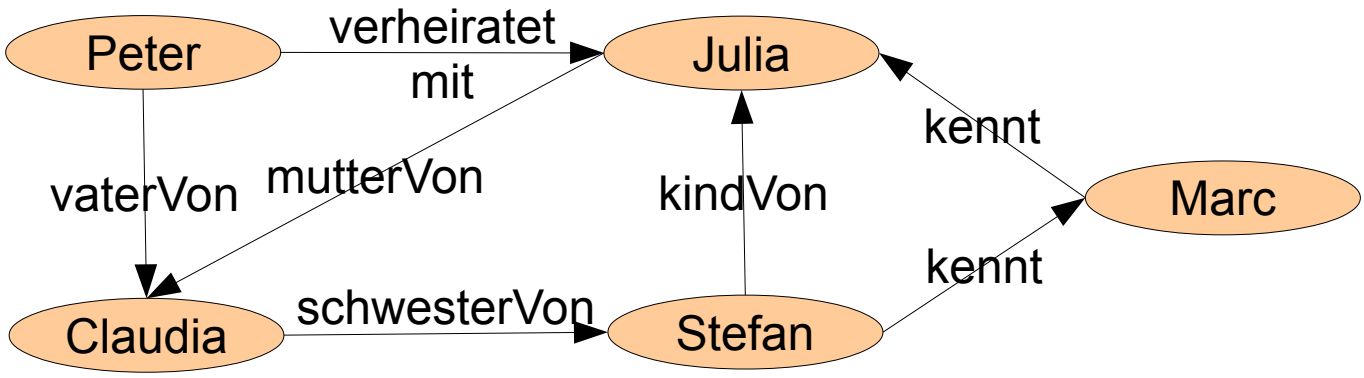
TECHNISCHE UNIVERSITÄT DARMSTADT

Semantic Web Übung 3

Besprechung dieser Übung am Donnerstag, 6.12.

1. Aufgabe

Gegeben folgender RDF-Graph:



Definieren Sie folgende Abfragen in SPARQL:

1. Wen kennt Stefan?
2. Wer ist Kind von Julia?
3. Hat Stefan Geschwister?
4. Ist Julia verheiratet?
5. Hat Stefan Geschwister?

Geben Sie auch an, wie die Antworten auf die Abfragen zu interpretieren sind.

Gegeben sind jetzt zusätzlich folgende Axiome in der T-Box:

```
:kennt a owl:SymmetricProperty .
:verheiratetMit rdfs:subPropertyOf :kennt .
:verheiratetMit a owl:SymmetricProperty .
:vaterVon rdfs:subPropertyOf :elternteilVon .
:mutterVon rdfs:subPropertyOf :elternteilVon .
:kindVon owl:inversePropertyOf :elternteilVon .
:kindVon rdfs:subPropertyOf :kennt .
:schwesterVon rdfs:subPropertyOf :geschwisterVon .
:geschwisterVon a owl:symmetricProperty .
:geschwesterVon rdfs:subPropertyOf :kennt .
```

Wenn Sie diese Axiome hinzuziehen, wie ändern sich dann die Ergebnisse Ihrer Abfrage?

Wäre es nützlich, das folgende Axiom noch hinzuzufügen?

```
:elternteilVon a rdfs:subPropertyOf :kennt .
```

2. Aufgabe

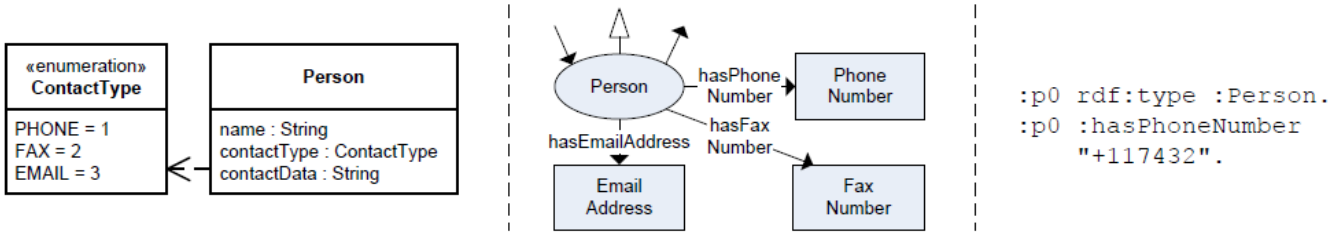
Das Datenset Linked Open Numbers (siehe Foliensatz „Linked Open Data“, Folie 40) definiert Objekte der Klasse Number, um Aussagen zu Zahlen und ihren Beziehungen zu definieren. Zwischen Objekten der Klasse *Number* sind dort unter anderem die Relationen *previous*, *next, lessThan, greaterThan* und *primefactor* definiert.

Formulieren Sie unter Verwendung (nur) dieser Relationen folgende Abfragen in SPARQL:

1. Finde alle geraden Zahlen.
2. Finde alle Zahlen, die direkter Nachfolger eines ihrer Primfaktoren sind.
3. Finde alle ungeraden Zahlen.
4. Finde alle Primzahlen.
5. Finde alle Nicht-Primzahlen.
6. Finde alle Primzahl-Zwillinge (d.h. zwei Primzahlen, bei denen die eine um zwei größer ist als die andere).

3. Aufgabe

Gegeben sind ein Klassenmodell (links) und eine Ontologie (Mitte). Beide werden zur Speicherung von Personen genutzt. Im Klassenmodell wird die Kontaktinformation (z.B. Mailadresse, Telefon- oder Faxnummer) im Attribut contactData gespeichert, während das Flag contactType angibt, um welche Art Kontaktdaten es sich handelt. In der Ontologie werden hierfür drei separate Literale genutzt. Eine Beispiel-Serialisierung ist rechts angegeben.



Erklären Sie, welche Probleme auftreten, wenn Sie in diesem Fall ein direktes Programmiermodell anwenden wollen.

4. Aufgabe

Sie sollen eine Ontologie für ein Autohaus entwickeln. Folgende Informationen sind gegeben:
Autos werden von verschiedenen Firmen hergestellt. Jedes Auto hat bestimmte Kennzahlen (Beschleunigung, Geschwindigkeit, Verbrauch). Autos können darüber hinaus Zubehör haben, z.B. Anhängekupplung, Kindersitze oder Dachgepäckträger. Familienautos haben immer einen Kindersitz als Zubehör inklusive, und alle Kombis verfügen mindestens über eine Anhängekupplung oder einen Dachgepäckträger.

Versuchen Sie, diese Informationen möglichst exakt in OWL zu modellieren. Welche Komplexität hat die entstehende Ontologie?

5. Aufgabe

Gegeben drei Häuser, ein braunes, ein gelbes, und ein blaues, die nebeneinander stehen und von 1-3 nummeriert sind. In jedem der Häuser wohnt eine Person. Jede der Personen hat ein anderes Hobby (Fußball, Rugby oder Eishockey) und ein anderes Lieblingsessen (Pfannkuchen, Waffeln oder Eis). Gegeben sind darüber hinaus noch folgende Fakten:

1. Die Person in Haus 2 isst am liebsten Pfannkuchen.
2. Haus 3 ist nicht braun.
3. Die Person in Haus 3 spielt nicht Rugby.
4. Die Person, die am liebsten Waffeln isst, wohnt im gelben Haus.
5. Haus 3 ist nicht gelb.
6. Die Person in Haus 2 spielt Fußball.

Versuchen Sie, einen Reasoner herausfinden zu lassen, wer in welchem Haus wohnt.