

S-58 Restructure

Guidelines on the check syntax

In the revised wording of these tests the syntax is arranged so that if outcome of the expression 'yes' then an error or warning is generated.

The principal terms used within this document are listed and defined below;

Operators

Equal
Not equal
Less than
Less than or equal to
Greater than
Greater than or equal to
AND
OR

Spatial Operators

Within this document the following ISO 19125-2 spatial operators are used;

Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects, Contains, Relate

Values

The following terms are used for types of values;

- Not populated – An attribute is not present.
- Null – An attribute has a value of null (255)
- notNull – The attribute has been populated with a value.

Statements

The checks should be structured using the following statements;

- If – A conditional statement which determines whether a further statement should be executed.
- For – repeat a statement until a statement is met. For the purposes of the checks the statement being met generates the error or warning specified.
- Switch – test against a variable if this does not match move on to the next test

Equals(g1 Geometry, g2 Geometry) :

Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments.

TRUE if g1 and g2 are equal

Disjoint(g1 Geometry, g2 Geometry) :

Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments.

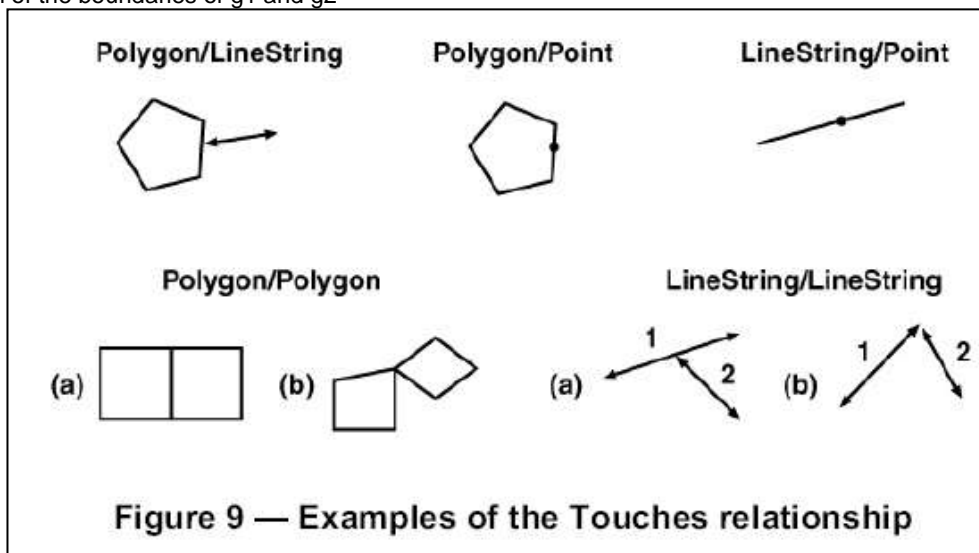
TRUE if the intersection of g1 and g2 is the empty set

Touches(g1 Geometry, g2 Geometry) :

Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments.

TRUE if the only Points in common between g1 and g2 lie in the union of the boundaries of g1 and g2

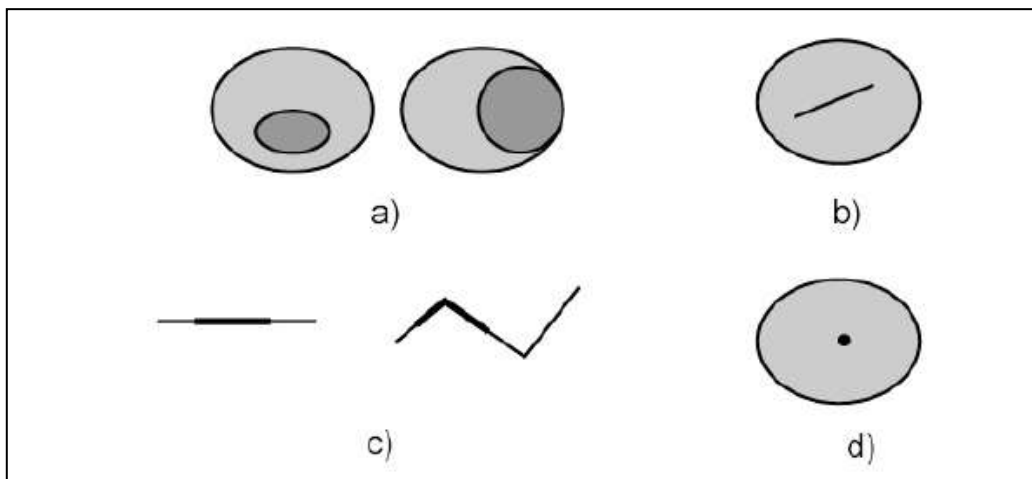


Within(g1 Geometry, g2 Geometry) :

Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments.

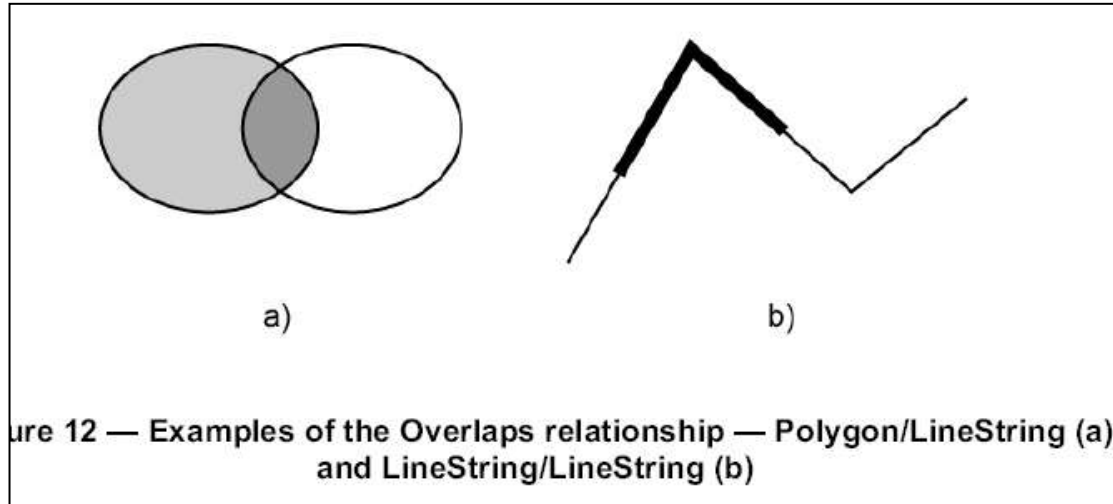
TRUE if g1 is completely contained in g2



Overlaps(g1 Geometry, g2 Geometry) :
Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments.

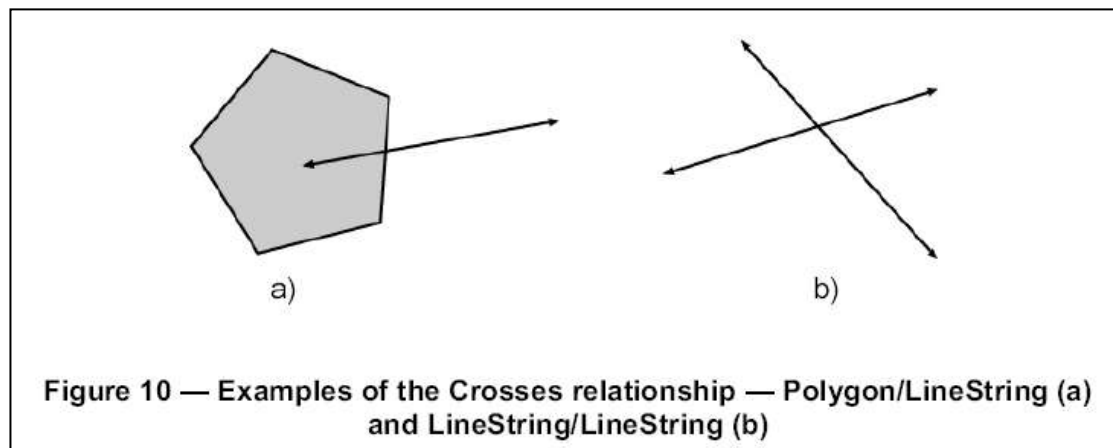
TRUE if the intersection of g1 and g2 results in a value of the same dimension as g1 and g2 that is different from both g1 and g2



Crosses(g1 Geometry, g2 Geometry) :
Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments.

TRUE if the intersection of g1 and g2 results in a value whose dimension is less than the maximum dimension of g1 and g2 and the intersection value includes Points interior to both g1 and g2, and the intersection value is not equal to either g1 or g2



Intersects(g1 Geometry, g2 Geometry) :
Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments;
convenience predicate: TRUE if the intersection of g1 and g2 is not empty

$Intersects(g1, g2) \equiv \text{Not } (Disjoint(g1, g2))$

Contains(g1 Geometry, g2 Geometry) :
Integer

The return type is Integer, with a return value of 1 for TRUE, 0 for

FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments;
convenience predicate: TRUE if g2 is completely contained in g1
Contains(g1, g2) □ *Within(g2, g1)*

Relate(g1 Geometry, g2 Geometry,
patternMatrix String) : Integer
The return type is Integer, with a return value of 1 for TRUE, 0 for FALSE, and -1 for UNKNOWN corresponding to a function invocation on NULL arguments;
returns TRUE if the spatial relationship specified by the patternMatrix holds