LARS: A Logic-based Framework for Analyzing Reasoning over Streams

Harald Beck     Minh Dao-Tran     Thomas Eiter     Michael Fink

AAAI 2015
January 27, 2015
Trams and buses appearing within the last 4 minutes, at the same station, at the same time?

CQL (continuous query language)

```sql
SELECT * FROM tram [RANGE 4], bus [RANGE 4] WHERE tram.S = bus.S
```

Query also reports bus at 42 with tram at 44.
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

Public transport scenario

- \( \text{bus}(b_1, s_1) \)
- \( \text{bus}(b_2, s_2) \)
- \( \text{bus}(b_3, s_2) \)
- \( \text{tram}(a_1, s_2) \)

▶ Trams and buses appearing within the last 4 minutes, at the same station, at the same time?

CQL (continuous query language)

\[
\text{SELECT * FROM tram \[RANGE 4\], bus \[RANGE 4\] WHERE tram.S = bus.S}
\]

Query also reports bus at 42 with tram at 44.
A public transport scenario with buses and trams appearing within the last 4 minutes at the same station at the same time. The continuous query language (CQL) query is:

\[
\text{SELECT * FROM tram [RANGE 4], bus [RANGE 4] WHERE tram.S = bus.S}
\]

The query also reports a bus at time 42 with a tram at time 44.
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

- bus \((b_1, s_1)\)
- bus \((b_2, s_2)\)
- bus \((b_3, s_2)\)
- tram \((a_1, s_2)\)

Trams and buses appearing within the last 4 minutes, at the same station, at the same time?

CQL (continuous query language)

\[\text{SELECT * FROM tram [RANGE 4], bus [RANGE 4] WHERE tram.S = bus.S}\]

Query also reports bus at 42 with tram at 44.
Public transport scenario

- bus (b1, s1)
- bus (b2, s2)
- bus (b3, s2)
- tram (a1, s2)

40 41 42 43 44 t = 45 46 47

Trams and buses appearing within the last 4 minutes, at the same station, at the same time?

CQL (continuous query language)

SELECT * FROM tram [RANGE 4], bus [RANGE 4] WHERE tram.S = bus.S

Query also reports bus at 42 with tram at 44.
Public transport scenario

\[ \text{bus}(b_1, s_1) \]

... 40 41 42 43 44 t = 45 46 47 ...
Public transport scenario

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \cdots \quad \cdots \]

\[ 40 \quad 41 \quad 42 \quad 43 \quad 44 \quad t = 45 \quad 46 \quad 47 \]

Trams and buses appearing within the last 4 minutes, at the same station, at the same time?

CQL (continuous query language)

\[ \text{SELECT * FROM tram [RANGE 4], bus [RANGE 4] WHERE tram.S = bus.S} \]

Query also reports bus at 42 with tram at 44.
Public transport scenario

\[
\begin{align*}
bus(b_1, s_1) & \quad bus(b_2, s_2) & \quad tram(a_1, s_2) \\
\cdots & \quad \cdots & \quad \cdots \\
40 & \quad 41 & \quad 42 & \quad 43 & \quad 44 & \quad t = 45 & \quad 46 & \quad 47
\end{align*}
\]
Public transport scenario

- $bus(b_1, s_1)$
- $bus(b_2, s_2)$
- $tram(a_1, s_2)$

Trams and buses appearing within the last 4 minutes,
Public transport scenario

- $\text{bus}(b_1, s_1)$
- $\text{bus}(b_2, s_2)$
- $\text{bus}(b_3, s_2)$
- $\text{tram}(a_1, s_2)$

$\cdots$ 40 41 42 43 44 $t = 45$ 46 47 $\cdots$

- Trams and buses appearing within the last 4 minutes,
- at the same station
Public transport scenario

- Trams and buses appearing within the last 4 minutes,
- at the same station, at the same time?
Public transport scenario

- Trams and buses appearing within the last 4 minutes,
- at the same station, at the same time?

**CQL** (continuous query language)

```
SELECT * FROM tram [RANGE 4], bus [RANGE 4]
WHERE tram.S = bus.S
```
Public transport scenario

- Trams and buses appearing within the last 4 minutes,
- at the same station, at the same time?
- **CQL** (continuous query language)

```sql
SELECT * FROM tram [RANGE 4], bus [RANGE 4]
WHERE tram.S = bus.S
```

:-(
Public transport scenario

- Trams and buses appearing within the last 4 minutes,
- at the same station, at the **same time**?
- **CQL** (continuous query language)

```
SELECT * FROM tram [RANGE 4], bus [RANGE 4]
WHERE tram.S = bus.S
```

- :-( Query also reports bus at 42 with tram at 44
Diverse Approaches - Lack of Unifying Theory

- **Data Management**: low-level, high frequency
- **Semantic Web**: SPARQL extensions
- **KR & R**: *high-level*, lower frequency
Diverse Approaches - Lack of Unifying Theory

- **Data Management**: low-level, high frequency
- **Semantic Web**: SPARQL extensions
- **KR & R**: *high-level*, lower frequency

- Hard to understand and compare semantics
- ⇒ Need language for formal analysis
Diverse Approaches - Lack of Unifying Theory

- **Data Management:** low-level, high frequency
- **Semantic Web:** SPARQL extensions
- **KR & R:** *high-level*, lower frequency

- Hard to understand and compare semantics
  - $\Rightarrow$ Need language for formal **analysis**
- Formal semantics for advanced reasoning over streams
bus\((b_1, s_1)\)    bus\((b_2, s_2)\)

40  41  42  43  44  \(t = 45\)  46  47
bus(b₁, s₁)        bus(b₂, s₂)

[t = 45]

Window operators

τ

-time-based window

time-based window
bus(b_1, s_1)  bus(b_2, s_2)

\[\exists t \in [45, 47] \text{ such that } \bigoplus_{\tau}(4, 2)\]

- **Window operators**
  - \[\bigoplus_{\tau}\] time-based window
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \text{exp}(b_2, s_3) \quad \text{jam} \]

\[ t = 45 \quad 46 \quad 47 \]

Window operators

- \( \tau \) time-based window
- \( \# \) tuple-based window
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \cdots \]

40 41 42 43 44 \( t = 45 \) 46 47

\[ \begin{array}{c}
\begin{array}{c}
\square^2 \\
\# \\
\end{array}
\end{array} \]

- **Window operators**
  - \( \tau \) time-based window
  - \( \# \) tuple-based window
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \text{exp}(b_2, s_3) \quad \text{jam} \]

\[ t = 45 \quad 46 \quad 47 \]

\[ \tau^4 \]

- **Window operators**
  - \( \tau \) time-based window
  - \( \# \) tuple-based window

- **Time reference**
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \text{exp}(b_2, s_3) \quad \text{jam} \]

\[ t = 45 \quad 46 \quad 47 \]

\[ \square^4_\tau \Diamond \text{bus}(b_2, s_2) \]

- Window operators
  - \( \square_\tau \) time-based window
  - \( \square \# \) tuple-based window

- Time reference
  - \( \Diamond \) some time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ bus(b_1, s_1) \]

\[ bus(b_2, s_2) \]

\[ \text{exp}(b_2, s_3) \]

\[ \text{jam} \]

\[ t = 45 \]

\[ \tau \]

\[ bus(b_2, s_2) \rightarrow \text{yes} \]

- Window operators
  - time-based window
  - tuple-based window
- Time reference
  - some time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[
\begin{align*}
bus(b_1, s_1) & \quad bus(b_2, s_2) \\
\ldots & \\
40 & 41 & 42 & 43 & 44 & t = 45 & 46 & 47 \\
\end{align*}
\]

\[
\begin{align*}
\lnot jam
\end{align*}
\]

- **Window operators**
  - \([ \tau ] \) time-based window
  - \([ \# ] \) tuple-based window

- **Time reference**
  - \(\Diamond\) some time
  - \(\Box\) all the time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\( \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \)

\( \cdot \quad \cdot \)

\( \cdot \quad \cdot \)

40 41 42 43 44 \( t = 45 \quad 46 \quad 47 \)

\( \begin{array}{c}
\begin{array}{c}
\square_4 \quad \neg \text{jam} \\
\rightsquigarrow \quad \text{yes}
\end{array}
\end{array} \)

- Window operators
  - \( \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\square_\tau \\
\#\#
\end{array}
\end{array}
\end{array} \) time-based window
  - tuple-based window
- Time reference
  - \( \Diamond \) some time
  - \( \Box \) all the time
bus\((b_1, s_1)\) \hspace{1cm} bus\((b_2, s_2)\)

\[ \begin{array}{cccccccc}
40 & 41 & 42 & 43 & 44 & t=45 & 46 & 47 \\
\end{array} \]

\[ \begin{array}{c}
\text{⊞}_\tau \quad \text{time-based window} \\
\text{⊞} \# \quad \text{tuple-based window} \\
\end{array} \]

\[ \begin{array}{c}
\text{♢} \quad \text{some time} \\
\text{□} \quad \text{all the time} \\
@_t \quad \text{exact time} \\
\end{array} \]
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[
\begin{align*}
\text{bus}(b_1, s_1) & \quad \text{bus}(b_2, s_2) \\
40 & \quad 41 & \quad 42 & \quad 43 & \quad 44 & t = 45 & \quad 46 & \quad 47 \\
\end{align*}
\]

\[
\text{обытие}^4 \quad @_{42} \text{bus}(b_2, s_2) \quad \leadsto \quad \text{yes}
\]

- **Window operators**
  - \(\text{ события } \tau\) time-based window
  - \(\text{ события } \#\) tuple-based window

- **Time reference**
  - \(\text{ some time }\)
  - \(\text{ all the time }\)
  - \(\text{exact time}\)
bus(b_1, s_1)  bus(b_2, s_2)

\[ \square_4 \tau \@_{40} \text{bus}(b_1, s_1) \]

- **Window operators**
  - \( \square \tau \) time-based window
  - \( \square \# \) tuple-based window

- **Time reference**
  - \( \diamond \) some time
  - \( \Box \) all the time
  - \( \@_t \) exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \tau^{4} \quad @_{40} \quad \text{bus}(b_1, s_1) \quad \leadsto \quad \text{no} \]

- **Window operators**
  - \( \tau \) time-based window
  - \( \# \) tuple-based window

- **Time reference**
  - \( \diamond \) some time
  - \( \square \) all the time
  - \( @_{t} \) exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

Window operators

- $\llbracket \tau \rrbracket$ time-based window
- $\llbracket \# \rrbracket$ tuple-based window

Time reference

- ♦ some time
- □ all the time
- @ $t$ exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

Window operators
- $\exists_{\tau}$ time-based window
- $\exists_{\#}$ tuple-based window

Time reference
- $\diamond$ some time
- $\square$ all the time
- $@t'$ exact time
$bus(b_1, s_1)$  $bus(b_2, s_2)$

$\square^1_{\#} @_T bus(B, S)$

- **Window operators**
  - $\square_\tau$ time-based window
  - $\square_{\#}$ tuple-based window
  - \ldots

- **Time reference**
  - $\Diamond$ some time
  - $\Box$ all the time
  - $@_{t'}$ exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \ldots \quad \bullet \quad \ldots \]

\[ \begin{array}{ccccccc}
40 & 41 & 42 & 43 & 44 & t = 45 & 46 & 47 \\
\end{array} \]

\[ T \leftarrow 1 \]

\[ \oplus_{\text{#P}} @T \text{bus}(B, S) \quad \leadsto \quad T = 42, B = b_2, S = s_2 \]

- Window operators
  - \( \oplus_{\tau} \) time-based window
  - \( \oplus_{\#} \) tuple-based window

- Time reference
  - \( \diamond \) some time
  - \( \square \) all the time
  - \( \oplus_{t'} \) exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[
\begin{align*}
bus(b_1, s_1) & \quad bus(b_2, s_2) \\
\ldots & \quad \ldots \\
40 & \quad 41 & 42 & 43 & 44 & t = 45 & 46 & 47
\end{align*}
\]

\[
\begin{aligned}
\begin{array}{c}
\begin{array}{c}
\exists^1 \# @T bus(B, S)
\end{array}
\end{array}
\end{aligned}
\]

- Window operators
  - \(\tau\) time-based window
  - \(#\) tuple-based window

- Time reference
  - \(\lozenge\) some time
  - \(\square\) all the time
  - \(\lozenge_{t'}\) exact time
\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \text{exp}(b_2, s_3) \quad \text{jam} \]

\[ t = 45 \]

\[ \text{reach}(S, S', D), \]

- Window operators
  - \( \square \) time-based window
  - \( \# \) tuple-based window

- Time reference
  - \( \diamond \) some time
  - \( \Box \) all the time
  - \( \@_{t'} \) exact time
\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \]

\[ \text{exp}(b_2, s_3) \]

\[ \text{jam} \]

\[ \ldots \]

\[ t = 45 \quad 46 \quad 47 \]

\[ \mathbb{1}_{\#} @ T \text{bus}(B, S), \text{reach}(S, S', D), \]

\[ T' = T + D \]

- Window operators
  - $\mathbb{\tau}$: time-based window
  - $\mathbb{\#}$: tuple-based window

- Time reference
  - $\square$: some time
  - $\Box$: all the time
  - $\mathbb{\tau}'$: exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \quad \text{exp}(b_2, s_3) \]

\[ \ldots \quad \bullet \quad \bullet \quad \ldots \]

40 41 42 43 44 \( t = 45 \) 46 47

\[ @_T \text{exp}(B, S') \leftarrow \boxplus^1 \# @_T \text{bus}(B, S), \text{reach}(S, S', D), \]
\[ T' = T + D \]

- Window operators
  - \( \boxplus \tau \) time-based window
  - \( \boxplus \# \) tuple-based window

- Time reference
  - \( \diamond \) some time
  - \( \square \) all the time
  - \( @_{t'} \) exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

- \( \text{bus}(b_1, s_1) \)
- \( \text{bus}(b_2, s_2) \)
- \( \text{exp}(b_2, s_3) \)

\[ \vdots \]

\( t = 45 \)

\( \omega_T \exp(B, S') \) \( \leftarrow \) \( \Box^1 \# \omega_T \text{bus}(B, S), \text{reach}(S, S', D), \quad T' = T + D, \not\omega^2_{\tau} \diamond \text{jam}. \)

- **Window operators**
  - \( \Box_{\tau} \) time-based window
  - \( \Box_{\#} \) tuple-based window

- **Time reference**
  - \( \diamond \) some time
  - \( \square \) all the time
  - \( @_{t'} \) exact time
LARS: A Logic-based Framework for Analyzing Reasoning over Streams

\[ \text{bus}(b_1, s_1) \quad \text{bus}(b_2, s_2) \quad \text{jam} \]

\[ \begin{array}{ccccccccc}
40 & 41 & 42 & 43 & 44 & 45 & t = 46 & 47
\end{array} \]

\[ \text{@}_T \exp (B, S') \leftarrow \boxplus_1 \text{bus}(B, S), \text{reach}(S, S', D), \]
\[ T' = T + D, \text{ not } \boxplus_2 \diamond \text{jam}. \]

- Window operators
  - \boxplus_\tau \quad \text{time-based window}
  - \boxplus_\# \quad \text{tuple-based window}

- Time reference
  - \diamond \quad \text{some time}
  - \square \quad \text{all the time}
  - \text{@}_t \quad \text{exact time}
In the paper . . .

- **Formal language**
  - Streams, windows, formulas, rules
  - Stable model semantics

- **Complexity**
  - Model Checking: co-NP-c
  - SAT: $\Sigma_2^p$-c
  - If Nesting of unbounded: both PSPACE-c

- Capturing CQL (continuous query language)

- Relation to ETALIS (complex event processing; intervals)
Summary

- **Goal:** Theoretical underpinning for stream reasoning
  - Analysis
  - Model-based, non-monotonic semantics

- **LARS**
  - Window operators $\mathbb{F}$
  - Time reference: $\Diamond, \Box, @_t$
  - Rule-based; ASP-like semantics
  - Not an end user language

- **Future work**
  - Compare existing semantics
  - Implementation of tractable fragments
  - Incremental evaluation
  - ...