Pellet: An OWL DL Reasoner

Evren Sirin and Bijan Parsia
MINDSWAP Research Group
University of Maryland, College Park, MD
evren@cs.umd.edu, bparsia@isr.umd.edu

1 Introduction

In order to gain experience with description logic reasoner, and to contribute to the OWL Candidate Recommendation process, a small team at MINDSWAP set out to implement a tableau reasoner for the Lite and DL dialects of OWL (corresponding roughly to the description logics SHIF(D) and SHION(D)). Our group found existing, available DL reasoners lacking for our purposes, because we needed an open-source tool that provides ABox reasoning, that does not make Unique Name assumption, supports entailment checks and works with XML Schema datatypes. Pellet has been developed to addresses these issues and has become both our test bed for experiments with DL and Semantic Web reasoning, as well as our standard reasoning component. While not (yet) in the performance range of Racer or Fact, it has many usability features that makes it a good choice for various lighter weight situations.

Technically, Pellet is a sound and complete tableau reasoners for SHIN(D) and SHON(D) (with ABoxes), and a sound but incomplete tableau reasoner for SHION(D) (with ABoxes). Pellet has the usual suite of optimizations including lazy unfolding, absorption, dependency directed backjumping, and semantic branching. It incorporates datatype reasoning for the built-in primitive XML Schema datatypes. Pellet is implemented in pure Java and available as open source software.

2 Special features

Pellet has a number of features either driven by OWL requirements or Semantic Web issues.

Ontology analysis and repair OWL has two major dialects, OWL DL and OWL Full, with OWL DL being a subset of OWL Full. All OWL knowledge bases are encoded as RDF/XML graphs. OWL DL imposes a number of restrictions on RDF graphs, some of which are substantial (e.g., that the set of class names and individual names be disjoint) and some less so (that every item have a "type" triple). Ensuring that an RDF/XML document meets all the restrictions is a relatively difficult task for authors, and many existing OWL documents are nominally OWL Full, even though their authors intend for them to be OWL DL. Pellet incorporates a number of heuristics to detect "DLizable" OWL Full documents "repair" them.

Datatype reasoning XML Schema has a rich set of basic datatypes including various numeric types (integers and floats), strings, and date/type types. It also has several mechanism, both standard and unusual for creating new types out of the base types. For example, it's possible to define a datatype by restricting the integers to the set of integers whose canonical representation has only 10 digits, or whose string representation matches a certain regular expression. Currently, XML Schema systems tend toward validation of documents and generation of PSVI instead of type checking (though, with the advent of XQuery, this might change). Pellet can test the satisfiability of conjunctions of thus constructed datatypes.

Entailment In Semantic Web, entailment is the key inference whereas the Description Logic community have focused on satisfiability and subsumption. While entailment can be reduced to satisfiability, most DL systems do not support it. In part to pass a large portion of the OWL test suite, we implemented entailment support in Pellet.

Conjunctive ABox query Query answering is yet another important feature for Semantic Web. We have implemented an ABox query answering module in Pellet using "rolling-up" technique. We have devised algorithms to optimize the query answering by changing how likely candidates for variables are found and tried. Exploiting the dependencies between different variable bindings helps us to reduce the total number of satisfiability tests thus speeding up the answer significantly.

3 Applications and Future Work

We expose most of Pellet's capabilities equally from a Java API, a command line interface, and a Web form. The Web form has been used by a number of people for species validation, consistency checking, and experimenting with OWL DL classification and entailment.

Pellet is the default reasoner in SwoopEd, a lightweight ontology browser and editor. Pellet is used for classification, class satisfiability testing, query, and species validation and repair.

We also use Pellet for web service discovery and composition. Pellet has been incorporated as the knowledge base for a version of the SHOP2 HTN planning system and Fujitsu Lab of America's Task Computing Environment (TCE).

We have begun experimenting with various multi-ontology/logic formalism, such as distributed description logic and E-Connections implementation techniques. Initial results have been both instructive and promising. We also aim to integrate the reasoner with rules to support Semantic Web Rule Language (SWRL). Other future work projects include generating explanations for concept satisfiability, querying using **K** operator and experimenting with non-monotonic reasoning using annotated logics.