Pellet reasoner using SWRL
Pellet Overview

Pellet is an open-source Java based OWL DL reasoner and it can be used with Jena and OWL API libraries.

Features:

- **Consistency checking:** ensures that an ontology does not contain any contradictory facts.
- **Concept satisfiability:** checks if it is possible for a class to have any instances.
- **Classification:** computes the subclass relations between every named class to create the complete class hierarchy.
- **Realization:** computes the direct types for each of the individuals.
Architecture of Pellet

Terms:

- **Assertional Box** (Abox) contains assertions about individuals, i.e. OWL facts such as type, property-value...
- **Terminological Box** (Tbox) contains axioms about classes, i.e. OWL axioms such as subclass, equivalent class...
- **Knowledge Base** (KB) A combination of an ABox and a TBox, i.e. a complete OWL ontology.
Architecture of Pellet

- **Main modules:**

  - **Parsing and Loading:** Pellet includes different submodules that can load ontologies from different representations (Jena, WonderWeb..).

  - **Tableaux Reasoner:** has the functionality of checking the consistency of an ontology by constructing a graph from the Abox. The reasoner repeatedly applies the tableaux expansion rules until a contradiction is detected.

  - **Datatype Reasoner:** is responsible for checking if the intersection of datatypes is consistent or not.

  - **Knowledge Base Interface:** makes decisions to check the consistency of the ABox, when to classify all the concepts or when to realize all the individuals and answer queries.

  - **ABox Query Engine:** The KnowledgeBase interface is coupled with an ABox Query Engine that answers queries.
Architecture of Pellet

Operation:

- The core of the system is the tableaux reasoner that checks the consistency of a knowledge base.

- The OWL ontologies are loaded into the reasoner after ontology validation and repair. This step ensures that all the resources have an appropriate type triple and missing type declarations are added.

- During the loading phase, axioms about classes are put into the TBox component and assertions about individuals are stored in the ABox component.

- The system provides a thin layer for programmatic access through the Service Programming Interface (SPI) that provides convenience functions to access the reasoning services provided.
Reasoners detect inconsistent ontologies but the diagnosis and resolution of the bug is not supported at all.

Pellet contains two debugging services that help explain why the inconsistency occurs:

- the first service clash detection is used to pinpoint the root contradiction or clash in the completion graph;
- and the second – axiom tracing is used to extract the relevant source axioms from the ontology responsible for the clash.

These services are used in the OWL Ontology Editor, Swoop, as a debugging aid for ontology users and modelers: [http://code.google.com/p/swoop/](http://code.google.com/p/swoop/)
Pizza Example

- Reasoning -> Check Consistency

  - Protégé-OWL will have performed a consistency check of the pizza ontology:
Pizza Example

- CheeseyVegetableTopping
Pizza Example

- Ice Cream
THE END

THANK YOU!