



Stream Reasoning For Linked Data

Jeff Z. Pan and Emanuele Della Valle

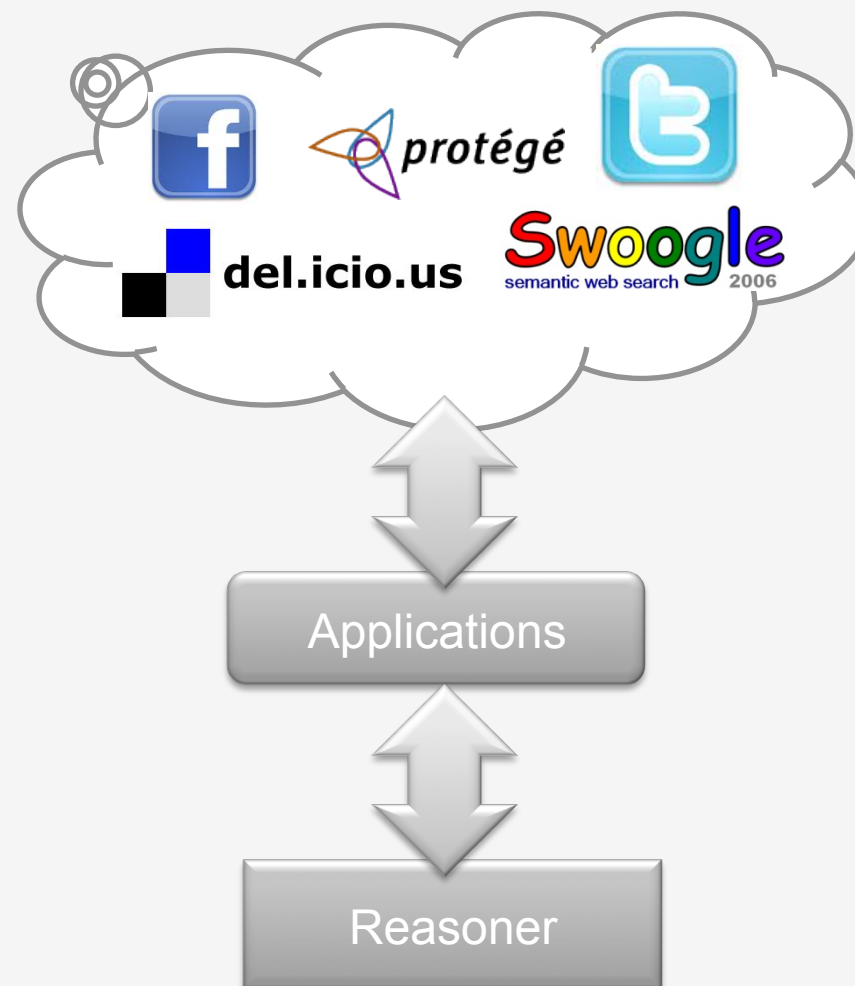


Agenda

- Introduction to Linked Data and OWL 2 (90m)
- C-SPARQL: A Continuous Extension of SPARQL (90m)
- Stream Reasoning techniques for RDFS and OWL2-RL (60m)
- **Approximate Reasoning and Approximate Stream Reasoning for OWL2-DL (60m)**
- Hands-on Session (60m)

- In semantic applications, data and knowledge are subject to change

- Ontology editing
- Web service registry
- Ontology evolution/versioning
- Mobile semantic web
- Sensor network
- ...

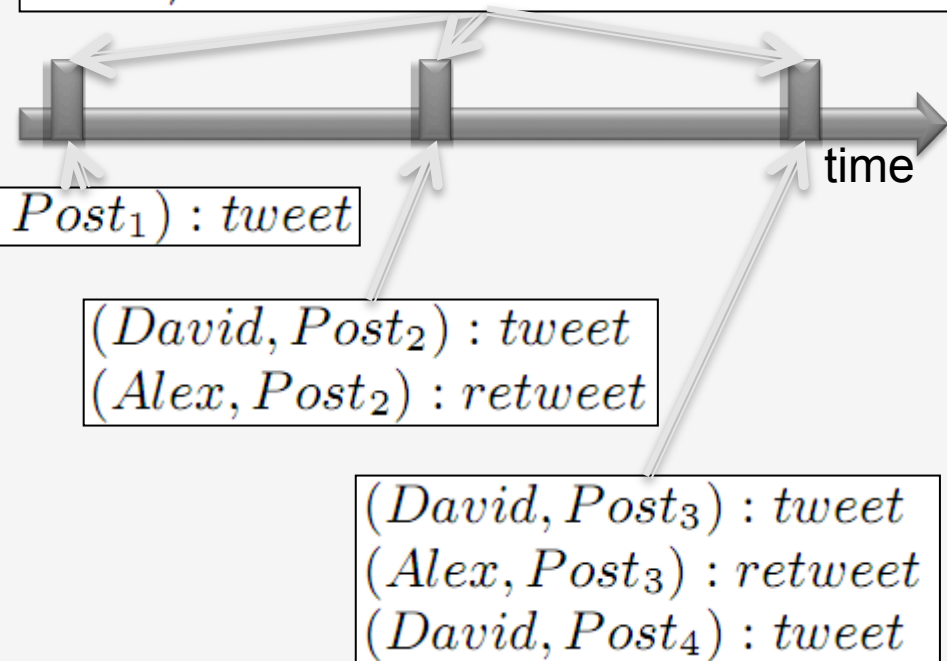


Example

- Suppose we want to find out if anyone is sharing some interest with David and following him on twitter.

$interest \circ interest^- \sqsubseteq shareInterest$
 $retweet \circ tweet^- \sqsubseteq follow$
 $(David, SemanticWeb) : interest$
 $(Alex, SemanticWeb) : interest$
 $Alex \neq David$

- Potential changes
 - David's interests
 - Other's interests
 - David's tweets
 - Other's retweets



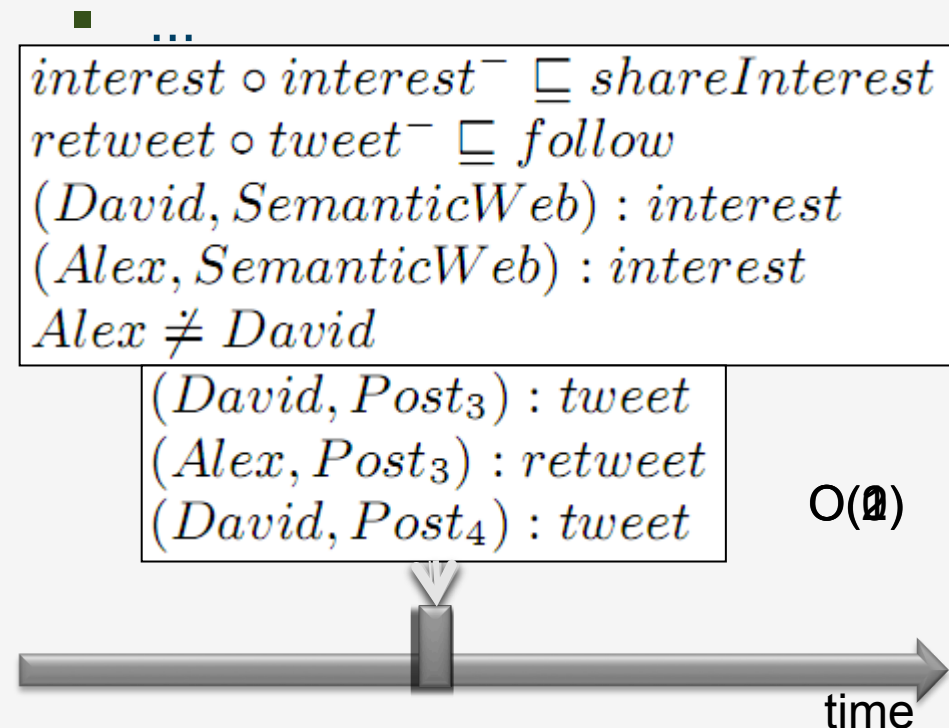
Ontology Stream

■ Definitions

- An ontology stream $O[0,n]$ is a sequence of classical ontologies $O(0), O(1), \dots, O(n)$:
 - $O(0)$ is the initial ontology
- $Ad(i)$ added axioms for $O(i)$
- $Er(i)$ erased axioms for $O(i)$
- $O(i+1) = O(i) \setminus Er(i) + Ad(i)$

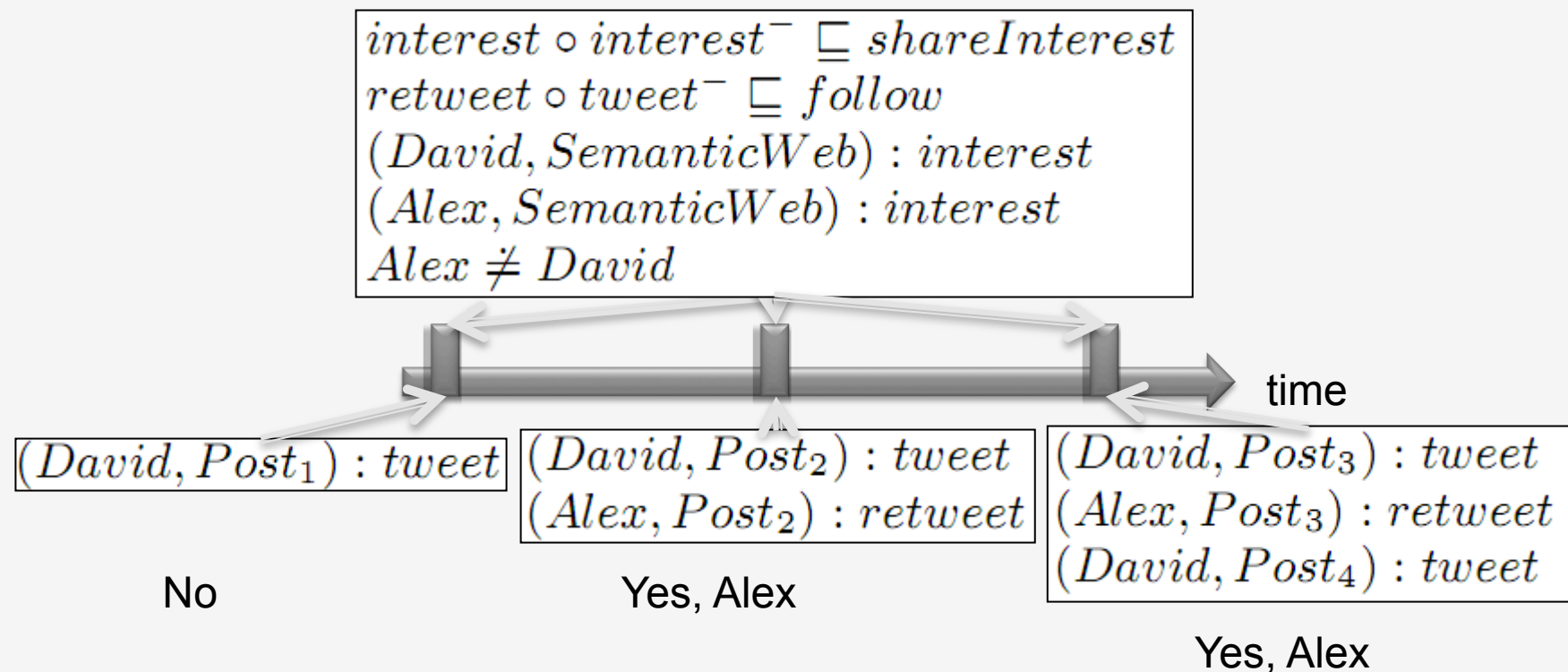
■ Key task

- Answer a set of test queries at each snapshot ontology $O(i)$



Example: Test Query

- ***Is there anyone sharing some interest with David following him on twitter ?***
 - Retrieve instances of

$$\neg\{David\} \sqcap \exists shareInterest.\{David\} \sqcap \exists follow.\{David\}$$


Two Approaches

- Naïve reasoning
 - Re-computing everything on $O(i+1)$
- Stream reasoning
 - Can we re-use results on $O(i)$?
 - Keep the preserved results
 - Eliminate expired results
 - Derive new results

Some Related Works

- Incremental Reasoning [1,2]:
 - **High** complexity, **low** change rate
 - Only deal with $Ad(i)$, **not** $Er(i)$
- Belief contraction [3]
 - Very difficult in DL
 - Different from erasing in stream reasoning
 - Belief contraction can remove any entailed formulas
 - Stream reasoning **ONLY** removes ASSERTED formulas
- C-SPARQL: Continuous SPARQL [4]:
 - Defining a query language for RDF data streams
 - Leaving implementation open

Stream Reasoning: Existing Works

- The DRed (Delete and Re-derive) approach [5]
 - Maintaining the materialisation of the knowledge base
 - Over-delete impacted entailments
 - Re-derive impacted entailments
 - Derive new entailments
- Time-Windowed variant [6]
 - Suitable for fixed time-window size
 - Labelling the entailments with expiration time
 - Delete obsolete entailments w.r.t. their time labels
- Limitations:
 - Expressive power: RDF(S), rule-reducible DLs
 - Relying on the fixed time window

Solution We Seek

- Expressive power
 - Supporting **OWL2 DL** ontology streams
- Less/No prior knowledge
 - Flexible addition and removal of axioms
- Scalability
 - Absolute volume of the stream
 - Relative volume of update
- Efficiency
 - **Low** computational complexity
 - Pragmatic **usefulness** and implementation

- **How to improve scalability**
 - Reducing complexity with approximation
- How to re-use results on $O(n)$?
 - Follow the DRed strategy
 - Maintain the derivation relations among entailments, to
 - Identify the preserved results
 - Identify the expired results
- How to perform incremental reasoning
 - Glass-box approach on preserved results

Why approximations?

- **In short, because**
 - there exist **no algorithms** for query answering in OWL 2 DL
 - TBox reasoning in OWL 2 DL has very high computational complexity (2-NEXPTIME)

What to approximate?

- **Inputs**
 - query
 - ontology
- **Algorithm**



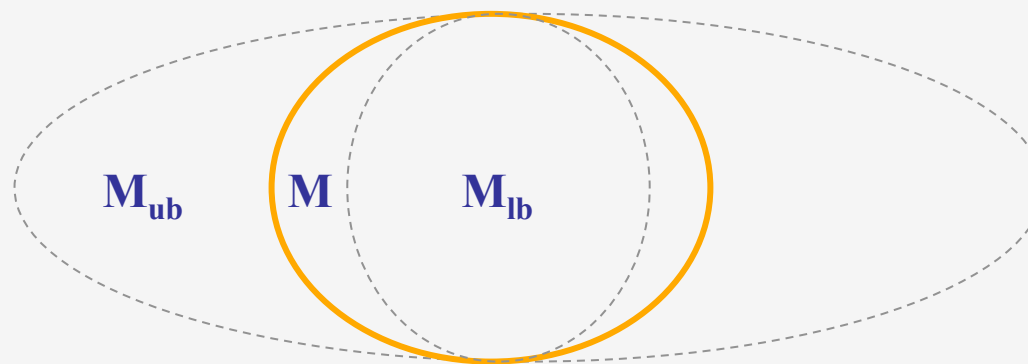
- **Outputs**
 - soundness preserving
 - completeness preserving



Summary: Existing Work

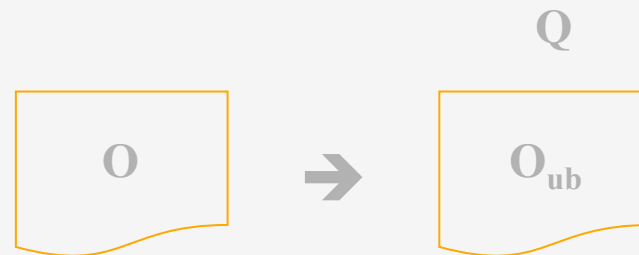
- Most based on syntactic approximations
- Few providing proofs on soundness or completeness

- Knowledge Compilation [Selman and Kautz, 1996]
 - Lower-bound KB_{lb} : $M(KB_{lb}) \subseteq M(KB)$ (hence $KB_{lb} \models KB$)
 - Greatest lower-bound: no KB' such that $M(KB_{lb}) \subseteq M(KB') \subseteq M(KB)$
 - Upper-bound KB_{ub} : $M(KB) \subseteq M(KB_{ub})$ (hence $KB \models KB_{ub}$)
 - Least upper-bound: no KB' such that $M(KB) \subseteq M(KB') \subseteq M(KB_{ub})$



Semantic Approximation [AAAI2007]

- Idea: to approximate a source ontology O (in more expressive L_S) with its (least) upper bound O_{ub} (in less expressive L_T)
 - $O \models O_{ub}$ (soundness guaranteed)

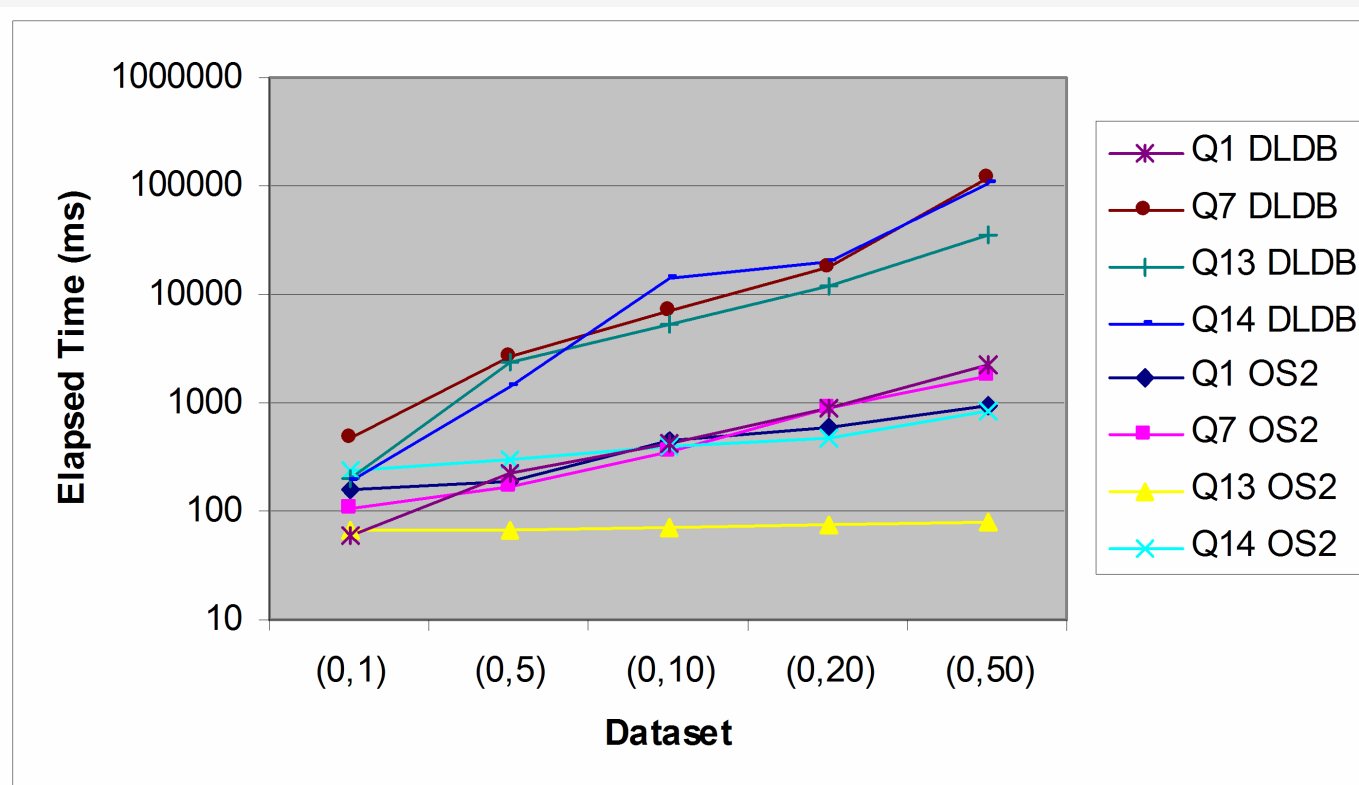


- Least upper bound approximation for QL ES(O , OWL 2 QL) of an OWL2 DL O is **finite** and **unique**.
- **Theorem 1**: Given an ontology O , a conjunctive query $q(X)$ and an evaluation $[X \rightarrow S]$, if $ES(O, OWL\ 2\ QL) \models q_{[X \rightarrow S]}$, then $O \models q_{[X \rightarrow S]}$.
- **Theorem 2**: Given an ontology O_S , a database-style conjunctive query $q(X)$ without non-distinguished variables and an evaluation $[X \rightarrow S]$, $ES(O_S, OWL\ 2\ QL) \models q_{[X \rightarrow S]}$ iff $O_S \models q_{[X \rightarrow S]}$.

- Lehigh University Benchmark
- OWL 2 DL/ SHIQ reasoners [Motik and Sattler, 2006]
 - Pellet and Racer: unable to complete queries 1, 2 and 3 when more than 2 universities
 - KAON2 can answer queries for up to 4 universities
- Syntactic approximation systems [Guo et al., 2004]
 - OWLJessKB can only handle 1 university
 - All results are covered
 - Returned incorrect answers for queries 4, 6, 8 and 12
 - DLDB can handle 50 universities on all queries but query 2
 - Returned incomplete answers for queries 11 and 13

Evaluation: Query Answering (III) (Quill – QL reasoner in TrOWL)

- Quill/ONTOSEARCH2 can handle at least 50 universities
 - Sound and complete results for all queries
- DLDB vs. Quill/ONTOSEARCH2



Faithful Syntactic Approximate Reasoning [AAAI2010]



- Syntactic approximation from OWL2 DL to OWL2 EL
 - Minor syntactic gap results in major complexity difference
 - Using approximation to bridge the gap

DL ROQ (large subset of OWL2 DL)	DL EL++ (large subset of OWL2 EL)
$\top \mid \perp \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\} \mid \boxed{\neg C \mid \geq n R.C}$	$\top \mid \perp \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$
$C \sqsubseteq D$	$r \sqsubseteq s, r_1 \circ \dots \circ r_n \sqsubseteq s$
$a : C$	$(a, b) : r$
N2EXPTIME-complete	PTIME-complete

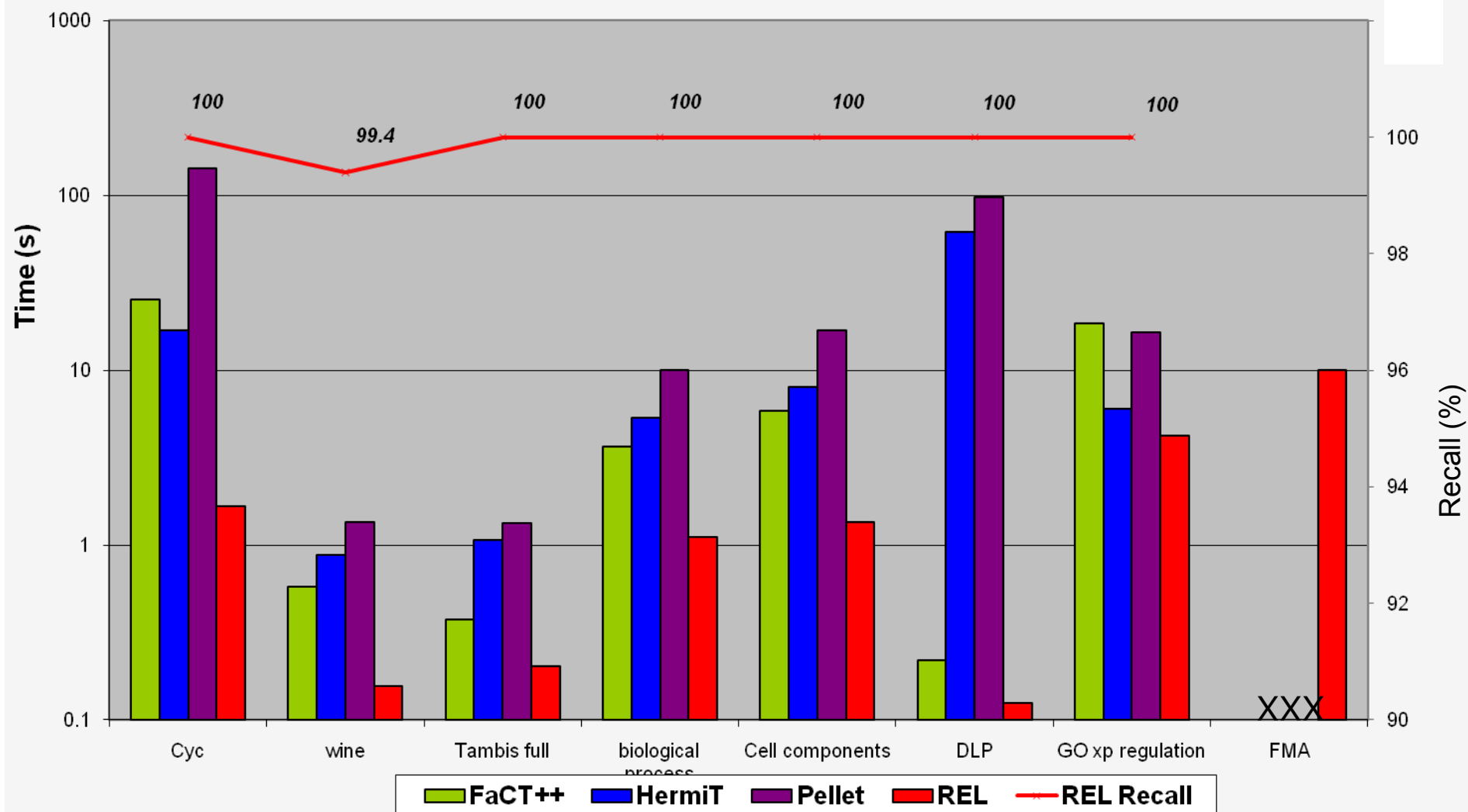
- We provide faithful syntactic approximation for OWL2 DL TBox reasoning
 - Soundness preserving
 - Practically close to complete
- **Challenge**
 - removing axioms with non-OWL2 EL descriptions could lead to low recall



How Does it Work

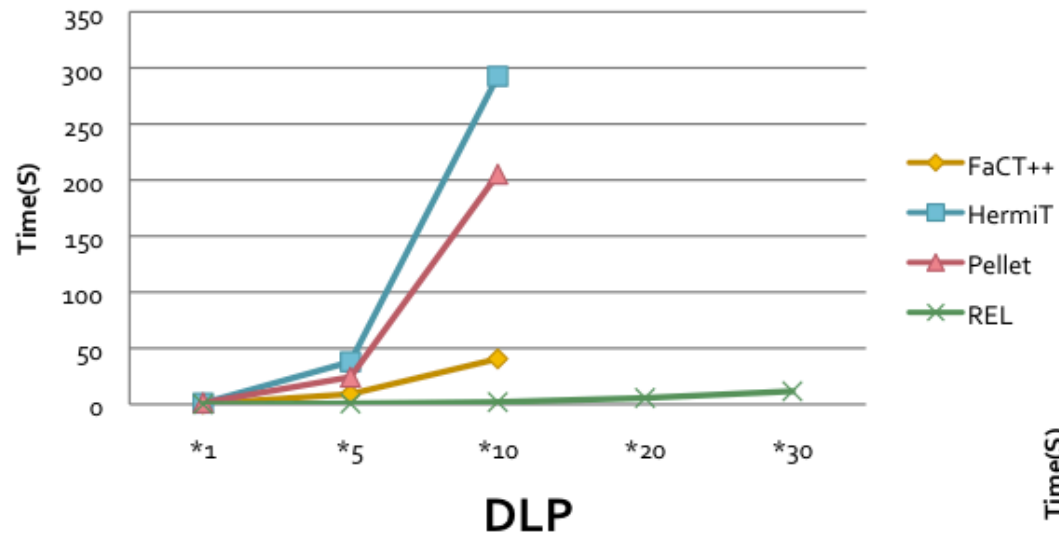
- TBox approximation
 - Directly represent non-OWL2-EL concepts with fresh named concepts
 - E.g., $\forall r.C \text{ subClassOf } D \rightarrow A_{\forall r.C} \text{ subClassOf } D$
 - Maintain semantic relations for these named concepts
 - *complementary relations*
 - *cardinality relations*
- ABox approximation
 - Internalise ABox into TBox
 - E.g., $C(a) \rightarrow \{a\} \text{ subClassOf } C$
 - $r(a,b) \rightarrow \{a\} \text{ subClassOf } \exists r.\{b\}, \dots$
- Reasoning
 - Using additional **tractable** completion rules to recover the semantics

Evaluations for the Oxford Benchmarks (REL – EL reasoner in TrOWL)

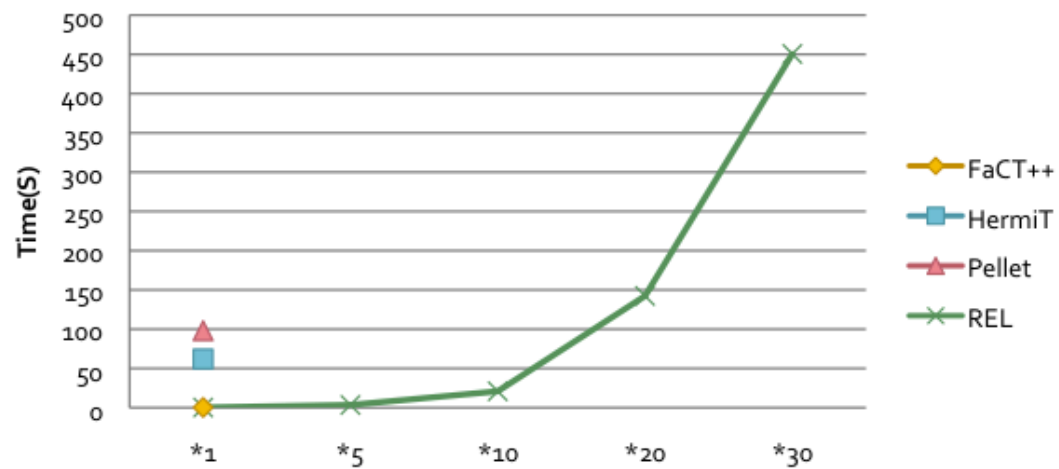


Evaluation: Scalability Comparison

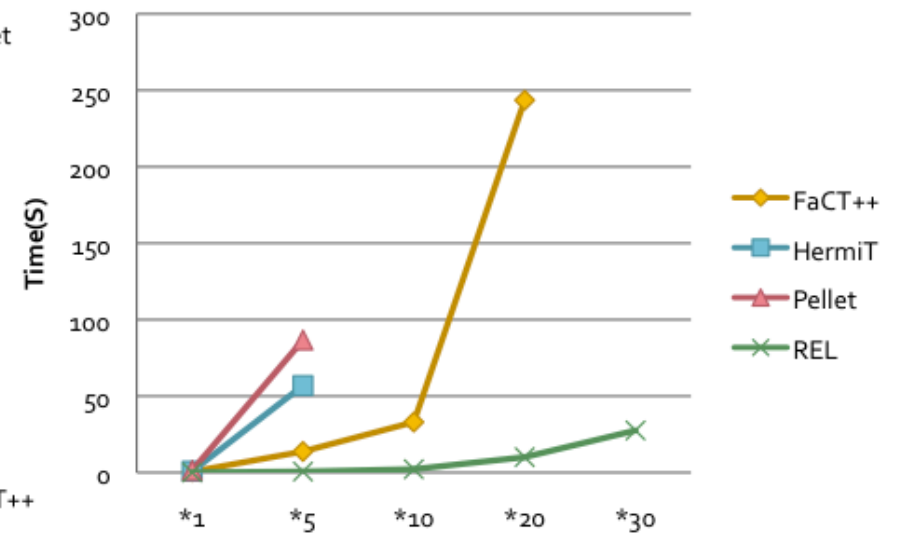
Tambius Full



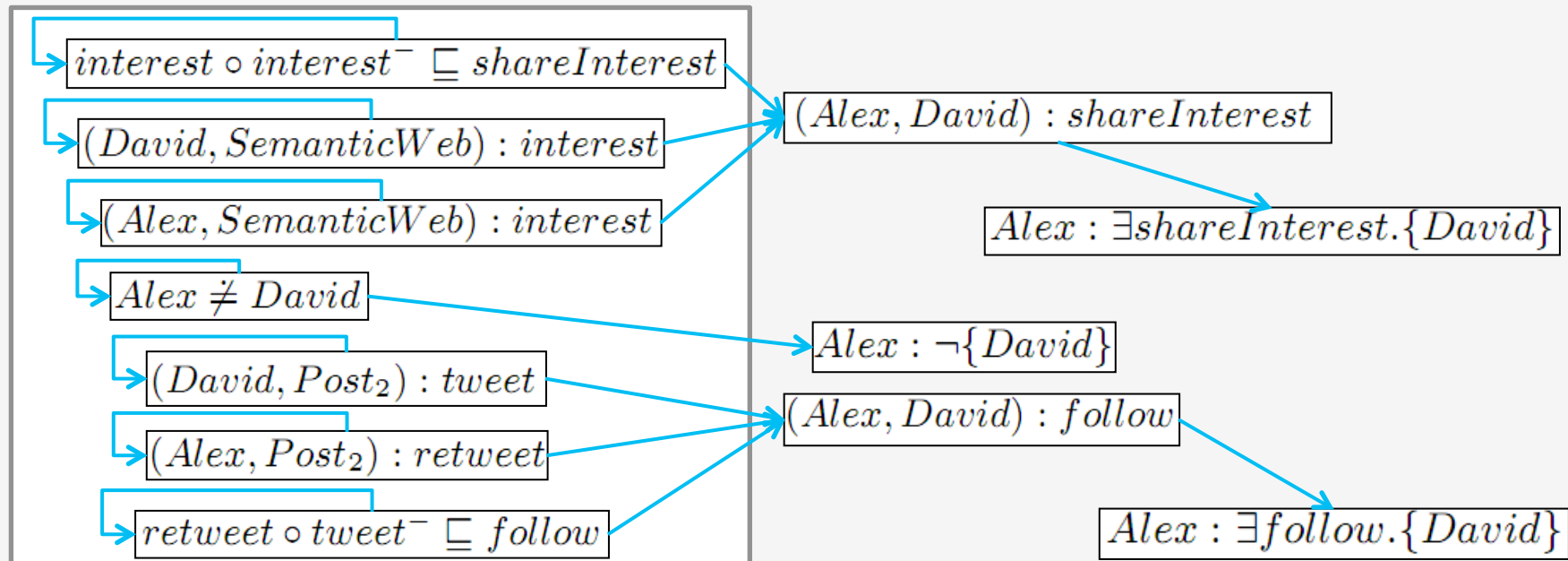
DLP



Wine

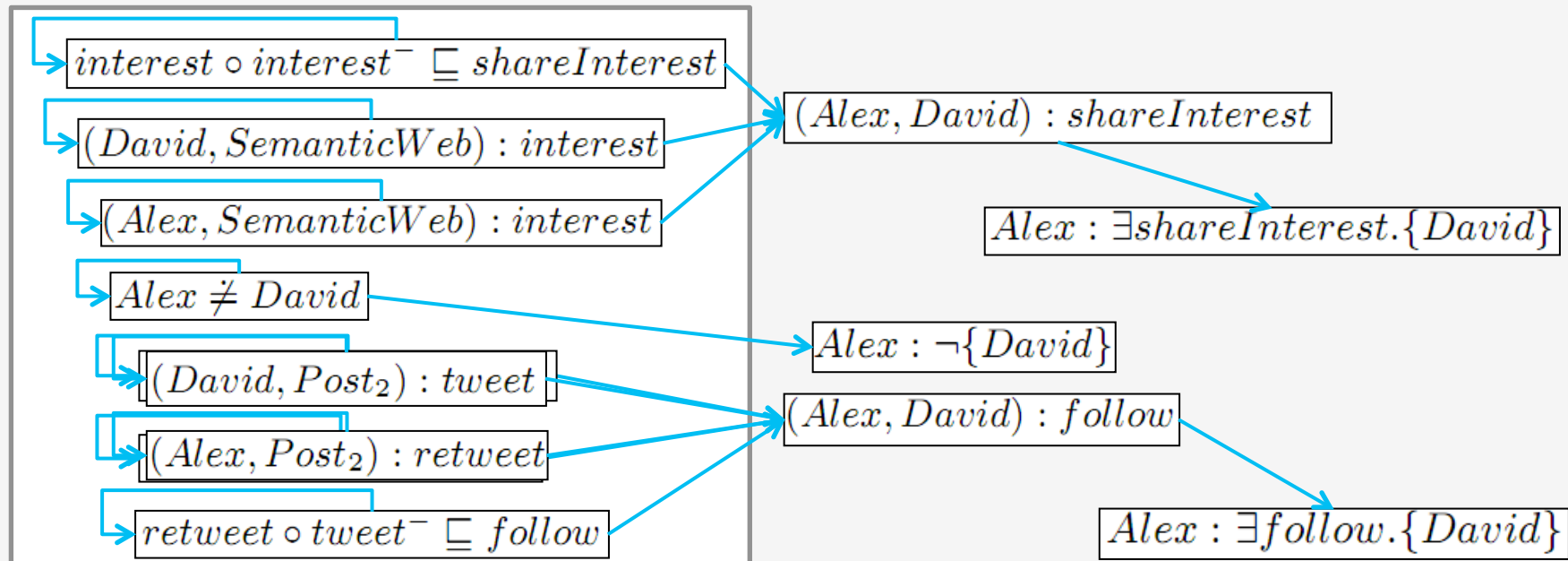


- A directed graph
 - Nodes: Entailments
 - Edges: Derivation relations among entailments
 - All entailments are reachable from their justifications
 - Easy to identify impacted entailments



Stream Reasoning with TMS

- Erasing
 - Remove all nodes reachable from the erased axioms
 - Removing all corresponding edges
- Adding
 - Adding added axioms as new nodes into the graph
 - Inferring new results
 - Establishing new edges

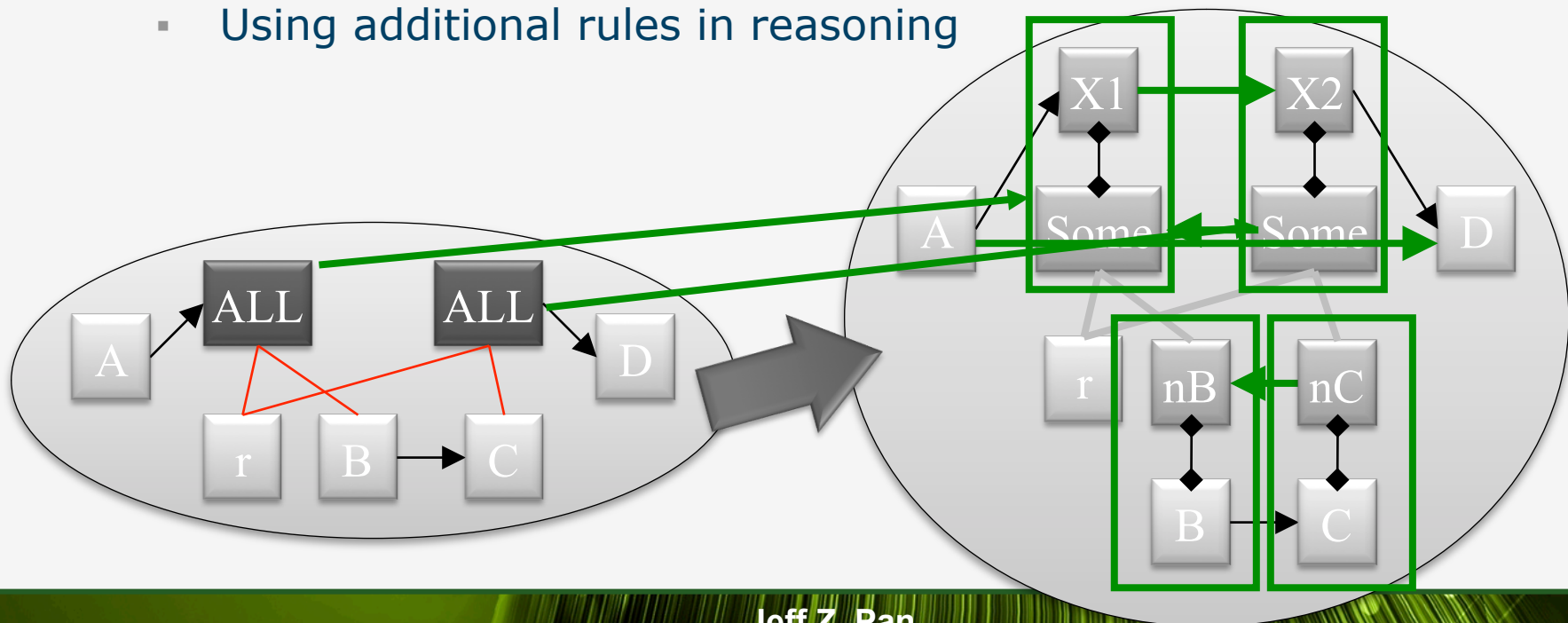


Features and Optimizations

- Advantage
 - Significantly **reduce** the needed number of entailment checking
- Disadvantages & optimizations
 - Require **additional computation** to form the edges
 - Using saturation-based algorithms to generate the TMS **on-the-fly**
 - EL+, Horn-SHIQ vs. Tableau-based or Datalog-based algorithms
 - Significantly increase the **memory consumption**
 - Saturation algorithms already make intermediate results more **“reusable”**
 - Optimizing the algorithm to further **minimise intermediate results**
 - **High complexity** in expressive languages
 - Syntactic approximation

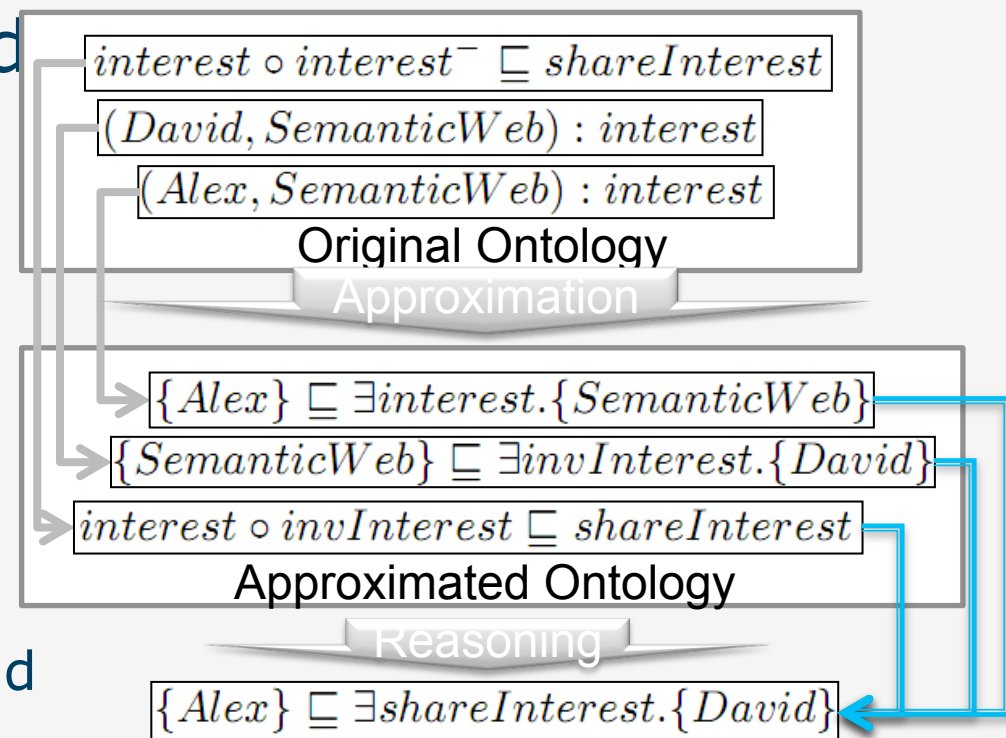
Approximating Stream Reasoning

- Syntactic Approximation [7]
 - Approximating OWL2-DL ontologies into OWL2-EL ontologies
 - Naming non-OWL2-EL predicates
 - Maintaining additional semantics in other data structures
 - Using additional rules in reasoning

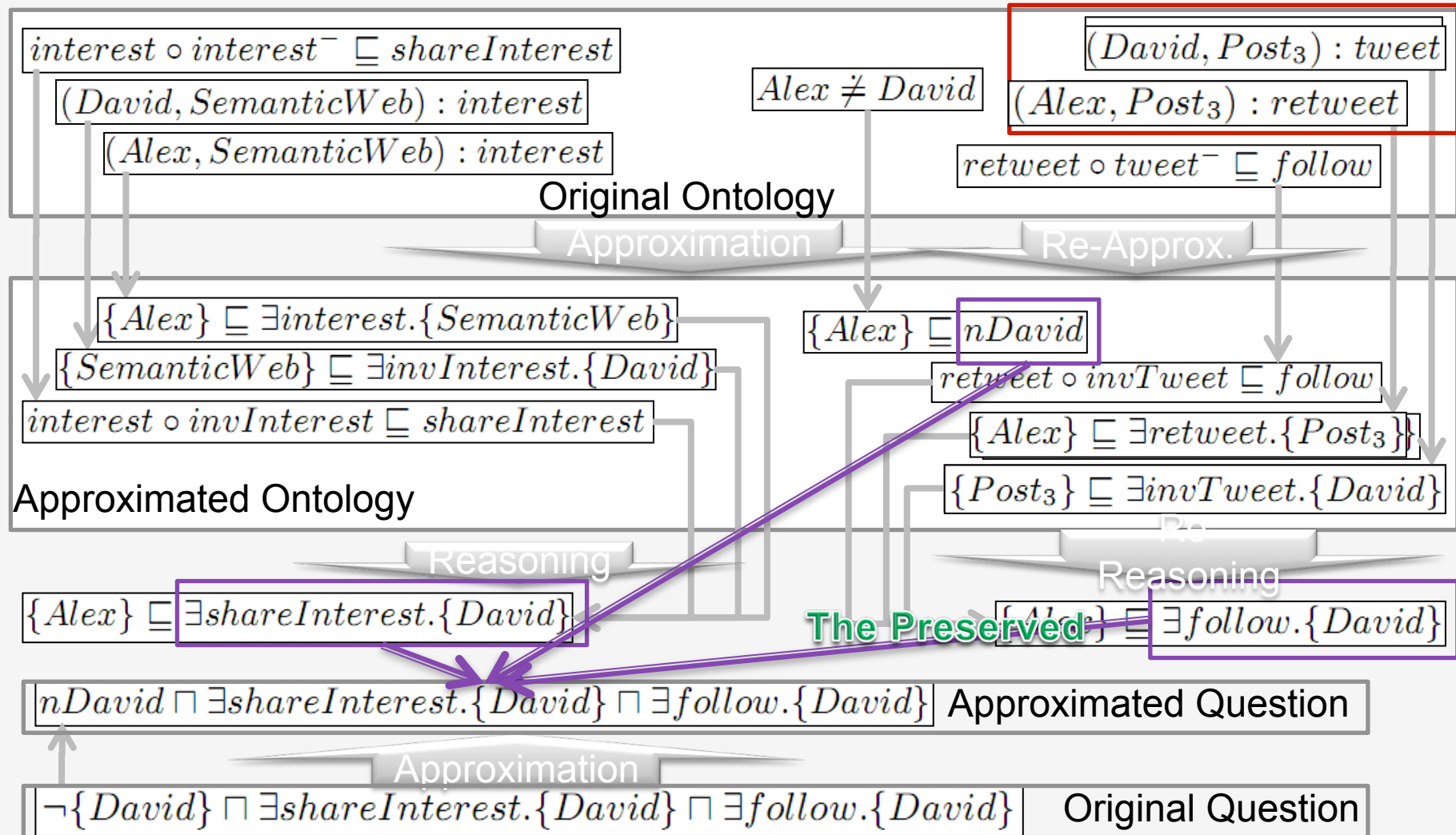


- Generate a TMS when doing approximation and reasoning

- Nodes:
 - Asserted axioms;
 - Approximated axioms;
 - Entailed axioms;
- Edges:
 - Created during approximation and reasoning



Stream Reasoning with Syntactic Approximated-TMS



Stream Reasoning with Syntactic Approximated-TMS

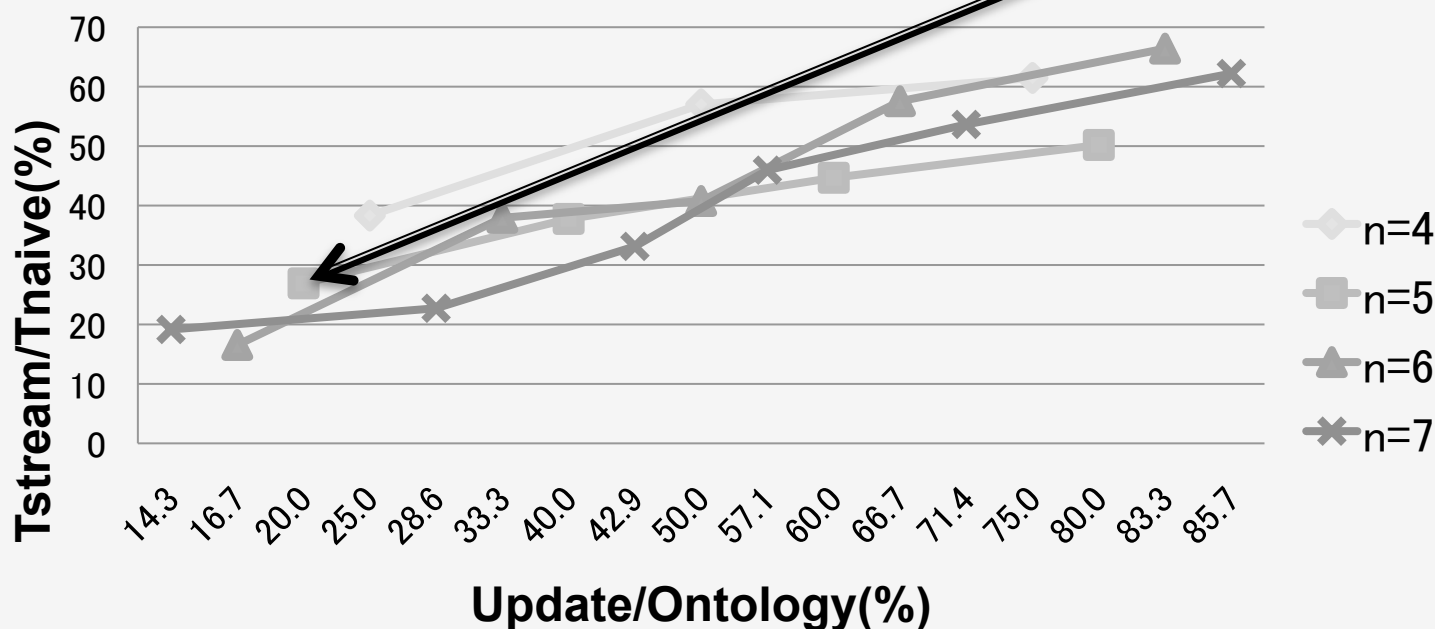
- Improvement:
 - The approximation can be reused when adding new axioms
 - Reasoning complexity is reduced
- Quality:
 - Soundness-guaranteed
 - Entailed axioms will never be over-looked in erasing

- LUBM
 - Arbitrarily large ABox
 - Non-trivial reasoning
 - Different possible sources of a same entailment
- Stream simulation
 - Generating a LUBM ontology
 - Preserve the TBox through the stream
 - To ensure the difficulty of reasoning
 - ABox stream
 - Partitioning the ABox
 - Swap sub-ABoxes in stream updating

Criteria

- Absolute volume of the stream
- Relative volume of the stream
- Absolute time of updating
- Relative time of updating

Approach	$\mathcal{O}(0)$	$Max(\mathcal{O}(i))$	$Min(\mathcal{O}(i))$	$Ave(\mathcal{O}(i))$
Naive		13.897	10.667	11.5997
Stream Reasoning	19.223	3.982	2.175	2.5076



- Stream reasoning can significantly improve the flexibility of reasoners
- TMS
 - Preserve intermediate results
 - Preserve derivation relations
- Approximation
 - substantially reduce reasoning complexity
 - higher updating ratio
- Future works:
 - Using ontology stream to maintain historic knowledge
 - Supporting reasoning regarding history

- TrOWL is a tractable OWL2 reasoning infrastructure developed at Aberdeen [ESWC2010]
 - It supports many pluggable reasoners
 - It works with OWL API, Protégé 4, Jena and Semantic Mediawiki
 - Configuration through property files and feature modelling:
 - Performance required
 - Determine the best reasoner for a specific task
- Quality guaranteed transformations
 - Faithful approximate reasoning
 - Quill: OWL 2 DL -> OWL 2 QL (semantic approximation) [AAAI2007]
 - REL: OWL 2 DL -> OWL 2 EL (syntactic approximation) [AAAI2010]
 - Divide and conquer
 - Modularisation [ISWC2009]
 - Forgetting [ESWC2008,ISWC2009b]
- Coming up soon: Local Closed world reasoning with NBox, SPARQL1.1, Stream Reasoning

Acknowledgement



- Aidan Hogan
- Ian Horrocks
- Nophadol Jekjantuk
- Axel Polleres
- Yuan Ren
- Edward Thomas
- Yuting Zhao





Stream Reasoning For Linked Data

Jeff Z. Pan and Emanuele Della Valle

