



Künstliche Intelligenz

– Aufgabenblatt 07 –

Prof. Dr. David Spieler
Hochschule München

9. März 2023

In diesem Aufgabenblatt werden wir einige semantische Netze mit RDF bauen und einige Abfragen auf dem Graphen mit SPARQL durchführen. Dafür verwenden wir die Python Bibliothek `rdflib`. Sie können die Bibliothek innerhalb von Jupyter mit Hilfe von `!pip install rdflib` installieren.

Aufgabe 1 (Die drei Freunde aus der Vorlesung) *Installieren Sie zunächst `rdflib` und erstellen Sie ein neues Jupyter Notebook.*

1. Importieren Sie die benötigten Teile und Namensräume von `rdflib` mit Hilfe von

```
from rdflib import URIRef, Literal, Namespace, Graph
from rdflib.namespace import FOAF, XSD, RDF, SDO
```

2. Erstellen Sie einen eigenen **Namensraum** `https://cs.hm.edu/module-ki/example/` mit Hilfe von `Namespace(...)` in der Variablen `KI`.
3. Erzeugen Sie die **Knoten** `P1001`, `P1002` und `P1003` in diesem Namensraum mit Hilfe von `variable = KI['name']` mit den Variablennamen `alice`, `bob` und `charly`.
4. Erzeugen Sie einen neuen **RDF-Graphen** `g` mit Hilfe von `Graph()`.
5. Fügen Sie dem Graph `g` mit Hilfe von `g.add(subject, predicate, object)` die **Tripel** wie in den Vorlesungsfolien hinzu. Benutzen Sie dafür die Variablen `alice`, `bob` und `charly` als auch `RDF.type`, `FOAF.Person`, `FOAF.knows`, `SDO.name`, `SDO.birthDate`, `Literal('<name>')` und `Literal('<date>', datatype=XSD.date)`.
6. **Serialisieren** Sie den Graphen mit `print(g.serialize(format='nt'))` im **N-Triples-Format** und vergleichen Sie das Ergebnis mit den Vorlesungsfolien. Korrigieren Sie evtl. vorhandene Fehler.

7. Teilen Sie dem RDF-Graphen *g* mit `g.bind('ki', KI)`, `g.bind('foaf', FOAF)` und `g.bind('schema', SDO)` besser zu lesende Abkürzungen (*Prefixe*) für die Namensräume mit.
8. Serialisieren Sie den Graphen im *Turtle-Format* mit `print(g.serialize())` und vergleichen Sie das Ergebnis mit den Vorlesungsfolien.
9. Erstellen Sie in der Variablen `knows_query` die *SPARQL-Abfrage* bzgl. *Wer-kennt-wen*. Verwenden Sie dafür mehrzeilige Zeichenketten in Python und starten die Abfrage durch `gres = g.query(knows_query)`. Geben Sie die Ergebnisse mit Hilfe von

```
for row in gres:
    print(f"{row.aname} knows {row.bname}")
```

aus und überprüfen Sie das Ergebnis mit Hilfe der Vorlesungsfolien.

10. Führen Sie eine Abfrage *Wer-kennt-wen-über-wen* aus und überprüfen Sie das Ergebnis.
11. Suchen Sie mit Hilfe einer Abfrage *mögliche Zwillinge* und überprüfen Sie das Ergebnis.