Efficient Identification of Implicit Facts in Incomplete OWL2-EL Knowledge Bases

John Liagouris, Manolis Terrovitis

IMIS - Research Center “Athena”

Web Ontology Language (OWL)

- Extends the RDF Schema (RDFS)
  - rdfs:Class, rdfs:Property
- Complex class expressions
  - Woman $\equiv$ Person $\sqcap$ Female
- Complex property expressions
  - hasMother $\circ$ hasSister $\sqsubseteq$ hasAunt
- Property characteristics
  - We can define a RDF property like “isSiblingWith” as *symmetric* and *transitive*

Web Ontology Language (OWL)

- The semantics of OWL imply additional knowledge, i.e., new RDF triples

<mary rdf:type Woman> + “Woman ≡ Person ⊓ Female”

$\Downarrow$

<mary rdf:type Person>
<mary rdf:type Female>

- Such “hidden” triples cannot be queried directly with traditional SPARQL engines
- Reasoning is needed

Complexity of Reasoning in OWL

- Exponential for OWL in general

- Three tractable fragments:
  - OWL2-QL
  - **OWL2-EL**
  - OWL2-RL

- Each fragment poses different restrictions in the syntax of OWL

Problem

- We are given a large collection of OWL2-EL axioms and a set of inference rules

- Goal: Infer all axioms that are implied by the rules

- How: Apply all rules to the collection of axioms exhaustively till no new axioms are produced (fix-point)

Running Example

We are given a collection of OWL2-EL axioms of the form
$\mathbf{X} \sqsubseteq \mathbf{Y}$:

1. InfectedWithVirusA $\sqcap$ NotVaccinated $\sqsubseteq$ Ill
2. $\exists$Vaccinated.VaccineTypeX $\sqsubseteq$ NotVaccinated
3. Vaccinated1994 $\sqsubseteq$ $\exists$Vaccinated.{va}
4. {va} $\sqsubseteq$ VaccineTypeX
5. {john} $\sqsubseteq$ Vaccinated1994
6. {john} $\sqsubseteq$ InfectedWithVirusA

Running Example

2. $\exists$Vaccinated.VaccineTypeX $\sqsubseteq$ NotVaccinated
3. Vaccinated1994 $\sqsubseteq$ $\exists$Vaccinated.{va}
4. {va} $\sqsubseteq$ VaccineTypeX

From axioms 3, 4 and 2 $\longrightarrow$ 7. Vaccinated1994 $\sqsubseteq$ NotVaccinated

Running Example

5. $\{john\} \sqsubseteq Vaccinated1994$

From axioms 3, 4 and 2 $\rightarrow$ 7. $Vaccinated1994 \sqsubseteq NotVaccinated$

From axioms 5 and 7 $\rightarrow$ 8. $\{john\} \sqsubseteq NotVaccinated$

Running Example

1. InfectedWithVirusA $\sqcap$ NotVaccinated $\sqsubseteq$ Ill

6. {john} $\sqsubseteq$ InfectedWithVirusA

From axioms 3, 4 and 2 $\longrightarrow$ 7. Vaccinated1994 $\sqsubseteq$ NotVaccinated

From axioms 7 and 5 $\longrightarrow$ 8. {john} $\sqsubseteq$ NotVaccinated

From axioms 6, 8 and 1 $\longrightarrow$ 9. {john} $\sqsubseteq$ Ill

Challenges

- Inference rules for OWL2-EL are complex and mutually recursive (each one affects the other)

- The collection of axioms does not always fit in main-memory

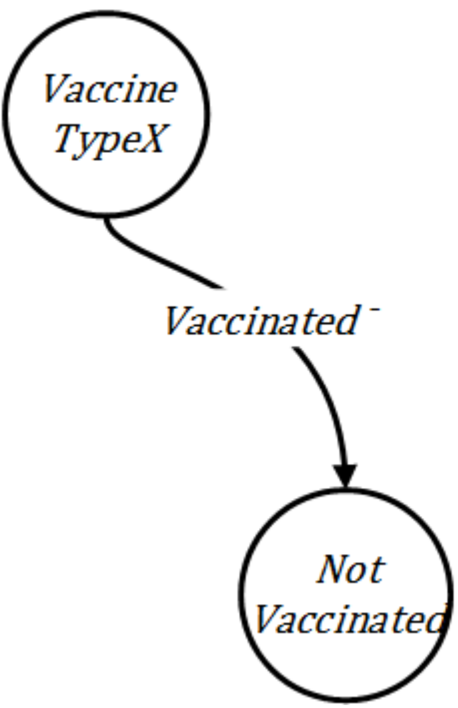
- The inference requires repetitive scans of the axioms
  - The problem becomes I/O bounded.

Our Contribution

- All existing rule engines apply the inference rules sequentially
- They scan the ontology on a per-rule basis
- We define a uniform access pattern which allows for the in-bulk application of many rules within the same scan

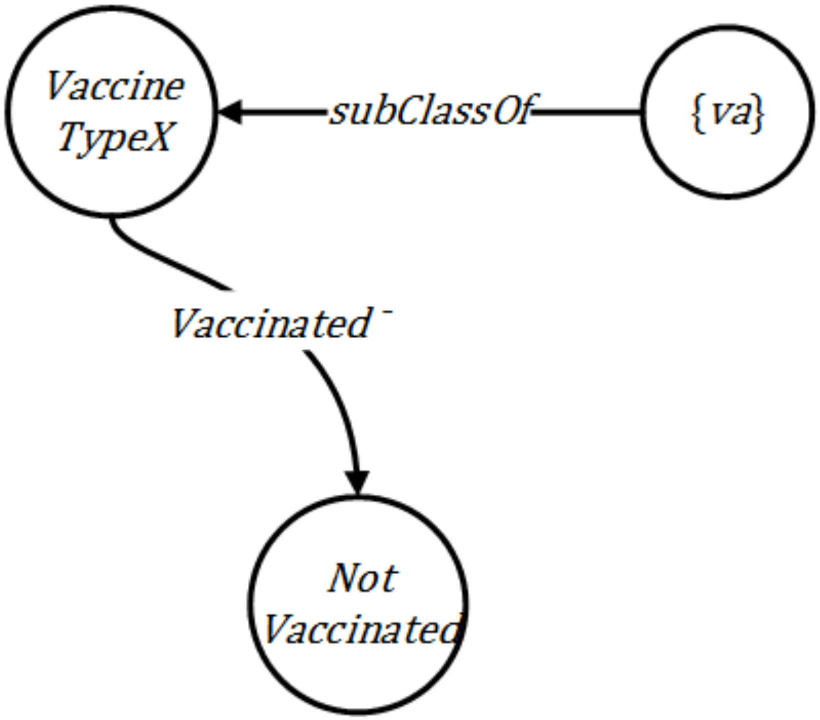
Graph Model

2. $\exists$Vaccinated.VaccineTypeX $\sqsubseteq$ NotVaccinated



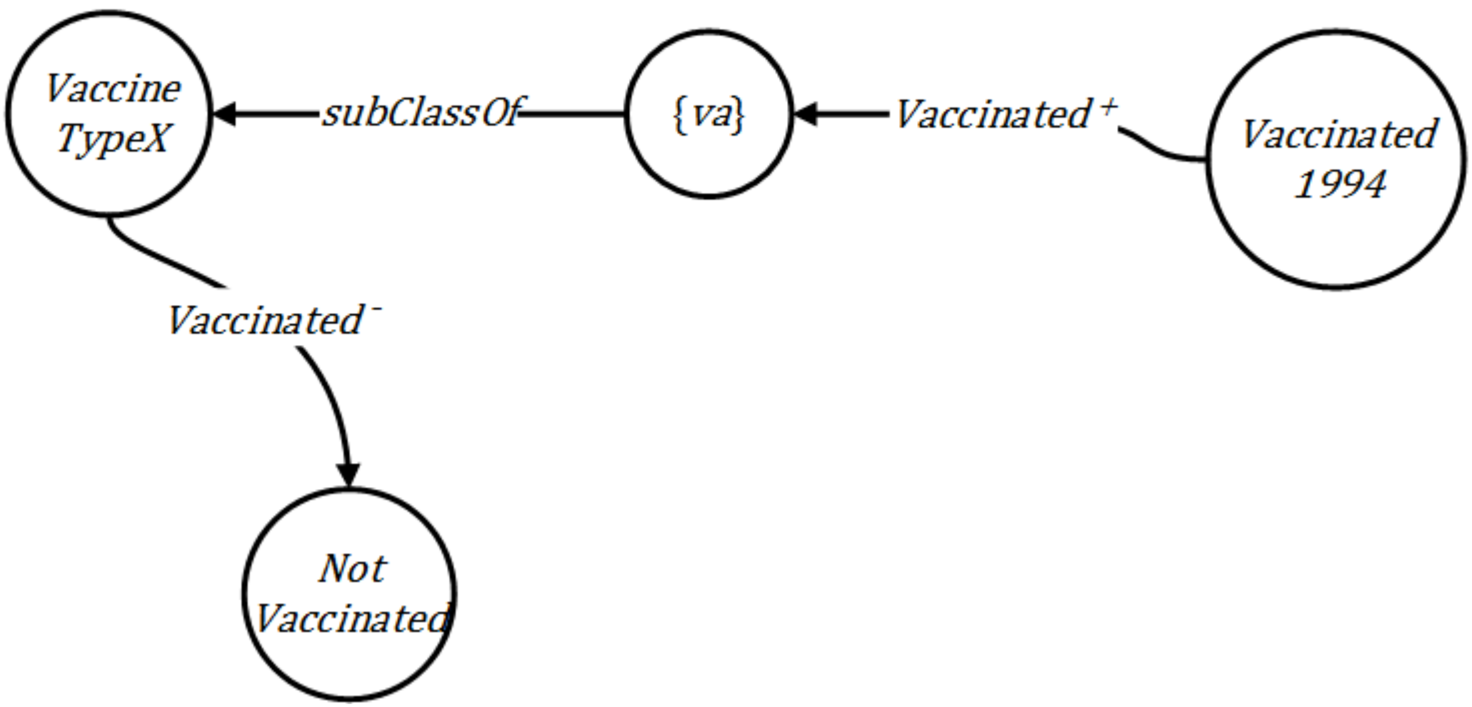
Graph Model

4. {va} ⊑ VaccineTypeX



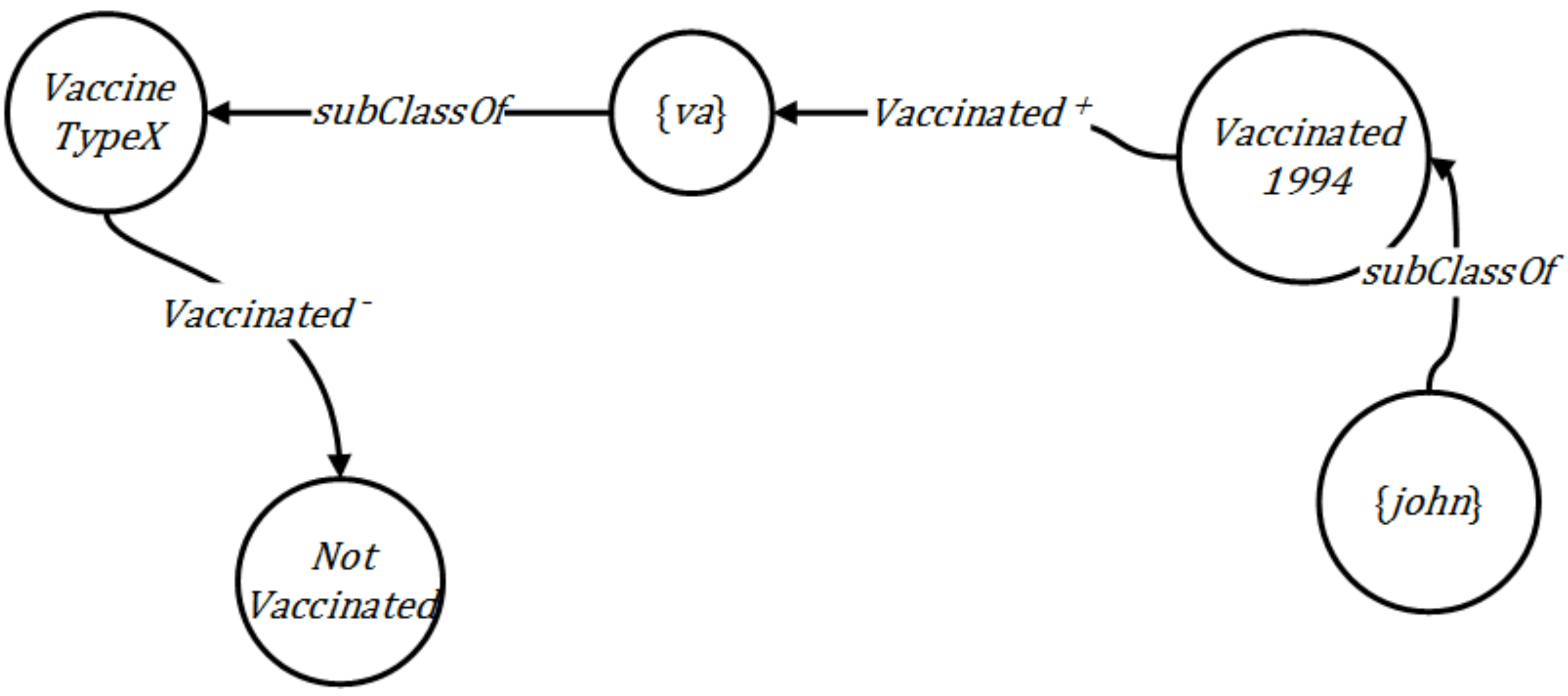
Graph Model

3. Vaccinated1994 ⊑ ∃Vaccinated.{va}



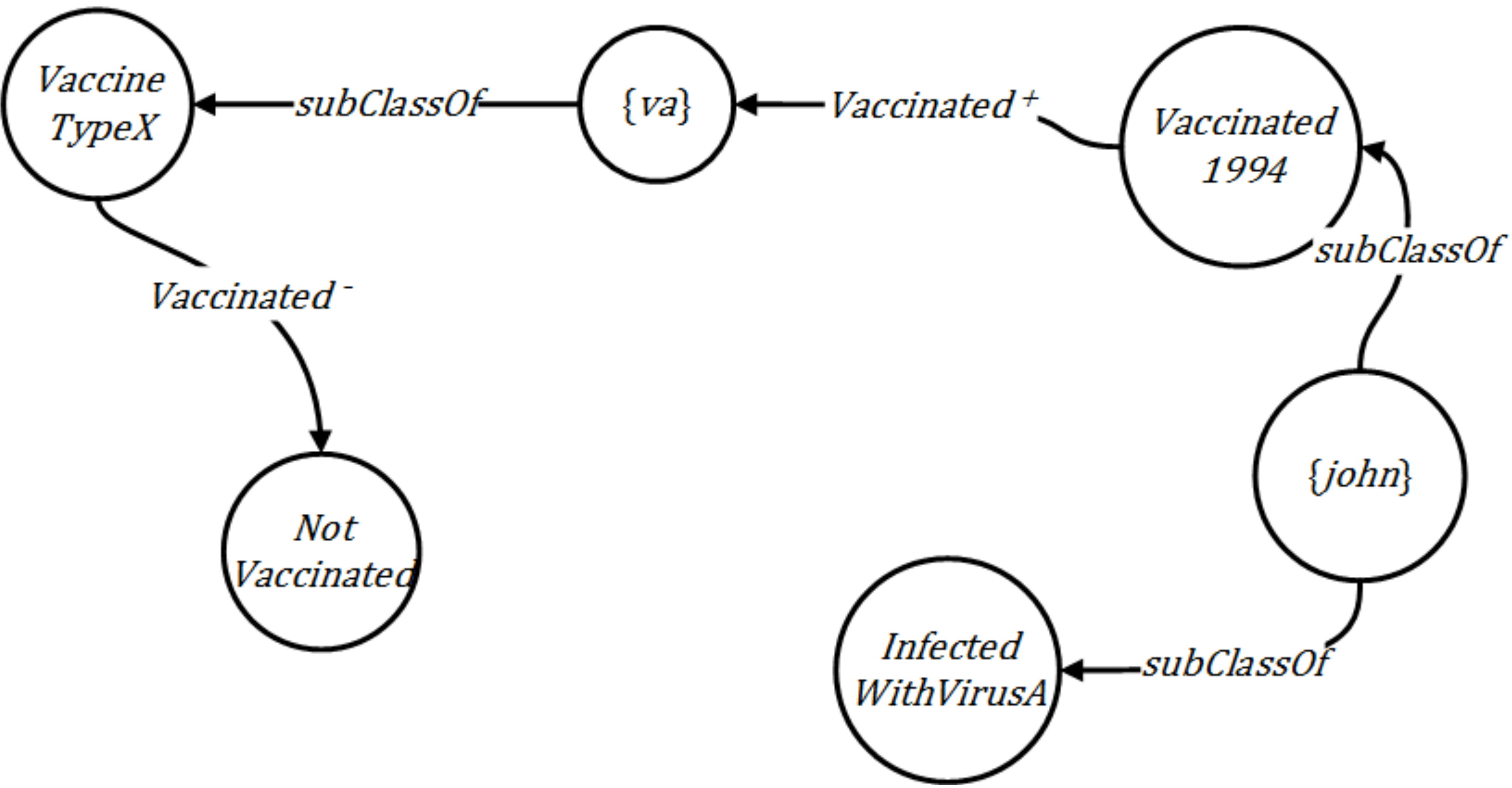
Graph Model

5. {john} ⊑ Vaccinated1994



Graph Model

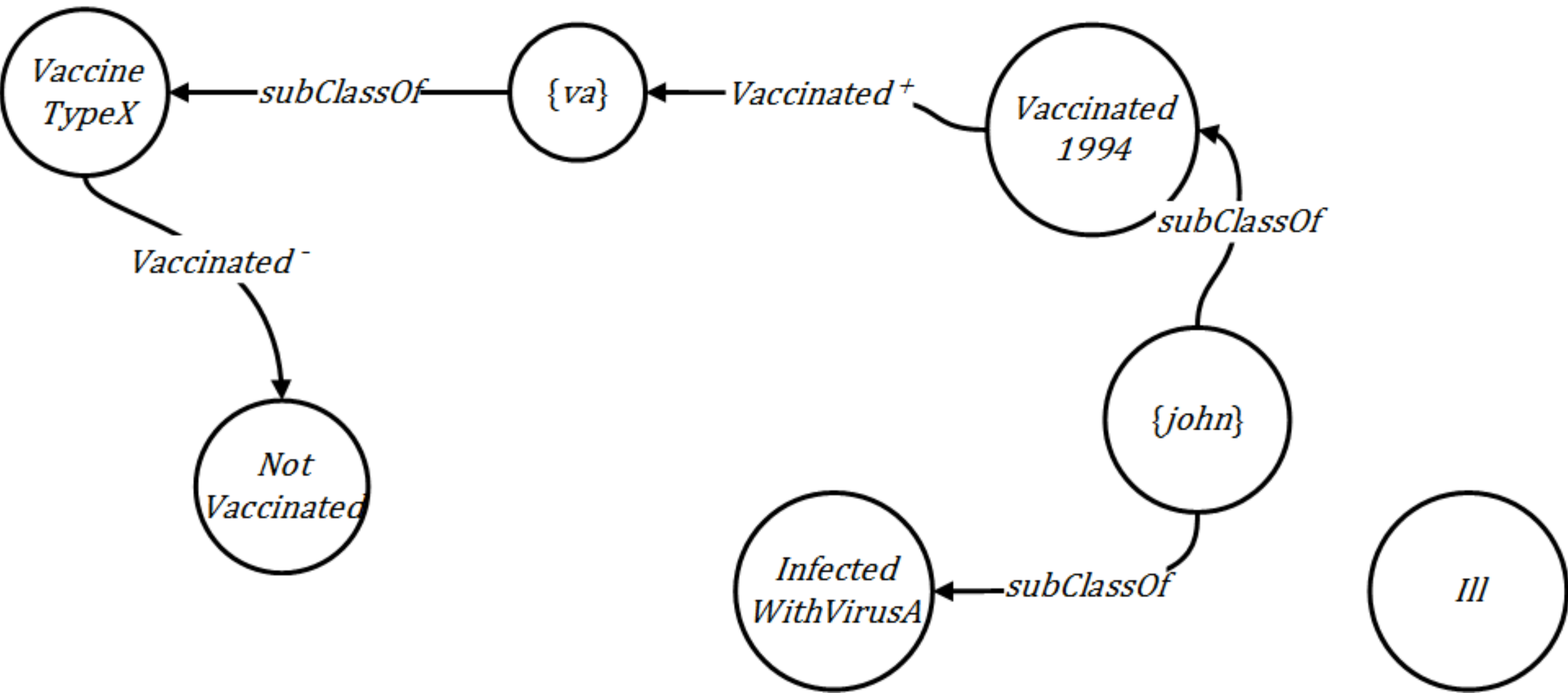
6. {john} ⊑ InfectedWithVirusA



Graph Model

1. $\mathit{InfectedWithVirusA} \sqcap \mathit{NotVaccinated} \sqsubseteq \mathit{Ill}$

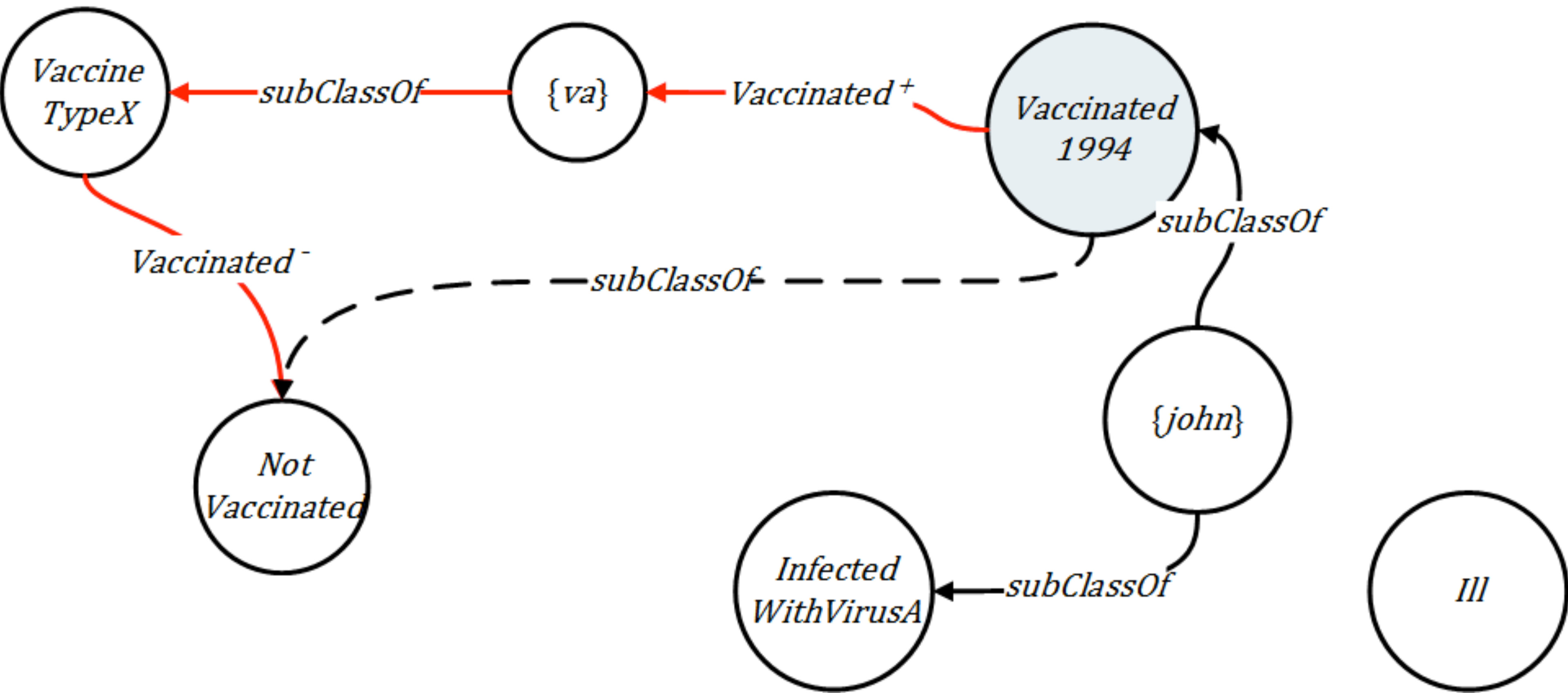
Metadata



Inference on the Graph

1. $InfectedWithVirusA \sqcap NotVaccinated \sqsubseteq Ill$

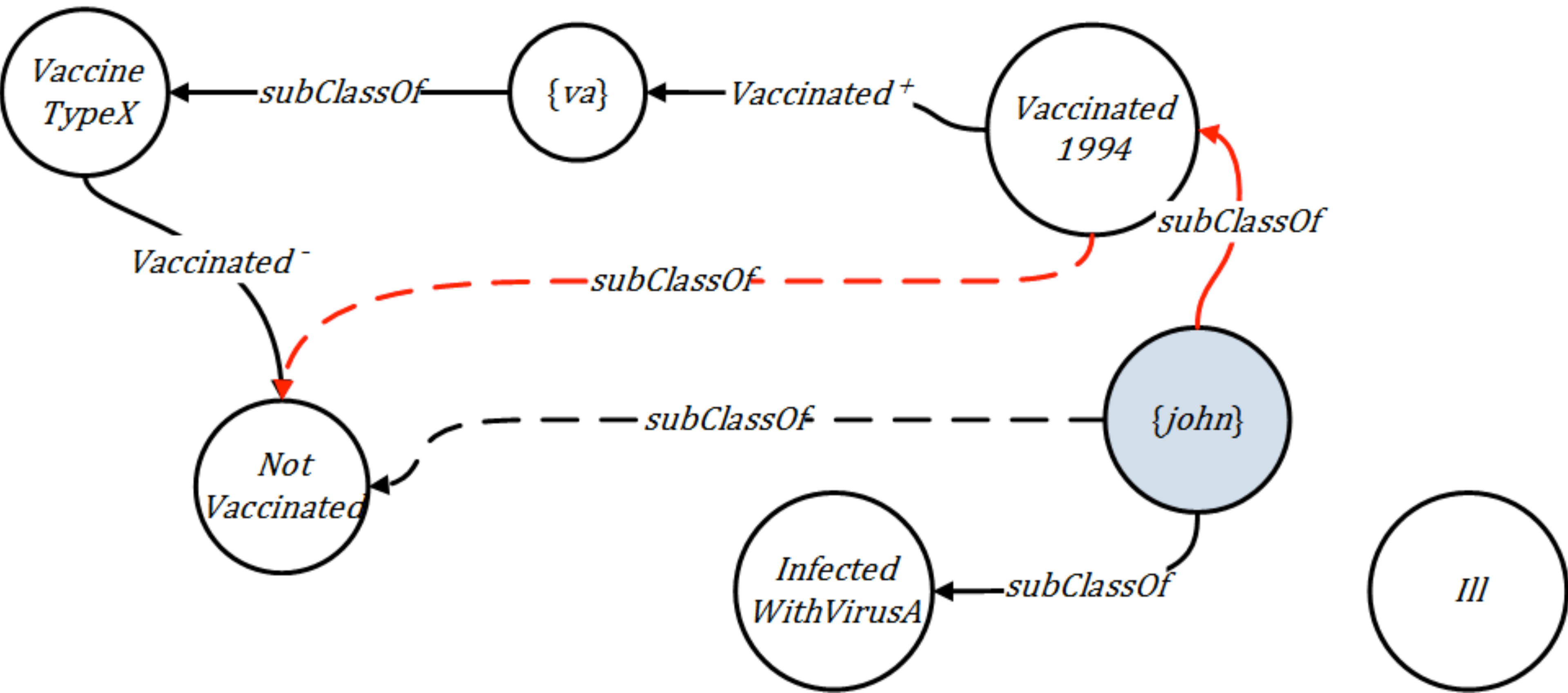
Metadata



Inference on the Graph

1. $InfectedWithVirusA \sqcap NotVaccinated \sqsubseteq Ill$

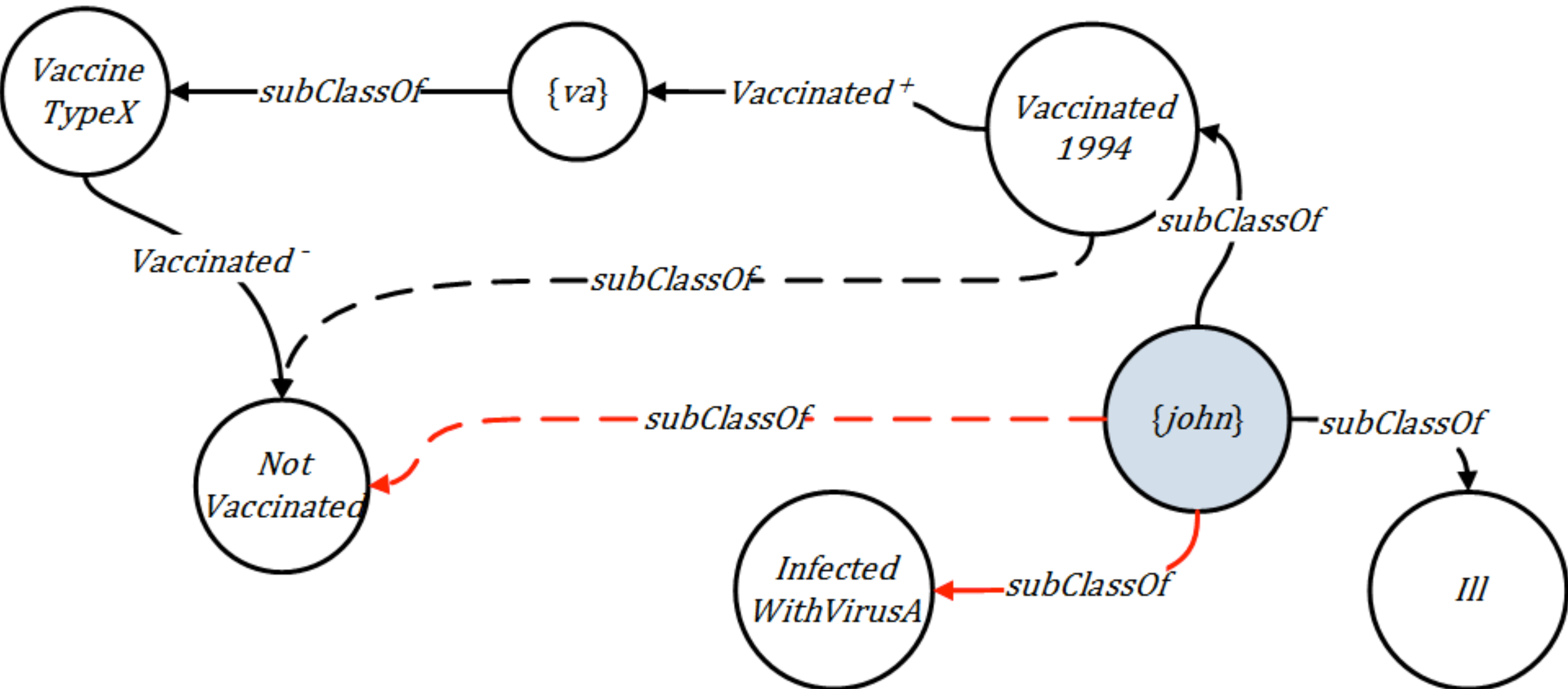
Metadata



Inference on the Graph

1. $InfectedWithVirusA \sqcap NotVaccinated \sqsubseteq Ill$

Metadata



Idea of the Algorithm

- Store the graph in a way that allows efficient lookups in the neighbourhood of each node

- Keep track of the changes made in the graph

- At each subsequent step check only the neighbourhoods affected from the previous step

Experiments

- Real ontologies
  - SNOMED CT
  - GALEN8

- Synthetic ontologies
  - Ontologies of different sizes whose graphs are isomorphic to the graphs of SNOMED CT and GALEN
  - Ontologies of different sizes by increasing the number of labeled edges per node

Experiments on Real Data
(SNOMED CT)

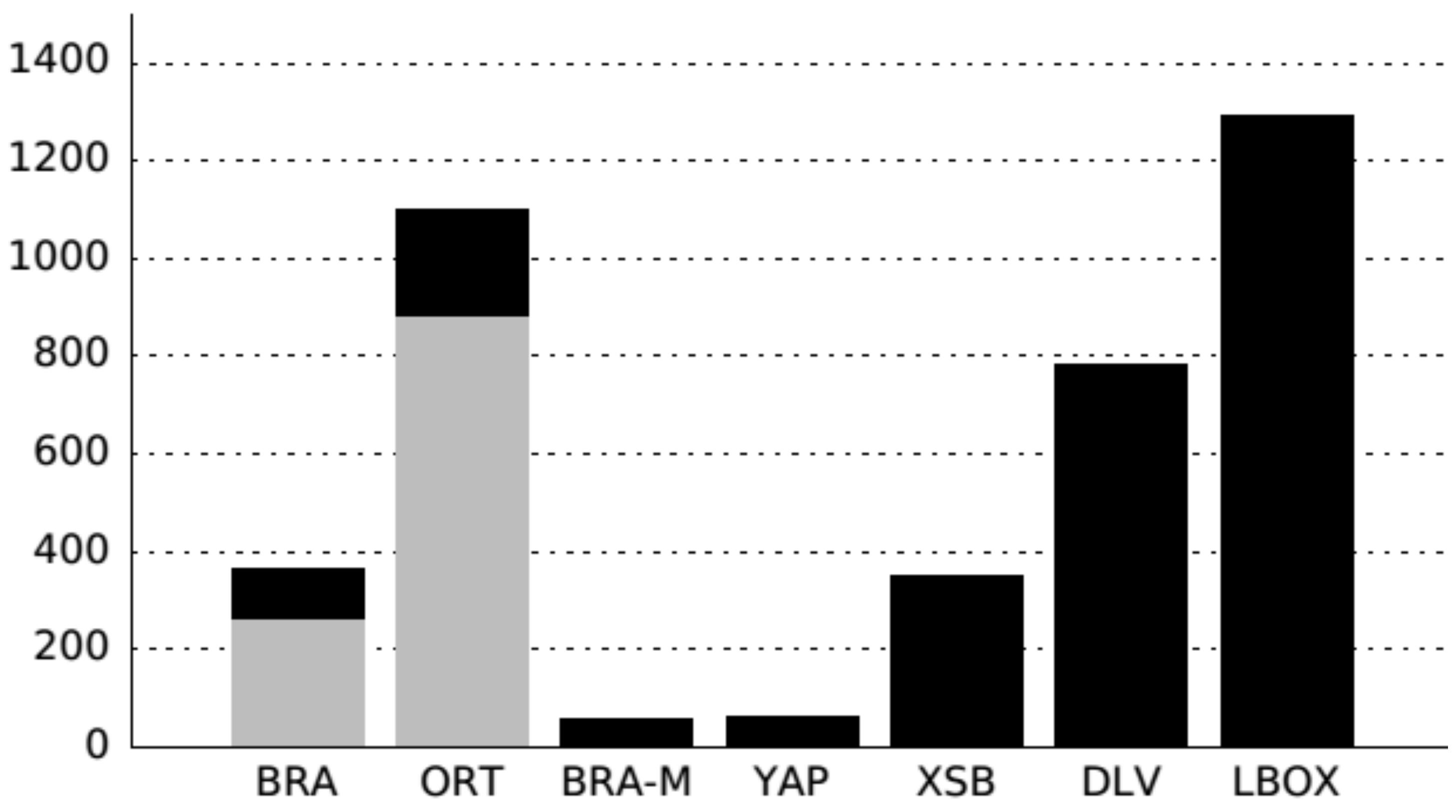
BRA: Batch Rule Application

ORT: One Rule at a Time

BRA-M: Main-memory version of BRA

YAP, XSB: Prolog-based systems

DLV, LogicBlox: Datalog engines



Experiments on Real Data (GALEN8)

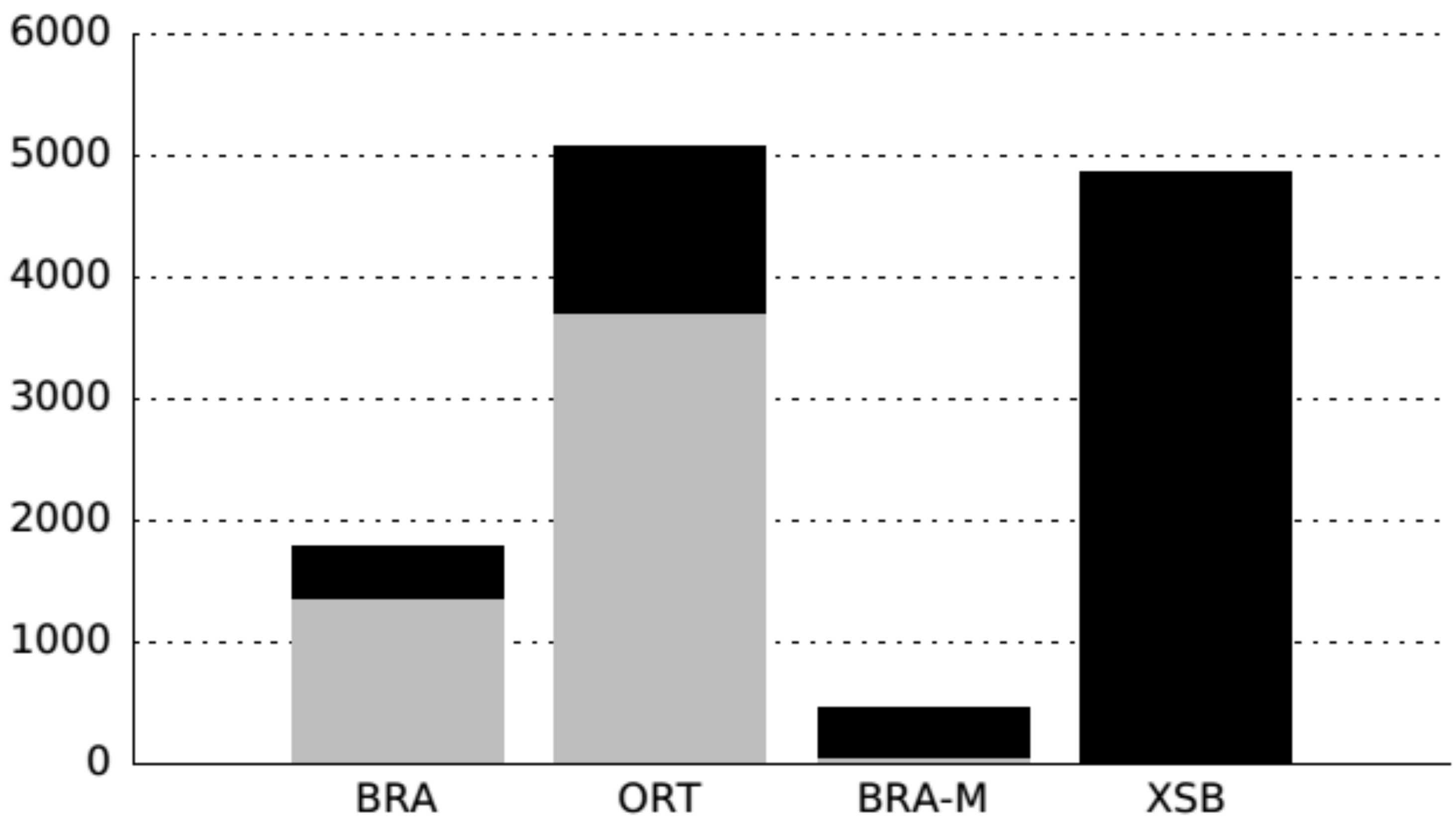
BRA: Batch Rule Application

ORT: One Rule at a Time

BRA-M: Main-memory version of BRA

YAP, XSB: Prolog-based systems

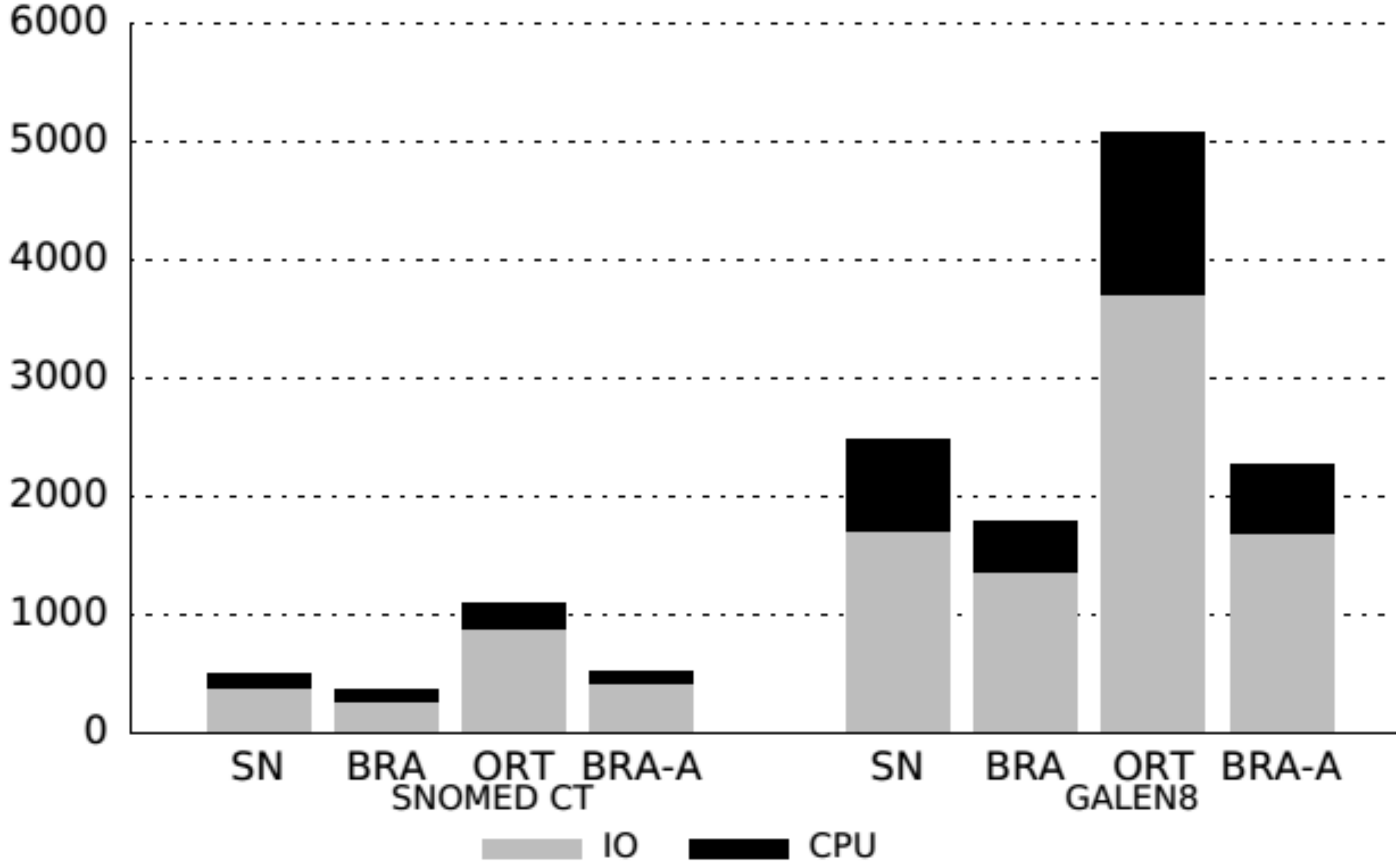
DLV, LogicBlox: Datalog engines



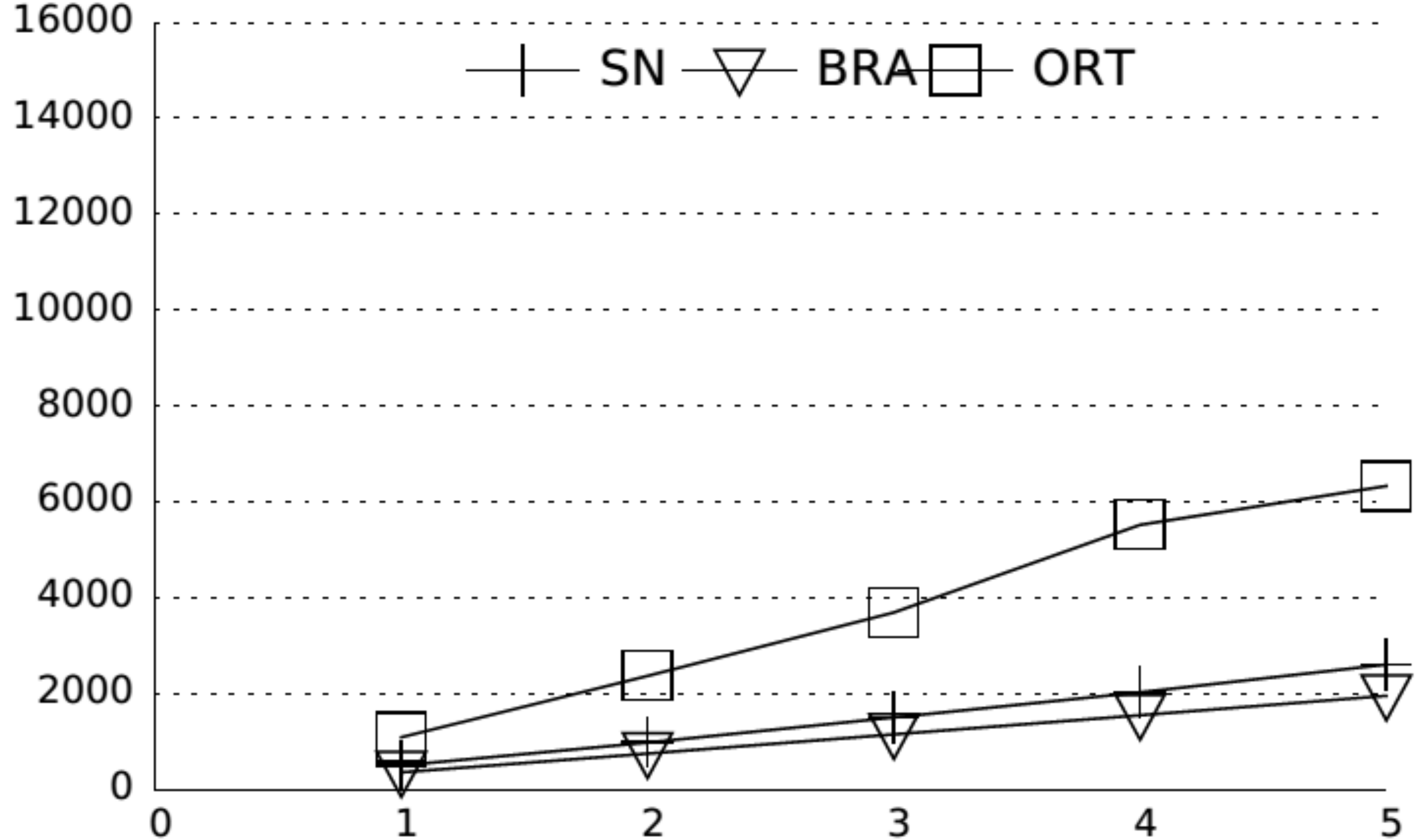
Experiments on Real Data
(Optimizations)

BRA: Batch Rule Application
BRA-A: BRA on the schema of ORT

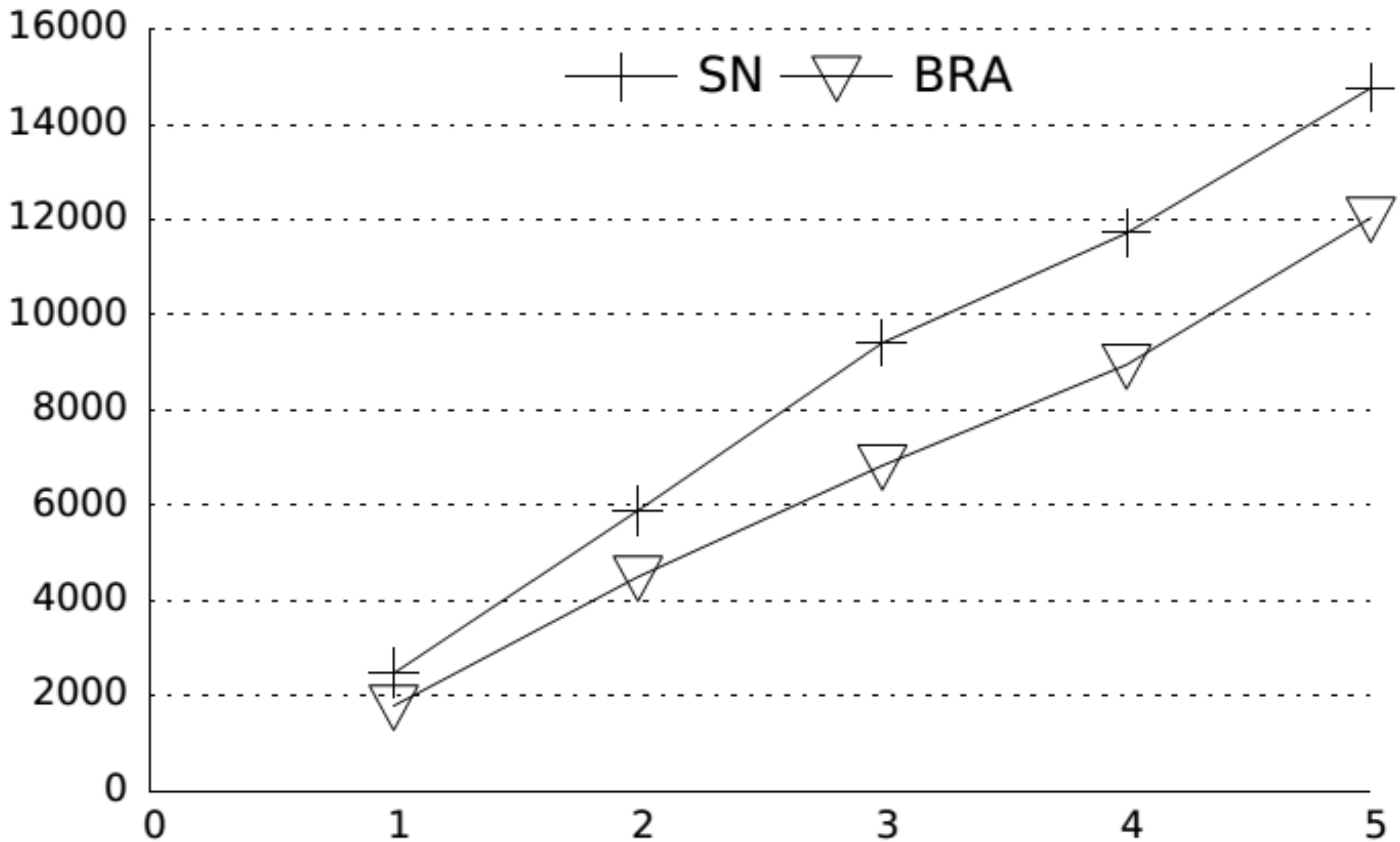
SN: BRA without optimizations
ORT: One Rule at a Time



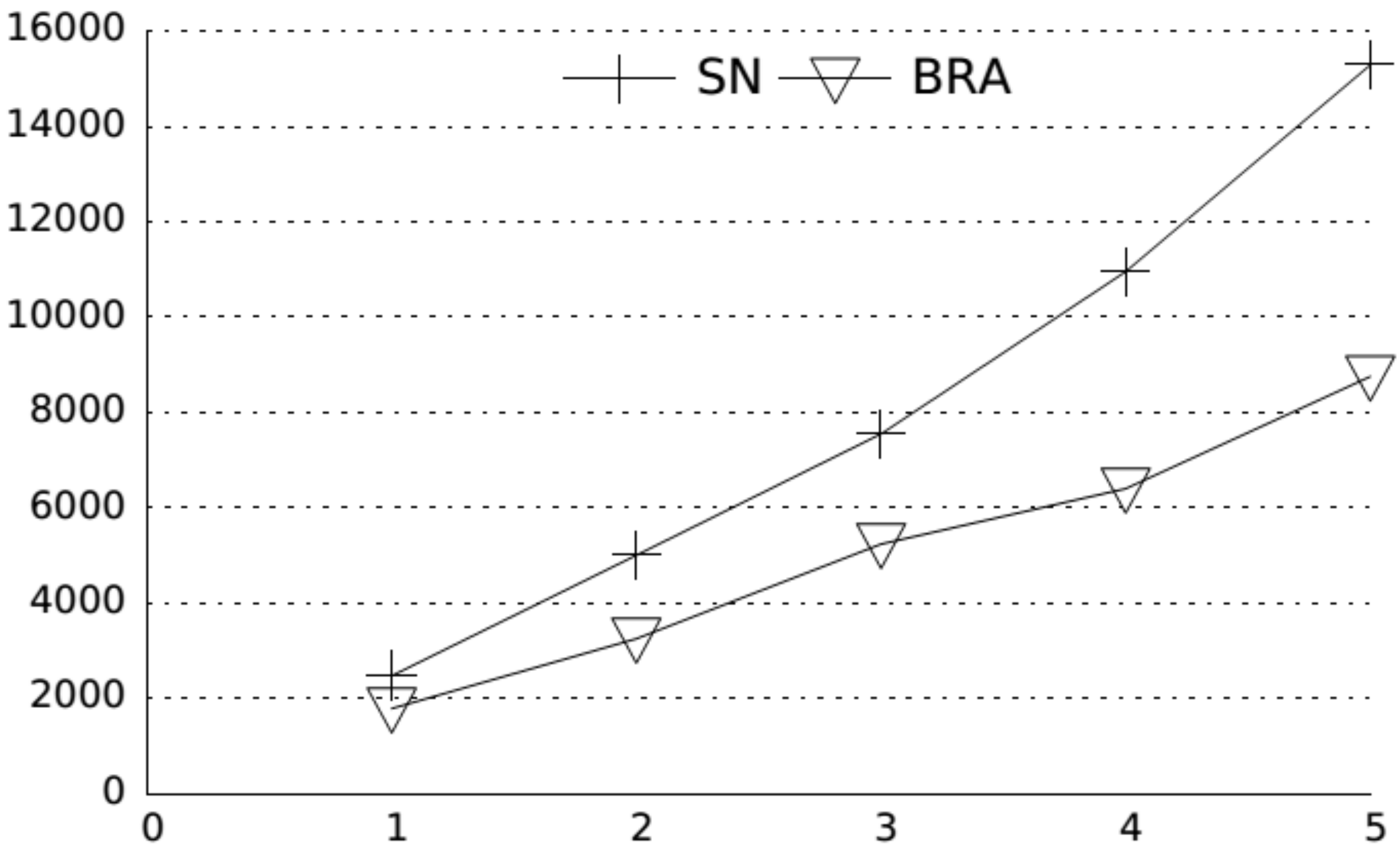
Experiments on Synthetic Data
(Isomorphic Graphs of SNOMED CT)



Experiments on Synthetic Data
(Isomorphic Graphs of GALEN8)



Experiments on Synthetic Data
(Increasing Node Degree for GALEN8)



OWL2-EL Syntax and Semantics

- Intersection of Classes
    - Father $\equiv$ Male $\sqcap$ Parent
    - “Father is the class of all individuals which are of type Male and also of type Parent”

- Existential Restrictions
    - Grandparent $\equiv$ $\exists$hasChild.Parent
    - “Grandparent is the class of all individuals which are linked through property “hasChild” with an individual of type Parent”

OWL2-EL Syntax and Semantics

- Reflexivity
  - $Narcissus \equiv \exists likes.Self$
  - “Narcissus is the class of all individuals which are linked through property ‘likes’ with themselves”

- Property axioms
  - $hasSister \sqsubseteq siblingWith$
  - $hasMother \circ hasSister \sqsubseteq hasAunt$

OWL2-EL Syntax and Semantics

- Singleton Nominals (individuals/instances)
  - $\{mary\} \sqsubseteq Woman \leftrightarrow$ <mary rdfs:type Woman>
  - $\{mary\} \sqsubseteq \exists siblingWith.\{tom\} \leftrightarrow$ <mary siblingWith tom>
  - $\{mary\} \equiv \{maria\} \leftrightarrow$ <mary owl:sameAs maria>
- These are the actual data modeled with the ontology

Why OWL2-EL??

- It is widely used in Life Sciences
- Large OWL2-EL ontologies like SNOMED CT have become vital parts of the Health Information Systems in many countries
- It supports the definition of Tuple-Generating Dependencies (TGDs)
    - Suitable for data integration scenarios