Stream Reasoning For Linked Data
J-P Calbimonte, D. Dell'Aglio, E. Della Valle, M.I. Ali and A. Mileo
http://streamreasoning.org/events/sr4ld2015

Stream Reasoning Naïve Approaches
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 a credits slide stating
    - These slides are partially based on “Streaming Reasoning for Linked Data 2015” by J-P Calbimonte, D. Dell'Aglio, E. Della Valle, M. I. Ali and A. Mileo
      http://streamreasoning.org/sr4ld2015

- 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)**
  - RDF data → Reasoner → Inferred data
  - S2R

- **Query-driven – backward reasoning**
  - RDF data → Reasoner → SPARQL
  - S2R

- **Query-driven – query rewriting (a.k.a. ontology based data access)**
  - data → Rewritten query → Reasoner → SPARQL
  - S2R
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 \( p_2 \).

Post

\( p_1 \)

\( p_2 \)

\( p_3 \)

\( p_4 \)

\( p_5 \)

\( p_6 \)

\( p_7 \)

\( p_8 \)

50 min ago

40 min ago

30 min ago

20 min ago

10 min ago

now

\( p_1 \) discusses \( p_2 \)

\( p_2 \) discusses \( p_4 \)

\( p_4 \) discusses \( p_7 \)

\( p_7 \) discusses \( p_8 \)

\( p_1 \) discusses \( p_3 \)

\( p_3 \) discusses \( p_5 \)

\( p_5 \) discusses \( p_6 \)

\( p_6 \) discusses \( p_8 \)
Example of Stream Reasoning 2/2

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 $p_1$.
Naïve data-driven stream reasoning

- Memo

\[ p_1 \]

S2R → RDF data → Reasoner → Inferred data → SPARQL

ontology
Naïve data-driven stream reasoning

- **Memo**

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

Legend:
- Discusses $p_2$
- Begins 10 min ago
- Current time: 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

$\text{discusses}$ $p_1$ $p_2$ $p_3$ $p_4$

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 \( p_4 \) discusses \( p_5 \) discusses \( p_8 \)
  - \( p_5 \) discusses \( p_6 \)

  - Time: 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
Naïve data-driven stream reasoning

- Memo

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

- Memo

- S2R → RDF data → Reasoner → ontology → SPARQL

- $p_1$

- now

http://streamreasoning.org/events/sr4ld2015
Naïve query-driven (backward) stream reasoning

- Memo

1. S2R \rightarrow RDF data \rightarrow Reasoner \rightarrow Inferred data \rightarrow SPARQL

- Naïve query-driven (backward) stream reasoning

- Discusses
  - p_1 \rightarrow p_2

10 min ago \rightarrow now
Naïve query-driven (backward) stream reasoning

**Memo**

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

- p₁
  - discuss p₂
  - discuss p₃

- p₂

- p₃

Timeline:
- 20 min ago
- 10 min ago
- now

Links:
- http://streamreasoning.org/events/sr4ld2015
Naïve query-driven (backward) stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data → SPARQL

- ontology

\[ \text{p}_1 \rightarrow \text{discusses} \rightarrow \text{p}_2 \rightarrow \text{discusses} \rightarrow \text{p}_4 \]

\[ \text{p}_2 \rightarrow \text{discusses} \rightarrow \text{p}_3 \rightarrow \text{discusses} \rightarrow \text{p}_1 \]

\[ \text{now} \rightarrow \text{10 min ago} \rightarrow \text{20 min ago} \rightarrow \text{30 min ago} \]
Naïve query-driven (backward) stream reasoning

- Memo

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

Memo

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

- $p_1$ discusses $p_2$
- $p_2$ discusses $p_4$
- $p_4$ discusses $p_7$
- $p_1$ discusses $p_3$
- $p_3$ discusses $p_5$
- $p_5$ discusses $p_8$
- $p_1$ discusses $p_6$
- $p_6$ discusses $p_8$

- 50 min ago
- 40 min ago
- 30 min ago
- 20 min ago
- 10 min ago
- Now

http://streamreasoning.org/events/sr4ld2015
Naïve query-driven (backward)
stream reasoning

- Memo

S2R → RDF data → Reasoner → Inferred data

ontology

SPARQL

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

p1 discusses p2

p2 discusses p3

p3 discusses p4

p4 discusses p5

p5 discusses p6

p6 discusses p7

p7 discusses p8

p8 discusses p1
Naïve query-driven (backward) stream reasoning

Memo

The backward reasoner would not even start!
Naïve query-driven stream reasoning by query rewriting

MEMO

- S2R
  - data
  - Rewritten query
  - Reasoner
  - ontology
  - SPARQL

- 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)

- Foundational works in MorphStream (see this morning)
Not so naïve stream reasoning

- Naïve data-driven approach
- From snapshots to changes
  - What has just been inserted?
  - What has just been deleted?

Diagram:

```
S2R -> RDF data -> Reasoner -> Inferred data -> SPARQL
```

```
S2R -> insertions
```

```
S2R -> deletions
```

```
Reasoner -> Inferred data
```

```
Incremental!!!
```

```
http://streamreasoning.org/events/sr4ld2015
```
Not so naïve stream reasoning

MEMO

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

DReD

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

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:

1. **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 $p_2 \rightarrow p_1$.

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

http://streamreasoning.org/events/sr4ld2015
Not so naïve data-driven stream reasoning

- Memo

- S2R
  - insertions
  - deletions

- Reasoner

- Inferred data

- Incremental !!!

- SPARQL

- \( p_1 \)

- Now
Not so naïve data-driven stream reasoning

**Memo**

- **S2R**
  - insertions
  - deletions

  **Reasoner**
  - Inferred data
  - Incremental !!!
  - SPARQL

- **ontology**

- **discusses**
  - $p_1$
  - $p_2$

10 min ago | now
Not so naïve data-driven stream reasoning

- Memo

![Diagram showing S2R, insertions, deletions, Reasoner, Inferred data, ontology, SPARQL, and p1, p2, p3 relationships.]

- Incremental !!!

- Reasoner

- Inferred data

- S2R

- insertions

- deletions

- ontology

- SPARQL

http://streamreasoning.org/events/sr4ld2015
Not so naïve data-driven stream reasoning

- **Memo**

  S2R \(\rightarrow\) Reasoner \(\rightarrow\) Inferred data \(\rightarrow\) SPARQL

  - insertions
  - deletions

  Inferred data

  **Incremental !!!**

  Ontology

  \(p_1\) \(\rightarrow\) \(p_2\) \(\rightarrow\) \(p_4\)

  \(p_2\) \(\rightarrow\) \(p_4\)

  \(p_1\) \(\rightarrow\) \(p_3\)

  \(p_3\) \(\rightarrow\) \(p_4\)

  \(p_1\) \(\rightarrow\) \(p_2\)

  **3**

  30 min ago  20 min ago  10 min ago  now

  [Link to streamreasoning.org/events/sr4ld2015]
Not so naïve data-driven stream reasoning

- **Memo**

  The inference process is performed incrementally.

The diagram illustrates the flow of data from S2R through insertions and deletions to the Reasoner, which processes the data incrementally. The inferred data is then fed into the SPARQL system. The ontology is the backbone of the reasoning process, connecting entities through the `discusses` relation.
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.

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

insertions

deletions

Reasoner

Inferred data

ontology

SPARQL

Incremental !!!

No inference is required when nothing changes
Not so naïve data-driven stream reasoning

- Memo

S2R → Reasoner → Inferred data

Incremental !!!

SPARQL

Inference is required to processes deletions :-(

Reasoner

ontology

insertions

deletions

S2R

p1

p2

p3

p4

p5

p6

p7

p8

discusses

discusses

discusses

discusses

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

http://streamreasoning.org/events/sr4ld2015
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 benefits 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 :-(
Naive reasoning on RDF streams

Emanuele Della Valle
emanuele.dellavalle@polimi.it
http://emanueledellavalle.org