0.1 Background

The Web is transitioning away from centralised services to a re-emergent decentralised platform. This generates demand for technologies that hide complexities of federated architectures (Verborgh, 2021) so developers can create rich Web 3.0 (Berners-Lee et al., 2001; Berners-Lee, 2001) applications.

In typical centralised databases containing real-world Semantic Web data, implicit facts comprise a sizeable portion of the database. For instance, in the UniProt (Consortium, 2015) knowledge graph of protein sequences, 46.1% of the 228 million facts within the database are implicit. Further, in the Classical Art Research Online Service (CLAROS) (Kurtz et al., 2009) knowledge graph, 81.4% of the 102 million facts are implicit. Considering only ABox reasoning on ontologies such as OpenCyc[1], an ontology of general human knowledge, the effect is even more pronounced; 99.8% of the 1176 million ABox facts in the database are generated by rules (Motik et al.). Therefore, effectively performing reasoning on a decentralised Semantic Web is necessary to producing equally rich results.

Given the importance of implicit data to centralised Semantic Web applications, there is an immense opportunity for decentralised applications to have similar reasoning functionality. To enable this, client-side Resource Description Framework (RDF) reasoning is an emerging area of research (Terdjimi et al., 2015), which has the capacity to unlock a wide range of IoT applications; ranging from user-empowered social media (Werbrouck et al., 2019) to live health diagnoses (Sondes et al., 2019). However, the limited work on client-side RDF reasoning (Terdjimi et al., 2015) has failed to produce a reasoner that meets the *performance*, *interoperability* and *usability* requirements for many of these applications.

0.2 RDF Reasoning

Throughout this project, we are interested in rules-based semantic reasoning on RDF data sets so as to infer logical consequences from an asserted set of facts (axioms).

There are two primary techniques used in literature (Mishra and Kumar, 2011; Rattanasawad et al., 2013) to achieve the process of semantic reasoning, which are known as forward chaining and backward chaining.

0.2.1 Forward Chaining

Forward Chaining (Finin et al., 1989) begins with a data source and recursively applies rules until new results are achieved. This is generally considered an eager method for fact materialisation, as all facts must be realised before the database can be queried.

[1] https://github.com/asanchez75/opencyc

Forward chaining is often called materialisation as systems materialise (pre-compute) facts and place them in memory.

0.2.2 Backward Chaining

Backward chaining (Russell and Norvig, 2009, p. 337) first begins with a set of goals (facts) - which it wishes to test to see if they are true. The inferencing engine then examines the consequents of rules in the rule-set. For any consequents that match the goals we wish to test, the engine then checks to see if the antecedent is supported. This can happen one of two ways:

1. Data supporting the antecedent is present within the dataset.
2. The consequent of another rule supports the antecedent - and the process is repeated (chained) to see if the antecedent of the downstream rule is supported.

0.2.3 Query Rewriting

For certain sets of rules, backwards chaining can be achieved using query rewriting. Query rewriting is the process of modifying a *SPARQL* query such that when executed over only *explicit data* the results are the same as if the original *SPARQL* query had been executed against the *implicit* and *explicit* data.

0.3 Problem Statement

RDF reasoning does not occur widely in browser-based applications, despite being critical to the next generation of Web technologies. The primary constraints are that client-side reasoners do not meet the *performance* requirements of modern applications. Whilst we have work evidencing that it is possible to perform reasoning in the browser.[2][3], they are largely implemented using forward-chaining techniques and thus require most of a dataset to be loaded into the client before reasoning takes place. This is not viable for larger datasets. Thus query-rewriting is needed as an optimisation to minimise the amount of data that must be sent to the client.

The goal of this project is to implement query-rewriting for the Comunica Engine (Taelman et al., 2018).

[2] https://github.com/comunica/comunica-feature-reasoning
[3] https://github.com/rdfjs/N3.js/pull/296

Bibliography

Berners-Lee, T., 2001. Web 3.0. (2001). [Cited on page i.]

Berners-Lee, T.; Hendler, J.; and Lassila, O., 2001. The semantic web. *Scientific american*, 284, 5 (2001), 34–43. [Cited on page i.]

Consortium, U., 2015. Uniprot: a hub for protein information. *Nucleic acids research*, 43, D1 (2015), D204–D212. [Cited on page i.]

Finin, T.; Fritzson, R.; Matuszek, D.; et al., 1989. Adding forward chaining and truth maintenance to prolog. *5th Artificial Intelligence Applications*, (1989), 123–130. [Cited on page i.]

Kurtz, D.; Parker, G.; Shotton, D.; Klyne, G.; Schroff, F.; Zisserman, A.; and Wilks, Y., 2009. Claros-bringing classical art to a global public. in escience, 20–27. *IEEE Computer Society*, (2009). [Cited on page i.]

Mishra, R. B. and Kumar, S., 2011. Semantic web reasoners and languages. *Artificial Intelligence Review*, 35, 4 (2011), 339–368. [Cited on page i.]

Motik, B.; Nenov, Y.; Piro, R.; and Horrocks, I. Handling owl: sameas in rdfox via rewriting. [Cited on page i.]

Rattanasawad, T.; Saikaew, K. R.; Buranarach, M.; and Supnithi, T., 2013. A review and comparison of rule languages and rule-based inference engines for the semantic web. In *2013 International Computer Science and Engineering Conference (ICSEC)*, 1–6. doi:10.1109/ICSEC.2013.6694743. [Cited on page i.]

Russell, S. and Norvig, P., 2009. Artificial intelligence: A modern approach. edition. [Cited on page ii.]

Sondes, T.; Elhadj, H. B.; and Chaari, L., 2019. An ontology-based healthcare monitoring system in the internet of things. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 319–324. IEEE. [Cited on page i.]

Taelman, R.; Van Herwegen, J.; Vander Sande, M.; and Verborgh, R., 2018. Comunica: a modular sparql query engine for the web. In *Proceedings of the 17th International Semantic Web Conference*. https://comunica.github.io/Article-ISWC2018-Resource/. [Cited on page ii.]

Terdjimi, M.; Médini, L.; and Mrissa, M., 2015. Hylar: Hybrid location-agnostic reasoning. In *ESWC Developers Workshop 2015*, 1. [Cited on page i.]

Verborgh, R., 2021. Reflections of knowledge. https://ruben.verborgh.org/blog/2021/12/23/reflections-of-knowledge/. [Cited on page i.]

Werbrouck, J.; Pauwels, P.; Beetz, J.; and van Berlo, L., 2019. Towards a decentralised common data environment using linked building data and the solid ecosystem. In *36th CIB W78 2019 Conference*, 113–123. [Cited on page i.]