Towards Ideal Semantics for Analyzing Stream Reasoning

Harald Beck  Minh Dao-Tran  Thomas Eiter  Michael Fink

International Workshop on Reactive Concepts in Knowledge Representation 2014

August 19, 2014
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
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
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
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
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
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
“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
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
  - 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 \( p_i \)

\[
\begin{align*}
bus(i_2, p_1) \\
tram(i_1, p_1) \\
tram(i_3, p_2)
\end{align*}
\]
Example: Trams and buses

Arrival times at different stations $p_i$

tram($i_1, p_1$)

bus($i_2, p_1$)

tram($i_3, p_2$)

0  2  8  11
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 $p_i$

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

- Normal DB: Query for
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 $P$
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 $p_i$

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

- 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

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

Arrival times at different stations $p_i$

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

- 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 $p_i$

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

- 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

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

Arrival times at different stations $p_i$

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

$\bullet$ 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 $p_i$

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

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 $p_i$

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

- Stream setting, at time 13: Query for

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

Answer: $i_3, i_4, p_2$
Example: Trams and buses

Arrival times at different stations $p_i$

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

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
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 $p_i$

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

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

0  2  8  11  13

- 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)

```
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 $2, \ldots, 7$

- **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 $p_i$

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

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)

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

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)

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

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

- Time-based

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

Timeline:
- 0  2  8  11  13
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?”
Idea 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
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 $t = 4$. 

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

```
a a • a
```
Ideas for Time Reference

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

![Diagram showing a window on interval $[1, 4]$ with atoms $a$ appearing at time points $1, 2, 5$.]

- 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 $[1, 4]$

![Diagram of time line with points 0, 1, 2, 3, 4, 5, 6 and atom $a$ appearing at 1, 2, 5, with a window on interval [1, 4] highlighted]

- 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 $t$?

...yes

...at some time point $t'$?

...yes

...at all time points $t'$?

...no
Ideas for Time Reference

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

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 \([1, 4]\)

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? $@_2 a$ 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 $[1, 4]$ 

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 $[1, 4]$

![Diagram showing time points 0 to 6 with atoms $a$ at points 1 and 2, and a window from 1 to 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
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? $\mathcal{a}_2 a$ yes
- ...at some time point $t'$? $\diamond a$ 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 $[1, 4]$

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 $t'$? $\Box a$
Ideas for Time Reference

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

![Time Window Diagram]

- 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 $[1, 4]$

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
Streams

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

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

Timeline:

- $0 \rightarrow 2$
- $8 \rightarrow 11 \rightarrow 13$
Streams

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

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

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

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

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

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

Example

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

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

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

Example

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

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

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

\[ \alpha ::= \]

>window operator: change view on stream

Utilizing window function with identifier \(i\)

Change considered substream based on current time point, and original stream

Window operator = window function + stream choice function

Why keep the original stream?
Formulas defined by the grammar (atom $a$, $t \in \mathbb{N}$ timepoint)

$$\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha$$
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)

\[
\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \Diamond \alpha \mid \square \alpha \mid @_t \alpha \mid \Box_i \alpha
\]

- $\Box_i$ window operator: change view on stream
Formulas

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

\[ \alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \diamond \alpha \mid \Box \alpha \mid @_t \alpha \mid @_i \alpha \]

- \(\Box_i\) 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)

$$\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \lozenge \alpha \mid \square \alpha \mid @t \alpha \mid \boxdot_i \alpha$$

- $\boxdot_i$ 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
Formulas

- Formulas defined by the grammar (atom $a, t \in \mathbb{N}$ timepoint)
  \[ \alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \lozenge \alpha \mid \Box \alpha \mid @t \alpha \mid \Box_i \alpha \]

- $\Box_i$ 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
Formulas

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

  $$\alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \diamond \alpha \mid \Box \alpha \mid @_i \alpha \mid \boxdot_i \alpha$$

- $\boxdot_i$ 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
  - Why keep the original stream?
Nested Windows and Stream Choice

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

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

⇒ 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?”

```
bus
tram
tram       bus
2  7 8 11 13
```

▶ 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 $M = \langle T, \nu, \hat{W} \rangle$, 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 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 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 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 \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
  - $w^5$ 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 $\alpha, \beta$

\[
M, S, t \models a \quad \text{iff} \quad a \in \nu_S(t),
\]
\[
M, S, t \not\models \neg \alpha \quad \text{iff} \quad M, S, t \not\models \alpha,
\]
\[
M, S, t \models \alpha \land \beta \quad \text{iff} \quad M, S, t \models \alpha \quad \text{and} \quad M, S, t \models \beta,
\]
\[
M, S, t \models \alpha \lor \beta \quad \text{iff} \quad M, S, t \models \alpha \quad \text{or} \quad M, S, t \models \beta,
\]
\[
M, S, t \models \alpha \rightarrow \beta \quad \text{iff} \quad M, S, t \not\models \alpha \quad \text{or} \quad M, S, t \models \beta,
\]
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 \( \alpha, \beta \)

\[
M, S, t \models a \quad \text{iff} \quad a \in \nu_S(t),
\]
\[
M, S, t \models \neg \alpha \quad \text{iff} \quad M, S, t \not\models \alpha,
\]
\[
M, S, t \models \alpha \land \beta \quad \text{iff} \quad M, S, t \models \alpha \quad \text{and} \quad M, S, t \models \beta,
\]
\[
M, S, t \models \alpha \lor \beta \quad \text{iff} \quad M, S, t \models \alpha \quad \text{or} \quad M, S, t \models \beta,
\]
\[
M, S, t \models \alpha \rightarrow \beta \quad \text{iff} \quad M, S, t \not\models \alpha \quad \text{or} \quad M, S, t \models \beta,
\]
\[
M, S, t \models \lozenge \alpha \quad \text{iff} \quad M, S, t' \models \alpha \quad \text{for some} \quad t' \in T_S,
\]
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 $\alpha, \beta$

\[
M, S, t \models a \quad \text{iff} \quad a \in \nu_S(t),
\]
\[
M, S, t \models \neg \alpha \quad \text{iff} \quad M, S, t \notmodels \alpha,
\]
\[
M, S, t \models \alpha \land \beta \quad \text{iff} \quad M, S, t \models \alpha \text{ and } M, S, t \models \beta,
\]
\[
M, S, t \models \alpha \lor \beta \quad \text{iff} \quad M, S, t \models \alpha \text{ or } M, S, t \models \beta,
\]
\[
M, S, t \models \alpha \to \beta \quad \text{iff} \quad M, S, t \notmodels \alpha \text{ or } M, S, t \models \beta,
\]
\[
M, S, t \models \diamond \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ for some } t' \in T_S,
\]
\[
M, S, t \models \square \alpha \quad \text{iff} \quad M, S, t' \models \alpha \text{ for all } t' \in T_S,
\]
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 $\alpha, \beta$

$M, S, t \not|\not\models a$ iff $a \not\in \nu_S(t)$,

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

$M, S, t \not|\models \alpha \land \beta$ iff $M, S, t \not|\models \alpha$ and $M, S, t \not|\models \beta$,

$M, S, t \not|\models \alpha \lor \beta$ iff $M, S, t \not|\models \alpha$ or $M, S, t \not|\models \beta$,

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

$M, S, t \not|\models \diamond \alpha$ iff $M, S, t' \not|\models \alpha$ for some $t' \in T_S$,

$M, S, t \not|\models \Box \alpha$ iff $M, S, t' \not|\models \alpha$ for all $t' \in T_S$,

$M, S, t \not|\models @_{t'} \alpha$ iff $M, S, t' \not|\models \alpha$ and $t' \in T_S$.
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 $\alpha, \beta$

$M, S, t \models a$ 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 $M, S, t \models \beta$,

$M, S, t \models \diamond \alpha$ iff $M, S, t' \models \alpha$ for some $t' \in T_S$,

$M, S, t \models \Box \alpha$ iff $M, S, t' \models \alpha$ for all $t' \in T_S$,

$M, S, t \models @_{t'} \alpha$ iff $M, S, t' \models \alpha$ and $t' \in T_S$,

$M, S, t \models \circledast_i \alpha$ iff $M, S', t \models \alpha$ where $S' = \hat{w}_i(S_M, S, t)$. 
Queries

- Query \( \alpha[t] \): \( M, S_M, t \models \alpha \)?
Queries

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

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

Timeline:

- 0
- 2
- 8
- 11
- 13
Queries

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

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

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

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

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

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

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

\[ \begin{array}{cccc}
0 & 2 & 8 & 11 & 13 \\
\end{array} \]

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

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

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

\[\begin{array}{cccc}
0 & 2 & 8 & 11 & 13
\end{array}\]

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

- Query $\alpha[t]$: "$M, S_M, t \models \alpha$"?  $\Box_1$: last 5 min

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

$M, S_M, 13 \models \Box_1 \Diamond \text{bus}(i_2, p_1)$?
Queries

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

$\diamondsuit_1$: last 5 min

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

$M, S_M, 13 \not\models \Box_1 \diamondsuit bus(i_2, p_1)$
Queries

- Query $\alpha[t]$: “$M, S_M, t \models \alpha$”?  
  $\Box_1$: last 5 min

$M, S_M, 13 \models \Box_1 \Diamond bus(i_4, p_2)$
Non-ground Queries

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

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

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

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

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

\[
\begin{array}{cccc}
0 & 2 & 8 & 11 & 13
\end{array}
\]

\[
M, S_M, 13 \models \Box_1 \Diamond bus(X, P)\
\]
Non-ground Queries

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

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

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

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

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

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

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

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

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

\[ M, S, U \models □_1 \diamond \text{bus}(i_2, p_1) ? \]

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

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

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

\[
\begin{align*}
0 & \quad 2 & \quad 8 & \quad 9 & \quad 10 & \quad 11 & \quad 12 & \quad 13
\end{align*}
\]

\[
M, S_M, 13 \models \Box_1 (\diamond tram(X, P) \land \diamond bus(Y, P))?
\]
Non-ground Queries

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

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

\( M, S_M, 13 \models \square_1 (\Diamond \text{tram}(X, P) \land \Diamond \text{bus}(Y, P)) \)?
Non-ground Queries

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

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

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

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

\[
\begin{align*}
bus(i_2, p_1) & \\
tram(i_1, p_1) & \\
tram(i_3, p_2) & \\
bus(i_4, p_2) & \\
non-ground query: Assignments s.t. substitution hold
\end{align*}
\]

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

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

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

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

\[M, S_M, 13 \models @_U(tram(X, P)) \land bus(Y, P))?\]
Non-ground Queries

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

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

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

\[ U \mapsto 2, \quad X \mapsto i_1, \quad P \mapsto p_1, \quad Y \mapsto i_2 \]
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?”

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

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

<table>
<thead>
<tr>
<th>bus</th>
<th>bus</th>
<th>bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>213</td>
<td>217</td>
</tr>
</tbody>
</table>

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

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

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

- \( \square_i \): 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?”

\[
\begin{aligned}
\text{bus} & & \text{bus} & & \text{bus} \\
206 & & 213 & & 217 & & t
\end{aligned}
\]

- \( \Box_i \): time-based window for last \( i \) minutes

- Query: \( \Box_{60} \) \( \square \) \( \Box_5 \)
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: $\square_{60} \square \square_{5}$
Example: Nested Window

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

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

- Query: $\Box_{60} \Box_5 \Box bus$
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>
</tbody>
</table>

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

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

- Limitation: $\square_{60} \square \square_{5} \diamond bus(X, P)$
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>

- $⊞_i$: time-based window for last $i$ minutes

- Query: $⊞_{60} □ ⊞_5 \diamond bus$

- Limitation: $⊞_{60} □ ⊞_5 \diamond bus(X, P)$
  - Result: List of fixed combinations $X, P$
  - Need a rule: $some\_bus \leftarrow bus(X, P)$
  - Then: $⊞_{60} □ ⊞_5 \diamond some\_bus$
Conclusion Stream

$t - k$

- Past
Conclusion Stream

? |= ?

Past: Lack of theoretical underpinning for stream reasoning
Conclusion Stream

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

\[ ? \models ? \quad \square (a \land \Diamond b) \]

- **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

\[
\begin{align*}
\begin{array}{ccc}
? & = ? & \Box (a \land \Diamond b) \\
\hline
(t - k) & (t \text{ (now)}) & (t + \varepsilon)
\end{array}
\end{align*}
\]

- **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**
Conclusion Stream

\(? \models ?\)  \quad \Box (a \land \Diamond b)  \quad b \leftarrow a

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

- **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

\[ ? \models ? \quad \square (a \land \diamond b) \quad b \leftarrow a \]

- **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

- 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)

\[ t \times \]

\( t \times \) query times
\( t' \times \) 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)

\[
\begin{align*}
0 & \quad 1 & \quad 2 & \quad 3 & \quad 4 & \quad 5 & \quad 6 & \quad 7 & \quad 8 & \quad 9 \\
& & & & \bullet & & & & \times & \\
\end{align*}
\]

- \( \bullet \): 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)

$\bullet$: 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)

- $\circ$: 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 $t'$

---

H. Beck (TU Vienna) Towards Ideal Semantics for Analyzing Stream Reasoning ReactKnow'14
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: 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: Query time $t = 4$
  - $\ell$ 2 time points into the past
  - $u$ 1 time points into the future
  - $d$ 3 step size (slide parameter)
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](image)