

Stream Reasoning For Linked Data

M. Balduini, J-P Calbimonte, O. Corcho,
D. Dell'Aglio, E. Della Valle, and J.Z. Pan
<http://streamreasoning.org/sr4ld2013>

 ISWC 2013
Sydney, Australia

Naive reasoning on RDF streams

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



- 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**



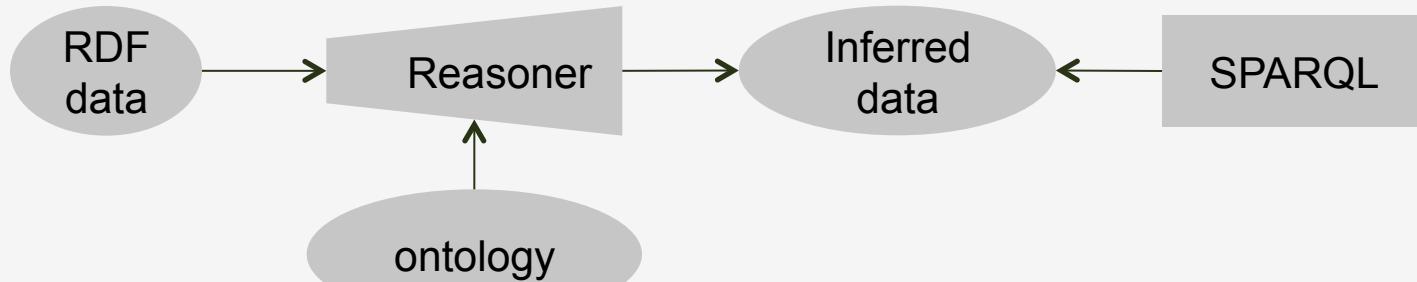
Attribution — You must attribute the work by inserting

- “[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>

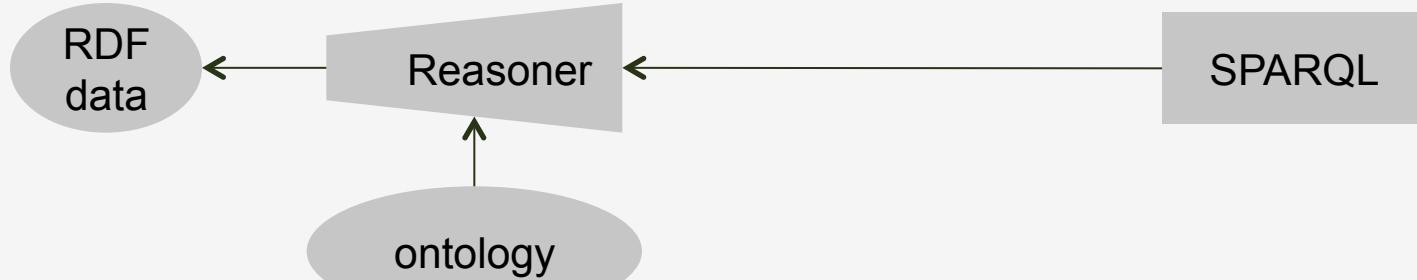
- To view a copy of this license, visit
<http://creativecommons.org/licenses/by/3.0/>

- 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

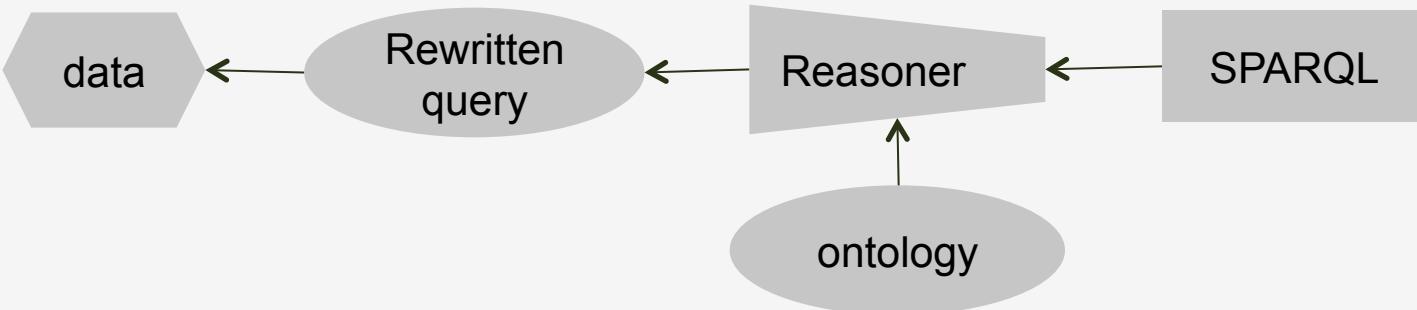
- Data-driven (a.k.a. forward reasoning)



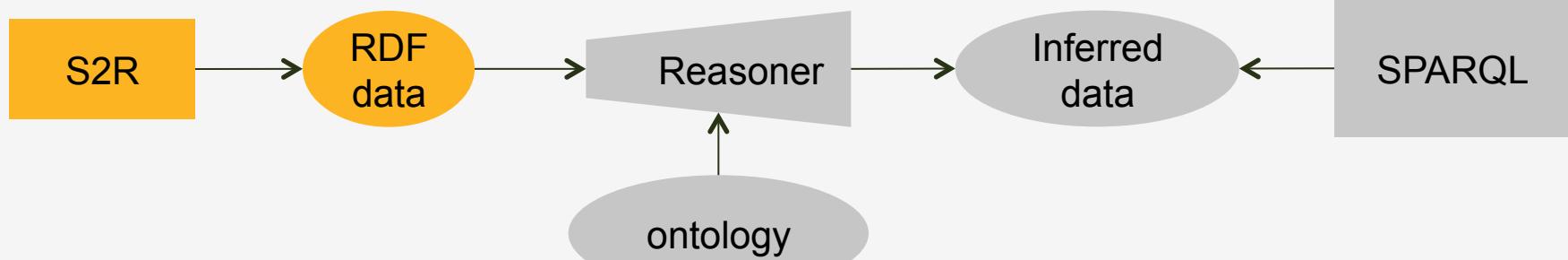
- Query-driven – backward reasoning



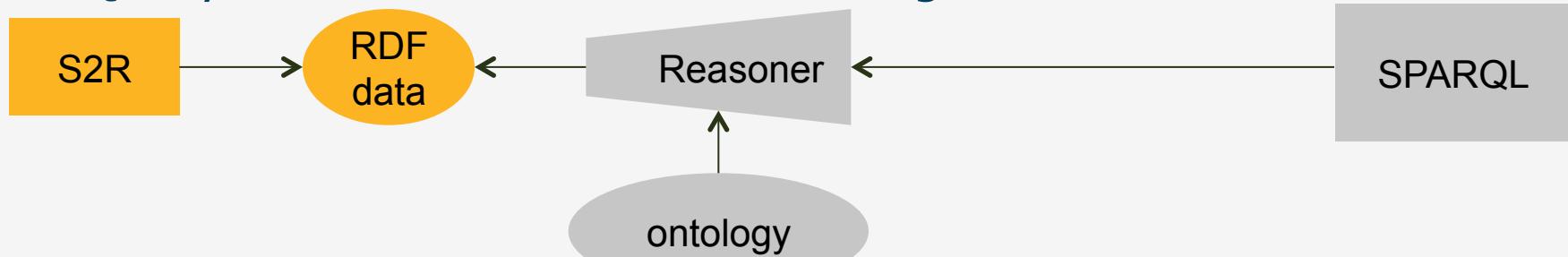
- Query-driven – query rewriting (a.k.a. ontology based data access)



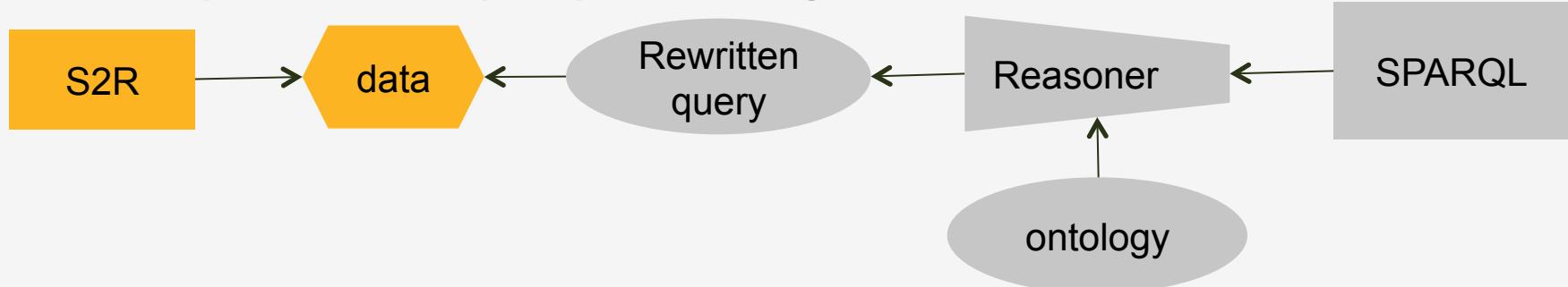
- Data-driven (a.k.a. forward reasoning)



- Query-driven – backward reasoning

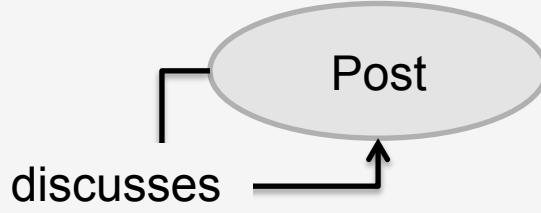


- Query-driven – query rewriting (a.k.a. ontology based data access)

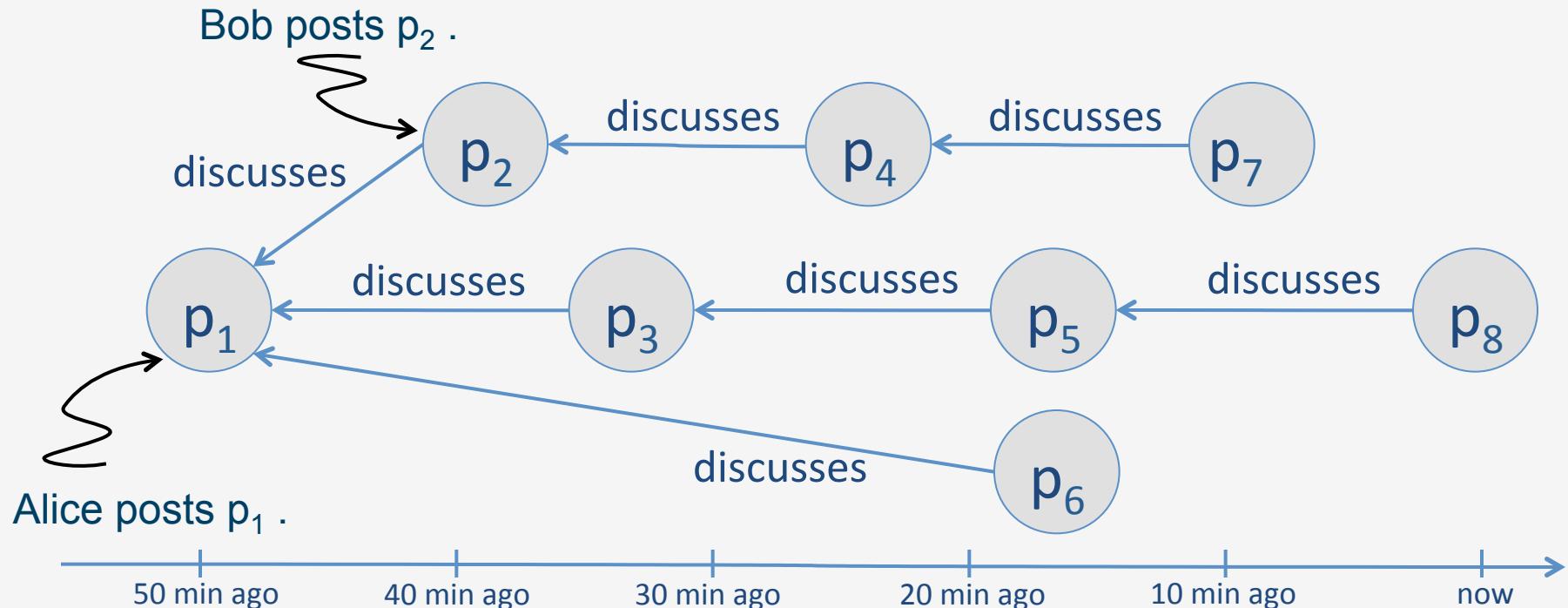


Example of Stream Reasoning 1/2

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



- For example



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 ...*

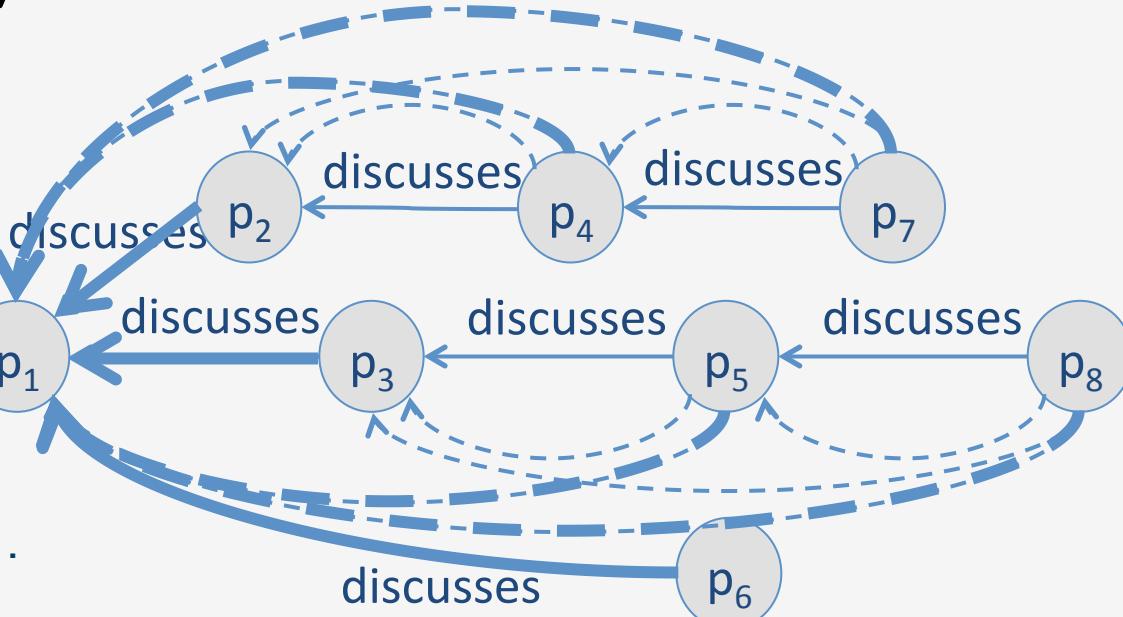
```
REGISTER STREAM ImpactMeter AS
SELECT (count(?p) AS ?impact)
FROM STREAM <http://.../fb> [RANGE 60m STEP 10m]
WHERE {
```

:Alice posts [sr:discusses ?p]

}

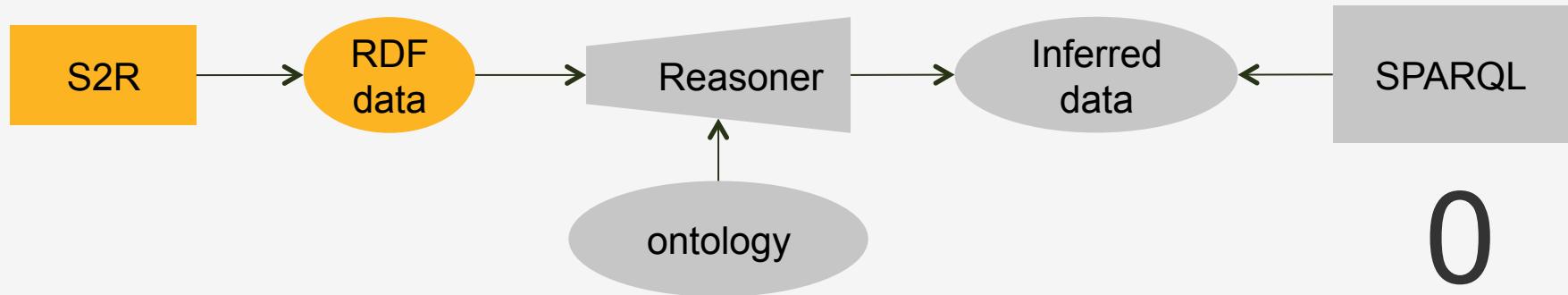
*Transitive
property*

Alice posts p_1 .



Naïve data-driven stream reasoning

- Memo

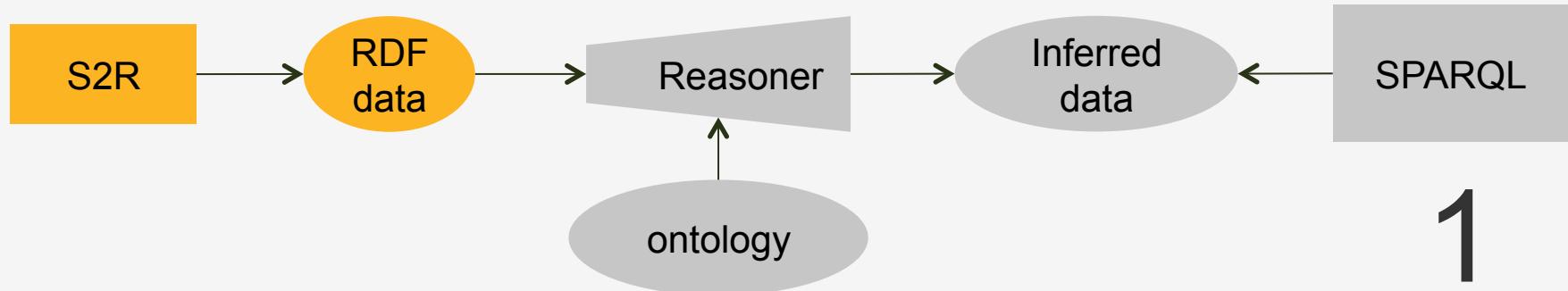


p_1

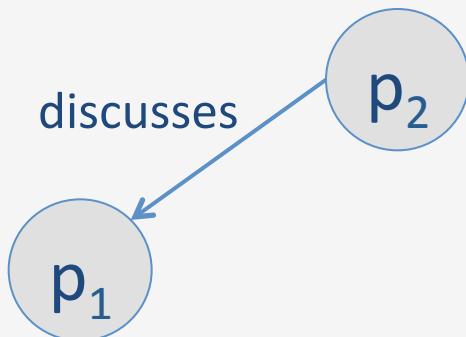
now

Naïve data-driven stream reasoning

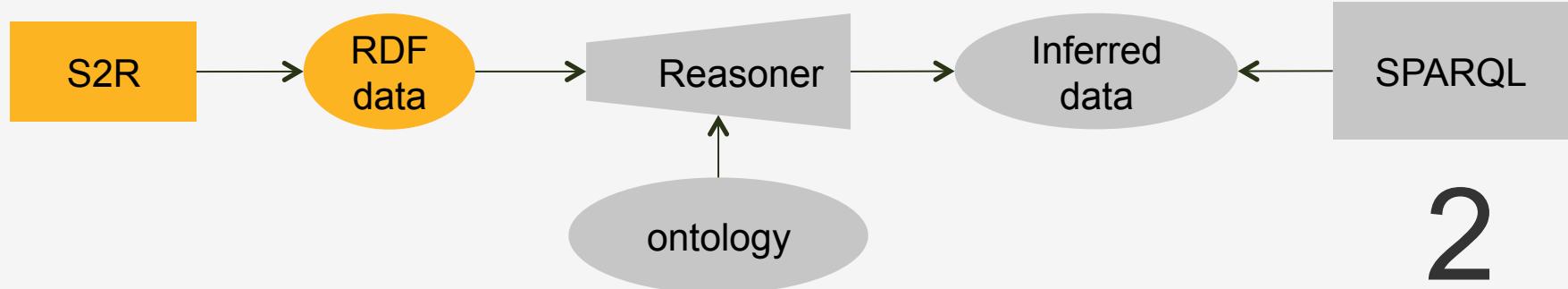
- Memo



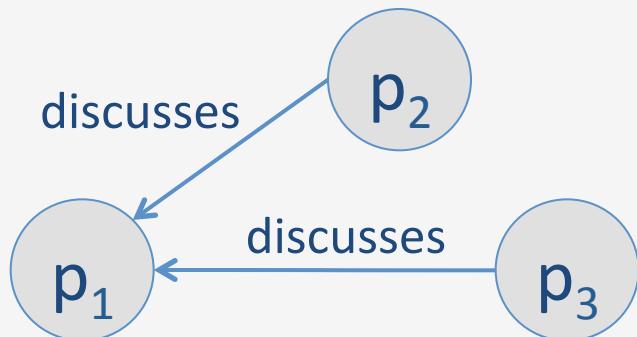
1



- Memo

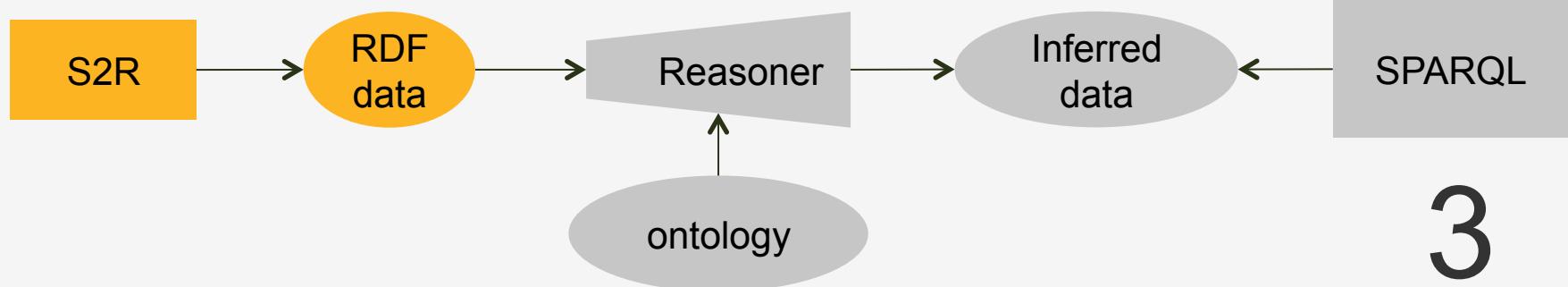


2

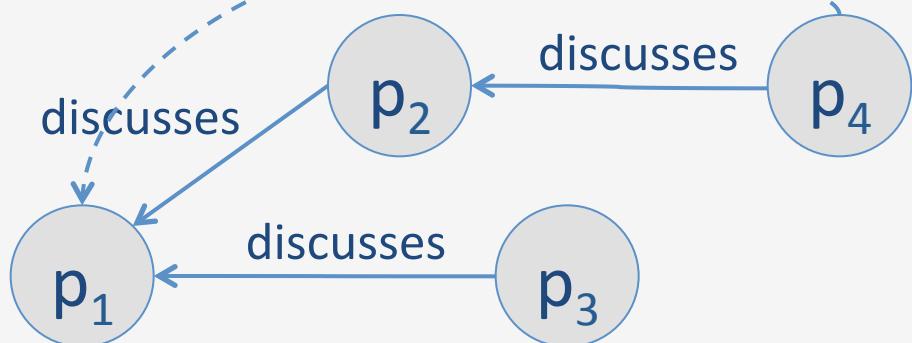


Naïve data-driven stream reasoning

- Memo

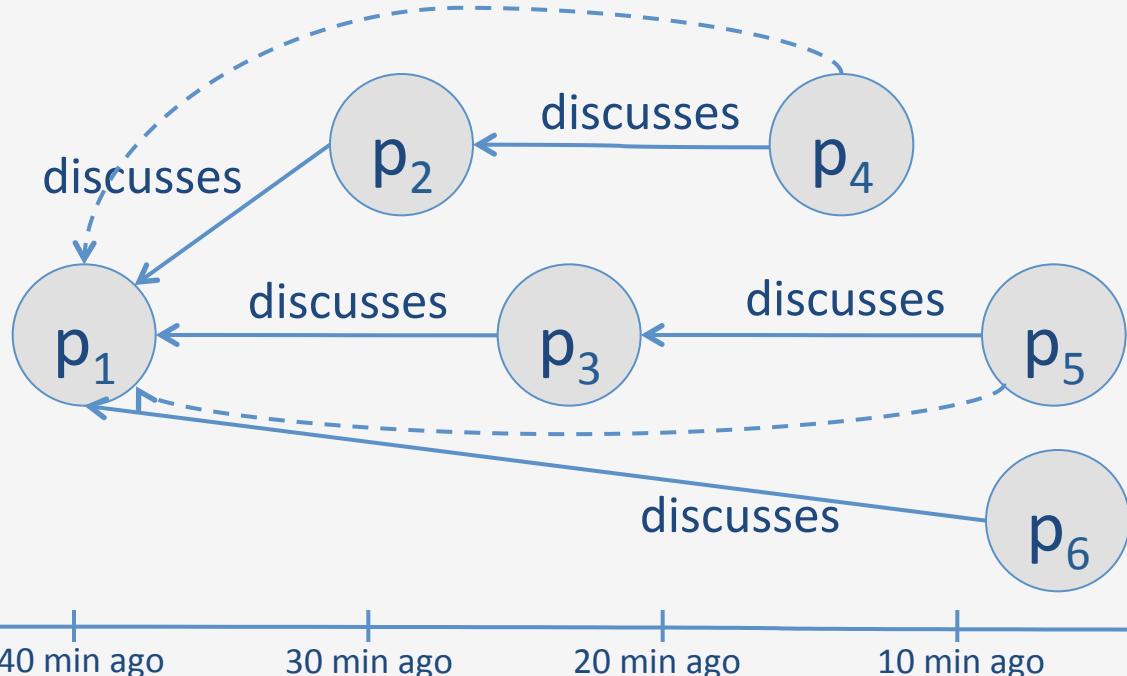
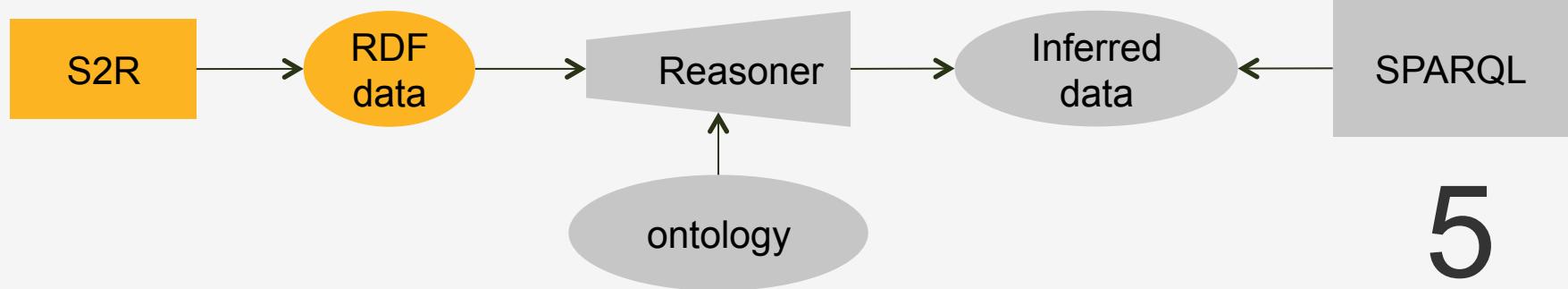


3



Naïve data-driven stream reasoning

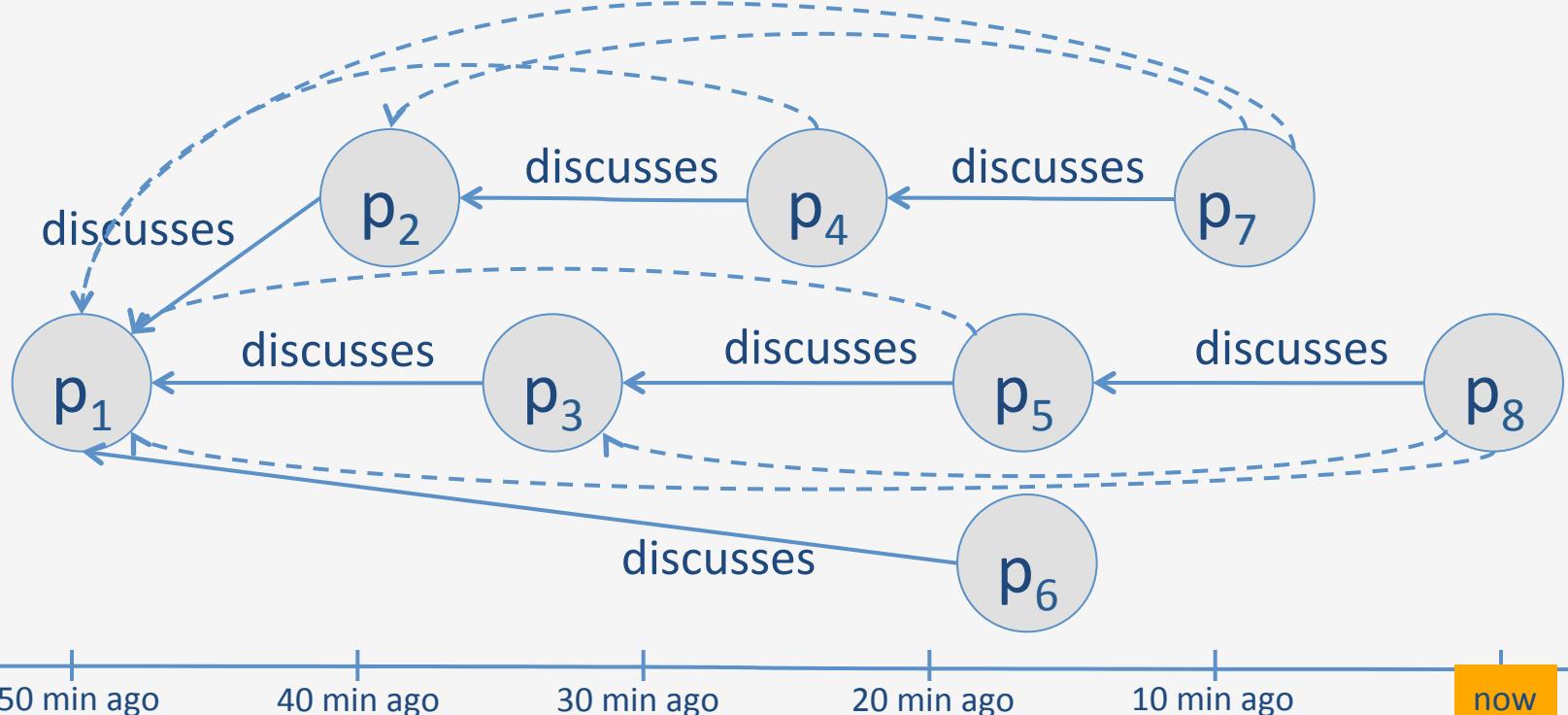
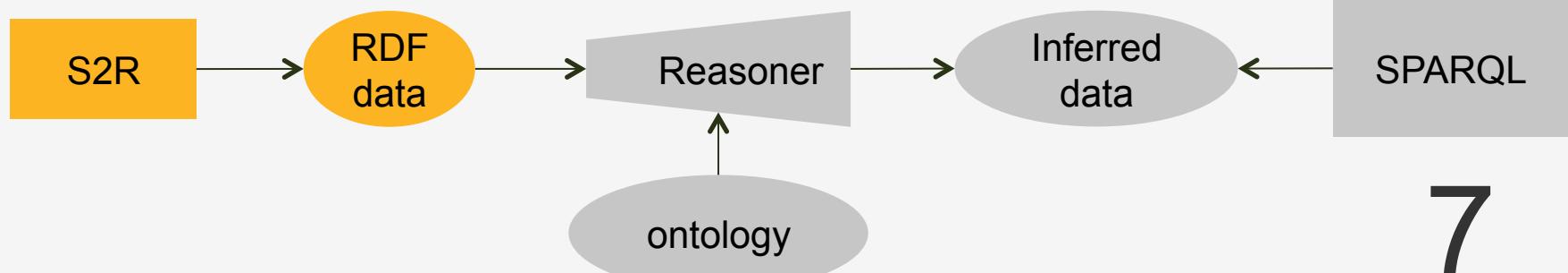
Memo



The entire inference process is repeated each time the SR2 operator delivers new RDF data

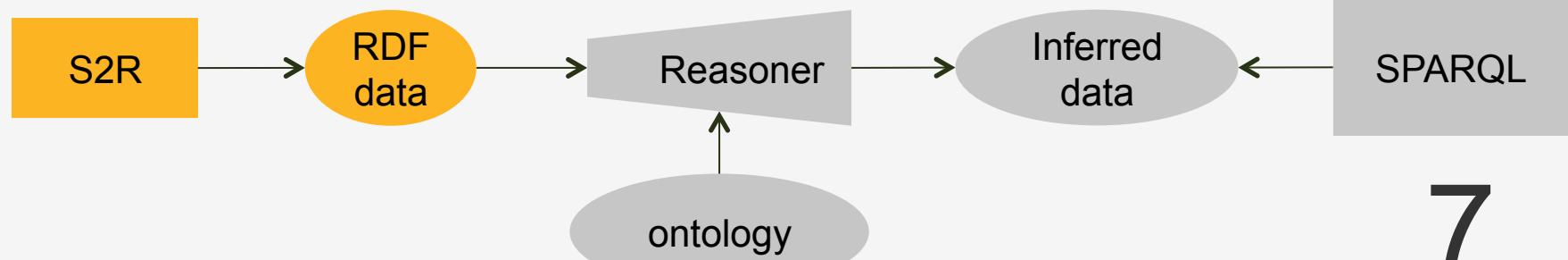
Naïve data-driven stream reasoning

Memo

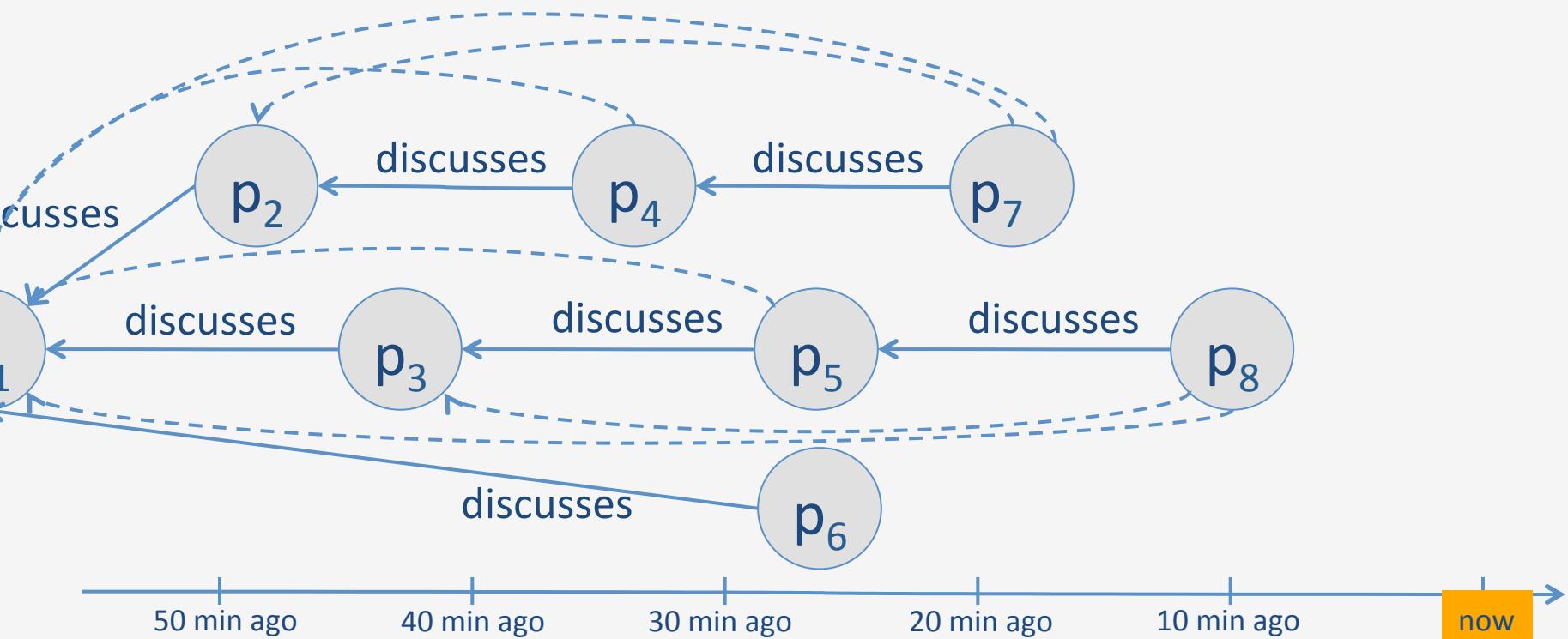


Naïve data-driven stream reasoning

Memo

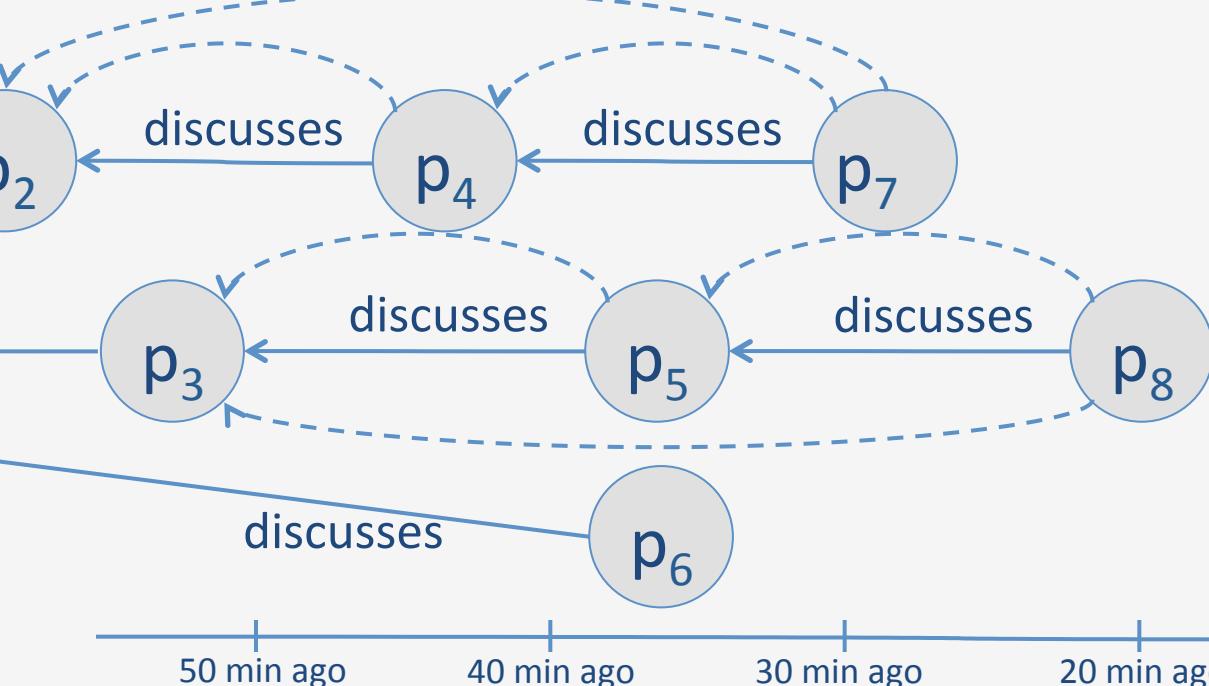
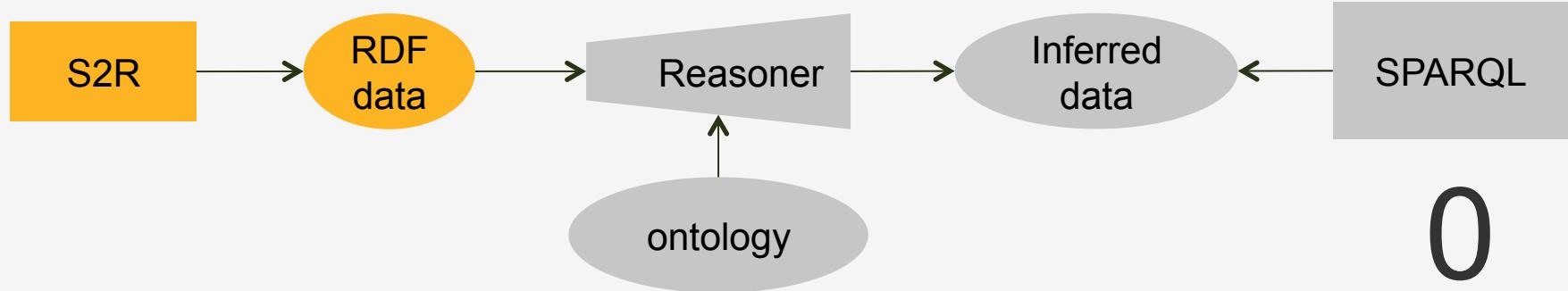


7



Naïve data-driven stream reasoning

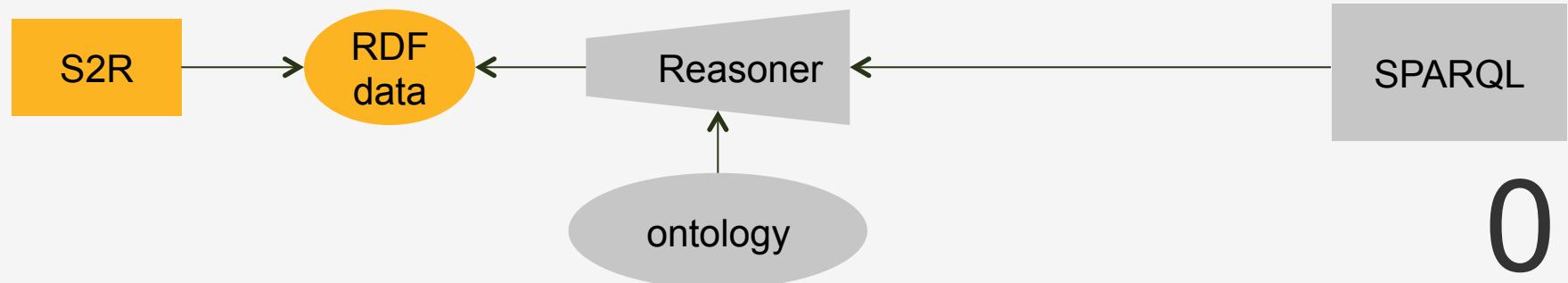
Memo



The reasoner infers data that is irrelevant to query answering

Naïve query-driven (backward) stream reasoning

- Memo

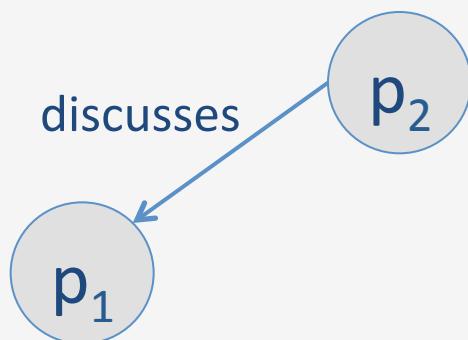
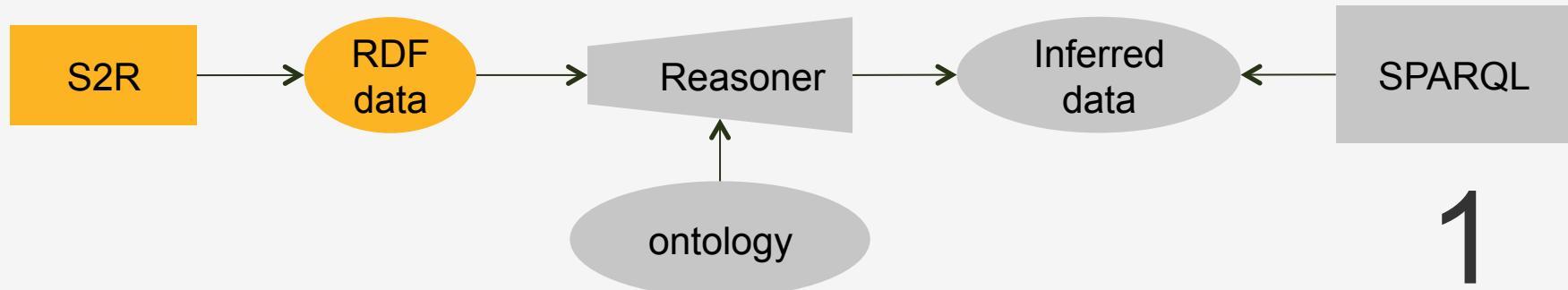


p_1



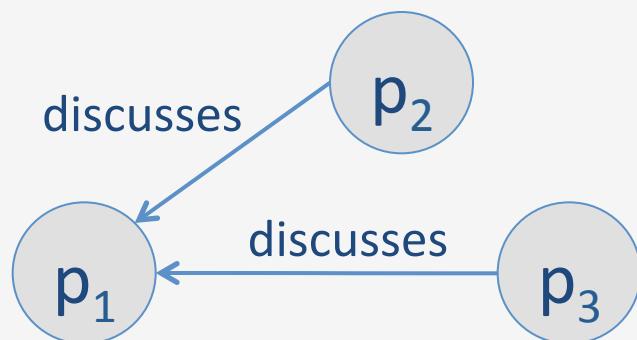
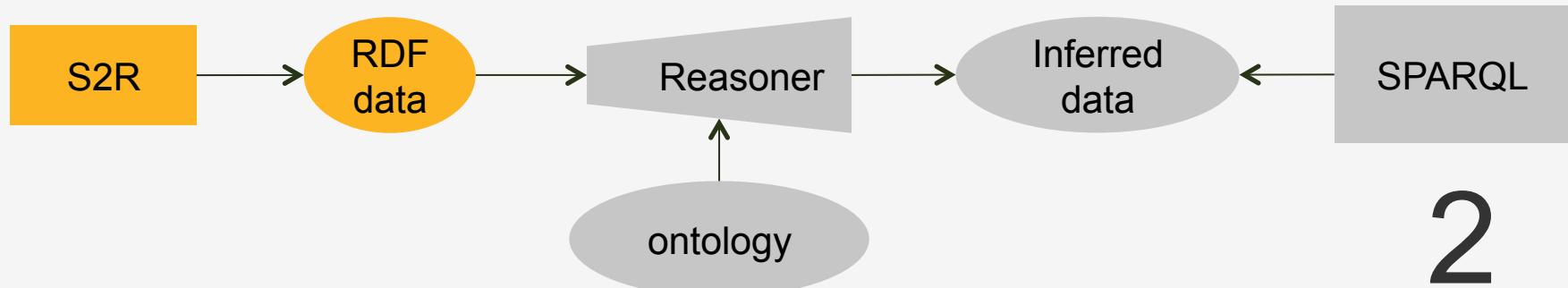
Naïve query-driven (backward) stream reasoning

Memo



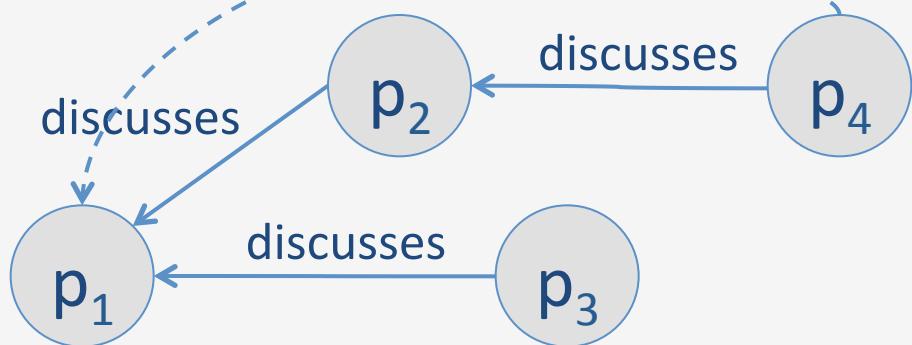
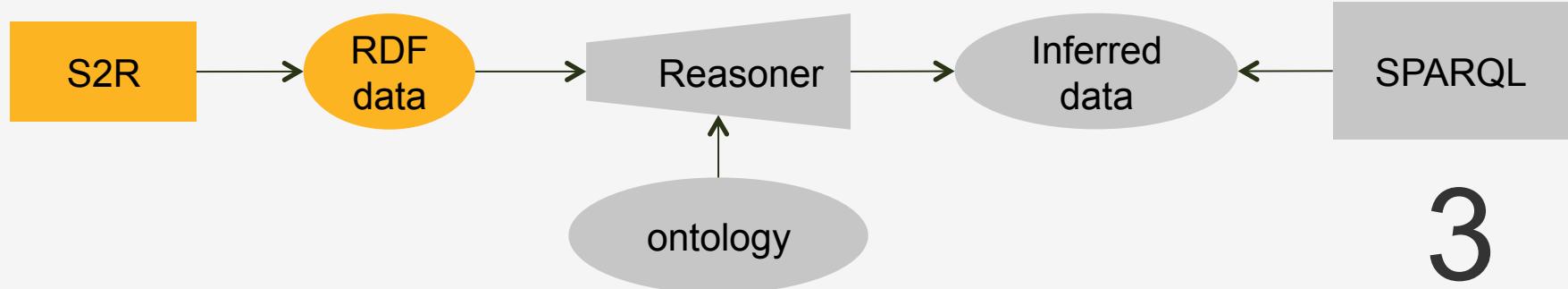
Naïve query-driven (backward) stream reasoning

- Memo



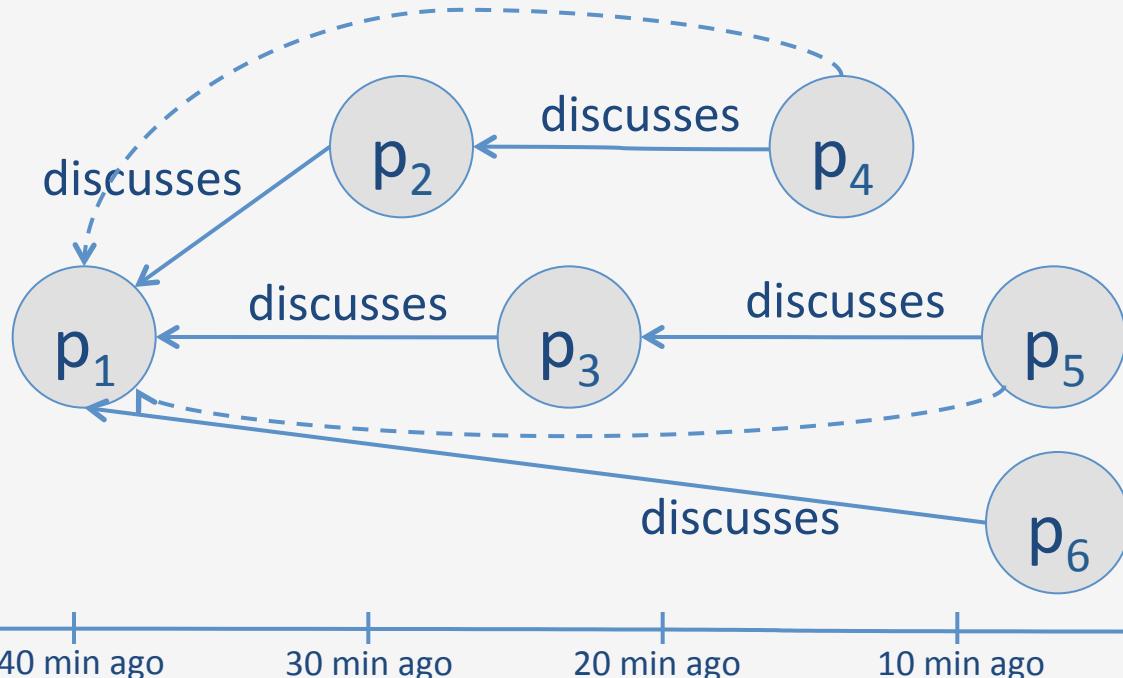
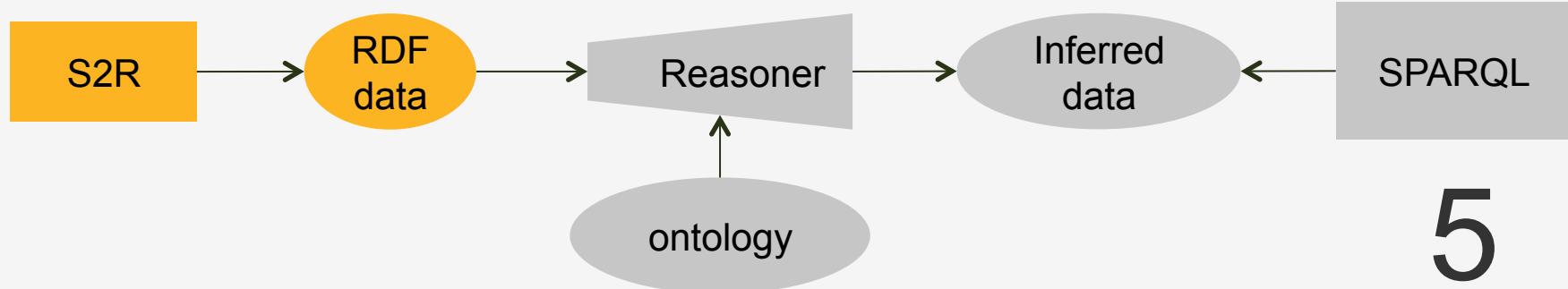
Naïve query-driven (backward) stream reasoning

- Memo



Naïve query-driven (backward) stream reasoning

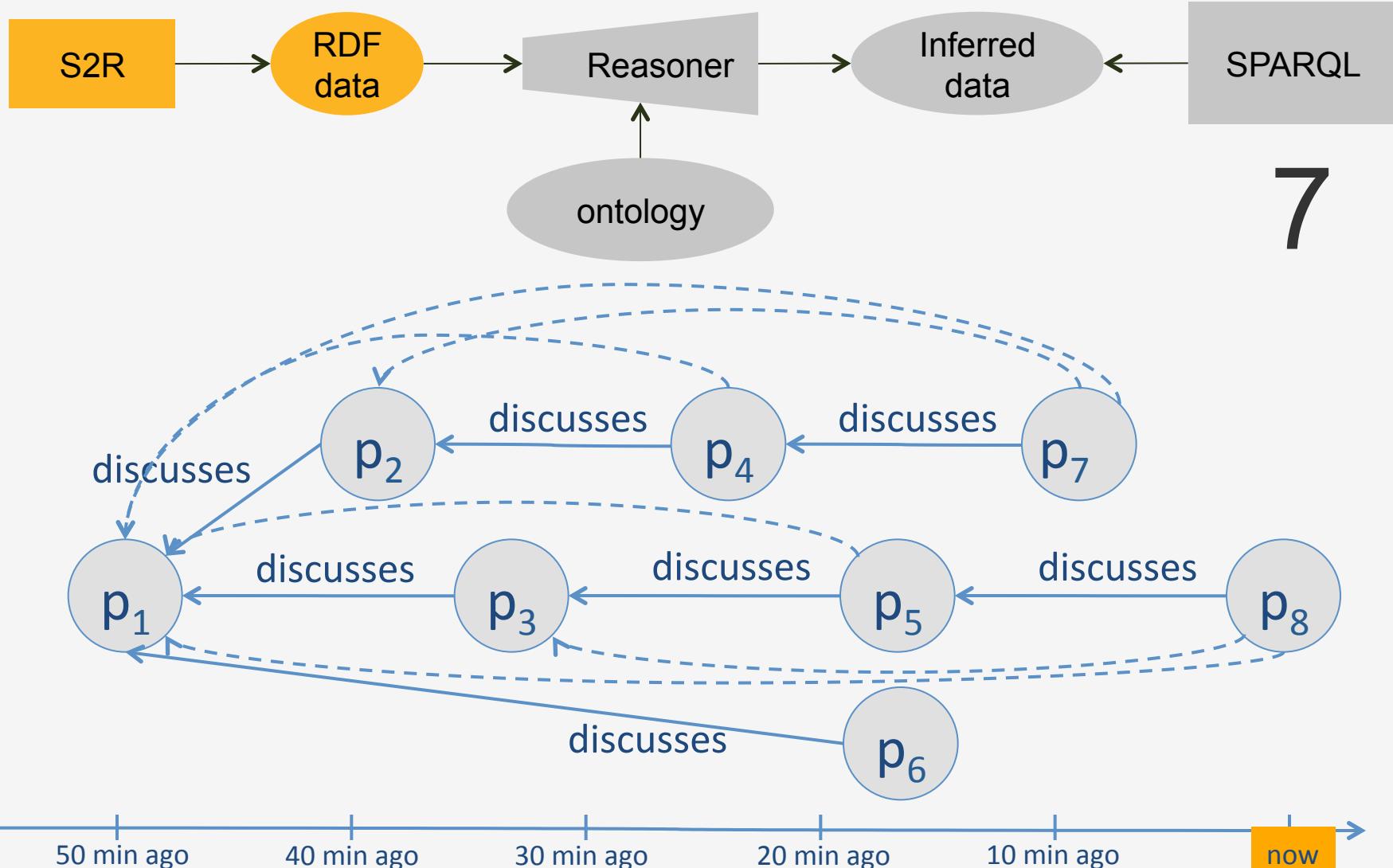
Memo



The entire inference process is repeated each time the SR2 operator delivers new RDF data

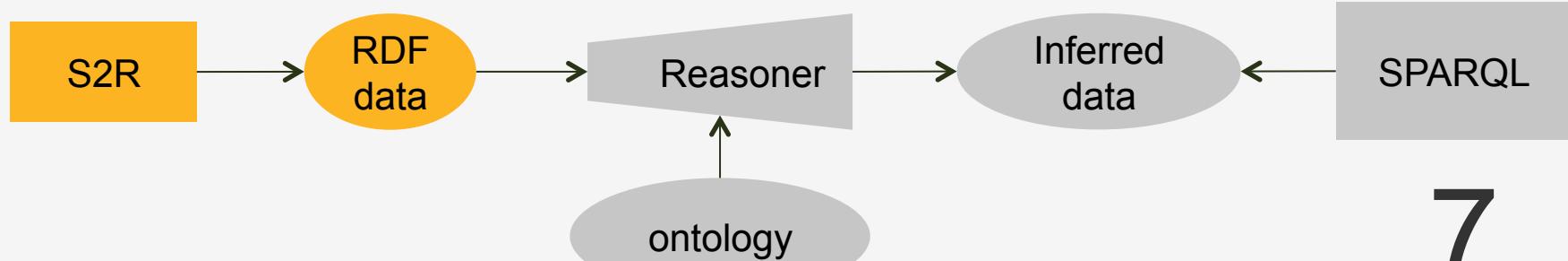
Naïve query-driven (backward) stream reasoning

Memo

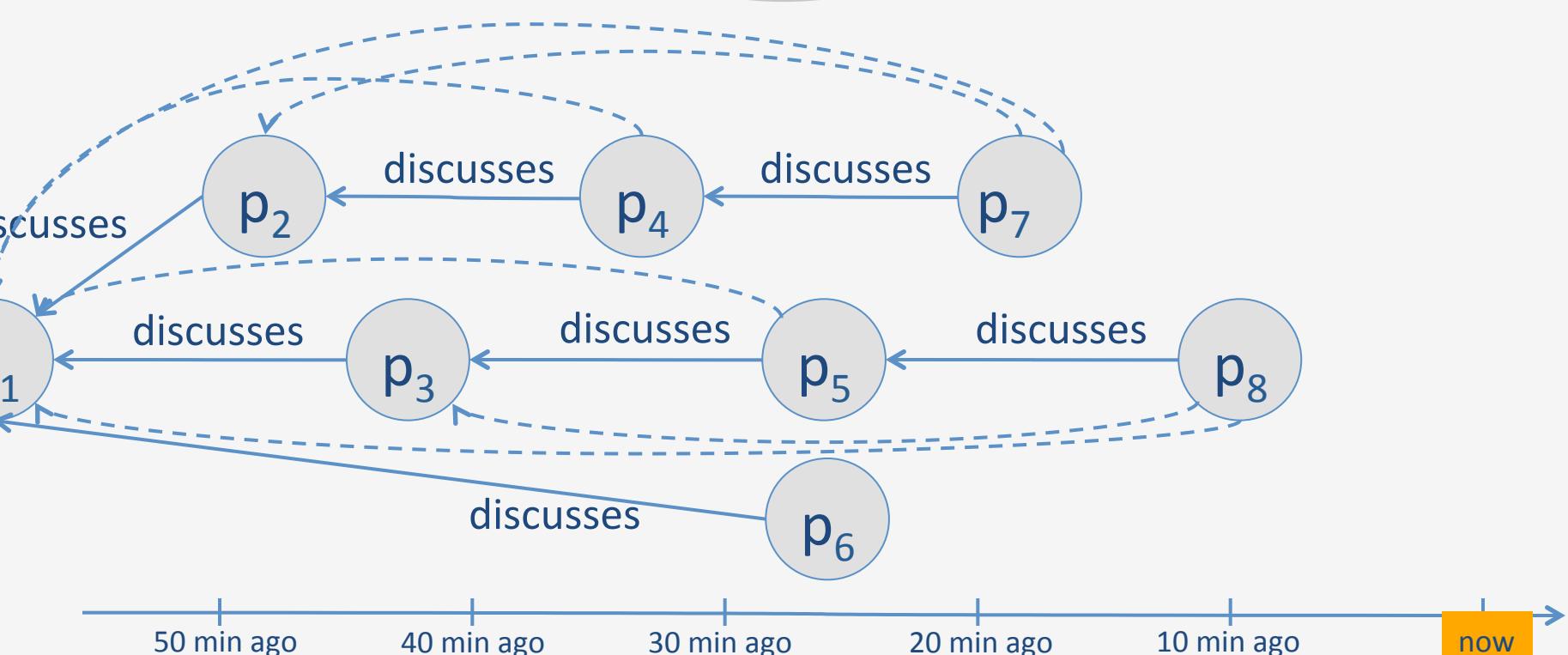


Naïve query-driven (backward) stream reasoning

Memo

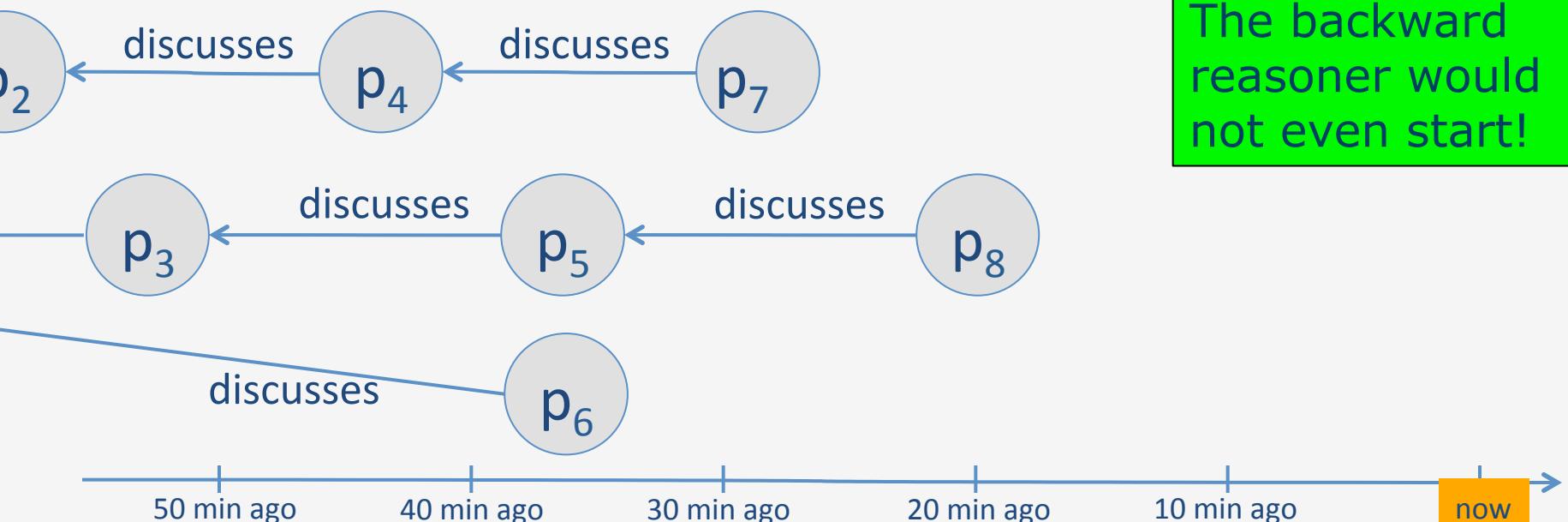
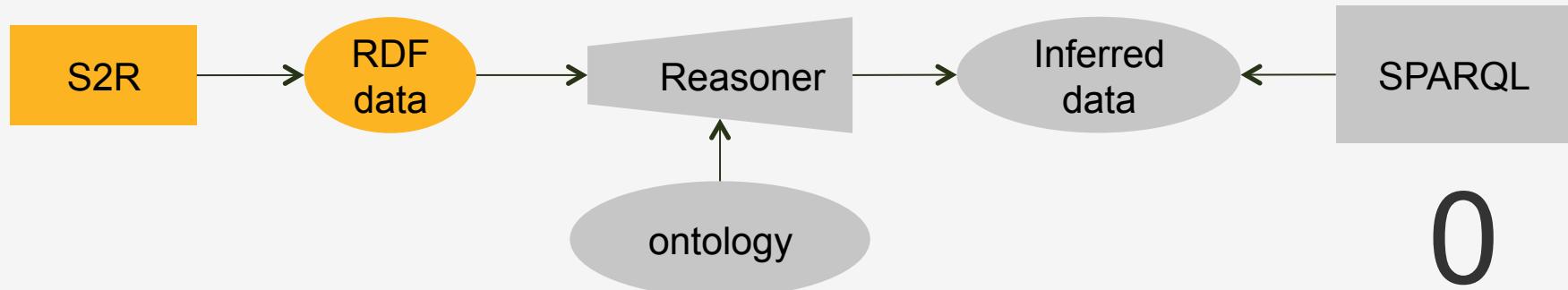


7

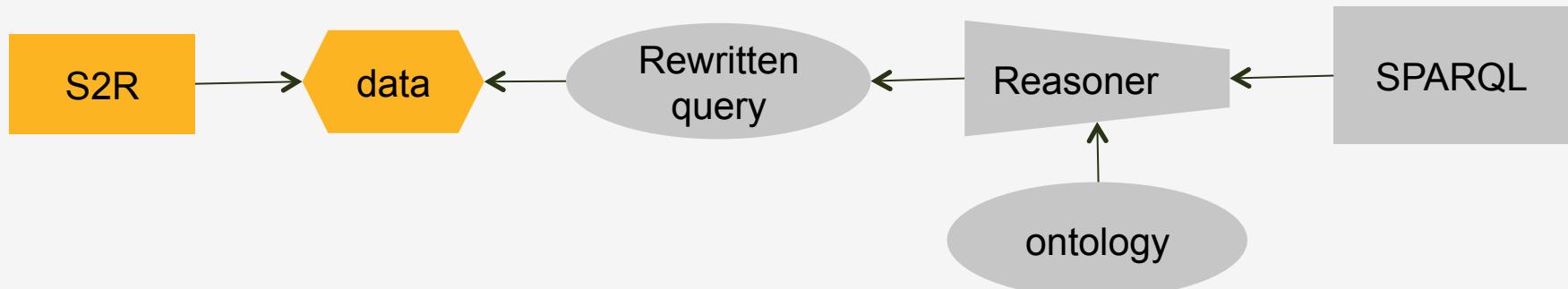


Naïve query-driven (backward) stream reasoning

Memo

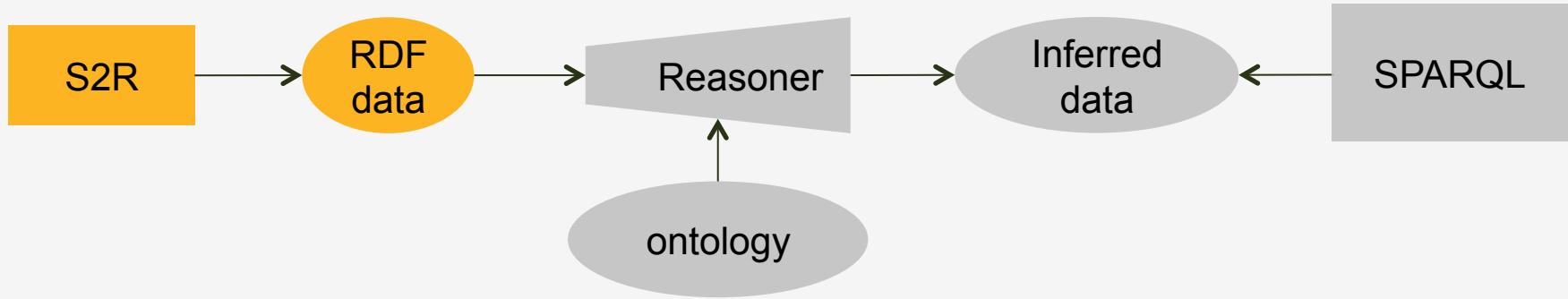


- MEMO

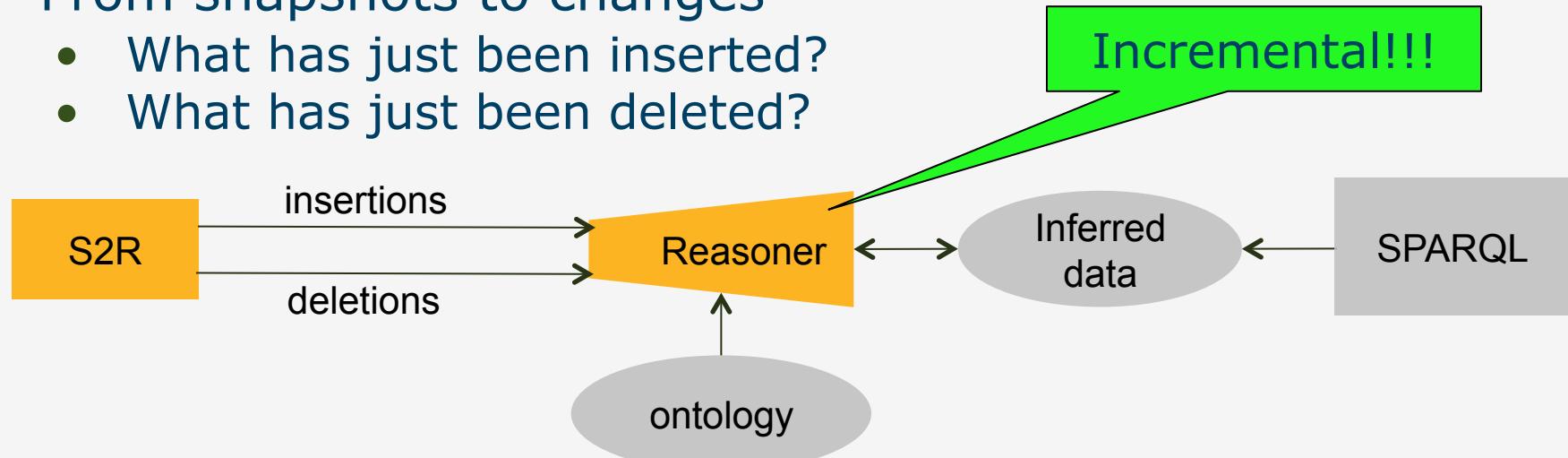


- It is not that straight forward :-(
 - Lack of a standard query language for DSMS and CEP
 - Lack of a well-understood operational semantics for DSMS and CEP (cf. SECRET by I. Botan et al., PVLDB 3(1), 2010)
 - Lack of expressiveness in OWL2QL
 - Temporal reasoning
 - Aggregates
 - Functions

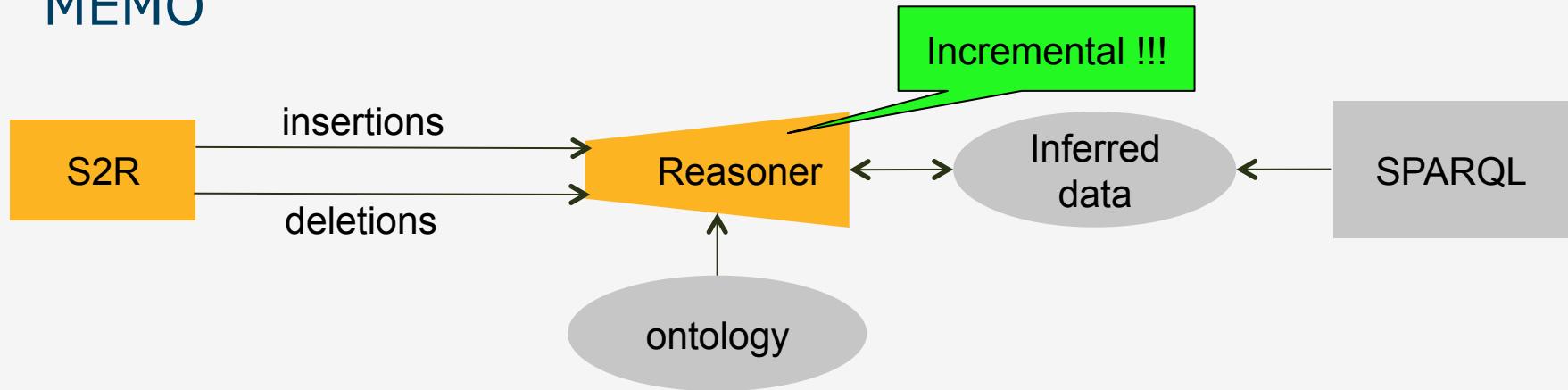
- Naïve data-driven approach



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



- MEMO



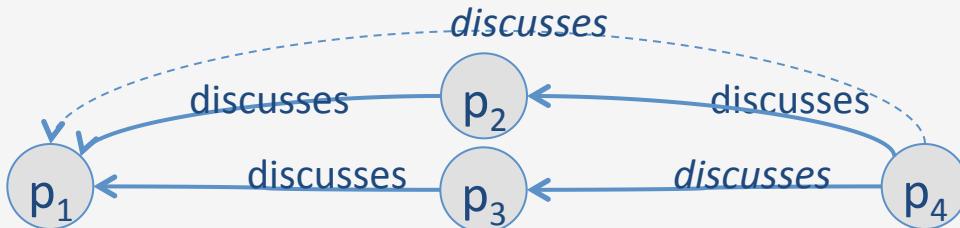
- The problem is that materialization (the result of data-driven processing) are very difficult to decrement efficiently.
 - State-of-the-art: DRed algorithm
 - Over delete
 - Re-derive
 - Insert

Ceri, S., Widom, J.: Deriving production rules for incremental view maintenance. In: Lohman, G.M., Sernadas, A., Camps, R. (eds.) VLDB, pp. 577–589. Morgan Kaufmann, San Francisco (1991)

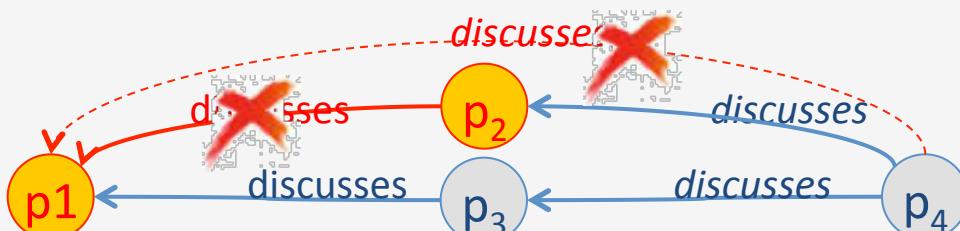
- *Overestimation of deletion*: Overestimates deletions by computing all direct consequences of a deletion.
- *Rederivation*: Prunes those estimated deletions for which alternative derivations (via some other facts in the program) exist.
- *Insertion*: Adds the new derivations that are consequences of insertions to extensional predicates.

The Intuition of DRed Algorithm

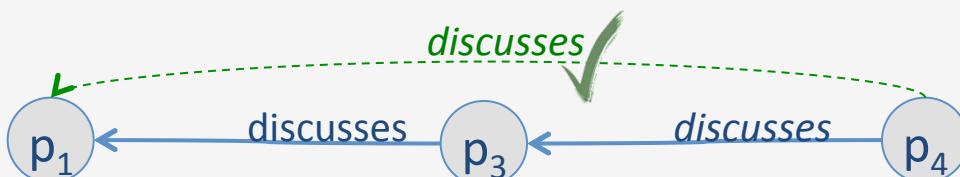
- Let's assume that we have the following materialized graph



- 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 $p_2 \rightarrow p_1$

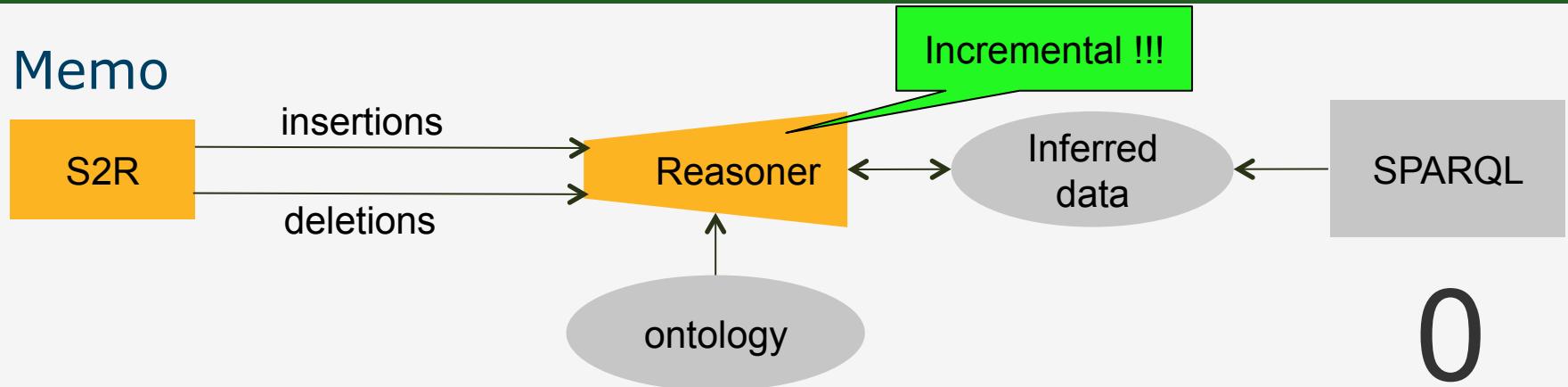


- look for alternative derivation** of $p_4 \rightarrow p_1$ and eventually find the chain $p_4 \rightarrow p_3$ and $p_3 \rightarrow p_1$

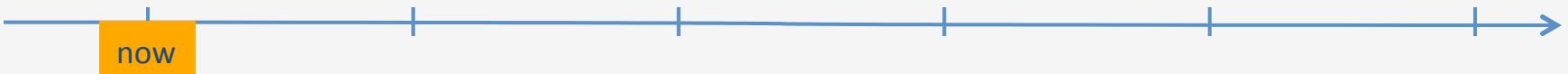


Not so naïve data-driven stream reasoning

- Memo

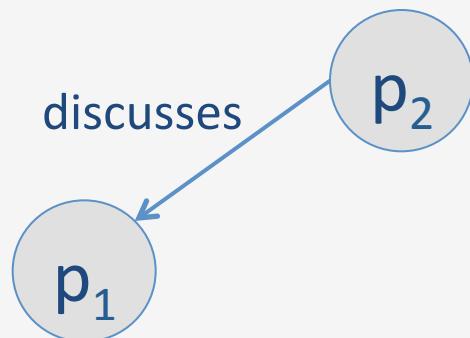
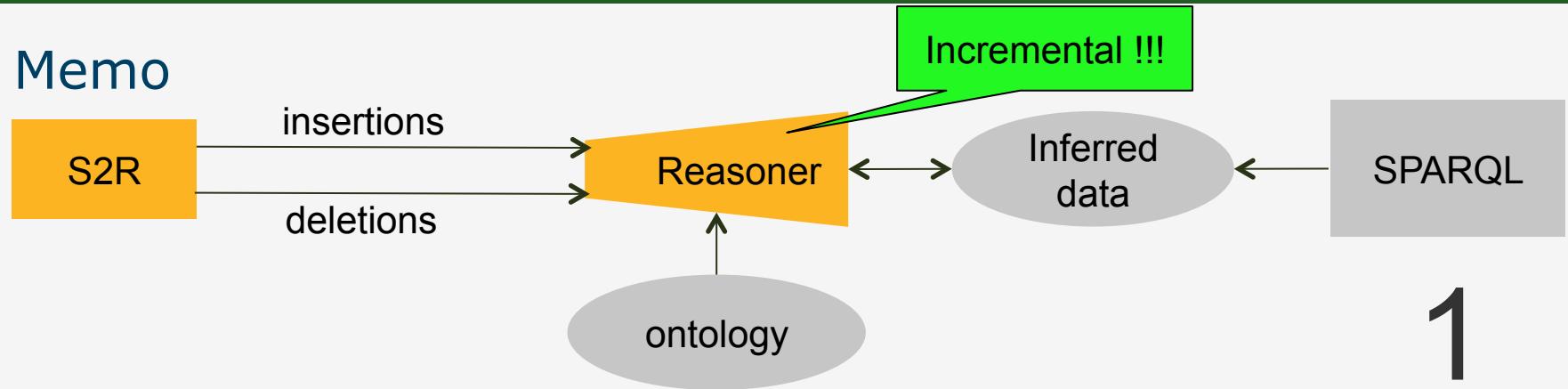


p_1



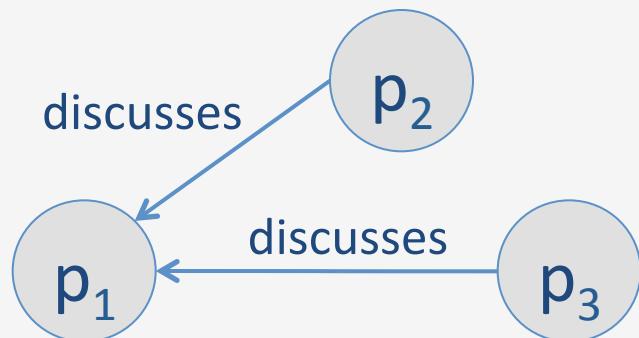
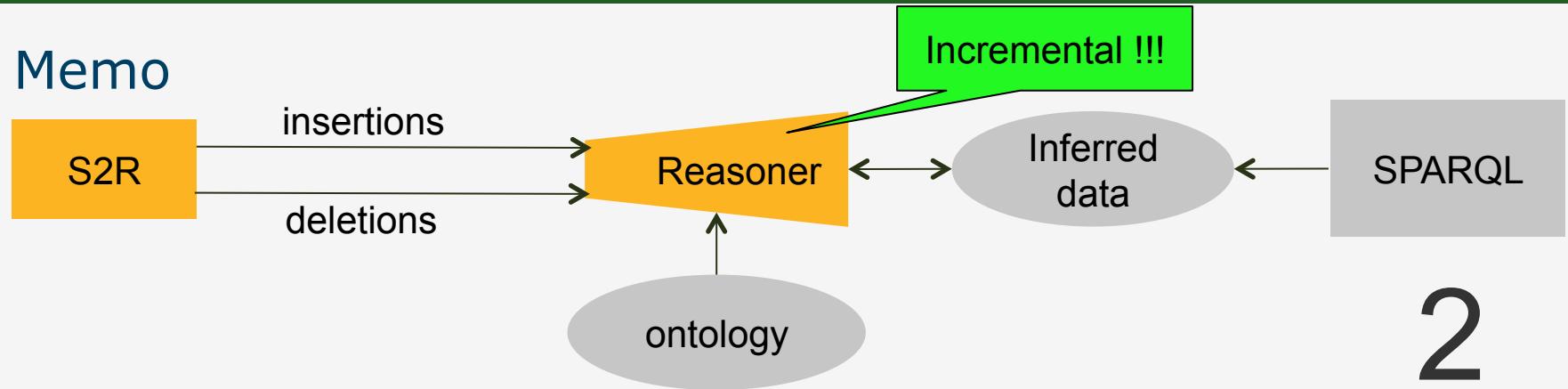
Not so naïve data-driven stream reasoning

- Memo



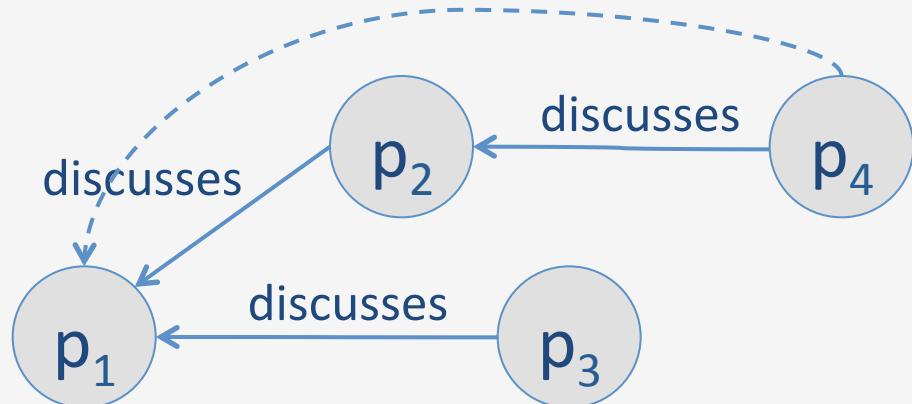
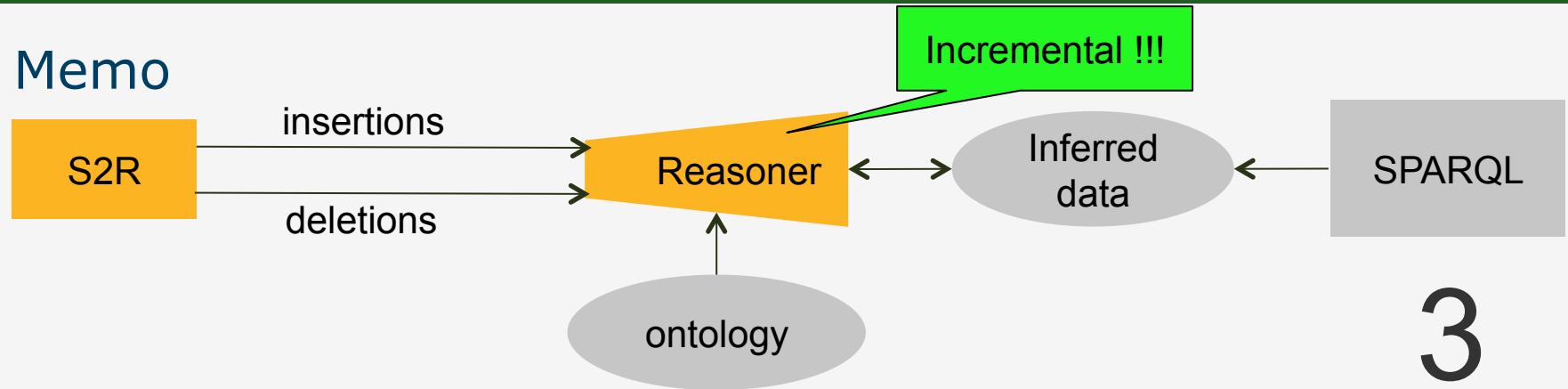
Not so naïve data-driven stream reasoning

- Memo



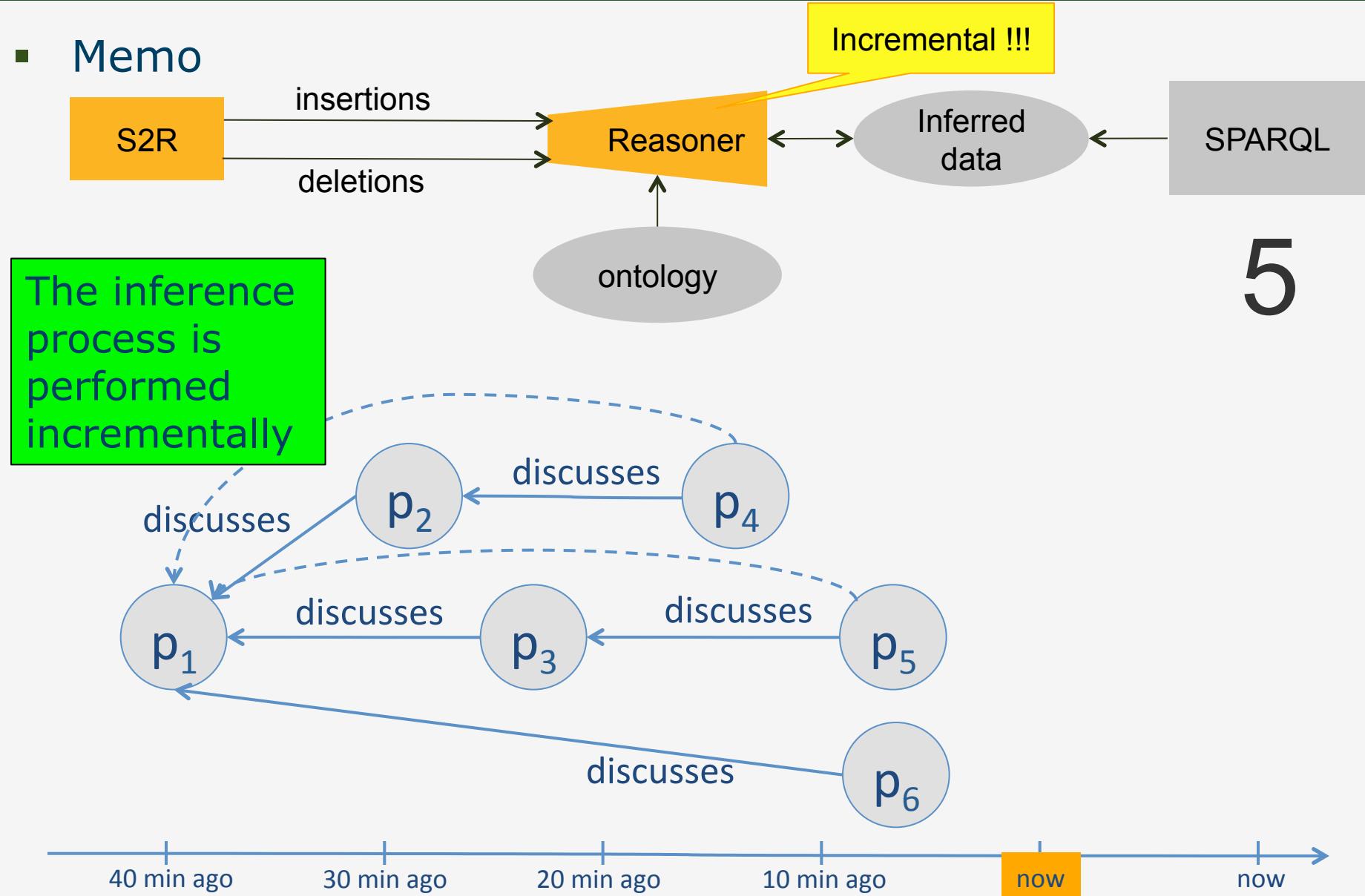
Not so naïve data-driven stream reasoning

- Memo



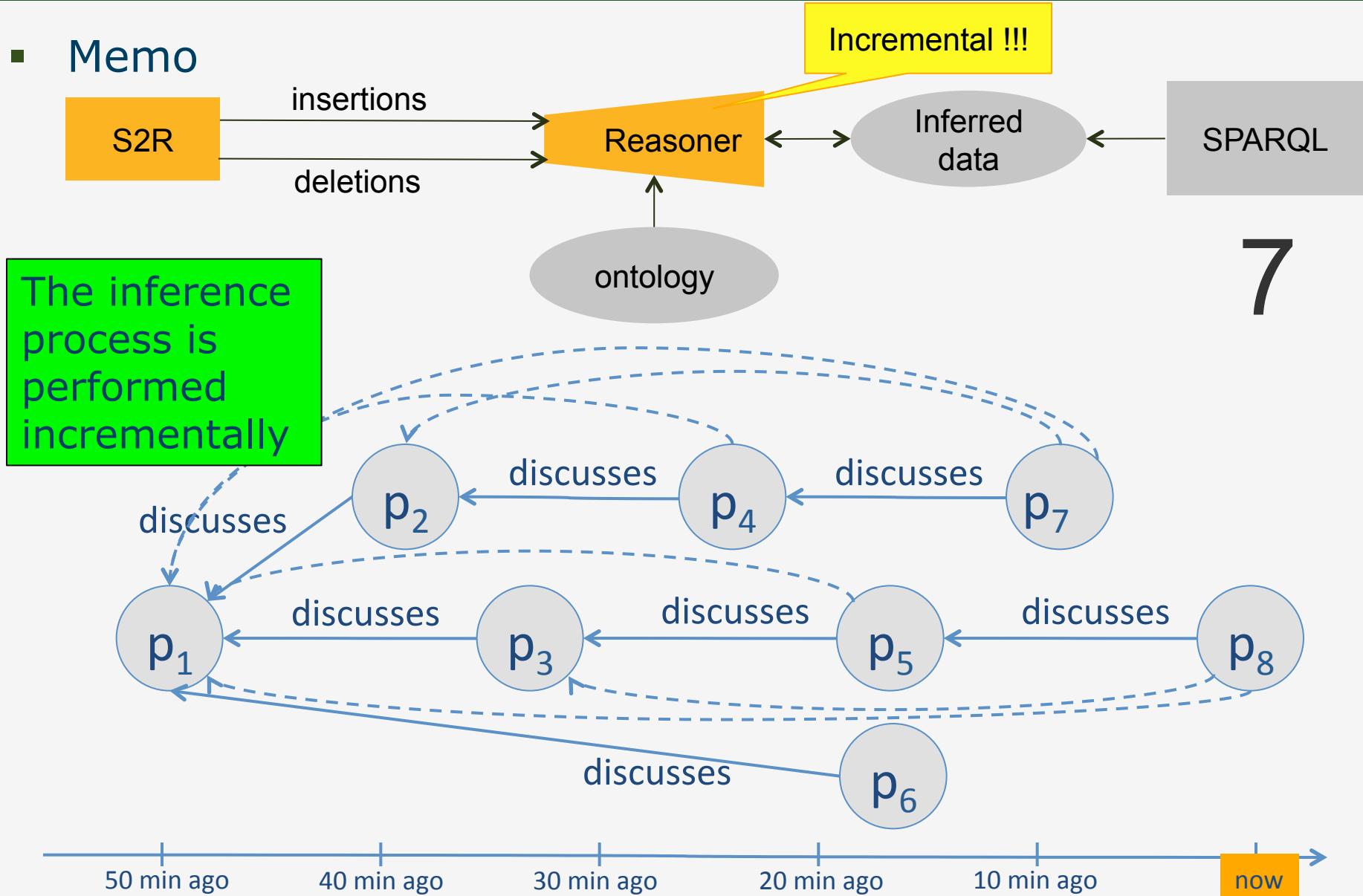
Not so naïve data-driven stream reasoning

Memo



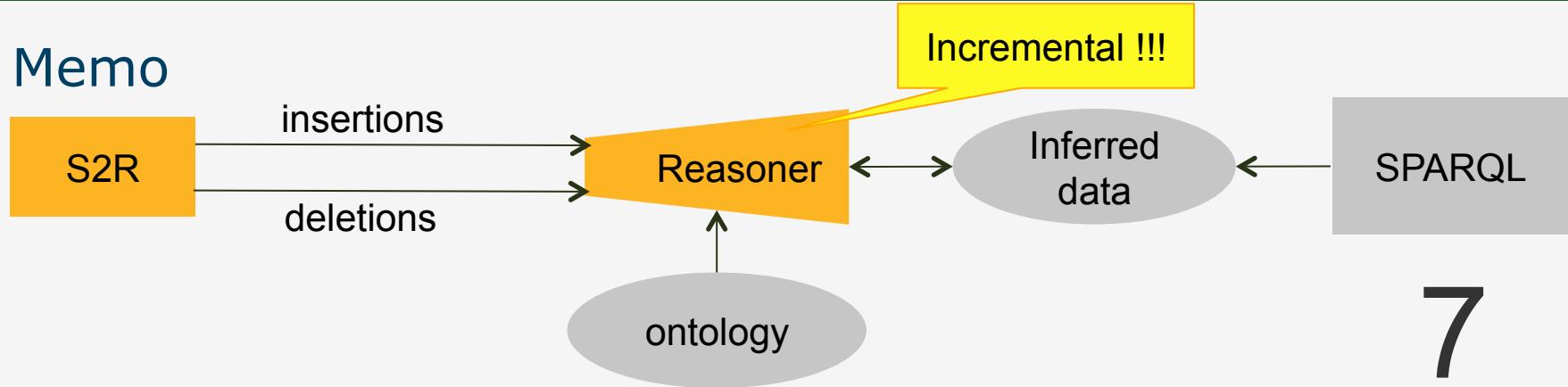
Not so naïve data-driven stream reasoning

Memo

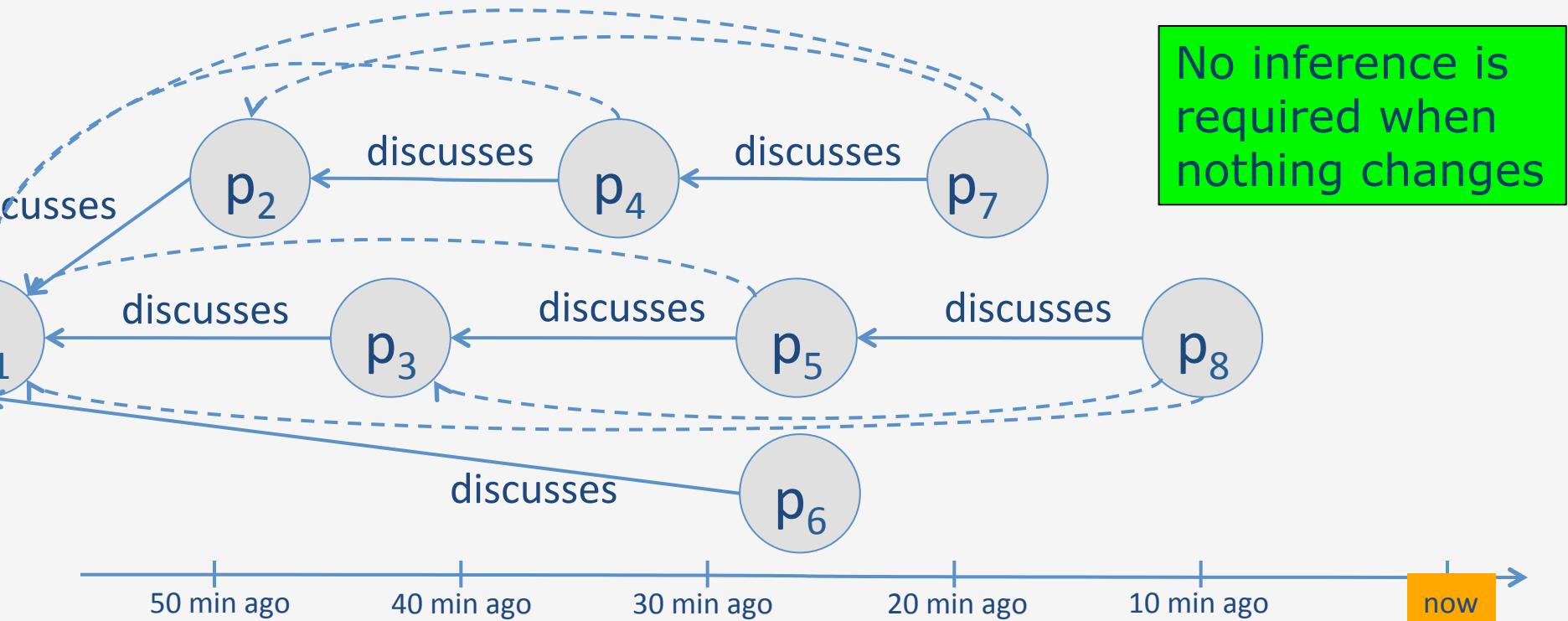


Not so naïve data-driven stream reasoning

Memo

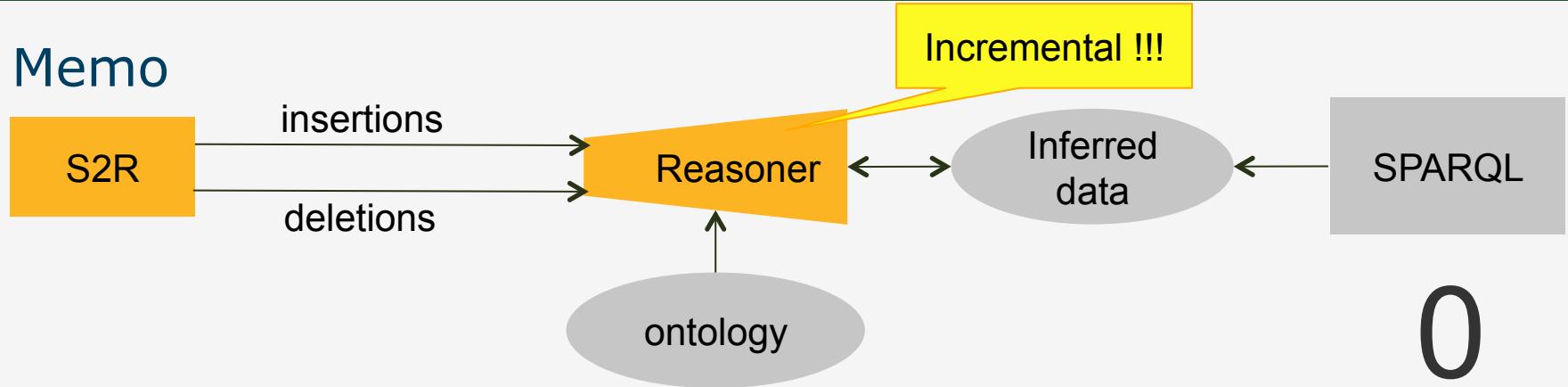


7

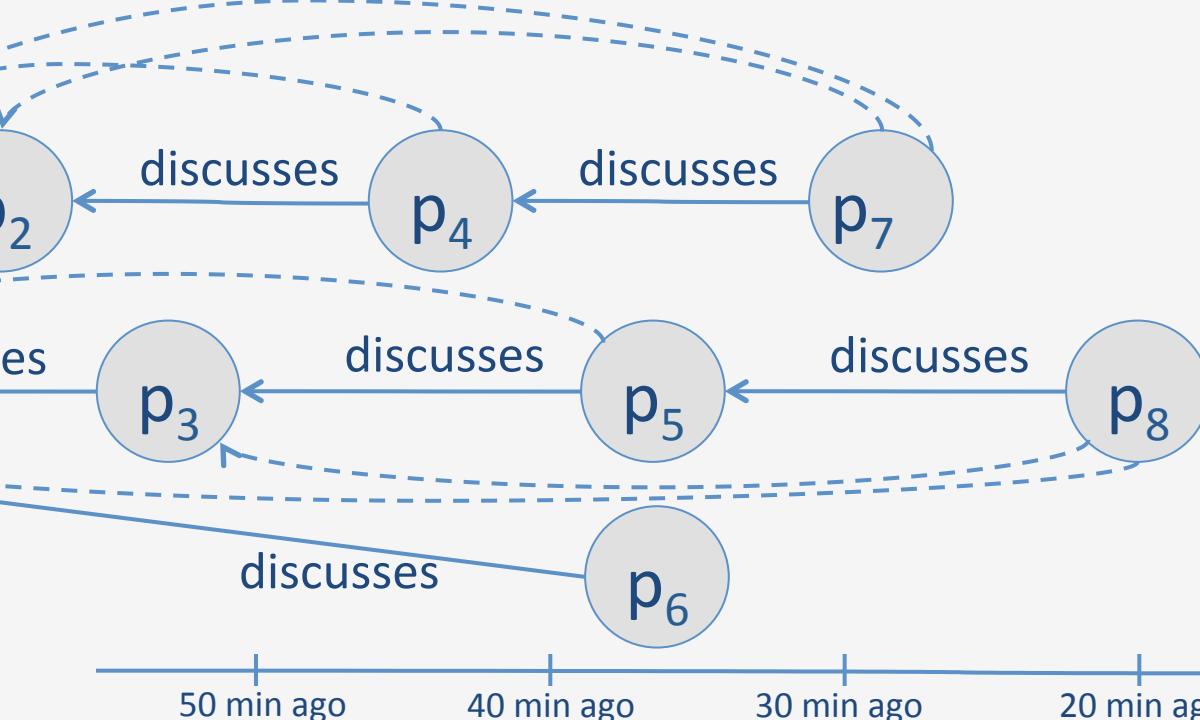


Not so naïve data-driven stream reasoning

Memo



0



Inference is required to process deletions :-(

- 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
- 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
- Query-driven by query rewriting is problematic
 - Lack of a standard query language and well-understood operational semantics for DSMS and CEP
 - Lack of expressiveness in OWL2QL
- 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 process deletions :-(

Stream Reasoning For Linked Data

M. Balduini, J-P Calbimonte, O. Corcho,
D. Dell'Aglio, E. Della Valle, and J.Z. Pan
<http://streamreasoning.org/sr4ld2013>

 ISWC 2013
Sydney, Australia

Naive reasoning on RDF streams

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

