Incremental Reasoning on RDFS
Jules Chevalier, Julien Subercaze, Christophe Gravier, Frederique Laforest

To cite this version:
Jules Chevalier, Julien Subercaze, Christophe Gravier, Frederique Laforest. Incremental Reasoning on RDFS. The Fourteenth International Symposium on Intelligent Data Analysis, Oct 2015, Saint Etienne, France. <hal-01261407>

HAL Id: hal-01261407
https://hal.archives-ouvertes.fr/hal-01261407
Submitted on 25 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Evolving Data

Inference time (in ms.)

10 000
20 000
30 000
RDFS
Inference time (in ms.)
Slider
OWLIM

Parallel and Scalable Execution: Each inference rule is mapped to an independent module, receiving intended triples and later distributing them to other modules for further processing.

Duplicates Limitation: Vertical partitioning [1] and multiple indexing limit the production of duplicates and avoid unnecessary computation.

Data Stream Support: Slider can handle both dynamic triple streams and static triples sets by employing parallel architecture.

Fragment’s Customization: Slider natively support both RDFS [4] and ρdf [5] fragments, and can be extended to any other fragments.

Comparison with OWLIM-SE [2]

Inference on both RDFS and ρdf

13 different ontologies
– 5 generated by BSBM [3]
– 2 from real-word datasets
– 6 subClassOf ontologies

106.86% improvement for ρdf
36.08% improvement for RDFS
71.47% improvement in average

References


Future Work

• Implementation of more complex inference rules, to provide reasoning over more complex fragments.
• Just-in-time optimisations of the rules execution’s scheduling.
• Use of previous runs informations to adapt and be more reactive.

Source Code and Demo

The source code is available here: https://github.com/juleschevalier/slider
A demo can be found here: http://demo-satin.telecom-st-etienne.fr/slider/

Acknowledgement

This work is supported by the OpenCloudware project. OpenCloudware is funded by the French FSN (Fond national pour la Société Numérique), and is supported by Pôles Minalogic, Systematic and SCS.