GSBL: GSB-Language

Inhalt

Einleitung

Die GSB-Language (kurz: GSBL) ist eine zweidimensionale Sprache, welche im GSB benutzt wird. Die Wörter der GSBL können mit Hilfe eines Übersetzers in SPARQL umgeformt werden und so Anfragen an einem SPARQL-Endpoint gestellt werden.
Neben der zweidimensionalen Repräsentation im Browser des Nutzers können wir uns vorstellen, dass zukünftig ein GSBL-Wort auch durch ein JSON- oder XML-Objekt dargestellt werden kann.
Im Anschluss an die Einleitung findet man die allgemeine Definition der GSBL: Der generelle Aufbau eines Wortes und dessen Teilwörter, eine kurze Einführung in die Übersetzung sowie präzise Definitionen mit Übersetzungsanweisung für die GSBL.

Definition der GSBL

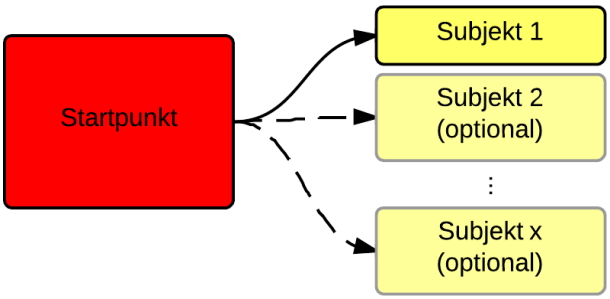
Wörter der GSBL

Wörter der GSBL bestehen aus einem Startpunkt, Subjekten und Eigenschaften.
Ein Wort ist nur dann ein Wort der GSBL wenn es genau einen Startpunkt, mindestens ein Subjekt und keine oder mehr Eigenschaften besitzt:
Startpunkt{1} Subjekt+ Eigenschaft*.
Desweiteren müssen alle Eigenschaften an ein Subjekt gebunden sein und es muss ein Pfad vom Startpunkt zu jedem Subjekt existieren.

Startpunkt

Startpunkte bestimmen die Art der SPARQL-Anfrage (z.B. SELECT oder ASK).
Es kann bei jedem Wort der GSBL nur einen Startpunkt geben. Startpunkte verweisen auf ein oder mehrere Hauptsubjekte.

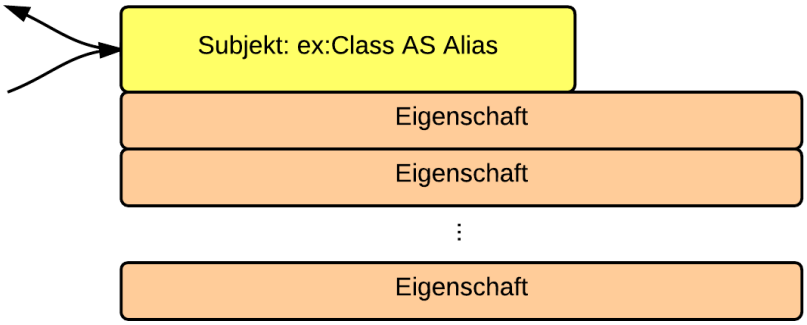
Allgemeiner Aufbau



Subjekt

Subjekte sind Klassen aus einem Namespace. Für jedes Subjekt gibt es die Möglichkeit mehrere Eigenschaften zu definieren, nach welchen die Entitäten eines Subjekts gefiltert werden bzw. welche Eigenschaften angezeigt werden sollen.
Jedes Subjekt kann durch einen Pfeil mit einer Eigenschaft oder einem Startpunkt verbunden werden.

Allgemeiner Aufbau



Technisches

Subjekte sind die Repräsentation einer *owl:class*.
Wählt man eine Klasse aus [im Beispiel *ex:Class*] so kann man diese optional mit einem alternativen Namen versehen [*Alias*]. Der Standardalias entspricht dabei dem *rdfs:label* der gewählten Klasse.
Eine Möglichkeit alle Klassen eines Namespaces abzufragen, deren englisches *rdfs:label* mit “ar” anfängt, wäre:

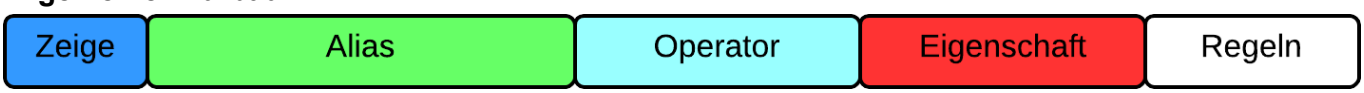
```
select distinct ?class ?alias {
   ?class a owl:Class .
   ?class rdfs:label ?label .
   FILTER REGEX(LANG(?label), "en", "i") .
   FILTER REGEX(STR(?label), "^Ar", "i") .
   BIND (STR(?label) as ?alias)
}
```

Beispiel an der dbpedia [hier].

Eigenschaft

Eigenschaften dienen zur Auswahl welche Properties eines Subjekts angezeigt werden sollen und nach welchen Kriterien das Subjektes eingeschränkt werden soll.

Allgemeiner Aufbau



Technisches

- *Zeige*: bestimmt, ob eine Eigenschaft in der Ausgabe angezeigt werden soll oder nicht
- *Alias*: Anzeigename der Eigenschaft (Standard ist hier *rdfs:label* der gewählten Eigenschaft)
- *Operator*: bestimmt Beziehung der Eigenschaft zum Subjekt (muss, kann, ...)
- *Eigenschaft*: Ein Element aus der Menge der aller Eigenschaften, deren *rdfs:domain* bzw. *rdfs:range* der *owl:class* des Subjektes entsprechen.
- *Regeln*: Je nach Operator und *rdfs:type* der gewählten Eigenschaft ist es möglich, dass weitere optionale Regeln bzw. Operationen ausgeführt werden.

Eine Möglichkeit alle Eigenschaften einers Subjekts zu bestimmen wäre:

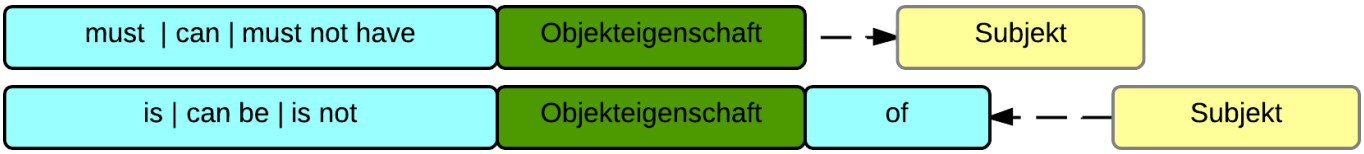
```
select distinct ?domain ?property ?range where {
   {
      dbpedia-owl:Person rdfs:subClassOf+ ?class.
      {
         ?property rdfs:range ?class .
         OPTIONAL { ?property rdfs:domain ?domain . }
         ?property rdfs:range ?range .
      } UNION {
         ?property rdfs:domain ?class .
         ?property rdfs:domain ?domain .
         OPTIONAL { ?property rdfs:range ?range . }
      }
   } UNION {
      ?property rdfs:domain dbpedia-owl:Person.
      ?property rdfs:domain ?domain .
      OPTIONAL { ?property rdfs:range ?range . }
   } UNION {
      ?property rdfs:range dbpedia-owl:Person.
      OPTIONAL { ?property rdfs:domain ?domain . }
      ?property rdfs:range ?range .
   }

}
```

Dieses Beispiel gibt alle Eigenschaften aus, deren *rdfs:domain* oder *rdfs:range* auf die Klasse Person und ihrer Überklassen zeigt.
Beispiel an der dbpedia [hier].

Je nach Art einer Eigenschaft sind verschiedene Operatoren und Regeln möglich. Grundsätzlich gibt es zwei Arten von Eigenschaften, mit denen der GSB umgehen kann. Eine Dateneigenschaft mit der *rdfs:range owl:DataTypeProperty* und eine Objekteigenschaft *owl:ObjectProperty*.

Objektdatentyp: Operatoren & Regeln

Unterstützt die Operatoren:



must have:	Die Objekteigenschaft wird zwingend vorausgesetzt
can have:	Die Objekteigenschaft ist optional (meist verwendet um eine Eigenschaft des Verweis-Subjekts optional anzuzeigen)
must not have:	Die Objekteigenschaft darf nicht erfüllt sein
is:	Das Verweis-Subjekt muss zwingend über die Objekteigenschaft mit dem Subjekt in Verbindung stehen
can be:	Das Verweis-Subjekt kann optional über die Objekteigenschaft mit dem Subjekt in Verbindung stehen
is not:	Das Verweis-Subjekt darf nicht über die Objekteigenschaft mit dem Subjekt in Verbindung stehen

Dateneigenschaft: Operatoren & Regeln

Unterstützt die Operatoren:

must have:	Die Dateneigenschaft wird zwingend vorausgesetzt
can have:	Die Dateneigenschaft ist optional
must not have:	Die Dateneigenschaft darf nicht erfüllt sein

Die folgende Tabelle erläutert welche arithmetischen und welche vergleichenden Operatoren für die verschiedenen Subtypen der Dateneigenschaft anwendbar sind.

Datentyp	Beschreibung	Arithmetische Operationen	Vergleichsoperatoren
string, normalizedString, Name	Zeichenkette	-	Gleichheit
boolean	Wahrheitswert	-	AND, NAND, OR, NOR, XOR, NOT
decimal, float, byte, double, int, short, long, negativeInteger, positiveInteger, nonpositiveInteger, nonnegativeInteger, unsignedLong, unsignedInt,	Zahlen	Addition, Subtraktion, Multiplikation, Division	größer (-gleich), kleiner (-gleich), Gleichheit

unsignedShort, unsignedByte,			
time, duration, date, dateTime, gYearMonth, gYear, gMonthDay, gDay, gMonth	Zeit Dauer Datum	-	größer (-gleich), kleiner (-gleich), Gleichheit
language	Sprache	-	Gleichheit

[Quelle: http://www.w3.org/TR/xmlschema-2/#built-in-derived]
[Quelle: http://www.w3.org/TR/rdf-sparql-query/#tests]

ANMERKUNG LUKAS:
Übersetzungsteil im Entwurf eines Entwurfs, also mit Vorsicht genießen!

Übersetzung der GSBL

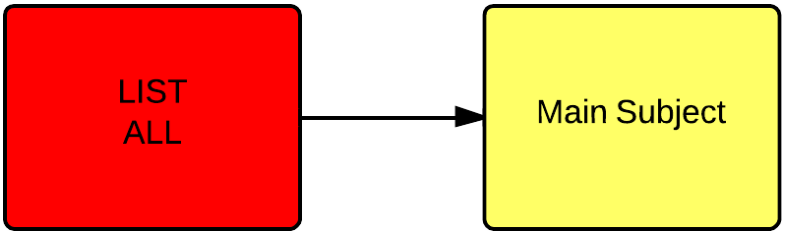
Die Übersetzung der GSBL zu SPARQL folgt festen Regeln. Die Übersetzung folgt an einem StartpunktFür die Übersetzung seien folgende Definitionen und Mengen definiert:

$shownValues = {* *}
$translated = {* *}
translateStartpunkt(* $id *)
translateSubjekt(* $id, $prefix = null*)
translateEigenschaft(* $id, $prefix *)
translateOperator(* $Operator, $Eigenschaft, $Regeln = null, $Prefix = null, $Alias = null *)

Die Übersetzung beginnt mit translateStartpunkt(* $name *)

Übersetzung Startpunkt

LIST ALL



Der Startpunkt “List ALL” steht für eine SELECT DISTINCT Abfrage und kann nur an ein Hauptsubjekt gebunden werden.

Übersetzung:

```
translateStartpunkt (* 'LIST ALL' *) =
      SELECT DISTINCT $shownValues
      WHERE {
            translateSubjekt('Hauptsubjekt')
      }
```

Übersetzung Subjekt

Ein Subjekt wird nur übersetzt, falls es noch nicht übersetzt wurde. Der Alias des Subjekts wird dem Ergebnis hinzugefügt, sowie

Übersetzung:

```
      translateSubjekt(* $id, $prefix = null *) =
      $alias sei der definierte "Alias" des Subjekts
      wenn $id nicht in $translated :
            füge $id zu $translated hinzu
            wenn $prefix != null: $alias = $prefix_$alias
            füge $alias zu $shownValues hinzu
            ?$alias a ex:class .
            für jede Eigenschaft:
                translateEigenschaft($name, $alias)
```