

# Open Geospatial Consortium

Date: 2011-01-28

Reference number of this OGC® project document: **OGC 09-157r4**

Version: 1.0.4

Category: OGC® Draft Implementation Specification

Editors: Matthew Perry and John Herring

## **GeoSPARQL - A geographic query language for RDF data**

A proposal for an OGC Draft Candidate Standard

### **Copyright notice**

Copyright © 2011 Open Geospatial Consortium.

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

### **Warning**

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OGC® Draft Candidate Standard
Document subtype:	Proposal
Document stage:	Draft
Document language:	English

## Contents

Foreword .....	ix
Introduction.....	x
1 Scope .....	1
2 Conformance .....	1
3 Normative references .....	2
4 Terms and definitions .....	2
5 Conventions .....	3
5.1 Symbols and abbreviated terms .....	3
5.2 XML namespaces .....	3
6 GeoSPARQL Vocabulary .....	4
6.1 Classes and datatypes .....	4
6.1.1 Class: <code>ogc:SpatialObject</code> .....	4
6.1.2 Class: <code>ogc:Feature</code> .....	4
6.1.3 Class: <code>ogc:Geometry</code> .....	5
6.1.4 Subclasses of <code>ogc:Geometry</code> .....	5
6.1.5 RDFS Datatypes.....	10
6.2 Properties .....	12
6.2.1 Standard Properties for <code>ogc:Feature</code> .....	12
6.2.2 Standard Properties for <code>ogc:Geometry</code> .....	13
6.2.3 Standard Properties for Spatial Relations .....	17
6.3 Illustrative Example .....	21
6.3.1 Example RDF Data.....	21
6.3.2 Example GeoSPARQL Queries .....	24
7 Domain-specific SPARQL Filter Functions .....	25
7.1 Filter Function: <code>ogcf:relate</code> .....	26
7.2 Filter Function: <code>ogcf:distance</code> .....	26
7.3 Filter Function: <code>ogcf:buffer</code> .....	26
7.4 Filter Function: <code>ogcf:convexHull</code> .....	26
7.5 Filter Function: <code>ogcf:intersection</code> .....	26
7.6 Filter Function: <code>ogcf:union</code> .....	27
7.7 Filter Function: <code>ogcf:difference</code> .....	27
7.8 Filter Function: <code>ogcf:symDifference</code> .....	27
7.9 Filter Function: <code>ogcf:envelope</code> .....	27
7.10 Filter Function: <code>ogcf:boundary</code> .....	27
7.11 Example GeoSPARQL FILTER queries .....	27
8 Query Transformation Rules .....	29
8.1 Transformation Rule: <code>ogc&gt;equals</code> .....	29
8.2 Transformation Rule: <code>ogc:disjoint</code> .....	30
8.3 Transformation Rule: <code>ogc:intersects</code> .....	31
8.4 Transformation Rule: <code>ogc:touches</code> .....	31
8.5 Transformation Rule: <code>ogc:crosses</code> .....	32
8.6 Transformation Rule: <code>ogc:within</code> .....	32
8.7 Transformation Rule: <code>ogc:contains</code> .....	33
8.8 Transformation Rule: <code>ogc:overlaps</code> .....	34
8.9 Example Query Rewrite .....	34
9 OWL-based Spatial Reasoning .....	35
10 Future Work .....	35
Abstract Test Suite .....	37
A.1 Abstract Test Suite for GeoSPARQL .....	37
A.1.1 WKT Geometry Serialization Conformance Class.....	37

<b>A.1.2 GML Geometry Serialization Conformance Class .....</b>	<b>41</b>
<b>Bibliography.....</b>	<b>47</b>

## Table of Requirements

<b>Req 1</b>	All feature and geometry objects shall subtype either directly or through intermediaries from the root class <code>ogc:SpatialObject</code> .....	4
<b>Req 2</b>	All feature objects shall subtype either directly or through intermediaries from the class <code>ogc:Feature</code> .....	5
<b>Req 3</b>	All geometry objects shall subtype either directly or through intermediaries from the class <code>ogc:Geometry</code> .....	5
<b>Req 4</b>	All RDFS Literals of type <code>ogc:WKTLiteral</code> shall consist of an IRI identifying the coordinate reference systems followed by Simple Features Well Known Text (WKT) describing a geometric value. ....	11
<b>Req 5</b>	An RDFS Literal of type <code>ogc:WKTLiteral</code> that represents an empty geometry shall use an empty string for its lexical value. ....	11
<b>Req 6</b>	All RDFS Literals of type <code>ogc:GMLLiteral</code> shall consist of a valid element from the GML schema that implements a subtype of <code>GM_Object</code> . ....	11
<b>Req 7</b>	An RDFS Literal of type <code>ogc:GMLLiteral</code> that represents an empty geometry shall use an empty string for its lexical value. ....	12
<b>Req 8</b>	Implementations claiming conformance to the GML GeoSPARQL conformance class shall document the GML profile supported by the implementation. ....	12
<b>Req 9</b>	Compliant implementations shall allow the RDF properties <code>ogc:hasGeometry</code> and <code>ogc:defaultGeometry</code> to be used directly in SPARQL triple patterns to retrieve information about instances of <code>ogc:Feature</code> . ....	12
<b>Req 10</b>	Implementations claiming conformance with the WKT conformance class shall allow the RDF property <code>ogc:asWKT</code> to be used directly in GeoSPARQL triple patterns to retrieve an <code>ogc:WKTLiteral</code> for elements of type <code>ogc:Geometry</code> ...	13
<b>Req 11</b>	Implementations claiming conformance with the GML conformance class shall allow the RDF property <code>ogc:asGML</code> to be used directly in GeoSPARQL triple patterns to retrieve an <code>ogc:GMLLiteral</code> for elements of type <code>ogc:Geometry</code> ..	14
<b>Req 12</b>	Compliant implementations shall support an <code>ogc:dimension</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.....	14
<b>Req 13</b>	Compliant implementations shall support an <code>ogc:coordinateDimension</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.....	15
<b>Req 14</b>	Compliant implementations shall support an <code>ogc:spatialDimension</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.....	15
<b>Req 15</b>	Compliant implementations shall support an <code>ogc:isEmpty</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.....	16
<b>Req 16</b>	Compliant implementations shall support an <code>ogc:isSimple</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.....	16
<b>Req 17</b>	Compliant implementations shall support an <code>ogc:is3D</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features. ....	17
<b>Req 18</b>	Compliant implementations shall allow the following RDF properties to be used directly in SPARQL triple patterns to test for existence of binary spatial relations between instances of <code>ogc:SpatialObject</code> . The named operators shall be consistent with those in Simple Features. The RCC operators shall be valid for 2D topological dimension geometric objects (Regions).....	18

**Req 19** Compliant implementations shall support the SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary, consistent with Simple Features. .... 25

**Req 20** Implementations claiming conformance with the WKT GeoSPARQL conformance class shall support the use of ogc:WKTLiteral in place of ogc:GeomLiteral in SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary. .... 25

**Req 21** Implementations claiming conformance with the GML GeoSPARQL conformance class shall support the use of ogc:GMLLiteral in place of ogc:GeomLiteral in SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary. .... 25

**Req 22** Compliant implementations shall use the following rules to test spatial relations between instances of ogc:SpatialObject..... 29

**Req 23** Implementations claiming conformance with the WKT GeoSPARQL conformance class shall support the use of ogc:asWKT in place of ogc:asText in the following rules to test spatial relations between instances of ogc:SpatialObject..... 29

**Req 24** Implementations claiming conformance with the GML GeoSPARQL conformance class shall support the use of ogc:asGML in place of ogc:asText in the following rules to test spatial relations between instances of ogc:SpatialObject..... 29

## Table of Requirement by URI

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialObjectClass">http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialObjectClass</a> .....	4
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureClass">http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureClass</a> .....	5
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryClass">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryClass</a> .....	5
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteral">http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteral</a> .....	11
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteralEmpty">http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteralEmpty</a> .....	11
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteral">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteral</a> .....	11
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteralEmpty">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteralEmpty</a> .....	12
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLProfile">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLProfile</a> .....	12
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureProperties">http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureProperties</a> .....	12
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsWKTLiteral">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsWKTLiteral</a> .....	13
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsGMLLiteral">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsGMLLiteral</a> .....	14
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoDim">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoDim</a> .....	14
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoCoordDim">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoCoordDim</a> .....	15
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoSpatDim">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoSpatDim</a> .....	15
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeolsEmpty">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeolsEmpty</a> .....	16
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeolsSimple">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeolsSimple</a> .....	16
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/Geols3D">http://www.opengis.net/spec/GeoSPARQL/1.0/req/Geols3D</a> .....	17
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialRelations">http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialRelations</a> .....	18
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterFunctions">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterFunctions</a> .....	25
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterWKT">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterWKT</a> .....	25
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterGML">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterGML</a> .....	26
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransRules">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransRules</a> .....	29
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransWKT">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransWKT</a> .....	29
<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransGML">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransGML</a> .....	29

## Preface

This draft standard defines an enhanced query language based on SPARQL that uses an RDF (Resource Description Framework) representation for geographic information based on the General Feature Model (from ISO 19109) and spatial predicates consistent with the spatial schema (from ISO 19107), and query predicates for geometric and topological operators (from Simple Feature, or ISO 19125)

## Submitting organizations

The following organizations submitted this Implementation Specification to the Open Geospatial Consortium Inc. as a Request For Comment (RFC):

Oracle America

## Submission contact points

All questions regarding this submission should be directed to the editor or the submitters:

CONTACT	COMPANY	EMAIL
John R. Herring	Oracle America	john.herring@oracle.com
Matthew Perry	Oracle America	matthew.perry@oracle.com
Xavier Lopez	Oracle America	xavier.lopez@oracle.com

## Revision history

Date	Release	Author	Paragraph modified	Description
27 Oct. 2009	Draft	Matthew Perry	Clause 6	Technical Draft
11 Nov. 2009	Draft	John R. Herring	All	Creation
6 Jan. 2010	Draft	John R. Herring		Comment responses

<b>Date</b>	<b>Release</b>	<b>Author</b>	<b>Paragraph modified</b>	<b>Description</b>
30 March 2010	Draft	Matthew Perry		Comment responses
26 Oct. 2010	Draft	Matthew Perry	All	Revision based on working group discussion
28 Jan. 2011	Draft	Matthew Perry	All	Revision based on working group discussion

### **Changes to the OGC<sup>®</sup> Abstract Specification**

The OGC<sup>®</sup> Abstract Specification does not require changes to accommodate this OGC<sup>®</sup> standard.

## **Foreword**

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

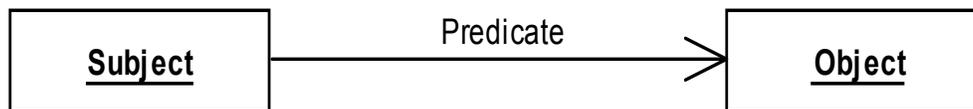
Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

## Introduction

RDF (Resource Description Framework) is, among other things, a data model built on edge-node "graphs." Each link in a graph consists of three things (with many aliases depending on the mapping from other types of data models):

1. Subject (start node, instance (OOPL), feature, blank node)
2. Predicate (verb, property, attribute, relation, member(OOPL), link, reference)
3. Object (value, end node, values can be used as a Subject)

Any of the three values in a single triple can be represented via an IRI (with an optional fragment identifier). Subjects and objects are called nodes and can be represented as a blank node (usually with a local identifier with no meaning). Objects can also be represented as a literal value. Note that the same node may play the role of a Subject in some edges, and the role of the Object in others.



**Figure 1: RDF Triple**

Almost all data can be presented or represented in RDF. In particular, it is an easy match to the (feature-instance-by-id, attribute, value) tuples of the General Feature Model (ISO 19109), and for the relational model as (table primary key, column, value).

Complex information can often be expressed as graphs, or lexically as list of triples using common identifiers for nodes used in common, having the same value or identifier.

From <http://dbpedia.org/page/SPARQL>:

SPARQL is an RDF query language; its name is a recursive acronym that stands for SPARQL Protocol and RDF Query Language. It was standardized by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium, and is considered a key semantic web technology. On 15 January 2008, SPARQL became an official W3C Recommendation. SPARQL allows [a] query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

SPARQL queries work on RDF representations of data by finding patterns that match templates in the query, in effect finding information graphs in the RDF data based on the templates and filters (constraints on nodes and edges) expressed in the query. This query template is represented in the SPARQL query by a set of parameterized "query variables" appearing in a sequence of RDF triples and filters. If the query processor finds a set of triples in the data (converted to an RDF graph in some predetermined standard manner) then the values that the "query variables" take on in those triples become a solution to the query request. The values of the variables are returned in the query result in a format based on the "SELECT" clause of the query (similar to SQL).

In addition to predicates defined in this manner, the SPARQL query may contain filter functions that can be used to constrain the values of the nodes. Several mechanisms are available to extend filter functions to allow for predicates calculated directly on data

values. The SPARQL specification in section 11.6 (<http://www.w3.org/TR/rdf-sparql-query/#extensionFunctions>) describes the mechanism for invocation of such a filter function.

Identifiers in SPARQL queries may be prefixed with globally defined namespace prefixes.

Spatial examples are given in the text of this draft candidate standard.

This draft candidate standard takes SPARQL as it is defined by the W3C RDF DAWG and extends it to deal with feature data as defined in Simple Features. The standard defines a vocabulary and datatypes for representing feature and geometry data in RDF, and it defines a set of SPARQL filter functions and query rewrite rules.

Two conformance classes for GeoSPARQL have been defined. The conformance class determines the literal representation of a geometry used in an implementation. A variety of text-based serializations of geometry data exist (e.g., WKT, GML, KML). This version of the GeoSPARQL standard defines two such conformance classes: one for WKT and one for GML. Any reasonable text-based serialization could potentially be used to define new conformance classes as long as: (1) it expresses geometry in a specific system, and (2) associates geometries to that spatial reference system.



---

## GeoSPARQL - A geographic query language for RDF data

### Scope

The purpose of this document is to define a spatial extension to the SPARQL query language for geographic information as defined in OGC.

There are three main components of GeoSPARQL. The first component is the definition of a vocabulary to represent features, geometries and their relationships. The second component of GeoSPARQL is a set of domain-specific, spatial filter functions for use in SPARQL queries, and the third component is a set of query transformation rules.

### Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing.

Many text-based serializations of geometry values exist (e.g. WKT, GML, KML, GeoJSON). This standard is therefore designed to allow multiple conformance classes based on different geometry serialization methods. This clause defines two conformance classes for GeoSPARQL: one based on Well-known Text serializations of geometry values and another based on GML serializations of geometry values. It is anticipated that new conformance classes will be created in the future using other text-based serializations of geometry values (e.g. KML, GeoJSON).

The WKT conformance class defines an `ogc:WKTLiteral` RDFS datatype and a corresponding `ogc:asWKT` RDF property. `ogc:WKTLiteral` values are used as arguments to many FILTER functions and as return types for many FILTER functions in this conformance class. Any implementation claiming conformance with the WKT conformance class shall pass all test cases in Subclause A.1 of the test suite.

The GML conformance class defines an `ogc:GMLLiteral` RDFS datatype and a corresponding `ogc:asGML` RDF property. `ogc:GMLLiteral` values are used as arguments to many FILTER functions and as return types for many FILTER functions in this conformance class. Any implementation claiming conformance with the GML conformance class shall document the GML profile supported by the implementation. Any implementation claiming conformance with the GML conformance class shall pass all test cases in Subclause A.2 of the test suite.

## Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this draft candidate standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this draft candidate standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies<sup>[1]</sup>.

- a. ISO 19107: Geographic information — Geomatics — Spatial schema
- b. ISO 19109: Geographic information — Geomatics — Rules for application schemas
- c. ISO 19125-1: Geographic information — Simple feature access — Part 1: Common architecture
- d. RDF/XML Syntax Specification (Revised), W3C Recommendation 10 February 2004.
- e. RDF Semantics, W3C Recommendation, 10 February 2004
- f. Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004
- g. RIF Overview, W3C Working Group Note 22 June 2010
- h. RIF Core Dialect, W3C Recommendation 22 June 2010
- i. SPARQL Query Language for RDF, W3C Recommendation 15 January 2008
- j. OWL 2 Web Ontology Language Document Overview, W3C Recommendation 27 October 2009

## Terms and definitions

For the purposes of this document, the terms and definitions given in any of the normative references and the following apply.

## Conventions

### 1.1 Symbols and abbreviated terms

In this draft candidate standard, the following common acronyms are used:

RDF	Resource Description Framework
RDFS	RDF Schema
OWL	Web Ontology Language
RIF	Rule Interchange Format
SPARQL	SPARQL Protocol and RDF Query Language
GML	Geography Markup Language
GFM	General Feature Model (as defined in ISO 19109)
RCC	Region Connection Calculus
W3C	World Wide Web Consortium ( <a href="http://www.w3.org/">http://www.w3.org/</a> )
WKT	Well Known Text (as defined by Simple Features or ISO 19125)
XML	eXtended Markup Language
KML	Keyhole Markup Language
GeoJSON	Geographic JavaScript Object Notation

### 1.2 XML namespaces<sup>[2]</sup>

The following namespace prefixes are used throughout this document.

ogc	<a href="http://www.opengis.net/rdf#">http://www.opengis.net/rdf#</a>	The OGC-wide data type and operators as RDF definitions including contributions from Simple Features
ogcf	<a href="http://www.opengis.net/rdf/functions#">http://www.opengis.net/rdf/functions#</a>	The external function definitions
xsd	<a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a>	
rdf	<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a>	
rdfs	<a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a>	
owl	<a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a>	
my	<a href="http://somewhere/ApplicationSchema#">http://somewhere/ApplicationSchema#</a>	The application schema for the data

This is the equivalent of having the following at the beginning of each query of the equivalent “xmlns” declarations in the heading of any “xml.”

PREFIX	ogc:	< <a href="http://www.opengis.net/rdf#">http://www.opengis.net/rdf#</a> >
PREFIX	ogcf:	< <a href="http://www.opengis.net/rdf/functions#">http://www.opengis.net/rdf/functions#</a> >
PREFIX	xsd:	< <a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a> >
PREFIX	rdf:	< <a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a> >
PREFIX	rdfs:	< <a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a> >
PREFIX	owl:	< <a href="http://www.w3.org/2002/07/owl#">http://www.w3.org/2002/07/owl#</a> >
PREFIX	my:	< <a href="http://somewhere/ApplicationSchema#">http://somewhere/ApplicationSchema#</a> >

## GeoSPARQL Vocabulary

This section defines a set of classes and properties for representing geospatial data. The resulting vocabulary can be used to construct SPARQL graph patterns for querying appropriately modeled geospatial data. RDFS and OWL vocabulary have both been used so that the vocabulary can be understood by systems that support only RDFS entailment and by systems that support OWL-based reasoning.

### 1.3 Classes and datatypes

Three main classes are defined: `ogc:SpatialObject`, `ogc:Feature` and `ogc:Geometry`. In addition, `ogc:Geometry` has sixteen subclasses defining a hierarchy of geometry types. Note that instances of `ogc:Feature` and `ogc:SpatialObject` are not required to have an associated `ogc:Geometry` value. A literal representation of a geometry is needed so that geometric values may be treated as a single unit. Such a representation allows geometries to be passed to external functions for computations and to be returned from a query. GeoSPARQL defines new RDFS Datatypes for this purpose. Each conformance class for GeoSPARQL has an associated RDFS Datatype.

#### 1.3.1 Class: `ogc:SpatialObject`

The class `ogc:SpatialObject` is defined by the following:

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#SpatialObject">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>SpatialObject</rdfs:label>
  <rdfs:comment>
    The class SpatialObject, superclass of everything feature
    or geometry that can have a spatial representation.
  </rdfs:comment>
</rdf:Description>
```

<p><b>Req 1 All feature and geometry objects shall subtype either directly or through intermediaries from the root class <code>ogc:SpatialObject</code>.</b></p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p><a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialObjectClass">http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialObjectClass</a></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------

#### 1.3.2 Class: `ogc:Feature`

The class `ogc:Feature` is defined by the following:

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#Feature">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>Feature</rdfs:label>
  <rdfs:subClassOf rdf:resource=
```

```

    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:comment>
    The class Feature, superclass of everything
    feature.
  </rdfs:comment>
</rdf:Description>

```

<b>Req 2 All feature objects shall subtype either directly or through intermediaries from the class <code>ogc:Feature</code></b>
----------------------------------------------------------------------------------------------------------------------------------

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureClass">http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureClass</a>
-------------------------------------------------------------------------------------------------------------------------------------

### 1.3.3 Class: `ogc:Geometry`

The class `ogc:Geometry` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#Geometry">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>Geometry</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:comment>
    The class Geometry, superclass of everything
    geometry.
  </rdfs:comment>
</rdf:Description>

```

<b>Req 3 All geometry objects shall subtype either directly or through intermediaries from the class <code>ogc:Geometry</code>.</b>
-------------------------------------------------------------------------------------------------------------------------------------

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryClass">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryClass</a>
---------------------------------------------------------------------------------------------------------------------------------------

### 1.3.4 Subclasses of `ogc:Geometry`

#### 1.3.4.1 Class: `ogc:Point`

The class `ogc:Point` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#Point">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>Point</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:comment>
    Topological 0-dimensional geometric primitive representing
    a position.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.2 Class: ogc:Curve

The class ogc:Curve is defined by the following:

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#Curve">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>Curve</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:comment>
    Topological 1-dimensional geometric primitive representing
    the continuous image of a line.
  </rdfs:comment>
</rdf:Description>
```

### 1.3.4.3 Class: ogc:LineString

The class ogc:LineString is defined by the following:

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#Curve">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>LineString</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Curve"/>
  <rdfs:comment>
    A curve composed of straight line segments.
  </rdfs:comment>
</rdf:Description>
```

### 1.3.4.4 Class: ogc:ArcString

The class ogc:ArcString is defined by the following:

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#ArcString">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>ArcString</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Curve"/>
  <rdfs:comment>
    A curve composed of arc segments.
  </rdfs:comment>
</rdf:Description>
```

### 1.3.4.5 Class: ogc:Surface

The class ogc:Surface is defined by the following:

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#Surface">
```

```

<rdf:type rdf:resource=
  "http://www.w3.org/2000/01/rdf-schema#Class"/>
<rdf:type rdf:resource=
  "http://www.w3.org/2002/07/owl#Class"/>
<rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
<rdfs:label>Surface</rdfs:label>
<rdfs:subClassOf rdf:resource=
  "http://www.opengis.net/rdf#Geometry"/>
<rdfs:comment>
  A 2-dimensional geometric primitive representing a continuous
  image of a region of a plane
</rdfs:comment>
</rdf:Description>

```

### 1.3.4.6 Class: ogc:Polygon

The class ogc:Polygon is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#Polygon">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>Polygon</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Surface"/>
  <rdfs:comment>
    A planar surface defined by 1 exterior boundary and 0 or more
    interior boundaries.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.7 Class: ogc:Triangle

The class ogc:Triangle is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#Triangle">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>Triangle</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Polygon"/>
  <rdfs:comment>
    A Polygon with an outer boundary consisting of a linear ring
    with exactly four points and no inner boundaries.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.8 Class: ogc:PolyhedralSurface

The class ogc:PolyhedralSurface is defined by the following:

```

<rdf:Description
  rdf:about="http://www.opengis.net/rdf#PolyhedralSurface">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>

```

```

<rdf:type rdf:resource=
  "http://www.w3.org/2002/07/owl#Class"/>
<rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
<rdfs:label>PolyhedralSurface</rdfs:label>
<rdfs:subClassOf rdf:resource=
  "http://www.opengis.net/rdf#Surface"/>
<rdfs:comment>
  A surface composed of polygon patches connected along their
  common boundary curves.
</rdfs:comment>
</rdf:Description>

```

### 1.3.4.9 Class: ogc:TIN

The class ogc:TIN is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#TIN">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>TIN</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#PolyhedralSurface"/>
  <rdfs:comment>
    A polyhedral surface that is composed only of triangles.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.10 Class: ogc:GeometryCollection

The class ogc:GeometryCollection is defined by the following:

```

<rdf:Description
  rdf:about="http://www.opengis.net/rdf#GeometryCollection">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>GeometryCollection</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:comment>
    A geometric object that is a collection of some number of
    geometric objects.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.11 Class: ogc:MultiPoint

The class ogc:MultiPoint is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#MultiPoint">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>

```

```

<rdfs:label>MultiPoint</rdfs:label>
<rdfs:subClassOf rdf:resource=
  "http://www.opengis.net/rdf#GeometryCollection"/>
<rdfs:comment>
  A MultiPoint is a 0-dimensional GeometryCollection whose
  elements are Points.
</rdfs:comment>
</rdf:Description>

```

### 1.3.4.12 Class: ogc:MultiCurve

The class ogc:MultiCurve is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#MultiCurve">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>MultiCurve</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#GeometryCollection"/>
  <rdfs:comment>
    A MultiCurve is a 1-dimensional GeometryCollection whose
    elements are Curves.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.13 Class: ogc:MultiLineString

The class ogc:MultiLineString is defined by the following:

```

<rdf:Description
  rdf:about="http://www.opengis.net/rdf#MultiLineString">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>MultiLineString</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#MultiCurve"/>
  <rdfs:comment>
    A MultiLineString is a 1-dimensional GeometryCollection whose
    elements are LineStrings.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.14 Class: ogc:MultiArcString

The class ogc:MultiArcString is defined by the following:

```

<rdf:Description
  rdf:about="http://www.opengis.net/rdf#MultiArcString">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>MultiArcString</rdfs:label>

```

```

<rdfs:subClassOf rdf:resource=
  "http://www.opengis.net/rdf#MultiCurve"/>
<rdfs:comment>
  A MultiArcString is a 1-dimensional GeometryCollection whose
  elements are ArcStrings.
</rdfs:comment>
</rdf:Description>

```

### 1.3.4.15 Class: **ogc:MultiSurface**

The class **ogc:MultiSurface** is defined by the following:

```

<rdf:Description
  rdf:about="http://www.opengis.net/rdf#MultiSurface">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>MultiSurface</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#GeometryCollection"/>
  <rdfs:comment>
    A MultiSurface is a 2-dimensional GeometryCollection whose
    elements are Surfaces.
  </rdfs:comment>
</rdf:Description>

```

### 1.3.4.16 Class: **ogc:MultiPolygon**

The class **ogc:MultiPolygon** is defined by the following:

```

<rdf:Description
  rdf:about="http://www.opengis.net/rdf#MultiPolygon">
  <rdf:type rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#Class"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>MultiPolygon</rdfs:label>
  <rdfs:subClassOf rdf:resource=
    "http://www.opengis.net/rdf#MultiSurface"/>
  <rdfs:comment>
    A MultiPolygon is a 2-dimensional GeometryCollection whose
    elements are Polygons.
  </rdfs:comment>
</rdf:Description>

```

## 1.3.5 RDFS Datatypes

This section defines two RDFS Datatypes: **ogc:WKTLiteral** is defined for the WKT conformance class, and **ogc:GMLLiteral** is defined for the GML conformance class.

### 1.3.5.1 RDFS Datatype: **ogc:WKTLiteral**

```

<rdfs:Datatype rdf:about=
  "http://www.opengis.net/rdf#WKTLiteral">
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>

```

```

<rdfs:label>Well-known Text Literal</rdfs:label>
<rdfs:comment>
  A Well-known Text serialization of a geometry object.
</rdfs:comment>
</rdfs:Datatype>

```

**Req 4 All RDFS Literals of type `ogc:WKTLiteral` shall consist of an IRI identifying the coordinate reference systems followed by Simple Features Well Known Text (WKT) describing a geometric value.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteral>

Valid `ogc:WKTLiterals` are formed by concatenating a valid IRI as defined in i, one or more spaces (Unicode U+0020 character) as a separator, and a WKT string as defined in a. The beginning IRI identifies the spatial reference system for the geometry. The example `ogc:WKTLiteral` below encodes a point geometry in the WGS 84 geodetic longitude-latitude spatial reference system:

```

"<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
Point(-83.38 33.95)"^^ogc:WKTLiteral

```

**Req 5 An RDFS Literal of type `ogc:WTKLiteral` that represents an empty geometry shall use an empty string for its lexical value.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteralEmpty>

### 1.3.5.2 RDFS Datatype: `ogc:GMLLiteral`

```

<rdfs:Datatype rdf:about="http://www.opengis.net/rdf#GMLLiteral">
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:subClassOf rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral"/>
  <rdfs:label>GML Literal</rdfs:label>
  <rdfs:comment>
    The datatype of GML literal values.
  </rdfs:comment>
</rdfs:Datatype>

```

Valid `ogc:GMLLiterals` are formed by encoding geometry information as a valid element from the GML schema that implements a subtype of `GM_Object`. For example, in GML 3.2.1 this is every element directly or indirectly in the substitution group of the element `{http://www.opengis.net/gml/3.2}AbstractGeometry`. In GML 3.1.1 and GML 2.1.2 this is every element directly or indirectly in the substitution group of the element `{http://www.opengis.net/gml}_Geometry`.

**Req 6 All RDFS Literals of type `ogc:GMLLiteral` shall consist of a valid element from the GML schema that implements a subtype of `GM_Object`.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteral>

The example `ogc:GMLLiteral` below encodes a point geometry in the WGS 84 geodetic longitude-latitude spatial reference system:

```

"<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

```

<b>Req 7 An RDFS Literal of type <code>ogc:GMLLiteral</code> that represents an empty geometry shall use an empty string for its lexical value.</b>
-----------------------------------------------------------------------------------------------------------------------------------------------------

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteralEmpty">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteralEmpty</a>
-------------------------------------------------------------------------------------------------------------------------------------------

<b>Req 8 Implementations claiming conformance to the GML GeoSPARQL conformance class shall document the GML profile supported by the implementation.</b>
----------------------------------------------------------------------------------------------------------------------------------------------------------

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLProfile">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLProfile</a>
---------------------------------------------------------------------------------------------------------------------------------

## 1.4 Properties

### 1.4.1 Standard Properties for `ogc:Feature`

Several properties are defined for describing features and geometries.

<b>Req 9 Compliant implementations shall allow the RDF properties <code>ogc:hasGeometry</code> and <code>ogc:defaultGeometry</code> to be used directly in SPARQL triple patterns to retrieve information about instances of <code>ogc:Feature</code>.</b>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureProperties">http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureProperties</a>
-----------------------------------------------------------------------------------------------------------------------------------------------

#### 1.4.1.1 Property: `ogc:hasGeometry`

The property `ogc:hasGeometry` is used to link a feature with a geometry that represents its spatial extent. A given feature may have many associated geometries.

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#hasGeometry">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>hasGeometry</rdfs:label>
  <rdfs:comment>
    A spatial representation for a given feature.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Feature"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
</rdf:Description>
```

#### 1.4.1.2 Property: `ogc:defaultGeometry`

The property `ogc:defaultGeometry` is used to link a feature with its default geometry. The default geometry is the geometry that should be used for spatial calculations in the absence of a request for a specific geometry (e.g. in the case of query rewrite Section 8).

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#defaultGeometry">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

```

<rdf:type rdf:resource=
  "http://www.w3.org/2002/07/owl#ObjectProperty"/>
<rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
<rdfs:label>primaryGeometry</rdfs:label>
<rdfs:comment>
  The default geometry to be used in spatial calculations
</rdfs:comment>
<rdfs:subPropertyOf rdf:resource=
  "http://www.opengis.net/rdf#hasGeometry"/>
<rdfs:domain rdf:resource=
  "http://www.opengis.net/rdf#Feature"/>
<rdfs:range rdf:resource=
  "http://www.opengis.net/rdf#Geometry"/>
</rdf:Description>

```

GeoSPARQL does not restrict the cardinality of the `ogc:defaultGeometry` property. It is thus possible for a feature to have more than one distinct default geometry. This situation does not result in a query processing error; SPARQL graph pattern matching simply proceeds as normal. Certain queries may, however, give logically inconsistent results. For example, given a feature `my:f1` with two distinct default geometries, the query below could have a non-empty result.

```

PREFIX ogc: <http://www.opengis.net/rdf#>
PREFIX ogcf: <http://www.opengis.net/rdf/functions#>
PREFIX epsg: <http://www.opengis.net/def/crs/EPSG/0/>
PREFIX my: <http://somewhere/ApplicationSchema#>
SELECT ?f2
WHERE {
  my:f1 ogc:defaultGeometry ?g1
    ?g1 ogc:asWKT ?w1 .
    ?f2 ogc:defaultGeometry ?g2 .
    ?g2 ogc:asWKT ?w2
  FILTER (ogcf:relate(?w1, ?w2, ogc:within) &&
    ogcf:relate(?w1, ?w2, ogc:contains)) }

```

Such cases are application-specific data modeling errors and are therefore outside of the scope of the GeoSPARQL specification.

## 1.4.2 Standard Properties for `ogc:Geometry`

### 1.4.2.1 Serialization Properties

Within each GeoSPARQL conformance class, properties are defined to link a geometry with its serialization. The WKT conformance class defines the `ogc:asWKT` property and the GML conformance class defines the `ogc:asGML` property.

#### 1.4.2.1.1 Property: `ogc:asWKT`

**Req 10 Implementations claiming conformance with the WKT conformance class shall allow the RDF property `ogc:asWKT` to be used directly in GeoSPARQL triple patterns to retrieve an `ogc:WKTLiteral` for elements of type `ogc:Geometry`.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsWKTLiteral>

The property `ogc:asWKT` is used to link a geometric element with its WKT serialization.

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#asWKT">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>asWKT</rdfs:label>
  <rdfs:comment>
    The WKT serialization of a geometry
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#WKTLiteral"/>
</rdf:Description>
```

The remainder of the standard set of properties for `ogc:Geometry` were originally defined in ISO 19107, and used in Simple Features and other OGC and ISO standards.

#### 1.4.2.1.2 Property: `ogc:asGML`

**Req 11 Implementations claiming conformance with the GML conformance class shall allow the RDF property `ogc:asGML` to be used directly in GeoSPARQL triple patterns to retrieve an `ogc:GMLLiteral` for elements of type `ogc:Geometry`.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsGMLLiteral>

The property `ogc:asGML` is used to link a geometric element with its GML serialization.

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#asGML">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>asGML</rdfs:label>
  <rdfs:comment>
    The GML serialization of a geometry
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#GMLLiteral"/>
</rdf:Description>
```

#### 1.4.2.2 Property: `ogc:dimension`

**Req 12 Compliant implementations shall support an `ogc:dimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoDim>

The dimension is the topological dimension of this geometric object, which must be less than or equal to the coordinate dimension. In non-homogeneous collections, this will return the largest topological dimension of the contained objects.

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#dimension">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>dimension</rdfs:label>
  <rdfs:comment>
    The topological dimension of this geometric object, which must
    be
    less than or equal to the coordinate dimension. In non-
    homogeneous
    collections, this will return the largest topological dimension
    of
    the contained objects.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#integer"/>
</rdf:Description>
```

### 1.4.2.3 Property: `ogc:coordinateDimension`

<p><b>Req 13 Compliant implementations shall support an <code>ogc:coordinateDimension</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.</b></p> <p><a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoCoordDim">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoCoordDim</a></p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The coordinate dimension is dimension of direct position (coordinate tuples) used in the definition of this geometric object.

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#coordinateDimension">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>coordinateDimension</rdfs:label>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#integer"/>
</rdf:Description>
```

### 1.4.2.4 Property: `ogc:spatialDimension`

<p><b>Req 14 Compliant implementations shall support an <code>ogc:spatialDimension</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.</b></p> <p><a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoSpatDim">http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoSpatDim</a></p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The spatial dimension is dimension of the spatial portion of the direct positions (coordinate tuples) used in the definition of this geometric object. If the direct positions do not carry a measure coordinate, this will be equal to the coordinate position.

```
<rdf:Description rdf:about=
  "http://www.opengis.net/rdf#spatialDimension">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>spatialDimension</rdfs:label>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#integer"/>
</rdf:Description>
```

#### 1.4.2.5 Property: `ogc:isEmpty`

<p><b>Req 15 Compliant implementations shall support an <code>ogc:isEmpty</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.</b></p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIsEmpty>

The `isEmpty` Boolean will be set to TRUE, only if the geometry contains no points.

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#isEmpty">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>isEmpty</rdfs:label>
  <rdfs:comment>
    (TRUE) if this geometric object is the empty Geometry. If true,
    then this geometric object represents the empty point set for
    the coordinate space.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#boolean"/>
</rdf:Description>
```

#### 1.4.2.6 Property: `ogc:isSimple`

<p><b>Req 16 Compliant implementations shall support an <code>ogc:isSimple</code> property for all elements of type <code>ogc:Geometry</code> consistent with its definition in Simple Features.</b></p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIsSimple>

The `isSimple` Boolean will be set to TRUE, only if the geometry contains no self-intersections, with the possible exception of its boundary.

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#isSimple">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
```

```

    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
<rdfs:label>isSimple</rdfs:label>
<rdfs:comment>
    (TRUE) if this geometric object has no anomalous geometric
    points, such as self intersection or self tangency.
</rdfs:comment>
<rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
<rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#boolean"/>
</rdf:Description>

```

### 1.4.2.7 Property: `ogc:is3D`

**Req 17 Compliant implementations shall support an `ogc:is3D` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIs3D>

The `is3D` Boolean will be set to TRUE, only if the geometry uses three spatial dimensions. This should be consistent with the `srid`, which would have to reference three spatial dimensions.

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#is3D">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>is3D</rdfs:label>
  <rdfs:comment>
    (TRUE) if this geometric object has z coordinate
    values.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#Geometry"/>
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema#boolean"/>
</rdf:Description>

```

### 1.4.3 Standard Properties for Spatial Relations

The following binary spatial relations are adopted from Simple Features. This specification generalizes these relations so that they may connect features as well as geometries.

#### **Relationship with Egenhofer Spatial Relations and RCC8 Spatial Relations:**

The Region Connection Calculus (RCC-8), see b.) and e.), and the original Egenhofer operators, see c.) and d.), both define the same set of eight operators based on the distinct operations on Regions in 2-space. Table 1 summarizes the equivalent Egenhofer and RCC8 spatial relations.

**Table 1: Egenhofer and RCC8 Spatial Relations**

RCC Short Name	RCC Long Name	Egenhofer Name
DC	disconnected	disjoint
EC	externally connected	meet
PO	partially overlapping	overlap
TPP	tangential proper part	coveredBy
TPPi	tangential proper part inverse	covers
NTPP	non-tangential proper part	inside
NTPPi	non-tangential proper part inverse	contains
EQ	equal	equal

Table 2 summarizes the equivalences between Egenhofer, RCC8 and GeoSPARQL spatial relations.

**Table 2: Equivalent GeoSPARQL, RCC8 and Egenhofer Relations**

GeoSPARQL	RCC8	Egenhofer
ogc:equals	EQ	equal
ogc:disjoint	DC	disjoint
ogc:intersects	¬ DC	¬ disjoint
ogc:touches	EC	meet
ogc:within	NTPP + TPP	inside + coveredBy
ogc:contains	TPPi + NTPPi	contains + covers
ogc:overlaps	PO	overlap

**Req 18** Compliant implementations shall allow the following RDF properties to be used directly in SPARQL triple patterns to test for existence of binary spatial relations between instances of `ogc:SpatialObject`. The named operators shall be consistent with those in Simple Features. The RCC operators shall be valid for 2D topological dimension geometric objects (Regions).<sup>[3]</sup>

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialRelations>

### 1.4.3.1 Property: ogc:equals

The Boolean predicate `ogc:equals` is defined by the following:

```
<rdf:Description rdf:about="http://www.opengis.net/rdf#equals">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>equals</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject spatially equals the
    object ogc:SpatialObject.
```

```

</rdfs:comment>
<rdfs:domain rdf:resource=
  "http://www.opengis.net/rdf#SpatialObject"/>
<rdfs:range rdf:resource=
  "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.2 Property: ogc:disjoint

The Boolean predicate `ogc:disjoint` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#disjoint">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>disjoint</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject is spatially disjoint
    from the object ogc:SpatialObject.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.3 Property: ogc:intersects

The Boolean predicate `ogc:intersects` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#intersects">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>intersects</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject spatially intersects
    the object ogc:SpatialObject.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.4 Property: ogc:touches

The Boolean predicate `ogc:touches` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#touches">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>

```

```

<rdfs:label>touches</rdfs:label>
<rdfs:comment>
  Exists if the subject ogc:SpatialObject spatially touches the
  object ogc:SpatialObject.
</rdfs:comment>
<rdfs:domain rdf:resource=
  "http://www.opengis.net/rdf#SpatialObject"/>
<rdfs:range rdf:resource=
  "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.5 Property: ogc:crosses

The Boolean predicate `ogc:crosses` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#crosses">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>crosses</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject spatially crosses the
    object ogc:SpatialObject.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.6 Property: ogc:within

The Boolean predicate `ogc:within` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#within">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>within</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject is spatially within
    the object ogc:SpatialObject.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.7 Property: ogc:contains

The Boolean predicate `ogc:contains` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#contains">
  <rdf:type rdf:resource=

```

```

    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>contains</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject spatially contains
    the object ogc:SpatialObject.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

```

### 1.4.3.8 Property: ogc:overlaps

The Boolean predicate `ogc:overlaps` is defined by the following:

```

<rdf:Description rdf:about="http://www.opengis.net/rdf#overlaps">
  <rdf:type rdf:resource=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdf:type rdf:resource=
    "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:isDefinedBy rdf:resource="http://www.opengis.net/rdf"/>
  <rdfs:label>overlaps</rdfs:label>
  <rdfs:comment>
    Exists if the subject ogc:SpatialObject spatially overlaps
    the object ogc:SpatialObject.
  </rdfs:comment>
  <rdfs:domain rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
  <rdfs:range rdf:resource=
    "http://www.opengis.net/rdf#SpatialObject"/>
</rdf:Description>

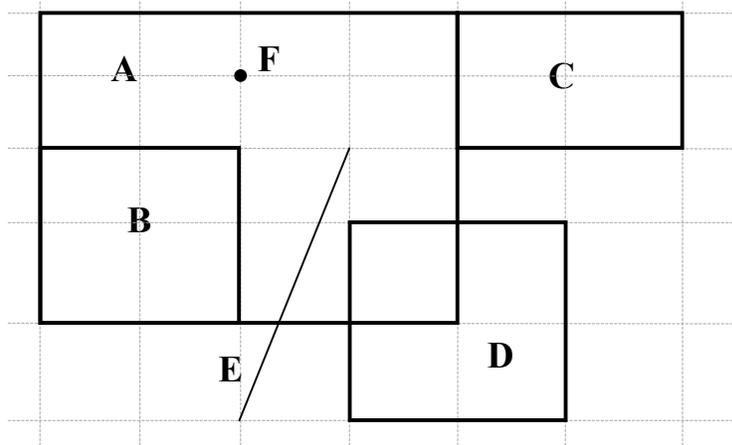
```

## 1.5 Illustrative Example

This Section illustrates the GeoSPARQL vocabulary with example RDF data and simple GeoSPARQL queries. This sample data will serve as a running example throughout the rest of the document. All examples use the WKT conformance class.

### 1.5.1 Example RDF Data

The following RDF/XML data encodes application-specific spatial data. The resulting spatial data is illustrated in Figure 2. The RDF statements define the feature class `my:PlaceOfInterest`, and two properties are created for associating geometries with features: `my:hasExactGeometry` and `my:hasPointGeometry`. `my:hasExactGeometry` is designated as the default geometry for the `my:PlaceOfInterest` feature class.



**Figure 2: Illustration of spatial data**

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:ogc="http://www.opengis.net/rdf#"
  xmlns:my="http://somewhere/ApplicationSchema#">

  <!-- Integration with GeoSPARQL classes and properties -->
  <rdfs:Class rdf:about=
    "http://somewhere/ApplicationSchema#PlaceOfInterest">
    <rdfs:subClassOf rdf:resource="http://www.opengis.net/rdf#Feature"/>
  </rdfs:Class>
  <rdf:Property rdf:about=
    "http://somewhere/ApplicationSchema#hasExactGeometry">
    <rdfs:subPropertyOf rdf:resource=
      "http://somewhere/ApplicationSchema#hasGeometry"/>
    <rdfs:subPropertyOf rdf:resource=
      "http://somewhere/ApplicationSchema#defaultGeometry"/>
  </rdf:Property>
  <rdf:Property rdf:about=
    "http://somewhere/ApplicationSchema#hasPointGeometry">
    <rdfs:subPropertyOf rdf:resource=
      "http://somewhere/ApplicationSchema#hasGeometry"/>
  </rdf:Property>

  <!-- Instance-level statements -->
  <!-- A -->
  <my:PlaceOfInterest rdf:about="http://somewhere/ApplicationSchema#A">
    <my:hasExactGeometry rdf:resource=
      "http://somewhere/ApplicationSchema#AExactGeom"/>
    <my:hasPointGeometry rdf:resource=
      "http://somewhere/ApplicationSchema#APointGeom"/>
  </my:PlaceOfInterest>
  <ogc:Polygon rdf:about="http://somewhere/ApplicationSchema#AExactGeom">
    <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
      <http://www.opengis.net/def/crs/OGC/1.3/CRS84>

```

## OGC 09-157r4

```
        Polygon((-83.6 34.1, -83.6 34.5, -83.2 34.5,
                -83.2 34.1, -83.6 34.1))
    </ogc:asWKT>
</ogc:Polygon>
<ogc:Point rdf:about="http://somewhere/ApplicationSchema#APointGeom">
    <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
        <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
        Point(-83.4 34.3)
    </ogc:asWKT>
</ogc:Point>

<!-- B -->
<my:PlaceOfInterest rdf:about="http://somewhere/ApplicationSchema#B">
    <my:hasExactGeometry rdf:resource=
        "http://somewhere/ApplicationSchema#BExactGeom"/>
    <my:hasPointGeometry rdf:resource=
        "http://somewhere/ApplicationSchema#BPointGeom"/>
</my:PlaceOfInterest>
<ogc:Polygon rdf:about="http://somewhere/ApplicationSchema#BExactGeom">
    <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
        <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
        Polygon((-83.6 34.1, -83.6 34.3, -83.4 34.3,
                -83.4 34.1, -83.6 34.1))
    </ogc:asWKT>
</ogc:Polygon>
<ogc:Point rdf:about="http://somewhere/ApplicationSchema#BPointGeom">
    <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
        <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
        Point(-83.5 34.2)
    </ogc:asWKT>
</ogc:Point>

<!-- C -->
<my:PlaceOfInterest rdf:about="http://somewhere/ApplicationSchema#C">
    <my:hasExactGeometry rdf:resource=
        "http://somewhere/ApplicationSchema#CExactGeom"/>
    <my:hasPointGeometry rdf:resource=
        "http://somewhere/ApplicationSchema#CPointGeom"/>
</my:PlaceOfInterest>
<ogc:Polygon rdf:about="http://somewhere/ApplicationSchema#CExactGeom">
    <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
        <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
        Polygon((-83.2 34.3, -83.2 34.5, -83.0 34.5,
                -83.0 34.3, -83.2 34.3))
    </ogc:asWKT>
</ogc:Polygon>
<ogc:Point rdf:about="http://somewhere/ApplicationSchema#CPointGeom">
    <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
        <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
        Point(-83.1 34.4)
    </ogc:asWKT>
</ogc:Point>
```

```

<!-- D -->
<my:PlaceOfInterest rdf:about="http://somewhere/ApplicationSchema#D">
  <my:hasExactGeometry rdf:resource=
    "http://somewhere/ApplicationSchema#DExactGeom"/>
  <my:hasPointGeometry rdf:resource=
    "http://somewhere/ApplicationSchema#DPointGeom"/>
</my:PlaceOfInterest>
<ogc:Polygon rdf:about="http://somewhere/ApplicationSchema#DExactGeom">
  <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
    <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    Polygon((-83.3 34.0, -83.3 34.2, -83.1 34.2,
      -83.1 34.0, -83.3 34.0))
  </ogc:asWKT>
</ogc:Polygon>
<ogc:Point rdf:about="http://somewhere/ApplicationSchema#DPointGeom">
  <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
    <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    Point(-83.2 34.1)
  </ogc:asWKT>
</ogc:Point>

<!-- E -->
<my:PlaceOfInterest rdf:about="http://somewhere/ApplicationSchema#E">
  <my:hasExactGeometry rdf:resource=
    "http://somewhere/ApplicationSchema#EExactGeom"/>
</my:PlaceOfInterest>
<ogc:LineString rdf:about=
  "http://somewhere/ApplicationSchema#EExactGeom">
  <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
    <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    LineString((-83.4 34.0, -83.3 34.3))
  </ogc:asWKT>
</ogc:LineString>

<!-- F -->
<my:PlaceOfInterest rdf:about="http://somewhere/ApplicationSchema#F">
  <my:hasExactGeometry rdf:resource=
    "http://somewhere/ApplicationSchema#FExactGeom"/>
</my:PlaceOfInterest>
<ogc:Point rdf:about="http://somewhere/ApplicationSchema#FExactGeom">
  <ogc:asWKT rdf:datatype="http://www.opengis.net/rdf#WKTLiteral">
    <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    Point((-84.4 34.0))
  </ogc:asWKT>
</ogc:Point>

</rdf:RDF>

```

## 1.5.2 Example GeoSPARQL Queries

Asserted RDF triples can be queried directly in triple patterns (**note:** examples in this document assume support for RDFS entailment).

**Example 1:** *Find all ogc:Features and their default geometry serializations.*

```
SELECT ?f ?s
WHERE { ?f rdf:type ogc:Feature .
        ?f ogc:defaultGeometry ?g .
        ?g ogc:asWKT ?s }
```

**Result:**

<b>?f</b>	<b>?s</b>
my:A	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> Polygon((-83.6 34.1, -83.6 34.5, -83.2 34.5, -83.2 34.1, -83.6 34.1))"^^ogc:WKTLiteral
my:B	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> Polygon((-83.6 34.1, -83.6 34.3, -83.4 34.3, -83.4 34.1, -83.6 34.1))"^^ogc:WKTLiteral
my:C	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> Polygon((-83.2 34.3, -83.2 34.5, -83.0 34.5, -83.0 34.3, -83.2 34.3))"^^ogc:WKTLiteral
my:D	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> Polygon((-83.3 34.0, -83.3 34.2, -83.1 34.2, -83.1 34.0, -83.3 34.0))"^^ogc:WKTLiteral
my:E	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> LineString((-83.4 34.0, -83.3 34.3))"^^ogc:WKTLiteral
my:F	"<http://www.opengis.net/def/crs/OGC/1.3/CRS84> Point((-84.4 34.0))"^^ogc:WKTLiteral

**Domain-specific SPARQL Filter Functions**

This document defines the following SPARQL filter functions for expressing spatial queries. These functions can be used in the FILTER clause of a SPARQL query.

**Req 19 Compliant implementations shall support the SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary, consistent with Simple Features.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterFunctions>

The remainder of this document uses the IRI `ogc:GeomLiteral` as a placeholder for the conformance class-specific RDFS Datatype: either `ogc:WKTLiteral` for the WKT conformance class or `ogc:GMLLiteral` for the GML conformance class. The exact function signatures for each conformance class can be obtained by substituting the appropriate RDFS Datatype.

**Req 20 Implementations claiming conformance with the WKT GeoSPARQL conformance class shall support the use of `ogc:WKTLiteral` in place of `ogc:GeomLiteral` in SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterWKT>

**Req 21 Implementations claiming conformance with the GML GeoSPARQL conformance class shall support the use of `ogc:GMLLiteral` in place of `ogc:GeomLiteral` in SPARQL FILTER functions for relate, distance,**

<b>buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary.</b>
-----------------------------------------------------------------------------------------------------------

<a href="http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterGML">http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterGML</a>
-----------------------------------------------------------------------------------------------------------------------------------------

An invocation of any of the following FILTER functions with invalid arguments produces an error. An invalid argument includes any of the following:

- An argument of an unexpected type
- An invalid geometry literal value
- A geometry literal from a spatial reference system that is incompatible with the spatial reference system used for calculations.
- A relation URI that is not included in the set defined in Section 6.2.3
- A units URI that is not included in the set defined **[need something here]**

For further discussion of the effects of errors during FILTER evaluation, consult Section 11 of the SPARQL specification (<http://www.w3.org/TR/rdf-sparql-query/#tests>).

### 1.6 Filter Function: ogcf:relate

```
ogcf:relate (geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
            relation: xsd:anyURI): xsd:boolean
```

Returns TRUE if `geom1` is spatially related to `geom2` by the given spatial `relation`. Otherwise, this function returns FALSE. Any of the spatial predicate IRIs defined in Section 6.2.3 are valid `relation` arguments. The spatial reference system for `geom1` is used for spatial calculations.

### 1.7 Filter Function: ogcf:distance

```
ogcf:distance (geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
              units: xsd:anyURI): xsd:double
```

Returns the shortest distance in `units` between any two Points in the two geometric objects as calculated in the spatial reference system of `geom1`.

### 1.8 Filter Function: ogc:buffer

```
ogcf:buffer (geom: ogc:GeomLiteral, radius: xsd:real,
            units: xsd:anyURI): ogc:GeomLiteral
```

This function returns a geometric object that represents all Points whose distance from `geom1` is less than or equal to the `radius` measured in `units`. Calculations are in the spatial reference system of `geom1`.

### 1.9 Filter Function: ogcf:convexHull

```
ogcf:convexHull (geom1: ogc:GeomLiteral): ogc:GeomLiteral
```

This function returns a geometric object that represents all Points in the convex hull of `geom1`. Calculations are in the spatial reference system of `geom1`.

### 1.10 Filter Function: ogcf:intersection

```
ogcf:intersection (geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
                  ): ogc:GeomLiteral
```

This function returns a geometric object that represents all Points in the intersection of geom1 with geom2. Calculations are in the spatial reference system of geom1.

### 1.11 Filter Function: ogcf:union

```
ogcf:union (geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
           ): ogc:GeomLiteral
```

This function returns a geometric object that represents all Points in the union of geom1 with geom2. Calculations are in the spatial reference system of geom1.

### 1.12 Filter Function: ogcf:difference

```
ogcf:difference (geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
                ): ogc:GeomLiteral
```

This function returns a geometric object that represents all Points in the set difference of geom1 with geom2. Calculations are in the spatial reference system of geom1.

### 1.13 Filter Function: ogcf:symDifference

```
ogcf:symDifference (geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
                  ): ogc:GeomLiteral
```

This function returns a geometric object that represents all Points in the set symmetric difference of geom1 with geom2. Calculations are in the spatial reference system of geom1.

### 1.14 Filter Function: ogcf:envelope

```
ogcf:envelope (geom1: ogc:GeomLiteral): ogc:GeomLiteral
```

This function returns the minimum bounding box of geom1. Calculations are in the spatial reference system of geom1.

### 1.15 Filter Function: ogcf:boundary

```
ogcf:boundary (geom1: ogc:GeomLiteral): ogc:GeomLiteral
```

This function returns the closure of the boundary of geom1. Calculations are in the spatial reference system of geom1.

### 1.16 Example GeoSPARQL FILTER queries

This Section illustrates the use of GeoSPARQL FILTER functions through a series of example queries.

**Example 2:** *Find all features that feature my:A contains, where spatial calculations are based on my:hasExactGeometry.*

```
SELECT ?f
WHERE {
  my:A my:hasExactGeometry ?aGeom .
      ?aGeom ogc:asWKT ?aWKT .
      ?f my:hasExactGeometry ?fGeom .
      ?fGeom ogc:asWKT ?fWKT .
      FILTER (ogcf:relate(?aWKT, ?fWKT, ogc:contains))
}
```

**Result:**

?f
my:B
my:F

**Example 3:** Find all features that are within a transient bounding box geometry, where spatial calculations are based on my:hasPointGeometry.

```
SELECT ?f
WHERE {
  ?f my:hasPointGeometry ?fGeom .
  ?fGeom ogc:asWKT ?fWKT .
  FILTER (ogcf:relate(?fWKT,
    "<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    Polygon ((-83.5 34.0, -83.5 34.3, -83.1 34.3,
              -83.1 34.0, -83.5 34.0))"^^ogc:WKTLiteral,
    ogc:within))
}
```

**Result:**

?f
my:A
my:B
my:D

**Example 4:** Find all features that touch the union of feature my:A and feature my:D, where computations are based on my:hasExactGeometry.

```
SELECT ?f
WHERE {
  ?f my:hasExactGeometry ?fGeom .
  ?fGeom ogc:asWKT ?fWKT .
  my:A my:hasExactGeometry ?aGeom .
  ?aGeom ogc:asWKT ?aWKT .
  ?my:D my:hasExactGeometry ?dGeom .
  ?dGeom ogc:asWKT ?dWKT .
  FILTER (ogcf:relate(?fWKT,
    ogcf:union(?aWKT, ?dWKT),
    ogc:touches))
}
```

**Result:**

?f
my:C

**Example 5:** Find the 3 closest features to feature my:C, where computations are based on my:hasExactGeometry.

```
SELECT ?f
WHERE {
  my:C my:hasExactGeometry ?cGeom .
  ?cGeom ogc:asWKT ?cWKT .
  ?f my:hasExactGeometry ?fGeom .
  ?fGeom ogc:asWKT ?fWKT .
  FILTER (?fGeom != ?cGeom) }
ORDER BY ASC (ogcf:distance(?cWKT, ?fWKT,
  my:simpleUnits))
LIMIT 3
```

**Result:**

<b>?f</b>
my:A
my:D
my:E

## Query Transformation Rules

GeoSPARQL is designed to accommodate systems based on qualitative spatial reasoning and systems based on quantitative spatial computations. Systems based on qualitative spatial reasoning, (e.g. those based on the Region Connection Calculus b.), e.) do not usually model explicit geometries, so queries in such systems will likely test for binary spatial relationships between features rather than between explicit geometries. To allow queries for spatial relations between features in quantitative systems, GeoSPARQL defines a series of query transformation rules that expand a feature-only query into a geometry-based query. With these transformation rules, queries about spatial relations between features will have the same specification in both qualitative systems and quantitative systems. The qualitative system will likely evaluate the query with a backward-chaining spatial reasoner, and the quantitative system can transform the query into a geometry-based query that can be evaluated with computational geometry. This Section defines query transformation rules for GeoSPARQL using RIF Core syntax h.

**Req 22 Compliant implementations shall use the following rules to test spatial relations between instances of `ogc:SpatialObject`.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransRules>

The remainder of this document uses the IRI `ogc:asText` as a placeholder for the conformance class-specific properties linking geometries with their text-based serializations: either `ogc:asWKT` for the WKT conformance class or `ogc:asGML` for the GML conformance class. The exact RIF rules for each conformance class can be obtained by substituting the appropriate property.

**Req 23 Implementations claiming conformance with the WKT GeoSPARQL conformance class shall support the use of `ogc:asWKT` in place of `ogc:asText` in the following rules to test spatial relations between instances of `ogc:SpatialObject`.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransWKT>

**Req 24 Implementations claiming conformance with the GML GeoSPARQL conformance class shall support the use of `ogc:asGML` in place of `ogc:asText` in the following rules to test spatial relations between instances of `ogc:SpatialObject`.**

<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransGML>

### 1.17 Transformation Rule: `ogc>equals`<sup>[4]</sup>

```
(forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
  (f1[ogc>equals->?f2] :-
```

```

Or(
  And
    # feature - feature rule
    (?f1[ogc:defaultGeometry->?g1]
     ?f2[ogc:defaultGeometry->?g2]
     ?g1[ogc:asText->?g1Serial]
     ?g2[ogc:asText->?g2Serial]
     External(ogcf:equals(?g1Serial,?g2Serial)))
  And
    # feature - geometry rule
    (?f1[ogc:defaultGeometry->?g1]
     ?g1[ogc:asText->?g1Serial]
     ?f2[ogc:asText->?g2Serial]
     External(ogcf:equals(?g1Serial,?g2Serial)))
  And
    # geometry - feature rule
    (?f2[ogc:defaultGeometry->?g2]
     ?f1[ogc:asText->?g1Serial]
     ?g2[ogc:asText->?g2Serial]
     External(ogcf:equals(?g1Serial,?g2Serial)))
  And
    # geometry - geometry rule
    (?f1[ogc:asText->?g1Serial]
     ?f2[ogc:asText->?g2Serial]
     External(ogcf:equals(?g1Serial,?g2Serial)))
)
)
)

```

### 1.18 Transformation Rule: ogc:disjoint

```

(Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
  (f1[ogc:disjoint->?f2] :-
    Or(
      And
        # feature - feature rule
        (?f1[ogc:defaultGeometry->?g1]
         ?f2[ogc:defaultGeometry->?g2]
         ?g1[ogc:asText->?g1Serial]
         ?g2[ogc:asText->?g2Serial]
         External(ogcf:disjoint(?g1Serial,?g2Serial)))
      And
        # feature - geometry rule
        (?f1[ogc:defaultGeometry->?g1]
         ?g1[ogc:asText->?g1Serial]
         ?f2[ogc:asText->?g2Serial]
         External(ogcf:disjoint(?g1Serial,?g2Serial)))
      And
        # geometry - feature rule
        (?f2[ogc:defaultGeometry->?g2]
         ?f1[ogc:asText->?g1Serial]
         ?g2[ogc:asText->?g2Serial]
         External(ogcf:disjoint(?g1Serial,?g2Serial)))
      And
        # geometry - geometry rule
        (?f1[ogc:asText->?g1Serial]
         ?f2[ogc:asText->?g2Serial]

```

```

        External (ogcf:disjoint (?g1Serial, ?g2Serial))
    )
)
)

```

### 1.19 Transformation Rule: ogc:intersects

```

Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
(f1[ogc:intersects->?f2] :-
  Or(
    And
      # feature - feature rule
      (?f1[ogc:defaultGeometry->?g1]
       ?f2[ogc:defaultGeometry->?g2]
       ?g1[ogc:asText->?g1Serial]
       ?g2[ogc:asText->?g2Serial]
       External (ogcf:intersects (?g1Serial, ?g2Serial)))
    And
      # feature - geometry rule
      (?f1[ogc:defaultGeometry->?g1]
       ?g1[ogc:asText->?g1Serial]
       ?f2[ogc:asText->?g2Serial]
       External (ogcf:intersects (?g1Serial, ?g2Serial)))
    And
      # geometry - feature rule
      (?f2[ogc:defaultGeometry->?g2]
       ?f1[ogc:asText->?g1Serial]
       ?g2[ogc:asText->?g2Serial]
       External (ogcf:intersects (?g1Serial, ?g2Serial)))
    And
      # geometry - geometry rule
      (?f1[ogc:asText->?g1Serial]
       ?f2[ogc:asText->?g2Serial]
       External (ogcf:intersects (?g1Serial, ?g2Serial)))
  )
)
)

```

### 1.20 Transformation Rule: ogc:touches

```

Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
(f1[ogc:touches->?f2] :-
  Or(
    And
      # feature - feature rule
      (?f1[ogc:defaultGeometry->?g1]
       ?f2[ogc:defaultGeometry->?g2]
       ?g1[ogc:asText->?g1Serial]
       ?g2[ogc:asText->?g2Serial]
       External (ogcf:touches (?g1Serial, ?g2Serial)))
    And
      # feature - geometry rule
      (?f1[ogc:defaultGeometry->?g1]
       ?g1[ogc:asText->?g1Serial]
       ?f2[ogc:asText->?g2Serial]
       External (ogcf:touches (?g1Serial, ?g2Serial)))
    And

```

```

    # geometry - feature rule
    (?f2[ogc:defaultGeometry->?g2]
     ?f1[ogc:asText->?g1Serial]
     ?g2[ogc:asText->?g2Serial]
     External(ogcf:touches(?g1Serial,?g2Serial)))
  And
  # geometry - geometry rule
  (?f1[ogc:asText->?g1Serial]
   ?f2[ogc:asText->?g2Serial]
   External(ogcf:touches(?g1Serial,?g2Serial)))
)
)
)

```

## 1.21 Transformation Rule: ogc:crosses

```

Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
(f1[ogc:crosses->?f2] :-
  Or(
    And
      # feature - feature rule
      (?f1[ogc:defaultGeometry->?g1]
       ?f2[ogc:defaultGeometry->?g2]
       ?g1[ogc:asText->?g1Serial]
       ?g2[ogc:asText->?g2Serial]
       External(ogcf:crosses(?g1Serial,?g2Serial)))
    And
      # feature - geometry rule
      (?f1[ogc:defaultGeometry->?g1]
       ?g1[ogc:asText->?g1Serial]
       ?f2[ogc:asText->?g2Serial]
       External(ogcf:crosses(?g1Serial,?g2Serial)))
    And
      # geometry - feature rule
      (?f2[ogc:defaultGeometry->?g2]
       ?f1[ogc:asText->?g1Serial]
       ?g2[ogc:asText->?g2Serial]
       External(ogcf:crosses(?g1Serial,?g2Serial)))
    And
      # geometry - geometry rule
      (?f1[ogc:asText->?g1Serial]
       ?f2[ogc:asText->?g2Serial]
       External(ogcf:crosses(?g1Serial,?g2Serial)))
  )
)
)
)

```

## 1.22 Transformation Rule: ogc:within

```

Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
(f1[ogc:within->?f2] :-
  Or(
    And
      # feature - feature rule
      (?f1[ogc:defaultGeometry->?g1]
       ?f2[ogc:defaultGeometry->?g2]
       ?g1[ogc:asText->?g1Serial]

```

```

        ?g2[ogc:asText->?g2Serial]
        External(ogcf:within(?g1Serial,?g2Serial))
    And
        # feature - geometry rule
        (?f1[ogc:defaultGeometry->?g1]
        ?g1[ogc:asText->?g1Serial]
        ?f2[ogc:asText->?g2Serial]
        External(ogcf:within(?g1Serial,?g2Serial)))
    And
        # geometry - feature rule
        (?f2[ogc:defaultGeometry->?g2]
        ?f1[ogc:asText->?g1Serial]
        ?g2[ogc:asText->?g2Serial]
        External(ogcf:within(?g1Serial,?g2Serial)))
    And
        # geometry - geometry rule
        (?f1[ogc:asText->?g1Serial]
        ?f2[ogc:asText->?g2Serial]
        External(ogcf:within(?g1Serial,?g2Serial)))
    )
)
)
)

```

### 1.23 Transformation Rule: ogc:contains

```

Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
(f1[ogc:contains->?f2] :-
    Or(
        And
            # feature - feature rule
            (?f1[ogc:defaultGeometry->?g1]
            ?f2[ogc:defaultGeometry->?g2]
            ?g1[ogc:asText->?g1Serial]
            ?g2[ogc:asText->?g2Serial]
            External(ogcf:contains(?g1Serial,?g2Serial)))
        And
            # feature - geometry rule
            (?f1[ogc:defaultGeometry->?g1]
            ?g1[ogc:asText->?g1Serial]
            ?f2[ogc:asText->?g2Serial]
            External(ogcf:contains(?g1Serial,?g2Serial)))
        And
            # geometry - feature rule
            (?f2[ogc:defaultGeometry->?g2]
            ?f1[ogc:asText->?g1Serial]
            ?g2[ogc:asText->?g2Serial]
            External(ogcf:contains(?g1Serial,?g2Serial)))
        And
            # geometry - geometry rule
            (?f1[ogc:asText->?g1Serial]
            ?f2[ogc:asText->?g2Serial]
            External(ogcf:contains(?g1Serial,?g2Serial)))
    )
)
)
)

```

## 1.24 Transformation Rule: ogc:overlaps

```

Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
  (f1[ogc:overlaps->?f2] :-
    Or(
      And
        # feature - feature rule
        (?f1[ogc:defaultGeometry->?g1]
         ?f2[ogc:defaultGeometry->?g2]
         ?g1[ogc:asText->?g1Serial]
         ?g2[ogc:asText->?g2Serial]
         External(ogcf:overlaps(?g1Serial,?g2Serial)))
      And
        # feature - geometry rule
        (?f1[ogc:defaultGeometry->?g1]
         ?g1[ogc:asText->?g1Serial]
         ?f2[ogc:asText->?g2Serial]
         External(ogcf:overlaps(?g1Serial,?g2Serial)))
      And
        # geometry - feature rule
        (?f2[ogc:defaultGeometry->?g2]
         ?f1[ogc:asText->?g1Serial]
         ?g2[ogc:asText->?g2Serial]
         External(ogcf:overlaps(?g1Serial,?g2Serial)))
      And
        # geometry - geometry rule
        (?f1[ogc:asText->?g1Serial]
         ?f2[ogc:asText->?g2Serial]
         External(ogcf:overlaps(?g1Serial,?g2Serial)))
    )
  )
)

```

## 1.25 Example Query Rewrite

This section illustrates the application of a query rewrite rule to a GeoSPARQL query.

**Example 6:** *Find all features or geometries that overlap feature my:A.*

### Original Query:

```

SELECT ?f
WHERE { ?f ogc:overlaps my:A }

```

### Transformed Query (application of transformation rule ogc:overlaps):

```

SELECT ?f
WHERE { { # check for asserted statement
         ?f ogc:overlaps my:A }
        UNION
        { # feature - feature
          ?f ogc:defaultGeometry ?fGeom .
          ?fGeom ogc:asWKT ?fSerial .
          my:A ogc:defaultGeometry ?aGeom .
          ?aGeom ogc:asWKT ?aSerial .
          FILTER (ogc:overlaps(?fSerial, ?aSerial)) }
        UNION
        { # feature - geometry
          ?f ogc:defaultGeometry ?fGeom .

```

```

    ?fGeom ogc:asWKT ?fSerial .
    my:A ogc:asWKT ?aSerial .
    FILTER (ogc:overlaps(?fSerial, ?aSerial)) }
UNION
{ # geometry - feature
  ?f ogc:asWKT ?fSerial .
  my:A ogc:defaultGeometry ?aGeom .
  ?aGeom ogc:asWKT ?aSerial .
  FILTER (ogc:overlaps(?fSerial, ?aSerial)) }
UNION
{ # geometry - geometry
  ?f ogc:defaultGeometry ?fGeom .
  ?fGeom ogc:asWKT ?fSerial .
  my:A ogc:defaultGeometry ?aGeom .
  ?aGeom ogc:asWKT ?aSerial .
  FILTER (ogc:overlaps(?fSerial, ?aSerial)) }
}

```

**Result:**

<b>?f</b>
my:D
my:DExactGeom
my:E
my:EExactGeom

## OWL-based Spatial Reasoning

It is possible to create OWL *j* axioms to aid in reasoning based on asserted spatial relations (e.g. asserting that `ogc:contains` is an `owl:TransitiveProperty`). The specification of such axioms is outside the scope of this document. However, GeoSPARQL has been designed so that queries for asserted and inferred spatial relations between features are possible. That is, one can query for spatial relations without directly specifying geometry information in the query. GeoSPARQL should therefore accommodate systems that completely rely on logical spatial reasoning and have no notion of quantitative geometries. For example, the query in Example 6 could be evaluated through logical reasoning or through query transformation based on the rules defined in Section 8.

## Future Work

Many future extensions of this standard are possible. Obvious extensions are to define new conformance classes for other standard serializations of geometry data (e.g. KML, GeoJSON). In addition, significant work remains in developing vocabularies for spatial data, and expanding the GeoSPARQL vocabulary with OWL axioms to aid in logical spatial reasoning would be a valuable contribution. There are also large amounts of existing feature data represented either in a GML file (or similar serialization) or in a

datastore supporting the general feature model. It would be beneficial to develop standard processes for converting (or virtually converting and exposing) this data to RDF.

## Abstract Test Suite

### A.1 Abstract Test Suite for GeoSPARQL

#### A.1.1 WKT Geometry Serialization Conformance Class

##### A.1.1.1 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialObjectClass>

All feature and geometry objects shall subtype either directly or through intermediaries from the root class `ogc:SpatialObject`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: ???<sup>[5]</sup>
- c.) Reference: Clause 1.3.1 Req 1
- d.) Test Type: Capabilities

##### A.1.1.2 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureClass>

All feature objects shall subtype either directly or through intermediaries from the class `ogc:Feature`

- a.) Test purpose: check conformance with this requirement
- b.) Test method: ???<sup>[6]</sup>
- c.) Reference: Clause 1.3.2 Req 2
- d.) Test Type: Capabilities

##### A.1.1.3 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryClass>

All geometry objects shall subtype either directly or through intermediaries from the class `ogc:Geometry`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: ???<sup>[7]</sup>
- c.) Reference: Clause 1.3.3 Req 3
- d.) Test Type: Capabilities

##### A.1.1.4 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteral>

All RDFS Literals of type `ogc:WKTLiteral` shall consist of an IRI identifying the coordinate reference systems followed by Simple Features Well Known Text (WKT) describing a geometric value.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query with a filter condition involving an invalid `ogc:WKTLiteral` returns no results.

- c.) Reference: Clause 1.3.5.1 Req 4
- d.) Test Type: Capabilities

**A.1.1.5**      **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/WKTLiteralEmpty>**

An RDFS Literal of type `ogc:WKTLiteral` that represents an empty geometry shall use an empty string for its lexical value.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query with an empty `ogc:WKTLiteral` returns the correct result.
- c.) Reference: Clause 1.3.5.1 Req 5
- d.) Test Type: Capabilities

**A.1.1.6**      **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureProperties>**

Compliant implementations shall allow the RDF properties `ogc:hasGeometry` and `ogc:defaultGeometry` to be used directly in SPARQL triple patterns to retrieve information about instances of `ogc:Feature`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving these properties returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.1 Req 9
- d.) Test Type: Capabilities

**A.1.1.7**      **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsWKTLiteral>**

Implementations claiming conformance with the WKT conformance class shall allow the RDF property `ogc:asWKT` to be used directly in GeoSPARQL triple patterns to retrieve an `ogc:WKTLiteral` for elements of type `ogc:Geometry`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.1.1 Req 10
- d.) Test Type: Capabilities

**A.1.1.8**      **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoDimension>**

Compliant implementations shall support an `ogc:dimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.2 Req 12
- d.) Test Type: Capabilities

### **A.1.1.9 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoCoordDim>**

Compliant implementations shall support an `ogc:coordinateDimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.3 Req 13
- d.) Test Type: Capabilities

### **A.1.1.10 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoSpatialDim>**

Compliant implementations shall support an `ogc:spatialDimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.4 Req 14
- d.) Test Type: Capabilities

### **A.1.1.11 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIsEmpty>**

Compliant implementations shall support an `ogc:isEmpty` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.5 Req 15
- d.) Test Type: Capabilities

### **A.1.1.12 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIsSimple>**

Compliant implementations shall support an `ogc:isSimple` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.6 Req 16
- d.) Test Type: Capabilities

### **A.1.1.13 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIs3D>**

Compliant implementations shall support an `ogc:is3D` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.7 Req 17
- d.) Test Type: Capabilities

#### **A.1.1.14 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialRelations>**

Compliant implementations shall allow the following RDF properties to be used directly in SPARQL triple patterns to test for existence of binary spatial relations between instances of `ogc:SpatialObject`. The named operators shall be consistent with those in Simple Features. The RCC operators shall be valid for 2D topological dimension geometric objects (Regions).

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving each of the following properties returns the correct result for a test dataset: `ogc:equals`, `ogc:disjoint`, `ogc:intersects`, `ogc:touches`, `ogc:crosses`, `ogc:within`, `ogc:contains`, `ogc:overlaps`.
- c.) Reference: Clause 1.4.3 Req 18
- d.) Test Type: Capabilities

#### **A.1.1.15 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterFunctions>**

Compliant implementations shall support the SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary, consistent with Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following filter functions return the correct result for a test dataset when using `ogc:WKTLiteral` geometry arguments: `ogcf:relate`, `ogcf:distance`, `ogcf:buffer`, `ogcf:convexHull`, `ogcf:intersection`, `ogcf:union`, `ogcf:difference`, `ogcf:symDifference`, `ogcf:envelope`, `ogcf:boundary`.
- c.) Reference: Clause 0 Req 19
- d.) Test Type: Capabilities

#### **A.1.1.16 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterWKT>**

Implementations claiming conformance with the WKT GeoSPARQL conformance class shall support the use of `ogc:WKTLiteral` in place of `ogc:GeomLiteral` in SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following filter functions return the correct result for a test dataset when using `ogc:WKTLiteral` geometry arguments: `ogcf:relate`, `ogcf:distance`, `ogcf:buffer`, `ogcf:convexHull`, `ogcf:intersection`, `ogcf:union`, `ogcf:difference`, `ogcf:symDifference`, `ogcf:envelope`, `ogcf:boundary`.
- c.) Reference: Clause 0 Req 20
- d.) Test Type: Capabilities

### **A.1.1.17 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransformationRules>**

Compliant implementations shall use the following rules to test spatial relations between instances of `ogc:SpatialObject`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following query transformation rules return the correct result for a test dataset when using `ogc:asWKT` as the geometry serialization property: `ogc:equals`, `ogc:disjoint`, `ogc:intersects`, `ogc:touches`, `ogc:crosses`, `ogc:within`, `ogc:contains`, `ogc:overlaps`.
- c.) Reference: Clause 0 Req 22
- d.) Test Type: Capabilities

### **A.1.1.18 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransformationWKT>**

Implementations claiming conformance with the WKT GeoSPARQL conformance class shall support the use of `ogc:asWKT` in place of `ogc:asText` in the following rules to test spatial relations between instances of `ogc:SpatialObject`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following query transformation rules return the correct result for a test dataset when using `ogc:asWKT` as the geometry serialization property: `ogc:equals`, `ogc:disjoint`, `ogc:intersects`, `ogc:touches`, `ogc:crosses`, `ogc:within`, `ogc:contains`, `ogc:overlaps`.
- c.) Reference: Clause 0 Req 23
- d.) Test Type: Capabilities

## **A.1.2 GML Geometry Serialization Conformance Class**

### **A.1.2.1 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialObjectClass>**

All feature and geometry objects shall subtype either directly or through intermediaries from the root class `ogc:SpatialObject`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: ???[8]
- c.) Reference: Clause 1.3.1 Req 1
- d.) Test Type: Capabilities

### **A.1.2.2 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureClass>**

All feature objects shall subtype either directly or through intermediaries from the class `ogc:Feature`

- a.) Test purpose: check conformance with this requirement
- b.) Test method: ???[9]
- c.) Reference: Clause 1.3.2 Req 2
- d.) Test Type: Capabilities

**A.1.2.3 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryClass>**

All geometry objects shall subtype either directly or through intermediaries from the class `ogc:Geometry`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: ???<sup>[10]</sup>
- c.) Reference: Clause 1.3.3 Req 3
- d.) Test Type: Capabilities

**A.1.2.4 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteral>**

All RDFS Literals of type `ogc:GMLLiteral` shall consist of a valid element from the GML schema that implements a subtype of `GM_Object`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query with a filter condition involving an invalid `ogc:GMLLiteral` returns no results
- c.) Reference: Clause 1.3.5.2 Req 6
- d.) Test Type: Capabilities

**A.1.2.5 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLLiteralEmpty>**

An RDFS Literal of type `ogc:GMLLiteral` that represents an empty geometry shall use an empty string for its lexical value.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving an empty `ogc:GMLLiteral` returns the correct result for a test dataset.
- c.) Reference: Clause 1.3.5.2 Req 7
- d.) Test Type: Capabilities

**A.1.2.6 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GMLProfile>**

Implementations claiming conformance to the GML GeoSPARQL conformance class shall document the GML profile supported by the implementation.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that an implementation specifies the GML Profile supported.
- c.) Reference: Clause 1.3.5.2 Req 8
- d.) Test Type: Capabilities

**A.1.2.7 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/FeatureProperties>**

Compliant implementations shall allow the RDF properties `ogc:hasGeometry` and `ogc:defaultGeometry` to be used directly in SPARQL triple patterns to retrieve information about instances of `ogc:Feature`.

- a.) Test purpose: check conformance with this requirement

- b.) Test method: Verify that a query involving these properties returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.1 Req 9
- d.) Test Type: Capabilities

**A.1.2.8**      **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeometryAsGMLLiteral>**

Implementations claiming conformance with the GML conformance class shall allow the RDF property `ogc:asGML` to be used directly in GeoSPARQL triple patterns to retrieve an `ogc:GMLLiteral` for elements of type `ogc:Geometry`.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.1.2 Req 11
- d.) Test Type: Capabilities

**A.1.2.9**      **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoDim>**

Compliant implementations shall support an `ogc:dimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.2 Req 12
- d.) Test Type: Capabilities

**A.1.2.10**     **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoCoordDim>**

Compliant implementations shall support an `ogc:coordinateDimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.3 Req 13
- d.) Test Type: Capabilities

**A.1.2.11**     **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoSpatialDim>**

Compliant implementations shall support an `ogc:spatialDimension` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.4 Req 14
- d.) Test Type: Capabilities

### **A.1.2.12 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIsEmpty>**

Compliant implementations shall support an `ogc:isEmpty` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.5 Req 15
- d.) Test Type: Capabilities

### **A.1.2.13 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIsSimple>**

Compliant implementations shall support an `ogc:isSimple` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.6 Req 16
- d.) Test Type: Capabilities

### **A.1.2.14 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/GeoIs3D>**

Compliant implementations shall support an `ogc:is3D` property for all elements of type `ogc:Geometry` consistent with its definition in Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving this property returns the correct result for a test dataset.
- c.) Reference: Clause 1.4.2.7 Req 17
- d.) Test Type: Capabilities

### **A.1.2.15 <http://www.opengis.net/spec/GeoSPARQL/1.0/req/SpatialRelations>**

Compliant implementations shall allow the following RDF properties to be used directly in SPARQL triple patterns to test for existence of binary spatial relations between instances of `ogc:SpatialObject`. The named operators shall be consistent with those in Simple Features. The RCC operators shall be valid for 2D topological dimension geometric objects (Regions).

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that a query involving each of the following properties returns the correct result for a test dataset: `ogc:equals`, `ogc:disjoint`, `ogc:intersects`, `ogc:touches`, `ogc:crosses`, `ogc:within`, `ogc:contains`, `ogc:overlaps`.
- c.) Reference: Clause 1.4.3 Req 18
- d.) Test Type: Capabilities

**A.1.2.16**     **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterFunctions>**

Compliant implementations shall support the SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary, consistent with Simple Features.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following filter functions return the correct result for a test dataset when using ogc:GMLLiteral geometry arguments: ogcf:relate, ogcf:distance, ogcf:buffer, ogcf:convexHull, ogcf:intersection, ogcf:union, ogcf:difference, ogcf:symDifference, ogcf:envelope, ogcf:boundary.
- c.) Reference: Clause 0 Req 19
- d.) Test Type: Capabilities

**A.1.2.17**     **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryFilterGML>**

Implementations claiming conformance with the GML GeoSPARQL conformance class shall support the use of ogc:GMLLiteral in place of ogc:GeomLiteral in SPARQL FILTER functions for relate, distance, buffer, convex hull, intersection, union, difference, symmetric difference, envelope and boundary.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following filter functions return the correct result for a test dataset when using ogc:GMLLiteral geometry arguments: ogcf:relate, ogcf:distance, ogcf:buffer, ogcf:convexHull, ogcf:intersection, ogcf:union, ogcf:difference, ogcf:symDifference, ogcf:envelope, ogcf:boundary.
- c.) Reference: Clause 0 Req 21
- d.) Test Type: Capabilities

**A.1.2.18**     **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransformationRules>**

Compliant implementations shall use the following rules to test spatial relations between instances of ogc:SpatialObject.

- a.) Test purpose: check conformance with this requirement
- b.) Test method: Verify that queries involving the following query transformation rules return the correct result for a test dataset when using ogc:asGML as the geometry serialization property: ogc>equals, ogc:disjoint, ogc:intersects, ogc:touches, ogc:crosses, ogc:within, ogc:contains, ogc:overlaps.
- c.) Reference: Clause 0 Req 22
- d.) Test Type: Capabilities

**A.1.2.19**     **<http://www.opengis.net/spec/GeoSPARQL/1.0/req/QueryTransformationGML>**

Implementations claiming conformance with the GML GeoSPARQL conformance class shall support the use of ogc:asGML in place of ogc:asText in the following rules to test spatial relations between instances of ogc:SpatialObject.

- a.) Test purpose: check conformance with this requirement

- b.) Test method: Verify that queries involving the following query transformation rules return the correct result for a test dataset when using `ogc:asGML` as the geometry serialization property: `ogc:equals`, `ogc:disjoint`, `ogc:intersects`, `ogc:touches`, `ogc:crosses`, `ogc:within`, `ogc:contains`, `ogc:overlaps`.
- c.) Reference: Clause 0 Req 24
- d.) Test Type: Capabilities

## Bibliography

- a.) ARQ - A SPARQL Processor for Jena, available at <http://jena.sourceforge.net/ARQ/>.
- b.) Cohn, Anthony G., Brandon Bennett, John Gooday, Nicholas Mark Gotts: **Qualitative Spatial Representation and Reasoning with the Region Connection Calculus**. *GeoInformatica*, 1, 275–316, 1997.
- c.) Egenhofer Max, **A Formal Definition of Binary Topological Relationships**, *Third International Conference on Foundations of Data Organization and Algorithms (FODO)*, Paris, France, W. Litwin and H. Schek (eds.), *Lecture Notes in Computer Science*, Vol. 367, Springer-Verlag, pp. 457-472, June 1989. available at <http://www.spatial.maine.edu/~max/RC2.html>
- d.) Egenhofer, Max and J. Herring **Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases**, Technical Report, Department of Surveying Engineering, University of Maine, 1990. (revised versions in NCGIA Technical Report 91-7 and NCGIA Technical Report 94-1)
- e.) Randell, D. A., Cui, Z. and Cohn, A. G.: **A spatial logic based on regions and connection**, Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, pp. 165–176, 1992.