Towards a Logic-Based Framework for Analyzing Stream Reasoning

Harald Beck  Minh Dao-Tran  Thomas Eiter  Michael Fink

3rd International Workshop on Ordering and Reasoning
October 20, 2014
What & Why

“Towards a Logic-Based Framework for Analyzing Stream Reasoning”

Stream Reasoning
What & Why

“Towards a Logic-Based Framework for Analyzing Stream Reasoning”

- **Stream Reasoning**: Logical reasoning on streaming data
What & Why

“Towards a Logic-Based Framework for Analyzing Stream Reasoning”

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

“Towards a Logic-Based Framework for Analyzing Stream Reasoning”

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

- Logic-Based
"Towards a Logic-Based Framework for Analyzing Stream Reasoning"

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

- Logic-Based: Lack of theory
“Towards a Logic-Based Framework for Analyzing Stream Reasoning”

- Stream Reasoning: Logical reasoning on streaming data
  - Streams = tuples (atoms) with timestamps
  - Essential aspect: window functions
- Logic-Based: Lack of theory
- Analysis
“Towards a Logic-Based Framework for Analyzing Stream Reasoning”

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

- Logic-Based: Lack of theory

- Analysis: Hard to predict, hard to compare
Example: Public Transportation Monitoring

Report trams' expected arrival time.
Report good connections between two lines at a given stop.
Example: Public Transportation Monitoring

<table>
<thead>
<tr>
<th>PLAN</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>ℓ₁</td>
<td>p₁</td>
<td>p₃</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>ℓ₂</td>
<td>p₂</td>
<td>p₃</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LINE</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>a₁</td>
<td>ℓ₁</td>
<td></td>
</tr>
<tr>
<td>a₂</td>
<td>ℓ₂</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OLD

| a₁   |
| ...  |

Report trams’ expected arrival time.
Report good connections between two lines at a given stop.
Example: Public Transportation Monitoring

<table>
<thead>
<tr>
<th>PLAN</th>
<th>LINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>$ID$</td>
</tr>
<tr>
<td>$X$</td>
<td>$L$</td>
</tr>
<tr>
<td>$Y$</td>
<td></td>
</tr>
<tr>
<td>$Z$</td>
<td></td>
</tr>
<tr>
<td>$\ell_1$</td>
<td>$a_1$</td>
</tr>
<tr>
<td>$p_1$</td>
<td>$p_3$</td>
</tr>
<tr>
<td>$\ell_2$</td>
<td>$a_2$</td>
</tr>
<tr>
<td>$p_2$</td>
<td>$p_3$</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

OLD

$\ell_1$

$\ell_2$

$p_1$

$p_2$

$p_3$

$\text{tram}(a_1, p_1)\quad \text{tram}(a_2, p_2)$

$36$ $40$

$t$

$\Rightarrow$

Report trams' expected arrival time.

Report good connections between two lines at a given stop.
Example: Public Transportation Monitoring

- Report trams’ expected arrival time.
Example: Public Transportation Monitoring

- Report trams’ expected arrival time.
- Report **good** connections between two lines at a given stop.
Streams

- Data Stream $D = (T, \nu)$

\[
T = [0, 50] \\
\nu = \{36 \mapsto \{\text{tram}(a_1, p_1)\}, 40 \mapsto \{\text{tram}(a_2, p_2)\}\}
\]
Streams

- **Data Stream** $D = (T, \nu)$

  $T = [0, 50]$

  $\nu = \{36 \mapsto \{\text{tram}(a_1, p_1)\}, 40 \mapsto \{\text{tram}(a_2, p_2)\}\}$

- **Interpretation Stream** $S^* = (T^*, \nu^*) \supseteq D$

  $T^* = [0, 50]$

  $\nu^* = \begin{cases} 
  36 \mapsto \{\text{tram}(a_1, p_1)\}, & 40 \mapsto \{\text{tram}(a_2, p_2)\}, \\
  43 \mapsto \{\text{exp}(a_2, p_3)\}, & 44 \mapsto \{\text{exp}(a_1, p_3)\} 
  \end{cases}$
Window Functions

\[ S' = w_t(S, t, \bar{x}) \]
Window Functions

\[ S' = w_\tau(S^*, 40, (1, 5, 1)) = ([39, 45], \begin{cases} 40 \mapsto \{ \text{tram}(a_2, p_2) \}, \\ 43 \mapsto \{ \text{exp}(a_2, p_3) \}, \\ 44 \mapsto \{ \text{exp}(a_1, p_3) \} \end{cases} ) \]
Window Operators

\[ \boxed{\vec{x}}_{l, ch} \iff w_l(ch(S^*, S), t, \vec{x}) \]
Window Operators

$$\square_{\lambda, ch} \xrightarrow{\quad} w_{\lambda}(ch(S^*, S), t, \vec{x})$$

- $ch$: stream choice

$$ch_1(S^*, S) = S^* \quad ch_2(S^*, S) = S$$
Window Operators

\[ \begin{align*} &\begin{array}{c} \left[ x \right]_{\text{ch}} \\ \text{ch} \end{array} \iff w_{\tau}(\text{ch}(S^*, S), t, \vec{x}) \\
\end{align*} \]

- \text{ch}: stream choice

\[ \begin{align*} &\text{ch}_1(S^*, S) = S^* \\
&\text{ch}_2(S^*, S) = S \\
\end{align*} \]

- \[ \begin{align*} &\begin{array}{c} 10 \\ \tau \end{array} = \begin{array}{c} 10, 0, 1 \\ \tau, \text{ch}_2 \end{array} \\
&w_{\tau}(\text{ch}_2(S^*, S), t, (10, 0, 1)) = w_{\tau}(S, t, (10, 0, 1)) \\
\end{align*} \]
Window Operators

\[ \boxed{\vec{x}_{\tau, ch}} \Leftrightarrow w_\tau(ch(S^*, S), t, \vec{x}) \]

- \( ch \): stream choice

\[ ch_1(S^*, S) = S^* \quad \quad ch_2(S^*, S) = S \]

- \( \boxed{+5}_\tau = \boxed{0,5,1}_\tau,ch_2 \) \n  \[ w_\tau(ch_2(S^*, S), t, (0, 5, 1)) = w_\tau(S, t, (0, 5, 1)) \]
Formulas

\[ \alpha ::= \]

\[ a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \lozenge \alpha \mid \Box \alpha \mid @t \alpha \mid \ldots \]


▶ various ways for time references
▶ nesting of window operators

\[ \top \top \tau \Box \top \top \tau \lozenge \]

▶ but need rules: tramAt(P) ← tram(X, P).
Formulas

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

\[ \alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \diamond \alpha \mid \square \alpha \mid \ominus_t \alpha \]

- various ways for time references
Formulas

\[ \alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \lozenge \alpha \mid \Box \alpha \mid \text{@}_t \alpha \mid \begin{array}{c} \mathbf{t} \varepsilon \text{ } \chi \\ \end{array} \alpha \]

- various ways for time references
- nesting of window operators

\[ \begin{array}{c} \begin{array}{c} \begin{array}{c} \mathbf{t} \varepsilon \text{ } \chi \\ \end{array} \end{array} \\ \text{tramAt}(p_1) \end{array} \]
Formulas

\[ \alpha ::= a \mid \neg \alpha \mid \alpha \land \alpha \mid \alpha \lor \alpha \mid \alpha \rightarrow \alpha \mid \diamond \alpha \mid [\alpha] \mid @[t] \alpha \mid x^{\tau}_{\upsilon, ch} \alpha \]

- various ways for time references
- nesting of window operators

\[ \begin{align*}
\begin{matrix}
\text{tramAt}(P_1) \\
\text{tramAt}(P) \leftarrow \text{tram}(X, P).
\end{matrix}
\end{align*} \]
Entailment

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
Entailment

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
- Substream $S = (T, \nu)$ of $S^*$: currently considered window
- Time point $t \in T$
Entailment

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
- Substream $S = (T, \nu)$ of $S^*$: currently considered window
- Time point $t \in T$

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

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
- Substream $S = (T, \nu)$ of $S^*$: currently considered window
- Time point $t \in T$

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

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

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

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

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

7 / 11
Entailment

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
- Substream $S = (T, \nu)$ of $S^*$: currently considered window
- Time point $t \in T$

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

$M, S, t \models \neg \alpha$ 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 $t' \in T$, 
Entailment

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
- Substream $S = (T, \nu)$ of $S^*$: currently considered window
- Time point $t \in T$

$M, S, t \models a$ iff $a \in \nu(t)$,
$M, S, t \models \neg \alpha$ 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 $t' \in T$,
$M, S, t \models \Box \alpha$ iff $M, S, t' \models \alpha$ for all $t' \in T$,
Entailment

- Structure $M = \langle T^*, \nu^*, W, B \rangle$
- Substream $S = (T, \nu)$ of $S^*$: currently considered window
- Time point $t \in T$

$M, S, t \models a$ iff $a \in \nu(t)$,
$M, S, t \models \neg \alpha$ 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 $t' \in T$,
$M, S, t \models \Box \alpha$ iff $M, S, t' \models \alpha$ for all $t' \in T$,
$M, S, t \models \Diamond_t \alpha$ iff $M, S, t' \models \alpha$ and $t' \in T$.
Entailment

- **Structure** $M = \langle T^*, \nu^*, W, B \rangle$
- **Substream** $S = (T, \nu)$ of $S^*$: currently considered window
- **Time point** $t \in T$

$M, S, t \vdash \alpha$ iff $\alpha \in \nu(t)$,
$M, S, t \vdash \neg \alpha$ iff $M, S, t \not\vdash \alpha$,
$M, S, t \vdash \alpha \land \beta$ iff $M, S, t \vdash \alpha$ and $M, S, t \vdash \beta$,
$M, S, t \vdash \alpha \lor \beta$ iff $M, S, t \vdash \alpha$ or $M, S, t \vdash \beta$,
$M, S, t \vdash \alpha \rightarrow \beta$ iff $M, S, t \not\vdash \alpha$ or $M, S, t \vdash \beta$,
$M, S, t \vdash \Diamond \alpha$ iff $M, S, t' \vdash \alpha$ for some $t' \in T$,
$M, S, t \vdash \Box \alpha$ iff $M, S, t' \vdash \alpha$ for all $t' \in T$,
$M, S, t \vdash @_{t'} \alpha$ iff $M, S, t' \vdash \alpha$ and $t' \in T$,
$M, S, t \vdash \Box^\varphi_{\nu, ch} \alpha$ iff $M, S', t \vdash \alpha$ where $S' = w^\varphi_\nu(ch(S^*, S), t, \vec{x})$. 
Entailment: Example

\[ M, S^*, 40 \models \bigotimes^5 \Delta \exp(a_1, p_3) \]
Entailment: Example

\[ M, S^*, 40 \vdash \Box^{\oplus 5}_T \Diamond exp(a_1, p_3) \]

\[ \uparrow \]

\[ M, S, 40 \vdash \Diamond exp(a_1, p_3) \]
Entailment: Example

\[ M, S^*, 40 \models \square^+_\tau \diamond exp(a_1, p_3) \]

↑

\[ M, S, 40 \models \diamond exp(a_1, p_3) \]

↑

\[ M, S, 44 \models exp(a_1, p_3) \]
Rules & Programs

- Rule: $\alpha \leftarrow \beta_1, \ldots, \beta_j$, not $\beta_{j+1}, \ldots$, not $\beta_n$
Rules & Programs

- Rule: $\alpha \leftarrow \beta_1, \ldots, \beta_j, \text{not } \beta_{j+1}, \ldots, \text{not } \beta_n$
- Program: set of rules.
Rules & Programs

- Rule: \( \alpha \leftarrow \beta_1, \ldots, \beta_j, \text{not } \beta_{j+1}, \ldots, \text{not } \beta_n \)

- Program: set of rules.

- Example:

\[
P = \begin{cases} 
@_{T \text{exp}}(ID, Y) & \leftarrow \bigoplus_{p}^{\text{idx},n} @_{T_1} \text{tram}(ID, X), \\
\text{line}(ID, L), \text{plan}(L, X, Y, Z), \\
T = T_1 + Z. 
\end{cases}
\]
Rules & Programs

- Rule: $\alpha \leftarrow \beta_1, \ldots, \beta_j, \text{not } \beta_{j+1}, \ldots, \text{not } \beta_n$

- Program: set of rules.

- Example:

\[
P = \begin{cases} 
\@_{Texp}(ID, Y) & \leftarrow \begin{array}{c} 
\text{id}x,n \@_{T_1} tram(ID, X), \\
\text{line}(ID, L), \text{plan}(L, X, Y, Z), \\
T = T_1 + Z.
\end{array} \\
\@_{Texp}(ID_1, X), \\
\@_T \uparrow^5 \text{exp}(ID_2, X), \\
\text{not } old(ID_2).
\end{cases}
\]
Answers (of a program $P$)

- Data stream: $D = (T, \nu)$
Answers (of a program $P$)

- Data stream: $D = (T, \nu)$
- Interpretation stream: $S^* = (T^*, \nu^*) \supseteq D$
Answers (of a program $P$)

- Data stream: $D = (T, \nu)$
- Interpretation stream: $S^* = (T^*, \nu^*) \supseteq D$
- Interpretation: $M = \langle T^*, \nu^*, W, B \rangle$
Answers (of a program $P$)

- Data stream: $D = (T, \nu)$
- Interpretation stream: $S^* = (T^*, \nu^*) \supseteq D$
- Interpretation: $M = \langle T^*, \nu^*, W, B \rangle$
- Model: $M, t \models P \iff M, S^*, t \models \beta(r) \rightarrow \alpha$, where $\beta(r) = \beta_1 \land \ldots \land \beta_j \land \neg \beta_{j+1} \land \ldots \land \neg \beta_n$. 
Answers (of a program $P$)

- Data stream: $D = (T, \nu)$
- Interpretation stream: $S^* = (T^*, \nu^*) \supseteq D$
- Interpretation: $M = \langle T^*, \nu^*, W, B \rangle$
- Model: $M, t \models P \iff M, S^*, t \models \beta(r) \rightarrow \alpha$, where $\beta(r) = \beta_1 \wedge \ldots \beta_j \wedge -\beta_{j+1} \wedge \ldots \wedge -\beta_n$.
- Minimal model: $\nexists M' = \langle T', \nu', W, B \rangle \subset M$ s.t. $M', t \models P$
Answers (of a program $P$)

- Data stream: $D = (T, \nu)$
- Interpretation stream: $S^* = (T^*, \nu^*) \supset D$
- Interpretation: $M = \langle T^*, \nu^*, W, B \rangle$
- Model: $M, t \models P \iff M, S^*, t \models \beta(r) \rightarrow \alpha$, where $\beta(r) = \beta_1 \land \ldots \land \beta_j \land \neg \beta_{j+1} \land \ldots \land \neg \beta_n$.
- Minimal model: $\exists M' = \langle T', \nu', W, B \rangle \subset M \text{ s.t. } M', t \models P$
- Reduct: $P_{M,t}^M = \{ r \in P \mid M, t \models \beta(r) \}$
Answers (of a program $P$)

- **Data stream**: $D = (T, \nu)$
- **Interpretation stream**: $S^* = (T^*, \nu^*) \supseteq D$
- **Interpretation**: $M = \langle T^*, \nu^*, W, B \rangle$
- **Model**: $M, t \models P \iff M, S^*, t \models \beta(r) \rightarrow \alpha$, where $\beta(r) = \beta_1 \land \ldots \beta_j \land \neg \beta_{j+1} \land \ldots \land \neg \beta_n$.
- **Minimal model**: $\exists M' = \langle T', \nu', W, B \rangle \subseteq M$ s.t. $M', t \models P$
- **Reduct**: $P^{M,t} = \{ r \in P \mid M, t \models \beta(r) \}$
- **Answer**: $M$ is an answer of $P$ (for $D$ at time $t$) iff $M$ is a minimal model of $P^{M,t}$. 
Conclusion Stream

? \models ?

$t - k$

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

\[
\begin{align*}
? & \models ? \\
& \iff \quad c \leftarrow \Box (a \land \Diamond b) \\
\hline \\
t - k & \quad t \ (\text{now})
\end{align*}
\]

- **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
  - rule-based semantics
  - more: capturing a fragment of CQL

Later: Language properties, complexity analysis, capture CQL, ETALIS, C-SPARQL, CQELS,...
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
  - rule-based semantics
  - more: capturing a fragment of CQL
- Later: Language properties, complexity analysis, capture CQL, ETALIS, C-SPARQL, CQELS,...
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
  - rule-based semantics
  - more: capturing a fragment of CQL
Later: Language properties, complexity analysis, capture CQL, ETALIS, C-SPARQL, CQELS,...
Eventually: Distributed setting, heterogeneous nodes