1 Geospatial Literals

GeoSPARQL currently only supports two Literal types:

- WKT Literal

```
"POLYGON((-77.089005 38.913574,-77.029953 38.913574,
-77.029953 38.886321,-77.089005 38.886321,
-77.089005 38.913574))"^^geo:wktLiteral
```

- GML Literal

```
"<gml:Point srsName=\"http://www.opengis.net/def/crs/OGC/1.3/CRS84\"
 xmlns:gml=\"http://www.opengis.net/gml\">
<gml:posList srsDimension=\"2\">-83.38 33.95</gml:posList>
</gml:Point>"^^ogc:GMLLiteral
```

Considering the large heterogeneity of geospatial data formats, the support of more geospatial data literals should be considered.

1.1 Change Request

The addition and standardization of more literal types. To me the following geospatial literal types should be included with high priority:

- GeoHash Literal[1]

```
"gbsuv"^^geo:geoHashLiteral
```

- GeoJSON Literal[2]
  - Why? - Very common webstandard for maps

```
{\"type\":\"LineString\",
\"coordinates\":[[39.046368900000004,22.2237116],
[39.0462247,22.223842]]}^^geo:geoJSONLiteral
```

- GPX Literal[3]
  - Why? - Useful for capturing and working with GPS data

```
"<gpx version=\"1.0\"><name>Example gpx</name> <trk>
    <name>Example gpx</name>
    <trkseg>
    <trkpt lat=\"40.133485900000004\" lon=\"22.5475202\"/>
    <trkpt lat=\"40.133099800000004\" lon=\"22.547705500000003\"/>
    </trkseg></trk></gpx>"^^geo:gpxLiteral
```

[1] http://geohash.org
[2] https://geojson.org
[3] https://www.topografix.com/gpx.asp

- KML Literal[4]

```
"<linestring>
<coordinates>23.5530286,38.0473921 23.5554701,38.0484895</coordinates>
</linestring>"^^geo:kmlLiteral
```

- TWKB Literal[5]

```
"\x02000202020808"^^geo:twkbLiteral
```

- (Hex)(E)WKB Literal[6]
  - Why? - Standard format in many other relational database systems

More data literals to be included could be:

- Polyshape(Polyline) Literal[7]

```
"1gwpzF{gr_Da@M"^^geo:polyshapeLiteral
```

- DXF Literal[8]
- Geobuf Literal[9]

```
"GAAiEAoOCgwIBBoIAAAAAgIAAAE="^^geo:geobufLiteral
```

- GeoURI Literal[10]

```
"geo:39.19972000122019,22.763096060395224;crs=EPSG:4326"
^^geo:geoURILiteral
```

- OSM/XML Literal[11]
- SVG Literal[12]
- TopoJSON Literal[13]

```
"{"type": "Topology",
"objects": { "example": {
"type": "GeometryCollection",
"geometries": [{ "type": "Point",
"properties": {"prop0": "value0"},
"coordinates": [102, 0.5]},
{
"type": "LineString",
"properties": {"prop0": "value0","prop1": 0},
```

[4] https://developers.google.com/kml/documentation/kmlreference?csw=1
[5] https://github.com/TWKB/Specification/blob/master/twkb.md
[6] https://mariadb.com/kb/en/library/well-known-binary-wkb-format/
[7] https://developers.google.com/maps/documentation/utilities/polylinealgorithm
[8] http://www.crlf.de/Verlag/DXF-intern/DXF-intern.html
[9] https://github.com/mapbox/geobuf
[10] https://tools.ietf.org/html/rfc5870
[11] https://wiki.openstreetmap.org/wiki/OSM_XML
[12] https://www.w3.org/Graphics/SVG/
[13] https://github.com/topojson/topojson

```
"arcs": [0]
},
{"type": "Polygon",
"properties": {"prop0": "value0","prop1": {"this": "that"}},
"arcs": [[-2]]}]}},
"arcs": [
[[102, 0], [103, 1], [104, 0], [105, 1]],
[[100, 0], [101, 0], [101, 1], [100, 1], [100, 0]]
]
}"^^geo:topoJSONLiteral
```

– X3D Literal[14]

Ontology changes The GeoSPARQL ontology would need to be extended by the described literal types. An extended ontology can be found in https://github.com/i3mainz/geosparql2.0

1.2 Implementation

Implementations of all aforementioned literals exist in the following projects:

– postgis-jena[15] - Extension for Apache Jena
– rdf4j-postgis[16] - Extension for RDF4j

[14] https://www.web3d.org/x3d/what-x3d
[15] https://github.com/i3mainz/postgis-jena
[16] https://github.com/i3mainz/rdf4j-postgis