p_1) \\
\text{tram}(i_1, p_1) \\
\text{tram}(i_3, p_2) \quad \text{bus}(i_4, p_2)
\]

\[M, S, U \models \blacklozenge_1 \Box_1 \Diamond \text{bus}(i_2, p_1)\]?

\[U \mapsto 2, \ldots, 7\]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[ \text{bus}(i_2, p_1) \]
\[ \text{tram}(i_1, p_1) \]
\[ \text{tram}(i_3, p_2) \]
\[ \text{bus}(i_4, p_2) \]

\[ M, S_M, 13 \models 1 (\Diamond \text{tram}(X, P) \land \Diamond \text{bus}(Y, P)) \]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[
\begin{align*}
\text{bus}(i_2, p_1) & \\
\text{tram}(i_1, p_1) & \\
\text{tram}(i_3, p_2) & \quad \text{bus}(i_4, p_2)
\end{align*}
\]

\[
M, S_M, 13 \models \Box_1 (\Diamond \text{tram}(X, P) \land \Diamond \text{bus}(Y, P))?
\]

\[
X \leftrightarrow i_3, \quad P \leftrightarrow p_2, \quad Y \leftrightarrow i_4
\]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[\text{bus}(i_2, p_1) \quad \text{tram}(i_1, p_1) \quad \text{tram}(i_3, p_2) \quad \text{bus}(i_4, p_2)\]

\[M, S_M, U \models \Box_1 (\text{tram}(X, P) \land \text{bus}(Y, P))?\]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[
\begin{align*}
&\text{bus}(i_2, p_1) \\
&\text{tram}(i_1, p_1) \\
&\text{tram}(i_3, p_2) \quad \text{bus}(i_4, p_2)
\end{align*}
\]

\[
M, S_M, U \models \Box_1 \Diamond (\text{tram}(X, P) \land \text{bus}(Y, P))?
\]

\[
U \leftrightarrow 2, \ldots, 7 \quad \times \quad X \leftrightarrow i_1, \ P \leftrightarrow p_1, \ Y \leftrightarrow i_2
\]
Non-ground Queries

Non-ground query: Assignments s.t. substitution hold

\[ M, S_M, 13 \models \mu_U(\text{tram}(X, P)) \land \text{bus}(Y, P)) \]
Non-ground Queries

- Non-ground query: Assignments s.t. substitution hold

\[ \text{bus}(i_2, p_1) \]
\[ \text{tram}(i_1, p_1) \]
\[ \text{tram}(i_3, p_2) \]
\[ \text{bus}(i_4, p_2) \]

\[ M, S_M, 13 \models \@_U(\text{tram}(X, P)) \land \text{bus}(Y, P) \]?

\[ U \mapsto 2, \quad X \mapsto i_1, \quad P \mapsto p_1, \quad Y \mapsto i_2 \]
Example: Nested Window

“In the last hour, did a bus always appear in the last 5 minutes?”

```
bus  bus  bus
206  213  217  t
```
Example: Nested Window

- “In the last hour, did a bus always appear in the last 5 minutes?”

  ![Diagram]

  - $\Box_i$: time-based window for last $i$ minutes
Example: Nested Window

- “In the last hour, did a bus always appear in the last 5 minutes?”

- $\square_i$: time-based window for last $i$ minutes

- Query: $\square_{60}$

$\text{bus} \quad \text{bus} \quad \text{bus}$

$\begin{array}{c}
206 \\
213 \\
217 \\
t
\end{array}$
Example: Nested Window

- “In the last hour, did a bus always appear in the last 5 minutes?”

- ▶️: time-based window for last $i$ minutes

- Query: $\square_{60}$
Example: Nested Window

- “In the last hour, did a bus always appear in the last 5 minutes?”

- ▶️ $i$: time-based window for last $i$ minutes

- Query: ▶️ $60$ □ ▶️ $5$
Example: Nested Window

“In the last hour, did a bus always appear in the last 5 minutes?”

- $bus_t$: time-based window for last $i$ minutes

- Query: $\preceq_{60} \; \preceq \; \preceq_{5}$
Example: Nested Window

- “In the last hour, did a bus always appear in the last 5 minutes?”

\[ \text{bus} \quad \text{bus} \quad \text{bus} \]
\[ \begin{array}{c}
206 \\
213 \\
217 \\
t
\end{array} \]

- \( \mathcal{W}_i \): time-based window for last \( i \) minutes

- Query: \( \mathcal{W}_{60} \quad \square \quad \mathcal{W}_5 \quad \diamond \quad \text{bus} \)
Example: Nested Window

▶ “In the last hour, did a bus always appear in the last 5 minutes?”

\[
\begin{array}{ccc}
bus(i, p) & bus(j, q) & bus(k, r) \\
206 & 213 & 217 & t
\end{array}
\]

▶ $i$: time-based window for last $i$ minutes

▶ Query: $\begin{array}{c} \lhd 60 \ \square \ \lhd 5 \ \diamond \bus \end{array}$

▶ Limitation: $\begin{array}{c} \lhd 60 \ \square \ \lhd 5 \ \diamond \bus(X, P) \end{array}$
Example: Nested Window

- “In the last hour, did a bus always appear in the last 5 minutes?”

---

<table>
<thead>
<tr>
<th>bus(i, p)</th>
<th>bus(j, q)</th>
<th>bus(k, r)</th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>213</td>
<td>217</td>
</tr>
<tr>
<td>t</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $\Box_i$: time-based window for last $i$ minutes

- Query: $\Box_{60} \square \Box_{5} \diamond \text{bus}$

- Limitation: $\Box_{60} \square \Box_{5} \diamond \text{bus}(X, P)$
  - Result: List of fixed combinations $X, P$
  - Need a rule: $\text{some } \text{bus} \leftarrow \text{bus}(X, P)$
  - Then: $\Box_{60} \square \Box_{5} \diamond \text{some } \text{bus}$
Conclusion Stream

- Past

- $t - k$
Conclusion Stream

\[ ? \models ? \]

- Past: Lack of theoretical underpinning for stream reasoning

\[ t - k \]
Conclusion Stream

? ⪯ ?

\[ t - k \quad t \text{ (now)} \]

- Past: Lack of theoretical underpinning for stream reasoning
- Now
Conclusion Stream

\[ \square (a \land \lozenge b) \]

- **Past:** Lack of theoretical underpinning for stream reasoning
- **Now:** First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction
- **Soon:** Rule-based extension (OrdRing @ ISWC, Oct.'14)
- **Later:** Language properties, capture CQL and ETALIS
- **Eventually:** Distributed setting, heterogeneous nodes
Conclusion Stream

? |= ? ⊢ (a ∧ ◊b)

Past: Lack of theoretical underpinning for stream reasoning

Now: First language for modelling semantics precisely
  ▶ flexible window operator (first class citizen)
  ▶ time reference / time abstraction

Soon
### Conclusion Stream

\[ \text{ ? } \models \text{ ? } \quad \Box (a \land \Diamond b) \quad b \leftarrow a \]

\[
\begin{align*}
    t - k & \\ t \text{ (now)} & \\ t + \varepsilon
\end{align*}
\]

- **Past:** Lack of theoretical underpinning for stream reasoning
- **Now:** First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction
- **Soon:** Rule-based extension (OrdRing @ ISWC, Oct.’14)
Conclusion Stream

- Past: Lack of theoretical underpinning for stream reasoning
- Now: First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction
- Soon: Rule-based extension  (OrdRing @ ISWC, Oct.'14)
- Later
Conclusion Stream

\[
? \models ? \quad \square (a \land \Diamond b) \quad b \leftarrow a \quad \text{CQL, ETALIS properties}
\]

- Past: Lack of theoretical underpinning for stream reasoning
- Now: First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction
- Soon: Rule-based extension (OrdRing @ ISWC, Oct.’14)
- Later: Language properties, capture CQL and ETALIS
Conclusion Stream

- **Past:** Lack of theoretical underpinning for stream reasoning
- **Now:** First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction
- **Soon:** Rule-based extension (OrdRing @ ISWC, Oct.’14)
- **Later:** Language properties, capture CQL and ETALIS
- **Eventually**
Conclusion Stream

Past: Lack of theoretical underpinning for stream reasoning

Now: First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction

Soon: Rule-based extension (OrdRing @ ISWC, Oct.’14)

Later: Language properties, capture CQL and ETALIS

Eventually: Distributed setting, heterogeneous nodes
Conclusion Stream

- Past: Lack of theoretical underpinning for stream reasoning
- Now: First language for modelling semantics precisely
  - flexible window operator (first class citizen)
  - time reference / time abstraction
- Soon: Rule-based extension (OrdRing @ ISWC, Oct.’14)
- Later: Language properties, capture CQL and ETALIS
- Eventually: Distributed setting, heterogeneous nodes
To je ono.

(That’s it.)
Time-based window

▶ Example

- $\ell$: 2 time points into the past
- $u$: 1 time points into the future
- $d$: 3 step size (slide parameter)

- $\circ$: query times $t$
- $\times$: pivot points $t'$

\[ t = 4 \]
Time-based window

- Example: Query time $t = 4$
  - $\ell$ 2 time points into the past
  - $u$ 1 time points into the future
  - $d$ 3 step size (slide parameter)
Appendix

Time-based window

- Example: Query time $t = 4$
  - $\ell$: 2 time points into the past
  - $u$: 1 time point into the future
  - $d$: 3 step size (slide parameter)

- $d$: 3 step size (slide parameter)
- $\bullet$: query times $t$
- $\times$: pivot points $t'$
Time-based window

- Example: Query time $t = 4$
  - $\ell$ 2 time points into the past
  - $u$ 1 time points into the future
  - $d$ 3 step size (slide parameter)

- $\bullet$: query times $t$
- $\times$: pivot points $t'$
Appendix

Time-based window

- Example: Query time $t = 4$
  - $\ell$: 2 time points into the past
  - $u$: 1 time points into the future
  - $d$: 3 step size (slide parameter)

Diagram:
- $\bullet$: query times $t$
- $\times$: pivot points $t'$
Time-based window

- Example: Query time $t = 4$
  - $\ell$ 2 time points into the past
  - $u$ 1 time points into the future
  - $d$ 3 step size (slide parameter)

- $\bullet$: query times $t$
- $\times$: pivot points $t'$
Time-based window

Example: Query time $t = 4$

- $\ell$: 2 time points into the past
- $u$: 1 time points into the future
- $d$: 3 step size (slide parameter)

Diagram showing a time-based window with query times and pivot points.
Time-based window

- Example: Query time $t = 4$
  - $\ell$ 2 time points into the past
  - $u$ 1 time points into the future
  - $d$ 3 step size (slide parameter)

$\bullet$: query times $t$  
$\times$: pivot points $t'$
Time-based window

Example: Query time $t = 4$

- $\ell$: 2 time points into the past
- $u$: 1 time points into the future
- $d$: 3 step size (slide parameter)

•: query times $t$   ×: pivot points $t'$
Time-based window

► Example

\( \ell \) 2 time points into the past

\( u \) 1 time points into the future

\( d \) 3 step size (slide parameter)

![Diagram showing query times and pivot points]

- ●: query times \( t \)
- ×: pivot points \( t' \)