Stream Reasoning
For Linked Data
M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, and E. Della Valle
http://streamreasoning.org/events/sr4ld2014

An Overview
On Stream Reasoning
Emanuele Della Valle
emanuele.dellavalle@polimi.it
http://emanueledellavalle.org
Share, Remix, Reuse — Legally

- This work is licensed under the Creative Commons Attribution 3.0 Unported License.

- Your are free:
  
  **to Share** — to copy, distribute and transmit the work
  
  **to Remix** — to adapt the work

- Under the following conditions

  **Attribution** — You must attribute the work by inserting
  
  - “[source http://streamreasoning.org/sr4ld2013]” at the end of each reused slide
  
  - a credits slide stating
    
    - These slides are partially based on “Streaming Reasoning for Linked Data 2013” by M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, E. Della Valle, and J.Z. Pan http://streamreasoning.org/sr4ld2013

- To view a copy of this license, visit http://creativecommons.org/licenses/by/3.0/
Agenda

- Forms of reasoning for Q/A
- Naïve forms of stream reasoning for Q/A
- A not so naïve form of stream reasoning for Q/A
- Wrap up
MEMO: forms of reasoning for Q/A

- Data-driven (a.k.a. forward reasoning)
  - RDF data
  - Reasoner
  - Inferred data
  - SPARQL
  - Ontology

- Query-driven – backward reasoning
  - RDF data
  - Reasoner
  - SPARQL
  - Ontology

- Query-driven – query rewriting (a.k.a. ontology based data access)
  - Data
  - Rewritten query
  - Reasoner
  - SPARQL
  - Ontology
Naïve Stream Reasoning

- **Data-driven (a.k.a. forward reasoning)**

- **Query-driven – backward reasoning**

- **Query-driven – query rewriting (a.k.a. ontology based data access)**
Example of Stream Reasoning 1/2

- **Query**: measure the impact of Alice's microposts
  
  **MEMO**: our running example data model

- **For example**

  Alice posts $p_1$.

  Bob posts $p_2$.

  $p_1$ discusses $p_2$.

  $p_2$ discusses $p_3$.

  $p_3$ discusses $p_4$.

  $p_4$ discusses $p_5$.

  $p_5$ discusses $p_6$.

  $p_6$ discusses $p_7$.

  $p_7$ discusses $p_8$.

  50 min ago 40 min ago 30 min ago 20 min ago 10 min ago now
What impact has been my micropost $p_1$ creating in the last hour? Let’s count the number of microposts that discuss it …

REGISTER STREAM ImpactMeter AS
SELECT (count(?p) AS ?impact)
FROM STREAM <http://.../fb> [RANGE 60m STEP 10m]
WHERE {
  :Alice posts [ sr:discusses ?p ]
}

Transitive property

Alice posts $p_1$. 7!
Naïve data-driven stream reasoning

Memo

- S2R
- RDF data
- Reasoner
- Inferred data
- SPARQL

 Ontology

\[ p_1 \]

\[ 0 \]
Naïve data-driven stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

ontology

discusses

$p_1$  $p_2$

10 min ago  now
Naïve data-driven stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

ontology

$p_1$ discusses $p_2$

$p_2$ discusses $p_3$

20 min ago 10 min ago now
Naïve data-driven stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

ontology

\[ p_1 \] discusses \[ p_2 \] discusses \[ p_4 \]

\[ p_2 \] discusses \[ p_3 \]

- now
  - 30 min ago
  - 20 min ago
  - 10 min ago
Naïve data-driven stream reasoning

- **Memo**

  The entire inference process is repeated each time the SR2 operator delivers new RDF data.
Naïve data-driven stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

ontology

$p_1$ → discusses → $p_2$ → discusses → $p_4$ → discusses → $p_7$

$p_3$ → discusses → $p_5$ → discusses → $p_6$ → discusses → $p_8$

50 min ago → 40 min ago → 30 min ago → 20 min ago → 10 min ago → now
Naïve data-driven stream reasoning

**Memo**

- S2R → RDF data → Reasoner → Inferred data → SPARQL

Diagram:

```
1 2
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>p2</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>p4</td>
</tr>
<tr>
<td></td>
<td>p7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>p5</td>
<td>p6</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>p8</td>
<td></td>
</tr>
</tbody>
</table>
```

Properties:
- p1 discusses p2
- p3 discusses p4
- p5 discusses p7
- p6 discusses p8

Time:
- 50 min ago
- 40 min ago
- 30 min ago
- 20 min ago
- 10 min ago
- Now
Naïve data-driven stream reasoning

### Memo

1. **S2R** → **RDF data** → **Reasoner** → **Inferred data** → **SPARQL**

   - **S2R** (Semantic Stream to RDF)
   - **RDF data**
   - **Reasoner**
   - **Inferred data**
   - **SPARQL**

   **ontology**

The reasoner infers data that is irrelevant to query answering.
Naive query-driven (backward) stream reasoning

- **Memo**

**Diagram:**
- **S2R**
- **RDF data**
- **Reasoner**
- **ontology**
- **SPARQL**

**Diagram Elements:**
- **p₁**
- **now**
Naïve query-driven (backward) stream reasoning

- **Memo**

  S2R → RDF data → Reasoner → Inferred data → SPARQL

  - ontology

  \[ p_1 \text{ discusses } p_2 \]

  10 min ago → now
Naive query-driven (backward) stream reasoning

- Memo

Diagram:
- S2R
- RDF data
- Reasoner
- Inferred data
- SPARQL

Ontology:
- \( p_1 \) discusses \( p_2 \)
- \( p_2 \) discusses \( p_3 \)

Timeline:
- 20 min ago
- 10 min ago
- Now
Naive query-driven (backward) stream reasoning

**Memo**

- S2R → RDF data → Reasoner → Inferred data → SPARQL

- Diagram showing relationships among nodes p1, p2, p3, p4 with arrows indicating "discusses" relationships.

- Timeline with timestamps: 30 min ago, 20 min ago, 10 min ago, now.
Naïve query-driven (backward) stream reasoning

**Memo**

The entire inference process is repeated each time the SR2 operator delivers new RDF data.
Naive query-driven (backward) stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

ontology

```
\[ p_1 \rightarrow \text{discusses} \rightarrow p_2 \rightarrow \text{discusses} \rightarrow p_3 \rightarrow \text{discusses} \rightarrow p_4 \rightarrow \text{discusses} \rightarrow p_5 \rightarrow \text{discusses} \rightarrow p_6 \rightarrow \text{discusses} \rightarrow p_7 \rightarrow \text{discusses} \rightarrow p_8 \] 
```

50 min ago → 40 min ago → 30 min ago → 20 min ago → 10 min ago → now

http://streamreasoning.org/events/sr4ld2014
Naive query-driven (backward) stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

<table>
<thead>
<tr>
<th>Time</th>
<th>Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>now</td>
<td>p_1</td>
</tr>
<tr>
<td>10 min</td>
<td>p_2</td>
</tr>
<tr>
<td>20 min</td>
<td>p_3</td>
</tr>
<tr>
<td>30 min</td>
<td>p_4</td>
</tr>
<tr>
<td>40 min</td>
<td>p_5</td>
</tr>
<tr>
<td>50 min</td>
<td>p_6</td>
</tr>
<tr>
<td>30 min</td>
<td>p_7</td>
</tr>
<tr>
<td>20 min</td>
<td>p_8</td>
</tr>
</tbody>
</table>

discusses

ontology
Naive query-driven (backward)
stream reasoning

- Memo

S2R \rightarrow RDF data \rightarrow Reasoner \rightarrow Inferred data

ontology

The backward reasoner would not even start!

$p_2$ discusses $p_4$ discusses $p_7$

$p_3$ discusses $p_5$ discusses $p_8$

$p_6$ discusses

50 min ago 40 min ago 30 min ago 20 min ago 10 min ago now

now
Naïve query-driven stream reasoning by query rewriting

- **MEMO**

- It is not that straight forward :-((
  - Lack of a standard query language for DSMS and CEP
  - Lack of a well-understood operational semantics for DSMS and CEP (cf. SECRET by I. Botan et al., PVLDB 3(1), 2010)
  - Lack of expressiveness in OWL2QL
    - Temporal reasoning
    - Aggregates (not straight forward, you need skolemization)
    - Functions (you need skolemization)

- Wait for MorphStream session to learn more
Not so naïve stream reasoning

- Naïve data-driven approach

- From snapshots to changes
  - What has just been inserted?
  - What has just been deleted?

S2R → RDF data → Reasoner → Inferred data → SPARQL

Incremental!!!
The problem is that materialization (the result of data-driven processing) are very difficult to decrement efficiently.

- State-of-the-art: DRed algorithm
  - Over delete
  - Re-derive
  - Insert

- **Overestimation of deletion**: Overestimates deletions by computing all direct consequences of a deletion.

- **Rederivation**: Prunes those estimated deletions for which alternative derivations (via some other facts in the program) exist.

- **Insertion**: Adds the new derivations that are consequences of insertions to extensional predicates.
The Intuition of DRed Algorithm

- Let’s assume that we have the following materialized graph:

![Materialized Graph](image)

- While inserts are not problematic, deletion are difficult to handle. If we delete $p_2$ discusses $p_1$ ($p_2 \rightarrow p_1$), we have:
  - overestimate the impact of the deletion and mark for deletion $p_4 \rightarrow p_1$ that can be derived by $p_4 \rightarrow p_2$ and $p_2 \rightarrow p_1$

![Deletion with Overestimation](image)

- look for alternative derivation of $p_4 \rightarrow p_1$ and eventually find the chain $p_4 \rightarrow p_3$ and $p_3 \rightarrow p_1$

![Alternative Derivation](image)
Not so naïve data-driven stream reasoning

Memo

- Incremental !!!

S2R

Reasoner

Inferred data

ontology

SPARQL

insertions

deletions

\( p_1 \)

now
Not so naïve data-driven stream reasoning

- Memo

S2R \rightarrow\leftarrow \text{Reasoner} \rightarrow\leftarrow \text{Inferred data} \rightarrow SPARQL

- Insertions: S2R \rightarrow \text{Reasoner}
- Deletions: S2R \leftarrow \text{Reasoner}

Incremental !!!

ontology

\text{Reasoner} \quad \text{Inferred data} \quad \text{SPARQL}

1

discusses

p_2

\text{discusses}

\text{p}_1

10\text{ min ago}

\text{now}
Not so naïve data-driven stream reasoning

Memo

- S2R
  - insertions
  - deletions

- Reasoner

- Inferred data
  - Incremental !!!

- SPARQL

```
discusses

p1

discusses

p2

p2

discusses

p3
```

Timeline:
- 20 min ago
- 10 min ago
- now
Not so naïve data-driven stream reasoning

**Memo**

- S2R
- Reasoner
- Inferred data
- SPARQL

**Summary**

- Incremental
- Insertions
- Deletions
- Ontology

**Diagram**

- $p_1$ discusses $p_2$
- $p_2$ discusses $p_4$
- $p_3$ discusses $p_4$

**Timeline**

- 30 min ago
- 20 min ago
- 10 min ago
- Now
Not so naïve data-driven stream reasoning

**Memo**

The inference process is performed incrementally.

- **S2R**
- **Reasoner**
- **Inferred data**
- **SPARQL**

The process involves insertions and deletions. The inference process is performed incrementally using the **ontology**.
Not so naïve data-driven stream reasoning

- Memo

The inference process is performed incrementally.

Reasoner

Inferred data

SPARQL

S2R

insertions

deletions

ontology

The inference process is performed incrementally.

Incremental !!!

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.

The inference process is performed incrementally.
Not so naïve data-driven stream reasoning

- **Memo**

  S2R \( \xrightarrow{\text{insertions}} \) Reasoner \( \xrightarrow{\text{Inferred data}} \) SPARQL

  \( \xrightarrow{\text{deletions}} \)

  ontology

  **Incremental !!!**

  **Reasoner**

  **Inferred data**

  **No inference is required when nothing changes**

  \( p_1 \) \( \xrightarrow{\text{discusses}} \) \( p_2 \)

  \( p_2 \) \( \xleftarrow{\text{discusses}} \) \( p_3 \)

  \( p_3 \) \( \xrightarrow{\text{discusses}} \) \( p_4 \)

  \( p_4 \) \( \xleftarrow{\text{discusses}} \) \( p_5 \)

  \( p_5 \) \( \xrightarrow{\text{discusses}} \) \( p_6 \)

  \( p_6 \) \( \xleftarrow{\text{discusses}} \) \( p_7 \)

  \( p_7 \) \( \xrightarrow{\text{discusses}} \) \( p_8 \)

  \( p_8 \) \( \xleftarrow{\text{discusses}} \) \( p_9 \)

  \( p_9 \) \( \xrightarrow{\text{discusses}} \) \( p_{10} \)

  \( p_{10} \) \( \xleftarrow{\text{discusses}} \) \( p_1 \)

  \( p_1 \) \( \xrightarrow{\text{discusses}} \) \( p_2 \)

  \( p_2 \) \( \xleftarrow{\text{discusses}} \) \( p_3 \)

  \( p_3 \) \( \xrightarrow{\text{discusses}} \) \( p_4 \)

  \( p_4 \) \( \xleftarrow{\text{discusses}} \) \( p_5 \)

  \( p_5 \) \( \xrightarrow{\text{discusses}} \) \( p_6 \)

  \( p_6 \) \( \xleftarrow{\text{discusses}} \) \( p_7 \)

  \( p_7 \) \( \xrightarrow{\text{discusses}} \) \( p_8 \)

  \( p_8 \) \( \xleftarrow{\text{discusses}} \) \( p_9 \)

  \( p_9 \) \( \xrightarrow{\text{discusses}} \) \( p_{10} \)

  \( p_{10} \) \( \xleftarrow{\text{discusses}} \) \( p_1 \)
Not so naïve data-driven stream reasoning

- **Memo**

  - **S2R**
    - insertions
    - deletions
  - **Reasoner**
  - **Inferred data**
  - **SPARQL**

  **Incremental !!!**

  **ontology**

  **Inference is required to process deletions :-((**

  \[ \text{p}_2 \text{ discusses } \text{p}_3 \text{ discusses } \text{p}_4 \text{ discusses } \text{p}_5 \text{ discusses } \text{p}_6 \text{ discusses } \text{p}_7 \text{ discusses } \text{p}_8 \text{ discusses } \text{p}_9 \]

  **http://streamreasoning.org/events/sr4ld2014**
Wrapping up

- Data-driven is very expensive
  - It materializes the full RDF snapshot delivered by the S2R operator
  - It infers data even if it is not relevant to the Q/A task
  - Naïve implementation in C-SPARQL Engine

- Query-driven (backward) is expensive
  - It infers only data relevant to the Q/A task
  - It does not benefit from data inferred for the previous snapshot delivered by the S2R operator
  - Naïve implementation in C-SPARQL Engine

- Query-driven by query rewriting is possible but, problematic
  - Lack of a standard query language and well-understood operational semantics for DSMS and CEP
  - Lack of expressiveness in OWL2QL
  - Naïve implementation in MorphStream

- Not so naïve stream reasoning using DRed is practicable
  - The inference process is performed incrementally
  - No inference is required when nothing changes
  - Inference is required to processes deletions :-(

http://streamreasoning.org/events/sr4ld2014
Stream Reasoning For Linked Data
M. Balduini, J-P Calbimonte, O. Corcho, D. Dell'Aglio, and E. Della Valle
http://streamreasoning.org/events/sr4ld2014

Naive reasoning on RDF streams
Emanuele Della Valle
emanuele.dellavalle@polimi.it
http://emanueledellavalle.org