

Beschreibungslogik

Übungsblatt 6

Abgabe bis 3.7.2017, 23:59 Uhr in Stud.IP, Ordner „Abgabe Übungsblatt 6“, im ZIP-Format. Bitte nur eine Abgabe pro Gruppe, Lizenz „Selbst verfasstes, nicht publiziertes Werk“.

1. (100 %) Implementiert und testet einen Reasoner für Klassifikation von \mathcal{EL} -TBoxen.¹ Geht dabei wie folgt vor.

- Implementiert zuerst Normalisierung (Regeln **NF1–NF5**).
- Implementiert dann das konsequenzbasierte Verfahren (Regeln **R1–R4**).
- Testet Euren Reasoner an Ontologien aus dem BioPortal-Korpus. Dazu findet Ihr auf <https://zenodo.org/record/439510> einen aktuellen Schnappschuss. Beachtet auch die Hinweise, die auf dieser Seite gegeben werden. Relevant sind folgende Dateien:
 - `owlxml.zip`: Dieses Archiv enthält alle verfügbaren Ontologien im OWL-XML-Format.
 - `bioportal2017.03.30.csv`: Diese CSV-Datei enthält Metadaten der Ontologien im Archiv; interessant ist hier vor allem die Spalte „owl2_el“, die angibt, ob die jeweilige Ontologie im EL-Profil von OWL 2 ist.

Wählt aus den Ontologien, die in dieser Spalte eine 1 haben, eine oder mehrere zum Testen aus. Die sollten nicht zu groß sein (höchstens 1000 Axiome, siehe Spalte „logical_axiom_count“). Vergleicht die Resultate Eures Reasoners mit denen verfügbarer Reasoner – entweder in Protégé oder über die OWL API. Beachtet dabei, dass das EL-Profil von OWL 2 mehr umfasst als \mathcal{EL} . Euer Reasoner darf diese zusätzlichen Operatoren ignorieren und ggf. Ergebnisse liefern, die von denen anderer Reasoner abweichen.

Um Euch das Implementieren und Testen zu erleichtern, befindet sich im Stud.IP-Ordner „Materialien zur Übung“ eine Vorlage (Boilerplate), die die wichtigsten Informationen für ein Java-Projekt enthält:

- Ordner `lib`: Bibliotheken für die aktuelle OWL API und den Reasoner `Hermit` (weitere könnt Ihr Euch bei Bedarf über Maven besorgen)
- Ordner `src`: Paket `de.unibremen.informatik.tdki.bl2017` mit den Dateien
 - `ELNormalizer.java` für Eure Normalisierung
 - `ELReasoner.java` für Euren Reasoner
 - `TestNormalizer.java` und `TestReasoner.java` für Tests

Gebt ein Archiv ab, das Eure Quellen, den kompilierten Code, die verwendeten Ontologien und eine kurze Dokumentation Eurer Tests enthält.

¹ In der ersten Version dieses Übungsblatts stand hier: „Subsumtion mit TBoxen in \mathcal{EL} “. Das war irreführend, denn das konsequenzbasierte Verfahren kann keine einzelnen Subsumtionen entscheiden, ohne die ganze TBox zu klassifizieren.