Spatial Concept Learning and Inference on Geospatial Polygon Data

Patrick Westphal[a,b], Tobias Grubenmann[c], Diego Collarana[a,d], Simon Bin[b,e], Lorenz Bühmann[b], Jens Lehmann[a,c,b]

[a] *Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Zwickauer Str. 46, 01069 Dresden, Germany*
[b] *Institute for Applied Informatics (InfAI) at the University of Leipzig, Goerdelerring 9, 04109 Leipzig, Germany*
[c] *Computer Science Institute, University of Bonn, Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Germany*
[d] *Institute for Computational Intelligence (ICI), Universidad Privada Boliviana, Avenida Juan Pablo II, Cochabamba, Bolivia*
[e] *Data Science (DICE) Group, Paderborn University, Warburger Str. 100, 33098 Paderborn, Germany*

Abstract

Geospatial knowledge has always been an essential driver for many societal aspects. This concerns in particular urban planning and urban growth management. To gain insights from geospatial data and guide decisions usually authoritative and open data sources are used, combined with user or citizen sensing data. However, we see a great potential for improving geospatial analytics by combining geospatial data with the rich terminological knowledge, e.g., provided by the Linked Open Data Cloud. Having semantically explicit, integrated geospatial and terminological knowledge, expressed by means of established vocabularies and ontologies, cross-domain spatial analytics can be performed. One analytics technique working on terminological knowledge is inductive concept learning, an approach that learns classifiers expressed as logical concept descriptions. In this paper, we extend inductive concept learning to infer and make use of the spatial context of entities in spatio-terminological data. We propose a formalism for extracting and making spatial relations explicit such that they can be exploited to learn spatial concept descriptions, enabling 'spatially aware' concept learning. We further provide an implementation of this formalism and demonstrate its capabilities in different evaluation scenarios.

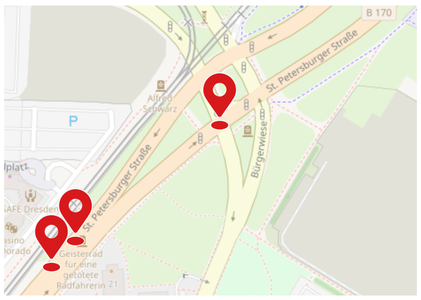
1. Introduction

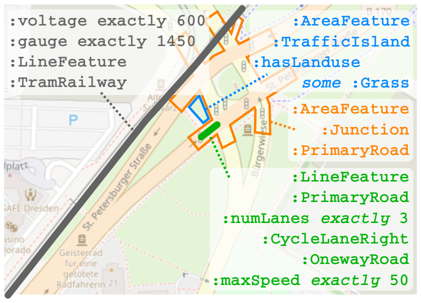
Geospatial knowledge has always been an important driver for many societal aspects. Especially since the beginning of the information age, vast amounts of digital geospatial data are being produced by companies and individuals every day. The ubiquity of mobile devices nowadays even enables the collection of *volunteered geographical information* (VGI) [1] from individuals on a large scale. Further, the idea of a *Web of Data* emerged and became reality in the form of the Linked Open Data (LOD) Cloud[1]. The LOD cloud contains open data sets covering many domains, including spatial data, which are expressed by means of the Resource Description Framework (RDF)[2] which allows to define vocabularies and ontologies with explicit semantics, and interlink data. Gaining insights and added value from such geospatial information is crucial in a world of pervasive mobile devices and location-aware information services. One recurring field of use cases is to harness geospatial *citizen sensing* data to guide and support administrative decisions in urban planning and urban growth [2, 3]. Urban development is of special importance since it is estimated that in 2050 about 68% of the earth's population will live in cities[3]. This poses challenges for governments, especially in developing countries, to provide proper infrastructure for better living standards of citizens. This aim is also reflected in the United Nations Sustainable Development Goals[4]. Further, different European Space Agency (ESA) initiatives employ spatial sensor data to support sustainable urban development[5].

The state-of-the-art with respect to the spatial analytics techniques that drive and support urban planning currently concentrates on VGI or sensor data, usually combined with special purpose spatial information that is lacking explicit semantics. Only little focus has been put on integrating VGI data with the rich terminological knowledge available on the Web of Data. We believe that by providing an enriched and integrated data infrastructure and the semantics-aware analytics capabilities on top, decision support services can be greatly improved. Such semantically explicit approaches usually operate on a symbolic level providing human and machine readable results to support decisions. One analytics method in artificial intelligence is the inductive learning of concept descriptions from labeled examples. Concept descriptions which then serve as a binary classifier are generated by means of the vocabulary of the terminological background knowledge base. Example outcomes could be rich descriptions of what constitutes a well accepted and popular road segment for cycling, or – to the contrary – what are the spatial and qualitative properties of bike accident hot spots. Since these descriptions are human and machine readable they can be taken into account by decision makers to decide on future actions in urban planning, as well as for an automatic detection and suggestions of other spots with similar properties.

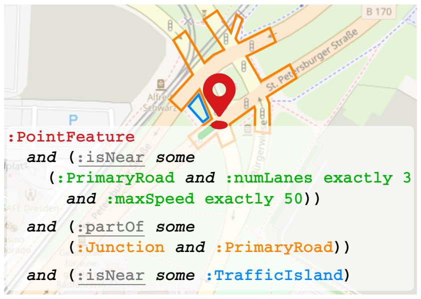
An example showing available complementary information about traffic accidents is shown in Figure 1. Whereas VGI usually provides mere geo-coordinates representing an appearance or event with a certain meaning (Figure 1a), the geospatial background information provided, e.g., by LinkedGeoData[6] [4] is se-

[1] `https://lod-cloud.net/`
[2] `https://www.w3.org/TR/rdf11-primer/`
[3] `https://ourworldindata.org/urbanization`
[4] `https://sdgs.un.org/goals`
[5] `https://eo4sd.esa.int/category/themes/urban/`
[6] `http://linkedgeodata.org`


(a) VGI reporting traffic accidents with bikes involved



(b) Background knowledge of feature types, based on OpenStreetMap



```
:PointFeature
  and (:isNear some
    (:PrimaryRoad and :numLanes exactly 3
      and :maxSpeed exactly 50))
  and (:partOf some
      (:Junction and :PrimaryRoad))
  and (:isNear some :TrafficIsland)
```

(c) Example of semantically rich description of a bike accident occurrence by means of the background knowledge using automatically inferred spatial relations (gray, underlined)

Figure 1: Illustration of the integrated usage of VGI and semantically rich background knowledge

mantically richer (Figure 1b). Here all kinds of spatial features are captured with their geometries and a comprehensive set of types, or concept expressions, describing them. Besides the open data sources available in the LOD cloud[7] this could also be further extended with authoritative data usually accessible to the municipalities. By combining both kinds of data sources one can get far richer descriptions of VGI data points and how they relate to their environment – not just in terms of spatial relations but also by means of terminological knowledge, e.g., describing POIs, road types etc. (Figure 1c).

Common Geographic Information Systems (GIS) provide a wide range of optimisations for processing geographical information, like indexes and special storage strategies to exploit data locality of spatial entities that are close to each other. In RDF, however, there are no means for a dedicated processing of spatial data and the polygon information describing the spatial entities' shapes and positions are stored as RDF literals in a string representation just like any other literal values. Hence, any optimisation techniques or spatial inferences

[7] `https://lod-cloud.net/`

have to be performed by the application using the data. However, since many geospatial RDF datasets as, for example, LinkedGeoData, were derived from GIS, or relational databases with GIS capabilities, characteristics like the spatial data's consistency are usually ensured. Other, implicit spatial characteristics, like spatial relations between entities modelled in RDF need to be made explicit, either 'on the fly' during data processing, or materialized and added to the dataset.

Structured geospatial RDF data and ontologies providing a conceptualisation of the spatial domain, e.g. expressed by means of the Web Ontology Language (OWL)[8], can be used to establish *spatially-aware* concept learning. However, to make use of the spatial information, *spatial reasoning* needs to be integrated into the OWL reasoning process. Furthermore, the concept learning algorithms need to be extended to better exploit spatial knowledge inferred by such reasoning components. In this regard, our contributions are the following: (i) We provide a formalisation of spatial relations that can be inferred by an extended spatially-aware OWL reasoner, (ii) we implemented such a spatial inference mechanism, and (iii) evaluated the impact on the reasoning performance in different experimental settings. (iv) We further formalise the notion of *spatial concept learning*, (v) describe our spatial concept learning extension for the DL-Learner framework and (vi) evaluate it in different learning scenarios on (geo-)spatial data.

This paper is structured as follows: We survey related work in Section 2. In Section 3 we cover the formal foundations for inferences on polygon data. Section 4 proposes an integration of the spatial inference mechanisms into Description Logics and Section 5 shows how the refinement-based inductive concept learning can be extended to also consider spatial relations. Implementation details are given in Section 6, and in Section 7, we discuss evaluation results of different spatial inference and spatial concept learning experiments. In Section 8 we conclude and give outlooks to future work.

2. Related Work

A considerable milestone providing an axiomatisation for the established field of topology dates back to the first half of the last century [5]. Here, the authors combine notions from set theory, topology and Boolean algebra to define an *algebra of topology*. Besides the introduction of the definition of a *topological space* the authors also presented connections to the modal logic **S4**.

Far later the spatial interval logic, called *Region Connection Calculus (RCC)* was defined [6], and refined [7], expressing a hierarchy of all possible relations of regions in space, as shown in Figure 2. The eight most special relations on the bottom of the hierarchy, usually referred to as *RCC-8*, have the favourable property of being *jointly-exhaustive and pairwise disjoint (JEPD)*. A further approach to provide a systematic collection of regions was the definition of the *Dimensionally Extended nine-Intersection Model (DE-9IM)* [8]. Whereas research

[8]`https://www.w3.org/TR/owl2-overview/`

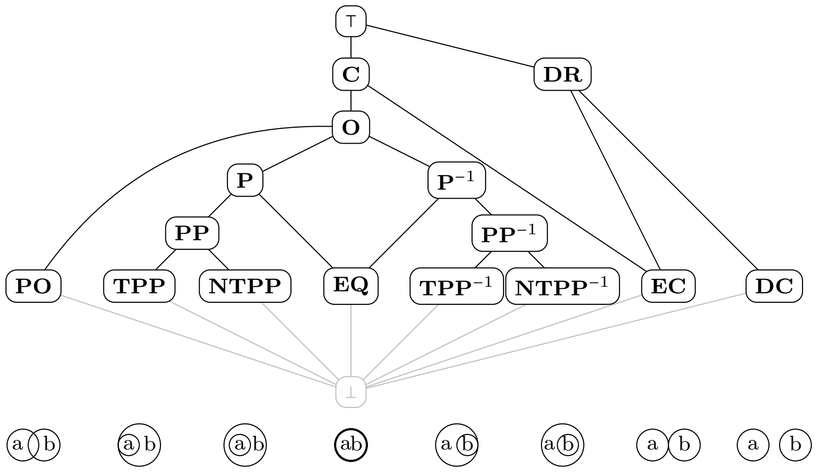


Figure 2: Region Connection Calculus relations: *Connected With* (**C**), *Discrete From* (**DR**), *Overlaps With* (**O**), *Part Of* (**P**), *Has Part* ($\mathbf{P}^{-1}$), *Proper Part Of* (**PP**), *Has Proper Part* ($\mathbf{PP}^{-1}$), *Partially Overlaps* (**PO**), *Tangential Proper Part Of* (**TPP**), *Non-tangential Proper Part Of* (**NTPP**), *Identical With* (**EQ**), *Has Tangential Proper Part* ($\mathbf{TPP}^{-1}$), *Has Non-tangential Proper Part* ($\mathbf{NTPP}^{-1}$), *Externally Connected With* (**EC**), *Disconnected From* (**DC**)

on RCC usually excludes all spatial entities not having a defined area (i.e., lines and points), DE-9IM explicitly considers area, line and point geometries and relations between them. The relations are defined by means of intersections of possible meaningful combinations of two geometries' *boundaries*, *interiors* and *exteriors*. This amounts to 512 possible spatial relations in 2-dimensional space (considering the different geometry types). But these can be summarised by a set of eight spatial relations bearing the jointly-exhaustive and pairwise disjointness (JEPD) feature: *disjoint*, *contains*, *inside*, *equal*, *meet*, *covers*, *coveredBy* and *overlap*.

In contrast, research in the field of human language and spatial cognition covers spatial relations in a broader sense [9]. Here, besides the topological aspect, further means to express spatial relations were collected. A proposed hierarchy of *subdomains of spatial language* from [9] is shown in Figure 3. This hierarchy assumes a 3-dimensional space and an observer looking at the relation of two spatial objects. Since we are considering a 2-dimensional space without an observer, certain spatial relations like *left to*, *behind* or *above* do not make sense. Nonetheless, to be able to learn intuitive, high quality concept descriptions we tried to include as many of the presented subdomains as possible. We added notes and examples in Figure 3 (in gray) to highlight this. In this regard, we trade a more general, and hopefully more intuitive, vocabulary to express spatial relations for the favorable JEPD feature mentioned above.

The RCC-8 semantics and its relations to Modal Logics structures were more thoroughly investigated in [10]. The outcomes were used for a further translation of the RCC-8 formalism into OWL-DL [11], which was eventually applied

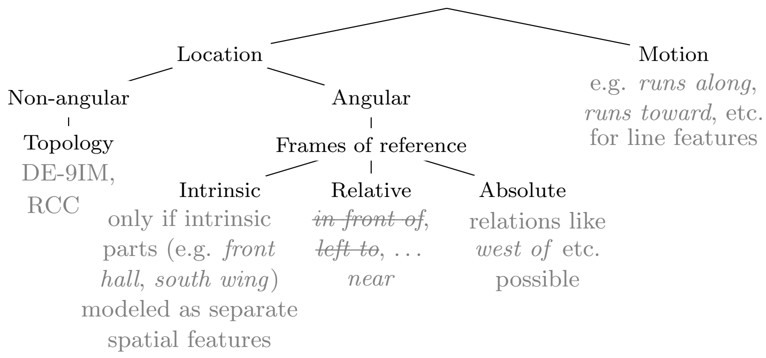

Location
Motion
e.g. *runs along*, *runs toward*, etc. for line features
Non-angular
Angular
Topology
DE-9IM, RCC
Frames of reference
Intrinsic
only if intrinsic parts (e.g. *front hall*, *south wing*) modeled as separate spatial features
Relative
~~*in front of*~~, ~~*left to*~~, … *near*
Absolute
relations like *west of* etc. possible

Figure 3: Hierarchy of subdomains of spatial language and example relations for spatial concept learning

to implement a ‘qualitative spatial reasoning engine’ for OWL ontologies [12]. These research avenues interpret regions as subsets of topological spaces, bridge their formal approaches to an **S4** modal logic, and eventually to OWL-DL. The main focus in terms of the reasoning task is to consider all topological constraints and derive whether a given set of logical assertions is satisfiable. Whereas [13] showed that the consistency problem for general topological constraint systems is NP-hard, if the goal was to find a *planar* model also taking geometric constraints into account, verifying the consistency of RCC-8 constraint systems was shown to be NP-complete [10]. In contrast, the consistency problem of the modal translation in [10] is PSpace-complete [14]. In [15], these complexity findings are discussed and related in depth. Further the authors provide an analysis on how to restrict the set of spatial relations of RCC-8 to gain a polynomial-time procedure to check consistency. However, for our spatial concept learning approach the main reasoning task regarding spatial inference is *instance classification*, or *instance checking*, not consistency validation. Accordingly, these formal frameworks for modeling spatial regions are not considered in our work.

Another line of research incorporated qualitative spatial reasoning into Description Logics modeling the spatial information by means of *concrete domains*. This is already covered by the standard literature on Description Logics like [16], presenting the RCC formalism as one example of concrete domains. Here, again, the main reasoning task is consistency checking and the tableau algorithm as a means to validate consistency is extended accordingly. However, to keep the consistency problem decidable strict requirements on the formalism to represent topological knowledge are imposed. Whereas these hold for RCC-8, approaches using more flexible vocabularies to express spatial relations will likely go beyond the complexity classes presented and envisioned there. In [17], requirements are further detailed and tightened to improve the complexity implications of tableau-based consistency checking in Description Logics with spatial concrete domains based on RCC-8.

A similar idea was presented in [18] where the authors concentrate on DE-9IM-like relations that can hold between spatial entities expressed by means of *polygon strings*. Polygon data is also modeled as a concrete domain. Further, the authors consider explicitly the semantic hierarchy of spatial relations, e.g. that *generally_inside* subsumes *equals*.

A more general view on spatial knowledge representation and reasoning, not only considering consistency checking is presented in [19]. Other than the previous approaches, spatial relations are not modeled as predicates of spatial regions inside a concrete domain, but rather as relations that hold between individuals of the abstract domain. We think this is a more natural approach especially for concept learning. Further, the authors describe a system that can be used for *ontology-based* search which also includes reasoning (e.g. to resolve sub-concept relations). Similar to one of the approaches in [12], they use a specialized RCC reasoner which can be used to infer such spatial relations between individuals of the abstract domain.

The task of making spatial relations between individuals explicit in a *link discovery* setting was studied in [20]. This approach used DE-9IM but other than inferring spatial relations 'on the fly' (i.e. during query time), discovered relations are usually meant to be stored and added to the knowledge base as new assertions to hold between individuals. This was further improved in [21] where the authors introduced the notion of *Progressive Holistic Geospatial Interlinking* which allows to compute topological relations in a 'pay-as-you-go' manner. We argue that this 'offline processing' is not always meaningful for settings like concept learning since it might tremendously increase the size of the knowledge base.

Another project focusing entirely on search, i.e. inferring (implicit) spatial relations from spatial information stored in an optimized spatial data store, not applying any Description Logics reasoning techniques, was presented in [22]. The demonstrated query engine allows to make use of spatial extensions of the SPARQL query language[9] like GeoSPARQL [23] or stSPARQL [22].

An approach to inductively learn classifiers for point data, based on Formal Concept Analysis [24] was proposed in [25]. Here, the classifiers are expressed by means of convex polygons containing the provided example points. Using more complex polygon shapes increases the expressivity over, e.g., simple bounding box approaches, whereas the requirement of them being *convex* allows to keep their algorithmic complexity moderate. In our attempt to inductively learn classifiers for polygon data we do not concentrate on polygon patterns but rather on logical descriptions.

The idea of concept learning, as it is used here, stems from the field of *Inductive Logic Programming* (ILP) [26], in which many popular implementations emerged [27, 28, 29]. However, to the best of our knowledge, there is only one attempt to combine ILP with spatial pattern learning [30]. Unlike our approach, the corresponding system only considers the learning part and assumes

[9] `https://www.w3.org/TR/sparql11-overview/`

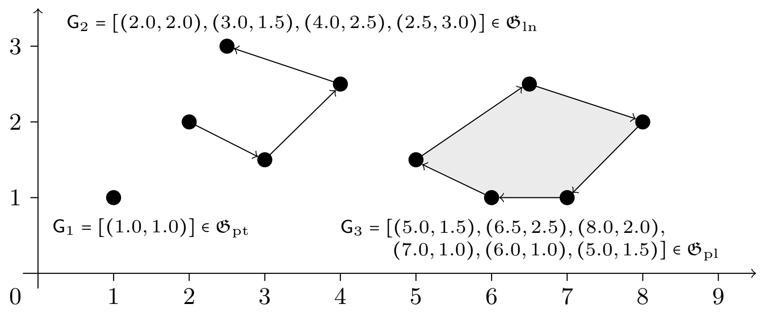

$\mathsf{G}_2 = [(2.0, 2.0), (3.0, 1.5), (4.0, 2.5), (2.5, 3.0)] \in \mathfrak{G}_{\mathrm{ln}}$
$\mathsf{G}_1 = [(1.0, 1.0)] \in \mathfrak{G}_{\mathrm{pt}}$
$\mathsf{G}_3 = [(5.0, 1.5), (6.5, 2.5), (8.0, 2.0), (7.0, 1.0), (6.0, 1.0), (5.0, 1.5)] \in \mathfrak{G}_{\mathrm{pl}}$
0
1
2
3
4
5
6
7
8
9

Figure 4: Examples of three geometry primitives: a primitive point (G_1), a primitive line string (G_2), and a primitive polygon (G_3)

all inferred spatial relations already materialized in the input data.

In the field of concept learning in Description Logics, the most prevalent framework is the DL-Learner[10] [31], which provides a variety of algorithms for supervised Machine Learning in OWL, RDF and Description Logics. The DL-Learner system follows the idea of *generalisation as search* [32] and applies, among others, refinement operator-based techniques [33, 34, 35, 36]. A further example of a concept learning system is YinYang [35]. However, whereas the DL-Learner can be configured to support different Description Logics or OWL profiles, YinYang focuses on learning $\mathcal{ALC}$ concepts only. The formalism introduced in the next sections was implemented as an extension of the DL-Learner framework to enable spatial concept learning by introducing and automatically inferring spatial relations that hold between individuals of the abstract domain. In particular we based our work on the CELOE algorithm [37] with an adapted set of refinement rules as described in Section 5.

3. A Calculus for Polygon Data

In this section we introduce the main building blocks to express the collection of spatial relations used in the following sections. We base our definitions on the notion of a *geometry primitive* which can either be a (primitive) point, a (primitive) line string, or a (primitive) polygon as sketched in Figure 4. Following the common understanding, each geometry $\mathsf{G} = [(x_1, y_1), \ldots, (x_n, y_n)]$ represents a subset of points in the 2-dimensional Euclidean space $\mathbb{R}^2$, being *inside* G. The sequence of points $(x_1, y_1), \ldots, (x_n, y_n)$ is called G's *boundary*. Every two consecutive points $[(x_i, y_i), (x_{i+1}, y_{i+1})]$ form a *line segment*.

The distinction between a (primitive) *point*, a (primitive) *line string*, and a (primitive) *polygon* is defined as follows: If G's boundary is *closed*, i.e. $(x_1, y_1) = (x_n, y_n)$ and $n \geq 3$, then G is a (primitive) *polygon*. We assume the points are describing G's boundary in clockwise order with the area of G always being on the

[10] `http://dl-learner.org`

right side of each line segment $[(x_i, y_i), (x_{i+1}, y_{i+1})]$ (resp. $[(x_n, y_n), (x_1, y_1)]$ in case of the last, *closing* segment). Further we assume, that the boundary line has no self-intersections other than the start/end point. If G contains only one entry, i.e. $\mathsf{G} = [(x_1, y_1)]$ then it is called a (primitive) *point*. Otherwise, G is called a (primitive) *line string*. For primitive line strings we also assume that there are no self-intersections. The set of geometry primitives is denoted by $\mathfrak{G}$. The sets of primitive points $\mathfrak{G}_{\text{pt}}$, primitive line strings $\mathfrak{G}_{\text{ln}}$, and primitive polygons $\mathfrak{G}_{\text{pl}}$ are all subsets of $\mathfrak{G}$ and mutually disjoint.

In real-world GIS, one may often find geometries that are not primitive, e.g. *multi-polygons*, *multi-line strings*, *multi-points* or *polygon collections* [38]. To represent those, the above as well as the following definitions needed to be extended to consider (generic) geometries as *sets* of geometry primitives. However, for brevity we introduce the notions of our polygon calculus only on geometry primitives. For complexity considerations we already report the most general cases also covering multi-geometries. Accordingly, in the following we will omit the 'primitive' adjective and always refer to their primitive versions whenever we talk about points, line strings, and polygons.

As mentioned above, a geometry primitive G's boundary, $\partial(\mathsf{G})$, is treated as a line string if G is a primitive polygon, i.e. $\mathsf{G} \in \mathfrak{G}_{\text{pl}}$. In this case, $\partial(\mathsf{G})$ is closed, i.e. a line string having the same start and end point. If G is a line string $\mathsf{G} \in \mathfrak{G}_{\text{ln}}$, its boundary $\partial(\mathsf{G})$ is the set containing G's start and end point. For a point $\mathsf{G} \in \mathfrak{G}_{\text{pt}}$, G's boundary $\partial(\mathsf{G})$ degenerates to a single point being identical to G itself. Further, we define the function $\mathcal{S} : \mathfrak{G}_{\text{ln}} \to 2^{\mathfrak{G}_{\text{ln}}}$ to return the set of all line segments of a line string. More formally, for a line string $\mathsf{G} = [(x_1, y_1), \ldots, (x_n, y_n)] \in \mathfrak{G}_{\text{ln}}$ the set of G's segments $\mathcal{S}(\mathsf{G})$ is defined as $\{[(x_i, y_i), (x_{i+1}, y_{i+1})]\}_{i=1}^{n-1}$.

In the field of computational geometry there are certain *fundamental* operations defined on geometry primitives that are well studied with respect to their computational complexity. The fundamental operations used to define our polygon calculus are given in Table 1. To define a generic *intersects* relation which holds whenever two geometry primitives intersect we make use of the fundamental operations as follows:

Definition 3.1 (Intersects, $\asymp$)**.** For two geometry primitives $\mathsf{G}_1 \in \mathfrak{G}$ and $\mathsf{G}_2 \in \mathfrak{G}$ the *intersects* relation $\asymp(\mathsf{G}_1, \mathsf{G}_2)$ holds if one of the following cases is true:

$\asymp(\mathsf{G}_1, \mathsf{G}_2)$ if...	$\mathsf{G}_2 = [(x_2, y_2)] \in \mathfrak{G}_{\text{pt}}$	$\mathsf{G}_2 \in \mathfrak{G}_{\text{ln}}$	$\mathsf{G}_2 \in \mathfrak{G}_{\text{pl}}$
$\mathsf{G}_1 = [(x_1, y_1)] \in \mathfrak{G}_{\text{pt}}$	$(x_1, y_1) = (x_2, y_2)$	$\mathsf{G}_1 \lessdot \mathsf{G}_2$	
$\mathsf{G}_1 = [(x_1, y_1), \ldots, (x_n, y_n)] \in \mathfrak{G}_{\text{ln}}$	$\asymp(\mathsf{G}_2, \mathsf{G}_1)$	$\asymp_{\text{ln}}(\mathcal{S}(\mathsf{G}_1), \mathcal{S}(\mathsf{G}_2))$	$\exists (x_i, y_i) \in \mathsf{G}_1 : \mathsf{G}_i = [(x_i, y_i)] \lessdot \mathsf{G}_2$, or $\asymp_{\text{ln}}(\mathcal{S}(\mathsf{G}_1), \mathcal{S}(\partial(\mathsf{G}_2)))$
$\mathsf{G}_1 \in \mathfrak{G}_{\text{pl}}$	$\asymp(\mathsf{G}_2, \mathsf{G}_1)$	$\asymp(\mathsf{G}_2, \mathsf{G}_1)$	$\exists (x_1, y_1) \in \mathsf{G}_1 : \mathsf{G}_i = [(x_i, y_i)] \lessdot \mathsf{G}_2$, or $\asymp_{\text{ln}}(\mathcal{S}(\partial(\mathsf{G}_1)), \mathcal{S}(\partial(\mathsf{G}_2)))$

[11]Please note that we are considering polygons as closed regions, i.e. the boundary is considered part of a polygon.

Table 1: Fundamental operations on geometries and their computational complexities

Operation	Definition	Complexity
$\propto_{\mathrm{ln}}$ *(has line intersections)*	Given two sets of line segments $\mathcal{S}(\mathsf{G}_1) \subset \mathfrak{G}_{\mathrm{ln}}$ and $\mathcal{S}(\mathsf{G}_2) \subset \mathfrak{G}_{\mathrm{ln}}$, the relation $\propto_{\mathrm{ln}} (\mathcal{S}(\mathsf{G}_1), \mathcal{S}(\mathsf{G}_2))$ holds if there exists a line segment $\mathsf{G}_i \in \mathcal{S}(\mathsf{G}_1)$ and a line segment $\mathsf{G}_j \in \mathcal{S}(\mathsf{G}_2)$ such that G_i and G_j intersect.	$O(\log(N))$ where $N = \lvert\mathcal{S}(\mathsf{G}_1)\rvert + \lvert\mathcal{S}(\mathsf{G}_2)\rvert$ [39]
$\lessdot$ *(point is inside polygon)*	The problem of deciding whether a point $\mathsf{G}_1 = [(x,y)] \in \mathfrak{G}_{\mathrm{pt}}$ is inside a polygon $\mathsf{G}_2 \in \mathfrak{G}_{\mathrm{pl}}$ is covered by the so called *point-in-polygon* problem [40]. $\mathsf{G}_1 \lessdot \mathsf{G}_2$ holds iff the point-in-polygon relation holds for (x,y) and G_2. We use the same notation for the case that $\mathsf{G}_2 \in \mathfrak{G}_{\mathrm{ln}}$ to denote that the point $\mathsf{G}_1 = [(x,y)] \in \mathfrak{G}_{\mathrm{pt}}$ equals one of G_2's points (i.e. $(x,y) \in \mathsf{G}_2$), or lies on one of G_2's line segments. To avoid confusions, in the following we will use $\in$ to express set or list containment, whereas $\lessdot$ refers to *spatial* containment, i.e. that a point is inside a line string or polygon.	$O(N)$ where $N = \lvert\mathcal{S}(\partial(\mathsf{G}_2))\rvert$ if $\mathsf{G}_2 \in \mathfrak{G}_{\mathrm{pl}}$[11], and $N = \lvert\mathcal{S}(\mathsf{G}_2)\rvert$ if $\mathsf{G}_2 \in \mathfrak{G}_{\mathrm{ln}}$
$\between_{\mathrm{ln}}$ *(intersection points)*	For two sets of line segments $\mathcal{S}(\mathsf{G}_1) \subseteq \mathfrak{G}_{\mathrm{ln}}$ and $\mathcal{S}(\mathsf{G}_2) \subseteq \mathfrak{G}_{\mathrm{ln}}$ the function $\between_{\mathrm{ln}}(\mathcal{S}(\mathsf{G}_1), \mathcal{S}(\mathsf{G}_2)) = \mathcal{G}_{\mathrm{pt}}$ returns the set of intersection points $\mathcal{G}_{\mathrm{pt}} \subseteq \mathfrak{G}_{\mathrm{pt}}$ between segments from $\mathcal{S}(\mathsf{G}_1)$ and segments from $\mathcal{S}(\mathsf{G}_2)$.	$O(N \log(N{+}K))$ with $N = \lvert\mathcal{S}(\mathsf{G}_1)\rvert + \lvert\mathcal{S}(\mathsf{G}_2)\rvert$, and $K = \lvert\mathcal{G}_{\mathrm{pt}}\rvert$ [41]
$\boldsymbol{\ell}$ *(length)*	The length function $\ell : \mathfrak{G}_{\mathrm{ln}} \mapsto \mathbb{R}$ is defined on line strings in the natural way.	$O(N)$ where $N = \mathcal{S}(\mathsf{G})$ for $\mathsf{G} \in \mathfrak{G}_{\mathrm{ln}}$
$\mho$ *(buffer)*	Given an offset $d \in \mathbb{R}$, the *buffer* function [40] $\mho(\mathsf{G}, d)$ expands or grows a geometry G along all its points by d units. For disk shapes, e.g. when applying the buffer operation to simple points we assume a suitable and accurate enough polygon approximation.	Superlinear in $N = \lvert\mathsf{G}\rvert$, however, empirically usually near $O(N)$ [42]
$\prec_{\mathsf{G}_{\mathrm{ln}}}$	Given a set of points $\mathcal{G} = \{\mathsf{G}_1, \mathsf{G}_2, \ldots \mathsf{G}_n\} \subset \mathfrak{G}_{\mathrm{pt}}$ each intersecting with a line string $\mathsf{G}_{\mathrm{ln}} \in \mathfrak{G}_{\mathrm{ln}}$, $\prec_{\mathsf{G}_{\mathrm{ln}}}$ imposes a total order on $\mathcal{G}$ w.r.t. the distance of its elements to G_{ln}'s start point, along G_{ln}, i.e. a sorting on their linear referencing on G_{ln}.	$O(\lvert\mathcal{G}\rvert \log(\lvert\mathcal{G}\rvert))$ as we usually know on which segment of G_{ln} a point $\mathsf{G}_i \in \mathcal{G}$ lies.

In terms of complexity the most expensive cases for $\propto$ are to determine whether a line string or a polygon G_1 intersects with another polygon $\mathsf{G}_2 \in \mathfrak{G}_{\mathrm{pl}}$. In both cases the point-in-polygon check, linear in the number of the boundary segments of G_2, has to be performed for each line string point if $\mathsf{G}_1 \in \mathfrak{G}_{\mathrm{ln}}$, and each point of the boundary if $\mathsf{G}_1 \in \mathfrak{G}_{\mathrm{pl}}$, which amounts to a quadratic time complexity in the worst case (i.e. $O(N{\cdot}M)$ with $N = \lvert\mathcal{S}(\mathsf{G}_1)\rvert$, or $N = \lvert\mathcal{S}(\partial(\mathsf{G}_1))\rvert$, respectively, and $M = \lvert\mathcal{S}(\partial(\mathsf{G}_2))\rvert$).

A function to retrieve the actual points where two polygon primitives intersect can, again, be defined based on the respective fundamental operation for sets of line segments [40]. The function to find intersection points on two geometries of arbitrary type can then be defined as follows:

Definition 3.2 (Intersection points, $\between$)**.** Given two geometries $\mathsf{G}_1, \mathsf{G}_2 \in \mathfrak{G}$, the function $\between : \mathfrak{G} \times \mathfrak{G} \mapsto 2^{\mathfrak{G}_{\text{pt}}}$ is defined by one of the following cases:

$\between(\mathsf{G}_1, \mathsf{G}_2) = \ldots$	$\mathsf{G}_2 = [(x_2, y_2)] \in \mathfrak{G}_{\text{pt}}$	$\mathsf{G}_2 \in \mathfrak{G}_{\text{ln}}$	$\mathsf{G}_2 \in \mathfrak{G}_{\text{pl}}$
$\mathsf{G}_1 = [(x_1, y_1)] \in \mathfrak{G}_{\text{pt}}$	$\{\mathsf{G}_1\}$ if $(x_1, y_1) = (x_2, y_2)$ $\varnothing$ otherwise	$\{\mathsf{G}_1\}$ if $\mathsf{G}_1 \lessdot \mathsf{G}_2; \varnothing$ otherwise	
$\mathsf{G}_1 \in \mathfrak{G}_{\text{ln}}$	$\between(\mathsf{G}_2, \mathsf{G}_1)$	$\between_{\text{ln}}(\mathcal{S}(\mathsf{G}_1), \mathcal{S}(\mathsf{G}_2))$	$\between_{\text{ln}}(\mathcal{S}(\mathsf{G}_1), \mathcal{S}(\partial(\mathsf{G}_2)))$
$\mathsf{G}_1 \in \mathfrak{G}_{\text{pl}}$	$\between(\mathsf{G}_2, \mathsf{G}_1)$	$\between(\mathsf{G}_2, \mathsf{G}_1)$	$\between_{\text{ln}}(\mathcal{S}(\partial(\mathsf{G}_1)), \mathcal{S}(\partial(\mathsf{G}_2)))$

Accordingly, $\between$ inherits the complexity implications of $\between_{\text{ln}}$.

With these fundamental operations a variety of spatial relations can be defined. These are not limited to the relations from the RCC but include relations from multiple subdomains of spatial language (cf. Figure 3). An overview of all proposed relations is given in Table 2. The respective time complexity considerations are summarised in Table 3. The proposed relations are discussed in the following.

Connected With/Disconnected From. The *Connected With* relation corresponds to the fundamental operation *intersects* introduced above. Accordingly, *Connected With* is of the same worst case time complexity class, i.e. $O(N \cdot M)$ (where N refers to the number of points defining the first, and M refers to the number of points defining the second geometry). However, this worst case complexity only applies to cases where the *Connected With* relation is checked between either line strings or polygons, and polygons. All other cases have more favorable complexities. Since *Disconnected From* only holds if *Connected With* does not hold, the same complexity considerations apply.

Identical With. Two points (line strings) are considered equal if they are defined by the same (sequence of) coordinates. This can obviously be evaluated in linear time. However, in case of polygons, we consider them as equal if they are defined by the same circular sequence modulo the starting point. This means that the polygons defined by the sequences $[p_1, p_2, p_3, p_1]$, $[p_2, p_3, p_1, p_2]$ and $[p_3, p_1, p_2, p_3]$ are all considered identical. However, since we are assuming polygons defined in clockwise order this does not affect the worst case time complexity of the *Identical With* relation. Note that in practice, definition **EQ**1 can be, and usually is, relaxed to also consider two line strings as equal if they are defined by the same points, but in inverse order.

Tangential Proper Part Of. The *Tangential Proper Part Of* relation requires that one geometry G_1 has to be inside a containing geometry G_2 (which already takes quadratic time to check in the worst case with respect to the number of (boundary) segments of the geometries) (cf. line **TPP**1), and at least one of G_1's defining points has to lie on G_2's boundary (cf. line **TPP**3). Further, at least one of G_2's defining points has to lie outside G_1 to establish a proper part relation ruling out spatial identity. As already noted above in the worst case

Table 2: Definitions of the considered spatial relations

Relation	Definition	
Connected With	$\mathbf{C}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \asymp(\mathsf{G}_1, \mathsf{G}_2)$	($\mathbf{C}$1)
Disconnected From	$\mathbf{DC}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \not\asymp(\mathsf{G}_1, \mathsf{G}_2)$	($\mathbf{DC}$1)
Part Of	$\mathbf{P}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{EQ}(\mathsf{G}_1, \mathsf{G}_2) \vee \mathbf{PP}(\mathsf{G}_1, \mathsf{G}_2)$	($\mathbf{P}$1)
Proper Part Of	$\mathbf{PP}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{TPP}(\mathsf{G}_1, \mathsf{G}_2) \vee \mathbf{NTPP}(\mathsf{G}_1, \mathsf{G}_2)$	($\mathbf{PP}$1)
Identical With	$\mathbf{EQ}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq ((\mathsf{G}_1, \mathsf{G}_2 \in \mathfrak{G}_{\mathrm{ln}} \vee \mathsf{G}_1, \mathsf{G}_2 \in \mathfrak{G}_{\mathrm{pt}}) \wedge \mathsf{G}_1 = \mathsf{G}_2) \vee (\mathsf{G}_1, \mathsf{G}_2 \in \mathfrak{G}_{\mathrm{pl}} \wedge \mathcal{S}(\partial(\mathsf{G}_1)) = \mathcal{S}(\partial(\mathsf{G}_2)))$	($\mathbf{EQ}$1)
Overlaps	$\mathbf{O}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{PO}(\mathsf{G}_1, \mathsf{G}_2) \vee \mathbf{P}(\mathsf{G}_1, \mathsf{G}_2) \vee \mathbf{P}^{-1}(\mathsf{G}_1, \mathsf{G}_2)$	($\mathbf{O}$1)
Discrete From	$\mathbf{DR}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq\approx \not\asymp(\mathsf{G}_1, \mathsf{G}_2) \wedge \asymp(\overset{\leftrightarrow}{\circ}(\mathsf{G}_1, \epsilon), \mathsf{G}_2)$ for some $\epsilon > 0$	($\mathbf{DR}$1)
Partially Overlaps	$\mathbf{PO}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq$	
	$(\mathsf{G}_1 \in \mathfrak{G}_{\mathrm{ln}} \wedge \exists \mathsf{G}_{\mathrm{ln}} \in \mathcal{S}(\mathsf{G}_1), \mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2} : \{\mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2}\} \subseteq \between(\mathsf{G}_{\mathrm{ln}}, \mathsf{G}_2) \wedge \mathsf{G}_{\mathrm{pt}_1} \neq \mathsf{G}_{\mathrm{pt}_2}) \vee$	($\mathbf{PO}$1)
	$(\mathsf{G}_1 \in \mathfrak{G}_{\mathrm{pl}} \wedge \exists \mathsf{G}_{\mathrm{ln}} \in \mathcal{S}(\partial(\mathsf{G}_1)), \mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2} : \{\mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2}\} \subseteq \between(\mathsf{G}_{\mathrm{ln}}, \mathsf{G}_2) \wedge \mathsf{G}_{\mathrm{pt}_1} \neq \mathsf{G}_{\mathrm{pt}_2}) \vee$	($\mathbf{PO}$2)
	$(\exists (x_i, y_i) \in \mathsf{G}_1, \exists (x_o, y_o) \in \mathsf{G}_1 :$ $\mathsf{G}_i = [(x_i, y_i)] \lessdot \mathsf{G}_2 \wedge \not\asymp(\mathsf{G}_i, \partial(\mathsf{G}_2)) \wedge \mathsf{G}_o = [(x_o, y_o)] \not\lessdot \mathsf{G}_2) \vee$	($\mathbf{PO}$3)
	$\mathbf{PO}(\mathsf{G}_2, \mathsf{G}_1)$	($\mathbf{PO}$4)
Externally Connected With	$\mathbf{EC}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq (\forall (x_i, y_i) \in \mathsf{G}_1 : \mathsf{G}_i = [(x_i, y_i)] \lessdot \mathsf{G}_2 \rightarrow \mathsf{G}_i \lessdot \partial(\mathsf{G}_2)) \vee \mathbf{EC}(\mathsf{G}_2, \mathsf{G}_1)$	($\mathbf{EC}$1)
Tangential Proper Part Of	$\mathbf{TPP}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq$	
	$(\forall (x_i, y_i) \in \mathsf{G}_1 : \mathsf{G}_i = [(x_i, y_i)] \lessdot \mathsf{G}_2) \wedge$	($\mathbf{TPP}$1)
	$(\exists (x_j, y_j) \in \mathsf{G}_2 : \mathsf{G}_j = [(x_j, y_j)] \not\lessdot \mathsf{G}_1) \wedge$	($\mathbf{TPP}$2)
	$((\exists (x_k, y_k) \in \partial(\mathsf{G}_1) : \mathsf{G}_k = [(x_k, y_k)] \lessdot \partial(\mathsf{G}_2)) \vee$ $(\exists (x_m, y_m) \in \partial(\mathsf{G}_2) : \mathsf{G}_m = [(x_m, y_m)] \lessdot \partial(\mathsf{G}_1)))$	($\mathbf{TPP}$3)

Table 2: Definitions of the considered spatial relations (cont.)

Relation	Definition	
Nontangential Proper Part Of	$\mathbf{NTPP}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \forall (x_i, y_i) \in \mathsf{G}_1 : \mathsf{P}_i = [(x_i, y_i)] \lessdot \mathsf{G}_2 \wedge \not\asymp (\partial(\mathsf{G}_1), \partial(\mathsf{G}_2))$	$(\mathbf{NTPP}1)$
Has Part	$\mathbf{P}^{-1}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{P}(\mathsf{G}_2, \mathsf{G}_1)$	$(\mathbf{P}^{-1}1)$
Has Proper Part	$\mathbf{PP}^{-1}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{PP}(\mathsf{G}_2, \mathsf{G}_1)$	$(\mathbf{PP}^{-1}1)$
Has Tangential Proper Part	$\mathbf{TPP}^{-1}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{TPP}(\mathsf{G}_2, \mathsf{G}_1)$	$(\mathbf{TPP}^{-1}1)$
Has Nontangential Proper Part	$\mathbf{NTPP}^{-1}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathbf{NTPP}(\mathsf{G}_2, \mathsf{G}_1)$	$(\mathbf{NTPP}^{-1}1)$
Runs Along	$\mathrm{RA}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq$	
	$\mathsf{G}_1, \mathsf{G}_2 \in \mathfrak{G}_{\mathrm{ln}} \wedge \exists \mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2} :$	(RA1)
	$\{\mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2}\} \subseteq \between (\mathsf{G}_1, \overset{\leftrightarrow}{\circ}(\mathsf{G}_2, d)) \wedge$	(RA2)
	$\mathsf{G}_{\mathrm{pt}_1} \prec_{\mathsf{G}_1} \mathsf{G}_{\mathrm{pt}_2} \wedge \nexists \mathsf{G}_{\mathrm{pt}_3} \in \between (\mathsf{G}_1, \overset{\leftrightarrow}{\circ}(\mathsf{G}_2, d)) : \mathsf{G}_{\mathrm{pt}_1} \prec_{\mathsf{G}_1} \mathsf{G}_{\mathrm{pt}_3} \prec_{\mathsf{G}_1} \mathsf{G}_{\mathrm{pt}_2} \wedge$	(RA3)
	$\ell(\mathsf{G}_{1_{\vdash \mathsf{G}_{\mathrm{pt}_1}, \mathsf{G}_{\mathrm{pt}_2} \dashv}}) > 2d$	(RA4)
Is Near	$\mathrm{N}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq$	
	$\asymp (\overset{\leftrightarrow}{\circ}(\mathsf{G}_1, \delta), \mathsf{G}_2)$ for some fixed δ	(N1)
Starts Near	$\mathrm{SN}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathsf{G}_1 = [(x_1, y_1), \ldots, (x_n, y_n)] \in \mathfrak{G}_{\mathrm{ln}} \wedge \mathrm{N}(\mathsf{G}_{\mathrm{pt}_1} = [(x_1, y_1)], \mathsf{G}_2)$	(SN1)
Ends Near	$\mathrm{SN}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq \mathsf{G}_1 = [(x_1, y_1), \ldots, (x_n, y_n)] \in \mathfrak{G}_{\mathrm{ln}} \wedge \mathrm{N}(\mathsf{G}_{\mathrm{pt}_n} = [(x_n, y_n)], \mathsf{G}_2)$	(EN1)
Crosses	$\mathrm{Cr}(\mathsf{G}_1, \mathsf{G}_2) \coloneqq$	
	$\mathsf{G}_1 = [(x_1, y_1), \ldots, (x_n, y_n)] \in \mathfrak{G}_{\mathrm{ln}} \wedge$	(Cr1)
	$\asymp (\mathsf{G}_1, \mathsf{G}_2) \wedge$	(Cr2)
	$\mathsf{G}_{\mathrm{pt}_1} = [(x_1, y_1)] \not\lessdot \mathsf{G}_2 \wedge \mathsf{G}_{\mathrm{pt}_n} = [(x_n, y_n)] \not\lessdot \mathsf{G}_2$	(Cr3)

Table 3: Worst case time complexity considerations of the proposed spatial relations. N and M refer to the respective number of coordinates (x_i, y_i) that define the point(s), line string(s), or the boundary line(s) of the polygon(s) evaluated for the respective spatial relation to hold.

Relation	Time Complexity
Connected With	$O(N \cdot M)$
Disconnected From	$O(N \cdot M)$
Part Of	$O(N \cdot M)$
Proper Part Of	$O(N \cdot M)$
Identical With	$O(N)$
Overlaps	$O(N \cdot M)$
Discrete From	$O(N \cdot (M + \log(n)))$ where n is the number of monotone chains in the (boundary of the) first geometry
Partially Overlaps	$O(N \cdot M)$
Externally Connected With	$O(N \cdot M)$
Tangential Proper Part Of	$O(N \cdot M)$
Nontangential Proper Part Of	$O(N \cdot M)$
Has Part	$O(N \cdot M)$
Has Proper Part	$O(N \cdot M)$
Has Tangential Proper Part	$O(N \cdot M)$
Has Nontangential Proper Part	$O(N \cdot M)$
Runs Along	$O(M \cdot \log(m) + (N + M) \cdot \log(N + M) + K \cdot \log(K))$ where m is the number of monotone chains in the second line string, and K is the number of intersection points of the first line string and the boundary of the second line string's buffer polygon
Is Near	$O(N \cdot (M + \log(n)))$ where n is the number of monotone chains in the (boundary of the) first geometry
Starts Near	$O(M)$
Ends Near	$O(M)$
Crosses	$O(N \cdot M)$

these checks require quadratic time in terms of the number of line segments of the involved geometries.

Nontangential Proper Part Of. For a geometry G_1 to be a *nontangential proper part of* a containing geometry G_2 all defining points of G_1 have to be inside G_2, whereas the boundaries of both geometries must be disconnected. The required point-in-polygon check for each of G_1's defining points, as well as the intersection check have a quadratic worst case complexity in the number of segments of the geometries involved.

Proper Part Of. A geometry G_1 is a *proper part of* a geometry G_2 either if G_1 is a tangential, or a nontangential proper part of G_2. As both *Tangential Proper Part Of* and *Nontangential Proper Part Of* have a worst case time complexity being quadratic in the number of (boundary) line segments of the geometries involved, this complexity class also applies to *Proper Part Of.*

Part Of. A geometry G_1 is *part of* a geometry G_2 either if G_1 is identical with G_2, or if G_1 is a proper part of G_2. Accordingly, the worst case time complexity is dominated by the harder of both relations, which is *Proper Part Of* being quadratic in the number of (boundary) line segments of the geometries involved.

Partially Overlaps. In general a line string or polygon G_1 *partially overlaps* with another geometry G_2 if either (i) there is a line segment of (the boundary of) G_1 that has at least two distinct intersection points with (the boundary of) G_2 (cf. lines **PO**1 and **PO**2), or (ii) if one can find two distinct points of (the boundary of) G_1 such that one of them lies inside G_2, and the other point lies outside G_2 (cf. line **PO**3). Obviously this rules out the possibility of points partially overlapping with any other geometries. Since the intersection and point-in-polygon tests need to be performed for each (boundary) segment and point, respectively, this renders *Partially Overlaps* quadratic in terms of its worst case time complexity and the number of (boundary) line segments of the geometries involved.

Overlaps. Two geometries G_1 and G_2 *overlap*, either if they partially overlap, or if one geometry is part of the other. As both options are quadratic in the number of (boundary) line segments of the geometries involved, the worst-case time complexity of *Overlaps* is also quadratic.

Discrete From. The relation of two geometries being *discrete from* each other cannot really be handled by our formalism. Following the definition of [6], two polygon primitives are considered *discrete from* each other if they are not overlapping. For open regions, i.e. where a region's boundary is not part of the region itself, as considered in [6], this allows two concretizations, namely (i) the case of being disconnected, i.e. where neither the regions nor their boundaries intersect, and (ii) the case of being discrete from each other where the regions do not intersect, but the boundaries do. When considering closed regions, i.e. when a geometry's boundary is considered being part of the geometry, the *Discrete From* relation does not really fit into the hierarchy in Figure 2 anymore. Applying the same definition of non-intersecting interiors and intersecting boundaries of the involved regions *Discrete From* would simply coincide with the *Externally Connected With* relation. Our closest interpretation would be that two geometries are *discrete from* each other if they are located directly side by side, as, e.g., cargo containers tightly put back to back on a wagon. How close two geometries must be to be considered 'directly side by side' obviously is a use case-specific setting, captured by the ϵ parameter in definition **DR**1. The worst case time complexity is then dominated by the *intersects* relation but also inherits the superlinear portion from the buffer operation. Please note that even though we provide a definition for *Discrete From* in Table 2, we consider the practical relevance of this spatial relation as limited and thus do not consider it in the following.

Externally Connected With. To test whether two geometries G_1 and G_2 are *externally connected* one has to verify that whenever a (boundary) point of G_1 lies inside G_2, this point has to be exactly on G_2's boundary. Accordingly, the

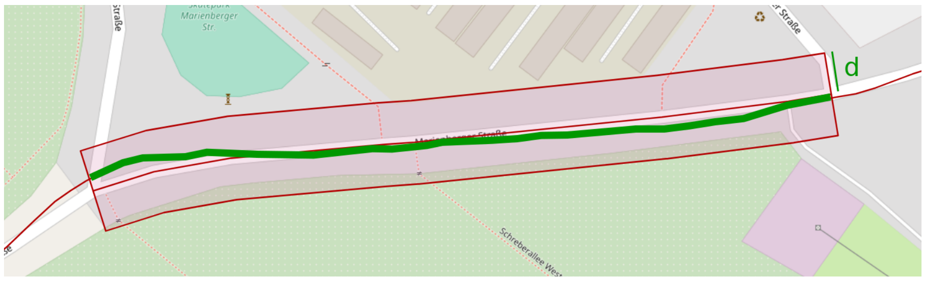


Figure 5: Example of a move (represented by the line string running from right to left) *running along* a road segment (represented by the line string in the center). The calculation of the *Runs Along* relation is based on intersection of the move line segment with the buffer (with offset d) of the road segment line string. The move line string *runs along* the road line segment if the intersection of both (thick green line) has a length greater than $2 \cdot d$.

point-in-polygon test has to be performed for each point in (the boundary of) G_1 and thus, the worst case time complexity for deciding the *Externally Connected With* relation is quadratic in the number of segments of the geometries involved.

Runs Along. The chosen definition of the *Runs Along* relation is motivated by the goal to be able to handle noisy trajectories and learn concept expressions for movement data. Since it is unlikely that the points of GPS traces are exactly *on* a line string, e.g. representing a bike lane, a robust definition with respect to spatial inaccuracies is crucial to guarantee meaningful spatial concept learning. First of all, this relation can only hold between two line strings, e.g. a movement line string G_M running *along* a line string G_S representing a street, side walk, river, etc. The main idea is to consider the buffer of the line string G_S, a movement line string *runs along*, i.e. the buffer of, e.g., a street, side walk, or river line string. The buffer offset corresponds to the degree of inaccuracy one wants to be able to handle and is a use case-specific setting. Further, it is assumed that the line string G_M representing a movement is longer than the line string G_S it runs along and thus has at least two intersection points with the buffer of G_S – ideally exactly two, one on each front side (cf. Figure 5). Now, if one cuts the movement line string G_M at two consecutive intersection points, say $\mathsf{G}_{\mathrm{pt}_1}$ and $\mathsf{G}_{\mathrm{pt}_2}$, (represented by the expression $\mathsf{G}_{M\vdash \mathsf{G}_{\mathrm{pt}_1},\mathsf{G}_{\mathrm{pt}_2}\dashv}$ in Table 2) it should have a certain length to be considered as *running along* the other line string G_S. We require this length to be greater than twice the buffer offset to rule out (orthogonally) intersecting line strings being counted as line strings running along G_S (cf. Figure 5). Accordingly, this involves the computation of the buffer operation, the *intersects* relation, linear referenced sorting to find consecutive intersection points on G_M, and the computation of a line string's length. Hence, the resulting time complexity is superlinear and depends on the number of segments in G_M and G_S, the number of monotone chains in G_M, and the number of intersection points.

Is Near. To capture the notion of two geometries G_1 and G_2 being *near* each other we make use of the buffer operation. If applied to one of the geometries, say G_1, then G_1 and G_2 are near each other, with respect to a use-case specific vicinity threshold δ, if $\overset{\leftrightarrow}{\circ}(\mathsf{G}_1, \delta)$ and G_2 intersect. Time complexity-wise this amounts to $O(N \cdot \log(n))$ for the buffer operation, and $O(N \cdot M)$ for the intersection check, with N being the number of segments in G_1, n being the number of monotone chains in G_1, and M being the number of segments in G_2.

Crosses. For a line string to cross another geometry we require that both intersect, and that the line string's start and end point are not inside the geometry to test against. The dominating fundamental operation is the intersection test which makes the computation of the *Crosses* relation quadratic in the number of the geometries' segments. We have to note that since our definition does not rule out points to be crossed, the *Crosses* relation might coincide e.g. with the *Connected With* relation. However, for geospatial concept learning, especially for cases involving motion data, we considered it useful and more intuitive to be able to state the crossing of spatial features modeled as simple points, e.g. to state that someone crossed a gate.

Having introduced the formal foundations of how to compute whether a certain spatial relation holds between two geometries, in the next section we will propose how to embed this formalism into Description Logics. This concerns the modeling of spatial entities in a domain of discourse and their underlying geometries, as well as the representation of spatial relations themselves.

4. Spatial Inference on Geospatial Polygon Data in Description Logics

Description Logics are a family of languages for representing knowledge of a considered domain of discourse, and furthermore provide means for reasoning about it. A knowledge base expressed in a Description Logic language usually comprises a defined terminology, called *TBox*, providing the vocabulary for describing the domain of discourse. This vocabulary can be used to make assertions about individuals of the considered domain, which are stored in the knowledge base's *ABox*. The vocabulary includes *concepts*, also called *classes*, and *roles*. Whereas concepts represent sets of individuals, roles stand for relations between individuals. Besides this, a Description Logic language provides a set of constructors to build complex concepts and roles.

Concepts are referred to via capital letters A, B, C, D (optionally with indexes). In particular, A and B shall denote *atomic*, i.e. non-complex, concepts. The set of all atomic concept names contained in a knowledge base is given by N_C. Roles are named by the upper case letters R, S (optionally with indexes), and the lower case letters a, b (optionally with indexes) represent individuals. The sets N_R and N_I contain, respectively, all role names, and the names of all named individuals that occur in a knowledge base. The Description Logic language considered here allows one to form concepts according to the following syntax rule:

$$
\begin{array}{lll}
C, D \longrightarrow & A \mid & \text{(atomic concept)} \\
& \top \mid & \text{(universal concept)} \\
& \bot \mid & \text{(bottom concept)} \\
& \neg C \mid & \text{(negation)} \\
& C \sqcap D \mid & \text{(intersection)} \\
& C \sqcup D \mid & \text{(union)} \\
& \forall R.C \mid & \text{(value restriction)} \\
& \exists R.C \mid & \text{(existential restriction)} \\
& \geq n\ R.C \mid & \text{(qualified number restriction)} \\
& \leq n\ R.C \mid & \text{(qualified number restriction)} \\
& = n\ R.C \mid & \text{(qualified number restriction)} \\
& \{a_1, \dots, a_n\} & \text{(nominals)}
\end{array}
$$

The semantics of a knowledge base $\mathcal{K}$ is defined by an interpretation $\mathcal{I}$ comprising the non-empty set $\Delta^{\mathcal{I}}$ called the domain of the interpretation, and an interpretation function $\cdot^{\mathcal{I}}$. The interpretation function assigns to every concept C a set $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to every role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

$$
\begin{array}{rcl}
\top^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} & = & \emptyset \\
(\neg C)^{\mathcal{I}} & = & \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} & = & C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} & = & \{a^{\mathcal{I}} \in \Delta^{\mathcal{I}} | \forall b^{\mathcal{I}}.(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \rightarrow b^{\mathcal{I}} \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} & = & \{a^{\mathcal{I}} \in \Delta^{\mathcal{I}} | \exists b^{\mathcal{I}}.(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \wedge b^{\mathcal{I}} \in C^{\mathcal{I}}\} \\
(\geq n\ R.C)^{\mathcal{I}} & = & \left\{a^{\mathcal{I}} \in \Delta^{\mathcal{I}} \mid |\{b^{\mathcal{I}} | (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \wedge b^{\mathcal{I}} \in C^{\mathcal{I}}\}| \geq n\right\} \\
(\leq n\ R.C)^{\mathcal{I}} & = & \left\{a^{\mathcal{I}} \in \Delta^{\mathcal{I}} \mid |\{b^{\mathcal{I}} | (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \wedge b^{\mathcal{I}} \in C^{\mathcal{I}}\}| \leq n\right\} \\
(= n\ R.C)^{\mathcal{I}} & = & \left\{a^{\mathcal{I}} \in \Delta^{\mathcal{I}} \mid |\{b^{\mathcal{I}} | (a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}} \wedge b^{\mathcal{I}} \in C^{\mathcal{I}}\}| = n\right\} \\
\{a_1, \dots, a_n\}^{\mathcal{I}} & = & \{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}
\end{array}
$$

Besides this so called *abstract* domain $\Delta^{\mathcal{I}}$ Description Logics were extended to support *concrete* domains like real numbers, integers, strings etc. We will introduce the polygon geometry domain here and refer to [43] for definitions of further concrete domains. To relate an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to its polygon geometry information $\mathsf{G}_a \in \mathfrak{G}$ we introduce the concrete role G with $G^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \mathfrak{G}$.

Apart from declaring a vocabulary, a knowledge base's TBox may contain statements about how concepts and roles relate to each other. One kind of such TBox statements are subsumption axioms $C \sqsubseteq D$, or $R \sqsubseteq S$ expressing that the concept C is a sub-concept of D, i.e. $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and R is a sub-role of S, i.e. $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$, respectively. Furthermore, one can express concept and role equivalences, or name complex concepts by means of definitions like $\mathsf{Mother} \equiv \mathsf{Woman} \sqcap \exists \mathsf{hasChild}.\mathsf{Person}$. Another kind of TBox axiom are role *domain* and *range* axioms. For an ABox assertion $R(a, b)$, stating that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$, the role domain declaration $\exists R.\top \sqsubseteq C$ allows to infer that $C(a)$, i.e. $a^{\mathcal{I}} \in C^{\mathcal{I}}$. A

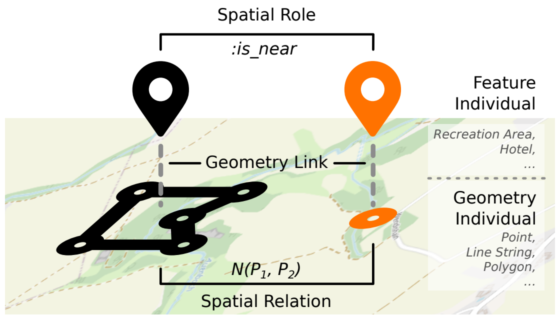


Figure 6: Overview of the relation between spatial features and geometries

corresponding range declaration $\top \sqsubseteq \forall R.D$, on the other hand allows to infer that $D(b)$, i.e. $b^{\mathcal{I}} \in D^{\mathcal{I}}$.

Following the suggestions of the *Basic Formal Ontology* approach [44] a *spatial feature* is a continuant (i.e. something that “continue[s] to exist through time” [44]) that can either be independent and material (e.g. buildings, statues, roads), independent and immaterial (e.g. countries, administrative areas) or a generically dependent continuant (e.g. any information artifact), and has a spatial region (be it 0-, 1-, or 2-dimensional) associated with it. A spatial feature’s geometry information is expressed in the concrete domain $\mathfrak{G}$ and is not directly attached to the feature individual itself. To be able to make assertions about a feature’s spatial extension a dedicated *geometry* individual is introduced which carries the region information. The role $R^{\mathcal{I}}_{\text{📍}:\text{⬠}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ expressing the association between a spatial feature and its geometry is termed *geometry link*. A geometry link can be an atomic role or a role chain. However, we assume the hasGeometry role defined in the GeoSPARQL specification [23] as a default which is declared as a relation between features and geometries. Allowing the geometry link to be an arbitrary role (chain) shall merely ensure flexibility to also support knowledge bases which make use of proprietary vocabularies or follow different modeling approaches.

To better distinguish spatial features from geometries, a dedicated notation is used. Individuals representing spatial features are expressed as lower-case letters with a map marker (optionally with indexes) $a^{\text{📍}}, b^{\text{📍}}$ with $a^{\text{📍}\mathcal{I}}, b^{\text{📍}\mathcal{I}} \in \Delta^{\mathcal{I}}$. By convention, all feature individuals are instances of a dedicated spatial feature class $C^{\text{📍}}$ with $C^{\text{📍}\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$. Geometry individuals are denoted by lower-case letters with a geometry icon (again, optionally with indexes) $a^{\text{⬠}}, b^{\text{⬠}}$ with $a^{\text{⬠}\mathcal{I}}, b^{\text{⬠}\mathcal{I}} \in \Delta^{\mathcal{I}}$. We expect the geometry link role $R_{\text{📍}:\text{⬠}}$ to be functional and define the function $f_{\text{📍}:\text{⬠}}(a^{\text{📍}\mathcal{I}})$ to return a spatial feature’s geometry following the geometry link. Again, we assume that the concrete role G relating a geometry individual to its polygon geometry value $\mathsf{G} \in \mathfrak{G}$ is functional and introduce the function $f_{\mathfrak{G}}(a^{\text{⬠}})$ to return this polygon G.

As mentioned above, we follow the approach of [19] and aim to model spatial relations as relations between individuals, not as relations defined on concrete domains. Moreover, in our conceptualization, spatial relations are defined between spatial features, not on the actual geometry individuals. We argue that this yields more intuitive expressions in the concept learning setting as one can come up with expressions like $\exists\mathsf{isNear.Station}$, $\forall\mathsf{runsAlong.CycleLane}$ which are shorter and better align with the usage of such spatial relations in natural language. So, given a knowledge base containing spatial information expressed in the way described above we extend the set of role names N_R by the set $N_R^{\text{📍}}$ such that for each spatial relation $S \in \mathsf{S} = \{\mathbf{C}, \mathbf{DR}, \mathbf{O}, \mathbf{P}, \mathbf{P}^{-1}, \ldots\}$ defined in Table 2 $N_R^{\text{📍}}$ contains a new role R_S. Further, we extend the TBox to also comprise domain and range declarations for all $R_S \in N_R^{\text{📍}}$. For all R_S domain and range are set to $C^{\text{📍}}$. In addition to the assertions stated in the ABox of the given knowledge base we introduce an additional set of assertions, ABox$^{\text{📍}}$. For each spatial relation $S \in \mathsf{S}$ the ABox$^{\text{📍}}$ then contains spatial role assertions $R_S(a^{\text{📍}}, b^{\text{📍}})$ that hold iff $S(f_{\mathfrak{G}}(f_{\text{📍}:\text{⌑}}(a^{\text{📍}\mathcal{I}})), f_{\mathfrak{G}}(f_{\text{📍}:\text{⌑}}(b^{\text{📍}\mathcal{I}})))$ holds. An overview of the relations between spatial features and geometries is given in Figure 6.

Having introduced means to express when a certain spatial relation holds between individuals of a domain of discourse, we can now extend the concept learning formalism to also use the spatial context of individuals to inductively learn classifying concept descriptions.

5. Spatial Concept Learning

The main concept learning problem considered here is to inductively learn a (usually complex) concept describing a set of observed individuals from N_I. Given

- the set $N^+ = \{a_1^+, a_2^+, \ldots\} \subseteq N_I$ of *positive examples*, and
- the set $N^- = \{a_1^-, a_2^-, \ldots\} \subseteq N_I$ of *negative examples*

a target concept C_t shall be learned, such that for an underlying knowledge base $\mathcal{K}$

- $\mathcal{K} \vDash C_t(a_i^+)$ for all $a_i^+ \in N^+$, and
- $\mathcal{K} \nvDash C_t(a_i^-)$ for all $a_i^- \in N^-$.

To achieve this, we follow a refinement-based approach for concept learning [36] and extended the existing CELOE algorithm [37] to account for the specifics of spatial data and the spatial inference considerations introduced in the previous section. A refinement operator is defined as follows:

Definition 5.1 (Refinement Operator)**.** Given a *quasi-ordered* space $(\Sigma, \preceq)$, a downward (upward) refinement operator ρ is a mapping from Σ to its power set 2^Σ such that for any $C \in \Sigma$ we have that $C' \in \rho(C)$ implies $C' \preceq C$ $(C \preceq C')$. C' is called a *specialization* (*generalization*) of C.

The quasi-ordered space considered here is $(\mathcal{ALCHOQ}(D), \sqsubseteq)$, i.e. the Description Logic language introduced in Section 4 and concept subsumption $\sqsubseteq$. CELOE uses an iterative *downward refinement* approach, i.e. starting with a very general concept C_t, in each step a number of more specific concepts C_t' are derived, such that $C_t' \sqsubseteq C_t$. Respective refinement rules can consider the concept hierarchy or apply concept or role constructors. To give a few examples of such refinement steps, assuming the concept subsumption $D \sqsubseteq C_1$, the role subsumption $S \sqsubseteq R$, and an arbitrary concept C_2, the following derivations are valid downward refinements steps ($\rightsquigarrow$ denotes a refinement step):

$$\begin{array}{lll} C_1 \rightsquigarrow D & C_1 \rightsquigarrow (C_1 \sqcap C_2) & \exists R.C_1 \rightsquigarrow \exists S.C_1 \\ \exists R.C_1 \rightsquigarrow \exists R.D & \exists R.C_1 \rightsquigarrow (\exists R.C_1 \sqcap C_2) & \top \rightsquigarrow C_2 \end{array}$$

Those refinements that improve a certain learning heuristic score (e.g. accuracy, concept complexity) will then be considered in future refinement rounds whereas poor refinements not improving the currently best solution, or with a score below a certain threshold, will be discarded. For our spatial concept learning approach we extended the refinement operator ρ introduced in [37] with the following rule:

$$\rho_1^{\text{📍}}(C) = \{\exists R_S.C^{\text{📍}} | C \equiv C^{\text{📍}} \wedge R_S \in N_R^{\text{📍}}\} \cup \{\forall R_S.C^{\text{📍}} | C \equiv C^{\text{📍}} \wedge R_S \in N_R^{\text{📍}}\} \qquad (1)$$

This means that each spatial feature individual is in some way related to at least another spatial feature individual via each of the spatial roles, and only to spatial feature individuals. For the *Part Of* relation, this would translate to the statement that a spatial feature is something that is *part of* some other spatial feature. This rule follows the intuition behind Tobler's 'First Law of Geography' [45] claiming that usually a spatial feature is somehow spatially related to its environment. This holds in particular in Description Logics where the open world assumption is applied. This means that the absence of an assertion does not imply that the assertion does not hold, but just that it is unknown whether it holds. Accordingly, by applying $\rho_1^{\text{📍}}$ we add existential and universal restrictions on the introduced spatial roles in $N_R^{\text{📍}}$.

Further, the roles in $N_R^{\text{📍}}$ also extend the role hierarchy reflecting the RCC hierarchy illustrated in Figure 2 which is taken into account by means of the following refinement rule:

$$\rho_2^{\faMapMarker}(\exists R.C) = \begin{cases} \begin{array}{lll} \{\exists R_{\mathbf{O}}.C, & \exists R_{\mathbf{P}}.C, & \exists R_{\mathbf{P}^{-1}}.C, \\ \exists R_{\mathbf{PP}}.C, & \exists R_{\mathbf{PP}^{-1}}.C, & \exists R_{\mathbf{PO}}.C, \\ \exists R_{\mathbf{TPP}}.C, & \exists R_{\mathbf{NTPP}}.C, & \exists R_{\mathbf{EQ}}.C, \\ \exists R_{\mathbf{TPP}^{-1}}.C, & \exists R_{\mathbf{NTPP}^{-1}}.C, & \exists R_{\mathbf{EC}}.C, \\ \exists R_{\mathrm{Cr}}.C\} & & \end{array} & \text{if } R = R_{\mathbf{C}} \\ \\ \begin{array}{lll} \{\exists R_{\mathbf{P}}.C, & \exists R_{\mathbf{P}^{-1}}.C, & \exists R_{\mathbf{PP}}.C, \\ \exists R_{\mathbf{PP}^{-1}}.C, & \exists R_{\mathbf{PO}}.C, & \exists R_{\mathbf{TPP}}.C, \\ \exists R_{\mathbf{NTPP}}.C, & \exists R_{\mathbf{EQ}}.C, & \exists R_{\mathbf{TPP}^{-1}}.C, \\ \exists R_{\mathbf{NTPP}^{-1}}.C, & \exists R_{\mathbf{EC}}.C, & \exists R_{\mathrm{Cr}}.C\} \end{array} & \text{if } R = R_{\mathbf{O}} \\ \\ \{\exists R_{\mathbf{PP}}.C, \exists R_{\mathbf{TPP}}.C, \exists R_{\mathbf{NTPP}}.C, \exists R_{\mathbf{EQ}}.C\} & \text{if } R = R_{\mathbf{P}} \\ \\ \begin{array}{ll} \{\exists R_{\mathbf{PP}^{-1}}.C, & \exists R_{\mathbf{EQ}}.C, \\ \exists R_{\mathbf{TPP}^{-1}}.C, & \exists R_{\mathbf{NTPP}^{-1}}.C\} \end{array} & \text{if } R = R_{\mathbf{P}^{-1}} \\ \\ \{\exists R_{\mathbf{TPP}}.C, \exists R_{\mathbf{NTPP}}.C\} & \text{if } R = R_{\mathbf{PP}} \\ \\ \{\exists R_{\mathbf{TPP}^{-1}}.C, \exists R_{\mathbf{NTPP}^{-1}}.C\} & \text{if } R = R_{\mathbf{PP}^{-1}} \\ \\ \{\exists R_{SN}.C, \exists R_{EN}.C\} & \text{if } R = R_{\mathrm{N}} \end{cases} \qquad (2)$$

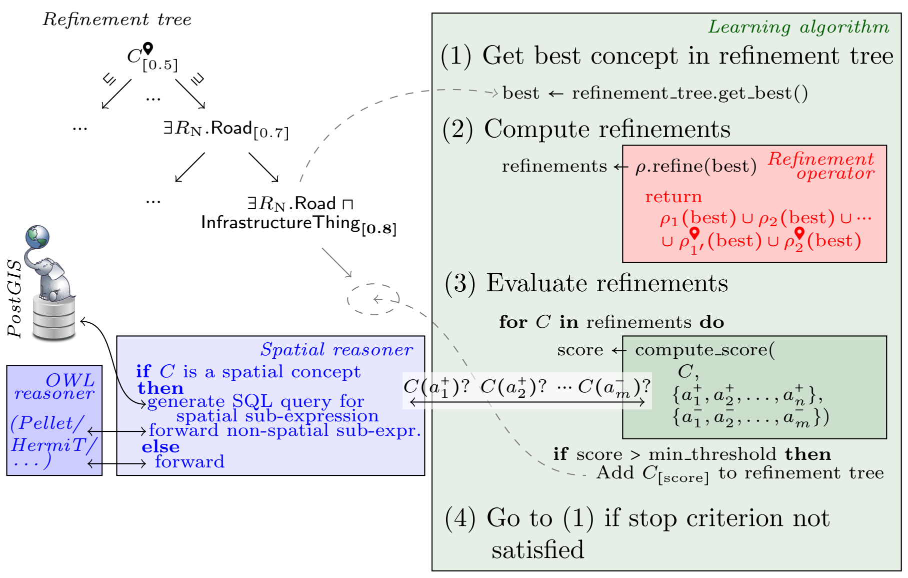
An analogue refinement rule was added for concepts of the form $\forall R.C$.

Computational Complexity of Spatial Concept Learning. The concept learning approach introduced above can be seen as an iterative 'generate-and-test' procedure where (spatial) concept expressions generated by the refinement operator are evaluated with respect to their fitness for discriminating between positive and negative examples. The additional refinement rules $\rho_1^{\faMapMarker}$ and $\rho_2^{\faMapMarker}$ have no impact on the overall learning complexity as they only require a concept equivalence check for $\rho_1^{\faMapMarker}$ which does not involve any spatial reasoning, and a check of the role name in $\rho_2^{\faMapMarker}$ which does not need any automated reasoning at all. However, instance checking, being the core of many of the evaluation metrics (e.g. accuracy, F_1-score, etc.), needs a closer consideration. For any concept expressions not containing any existential, universal or number restrictions on a *spatial role* $r \in N_R^{\faMapMarker}$, the general complexity implications for the supported Description Logic language apply. In our case, the extended downward operator $\rho^{\faMapMarker}$ generates concepts from $\mathcal{ALCHOQ}$ and further supports concrete roles. To the best of our knowledge, the complexity class for instance checking in $\mathcal{ALCHOQ}$ is unknown, however, for the less expressive language $\mathcal{ALCOQ}$ it is already PSpace complete (with respect to the knowledge base size). For any sub-concept being an existential, universal or number restriction on a spatial role $R_S \in N_R^{📍}$, for any instance of the respective role filler D, all members of the spatial relation R_S (and its super-roles $R_{S\uparrow}$) need to be found. So, an operation being quadratic on the number of segments of the involved features' geometries would be executed $|\{a|D(a)\}| \cdot \left|\{b|C^{📍}(b)\}\right| \cdot \left|\{R_{S\uparrow}|R_S \sqsubseteq R_{S\uparrow} \wedge R_{S\uparrow} \in N_R^{📍}\}\right|$ times. In practical terms, this computation of all members of a given individual and spatial relation lends itself for caching, such that in the long run the lookup of the related spatial features for a spatial role R_S is constant in most of the cases, given that the underlying knowledge base does not change and the cache size is big enough to stabilize on an acceptable cache hit rate.

Myopia in Spatial Concept Learning. As already mentioned above, one particular characteristic regarding spatial data is that usually spatial features are somehow spatially related to other spatial features, or as Tobler put it in his 'First Law of Geometry': "everything is related to everything else, but near things are more related than distant things" [45]. This has some implications on the refinement-based concept learning approach. Taking an arbitrary refinement for a concept C with $C \equiv C^{📍}$, e.g. $\rho_1^{📍}(C) = \exists R_{\mathrm{N}}.C^{📍}$, i.e. refining a class covering spatial feature individuals to the class of individuals that are *near* some spatial feature individual, this is unlikely to increase the heuristic score and thus might be ignored in future refinements rounds. Usually this rather generic spatial refinement is not discriminative with respect to the positive and negative examples used in the respective concept learning setting and thus could well be discarded and ignored on future refinement rounds. However, there might be a concept $D \sqsubset C^{📍}$ which could lead to a discriminative concept expression $\exists R_{\mathrm{N}}.D$ if a further refinement step was performed. To give a more illustrative example, consider the set of positive examples are individuals representing traffic lights, and the set of negative examples being arbitrary spatial features not being a traffic light. Having the refinement $\exists R_{\mathrm{N}}.C^{📍}$ (i.e. something that is near a spatial feature) would likely cover positive and negative examples to the same extent and thus lead to a poor score. Nonetheless, if we had the subsumption axiom $\mathsf{Road} \sqsubset C^{📍}$ in our TBox, a further refinement of $\exists R_{\mathrm{N}}.C^{📍}$ would lead to $\exists R_{\mathrm{N}}.\mathsf{Road}$ which would in fact give a distinctive concept description of traffic lights. To overcome this myopic behavior, we modify $\rho_1^{📍}$ to perform a multi-step refinement as follows:

$$\begin{aligned}\rho_{1'}^{📍}(C) = &\{\exists R_S.C_\downarrow^{📍}|C \equiv C^{📍} \wedge R_S \in N_R^{📍} \wedge C_\downarrow^{📍} \in \rho(C^{📍})\} \cup \\ &\{\forall R_S.C_\downarrow^{📍}|C \equiv C^{📍} \wedge R_S \in N_R^{📍} \wedge C_\downarrow^{📍} \in \rho(C^{📍})\}\end{aligned} \tag{3}$$

Here $C_\downarrow^{📍}$ stands for any direct sub-concept of the spatial feature concept. Hence, in the above example it would not need two refinements, as in $C^{📍} \rightsquigarrow \exists R_{\mathrm{N}}.C^{📍} \rightsquigarrow \exists R_{\mathrm{N}}.\mathsf{Road}$, but it could be directly derived that $C^{📍} \rightsquigarrow \exists R_{\mathrm{N}}.\mathsf{Road}$ which remedies myopia in the concept learning process.



Refinement tree
$C_{[0.5]}$
$\sqsupseteq$
$\sqsupseteq$
...
...
$\exists R_{\mathrm{N}}.\mathsf{Road}_{[0.7]}$
...
$\exists R_{\mathrm{N}}.\mathsf{Road} \sqcap \mathsf{InfrastructureThing}_{[0.8]}$
PostGIS
OWL reasoner
(Pellet/ HermiT/ ...)
Spatial reasoner
if C is a spatial concept
then
generate SQL query for spatial sub-expression
forward non-spatial sub-expr.
else
forward
$C(a_1^+)?\ C(a_2^+)?\ \cdots\ C(a_m^-)?$
Learning algorithm
(1) Get best concept in refinement tree
best ← refinement_tree.get_best()
(2) Compute refinements
refinements ← ρ.refine(best)
Refinement operator
return
ρ_1(best) ∪ ρ_2(best) ∪ ···
∪ ρ_1'(best) ∪ ρ_2'(best)
(3) Evaluate refinements
for C in refinements do
score ← compute_score(
C,
$\{a_1^+, a_2^+, \ldots, a_n^+\}$,
$\{a_1^-, a_2^-, \ldots, a_m^-\}$)
if score > min_threshold then
Add $C_{[\mathrm{score}]}$ to refinement tree
(4) Go to (1) if stop criterion not satisfied

Figure 7: Overview of the main components of the DL-Learner framework which were extended to provide the spatial concept learning capabilities

6. Implementation

The proposed adaptions for the *Class Expression Learning for Ontology Engineering (CELOE)* approach [37] mentioned above can be summarized as follows: (i) A calculus to embed the inference of spatial relations between feature individuals into Description Logics reasoning, and (ii) extended refinement rules for a refinement-based inductive class expression learning algorithm. Being implemented in the DL-Learner [31] framework for supervised Machine Learning in OWL, RDF and Description Logics, CELOE makes use of certain components as sketched in Figure 7. The main components of interest here are an extended 'spatially aware' reasoning component which implements (i), and an adapted refinement operator providing the extended rule set of (ii). Moreover, as mentioned above, we provide a CELOE-based implementation of a learning algorithm component. Figure 7 also sketches CELOE's generate-and-test loop making use of the extended refinement rules and the extended reasoner component.

For the extended reasoning component to infer spatial relations, we make use of the established free software database system PostgreSQL[12] with its PostGIS[13] database extension. To integrate PostGIS' capabilities for running complex queries on (even large scale) spatial information in an optimized manner

[12] `https://www.postgresql.org/`
[13] `https://postgis.net/`

we mirror spatial information of geometry individuals in a PostGIS database when a knowledge source is loaded. Hence, internally all reasoning requests are routed to PostGIS, or to an off-the-shelf Description Logic/OWL reasoner used behind the scenes, or both, depending on whether the request refers to spatial aspects, non-spatial aspects, or both, respectively. In our experiments we used the Pellet[14] OWL reasoner for the non-spatial reasoning part. For the requests referring to spatial inference bits, tailored SQL queries are instantiated from query templates, in accordance with the semantics of the definitions in Table 2, and run against the PostGIS back-end for spatial inference. The extended reasoner component is able to resolve geometry links in both directions to integrate the 'spatial' role assertions between spatial feature individuals into the overall reasoning task.

Even though the PostGIS extension provides a highly optimized query performance for spatial queries, database access still imposes a performance penalty during the concept learning process compared to native memory access. However, we argue that having a specialized and established database management system in place for this task is reasonable especially for bigger spatial datasets, since implementing all the database system features like memory management and indexing structures inside the extended reasoner component would be infeasible. To speed up the spatial inference part of reasoning requests we implemented caching mechanisms in the extended reasoner component to avoid database access where possible. Further, dereferencing the geometry link, back and forth, requires (non-spatial) reasoner access which may be costly, especially in the reverse direction. As such dereferencing is needed for almost all spatial inference operations, individuals related via a geometry link are cached as well.

7. Evaluation

In this section, we provide an overview of different experiments we performed to evaluate the implementation of the proposed spatial concept learning approach. First, we consider the spatial inference part where we concentrate on scalability concerns. In the second part, we report different spatial concept learning scenarios and discuss their outcomes. All experiments were executed on a machine with an Intel Xeon CPU (2.10 GHz, 32 cores) and 128 GB of RAM. We ran experiments with the latest version of the DL-Learner framework with its spatial concept learning extensions available in the *feature/spatial2* branch on GitHub[15].

7.1. Spatial Inference

Goal. In Section 3, we defined 19 spatial relations and estimated their worst case time complexities (cf. Table 3). The main outcome was that the computation of most of the relations requires quadratic time in terms of the overall

[14] `https://github.com/stardog-union/pellet`

[15] `https://github.com/SmartDataAnalytics/DL-Learner/tree/feature/spatial2`

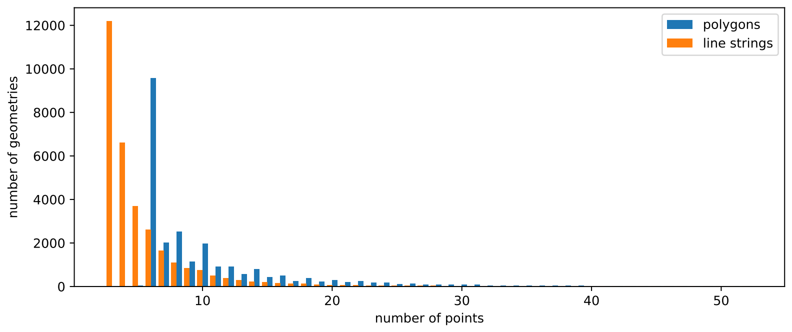


Figure 8: Histogram showing the occurrences of geometries of different sizes (with respect to the number of defining points) in the OpenStreetMap dataset of Saxony, Germany

number of segments of the geometries involved. On the more favorable side, there are exceptions like *Identical With*, *Starts Near* and *Ends Near* which only require linear time in the number of segments of one of the involved geometries. More demanding relations require a time complexity slightly worse than quadratic effort in the overall number of segments of the geometries involved. To get a more realistic impression on actual scalability concerns of the extended reasoner component, especially with respect to the PostGIS back end and its implementations of functions for retrieving spatial relations [46], we ran experiments on synthetic datasets with increasing geometry sizes. We measured the runtimes to retrieve all pairs of features spatially related via a certain spatial role.

Dataset. We generated a range of datasets of different geometry sizes and computed all spatial relations. To mimic real world use cases we first examined data from OpenStreetMap. As an example, Figure 8 shows the histograms for sizes of line string and polygon geometries of the dataset covering the German state Saxony. One can see that the line string and polygon sizes form distributions with a long tail. Hence, for our scalability experiments we considered geometry sizes from 3 to 20 defining points, which covers characteristics of the majority of geometries, leaving bigger and less likely geometry sizes out. For each such dataset, a point, a line string, or a polygon was created with the same probability of $\frac{1}{3}$ until a total of 500 geometries was reached. Each point was picked uniformly from a rectangle with a defined center point and dimensions. For line strings the first geo-coordinate was picked in the same manner. However, all the other geo-coordinates were picked from a neighborhood defined by a normal distribution having the previously generated coordinate as its center and $\mu = 0.0005$ degrees (both for longitude and latitude). To generate smooth line strings and avoid self-intersections, all generated coordinates are appended to the line string based on their shortest distance to the currently last coordinate. The number of coordinates is always fixed according to the considered geometry size, i.e. 3, 4, . . . , 20. The start coordinates of random polygons to

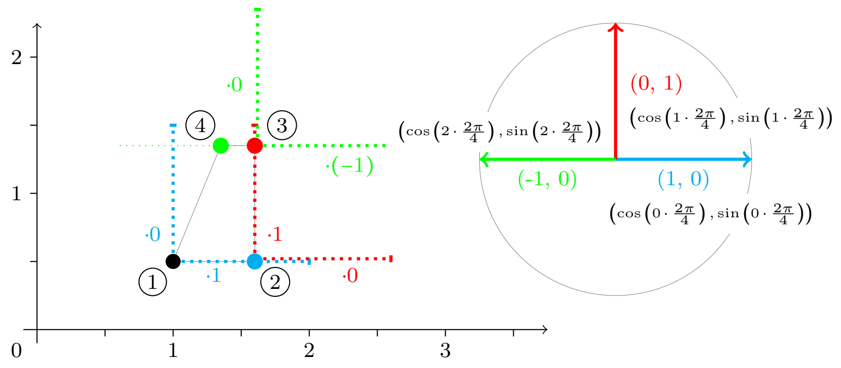

(0, 1)
$\left(\cos\left(1\cdot\frac{2\pi}{4}\right),\sin\left(1\cdot\frac{2\pi}{4}\right)\right)$
$\left(\cos\left(2\cdot\frac{2\pi}{4}\right),\sin\left(2\cdot\frac{2\pi}{4}\right)\right)$
(-1, 0)
(1, 0)
$\left(\cos\left(0\cdot\frac{2\pi}{4}\right),\sin\left(0\cdot\frac{2\pi}{4}\right)\right)$
·0
·(−1)
·1
·0
·1
·0
1
2
3
4

Figure 9: Example of generating a random non-intersecting quadrangle. Dotted lines represent offset ranges (in longitude and latitude direction) to randomly choose from. The colored numbers next to the dotted lines are the scaling factors for the respective longitude and latitude offsets. Numbers in circles denote the index of the polygon's boundary points.

generate are chosen in the same way. Offsets are randomly picked between 0 and 1, for the longitude and latitude. Having, e.g., the coordinate $(1.0, 0.5)$ and the randomly picked offsets $(0.62, 0.23)$ the candidate coordinate for the next boundary point would be $(1.62, 0.73)$. However, to ensure that we have mostly non-self-intersecting, convex polygons, we scale the c-th offset's longitude by $\cos\left(c \cdot \frac{2\pi}{n}\right)$ and its latitude by $\sin\left(c \cdot \frac{2\pi}{n}\right)$ where n is the overall number of defining boundary points and c starts with 0. For $n = 4$ this means that the longitude offset of the second coordinate is kept as is, whereas its latitude offset is is squashed to 0; the third longitude offset is squashed to 0, but the third latitude offset is kept as is, and so on. The procedure is sketched in Figure 9. For each geometry size five datasets were generated.

Results. The average runtimes, per geometry size, to infer all feature pairs for which the respective spatial relation holds are reported in Figure 10. The first main outcome of the experiments is that in terms of their required runtime the spatial relations can roughly be grouped into three categories: those that seem to be easy to infer (upper diagram in Figure 10), the midfield (middle diagram in Figure 10), and those that seem to be expensive to infer (lower diagram in Figure 10). However, none of them show clear quadratic behavior which suggests that the worst case time complexities are rarely reached on real world data, or, that the quadratic behavior is not dominant for common geometry sizes, as, e.g., found in OpenStreetMap. This further suggests that the overall approach is applicable in practice despite its demanding theoretical computational characteristics. Other than expected the three main complexity classes reported in Table 3 (linear, quadratic, worse than quadratic) do not clearly match the three classes we found in our evaluation. Whereas the *Identical With* relation seems to be the easiest to infer, which does reflect its theoretical linear worst time complexity we reported in Table 3, the other two relations having linear worst time complexity in theory, namely *Starts Near* and *Ends Near*, appear in the midfield and thus show much worse time complexity behavior

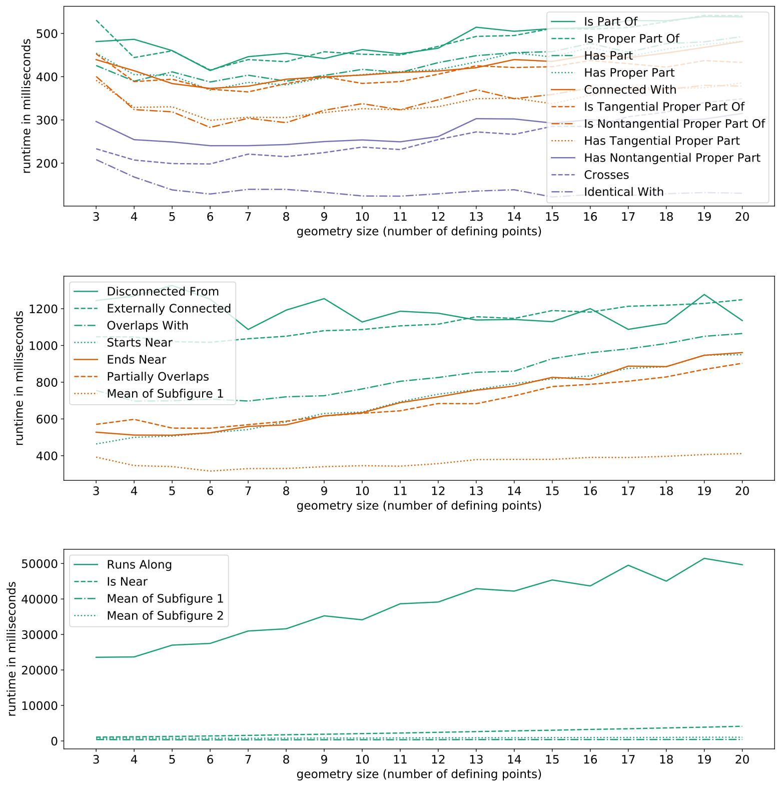
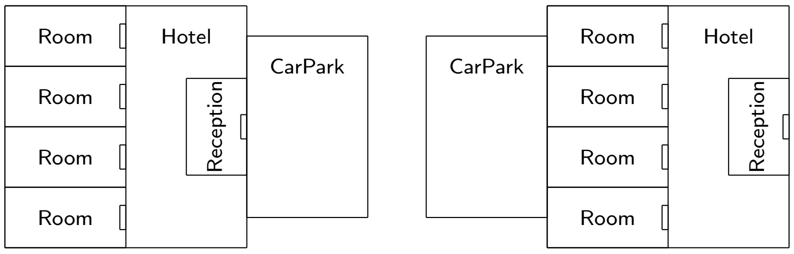


Figure 10: Runtimes for finding all feature pairs for which a certain spatial relation holds. The features' geometries are scaled from 3 to 20 points. The overall dataset sizes are always 500 features. The reported numbers are the average over five different randomly generated datasets per geometry size.

than anticipated. This may be attributed to the fact that we assumed a buffer disc approximation for buffer operations applied to points, which is fixed in the number of defining boundary points. The `ST_Buffer` function of the PostGIS database system we used to implement *Starts Near* and *Ends Near*, however, might provide a less approximative solution which is more expensive, time-wise. Regarding the spatial relations that have the least favorable complexity classes, namely *Runs Along* and *Is Near*, the experiments support our estimations, however rather seem to suggest that in practice the relations can still be inferred in linear time. The upper chart in Table 3 shows an interesting behavior of the PostGIS database system, namely that for some relations the execution times slightly drop as the size of geometries increase. We attribute this behavior to internals of the database system which we did not further investigate.



(a) Car-friendly hotel (b) Car-unfriendly hotel

Figure 11: Example illustrations of hotel environments (small upright rectangles denote doors)

7.2. Spatial Concept Learning

In this section we present experiments demonstrating concept learning with automatic spatial inference as introduced in Section 5. We created two main learning scenarios with positive and negative examples, as well as a background knowledge base with spatial information. We applied the proposed extension of the CELOE algorithm to learn concept descriptions including spatial relations inferred by the reasoner component. For evaluation purposes we concentrate on the accuracy of the learned concept descriptions which serve as binary classifiers. Furthermore, we report runtime measures.

7.2.1. Car-Friendly Hotel Learning Scenario

Goal. In the car-friendly hotel learning scenario we refer to an artificial example environment introduced by Lutz and Miličić for their attempt to perform automatic reasoning with concrete domains and RCC-8 [17]. The main idea is to imagine a hotel with a reception area, rooms and a car park. Examples of both a car-friendly and a 'car-unfriendly' hotel are given in Figure 11. Whereas in Figure 11a one can reach the reception area directly from the car park through the main entrance, in Figure 11b people arriving by car have to walk around the whole hotel to enter. Other examples for car-unfriendly hotels are possible, too. Besides the option of not having a car park, the car park could be externally connected to the hotel in the north or the south, or not connected at all. Lutz and Miličić provide a target description which can be translated to the following concept in our formalism:

$$\begin{aligned}\mathsf{CarFriendlyHotel} \equiv\, & \mathsf{Hotel} \sqcap \\ & \exists \mathsf{isExternallyConnectedWith}.(\\ & \quad \mathsf{CarPark} \sqcap \\ & \quad \exists \mathsf{isExternallyConnectedWith}.\mathsf{Reception})\end{aligned}$$

with Hotel, Reception and CarPark being sub-concepts of the feature class $C^{\text{📍}}$.

Dataset. To inductively learn this concept expression through positive (i.e. car-friendly) and negative (i.e. car-unfriendly) examples, we wrote a dataset generator which can create either type of hotel and puts them on random geo-coordinates ensuring that no two hotel instances intersect. Since hotels (together

Table 4: Average rankings of those learned concepts being within the top 10 results across all ten folds of the car-friendly hotel learning scenario

Avg. rank.	Concept
1.0	$\exists$isExternallyConnectedWith.($\exists$isExternallyConnectedWith.Reception)
2.0	$\exists$isExternallyConnectedWith.($\exists$isConnectedWith.Reception)
4.8	$\exists$isExternallyConnectedWith.(CarPark$\sqcap$($\exists$isExternallyConnectedWith.Reception))
5.8	$\exists$isExternallyConnectedWith.(CarPark $\sqcap$ ($\exists$isConnectedWith.Reception))
6.8	$\exists$isConnectedWith.(CarPark $\sqcap$ ($\exists$isExternallyConnectedWith.Reception))
7.8	$\exists$isConnectedWith.(CarPark $\sqcap$ ($\exists$isConnectedWith.Reception))
8.8	$\exists$isExternallyConnectedWith.((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\exists$isExternallyConnectedWith.Reception))
9.8	$\exists$isExternallyConnectedWith.((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\exists$isConnectedWith.Reception))

with their rooms, receptions and parking lots) are the only spatial features in this experiment, the target concept simplifies to

$$\begin{aligned}\text{CarFriendlyHotel} \equiv &\exists\text{isExternallyConnectedWith.(}\\ &\qquad\exists\text{isExternallyConnectedWith.Reception)}\end{aligned}$$

which is equivalent to (but shorter than) the target concept description derived from [17]. For our experiment we generated 50 car-friendly and 50 car-unfriendly hotels. The car-unfriendly hotels either have no car park, have the car park externally connected to the hotel, but not to the reception area, or have a car park disconnected from the hotel. Each option is chosen with the same probability of $\frac{1}{3}$.

Results. We ran the spatial concept learning approach presented in Section 5 in a 10-fold cross-validation setting. For each fold we allotted an execution time of 10 seconds. In all folds the ten best learned concepts all achieved a classification accuracy and F_1-score of 1.0. The results occurring in the top 10 results of all folds are shown in Table 4 with their average ranking. It can be seen that the best learned concept across all folds equals the simplified target concept mentioned above. The more complex concept closest to the description derived from [17] was among the top five solutions in all folds (with an average ranking of 4.8). This concept is ranked lower despite having a perfect accuracy because it got a lower score as the learning algorithm favors short and human readable concepts over more complex ones.

To better show the capabilities of the proposed learning algorithm to effectively overcome myopia in spatial concept learning we slightly adapted the experiment setting. We, again created 50 positive and 50 negative examples, however this time all negative examples had a car park externally connected with the hotel. The only distinction would be whether this externally connected car park is by itself externally connected with the reception area, or not (cf. Figure 11). Accordingly, the intermediate result $\exists$isExternallyConnectedWith.$C^{📍}$

Table 5: Top 10 learned concepts of the modified car-friendly hotel learning scenario

Rank.	Concept
1	$\forall$isExternallyConnectedWith.($\exists$isExternallyConnectedWith.Reception)
2	$\forall$isExternallyConnectedWith.($\exists$isConnectedWith.Reception)
3	$\forall$isExternallyConnectedWith.(CarPark$\sqcap$($\exists$isExternallyConnectedWith.Reception))
4	$\forall$isExternallyConnectedWith.(CarPark $\sqcap$ ($\exists$isConnectedWith.Reception))
5	$\forall$isExternallyConnectedWith.((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\exists$isExternallyConnectedWith.Reception))
6	$\forall$isExternallyConnectedWith.((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\exists$isConnectedWith.Reception))
7	$\forall$isExternallyConnectedWith.(((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\forall$startsNear.Room)) $\sqcap$ ($\exists$isExternallyConnectedWith.Reception))
8	$\forall$isExternallyConnectedWith.(((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\forall$startsNear.Room)) $\sqcap$ ($\exists$isConnectedWith.Reception))
9	$\forall$isExternallyConnectedWith.(((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\forall$startsNear.Reception)) $\sqcap$ ($\exists$isExternallyConnectedWith.Reception))
10	$\forall$isExternallyConnectedWith.(((CarPark $\sqcap$ ($\exists$isNear.Reception)) $\sqcap$ ($\forall$startsNear.Reception)) $\sqcap$ ($\exists$isConnectedWith.Reception))

would not be distinctive as it fully covers both the set of positive and negative examples. Hence, a greedy learning algorithm with a myopic behavior would never pass this 'plateau' getting stuck with a rather weak solution.

This setting required a longer execution time than 10 seconds as fully accurate solutions were only found after 15.265 seconds (averaged over the ten folds). So we adjusted the maximum execution time to 30 seconds. In all ten folds the top 10 ranked solutions are the same. Table 5 lists the learned concepts with their rankings. First of all, one can see that the search space was explored in a slightly different way. Instead of focusing on refining $\exists$isExternallyConnectedWith.C all top 10 solutions contain a universal restriction. This would render hotels, not having a car park at all, car-friendly. However, since in this adapted scenario such hotels were not explicitly labeled as negative examples the results are nonetheless correct. Further one can find longer result concepts, ranked from 7 to 10 even with redundant parts. ($\forall$startsNear.Room can only be fulfilled trivially since CarPark cannot start anywhere by the definition of the *Starts Near* relation.) Overall, all top 10 results in all folds had a perfect classification accuracy and the experiment showed that the multi-step refinement rule introduced in Section 5 is indeed effective.

7.2.2. Transportation Mode Detection Learning Scenario

Goal. To sketch the capabilities of our spatial concept learning approach to inductively learn from VGI and spatio-terminological information we set up a scenario with citizen sensing data and background knowledge from the LinkedGeoData project. The goal in this learning scenario is to get a description for commutes performed with public transport. The learned concept should provide a description by means of spatial relations to traffic infrastructure elements captured in OpenStreetMap, as well as points of interest. Compared to the previous experiment the background knowledge base is far more extensive and thus contains much more information not relevant for the task, which increases the search space considerably.

Dataset. We tracked trips within Dresden, Germany, performed with different means of transportation via a GPS tracking mobile app. The means of transportation were *bike*, *bus*, *car*, *tram* and *train*. The duration of each trip was at least several minutes. The respective GPS traces were represented as line string features (with their respective geometries) in RDF with 693.4 segments on average. Due to the time overhead needed to track movements with different means of transportation only a small amount of trips was recorded that served as our positive and negative examples. For the spatio-terminological background knowledge we used spatial data covering the Dresden area from OpenStreetMap and converted it to RDF by means of the mapping definitions of the LinkedGeoData project[16] and the Sparqlify RDB2RDF mapping tool [4]. The generated dataset contains $542,553$ triples and 519 named classes covering amenities, buildings, shops, offices and traffic infrastructure.

Results. We set tram, train and bus rides as positive examples, whereas car and bike rides were used as negative examples. Because of the small set of examples we did not perform cross validations but ran the experiment on the whole data and checked the result manually for plausibility. The maximum execution time of the learning algorithm was set to 30 minutes to account for the big spatio-terminological background data and the longer example line strings. A fully accurate target description (i.e. with an accuracy, recall and F_1-score of 1.0), $\exists\mathsf{startsNear}.\mathsf{PublicTransportThing}$, was found after 15 minutes and 17 seconds. The result is plausible considering the subclasses of PublicTransportThing which comprise Halt, Station, and Platform. Given the flat structure of the LinkedGeoData ontology and thus the lack of a common super-class like, e.g., PublicTransportStop the target description is indeed a meaningful solution.

8. Conclusion

In this work, we introduced a formalism to automatically infer spatial relations that hold in polygon data. The formalism was used to extend state-of-the-art concept learning on spatial RDF data allowing to include such inferred spatial relations in the learned concept descriptions. We showed empirically that even though the time complexities for computing whether relations hold between geometries are worse than linear for most relations, inferring such spatial relations on real world data is usually still feasible. We showed that the extended concept learning approach generates meaningful concept descriptions introducing inferred spatial relations. Moreover, we showed that we can overcome the risk of getting stuck in a non-optimal solution due to myopic behavior.

[16] `https://github.com/GeoKnow/LinkedGeoData`

In future work we plan to further optimize the reasoning performance by extending caching mechanisms and more in-depth analyses of spatial concept expressions. Other than the current generic approach which generates SQL queries against the back end database system for each spatial role, a spatial concept description can be inferred to be empty by its mere structure thus avoiding costly database round trips. A further approach to optimize the usage of the database back end would be to combine queries whenever possible such that they are more selective and thus lead to smaller intermediate results.

Another route of research would concern the invention of spatial concepts based on spatial properties going beyond the spatial relations between features. One such example is given in [7] where a *bay* is described by its convexity and composition structure as “a maximal one-piece sea region which is part of the convex-hull of a land region”. Further, considering aggregates of multiple spatial features and introducing spatial relations like *Amidst*, *Surrounded By* etc. can also serve as meaningful means to produce more expressive concept descriptions, as in “[t]he house [...] in line with the trees.” [47].

References

[1] M. Goodchild, Citizens as sensors: The world of volunteered geography, GeoJournal 69 (4) (2007) 211–221.

[2] C. Capineri, M. Haklay, H. Huang, V. Antoniou, J. Kettunen, F. Ostermann, R. Purves (Eds.), European Handbook of Crowdsourced Geographic Information, ubiquity press, London, UK, 2016.

[3] G. Foody, L. See, S. Fritz, P. Mooney, A.-M. Olteanu-Raimond, C. Costa Fonte, V. Antoniou (Eds.), Mapping and the Citizen Sensor, ubiquity press, London, UK, 2017.

[4] C. Stadler, J. Lehmann, K. Höffner, S. Auer, LinkedGeoData: A core for a web of spatial open data, Semantic Web Journal 3 (4) (2012) 333–354.

[5] J. C. C. McKinsey, A. Tarski, The algebra of topology, Annals of Mathematics 45 (1944) 141–191.

[6] D. A. Randell, Z. Cui, A. G. Cohn, A spatial logic based on regions and connection, in: B. Nebel, C. Rich, W. R. Swartout (Eds.), KR’92: Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc., 1992, pp. 165–176.

[7] A. G. Cohn, B. Bennett, J. Gooday, N. M. Gotts, Qualitative spatial representation and reasoning with the region connection calculus, GeoInformatica 1 (3) (1997) 275–316.

[8] E. Clementini, J. Sharma, M. J. Egenhofer, Modelling topological spatial relations: Strategies for query processing, Computers and Graphics 18 (6) (1994) 815–822.

[9] S. C. Levinson, Spatial language, in: Encyclopedia of Cognitive Science, John Wiley & Sons, Ltd, 2006, pp. 131–137.

[10] W. Nutt, On the translation of qualitative spatial reasoning problems into modal logics, in: W. Burgard, A. B. Cremers, T. Cristaller (Eds.), KI-99: Advances in Artificial Intelligence – 23rd Annual German Conference on Artificial Intelligence Bonn, Germany, September 13–15, 1999 Proceedings, Vol. 1701 of Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg, 1999, pp. 113–124.

[11] Y. Katz, B. Cuenca Grau, Representing qualitative spatial information in OWL-DL, in: B. C. Grau, I. Horrocks, B. Parsia, P. Patel-Schneider (Eds.), Proceedings of the OWLED*05 Workshop on OWL: Experiences and Directions, Vol. 188 of CEUR Workshop Proceedings, 2005.

[12] M. Stocker, E. Sirin, Pelletspatial: A hybrid RCC-8 and RDF/OWL reasoning and query engine, in: P. F. P.-S. Rinke Hoekstra (Ed.), Proceedings of the 6th International Workshop on OWL: Experiences and Directions (OWLED 2009), Vol. 529 of CEUR Workshop Proceedings, 2009.

[13] M. Grigni, D. Papadias, C. Papadimitriou, Topological inference, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (I), 1995.

[14] J. Y. Halpern, Y. Moses, A guide to completeness and complexity for modal logics of knowledge and belief, Artificial Intelligence 54 (3) (1992) 319–379.

[15] J. Renz, B. Nebel, On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus, Artificial Intelligence 108 (1-2) (1999) 69–123.

[16] F. Baader, R. Küsters, F. Wolter, Extensions to Description Logics, in: F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook, 2nd Edition, Cambridge University Press, 2007, Ch. 6.

[17] C. Lutz, M. Miličić, A tableau algorithm for Description Logics with concrete domains and general TBoxes, Journal of Automated Reasoning 38 (1-3) (2007) 227–259.

[18] V. Haarslev, C. Lutz, R. Möller, Foundations of spatioterminological reasoning with Description Logics, in: A. G. Cohn, L. K. Schubert, S. C. Shapiro (Eds.), KR'98: Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann Publishers Inc., 1998.

[19] R. Grütter, B. Bauer-Messmer, Towards spatial reasoning in the Semantic Web: A hybrid knowledge representation system architecture, in: S. I.

Fabrikant, M. Wachowicz (Eds.), The European Information Society, Lecture Notes in Geoinformation and Cartography, Springer Verlag, Berlin, Heidelberg, 2007, pp. 349–364.

[20] M. A. Sherif, K. Dreßler, P. Smeros, A.-C. N. Ngomo, Radon — rapid discovery of topological relations, in: S. Singh, S. Markovitch (Eds.), AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Vol. 1, AAAI Press, 2017.

[21] G. Papadakis, G. Mandilaras, N. Mamoulis, M. Koubarakis, Progressive, holistic geospatial interlinking, in: WWW '21: Proceedings of the Web Conference 2021, ACM, New York, NY, USA, 2021, pp. 833–844.

[22] K. Kyzirakos, M. Karpathiotakis, M. Koubarakis, Strabon: A semantic geospatial DBMS, in: P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, E. Blomqvist (Eds.), The Semantic Web – ISWC 2012, 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Vol. 7649 of Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg, 2012, pp. 295–311.

[23] Open Geospatial Consortium, `https://portal.opengeospatial.org/files/?artifact_id=47664`, OGC GeoSPARQL - A Geographic Query Language for RDF Data (2012).

[24] B. Ganter, R. Wille, Formal Concept Analysis – Mathematical Foundations, Springer-Verlag, 1999.

[25] A. Belfodil, S. Kuznetsov, C. Robardet, M. Kaytoue, Mining convex polygon patterns with formal concept analysis, in: The Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI), 2017, pp. 1425–1432.

[26] S. Muggleton, L. D. Raedt, D. Poole, I. Bratko, P. Flach, K. Inoue, A. Srinivasan, ILP turns 20, Machine Learning 86 (1) (2012) 3–23.

[27] S. Muggleton, C. Feng, Efficient induction of logic programs, in: First International Workshop on Algorithmic Learning Theory, 1990, pp. 368–381.

[28] S. Muggleton, J. Santos, A. Tamaddoni-Nezhad, ProGolem: A system based on relative minimal generalisation, in: L. D. Raedt (Ed.), Inductive Logic Programming – 19th International Conference, ILP 2009, Leuven, Belgium, July 02-04, 2009, Vol. 5989 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, 2009, pp. 131–148.

[29] S. H. Muggleton, J. C. A. Santos, A. Tamaddoni-Nezhad, TopLog: ILP using a logic program declarative bias, in: M. G. de la Banda, E. Pontelli (Eds.), Logic Programming – 24th International Conference, ICLP 2008

Udine, Italy, December 9-13 2008, Vol. 5366 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, 2008, pp. 687–692.

[30] D. Malerba, F. A. Lisi, Discovering associations between spatial objects: An ILP application, in: C. Rouveirol, M. Sebag (Eds.), Inductive Logic Programming – 11th International Conference, ILP 2001 Strasbourg, France, September 9–11, 2001, Vol. 2157 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, 2001, pp. 156–163.

[31] L. Bühmann, J. Lehmann, P. Westphal, DL-Learner – a framework for inductive learning on the Semantic Web, Web Semantics: Science, Services and Agents on the World Wide Web 39 (2016) 15–24.

[32] T. M. Mitchell, Generalization as search, Artificial Intelligence 18 (2) (1982) 203–226.

[33] N. Fanizzi, S. Ferilli, L. Iannone, I. Palmisano, G. Semeraro, Downward refinement in the $\mathcal{ALN}$ Description Logic, in: M. Ishikawa, S. Hashimoto, M. Paprzycki, E. Barakova, K. Yoshida, M. Köppen, D. W. Corne, A. Abraham (Eds.), Fourth International Conference on Hybrid Intelligent Systems (HIS'04), IEEE Computer Society, 2005, pp. 68–73.

[34] F. Esposito, N. Fanizzi, L. Iannone, I. Palmisano, G. Semeraro, Knowledge-intensive induction of terminologies from metadata, in: S. A. McIlraith, D. Plexousakis, F. van Harmelen (Eds.), The Semantic Web – ISWC 2004 – Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Proceedings, Vol. 3298 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Heidelberg, 2004, pp. 441–455.

[35] L. Iannone, I. Palmisano, N. Fanizzi, An algorithm based on counterfactuals for concept learning in the Semantic Web, Applied Intelligence 26 (2) (2007) 139–159.

[36] J. Lehmann, P. Hitzler, Concept learning in Description Logics using refinement operators, Machine Learning Journal 78 (1-2) (2010) 203–250.

[37] J. Lehmann, S. Auer, L. Bühmann, S. Tramp, Class expression learning for ontology engineering, Journal of Web Semantics 9 (1) (2011) 71–81.

[38] Open Geospatial Consortium Inc., `http://portal.opengeospatial.org/files/?artifact_id=25355`, OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture (May 2011).

[39] M. I. Shamos, D. Hoey, Geometric intersection problems, in: 17th Annual Symposium on Foundations of Computer Science, IEEE, 1976.

[40] J. E. Goodman, J. O'Rourke, C. D. Toth (Eds.), Handbook of Discrete and Computational Geometry, 3rd Edition, Chapman and Hall/CRC, 2017.

[41] M. de Berg, O. Cheong, M. van Kreveld, M. Overmars, Computational Geometry – Algorithms and Applications, 3rd Edition, Springer Verlag, Heidelberg, Berlin, 2008.

[42] B. K. Choi, S. C. Park, A pair-wise offset algorithm for 2D point-sequence curve, Computer-Aided Design 31 (1999) 735–745.

[43] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook, 2nd Edition, Cambridge University Press, 2007.

[44] R. Arp, B. Smith, A. D. Spear, Building Ontologies with Basic Formal Ontology, MIT Press, 2015.

[45] W. R. Tobler, A computer movie simulating urban growth in the detroit region, Economic geography 46 (1970) 234–240.

[46] International Organization for Standardization (ISO), https://www.iso.org/standard/60343.html, Information technology – Database languages – SQL multimedia and application packages – Part 3: Spatial, iSO/IEC 13249-3:2016 (2019).

[47] B. Landau, R. Jackendoff, "what" and "where" in spatial language and spatial cognition, Behavioral and brain sciences 16 (1993) 217–217.