A Semantic Web Rule Language for Geospatial Domains

A thesis submitted in partial fulfilment
of the requirement for the degree of Doctor of Philosophy

Philip David Smart

June 2009

Cardiff University
School of Computer Science

ABSTRACT

Retrieval of geographically-referenced information on the Internet is now a common activity. The web is increasingly being seen as a medium for the storage and exchange of geographic data sets in the form of maps. The geospatial-semantic web (GeoWeb) is being developed to address the need for access to current and accurate geo-information. The potential applications of the GeoWeb are numerous, ranging from specialised application domains for storing and analysing geo-information to more common applications by casual users for querying and visualising geo-data, e.g. finding locations of services, descriptions of routes, etc.

Ontologies are at the heart of W3C's semantic web initiative to provide the necessary machine understanding to the sheer volumes of information contained on the internet. For the GeoWeb to succeed the development of ontologies for the geographic domain are crucial. Semantic web technologies to represent ontologies have been developed and standardised. OWL, the Web Ontology Language, is the most expressive of these enabling a rich form of reasoning, thanks to its formal description logic underpinnings .

Building geo-ontologies involves a continuous process of update to the originally modelled data to reflect change over time as well as to allow for ontology expansion by integrating new data sets, possibly from different sources. One of the main challenges in this process is finding means of ensuring the integrity of the geo-ontology and maintaining its consistency upon further evolution.

Representing and reasoning with geographic ontologies in OWL is limited. Firstly, OWL is not an integrity checking language due to it's non-unique name and open world assumptions. Secondly, it can not represent spatial datatypes, can not compute information using spatial operators and does not have any form of spatial index. Finally, OWL does not support complex property composition needed to represent qualitative spatial reasoning over spatial concepts. To address OWL's representational inefficiencies, new ontology languages have been proposed based on the intersection or union of OWL (in particular the DL family corresponding to OWL) with logic programs (rule languages). In this work, a new Semantic Web Spatial Rule Language (*SWSRL*) is proposed, based on the syntactic core of the Description Logic Programs paradigm (DLP), and the semantics of a Logic Program. The language is built to support the expression of geospatial ontological axioms and geospatial integrity and deduction rules. A hybrid framework to integrate both qualitative symbolic information in *SWSRL* with quantitative, geometric information using spatial datatypes in a spatial database is proposed. Two notable features of *SWSRL* are 1) the language is based on a prioritised default logic that allows the expression of default integrity rules and their exceptions and 2) the implementation of the language uses an interleaved mode of inference for on the fly computation (either qualitative or quantitative) deduction of spatial relations.

SWSRL supports an OGC complaint spatial syntax, and a standardised definition of rule meta data. Both features aid the construction, description, identification and categorisation of designed and implemented rules within large rule sets.

The language and the developed engine are evaluated using synthetic as well as real data sets in the context of developing geographic ontologies for geographic information retrieval on the Semantic Web. Empirical experiments are also presented to test the scalability and applicability of the developed framework.

Acknowledgements

I would firstly like to thank my supervisor Alia Abdelmoty for her constant and patient help, support and hard work throughout this research, and for her ability to help me write research papers into the early hours of the morning! I would also like to thank my second supervisor Chris Jones for his valuable comments on my thesis and for giving me an opportunity to further my research career over the last 2 years in Cardiff.

I also would like to thank Nick Fiddian who always kept an interest in my work and allowed me to expand my teaching duties within the department. In addition, I would like to thank the entire office staff who have always gone out of their way to help me and provide a friendly atmosphere in which to work.

Further, I am grateful for the advice and for the many discussions with my colleagues at Cardiff, Jon and Gareth. Also I would like to say a big thank you to Florian who provided constant encouragement for me to keep writing and eventually finish my thesis, and to Steven in Ghent University who provided many fruitful discussions during the time we spent working together.

Finally, I'd like to thank my parents, Viv and Paul, for maintaining a constant interest in my work and supporting me throughout, and my girlfriend Elena, who had to go without any interesting weekends and a holiday for the last 4 years!

CONTENTS

Chapter 1

Introduction

Motivated by new Semantic Web technologies and the ever increasing need to share geo-referenced information on the Internet, this thesis proposes a new geographic ontology paradigm for maintaining, storing and enriching geographic information. The new paradigm is based on existing geographic data model standards and represents a new logical geographic ontology language. The framework is based on the fusion of both the logical geographic ontology language and an ad-hoc Geographic Information System (GIS), to form a complete maintenance and reasoning synergy between qualitative (symbolic relational) and terminological knowledge in the ontology and quantitative (geometric) knowledge in the GIS. The geographic ontology language combines features from existing, tractable knowledge representation languages found suitable for representing and reasoning with geographic knowledge. To accompany the geographic ontology language, a spatial reasoning engine is built to support its execution.

The language and its associated reasoning engine detail how these techniques can be applied to geographic information sharing and retrieval. Synthetic and real world data have been used to populate the geographic ontology in order to test the efficiency of the language, framework and its associated spatial reasoning engine. This chapter motivates the need for such a geographic ontology paradigm and framework, outlines the expected results and contributions of the work, then lastly explains the structure of the thesis.

1.1 MOTIVATION OF THE CURRENT WORK

The world wide web is the largest information resource in the world, containing over 25 billion web pages* of unstructured or semi-structured knowledge. However it is not being used to its full potential. Currently most web information is written using syntactical micro format languages such as HTML. These languages are machine understandable but only for presentation purposes intended for human consumption. To fully unlock the potential of such a large knowledge resource, the meaning of web information needs to be machine understandable. Machine understanding of web content shifts knowledge repositories away from traditional databases onto a new web-base. In order to gain machine understanding semantic markup languages are needed. Agents (human or machine) could then use this vast, cross referenced information resource in a variety of unique ways.

Web pages are often rich in geographic information [239] and can be geo-referenced by the geographic terms that appear on the page [24]. Geo-referencing finds and extracts geographical objects from web documents as well as grounding them to their location in space. For example by mining web pages to extract telephone numbers, post codes or place names. Such information can then be placed into a suitable geographic data model and storage structure for later retrieval.

Already 15% of all web searches contain a geographic search term in the form of a place name [228]. Increasingly therefore, web search engines are becoming spatially aware, allowing both textual and geospatial constraints in queries [143]. A geospatial query is a query with a set of geospatial constraints. Geospatial queries involve place names or geographic concepts, the `what` and `where` components, along with a spatial preposition forming the `relation` component [139, 174]. Less formally, a geospatial query is a; *'query about the spatial relations of entities geometrically defined and located in space'* [71]. Such relationships between features are either determined directly from qualitative relations in the document index, or by computing them from the geometric (location) information for each feature in the document index. Spatial ranking techniques can then be used in addition to standard keyword ranking techniques to order result sets. For example, in a query to find all '*Rugby Stadiums near Cardiff*', those stadiums closer (in spatial

*As of the size of Googles index in 2006, www.cis.upenn.edu/ zives/cis555/slides/I-Crawlers-Sync.ppt distance) to Cardiff would be ranked higher than those further away from Cardiff.

In addition to query tasks, the Web is increasingly being seen as a medium for the storage and exchange of geographic data sets in the form of maps. The simplicity and effectiveness of applications such as Google Earth led to a hype of activity in geo-referencing information on the web.

In all, the geospatial semantic web (GeoWeb) is being developed to address the need for access to current and accurate geo-information [56]. The potential applications of the GeoWeb are numerous, ranging from specialised application domains for storing and analysing geo-information to more common applications by casual users for querying and visualising geo-data, e.g. finding locations of services, descriptions of routes, etc.

Key to the development of the GeoWeb are geographic ontologies (geo-ontologies) [140]. Geo-ontologies provide formal logical semantics to geographical information thus enabling machine understanding and automated reasoning procedures. Building geo-ontologies involves a continuous process of update to the originally modelled data to reflect change over time as well as to allow for ontology expansion by integrating new data sets, possibly from different sources. One of the main challenges in this process is finding means of ensuring the integrity of the geo-ontology and maintaining its consistency upon further evolution. Integrity constraints that maintain the thematic and spatial consistency of geographic information should be an important facet of any complete Geographic Information System (GIS) and geographic ontology [235], but is however often a neglected area of GIS [79]. Developing methods for managing the spatial and thematic integrity of geo-ontologies will contribute towards the development of reliable geographical search engines and to the success of the GeoWeb in general.

This thesis contributes to geographic ontology development, and the geo-web in general, by proposing a new geospatial rule language and framework for the management of geo-ontologies for the purpose of geographic information retrieval. We aim to use the algebraic properties of spatial relations in the ontology, i.e. the transitive nature of containment relations, or the symmetric nature of neighbouring relations, in combination with well defined spatial calculi to derive general rules governing the structure of geographic entities and their interaction in space. In doing so we aim to maintain the consistency of spatial relations in the geo-ontology, and also to derive implicit relations from those raw explicit relations presented qualitatively or quantitatively in the geo-ontology.

1.2 HYPOTHESIS AND RESEARCH QUESTION

References to geographic information on the web are now common in web content and in search queries. The utilisation of this information on the semantic web is deriving the development of geospatial ontologies. Current general standard ontology languages are not designed for spatial or geospatial domains. They are limited with respect to the representation of spatially-referenced information and are not suited to answering spatial queries and retrieval tasks.

The logical consistency of axioms representing spatial ontologies encoded in these languages will not guarantee their spatial consistency, leading to possibly inconsistent knowledge bases. Hence, ontology languages are needed that allow for the natural representation of different dimensions of geospatial knowledge and appropriate frameworks that support the manipulation and reasoning over these ontologies are essential for the effective utilisation of this knowledge on the semantic web.

1.3 APPROACH

The approach adopted in this thesis is as follows. The thesis begins by surveying existing, standardised spatial database models and geographic ontologies with the aim of identifying a suitable representation of geographic knowledge. As part of this survey spatial reasoning is also considered, where suitable types and fragments of spatial calculi are identified for inclusion into a practical geo-ontology paradigm.

From this initial survey older, mature logical knowledge representation paradigms, along with newer, state of the art logical and Semantic Web knowledge representation paradigms are investigated, with aim to uncover suitable logical components necessary for representing geo-ontologies.

Appropriate aspects of these languages are compiled together to form a new rule based geographic ontology (geo-ontology) language which then forms a new geographic knowledge representation paradigm. A geo-ontology framework is proposed that combines the newly developed geo-ontology language with an ad-hoc geographic information system in order to combine both qualitative relational and terminological knowledge with quantitative geometric knowledge. From developing the geo-ontology language and framework, a spatial reasoning engine is implemented as an extension to existing Semantic Web tools.

In order to evaluate the approach, the language, framework and reasoning engine are tested using both synthetic and real world datasets. Real world datasets are constructed from more accurate administrative sources, and from less regulated sources of geographic information from the web. These are augment with synthetically generated geo-ontologies intended to test the runtime behaviour of the system under different conditions. An evaluation of the application of the new language to maintain real world geographic ontologies is show, along with the pragmatic scalability of the implementation engine. This is followed with possible avenues for future work.

1.4 CONTRIBUTIONS

This thesis presents the following novel contributions to the field of geospatial integrity maintenance and geographic information retrieval.

The first half of this thesis evaluates and uncovers features of existing logical knowledge representation paradigms that are suitable for representing the semantics of geographic knowledge bases. From this, a geographic ontology language is designed and developed, based on an integrity checking knowledge representation paradigm that stems from the intersection of Description Logics and Logic Programs.

The language has a spatially oriented syntax compatible with existing geospatial modelling standards. Standardisation compliance is important for the following two reasons. Firstly, it allows the re-use of existing semantics for both geographical features and geographic spatial relationships. Secondly, the syntax will guide non expert users in the proper construction of the individuals that populate the geo-ontology, along with allowing easier authoring of integrity and deduction rules. Rules within the language can be augmented with metadata that describes the type and purpose of the rule. This metadata helps to better identify, categorise and visualise individual rules within large rule sets. The language allows a complete representation of an identified type of spatial calculi that exploits properties of spatial relationships to derive new relations or to check the integrity of existing relations.

Based on the survey of existing works, two major logical extensions are incorporated into the new language. The first allows a form of prioritised defeasible reasoning in integrity rules, such that general or default integrity assumptions can be made while still allowing specific exceptions to these defaults to be represented. The second allows interleaved mixed mode reasoning that permits backward chaining query based rules to be interleaved into continuous entailment forward chaining production rules. This is thought to be useful as it enables dynamic spatial relationship computation on the fly, either derived by rules that represent qualitative spatial calculi, or computed by the location storage system that is accessible from within a proposed geo-ontology maintenance framework.

The geo-ontology maintenance framework combines the relational and terminological capabilities of the developed logical geo-ontology language, with a quantitative location store (typically a spatial database). The geo-ontology is a semantic repository of geospatial information, based on standardised geospatial data models and populated with realistic geographic knowledge from both official and web sources. The location store is then used to hold the geometric footprints of places stored in the geo-ontology. Geo-ontologies represented in the framework are monitored continuously using a set of integrity rules defined in the geo-ontology language. Any violated integrity rule derives an error which is added back to the geo-ontology. Two statistical techniques are then developed for topological errors that help to identify the source of inconsistent topological relations, and then help guide the user in their rectification.

The applicability of the language and framework in maintaining realistic geo-ontologies generated from Wikipedia entries and the official Ordnance Survey administrative boundaries of Cardiff, Glamorgan and South Wales is shown. Furthermore, a Genetic Algorithm is developed to generate synthetic geo-ontologies with a controllable number of geographic regions and relations. These geo-ontologies are then used to perform empirical experiments that test the scalability of the language when run with existing Semantic Web tools and reasoning engines. Testing primarily evaluates the scalability of production systems and logic programming reasoning engines under a real geo-world environment.

1.5 Organisation of the Thesis

This thesis is organised in the following way. Chapter 2 overviews some of the fundamental ideas behind geographic information retrieval and geographic data models. It also surveys existing techniques for qualitative spatial representation and reasoning, which are then later used to help enriched and maintain qualitative spatial relations in the geo-ontology language and maintenance framework.

Chapter 3 overviews the current ontology representation language and paradigm for the Semantic Web, the Web Ontology Language (OWL), and provides a more detailed discussion of the semantics, complexity and definitions of the logical formalisms underpinning all formal ontology and rule languages.

Chapter 4 overviews OWL as an integrity checking language and describes existing research into integrity checking variants of OWL, along with an evaluation of OWL from a geospatial perspective. The chapter then motivates the need for a geospatial rule layer on the Semantic Web and describes, in detail, existing approaches to integrate rules and ontologies. A conclusion is drawn based on the most suitable choice for a geo-ontology language, which is then used and further extended in the reminder of the thesis.

Chapter 5 starts with a survey of more recent proposals to combine spatial logics and ontologies enabling a form of spatioterminological reasoning, and proposes a new framework for the combination of geo-ontologies, rules and spatial databases.

Chapter 6 gives a detailed discussion of the the newly developed Semantic Web Spatial Rule Language ($SWSRL$), which at its core is based on existing work in integrating ontologies and rules, namely Description Logic Programs. The abstract syntax and concrete syntax of the language is given along with the language's general and spatial features and semantics.

Chapter 7 gives a formal discussion of issues and approaches to the representation of topological qualitative spatial reasoning rules in the newly defined language $SWSRL$. Finally, spatial rule sets are defined that form the core maintenance and reasoning mechanisms for any geo-ontology represented in $SWSRL$.

Chapter 8 describes in detail the spatial reasoning engine, which is itself based on existing, mature and scalable logic programming and production system algorithms. Its extensions, necessary to capture all aspects of the language, are shown algorithmically, and two methods to help localise topological inconsistencies in the geo-ontology are shown.

Chapter 9 describes three different techniques to instantiate both synthetic and real world geo-ontologies. These are generated using a Genetic Algorithm, extracted from official administrative sources and extracted from geo-information on the web.

Chapters 10 and 11 illustrate the application of the newly developed geo-ontology language *SWSRL* and its reasoning engine, by using the example geo-ontologies proposed in chapter 9. The results are shown as both realistic application test cases using real world administrative and web mined geo-ontologies, and empirically using the synthetically generated geo-ontologies used to determine realistic runtime behaviour and scalability of the new language and engine.

Chapter 12 then concludes this thesis by summarising its major contributions and findings, and presents an outlook for future research to be carried out in this area.

CHAPTER 2

GEOGRAPHIC INFORMATION SYSTEMS, INTEGRITY MAINTENANCE AND QUALITATIVE REASONING FOR SPACE

Within the context of computer science, an information system is: *'the software and hardware system that supports data-intensive application'* [237]. Information systems typically store, query and manage records of entities within a particular domain or context, for example employee records within a company. Geographic Information Systems (GISs) are a type of information system that deals primarily with geographic information. Geographic information is located in space and can be referenced to the earth. This spatial aspect imparts unique requirements on geographic information that is not contained in other domains [116]. It is often quoted that the first GIS was developed and used as far back as the 19th Century when John Snow established a simple map of the cholera outbreak in London*. However, it was not until Howard Fisher applied the necessary spatial theory to spatial datatypes that the first computer based GIS software started to emerge.

In addition to practical GISs the field of Geographic Information Retrieval (GIR) has emerged within the last decade†. GIR is an important sub field of information retrieval that deals with finding, spatially referencing and then query-

*http://www.nationmaster.com/encyclopedia/Geographic-information-system

†see for example the GIR workshops `http://www.geo.uzh.ch/~rsp/gir08/`

ing geographic information contained within document collections.

As well as simply storing, analysing and retrieving geographic information, the integrity of stored geographic information must be maintained, so as to guarantee the accuracy of any queries over such information. Maintaining the integrity or data quality of geographical information is one of the primary goals of this thesis. Works in this area include, maintaining the consistency of the geometry associated to geographical phenomena (their locational attribute), as well as checking the consistency of qualitative spatial relations between geographical objects.

Recently, geographic information maintenance and storage has been supported and aided by ontologies. Ontologies provide ways to formally define knowledge, work which dates as far back as Aristotle. Recently ontologies have gained interest within computer science research and application thanks largely to the new Semantic Web Initiative started by Sir Tim Berners-Lee [14]. Logical ontologies can be understood by both human and machine. When populated with large knowledge bases of information i.e. facts and axioms, a machine can be used to automatically derive common sense knowledge, akin to that performed by the human cognition. Geographic ontologies represent semantically enriched geographic information and assists geographic information retrieval tasks [141]. In addition, the semantics present in geographic ontologies can also be used to maintain geographic information, for example, a house would normally not be covered by a lake, even if their geometries and spatial relations are accurate and consistent.

Metric representations of spatial relationships between geographic phenomena can also be symbolised qualitatively. Within the last four decades, and particularly since the 1980's, ways of symbolising and then reasoning over qualitative information have emerged. Dealing with a qualitative interpretation of metric (quantitative) information is beneficial when dealing with natural language descriptions of geographic phenomena, or when recourse to quantitative (geometric) computation is expensive.

In this chapter, an introduction to the field of GISs as well as geographic information is given. Recent research into geographic knowledge maintenance is reviewed with an emphasis on work to maintain qualitative geographic knowledge. Ontologies as used on the semantic web are described, along with recent state of the art advancements in geographic ontologies. This is followed by an in depth review of qualitative spatial representation and reasoning.

2.1 Geographic Information Systems

A Geographic Information System (GIS) is fundamentally concerned with the storage, access and spatial analysis of geographical information. Geographical information typically concerns many thousands (if not millions) of geographical features or geofeatures*. A geofeature may be regarded as a encapsulation of knowledge of one particular instance of geographical phenomena, and are typically described using thematic attributes (name, ID etc), and spatial attributes which model the features spatial extent or spatial footprint (its location in space).

As already noted, special to geographic phenomena is location [116], which describes *where* that phenomena is located on the earth's surface. Two distinct spatial representations of geographic phenomena exist, namely the field view and object view[39]. The field view can be represented as a function of some parameter(s) i.e. temperature, population or humidity over some coordinate space. The object view splits space by geometric boundaries into discrete groups of entities or objects, where each object has its own identity, characteristics and attributes. As with the standard geofeature model, in this work we take an object or concept view of geographic information, whereby each geofeature encapsulates information about a single real world object, each object is classified into sets of features, and the spatial extent of geographic phenomena is represented by discrete vector geometry. The efficient storage of geographic features and their associated geometries is a fundamental requirement for a GIS. Typical GIS applications involve information retrieval and spatial analysis tasks, examples of this include, mapping, land management and city planning.

Databases are likely to contain many thousands, if not millions, of geographic instances. Information of that magnitude is expensive to query without a suitable model, clever management system and efficient record indexing methods. A Spatial database system is normally used as part of a GIS. Most spatial databases are extensions to the standard database schemas that provides support to represent and process object geometries. In all, spatial databases add the following functions and features.

*Of note the ISO 19109 definition of a feature is,"a meaningful object in the selected domain of discourse", which is further specialised when dealing wholly in the geographical domain to a geographical feature or geofeature

- Spatial datatypes.
- Optimized and efficient spatial indexing structures for spatial selection and determination of the spatial interaction between two object geometries.
- Querying over a mixture of textual and spatial constraints. Which includes efficient algorithms for performing spatial joins alongside standard joins*.
- Native spatial operators that scale well to complex geometries and large numbers of geometric instances.

2.2 GEOGRAPHIC INFORMATION RETRIEVAL

Geographic Information Retrieval (GIR) deals with the extraction, disambiguation, storage and querying of geospatial information from document collections such as the web [160].

This process can be formalised by a number of steps. Firstly, geographic references within web documents e.g. placenames, post codes or telephone numbers, must be identified (a process known as geo-referencing). These geo-references are then geo-tagged by assigning them a location on the Earth's surface (referred to as its geographic footprint [120], often in the form of a latitude longitude pair). Geo-tagging placename references is not always straight forward as placenames can be ambiguous and have many possible locations. For example the placename *London* could refer to, amongst others, *London* in the UK or *London* in Canada. Hence placename references often need to be disambiguated.

Disambiguation techniques try to identify the exact intended location of a placename in the context of the current document. Placename disambiguation has been approached from two principle directions:

- Rule base - using heuristics to help disambiguated placenames. For example, grounding a placename to the location with the largest population, or the location closest to all others in the document [200].
- Data driven - using machine learning techniques e.g. Support Vector Machines [17]). This approach requires large, accurately pre-tagged corpus's

*A spatial join connects two features or rows in the database together based on the relative locations of the two features to learn from. However large pre-tagged corpus's are not common for GIR applications, although a reference corupus of hand annotated and disambiguated placenames (as a gold standard) was produce by Leidner as part of his PhD thesis [162].

- Spatial - finding the spatial focus of a document, then resolving placename ambiguities using spatial autocorrelation by removing any possible locations that are not contained in a set distance from that focus (see [163] for an overview).

Once identified (geo-referenced), disambiguated and assigned a footprint (geo-tagged)*, each item of geographic information is stored using an appropriate model and indexed. This model and index then forms a searchable structure to perform geospatial queries over. This complete process is illustrated in Figure 2.1.

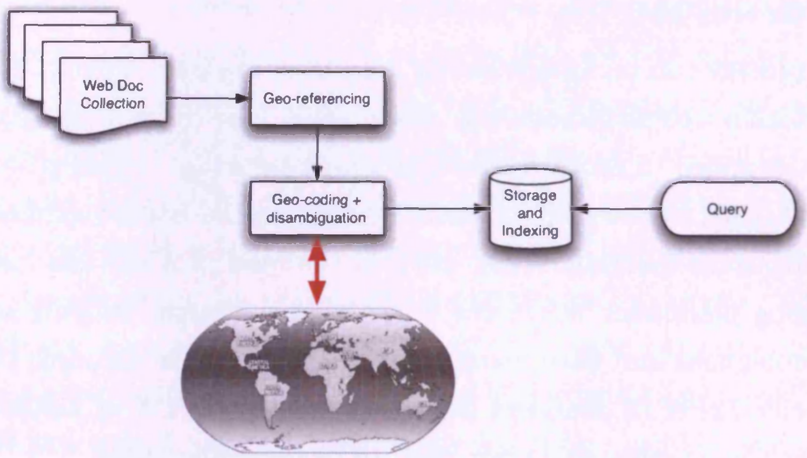


Figure 2.1: Overview of a typical Geographic Information Retrieval System

Current web search engines (although improving) find geographical search terms in a search query as text strings, ignoring the obvious semantics of geographical constraints. For example, a query for *all Hotels in Cardiff*, the spatial preposition *in* indicates a possible geographical search region that can be used to locate relevant resources, in addition to a simple text search for each word in the query string. Increasingly geographically aware search engines are being developed [143]. Enriching search engines with spatial semantics can facilitate more sophisticated retrieval

*For placenames only, this complete process is known as placename resolution [163] techniques. These include; spatial query term expansion, alternative name support, disambiguation and spatially-related place lookup (e.g. places within other places).

GIR tasks can be aided by suitable world knowledge captured in the form of ontologies. Ontologies are used for representing knowledge and its inherent semantics to facilitate geographic information retrieval, as discussed in more detail in the next section.

2.2.1 ONTOLOGIES

The word Ontology is prevalent within philosophy as long back as Socrates and then later Plato and Aristotle. Ontology in the philosophical context is the study of reality and nature of being, a systemic account of existence [109]. Those things that *exist* can also take a formal representation within the context of machine intelligence*. Knowledge commits to an ontology [190] if it adheres to the structure, vocabulary and semantics intrinsic to such ontology (conforms to the ontology definition).

Gruber defines an ontology as: 'An ontology is a specification of a conceptualization' [108]. Guarino separates the meaning of Ontology in a philosophical context to that used in a computer science context [112]. Ontology in a computer science context, spelt with a small 'o', is a logical theory which represents a conceptualisation of all real world concepts. An ontology in practice is a data model which describes objects, their attributes and the relations between those objects.

The best possible outcome for ontology modelling would be one unifying ontology model that represents all objects in the universe of discourse. In practice however, as an unrestricted universe of discourse is too complex to be represented in one ontological model, three levels of ontologies have emerged, namely; the application, domain and upper levels. Upper level ontologies are useful in defining global shared and common vocabulary from which any number of specialised ontologies can be built [173]. Domain ontologies limit the scope of upper level ontologies, and are developed to represent the terms and vocabulary of a given domain [145]. Domain ontologies are not biased per application, where different practical implementations may have their own sets of requirements. Consequently,

*See `http://www-ksl.stanford.edu/kst/what-is-an-ontology.html` an application level ontology is developed which further specialise a domain ontology to include more intricate implementation semantics.

An ontology is any logical theory, although a common representational formats assume a taxonomic class hierarchy. Indeed the web ontology language OWL is roughly based on a set of concept inclusions to construct a concept hierarchy (OWL is discussed in more depth in section 3.1). The popularity of ontologies within computer science has stemmed from their numerous application possibilities. Principally, as used in the context of the semantic web, ontologies as knowledge repositories have been developed to support the primary goal of sharing knowledge in a manner that aids understanding [109, 108].

2.2.2 Geographic Ontologies

Until more recently, geographic ontology development has not kept pace with ontology development in other domains [115]. However, as of the last 5 years, ontologies are being used more and more in the context of GI Science [276]. All geographic concepts are inherently routed in space, as such it is therefore common to refer to a geographic ontology as a geospatial ontology to represent both the geographic and spatial characteristics of the information they represent. Sometimes geospatial ontologies are enriched with temporal characteristics and are hence denoted spatiotemporal ontologies. Geospatial ontologies are specialisations of general, upper level ontologies which deal only with geographic phenomena. Place ontologies (a particular type of geospatial ontology) represent the association between placenames (toponyms) and their locations on the Earth's surface which, as already stated, is often encapsulated as a geofeature.

Geospatial ontologies or any specialization thereof e.g. a place ontology, have many distinguishing characteristics not found in other domains [116, 45], these can be enumerated as:

(a) Geospatial ontologies must be able to handle spatial datatypes and their operations.

(b) Spatial information is often only vaguely described as opposed to a commonly assumed crisp representation, hence a suitable model for vague, partial or incomplete knowledge must be identified.

(c) Geospatial data models are usually associated with large amounts of concrete data instances.

(d) A conceptualization of geographic phenomena leads to conceptual fuzziness, whereby a neat categorization of geographic phenomena is not always possible e.g. what are the difference between streams and lakes.

(e) Geographic concepts may have multiple meanings and may be themselves only weak concepts - where a weak concept does not correspond to a concrete real world object and therefore can not be instantiated e.g. a water body as a weak concept as opposed to a lake which can have individual instances.

(f) People often employ placenames to describe geographical phenomena and natural language expressions to represent relationships between those places. Hence, qualitative as well as quantitative geographic knowledge is required in a geospatial ontology.

Furthermore, geospatial ontologies must not only consider the extension of existing ontologies by locational (geometric) properties and functions, but by how the user perceives, interprets and interfaces with spatial information contained within the ontology [77]. From which naive geography and spatial models based on commonsense cognitive processes have emerged [54, 178].

General geographic ontologies (geo-ontologies) are recognised as essential components in the development of the geospatial semantic web [56] (GeoWeb), where they are used to capture key conceptualisations of geographic domains to facilitate the reuse and sharing of geographic information. Over the last few years a number of research activities and European projects have tried to address this issue, for example SPIRIT [143], OntoGeo[*], OGC[†], Inspire[‡], and Geo-Tumba[§]. In the context of geographic information retrieval, geo-ontologies are typically used for query expansion, term disambiguation and relevance ranking. Most existing geo-ontology designs are essentially Gazetteer models which include, explicitly in hierarchies, the thesaurus relationships; Related Term, Narrower Term and Broader Term, along with a spatial footprint. The additional expressive power of machine ontologies

[*] `http://ontogeo.ntua.gr/`
[†] `http://www.opengeospatial.org/`
[‡] `http://inspire.jrc.ec.europa.eu/`
[§] `http://www.tumba.pt/`

(thanks to their logical underpinnings) will push forward the reasoning potential of geo-ontologies.

2.3 Geospatial Integrity Maintenance

A major challenge to the realisation and the success of the GeoWeb is the reliability and consistency of the geo-information encoded in ontologies and in general of the geo-information that is being shared and used. Inaccuracy or error in geographic data can be accumulated at different stages of data handling and use [167, 134, 171], from the data collection phase, to maintenance and update processes on stored data. Errors in the description of the location and shape of geo-objects can propagate to errors in the spatial relationships between those objects, and consequently to wrong information being retrieved and analysed by users. In all, erroneous updates to geographic data sets may go undetected unless appropriate spatial integrity constraints are declared and applied, where 'Consistency describes the absence of any logical contradictions within a model of reality' [257].

Database systems have long held the notion of domain and referential integrity constraints, which help to maintain the consistency of stored knowledge. Constraints can act at three different states ([66, 68] in [35]), these are a static, transition and dynamic. A static state constraint is satisfied at every single state of the database e.g. an identifier must be have an integer value. A transition state constraint restricts the possible transitions from one database to another e.g. during update to the polygon of a Country, that polygon can not be made smaller than the sum of its containing regions. Finally a dynamic constraint restricts the possible state transitions of the database.

A Geographic Information System (GIS) should inherently contain the notion of consistency, however consistency constraints are often a neglected area of GIS, in particular when developing ontologies [79]. More recently, certain types or geographic constraints are being included into spatial database management systems such as Oracle 10g, and LaserScan's Radius Topology. Where these systems focus on geometric integrity of the spatial representation of geographic objects.

Geographic constraints can be expressed on thematic properties e.g. population or feature type, as well as on spatial properties e.g. the objects topology properties or size [168]. GIS constraints can be described by a set of consistency rules [277], which can be enforced by the transaction engine of the database management system [79].

Rule based constraint checking dates back to Event Condidtion Action (ECA) databases [176, 203]. Such ideas have been applied to a number of existing spatial databases, for example in [204, 260]. Rules for maintaining the consistency of the topological relations in a spatial database, including a process for topological error correcting can be seen in [260], and [36] deals with the use of business rules for spatial information systems.

Important to the definition of individual instances of spatial integrity rules, is the categorisations of integrity rules. Such a categorisation is shown below, which is taken and slightly adapted from [35, 235]:

- **Geometric Integrity Constraints:** Concerned with maintaining the accuracy of an objects geometry or location. For example, checking a polygon has 3 or more sides or looking for polygon overshoots and slivers.
- **Semantic Integrity Constraints:** Concerned with the meaning of geographical features and how they should legally be allowed to interact. For example a topo-semantic constraint defines the legal interaction between objects topological configuration i.e. a road cannot pass through a river, a house is not contained within a lake. Other spatial relations can also be considered, for example size-semantic where a city can not be smaller than the union of it's member neighbourhoods.
- **User Defined Constraints:** For expressing user-defined or business geospatial rules on geographic objects, which can be a mixture of either semantic or geometric constraints.

Existing semantic web technologies such as XML have well defined constraint models [157]. More recently, the ideas of integrity constraints have been brought to the area of ontologies, where ontologies themselves are seen as a necessary step toward maintaining geographic information [79], an issue that that will be taken forward in this thesis.

Maintaining consistency of spatial relations can also be tackled from a qualitative perspective using developed spatial calculi. A more in depth discussion of spatial calculi, or more generally qualitative spatial representation and reasoning, is given in the section to follow.

2.4 Qualitative Spatial Representation and Reasoning

From early on in GIS there was a need to find ways in which to represent continuous properties of the world by a discrete system of symbols - a qualitative approach[37]. This would replace or supplement the quantitative e.g. geometric, approach currently in use by most computer systems. Furthermore, in a GIS, there is a need to shield users from the complexity of the geometric representation associated with geographic objects, and instead present a higher level qualitative abstraction that is closer to human representation used for example in everyday communication and natural language. These requirements along with the study of spatial concepts from the cognitive point of view have provoked the birth of the qualitative spatial reasoning field of study [37].

Qualitative spatial representation is concerned with capturing everyday common sense spatial knowledge. This explicitly captured knowledge can be used within computer systems to, using appropriate reasoning techniques, make predictions or diagnose and explain the behaviour of physical systems [37]. Qualitative knowledge also has the advantage of being less dependent than quantitative knowledge which needs some reference or scale i.e. a mathematical coordinate system.

Symbolic qualitative spatial relationships are useful to abstract the user away from geometrical attributes. The human cognition works best using an object-centric interpretation of the world as opposed to a quantitative metric interpretation [83]. For example, it can be sufficient to refer to inclusion relationships between objects, such as *Cardiff* is **in** *Wales*, for a person to understand their relative spatial relationships. This is in contrast to presenting a person with precise coordinates of the objects, from which to compute the same information. Qualitative spatial reasoning offers an alternative symbolic way of dealing with the underlying data which is often out of scope to all but domain experts. As a consequence, symbolic reasoning methods do not require recourse to expensive geometric computational processing. Moreover, precise quantitative information can be too unrestricted and complex to be of help in a decision process compared to more restricted qualitative knowledge [82].

A complete categorization of spatial relationships was proposed by Egenhofer [55]. He defined the spatial relations; topological, metric (distance and direction) and relations concerning the partial and spatial ordering of objects e.g. direction. A crucial first step in any qualitative system is how to best define the representation of these qualitative spatial relations which typically involves mapping from the quantitative scale domain to qualitative symbols. As well as representation, reasoning with the defined relations is an important task which must also be addressed. The late 1980's and early 1990's produced much work in qualitative spatial representation and reasoning for all different types of spatial relationships, this work is reviewed and discussed in the sections to follow.

2.4.1 Qualitative Topological Relations

Topological space describes neighbourhood and incidence without the notion of distance measures and metrics. The lack of instantly definable distance measures allows topological spaces to be homomorphic in that they are invariant under scale, rotation and translation. Topology can be described at the qualitative level using topological complexes useful to support computation of geometry, or on the qualitative (symbolic) level closely mimicking human cognition of spatial relationships. Qualitative topological spatial relationships are often used in describing spatial configurations through natural language expressions, for example.

Cardiff is *inside* Wales
Cardiff *boarders* Newport

Until 1991 there was little work on the theory of spatial relations with the exception of Clarkes work on connection [33] and work on one-dimensional temporal/spatial relations based on Allens interval calculus [3]. Since then, topological spatial relations were the first type of relation to undergo extensive research.

Motivated by the need to find a complete coverage of topological spatial relations between spatially extended objects (regions as opposed to points) with strong syntactic and semantic foundations*, Egenhofer published work in 1991 on

*As Egenhofer notes, earlier works only considered weak, often verbal, definitions of relations the 4-intersection model rooted in topological space and theory [55]. This was a generalization of his earlier attempt to represent topological relations with Pullar in 1988 [211].

The 4-intersection model describes topological spatial relationships between polygonal areas in the plane (thus restricting topological space), and does so by exploiting the standard point set topological concepts of interior (where Y is a subset of the topological space X, denoted Y^o), boundary (denoted δY) and closure which is the union of both the interior and boundary (denoted $\overline{Y} = Y^o \cup \delta Y$). From the notions of boundary and interior, 4-intersections can be determined between the subsets A and B of the topological space X:

$$A^o \cap B^o(Interior\ Interior) \tag{2.1}$$

$$A^o \cap \delta B(Interior\ Boundary) \tag{2.2}$$

$$\delta A \cap B^o(Boundary\ Interior) \tag{2.3}$$

$$\delta A \cap \delta B(Boundary\ Boundary) \tag{2.4}$$

The result of an intersection is either an empty or non-empty set (non-empty sets indicate that the intersection holds). From the above 4-intersections a total of 16 possible relations can be defined between any two subsets of the topological space X, described by an intersection matrix. The 16 relations are mutually exclusive for any two sets A and B, in that only one of the relations holds. This is a very important result not previously obtained, which states that these 16 relations are complete with respect to the coverage of all topological spatial relations. The 16 relations however can be further reduced into different subsets depending on the restrictions placed on the topological space. For example as Egenhofer notes, the relations in one dimension are different from those in two dimensions and so on.

By further restricting the domain to only include topologically connected spaces (a boundary must be definable) and one piece polygonal areas in the plane that are closed sets (including their boundary $A = \overline{A^o}$), only 9 of the 16 topological spatial relations exist. Egenhofer gives qualitative descriptions (terminology) of the values produced by the 9 binary spatial relations where the semantics of these relations are defined by the interior boundary intersection matrix. Only 8 relations are definable when considering two dimensional euclidean spaces ($\mathbb{R}^2$).

The 4-intersection model was restricted to objects with the same dimension as the space they occupied (co-dimension of 0), in particular between two regions, or polygons in a GIS, in $\mathbb{R}^2$. Later Egenhofer extended the 4-intersection model to exploit the topological notions of interior (Y^o), boundary (δY) and, exterior (Y^-). This lead to a possible 9-intersections [57]. The 9-intersection model is more fine grained than the 4-intersection model, and can detect relations between objects which do not share the same dimensions* (in this sense it is dimensionally extended) as the topological space they exist in, for example between lines and regions. In total, the 9-intersection model can detect a complete coverage of topological relations between; line-line, region-region, point-point, line-region, point-region and point-line (see [34] for a study on these intersections).

The dimensionally extended 9-intersection model is widely used within GIS's for the definition of both semantics and terminology of topological spatial relations. For example the OGC filter specifications [267] and the popular commercial spatial databases Oracle* and PostGIS† use the syntax and semantics of the 9-intersection model. Moreover, the intersection model is suitable for modelling relationships between spatial extended objects in 3-dimensional space [280], and has found application in image databases where its point set topological theory has been interpreted into pixels sets for application over images [15]. More recently, based on the idea of intersection, methods have been proposed for the automatic derivation of composition tables for different types of spatial objects, and objects with arbitrary complexity [65, 64].

Developed at a similar time, but with a stronger adoption by the Artificial Intelligence (AI) community was the work of Randell et al. on the Region Connection Calculus [214]. Region Connection Calculus (RCC) is an extension of Clarkes original theory of connection in 1981 [33] which has a topological point set theory. They note the original theory of Clarke to have problems, conceptually, pragmatically and computational, which they aim to rectify with a new theory based on connection. In overview they aim to make an more intuitive theory that only considers closed sets (as opposed to open, closed and semi open in Clarkes theory) -

*of note the 4-intersections can distinguish between line-line relations and region-region relations as they are in the same dimension

*`www.oracle.com`

†`postgis.refractions.net/`

which are used in a GIS as physical objects represented by polygons corresponding to closed regions.

In a departure from the n-intersection model, RCC theory is dimensionless and, even though it can take a mathematical topological interpretation (often used in real world situations), the theory is based around a connection predicate C which can assume a number of different interpretations. C(x,y) is a primitive, dyadic (binary), symmetric ($\forall xy\ C(x,y) \rightarrow C(y,x)$) and reflexive ($\forall x C(x,x)$) relation meaning x and y are connected, where x and y are either proper (have an inside) or improper regions which could consist of one or more regions. Its exact interpretation is variable depending on its domain of use, for example using a topological interpretation $C(x,y)$ means that x shares a common point with y. Other possible interpretations exit, for example x and y have a Euclidean distance measure of zero [247].

From the primitive relation $C(x,y)$ 14 relations can be built, namely the set:

$$RCC = \{DC, P, PP, EQ, O, PO, DR, TPP, EC, NTPP, P^{-1}, PP^{-1}, TPP^{-1}, NTPP^{-1}\}$$

Where each relation is defined in terms of the connection predicate C, for example:

$$DC(x,y) \equiv_{def} \neg C(x,y) \tag{2.5}$$

$$P(x,y) \equiv_{def} \forall z\ [C(z,x) \rightarrow C(z,y)] \tag{2.6}$$

$$PP(x,y) \equiv_{def} P(x,y) \wedge \neg P(y,x) \tag{2.7}$$

These are,:

- Two regions x and y are disconnected (DC) if they are not connected (2.5)
- A region x is part of (P) a region y if for every region z connected to x, z is also connected to y (2.6)
- The region x is a proper part (PP) of region y if x is a part of y, and y is not a part of x (2.7)

All 14 relations are definable in this way form a lattice with weak (most general) relations at the top and strong (most specific) at the bottom (see Figure 2.5 in section 2.4.3.2). At the bottom level 8 relations which are not subsumed further are defined which are, as with the 4/9-intersection model, a complete coverage of topological spatial relations (Jointly Exhaustive and Pairwise Disjoint JEPD). These form the set RCC-8 which is the most commonly used fragment of RCC (illustrated for simple regions in $\mathbb{R}^2$ in Figure 2.2).

$$\text{RCC-8} = DC, EC, EQ, PO, TPP, NTTP, TPP^{-1}, NTPP^{-1} \tag{2.8}$$

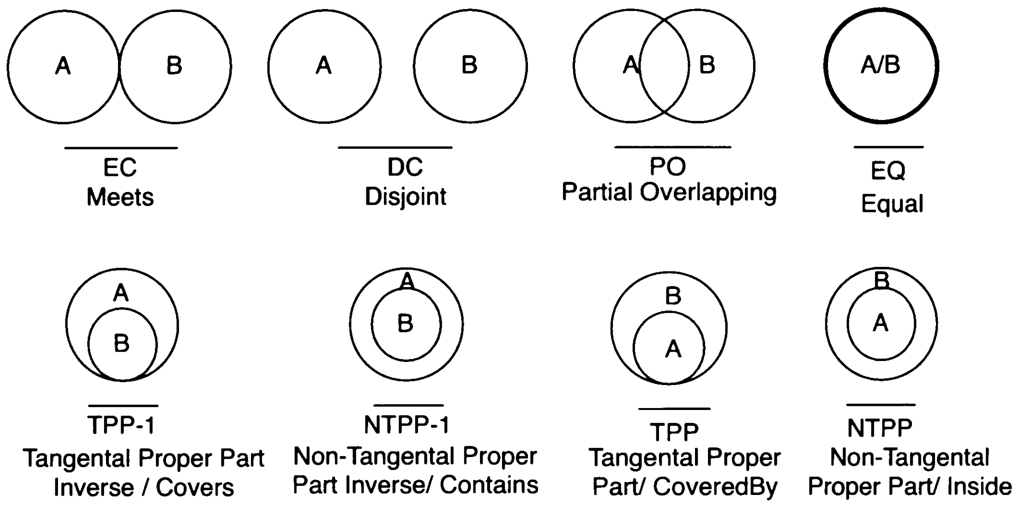


Figure 2.2: Illustration of the 8 RCC Base Relations - with alternative terminology

All spatial regions are mesaruable sets in $\mathbb{R}^2$ (although higher dimensionality is possible [218]). However, importantly RCC-8 formulas hold for the discrete domain Z^2 as well as R^2, i.e. the digital plane in addition to a continuous space [165]. This proof is of particular importance for the purpose of geographic information retrieval, as a GIS deals with digitised information sources that are discrete representations of real world phenomena.

In practice RCC and the 4/9-intersection model have an identical set of 8 JEPD relations between two regions, albeit using different relationship terminology. Of the two, the 9-intersection has a more intuitive and commonly used terminology. Of note, RCC has not been used to formalise topological relations between lines and regions, as opposed to the 9-intersection model.

2.4.2 Other Qualitative Spatial Relations

In addition to topological relations, three other types of relations, not used in the reminder of this thesis, have been defined, these are; relative size, proximity or distance, and direction.

Relative Size: Relative size information between two objects is typically represented as a binary metric value relating the size of the primary object compared to the size of the reference object. For example, *The size of Wales is at minimum 10 times the size of Cardiff* **and** *The size of the Wales minus the size of all its islands is equal to the size of mainland Wales.* Qualitative relative size information, which maps linguistic variables to size measures, is more commonly used within natural language descriptions of spatial configurations, compared to it's metric counterpart [90]. A study of relative size relations, and how they can be combined effectively with topological relations can be seen in [90].

Proximity and Distance: Proximity relations are often used in linguistic expressions to define spatial configurations, or in asking questions about spatial objects. For example, *The petrol station is near your current location*, or *find me a quiet location furthest away from places with large populations.* Proximity is subjective, hard to define and various with context [278]. Qualitative proximity relationships are often symbolised as surrogates for distance measures [83]. Two well known approaches to defining proximity exist. Relative distance that ranks objects by their distance to the primary object (an 'ordinal' approach), and absolute distance which assumes a simple linear relationship between distance (for example euclidean distance) and proximity [83].

Direction: Direction deals with order in space, and is used in natural language descriptions and cognitive reasoning procedures. Direction is a metric relation which varies under a number of different geometric transformations, and is an important part of the specification of spatial configurations [97]. Directional or orientation relations are important in wayfinding, for example *Newport is to the East of Cardiff* or *from your current position and orientation, keep going east until you reach the petrol station.* Early works on qualitative direction relations include those of Frank and Freksa, both in 1992. Each took a slightly different stance on which system better mimics human understanding and reasoning. Frank defines a cardinal direction system based on a cone-shaped area of acceptance or a projection based directional system [80]. Freska proposed an orientation model which he suggests matches more closely the human cognition as opposed to Franks cardinal direction [82]. Both of these methods are 0-dimensional (point based). Consequently they have been extended in newer works to n-dimensional objects, for example using minimum bounding rectangles (MBR) [201], directional matrices (which overcome representational problems with the MBR approaches) [96], and methods that can deal with complex object shapes [242].

2.4.3 Qualitative Spatial Reasoning

Qualitative reasoning takes advantage of the transitive nature of the partial or total ordering of the quantity space, in order to infer new qualitative information from the raw qualitative information presented. Qualitative spatial representation and reasoning (QSRR) works on the premise that descriptions of spatial configurations are often not based on quantitative information but qualitative / symbolic descriptions [221], mostly taken from natural language scene descriptions. Hence, such qualitative spatial descriptions are suitable for qualitative spatial reasoning mechanisms. The most prevalent form of QSRR, indeed the current paradigm for qualitative reasoning, was based on Allen's work on interval calculi in 1981 [3]. He devised a composition table, then known as a transitivity table*, from the analysis of temporal relations that shows how relations can be inferred. Work on QSRR was then delayed thanks to the now much refuted poverty conjecture promulgated by Forbus [74] which, in short, ruled out the possibility of reasoning with anything other than numbers - as a total order can not be defined on anything other than one dimensional space. One of the first QSRR techniques to re-emerge was the work of Clarke [33], which led to the development of the Region Connection Calculus (RCC) as described previously.

*Although Cohn notes in his survey paper of 2001 [37] that transitivity table is a poor term, as it does not exploit the transitivity of one relation but the possible composition of any two JEPD relations

Composition Tables: As important as the definition of the syntax and semantics of each spatial relation (as described in the previous sections), is the definition of the semantics of spatial relationship composition. Spatial relationship composition is a key step toward computing compositional inferences [37]. For example, topological compositional inferences in RCC can be computed from the formal definition of composition (see for example [221]):

$$R_1 \circ R_2 = \{< x,y > | \exists z :< x,z > \in R_1, < z,y > \in R_2\} \tag{2.9}$$

Where x, y and z are spatial variables from the domain of all spatial regions U, and $R_n \in$ RCC-8* or any defined subset S of RCC-8 ($S \subseteq$ RCC-8). Figure 2.3 depicts a spatial composition between three spatial regions A,B and C with explicit relations Rel(A,B), Rel(B,C) and the newly inferred relation Rel(A,C), representing R_1, R_2 and $R_1 \circ R_2$ from expression 2.9.

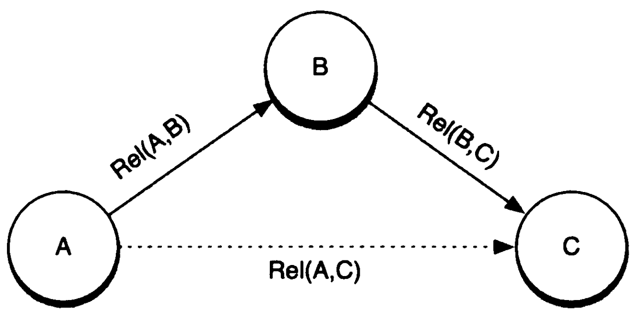


Figure 2.3: Example of a spatial composition, where Rel(A,C) is derived from the composition of the relations Rel(A,B) and Rel(B,C)

Once computed (which is often difficult [13, 12, 213]), compositional inferences are stored in a lookup table known as a composition table. A composition table stores inferences from two relational facts of the form R_1(A,B) and R_2(B,C) to a relational fact of the form R_3(A,C). Most developed spatial calculi have an associated composition table. The composition table for the base eight RCC relations is shown in table 2.1. Each entry in table 2.1 corresponds to the relation that results from the composition (as defined by operator 2.9) of the base relations shown in the related row and column. Composition of two abitrary RCC-8 relations (which

*Where RCC-8, as previously stated, is the set of topological relations formed from the eight base relations or any disjunction thereof

Table 2.1: RCC-8 Composition Table, where * represents the universal relation (the set of all 8 RCC base relations)

—	DC	EC	PO	TPP	NTPP	TPP^{-1}	$NTPP^{-1}$	EQ
DC	*	DC EC PO TPP NTPP	DC EC PO TPP NTPP	DC EC PO TPP NTPP	DC EC PO TPP NTPP	DC	DC	DC
EC	DC EC PO TPP^{-1} $NTPP^{-1}$	DC EC PO TPP TPP^{-1} EQ	DC EC PO TPP NTPP	EC PO TPP NTPP	PO TPP NTPP	DC EC	DC	EC
PO	DC EC PO TPP^{-1} $NTPP^{-1}$	DC EC PO TPP^{-1} $NTPP^{-1}$	*	PO TPP NTPP	PO TPP NTPP	DC EC PO TPP^{-1} $NTPP^{-1}$	DC EC PO TPP^{-1} $NTPP^{-1}$	PO
TPP	DC	DC EC	DC EC PO TPP NTPP	TPP NTPP	NTPP	DC EC PO TPP TPP^{-1} EQ	DC EC PO TPP^{-1} $NTPP^{-1}$	TPP
NTPP	DC	DC	DC EC PO TPP NTPP	NTPP	NTPP	DC EC PO TPP NTPP	*	NTPP
TPP^{-1}	DC EC PO TPP^{-1} $NTPP^{-1}$	EC PO TPP^{-1} $NTPP^{-1}$	PO TPP^{-1} $NTPP^{-1}$	PO EQ TPP TPP^{-1}	PO TPP NTPP	TPP^{-1} $NTPP^{-1}$	$NTPP^{-1}$	TPP^{-1}
$NTPP^{-1}$	DC EC PO TPP^{-1} $NTPP^{-1}$	PO TPP^{-1} $NTPP^{-1}$	PO TPP^{-1} $NTPP^{-1}$	PO TPP^{-1} $NTPP^{-1}$	PO TPP^{-1} TPP NTPP $NTPP^{-1}$ EQ	$NTPP^{-1}$	$NTPP^{-1}$	$NTPP^{-1}$
EQ	DC	EC	PO	TPP	NTPP	TPP^{-1}	$NTPP^{-1}$	EQ

includes disjunctive sets of relations) can be obtained by computing the union of the composition of the base relations [221]. As an example, the composition of the relation DC between regions A and B, with the relation $NTPP$ between the regions B and C, gives the disjunction of relations $= \{DC, EC, PO, TPP, NTPP\}$ between the regions A and C - this is shown by the entry in row 2 column 6 in table 2.1.

Within the context of this work, we focus on topological relations. In particular those defined by the RCC as opposed to Egenhofers n-intersection model. This is simply because the RCC has received a lot more attention in the AI community and a number of useful results exist based on the more general theory of connection, whereas the n-intersection model has been mostly used in a spatial database context. Moreover we do not aim to represent other spatial relations and or their combinations.

Spatial consistency using compositional inferences: Knowledge of the relation(s) that should hold between two regions, by compositional inference, enables a system to check the consistency of existing relations and thus decide the consistency of a spatial scene. We now discuss this by first defining standard notation used throughout the literature and the remainder of this thesis to describe the process of spatial reasoning and consistency checking over a set of spatial relations.

Spatial configurations - a qualitative description of the arrangement of a certain domain of spatial regions - can be described as a set Θ of spatial formulae or relational constraints. A spatial formula / spatial constraint is written $x\mathrm{R}y$, where R is an RCC-8 relation (of which, using RCC-8, there are 2^8 possible disjunctive sets of relations that form R, which includes each definite base relation, the universal relation and all possible disjunctions of the eight base relations) and x, y are variables in the domain of spatial regions U - in later chapters this is the domain of geographical regions in our developed geo-ontology.

Deciding consistency of a set Θ of relational constraints is denoted RSAT or Region Satisfiability (an alternative to the typical boolean satisfiability denoted SAT) where a consistent instantiation of regions to the variables x and y is sought which does not violate the constraints in Θ. RSAT is, in general, NP-Hard* and hence is intractable [221]. RSAT can be formalised as a Constraint Satisfaction Problem (CSP) and as such RSAT can be approximated using a path-consistency technique which can be used as a polynomial time heuristic test for whether a spatial network is consistent [255] (as first used for interval algebra by Allen [3]).

Constraint satisfaction is a process whereby a correct assignment of variables can be found for a given scene, such that the variables do not invalidate a specified set of constraints (in our case the topological relational constraints in Θ). Less formally, Constraint Satisfaction in a spatial context is concerned with finding the correct assignment of spatial (topological, orientation, size or proximity) relations between geographic features within a geographic scene, such that those relationships are not inconsistent with those found in the closed set of constraints Θ. Where Θ is closed using the algebraic operations, composition ($\otimes$), intersection ($\cap$) and converse ($\smile$) - closure of Θ occurs during the path-consistency process. Where the composition operator has already been defined in expression 2.9, and the converse and intersection operators for RCC-8 are defined as follows (where R and S are RCC-8 relations, and X and Y are spatial variables):

$$\forall X, Y: \quad XR^{\smile}Y \quad \text{iff} \quad YRX \tag{2.10}$$

*That is in the worst case, it is as hard as the hardest problem that can be determined in non-deterministic polynomial time. Investigation of SAT problems are credited to Stephen Cook in 1971

$$\forall X, Y: \quad X(R \cap S)Y \quad \text{iff} \quad XRY \wedge XSY \tag{2.11}$$

Path-consistency algorithms for topological relations then operate over constraint graphs where spatial variables in the domain U are nodes, and relations in RCC-8 are edges. Hence, a relational constraint $x\text{R}y$ is formed by two nodes and one connecting edge. An example constraint graph, where the nodes represent the spatial regions {Cardiff, Newport, Roath, Cathays} and the set of relational constraints Θ= { {NTPP^{-1}} (Cardiff, Roath), {EC,DC,PO} (Cardiff, Newport), {DC} (Cathays,Newport), {EC,DC} (Roath, Cathays), {DC} (Roath, Newport),{NTPP^{-1}} (Cardiff, Cathays)}, is shown in Figure 2.4 .

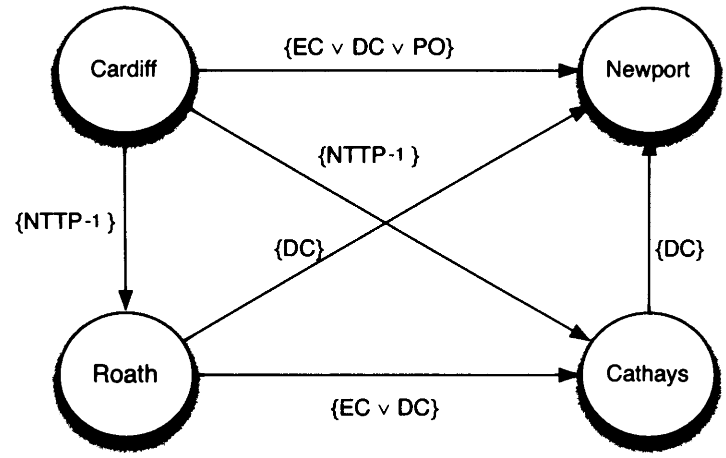


Figure 2.4: Example constraint graph (also referred to as a spatial network) for the set of relational constraintsΘ

Singh and Kumar [240, 156] surveyed different algorithms for solving the path consistency problem, namely PC-1 through to PC-5. The basic path consistency algorithms (PC-1 and PC-2) are described in [183, 169] and have been further developed and improved in [16]. Algorithms 2.1 and 2.2 show both Vilain and Kautz's path-consistency [265] and Mackworth's Revise [169] algorithms commonly used or extended for spatial reasoning tasks e.g. as shown in more detail in [90]. The algorithm can be solved in polynomial $O(n^3)$ time and has a space complexity of $O(n^2)$ [170]. For the spatial case, Revise is used to decide the consistency of spatial relations between regions (where A,B,C are spatial variables that correspond to spatial regions in the universe of discourse U) by performing operation 2.12, which successively removes (or refines) relations from all edges in the constraint graph until either an edge is left empty (no relations hold), indicating that the constraints can not be satisfied, or until a fixed point is reached when no more relations can be removed, indicating a consistent network.

$$A_rC = A_rC \cap (A_rB \otimes B_rC) \tag{2.12}$$

Where r is substituted for the relation holding between two regions, and $A_rB \otimes B_rC$ becomes the relation formed from the composition of the relations between A and B with B and C (as found in the precomputed composition table). Once a path-consistent set of constraints are found, the algorithm SCENARIO [219] can be used to find a particular consistent instantiation of those constraints. Interestingly, on deciding the first instance of inconsistency a path-consistency algorithm will exit without identifying the exact source of the inconsistency, or suggesting to the user how to rectify the inconsistency. This is observable in [16], where each path-consistency algorithm surveyed exits on the first sign of inconsistency. Localising errors and inconsistencies in spatial scenes is an important issue for geo-ontologies and is addressed in this thesis.

Algorithm 2.1 Path-consistency

procedure Consistency(i,k,j)
Q ← $\{(i,j) | i < j\}$
while Q $\neq \theta$ do
select and delete an arc (i,j) from Q
for $(k \neq i, k \neq j (k \in \{1...n\})$ **do**
  if (REVISE(i,j,k) **then**
    if $(R_{ik} = \theta)$ **then**
      return fail
    end if
    else add(i,k) to Q
  end if
  if (REVISE(k,i,j) **then**
    if $(R_{kj} = \theta)$ **then**
      return fail
    end if
    else add(k,j) to Q
  end if
end for

Algorithm 2.2 Revise

procedure REVISE(i,k,j)
oldt := t_{ij}
$t_{ij} := t_{ij} \cap (t_{ik} \otimes t_{kj})$
if (oldt :=t_{ij}) **then**
  return false // no revisions
end if
t_{ji} := Converse(t_{ij}) //by lookup to the converse relation
return true

2.4.3.1 From Path-consistency to Global Consistency

Path-consistency is, on its own, only an approximation of global consistency. More specifically, deciding path-consistency over one set of possible relations is not always sufficient to determine whether a scene is actually consistent, so called global consistency [221]. Hence either a process of branching and backtracking over disjunctive (sets of more than one) relations is required, or more tractable subsets of constraints need to be identified where, importantly, path-consistency is sufficient to decide global consistency and hence global consistency can be determined in polynomial time using a path-consistency method. Renz and Nebel [221] defined a maximal tractable subset of RCC-8 by identifying a boundary subset of relations between those NP-Hard subsets and polynomial subsets. Such subset was named $\mathcal{H}_8$ which contains 148 of the possible 256 RCC-8 relations. Tractability for $\mathcal{H}_8$ was shown by transformation of the set of relations in $\mathcal{H}_8$ to propositional Horn clauses, which is itself known to be solvable in polynomial time - such analysis also proved that for these cases path-consistency was sufficient for global consistency. This was later followed by Renz's complete analysis of tractable subsets of RCC-8 [219] where he identified two slightly larger tractable subsets of RCC-8 namely $\mathcal{C}_8$ and $\mathcal{Q}_8$, containing 158 and 160 relations respectively. Again as with the subset $\mathcal{H}_8$, Renz proved that for each of these two additional fragments ($\mathcal{C}_8$ and $\mathcal{Q}_8$) path-consistency is sufficient for deciding global consistency. Less formally, providing the initial set of topological relational constraints exist in one of the three tractable subsets (which includes even disjunctive relations), deciding path-consistency, without the need to backtrack over disjunctive relations, is sufficient to determine global consistency. These three sets are complete and thus no others are known to exist.

In addition to deciding RSAT in tractable time over a initial set of relational constraints in one of the maximal tractable subsets of RCC-8, knowledge of each tractable subset can also be used to speed up the use of backtracking [158] for relational constraints (or spatial configurations) that include relations outside of the maximal tractable subsets (those that are NP-Hard). That is, backtracking is only required until all relations are in one of the maximal tractable subsets, and then determining path-consistency of this set is again sufficient for determining global consistency.

2.4.3.2 Generalising Topological Relations

Generalising RCC-8: Generalising RCC-8 into other sets of JEPD relations is possible, for example the RCC-5 [12]. RCC-5 is boundary insensitive, and therefore only five JEPD relations can be determined, namely: DR - discrete from, PO - partially overlapping, PP - proper part, PPI - proper part inverse, and EQ - equal. These generalised relations encapsulate a number of lower level RCC relations, as illustrated in Figure 2.5 which shows how each RCC relation is defined from the root connection predicate C (so termed the RCC relational lattice). The most general relations are at the top of the lattice, and the most specific are toward the bottom.

For example from Figure 2.5, PP is formed from the RCC-8 set of base relations $\{NTPP,TPP\}$. The RCC-5 composition table is smaller and therefore more tractable to reason with than the RCC-8 composition table [220]. That said satisfiability checking in RCC-5 is still NP-Complete [221] as certain compositions between RCC-5 relations again lead sets of indefinite, disjunctive RCC-5 base relations. In its favour, most natural language descriptions of spatial configurations do not make distinctions for example between a region being a *part of* another region, or that region being a *non-tangental proper part of* (completely inside) another region. Hence RCC-5 is a more natural way to express such relations. Like RCC-8, tractable subset of RCC-5 have been studied, for example in [144].

RCC reasoning over definite compositions: In [231] Schockaert recognised the need to represent and reason with vague qualitative spatial information, as is prevalent for example in web documents. To do this, Schockaert developed a

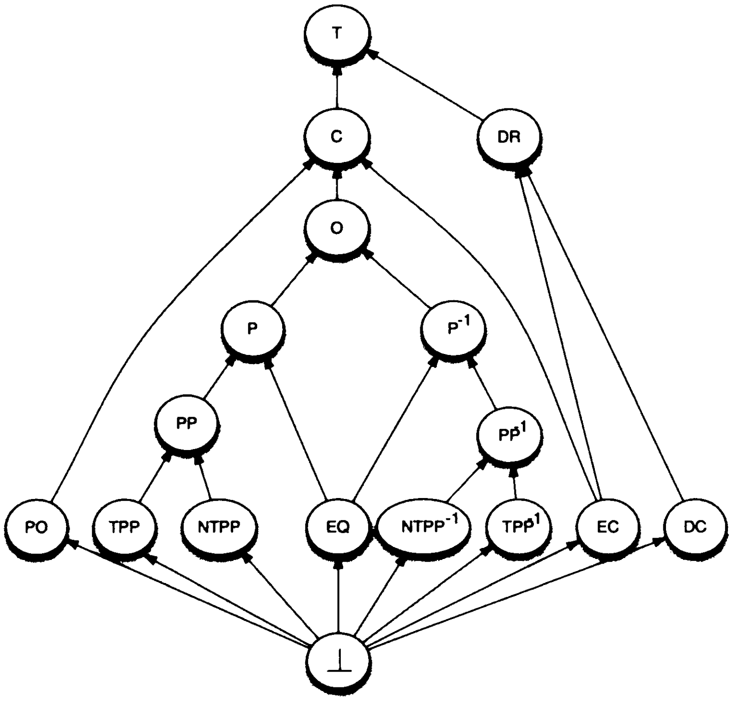


Figure 2.5: RCC-8 subsumption lattice of dyadic relations defined in terms of the connection predicate C. Where $\top$ represents a tautology and $\bot$ represents a contradiction

qualitative spatial reasoning framework based around a fuzzyfication of RCC-8.

During the development of the fuzzy RCC-8, Schockaert showed an equivalent composition table for RCC-8, where, importantly, the unions of RCC-8 base relations (which resulted from the composition of two arbitrary RCC-8 base relations) are replaced by the conjunction of generalised RCC relations (as first shown in [232]). As with RCC-5 and above, generalised RCC relations are definable thanks to the JEPD property of RCC-8 base relations. For example the complement of P^{-1} written as coP^{-1} or logically $\neg P^{-1}$ can be defined as:

$$DC \cup EC \cup PO \cup TPP \cup NTPP = coP^{-1} \tag{2.13}$$

Hence coP^{-1} can be used in place of the RCC-8 relation $\{DC, EC, PO, TPP, NTPP\}$. The RCC lattice in Figure 2.5 is again a visual illustration of the RCC-8 relations that form each generalised relation.

In order to fully replace all unions of base RCC-8 relations in the composition table with conjunctions of generalised relations, the definition of 12 RCC generalised relations was required, namely; C,P,P^{-1},O,NTP,NTP^{-1}, DC, $\neg P$, $\neg P^{-1}$, DR, $\neg NTP$, and $\neg NTP^{-1}$. Table 2.2 defines each of these generalised relations in terms of their equivalent RCC-8 relations. As an example, the disjunction

Relation	Name	RCC-8 Disjunctive Relations
C(a,b)	Connected	a$\{PO, TPP, NTPP, EQ, NTPP^{-1}, TPP^{-1}, EC\}$b
DC(a,b)	Disconnected	a{DC}b
P(a,b)	Part-of	a{ TPP,NTPP,EQ}b
P^{-1}(a,b)	(Part-of)$^{-1}$	a$\{TPP^{-1}, NTPP^{-1}, EQ\}$b
coP(a,b) = ¬ P(a,b)	¬ Part-of	a$\{PO, NTPP^{-1}, TPP^{-1}, EC, DC\}$b
coP^{-1}(a,b)= $\neg P^{-1}$(a,b)	¬ (Part-of)$^{-1}$	a$\{PO, NTPP, TPP, EC, DC\}$b
O(a,b)	Overlapping	a$\{PO, TPP, NTPP, EQ, NTPP^{-1}, TPP^{-1}\}$b
DR(a,b)	Discrete From	a$\{EC, DC\}$b
NTP(a,b)	Non-tangential Part-of	a$\{NTPP\}$b
NTP^{-1}(a,b)	(Non-tangential Part-of)$^{-1}$	a$\{NTPP^{-1}\}$b
coNTP(a,b)	¬ Non-tangential Part-of	a$\{PO, TPP, EQ, NTPP^{-1}, TPP^{-1}, EC, DC\}$b
$coNTP^{-1}$(a,b)	¬ (Non-tangential Part-of)$^{-1}$	a$\{PO, TPP, EQ, NTPP, TPP^{-1}, EC, DC\}$b

Table 2.2: Generalised relations and their corresponding set of RCC-8 relations, where a and b are regions $\in U$

($\{EC, PO, TPP, NTPP\}$ of RCC-8 base relations in row 3 column 5 of Table 2.1 can be replaced by the conjunction of the generalised relations C and coP^{-1} so:

$$\begin{aligned} C \cap coP^{-1} \quad &\equiv \\ \{PO, TPP, NTPP, EQ, NTPP^{-1}, TPP^{-1}, EC\} \quad &\cap \quad \{PO, NTPP, TPP, EC, DC\} \\ = \{EC, PO, TPP, NTPP\} \end{aligned}$$

As a further step, Schockaert developed a full composition table (Table 2.3) for the 12 generalised relations (henceforth denoted RCC-12) allowing reasoning over sets of relational constraints where the relations R are from RCC-12. Importantly, all possible relations R in RCC-12 includes each of the base relations (P, C, coP^{-1} etc), the universal relation, and any conjunction of base relations. Then for RCC-12 relations, a set representation of RCC-12 relations represents a **conjunction** of base relations, as opposed to a **disjunction** in RCC-8. . Also of interest to this work, each composition results in a definite RCC-12 relation and hence compositional inferences are definite Horn inferences (Horn inferences and other knowledge representation paradigms are detailed in Chapter 3), as opposed to disjunctive inferences for RCC-8, of the form:

$$R_1(x, y) \wedge R_2(y, c) \rightarrow Rh(x, c) \tag{2.14}$$

where x, y and c are spatial variables, R_1, R_2 and Rh are substituted for an

RCC-12 relation. Equivalence of reasoning with the generalised RCC composition table and the classical RCC-8 composition table was also proven, hence full compositional reasoning is possible using RCC-12. . Less formally, computing the closure of a set of relational constraints using classical RCC-8 reasoning generates the same compositional inferences (the same refined set of relational constraints, using the RCC-8 base relations) as the closure of the same set of relational constraints using the generalised composition table. Moreover, all 12 base generalised relations are in the set $\mathcal{H}_8$, and as the set $\mathcal{H}_8$ is closed under intersection, the intersection of any base RCC-12 relation is also in the set $\mathcal{H}_8$. Hence, providing a mapping exists between a set of RCC-8 relations and a corresponding conjunctive set of generalised base relations, then deciding path consistency over the resultant generalised relational constraints, is sufficient for deciding global consistency of the set of relational constraints.

2.5 SUMMARY

In this chapter the field of information retrieval, and in particular geographic information retrieval, in the context of computer science was shown. Work on machine ontologies and recent advancements in domain specialised geographic ontologies was overviewed. The need to maintain consistency of geographic information was highlighted, where ontologies themselves are seen as an opportunity to represent constraints over geographic information. Work in representing and reasoning about relations between objects in space was discussed, with a particular emphasis on topological relations. Reasoning techniques to maintain consistency of topological relations was explored in detail, with a view that any geographic ontology paradigm should include such techniques.

In the next chapter we look at current web knowledge representation paradigms, in addition to their formal logical underpinnings, with a view to representing geographic ontologies and qualitative spatial relations and reasoning.

	C	DC	P	P^{-1}	coP	coP^{-1}	O	DR	NTP	NTP^{-1}	coNTP	$coNTP^{-1}$
C	**1**	**coP**	**C**	**1**	**1**	**1**	**1**	**coNTP**	**O**	**1**	**1**	**1**
DC	coP^{-1}	1	coP^{-1}	DC	1	1	coP^{-1}	1	coP^{-1}	DC	1	1
P	**1**	**DC**	**P**	**1**	**1**	$\mathbf{coP^{-1}}$	**1**	**DR**	**NTP**	**1**	**1**	$\mathbf{coNTP^{-1}}$
P^{-1}	C	coP	O	P^{-1}	coP	1	O	coP	O	NTP^{-1}	coNTP	1
coP	**1**	**1**	**1**	**coP**	**1**	**1**	**1**	**1**	**1**	**coP**	**1**	**1**
coP^{-1}	1	1	coP^{-1}	1	1	1	1	1	coP^{-1}	1	1	1
O	**1**	**coP**	**O**	**1**	**1**	**1**	**1**	**coP**	**O**	**1**	**1**	**1**
DR	$coNTP^{-1}$	1	coP^{-1}	DR	1	1	coP^{-1}	1	coP^{-1}	DC	1	1
NTP	**1**	**DC**	**NTP**	**1**	**1**	$\mathbf{coP^{-1}}$	**1**	**DC**	**NTP**	**1**	**1**	$\mathbf{coP^{-1}}$
NTP^{-1}	O	coP	O	NTP^{-1}	coP	1	O	coP	O	NTP^{-1}	coP	1
coNTP	**1**	**1**	**1**	**coNTP**	**1**	**1**	**1**	**1**	**1**	**coP**	**1**	**1**
$coNTP^{-1}$	1	1	$coNTP^{-1}$	1	1	1	1	1	coP^{-1}	1	1	1

Table 2.3: Composition table for generalised RCC relations

Chapter 3

OWL and Logical Knowledge Representation

In the previous Chapter we discussed the general notion of ontologies along with ontologies specific to the geographic domain, geo-ontologies. With the advent of the Semantic Web, new web based ontology languages and technologies have been developed. In this Chapter we explore such technologies, their heritage and formal logical underpinnings.

3.1 The Web Ontology Language, OWL

The Semantic Web is an initiative by the World Wide Web Consortium (W3c) based on the vision of Sir Tim Berners-Lee [14]. It aims to provide meaning to the comprehensive amount of information on the Internet. At the heart of the Semantic Web are ontologies. Committing knowledge of concepts and their relationships on the web to an ontology brings a shared understanding of those concepts. Furthermore, ontologies enable greater potential for reasoning with those concepts to uncover implicit information. As a result, current syntactic searches on the web will be replaced with intelligent searches where the meaning of search constraints is understood by the machine, and can be related more accurately to search results.

To facilitate this level of machine knowledge and understanding, the W3c developed numerous Semantic Web technologies as illustrated by the Semantic Web layer cake in Figure 3.1. Toward the bottom and middle of the layer cake are ontological knowledge representation, sharing, and reasoning languages, namely; the resource description framework (RDF), the resource description framework schema (RDFS) and the web ontology language (OWL). Each of which provides increasingly expressive modelling and reasoning potential.

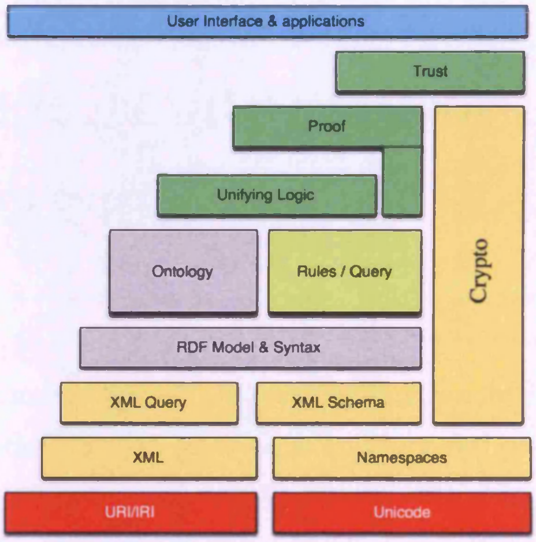


Figure 3.1: W3C's Semantic Web layer cake

RDF is a machine understandable base representation language for asserting knowledge about web resources. A web resource can be anything from online resources, to offline concepts and notions. A resource can be anywhere on the web and is identified by a Uniform Resource Identifier (URI) - one of the most powerful concepts in use on the web today (relating and linking web knowledge together [14]).

RDF provides a simple knowledge representation model using binary predicates, for example to express a containment relation between Cardiff and Wales, **Inside(Cardiff, Wales)**. A binary predicate is represented as a triple in RDF which has the syntactic form: $triple < subject, predicate, object >$. The triple asserts or affirms knowledge, described by the *predicate* about the *subject* and *object* - as is typical in English grammar. RDF models form graphs between *subjects* and *objects*, linked by *predicates*. A triple either relates a subject to another resource (object) or a literal value. An example RDF graph about Mountains and Countries is shown in Figure 3.2, where literal values are shown in ovals and resources are shown in rectangles.

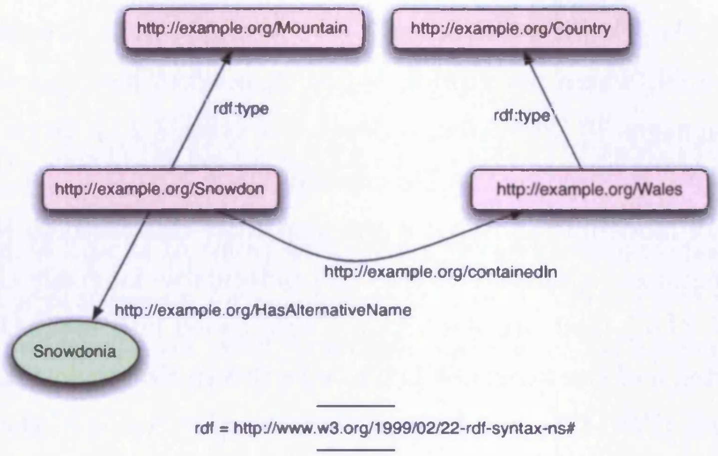


Figure 3.2: RDF Graph mountain and country example. Ovals represent literal values and rectangles represent resources.

The semantics of RDF revolve around assigning an appropriate interpretation of its graph model. The graph model, although possibly subject to different interpretation [161], has a sound model theory which allows an RDF graph to be translated to a logical expression with identical meaning [117].

RDF is designed specifically for knowledge sharing and reuse. RDF, as indeed with all Semantic Web technologies, has an XML surface syntax to enable interoperability over the web. All languages layered on top of RDF have an RDF triple representation and can be queried using SQL like syntax (but for triples patterns) using either RDQL [234] or SPARQL [207] - even if the meaning of their higher level constructs is not understood.

RDF Schema (RDFS) [22] is an extension to RDF that provides base ontological constructs for defining custom vocabularies. RDFS allows user defined classes and properties, giving it the same characteristics as a simple object orientated language. RDFS provides the backbone of the Semantic Web in use today [161] and, although some argue against RDFS as the base ontology language for the Semantic Web due to unsound model theory [128], it forms the base from which the richer ontology language OWL is built.

The Web Ontology Language OWL is based on a revised version of its predecessor ontology language DAML+OIL, incorporating new ideas gained from its application. DAML+OIL itself stems from the merger of two languages, DAML produced by DARPA in America, and the European venture OIL [69], led primarly by Ian Horrocks. DAML+OIL is built on top of 15 years of research into description logics [188], where description logics themselves have already been used as ontology languages [9] (discussed in depth in section 3.2.1). In particular, the first version of OWL is based on the Description Logic $\mathcal{SHOIN}$(D) [131]. It provides a richer set of modelling constructs and semantics compared to RDFS. OWL is a family of languages, consisting of the full, undecidable language OWL-Full (a true extension of RDF), and two description logic based languages OWL-DL and the more restricted and tractable OWL-Lite (the description logic $\mathcal{SHIQ}$). In essence, OWL-DL and OWL-Lite can be seen as webized versions of their respective description logics. OWL-DL and OWL-Lite have both a model theoretic semantics, and a semantics based on a vocabulary extension to the existing RDFS semantics [202]. OWL uses RDFS to provide the vocabulary modelling language, and uses XML/RDF* as its surface syntax for interoperability on the web

Reasoning with OWL is decidable but not tractable. Tractable languages are typically seen as having a polynomial, or lower, computational complexity*. OWL-DL on the other hand has a worst case reasoning complexity of NEXPTIME [253]. As a result, reasoning in the worst case results in long, undesirable computation times. Since OWL's advent, much research has centred around finding highly tractable subsets of description logics and thus OWL, for example DL-Lite[26], EL++ [7] and Horn-$\mathcal{SHIQ}$[135].

Summary: Description logic ontologies are subsets of First Order Logic (FOL) and have seen widespread use within Artificial Intelligence and now the Semantic Web, as a mean to encode and reason with knowledge. Within this thesis we aim to utilise these languages to represent and reason with Semantic Web geographic ontologies. As the web ontology standard OWL is, in effect, a webised version of a particular description logic, in the next section we describe in greater detail the formal underpinnings of description logics as a knowledge representation paradigm. Further to this we explore another popular knowledge representation and reasoning

*http://www.w3.org/XML/

*Although there is not guarantee in practice that a polynomial theoretical complexity actually translates to real world tractability paradigm, namely logic programs. Of interest to this thesis, logic programs or rule based systems are increasingly being used on the Semantic Web as a way to add certain reasoning possibilities that description logics are lacking, and as a way to achieve extra-logical functions.

3.2 Logic and Knowledge Representation

'The purpose of logic is to characterise the difference between valid and invalid arguments. A logical system for a language is a set of axioms and rules designed to prove exactly the valid arguments statable in the language' *

The first use of the word logic dates back to Aristotle. Aristotle was an ancient Greek philosopher who studied, amongst others, logics or more accurately dialectics - a logical method of philosophy. He was the first to study logics as an independent discipline. Aristotle's compiled six works on logic which were known collectively as the Organon.

The next major work on logic was by the mathematician George Boole. Boole was originally interested in replacing Aristotelian syllogistic logic (deductive reasoning) with a mathematical-style 'algebra'. Boole went on to invent boolean algebra / logic which was later adopted by Claude Shannon in his 1937 PhD thesis [28] which later went on to form the basis of all modern digital computers.

Gottlob Frege is widely regarded as the founder of conventional logic. Frege amongst others, is accredited with being the inventor of modern quantification theory which solved the problem of multiple generality, making the distinction between inference rules and axioms, and placing a distinction between concept and object . Frege's predicate logic was later restricted by Bertrand Russell and John von Neumann, after Russell found a paradox with the original theory, to First Order logic. First Order Logic is a powerful enough language to formalize all of set theory and therefore most if not all of mathematics. The current hierarchy of logics date back to this seminal work.

In this section we overview two subsets of First Order Logic (FOL) used for knowledge representation, namely description logics and logic programs.

*from `http://plato.stanford.edu/entries/logic-modal/`

3.2.1 DESCRIPTION LOGICS

Early (late 1960s) computerized ways of conceptualizing and representing domain knowledge were based on Semantic Networks [245]. A semantic network defined concepts (as nodes in a graph) and the relationships (as arcs in a graph) between those concepts. Semantic networks were however not very formally defined and suffered from ambiguity in the interpretation of their constructs.

Frame based systems become a prominent knowledge representational paradigm in the 1970's. Modelling in frame based systems is closely related to modelling in the object oriented (OO) paradigm. A frame (a class in OO terms) represents a set of objects and each frame can contain properties known as slots. A slot represents either a value or a relationship between two frames. A hierarchy of frames is possible using the principles of inheritance. Indeed the key inference task of a frame based system is to determine any implicit inheritance hierarchies (subsumption reasoning). In 1995 Frame based languages and the object oriented paradigm came together to produce F-Logic [149]. Frame based systems try to closely mimic the human representation of the real world [182, 70], and while they are good at representing structural information, they are restricted in their ability to deal with asserted knowledge (ground individuals) [11]. This led to a dichotomy of representation, a terminological or structural component containing a hierarchy of concepts, and an assertional or ground component containing observations of the real world [11].

To deal effectively with both structural and ground knowledge, description logics [188] were proposed that encompassed both types of knowledge, denoted by the TBox (terminological box) an ABox (assertional box). Description logics are subsets of full first order logic and hence have well defined semantics. Description logics stem from the seminal work of Brachman et al. in 1995 when developing their system KL-ONE [21]. KL-ONE logically formalized the ideas of both frame based and semantic networks. Inherently a description logic can describe the world in terms of properties or constraints that specific individuals have to satisfy. A description logic can describe concepts, concept hierarchies, roles and individuals. Thanks to their formal logical semantics, description logics support the following key inference tasks [188]:

(a) Subsumption reasoning - given concepts C and D, determine if C is a subset of D. Checking if the concept D is more general than C.

(b) Membership checking - checking whether an individual i is a member of the concept C, or finding all individuals that are instances of C (a query).

(c) Satisfiability checking - given concept C determine if C is logically consistent with other statements in the current knowledge base. Checking whether a concept expression does not denote the empty set.

For certain families of description logics (for example $\mathcal{SHIQ}$ in OWL-Lite) all major DL reasoning problems can be reduced to satisfiability checking [127]. From a practical perspective, it is important that these inference problems remain decidable. However as the expressiveness of the language increases, so too does the worst case complexity of reasoning with the language [20]. Throughout the literature various families and variants of description logics have been defined, each identified by their set of modelling constructs.

A number of mature description logic reasoning engines exist, for example FACT [133] and Racer [114]. Early forms of description logic reasoning were based on structural comparison techniques. However if concept negation is allowed e.g. ¬City, structural comparison is no longer sound, instead tableau based algorithms are used. These days nearly all description logic reasoners are based on tableau calculi [229].

3.2.1.1 Description Logic Families and Notation

In this section we describe families of description logics and their modelling constructs, through an evolving series of example description logic knowledge bases (simple ontologies).

$\mathcal{ALC}$ (Attributive Language with Complements) is the base description logic. All description logics based on $\mathcal{ALC}$ contain the following constructors (as shown using Backus Naur Form*):

$$ALC ::= \bot \mid A \mid \neg C \mid C \wedge D \mid C \vee D \mid \exists R.C \mid \forall R.C \tag{3.1}$$

Where C and D represent concepts (unary relationships or classes), R represents

*Backus Naur Form is a popular way of defining language grammar and syntax a role (binary relationships between two concepts), and A represents an atomic concept. $C \wedge D$ defines a complex class formed from the intersection of the classes C and D. $C \vee D$ defines a complex class as the union of the classes C and D. $\neg C$ defines the complement of the concept C. $\exists R.C$ states that some of the roles R must have a range of type C. $\forall R.C$ states that every role R has a range of type C.

Different variants of the base $\mathcal{ALC}$ are then possible by combining different modelling constructs together. An overview of the varying constructors that can be added to $\mathcal{ALC}$ is shown in Table 3.1*

DL Concept Constructors	Meaning	OWL Equivalent Constructor
F	functionality ($\leq$ R)	Functional Property
N	unqualified number restrictions ($\geq$n R), ($\leq$n R)	AllValuesFrom and SomeValues From
Q	qualified number restriction ($\geq$n R.C), ($\leq$n R.C)	
S	Role Transitivity	Transitive Property
H	Role Hierachy $R \subseteq S$	subproperty of
R	Complex role inclusion	as of OWLv1.1 complex property composition
I	Inverse roles R^{-}	Inverse property
O	nominals $\{a\}$ or $\{a_1, ..., a_n\}$	enumerated classes
s	other features	

Table 3.1: Description Logic constructors and their OWL counterparts

As previously described, OWL-DL is based on the description logic $\mathcal{SHOIN}$(D) where (D) symbolises a concrete datatype domain. In the concrete domain of datatypes, the entire lexical space and lexical values of each datatype are known. S is a substitute for $\mathcal{ALC}$, while also adding role transitivity R - therefore $\mathcal{SHOIN}$(D) is actually the DL $\mathcal{ALCRHOIN}$(D).

Subscripts typically represent role constructors, for example $\mathcal{ALC}_{\cup}$ represents the DL $\mathcal{ALC}$ with role union. $\mathcal{ALC}_{\cup\cap}$ represents the DL $\mathcal{ALC}$ with both role union and role intersection. Sometimes this information is represented within preceding

*Please see `http://www.cs.man.ac.uk/~ezolin/dl/` for a guide to description logic reasoning constructs and complexities brackets of the base description logic, for example $\mathcal{ALC}(\sqcup\sqcap)$ which represents the same as the above.

Semantics: Description logics have a model theoretic semantics based on a Tarski-style interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. For example for the base description logic $\mathcal{ALC}$, $\Delta^{\mathcal{I}}$ is a nonempty set (the domain), and $\cdot^{\mathcal{I}}$ is an interpretation function that maps:

- A concept name $\mathbf{A} \mapsto \mathbf{A}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ (subset $\mathbf{A}^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$)
- A role name $\mathbf{r} \mapsto \mathbf{r}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ (a binary relation)
- An individual name $\mathbf{a} \mapsto \mathbf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ (an element of $\Delta^{\mathcal{I}}$)

As description logics i.e. OWL, can include a concrete datatype domain (denoted with a (D) postfix), the interpretation can become the tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \Delta_D, \cdot^{\mathcal{I}})$, where Δ_D is a nonempty set of data values. The exact semantics of complex classes, properties and datatypes per description logic is then defined, see for example OWL-DL in [98].

3.2.1.2 Description Logic Constructor Examples

In order to demonstrate the modelling benefits of using each of the DL constructs shown in table 3.1, this section will build, from the ground up, an example ontology while progressively using different description logic constructors.

Concept expressions: A class of objects can be captured by a concept. For example, if you want to represent an ontology about Cities and Towns, the following DL concepts or classes can be used e.g.

$$City$$
$$Town$$

Here, each concept is not related to each other. However, description logics are designed to represent terminological hierarchies using the general concept inclusion axiom $C \sqsubseteq D$ (indeed a TBox is a finite set of concept inclusion axioms). Hence, it is possible to either combine concepts to create more general concepts, or subdivide concepts to create more specialized concepts, allowing for the representation of the geographic domain on different levels of granularity or abstraction. The resulting hierarchy of classes is of benefit in the following ways [73].

(a) It better matches the human view of the world.

(b) Hierarchies are extensible, allowing global upper level ontologies to be further specialised into domain and application specific ontologies.

(c) The hierarchy allows for more 'intelligent' reasoning / query expansion. By traversing the hierarchy a number of unwritten facts can be inferred.

(d) The potential for information integration between two ontologies is greater.

For example, if we wanted to represent both the City and Town concepts as subclasses (specialisations) of a geographic feature or Geofeature, the following DL ontology can be constructed.

$$\begin{aligned} Geofeature \\ City \\ Town \\ City \sqsubseteq Geofeature \\ Town \sqsubseteq Geofeature \end{aligned}$$

Roles: Roles are used to represent binary relationships between two individuals (denoted an ObjectProperty in OWL). If the DL contains a concrete datatype domain, then a role can represent a binary relationship between an individual and a datatype (denoted a DatatypeProperty in OWL), akin to a simple class attribute. For example, we can add both a *name* property and a *partOf* spatial relationship to the City concept.

$$City = Name \cap partOf$$

To constrain the range of the properties further, we can add both qualified and unqualified number constraints (N, Q). Qualified constraints can ensure that the range of the role $partOf$ has a value which is a member of a new concept Country (as shown below). Also note the universal quantifier was placed on the range restricted $partOf$ role. Intuitively, this states that **all** $partOf$ roles of that concept must adhere to this restriction. Furthermore, each city has 1 or more names e.g. vernacular names or alternate spellings etc.

$$\begin{aligned} &Country \\ Country &\sqsubseteq AdministrativeRegion \\ City &= \geqslant 1.Name \sqcap \forall partOf.Country \end{aligned}$$

Adding the H construct enables representation of role hierarchies - similar to concept hierarchies. For example, we could represent topological spatial relationships with a varying degree of granularity by the following.

$$\begin{aligned} Overlap &\sqsubseteq SpatialRelationship \\ PartOf &\sqsubseteq Overlap \\ Non\_Tangental\_Proper\_Part &\sqsubseteq PartOf \\ Tangental\_Proper\_Part &\sqsubseteq PartOf \\ Equal &\sqsubseteq PartOf \end{aligned}$$

A $partOf$ spatial relationship is inherently transitive and can form the basis of a spatial containment hierarchy. Adding the S DL constructor allows representation of transitive roles, suitable for example to capture the transitive nature of a containment hierarchy.

$$PartOf^{+} \sqsubseteq PartOf \tag{3.2}$$

With this added, we can now infer that Cardiff City is $partOf$ the United Kingdom if Cardiff City is $partOf$ Wales, and Wales is $partOf$ the United Kingdom (assuming Cardiff, Wales and United Kingdom were contained in the ABox of the description logic).

The topological spatial relationship *contains* is the inverse relation of the topological spatial relationship $partOf$. Inverse roles are possible using the I DL constructor.

$$partOf \equiv contains^{-}$$

Functional Properties: Functional properties, the F DL constructor, allows the representation of a property which has at most one value*. Such a construct is useful in defining unique properties. For example, if the City concept were to contain a unique identifier, the ontology would look as follows:

$$City = \leqslant 1.ID \sqcap \geqslant 1.Name \sqcap \forall partOf.Country$$

Nominals: Nominals can be used to restrict either a concept type or a role's range, to a certain set of individuals (also referred to as one-of). For example, if we wanted to restrict the range of an isCapital role to only the values True or False, our City concept can be augmented with the following:

$$City = \leqslant 1.ID \sqcap \geqslant 1.Name \sqcap \forall partOf.Country \sqcap isCapital.\{True, False\}$$

Adding this construct in addition to inverse roles adversely effects the theoretical computational complexity of the description logic [127]. Treatment of nominals with other constructs within DL reasoning engines is an ongoing research effort [197]. Indeed OWL-Lite, which corresponds to the DL ($\mathcal{SHIQ}$), syntactically omits the use of nominals.

Complex Role Inclusion: Complex role inclusion axioms, the R DL constructor, have the general form $R \circ S \sqsubseteq T$. Less formally, the class or role concept T is made from the chaining or composition of the properties R and S. Complex role inclusion axioms, when combined with certain DL constructs e.g. existential quantification, make the language undecidable [130]. However restriction to inclusion axioms of the form $RoS \sqsubseteq R$ or $RoS \sqsubseteq S$ and by making sure that any axioms of such type are acyclic, maintains decidability. Complex role inclusion axioms are not present in OWL 1.0, however restricted complex role inclusion axioms are included in the new standard OWL 2.0.

*As per the mathematical definition of a functional property

Necessary and Sufficient Concept Expressions: Within FOL the equality ($\equiv$) operator between two concepts C and D (i.e. $C \equiv D$), is represented by the following two first order inferences.

$$D(x) \rightarrow C(x) \tag{3.3}$$

$$C(x) \rightarrow D(x) \tag{3.4}$$

That is, formula 3.3 states that belonging to D is necessary to be a member of C. Whereas formula 3.4 states that belonging to C is sufficient to be a member of D. In essence these two concepts are equivalent and therefore share the same set of individuals (members). For example in the DL ontology, two synonymous names, one in English the other in German for the same concept City, can be equated using the following:

$$City \equiv Stadt$$

3.2.2 Logical Reasoning, Logic Programming and Rules of Inference - Syllogisms

Reasoning is an important function of human intelligence. Human cognition uses a form of autoepistemic (self knowledge) and defeasible (default assumption) reasoning to generate new knowledge based introspectively on previously learnt knowledge. Aristotle's original logic, followed later by all developed logics, is designed to understand human reasoning processes more formally. In all there are ten known distinct ways to derive implicit knowledge from raw explicit knowledge, these are: Deduction, Induction, Intuition, Heuristics, Generate and Test, Abduction, Autoepistemic, Nonmonotonic, Analogy and Default. As logic provides the foundations of modern computers, ways of applying logical reasoning methods to machines have been the subject of a vast body of research for the past half century - known more generally as the field of Artificial or Machine Intelligence. Within this thesis, we focus on two of these, namely deduction (standard inference rules) and default reasoning.

An inference rule captures axioms or heuristic knowledge of a domain. An inference rule is made up of an antecedent (or body) and consequent (or head).

If the antecedents of the rule are true, then the consequent is entailed as a result. More formally, a rule r infers a new fact f ($r \models f$) iff all the antecedents of r are true. A rule set therefore consists of a number of individual rules of the form:

$$Antecedent \rightarrow Consequent$$

Early inference rules where based on propositional* logic. Propositional logic can not deal with parts of a statement such as the subject or predicate, instead only dealing with the truth of individual statements and complex statements joined using logical operators (e.g. and, or) e.g. cardiff ∧ capital → city. Propositional logic has a well understood and relatively simple semantics which can be shown and proven using a truth table. Propositional logic uses a number of rules of inference to derive new statements, for example Modus Ponens † and Modus Talons.

Propositional logic was extended to predicate logic, which can focus not only on statements as a whole, but on the structure of the atomic propositions, hence allowing the expression of predicates and subjects within each atomic proposition. Predicate logic includes constants, variables and functions (although it is well known that functions of arbitrary arity make the language, and indeed all First Order Logic, undecidable). For example, in the geographic domain, representing the fact that everything is a geofeature can be easily expressed in predicate logic using variables. That is, the variable x as the subject of the predicate *Geofeature* to give $Geofeature(x)$. Essentially this predicate represents a set of *things* that are Geofeatures - a subset of individuals in the current domain of discourse. To represent the same information in propositional logic, a large number of propositional statements would need to be made e.g. Cardiff is a Geofeature, Wales is a Geofeature ... the Town Hall is a Geofeature etc. First Order Predicate Logic also allows variables to be quantified using For All (denoted $\forall$) and There Exists (denoted $\exists$), as also shown from a description logic context in the previous section. For example all *Geofeatures* are Things: $\forall x, Geofeature(x) \rightarrow Thing(x)$, and some Geofeatures are Cities: $\exists x, Geofeature(x) \wedge City(x)$.

Reasoning with full first order predicate logic requires an extended set of inference rules over propositional logic to deal with variable quantification, these

*A proposition is a concept that is either true or false

†Incidentally Modus Ponens is only complete for Horn clauses, and can not deal with more expressive logic features such as disjunction

are: Universal Introduction, Universal Generalization, Existential Instantiation, and Existential Generalization. However, an important result in the last century found that, all classical rules of inference can be replaced by one rule of inference called Resolution. Resolution is both sound and complete for logical programs in normal form (all predicate logic) [222]. This simplifies the effort to construct a theorem prover or reasoning engine, and led to the development of Prolog. Furthermore, in practice, predicate logic requires an additional unification operator in order to substitute variables for literals, a process known as grounding. Inference rules written in first order predicate logic are often simply referred to as rules.

Rules are written within a particular rule language. Rule languages, like their logical foundations, are declarative. That is, a rule specifies what to do but not how to do it. Most rule languages trade expressibility for decidability, and most are commonly built upon decidable fragments of first order predicate logic. Horn logic, named after its inventor Alfred Horn in 1972, is such a subset of FOL which omits any form of negation [238, 87] and only allows a definite conclusion (one predicate in the consequence). A Horn logic program (a set of Horn logic statements or Horn clauses) is monotonic in the sense that only new facts are entailed and added to the knowledge base, none are removed during the execution of a rule set. More formally a Horn clause, as expressed using propositional logic, has the form:

$$A \vee \neg\, B \vee \ldots \neg\, B_n : \mathrm{n} \geq 0.$$

Which is more commonly written (with identical semantics):

$$A \leftarrow B \wedge \ldots \wedge B_n : \mathrm{n} \geq 0.$$

where A is the head proposition, and $B \ldots B_n$ are body propositions. A Horn clause can be expressed also in predicate logic, having the form:

$$A(t_1 \ldots t_n) \leftarrow B(t_1 \ldots t_n)_1 \wedge \ldots \wedge B(t_1 \ldots t_n)_m$$

Where $A(t_1 \ldots t_n)$ and $B(t_1 \ldots t_n)$ are positive predicates, $t_1 \ldots t_n$ are terms, and $m \geq n \geq 1$. The implication operator $\leftarrow$ affirms truth to the head predicate A, on the truth of the body predicates $B \ldots B_n$. Alternatively, a Horn rule can be written identically in a forward direction, for example:

$$B(t_1 \ldots t_n)_1 \wedge \ldots \wedge B(t_1 \ldots t_n)_m \rightarrow A(t_1 \ldots t_n)$$

Where the implication operator $\rightarrow$ has an operational behaviour which, on matching of the body, the head is entailed. A rule with no body predicates i.e. where $m = 0$, is a fact.

In addition to the base Horn case, two extensions have been proposed. A 'general logic program' [86] is an extension to the basic Horn clause in that it allows negation as failure, denoted naf, which allows negatively represented predicates e.g. $City(x) \wedge not(Large(x)) \rightarrow smallCity(x)$. An 'extended logic program' [85, 269] adds both negation as failure along with a stronger negation more akin to, but not synonymous with, classical negation. By extending the logic with either form of negation leads to more complicated semantics, and inherently makes the language non-monotonic and thus capable of defeasible (default) reasoning, where a form of defeasible reasoning is described in section 3.2.3.

3.2.2.1 Datalog and Disjunctive Logic Programming

Datalog is a rule language designed in the mid 1980's for expressing recursive rules over databases, leading to the invention of deductive databases (see deductive databases in [212]). Datalog is based on the logic programming paradigm. A plain Datalog rule corresponds to a Horn rule in that both share the same syntax and declarative semantics

Some of the capabilities of Datalog were incorporated into recursive SQL (SQL-99) as described in numerous technical documents e.g. [58]. However despite this, Datalog is often referred to in the literature due to its strong and well studied theoretical foundations.

Datalog is restricted syntactically to preserve decidability and maintain tractability, where plain Datalog has a polynomial computational complexity. Datalog restrictions are often referred to throughout the literature. If a rule language conforms to the Datalog restriction then the rule language in question does not include functions symbols or any form of negation. Function symbols are well known to make a rule language and full First Order Logic undecidable by reduction to the Halting problem. Datalog rules also conform to the safety condition which states that, each variable occurring in the head of the clause must occur in the body of the same clause. The safety condition guarantees that the set of all facts that is derivable from the Datalog program is finite, and hence the program is decidable (such safety is also discussed again in Chapter 4).

Extending Datalog: As a Horn subset of FOL, Datalog is a relatively inexpressive language compared to most description logics. In the late 1980's extensions to the base Horn logic programming paradigm were being investigated. One such extension, disjunctive logic programming, came to the forefront when Minker devised a consistent theory of negation for disjunctive deductive databases - the generalised closed world assumption (GCWA) [181]. This was a continuation of his previous work in the field of disjunctive logic programming which he started in 1982 [180]. A disjunctive logic program extends a Horn logic program by allowing the disjunction of predicates in the head of a rule, for example:

$$B(t_1...t_n)_1 \wedge ... \wedge B(t_1...t_n)_n \rightarrow H(t_1...t_n)_k \vee ... \vee H(t_1...t_n)_m \quad (3.5)$$

Where $B(t_1...t_n)$ and $H(t_1...t_n)$ are positive body and head predicates, $t_1...t_n$ are terms, and k = 1, m $\geq$ n $\geq$ 1. Allowing head disjunction leads to inferences with multiple possible states, or minimum models - one for each disjunct in the head. Then, as each possible disjunct of one inference can be combined with disjuncts of further inferences, a disjunctive logic program is non deterministic and typically exhibits a computational complexity of NEXPTIME [61]. Disjunctive logic programs where incorporated in Datalog In 1997, when Thomas Eiter et al. defined disjunctive Datalog [60, 61], denoted Datalog$^{\vee}$ as well as Datalog with or without negation $^{\vee,\neg}$.

The most prevalent semantics for disjunctive Datalog is that of the disjunctive stable model semantics [210]. Such semantics are used as the basis for the now popular answer set programming [206] using answer set semantics, where answer set semantics are a variant of stable model semantics for negation and disjunction - as described in more detail in section 3.2.2.2. As a disjunctive Datalog program has potentially multiple minimal stable models or answer sets, the result of a computed query can be defined as either the union or the intersection of those models. The union is usually referred to as brave or credulous reasoning while intersection is referred to a cautious or skeptical reasoning [61].

In general, disjunctive rules can mimic more closely human reasoning than a definite rule. Indeed, recently disjunctive rules have received a lot of attention as a core representational paradigm from which to integrate rules and ontologies [6]. However query answering in Disjunctive Datalog still has a poor worst case complexity for query answering of coNEXPTIME^{NP} [61].

3.2.2.2 Logical Rule Semantics

In this section we discuss, more formally, a number of key semantics develop to formally describe the various logical subsets of FOL used by existing rule based systems.

All logic programs (a set of logical rules) should be given both a declarative and procedural semantics. The declarative semantics should define the answer set of a logic program independent of how it is implemented and executed. That is, separating the *what* from the *how*. A procedural semantics should then define a logic programs answer set when considering the steps involved in its construction - input output. Sometimes a procedural semantics gives an indication of how the inferencing is performed, and thus is often referred to as operational semantics.

As already noted, most logic programs use rules based upon a decidable subset of first order logic, namely Horn logic. A Horn Logic program consists of a rule set syntactically and semantically akin to Horn clauses - Horn rules. A Horn rule is often referred to as a definite rule as it only has one head (definite) and a conjunction of body literals. Horn rules are monotonic in the sense that knowledge is only ever added to the underlying system of facts, knowledge can not be removed. They are restricted to function symbols of arity 0 (function symbols with no terms), and as such their semantics are computationally tractable and well understood. There are two prevalent procedural and declarative semantics of a Horn logic program which are, least fixed point semantics and minimum model semantics (least Herbrand model), for example see [209, 88]. Importantly, a Herbrand model has both a Herbrand Universe, which is the set of all ground* terms found from the constants and function symbols in the logic program, and a Herbrand base, which is the set of all ground goals formed from predicates in each rule over the Herbrand Universe.

*A ground fact or predicate contains no variable, only constants

Non-monotonic reasoning, as is inherent in common sense reasoning, allows not only the addition of knowledge but also its removal. Negation, classical ($\neg$) or weak (naf), is the most famous non-monotonic operator, and can refute a conclusion that has already been drawn after the addition of new knowledge. For example, take the following simple propositional logic program Π:

$$s \leftarrow naf(a)$$
$$a \leftarrow$$

Where the negation as failure (naf) operator succeeds iff it does not find its subject term (a in this case) in Π . When a is added to Π, s is no longer valid and is therefore refuted.

The semantics of logic programs that contain non-monotonic operators, in particular weak negation or negation as failure, are difficult to define formally. Logic programs with non-monotonic features can contain multiple minimum models based on which order the program was executed. That is to say, most paradoxes occur when the negation of a predicate has not been determined before it is used. For example the simple logic program Π above has two minimum models M, one where a exists ($M = \{a\}$), and one where both a and s exist ($M = \{a, s\}$), which of course depends on the order the program was run. Finding their unique minimal model, or guaranteeing a consistent answer set is not trivial. Early work in the area was based around Clarke's work on program completion [32]. Clarke defined the notion that facts not entailed by the rules of the program were treated to be false.

Motivated by autoepistemic logic, Gelfond and then furthered by both Gelfond and Lifschitz, defined a semantics for negation through stable models [84, 86]. A stable model, relative to a minimal model Pos(M) (the amount of positive atoms in a model), is an interpretation of a program which reproduces itself under a three stage transform know as the Gelfond and Lifschitz Reduct. After applying the reduct, assuming a minimal model M of program Π, if a stability transform of M denoted $S(M)$ is equal to the original model M of Π i.e. $S(M) = M$, then M is said to be a stable model of the program Π. Each minimal model of Π is tested for stability, if only one matching model is found then Π is said to have a unique (stable) minimal model. Each model is an intended meaning (possible interpretation) of the program.

As previously noted, stable models have paved the way for a new programming paradigm which is gathering pace for use on the Semantic Web, namely answer set programming. Answer set semantics incorporate both types of negation namely, classical negation and negation as failure naf.

Stratifying the logic program is another semantic treatment of programs with negation [208]. A program can be stratified if there exists a partition of the program:

$$P = P_1 \cup ... \cup P_i \tag{3.6}$$

Such that 1) if a predicate symbol occurs positively in a clause in P_i, then its definition is contained within $\bigcup_{j<=i} P_j$ and 2) if a relation symbol occurs negatively in a clause in P_i, then its definition is contained within $\bigcup_{j<i} P_j$. Informally, the existence of a negative predicate is proved or disproved before any predicate that depends on it is evaluated. Stratification was extended by Przymusinski [209], termed local stratification. Importantly, when locally stratified, predicates may depend negatively on themselves.

This far all semantic treatments mentioned are based on classical two valued logics. However, many see recourse to a three valued or even multivalued logic to be an intuitive way of dealing with negation. The well-founded semantics [263] is widely adopted as the intended semantics of a logic program that uses negation as failure, when only a unique answer set is required. The well-founded semantics utilises a three valued logic (True False or $\perp$ - undefined) and tries to provide a clean semantics for negation as failure. The well-founded semantics is seen as an intuitive semantics where certain truths, which can not be determined by the facts and rules, are returned unknown. In [223], a infinite valued logic is defined to give a purely model theoretic and thus order independent characterisations of logic programs with negation as failure.

3.2.3 Defeasible Reasoning

Defeasible (default) reasoning tries to closely mimic a common sense human approach to reasoning. The link between body and head or an inference rule is tentative, and may later be refuted by the addition of new knowledge. For example, the following is an example of a spatial integrity constraint with both a default rule and an exception to that rule.

$$\text{default: } \forall X, Y \;\; \mathit{Road}(X) \wedge \mathit{River}(Y) \wedge \mathit{Crosses}(X,Y) \rightarrow \mathit{error}(\mathit{roadRiverCrossError}) \tag{3.7}$$

$$\text{exception: } \mathit{Road}(A40) \wedge \mathit{River}(\mathit{Taff}) \wedge \mathit{Crosses}(A40, \mathit{Taff}) \rightarrow \neg \mathit{error}(\mathit{roadRiverCrossError}) \tag{3.8}$$

Here, rule (3.7) is a default rule and rule (3.8) is the exception to the default. That is, the ground instantiation of rule 3.7 which substitutes variables X and Y for A40 and Taff respectively should be overridden by rule 3.8, and no error inferred for this road and river couplet.

Reiter proposed a defeasible, non-monotonic logic for reasoning with default assumptions, namely; default logic [215]. Such a logic is useful for dealing with situations where, by default something is true, but it may not be in all cases. Reiters logic has been widely criticised in that, dependent on the order in which inferences are run both conclusions are possible - in Reiter's default logic there is no given ordering. The prioritisation of rules such that certain rules can be explicitly stated to run before others has been widely looked at in the literature [101, 179, 8, 177]. In these, more than one rule premise can succeed, but only the rule with highest priority is actually fired.

In 1997, Grosof developed Courteous Logic Programs (CLP). A CLP is one where contradictions resulting from rules can be solved by imposing partially ordered pairwise prioritisation between rules using the reserved binary predicate *overrides(i,j)*. That is, if a rule implies $\neg p$ and another refutes such a claim, therefore implying p, a contradiction occurs. If no priority is specified then neither are inferred (treated sceptically), otherwise the rule with greater priority succeeds. A CLP guarantees a consistent and unique set of conclusions.

A possible implementation of Courteous Logic conforms to the well founded semantics [263]. However, Courteous logic programs had little impact outside the knowledge representation (KR) community, especially to non KR experts who predominately used Prolog, as they had little knowledge of how to deal with a CLP. In an attempt to overcome this, DIPLOMAT [102] was developed in 1997 as a Java based system that compiles a CLP into an ordinary logic program as a pre-processing stage. It also extends the original ideas of a CLP into generalised courteous logic which has the following additions; reasoning is allowed about the *overrides* predicate (*overrrides* can be a rule as well as a fact), cyclic rules are allowed, and the addition of a mutex predicate, which allows the user to specify their own pairwise mutual exclusions along with the implicit classical negation *mutex*. An opensource courteous logic compiler has been included into the now popular SweetRules implementation*.

3.3 SUMMARY

Description logics (DL's) provide a good base for ontology languages thanks to their numerous modelling constructs and inference mechanisms. OWL is a very expressive description logic variant and hence has a number of useful modelling features for representing geospatial information, as shown during construction of an example geographic ontology in section 3.2.1.2. However the DL's used by OWL and indeed the new OWL 2.0 are both intractable [155], and certain subsets of them are needed to regain tractability.

Rule languages and logic programs are practical, mature technologies for large scale reasoning tasks, however they often lack the modelling abilities of description logics - typically being only subsets of expressive description logics. With regard to expressive rule languages with complex semantics, we believe that, motivated by the need to enforce a single consistent answer (unknown is too weak an answer to return) three valued logics can be rules out. Most end-users of geographic information systems will not have practical knowledge in the application of knowledge based systems. We argue they will often expect a definite positive result without ambiguity, or the need for further manual refinement over complex and large data sets, whereby errors may occur from intertwined strands of knowledge chaining and inference.

*see `http://sweetrules.projects.semwebcentral.org/`

3.3 Summary

Classical 2-valued semantic approaches still have their limitations. Stable models can be intractable to compute (NP-complete see [172] in [256]) unless aided by pre-processing techniques, and do not always produce unique answer sets if negation is used. Stratifying a logic program may not always be possible if a suitable partition can not be found, and requires an extra step to compute.

Defeasible reasoning is a very good mechanism to represent default assumptions and their exceptions. This can be useful in integrity checking scenarios where default integrity rules, which hold for the general case, can be overridden by individual case by case exceptions.

In the next Chapter we review OWL as both a spatial and integrity checking language, and conclude by motivating a new spatial ontology language for use on the Semantic Web.

Chapter 4

OWL and Rules for Spatial Data Management

As shown in the previous chapter, OWL is an expressive knowledge representation language and succeeds in providing a rich modelling environment which can, and is, being used to formally represent millions of Semantic Web ontologies* .

This chapter explores the potential of using OWL as a language to represent and manage geographic ontologies. This is followed by motivating the the use of a rule language and logic program as a suitable knowledge paradigm, to help overcome found representation and reasoning issues discovered when OWL is used for geographic information.

4.1 Integrity Maintenance in OWL

In this section we explore how suitable the logical characteristics of OWL are for spatial data management. In particular, we look at the effects the open world and non-unique names assumption in OWL has on integrity checking.

First Order Logic (FOL) adheres to an open world non-unique name semantics. All families of description logics are subsets of first order logic and hence fit firmly within the same semantic framework. The open world assumption assumes that all known knowledge maybe incomplete, and so does not make any introspective

*As taken in 2009 from the statistics of the Swoogle Semantic Web Search engine `http://swoogle.umbc.edu/`

judgements based on the absence of knowledge. As an alternative, a closed world assumption is introspective and assumes that all known knowledge is total, the domain of individuals is finite and known, and hence the absence of knowledge provides conclusive proof that it does not exist. The open world assumption is actually beneficial on the web. The web is incomplete and continually evolving, data providers (from personal web pages to commercial outlets) are in the process of updating old information and exposing new information. At any one point in time, the knowledge that already exists on the web can not be assumed complete. This fundamental characteristic of web knowledge, representative of the open world assumption, is important to retain in a general web ontology language, so as to truly mimic the ever evolving formation of the web. However, any description logic based ontology language i.e. OWL, using these semantics can not be used to test the consistency of, for example, individuals that commit to that ontology [46].

A non-unique name assumption does not assume that individuals with different names or identifiers are distinct. This is beneficial on the web as often you find two equivalent objects created by different users with different names. However, without a unique name assumption, objects can not be counted. That is, you do not know which are discrete and can be counted as separate entities, or which are the same and should be counted as the same entity. Consequently, OWL can not easily express number restrictions on properties which would restrict and thus prevent the addition of values to that property beyond that it is restricted [46], as shown later. Furthermore, judgements about the relations between individuals can not be made if they can not be assumed separate entities.

Of interest, constraints in OWL are allowed on datatype properties. The domain of datatype properties is concrete and outside of FOL. The lexical space and values for all datatypes is known, a closed world, and hence constraints on datatype properties are checked.

Effects of the Open World and non-unique Name Assumption: To demonstrate the effects of the open world and unique name assumption on integrity checking, we highlight qualified cardinality constraints in OWL *, and how they can not be used to constrain and check the possible instantiations of a class.

*Qualified by a number restriction

Within an integrity checking version of OWL, maximal or minimal cardinality constraints should be violated if the asserted (instance based) information does not conform to such restrictions. However in OWL, where a cardinality constraint is present that is not adhered to, either information is inferred to satisfy the constraint (if the restriction is met), or no negative assumption is made about the absence of information (if the restriction is not met). As an example, take the following OWL definition of a Polygon:

$$\text{Polygon} \sqsubseteq \geq 3.\text{XYCoords}$$

If an individual of type Polygon had two *XYCoords* the open world assumption would concede that information may exist external to the ontology which can later be added to satisfy the restriction. If an individual had more than three *XYCoords* then, as OWL does not support the unique name assumption, equality would be inferred for all those coordinates greater than the restriction e.g. the fourth *XYCoords* instance would be inferred to be an equal individual to one of the first three, and so on.

As noted by Reiter, schema integrity constraints are inherently epistemic in nature [216], that is they rely on self knowledge or known knowledge to decide integrity. This requires a closed world and unique names assumption, such as those employed by typical relational databases.

4.2 OVERCOMING OWL'S INTEGRITY LIMITATIONS

The ability to switch on the closed world assumption, or to alternate between both closed and open world formalism has received much attention in recent years[147, 100]. The closed world assumption is said to be more intuitive as a modelling formalism [46], particularly as over the past two decades knowledge has principally been stored in relational or object relational database management systems, where closed world non-monotonic semantics are the norm. Moreover, domain experts will be more familiar with the typical notion of a constraint in a closed environment, that is, constraints that represent integrity rules rather than logical axioms [52].

In this section we present an overview of the various methods that facilitate forms of traditional closed world integrity maintenance within OWL. Such approaches can be categorised as follows:

- Modelling domain closure intrinsically within OWL.

- Extending OWL with additional operators to locally or globally close the domain.

- Translating subsets of OWL into a logic programming formalism which assumes both the unique name and closed world assumptions.

4.2.1 DOMAIN CLOSURE

The first version of OWL (OWL 1.0) contains two axioms that can be used to express explicit disjointness between individuals, and hence for these individuals apply the unique names assumptions, namely; *owl:differentFrom* and the larger scoped *owl:allDifferent*. *owl:differentFrom* allows the user to specify pairwise disjointness of individuals or concepts. However, this is rather verbose when specifying disjointness between possibly hundreds of individuals, which is needed for complete domain closure*. *owl:allDifferent* overcomes this issue by allowing the explicit representation of disjointness between a collection of individuals. Although better, it still does not provide a simplified way to represent the unique name assumption for a large number of individuals - statements for each individual would still need to be added to the ontology. Moreover, this is not very intuitive from a modelling perspective. That is, a domain expert would assume that each individual in their ontology is disjoint (distinct) from any other. A problem then arises during the assimilation of new information from an ever expanding domain (the web), where another expert has expressed the same individual but using a different name.

As of late 2008 OWL 2.0 was submitted as a member submission to the W3c†, motivated by user experiences and feedback ‡. OWL 2.0 is now based on the more expressive description logic $\mathcal{SROIQ}$. Of note, OWL 2.0 contains convenience axioms for expressing disjointness between sets of classes *owl:disjointUnion*, thus bringing a more developer intuitive form of domain closure to OWL.

*see http://www.w3.org/TR/owl-ref/#distinctMembers-def

†see `http://www.w3.org/TR/owl2-semantics/`

‡See OWL: Experiences and Directions workshops: `http://www.webont.org/owled/`

4.2.2 Extending OWL with Epistemic Operators

Extending OWL with auto-epistemic non-monotonic constructs facilitates both open world and local closed world reasoning, enabling default rules and consistency checking integrity constraints [100]. Such an extension is based on auto-epistemic description logics (ADL) (themselves an extension of the base DL $\mathcal{ALC}$) which are proper extensions of OWL, adding both a K operator representing known knowledge and an A operator representing assumed knowledge (where the A operator is similar to the use of negation as failure in a logic program, and hence can be used to represent default assumptions). Take for example the geospatial integrity constraint formalised using the auto-epistemic operator K.

$$LargeBodyOfWater \sqsubseteq \mathbf{K}Lake \vee \mathbf{K}Ocean \tag{4.1}$$

Then, a LargeBodyOfWater must either be a lake or an ocean, otherwise the ontology is invalid - an ontological integrity constraint.

Further to this work, the description logic ALC was extended but with only the K operator ($\mathcal{ALCK}$) in [147], still allowing the representation of integrity constraints. Omitting the A operator makes the language less complex, but prevents the language from being able to capture default rules. Further, they note that auto-epistemic operators are difficult to serialize into standard OWL/XML syntax. Therefore the K operator was later added to their own KRSS format. Neither of these extensions assume the unique name assumption.

4.2.3 Logic Programming and Rules

As defined in Chapter 3, logical rules, typically captured in logic programs, have been studied for a number of decades in the area of artificial intelligence. The use of logic programs over description logics are proving a popular method to overcome the integrity limitations of OWL, thanks to their closed world and unique name assumptions [46, 154]. Furthermore, logic programs can represent complex property compositions rules, and have existing, mature and scalable logic programming engines that deal well with large instance bases [154].

In the next section we motivate the use of logic programs as an integrity checking ontology paradigm for the semantic web, followed in section 4.6 by a survey of where the head role (or predicate) is different from any of the body roles (or predicates) - in OWL this is not possible.

Interestingly, OWL 2.0 adds a restricted complex property inclusion axiom * that can capture a limited form of inference rule 4.2, see axioms 4.3 and 4.4.

$$\forall x, y, c \quad R(x,y) \wedge S(y,c) \rightarrow S(x,c) \tag{4.3}$$

or

$$\forall x, y, c \quad R(x,y) \wedge S(y,c) \rightarrow R(x,c) \tag{4.4}$$

However, such an axiom only permits the conclusion of a property used in the body of the composition, guaranteeing decidability when combined with other constructs of OWL 2.0. Hence, even with this extension, OWL will not be able to capture complex property compositions of the form show in 4.2, and is still not expressive enough to capture the spatial compositional inferences from composition tables show in chapter 2.

4.3.1 Procedural Attachments

Procedural attachments are a means to integrate a certain level of procedural code into a declarative, logical programming environment. Procedural code can be used to express complex criteria. For example in the geospatial domain, a procedural attachment could be used to compute the Euclidean distance between a pair of coordinates and return the result back to the logic program. Procedural attachments make the following two expressive contributions to logic: 1) they support the computation of property values, and 2) they support comparison operations on properties.

Unfortunately procedural attachments can be difficult to define formally, hence a logic program with procedural attachment can lead to a complicated semantic treatment. A Situated Logic Program [103] tries to provide a formal understanding and clean semantic treatment to such programs, by not allowing side-effecting procedural attachments that can alter the knowledge base outside the logic of the program e.g. the use of a remove function.

The use of procedural attachments to compare property values between indi-

*OWL 2.0 will also include; reflexive, irreflexive and anti-symetric property constructs viduals was recognized as a feature that should be included directly into the specification of OWL. OWL 2.0 therefore has proposed extended datatype support, for example, constraints relating individual values A < B. However, this is limited to comparison of concrete datatype properties, arbitrary procedural attachments are not allowed.

4.3.2 Integrity Constraints

Most logic programs do not inherently support rules as integrity constraints [154] (which are classically headless rules). However an integrity rule can be constructed in a logic program by adding an integrity predicate in the head of the rule [154], which can then be queried for (or is entailed) if the body of the rule evaluates to true, for example:

$$\forall x, y...z \quad error(x) \leftarrow B(x)_1 \wedge ... \wedge B(z)_n \quad n > 1 \tag{4.5}$$

That is, if the body of the integrity rule holds true, the integrity violation predicate $error(x)$ is inferred. By assuming a first order system (variables can be used within rule predicates as is common in all rule systems), $error(x)$ can capture information about the error, as opposed to just the fact that an error has been inferred.

4.4 OWL for the Spatial Domain

OWL's non-unique name and open world assumptions are also not suitable for capturing spatial constraints, as outlined for the general case in section 4.3. Consequently the use of OWL in determining the integrity of geographic ontologies is limited. Further to this, OWL does not support concrete spatial datatypes [113]. Even if a representation of spatial datatypes are possible using a suitable OWL model, as attempted in [1], OWL lacks a spatial indexing function (clearly so do all description logics, they are purely logical) crucial to speed up spatial selection over large numbers of geographic features. Furthermore, geographic knowledge bases consist of possibly millions of geographic features, tableaux based reasoners for description logics are known not to scale as well as logic programming reasoning engines in reasoning over large instance bases [46].

Finally, as stated in section 4.3.1, OWL, or OWL 2.0, do not support newly defined procedural operations. However such functions are useful for the geospatial domain, where certain spatial operators form a crucial part of any GIS, and have been added the proposed semantic web rule language SWRL in GeoSWSRL*. The OpenGIS consortium† defined a number of spatial operators to determine topological relationships between geographical features e.g. the distance between two geographical features etc. for use within spatial databases. These should also be used in any complete geographical knowledge representation and management environment.

4.5 Conclusion of OWL's Integrity Maintenance Capabilities

The open world and non-unique name assumptions of classical first order logic and OWL, although theoretically compatible with a dynamically evolving environment such as the web, are arguably not beneficial to the end user of geographic ontologies. Most an end-users would assume the information contained in a geographic ontology is consistent as a basis for further analysis and reasoning.

Using OWL axioms to specify disjointness between individuals is too verbose, and still does not offer a default assumption that individuals are unique, instead relying on the ontology author to continually assert this knowledge between every distinct concept, property and individual. This is still the case for OWL 2.0, even though the *owl:disjointUnion* goes some way to improving this.

Auto-epistemic extensions are not trivially implemented inside a practical framework. Indeed new algorithms need to be designed to cope with the extension of OWL with auto-epistemic operators. Some tableaux based reasoning algorithms for auto-epistemic description logics do exist, but they are based on expressively restricted description logic $\mathcal{ALC}$ [51] (Only the K operator extension within the Pellet engine for reasoning has been shown in [147]). Neither of these approaches can be used to enforce checking of qualified cardinality constraints. Therefore we still can not restrict an individual to have a certain number of properties - there is

*http://projects.semwebcentral.org/projects/geoswrl/
†www.opengeospatial.org/

still no way to represent the fact that a polygon must have no less than 3 vertices.

Fundamentally for this work, OWL, or OWL 2.0, can not represent complex property composition axioms which are a necessity if qualitative spatial reasoning compositional inferences are to be represent. Adding rules within a typical logic programming environment, overcomes some of the representational limitations of OWL for representing qualitative spatial reasoning compositional inferences. Furthermore, the typical implementation of the rule based paradigm (logic programming engines) assumes both the closed world and open world assumption, can include arbitrary procedural attachments, and scales better than description logic reasoners to large instance bases. As a result we argue in favour of combining a description logic with a logic program as the bases of a new geographic ontology paradigm.

In the next section we survey existing approaches to the combination of description logics and rules while maintaining first order semantics, as well as the integration of description logics and syntactically equivalent logic programs, while assuming a logic programming semantics.

4.6 The Integration of Rules and Ontologies

Description logics, including OWL, are proven languages for modelling concepts and terminological structures [184]. However, as already discussed, OWL is limited in its ability to represent property compositions, perform integrity checking over individuals, and can not easily express the unique requirements of spatial information. Adding rules to OWL is a step toward the representation of property composition. In addition, assuming standard logic program semantics allows integrity checking. Adding rules does not however allow the representation of spatial information, in particular concrete spatial datatypes (such a step is shown later in chapter 5).

Work on adding a rule layer to the semantic web technology stack was initiated by the W3c. Some argued against the introduction of more than one ontology language, reinforcing the idea that the integration of disparate paradigms should spawn a new pragmatic language [154]. Nevertheless, the rule layer has received significant attention from the semantic web community. Figure 4.1 illustrates the rule (and logic program correspondent) and description logic fragment of FOL where both are treated as separate systems.

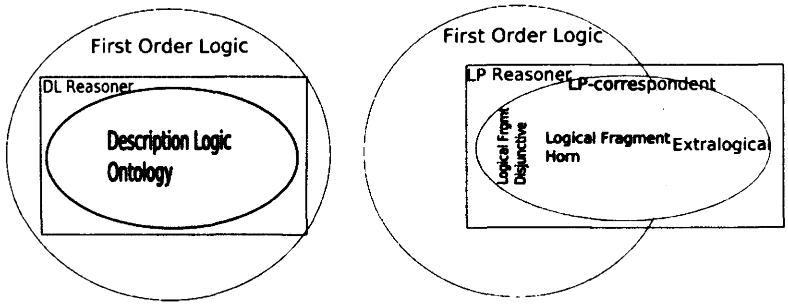


Figure 4.1: Rules and ontologies (Description Logics) treated as separate systems - no integration

As it stands there is much disagreement with how a rule layer should integrate with the ontology layer. Such conflict has given rise to contrasting approaches to their integration which broadly falls into two categories [5]:

(a) Hybrid approaches - A hybrid approach is characterised as a modular approach to the integration of rules and ontologies. Both rule and ontology components are kept separate, reasoning is performed separately and entailments by one component are treated as constraints to the other component. This approach is sometimes referred to as loose integration [62].

(b) Homogeneous approaches - A homogeneous approach is characterised by the complete translation of one language into the other. Certain approaches are based on the expressive union of the two languages or built around their common intersection. If a limited, decidable and tractable fragment of FOL is chosen as the rule language, their intersection also guarantees decidability and tractability. This is of obvious benefit, particularly when the fragment can be represented and reasoned with within existing mature reasoner implementations, for example Prolog. This approach is sometimes referred to as a tight integration [62] or translation [63, 46]

4.6.1 INTEGRATION ISSUES

Within the context of integrating rules, as used in logic programs, and ontologies, overcoming semantic differences between logic programs and classical FOL is a notable issue. Fundamentally their semantics differ in the following ways (see for example [225]).

(a) First order languages adhere to the open world semantics, while logic programs adhere to the close world semantics .

(b) Description logics do not assume unique names, whereas all individuals within the knowledge base (also referred to as the Herbrand Universe) of a logic program are assumed unique.

In addition, their integration brings forward the following issues.

(a) How to maintain decidability of the combined systems?

(b) How to maintain modularity of reasoning, or combine both systems into one logical language for use within a single (modified or unmodified) reasoner [104]?

(c) How to maintain tractability? Decidability does not imply tractability, moreover high worst case complexities (for example the NEXPTIME complexity of the description logic underpinning OWL) also do not imply tractability. End users of semantic web technologies will demand pragmatic reasoning procedures with at most a polynomial time complexity [122].

Each approach tackles these issues in different ways. In the sections to follow we survey existing techniques for the integration of description logics and logic programs or less formally, ontologies and rules.

4.6.2 HYBRID APPROACH

A hybrid approach (loose integration) is both modular and layered in that the syntax, semantics and reasoning distinction between ontology (Description Logic or DL) and relational (rule or Logic Programming) component is maintained [225]. The ontology component is some description logic variant i.e. $\mathcal{ALC}$ and richer, the rule component is typically some identified flavour of Datalog (for example Datalog or Datalog$^{\vee}$), see Figure 4.2.

A complete hybrid knowledge base K is represented by the pair $K = <\Sigma, \Pi>$, where Σ represents the ontological (structural) component, and Π represents the

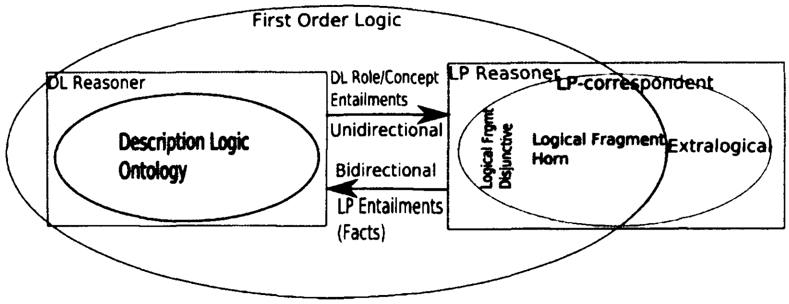


Figure 4.2: Hybrid integration of rules and ontologies

relational (rule) component. Π contains both rule and ontology predicates, where a strict separation is maintained between these predicates. Typically a rule r within the rule component Π has the form.

$$H \leftarrow B_1 \wedge ... \wedge B_n : O_1 \wedge O_m \quad n, m \geq 1 \tag{4.6}$$

Where, H and B are both rule predicates (head and body predicates respectively) and O represents an ontology predicate. Each O_i is a constraint of the form: $C(a)$ or $R(a, b)$, where C is a concept expression and R is a relational expression, both from the structural component.

The ontology predicates act as constraints that the interpretation of the rule must obey. Interaction between rule and ontology reasoners takes place through a a safe interface [62]. The flow of information between each component is either unidirectional or bidirectional. For a unidirectional approach, reasoning is performed over the ontology using an ontology reasoning engine (DL reasoner), entailments from the reasoner ω ($\Sigma \models \omega$) are fed, as a starting point, into the rule reasoner. Rules are interpreted such that they must satisfy the ontology predicates $p \in \omega$. If they don't the conclusion of the rule is not inferred. A bidirectional approach allows the flow of information from the rule component back to the ontology component. Iterative reasoning is then performed on both components until no more inferences can be drawn - a global fixed point. Bidirectional interaction is desirable as it allows a synergy of reasoning, and such synergy typically yields a larger set of inferences.

The integration of relational and structural components was first pioneered in 1998 in the AL-log system [50]. AL-log combines the foundational description logic

$\mathcal{ALC}$ [229] (a decidable DL) with Horn (plain) Datalog. The structural component allows the definition of an $\mathcal{ALC}$-knowledge base Σ that is a pair $\Sigma =< T, A >$, where T is the intentional component, a set of inclusions, and where A is the extensional component (Tbox) and a set of assertions (Abox). The structural component supports the Datalog rule component by providing a background theory or constraints. Therefore a complete AL-log knowledge base K is defined as the pair $K =< \Sigma, \Pi >$, where Π represents the Datalog subsystem. More formally, as again is common with a hybrid approach, for each hybrid rule r that is ground in the domain O_Σ (the set of individuals in the grounded ontology, so $r \in ground(P, O_\Sigma)$), if there is a constraint $C(a)$ in the rule r such that the interpretation I of the DL component does not satisfy $C(a)$, then r is eliminated. Otherwise all constraints are eliminated from r, that is they are satisfied and so are ignored.

To maintain decidability, AL-log implements and indeed defines the safety condition. That is, each first order variable that appears in the head of the rule must also occur in the body of the same rule. This ensures that the set of all facts derivable from the Datalog program Π is finite.

AL-log is limited with respect to the interaction between structural and relational components. Firstly, they use a constrained Datalog flavour with one of the least expressive description logics $\mathcal{ALC}$. Secondly, the set of Datalog predicate symbols appearing in Π is disjoint form the set of concepts and roles appearing in Σ. Therefore the interaction is unidirectional and intentional or derived relations from the Datalog subsystem can not be used to define terminological structures. Moreover constraints can only take the form of an $\mathcal{ALC}$ class or concept and not role constraint - where roles or spatial relations are an important part of geographic domains. AL-log employs a hybrid reasoner for query answering over both structural and relational component. The reasoner is based on the resolution principle of inference.

By considering new advancements in disjunctive reasoning and description logic engines, AL-log was extended by Rosati to include disjunctive Datalog and negation as failure ($Datalog^{\vee\neg}$) as well as the use of role constraints[224]. This approach is also unidirectional between structural and relational component.

At a similar time to the development of AL-Log, Levy & Rousset (1998) were developing their system CARIN [164]. CARIN integrates plain Datalog (Horn rules) with the description logic $\mathcal{ALCNR}$, a more expressive DL than used by AL-log. Significantly, CARIN allows both concept and role constructs from $\mathcal{ALCNR}$ to be used within rule antecedents as constraints. Full unrestricted CARIN is undecidable if either of the following description logic constructors are used (a source of undecidability in many later works): $\forall R.C$, $\geq nR$ or the terminology contains cycles in predicate definitions (recursive definitions). Decidability of CARIN can be attained by either syntactically restricting the structural component to remove these constructs, by allowing only acyclic concept definitions or by employing role-safety. Role-safety is of particular importance as it is used in a number of subsequent works. A rule is role-safe if at least one variable that appears in a role constraints also appears in an ordinary rule predicate. This serves to finitely bound the variable. CARIN-MARC is the the identified sub-language of CARIN which syntactically omits each of the constructors shown above. CARIN-MARC has a sound and complete inference procedure and the complexity of reasoning is co-NP complete.

Certain systems allow the bidirectional flow of information from structural and relational components. Allowing ontology predicates in the head of a rule is a way of achieving this. Eiter et al. developed Description Logic Programs in 2004, which caters for a bidirectional flow of information between structural and rule component [63]. Description Logic Programs is an approach to integrate rules and ontologies using answer set programming and the description logic $\mathcal{SHIF}$(D). Their approach defines both a DL knowledge base Σ, and a finite set of description logic rules (DL-rules) A - hence the combined knowledge base is $K =< \Sigma, A >$. DL-rules are akin to typical LP rules, but they may also contain queries to Σ within the body of a rule. The query is bidirectional in that inputs to Σ are allowed (enhancing DL inferences), as well as using the query as a rule constraint. For query answering their approach has a complexity of EXPTIME if the program is positive and stratified, or NEXPTIME if stratified negation as failure is used. The combination has both a strong and weak semantics. Weak semantics remove negation and DL-atoms from the rule, which then allows conformance to ordinary answer set semantics. The strong semantics removes those negated literals and DL-atoms which would give the logic program a non-monotonic characteristic, in so doing providing a smaller number of minimum models (possible interpretations of the program). In practice they utilize RACER for finding DL entailments and

DVL (a disjunctive Datalog reasoning system *) for rule entailments.

In 2005 Rosati dealt with the hybrid integration of rules and ontologies [225] while addressing some of the issues outlined in 4.6.1. In particular, he overcame the semantic difficulties in integrating closed world and open world reasoning, preserving decidability and maintaining reasoning with and without the unique name assumption. To achieve this he developed a safe hybrid knowledge base, combining *Datalog*$^{\neg\vee}$, with any function free subset of FOL*. Again as is common with a hybrid approach, structural predicates are allowed as constraints within the antecedent of rules. A bidirectional flow of information is catered for as structural components can appear in the head of a rule, hence rules act in reverse as constraints to the structural component. Rule predicates however are not allowed within the concept definitions in the structural component. Safe interaction between relational and structural components is assumed (as defined by CARIN and AL-Log). Rule safety is also assumed, where each variable occurring in a rule R must occur in a positive rule predicate in the body of R. In order to successfully mix open world and closed world assumptions, the structural component is interpreted with classical open world FOL interpretation I. Stable models for the *Datalog*$^{\neg\vee}$ component are then computed using the FOL interpretation I as a base, and where stable models are computed assuming a closed world. A rectification algorithm is used on the *Datalog*$^{\neg\vee}$ component to handle the effects of the non-unique names semantics of the structural component on the relational component. Essentially this algorithm generates equalities between variables and constants in the *Datalog*$^{\neg\vee}$ component, simulating the unique name assumption within a framework of assumed non-unique names. As a result *Datalog*$^{\neg\vee}$ can be interpreted with the standard Datalog semantics, and the explicit equality allows interpretation of the structural component within full FOL in an open domain, such that, the interpretation does not break the equality of variables and constants generated from the rectification algorithm. Checking satisfiability of a safe hybrid knowledge base is NEXPTIME^{NP}-hard if disjunctive Datalog is used, or if using non-disjunctive Datalog$^{\neg}$, checking satisfiability is NEXPTIME-complete.

In 2006 Boris Motik et al. developed a hybrid logic MKNF (Minimal Knowledge and Negation as Failure), which integrates a logic program with OWL-DL

*`http://www.dbai.tuwien.ac.at/proj/dlv/`

*Allowing arbitrary function symbols in FOL makes the language undecidable

[184] . This approach is of significance because of its use of an auto-epistemic operator K which can locally close parts of the domain. Auto-epistemic reasoning has been added to overcome shortcomings of OWL, specifically to handle integrity constraints - see section 4.2.2. A MKNF knowledge base K contains a DL knowledge base Σ and a rule component Π (hence again $K= <\Sigma,\Pi>$). Σ can be any DL, whereas Π is a disjunctive rule with both negation as failure and the auto-epistemic operator K. Predicates in Π can be DL-atoms which query Σ. However, this approach is unidirectional, information does not flow through the query to the DL to enhance DL entailments. MKNF is decidable under the DL-safety assumption. Semantically, both DL and rules are mapped to a set of FOL formulae. To overcome the unique names assumption of LP's and the non-unique names assumptions of DLs, their integration assumes the standard names assumption. That is, two individuals are equal only if there is explicit evidence to say so. Interestingly as the authors note, this does not change standard OWL inferences. A three valued, well-founded semantics for MKNF knowledge bases has been developed in [151].

4.6.3 Homogeneous approach

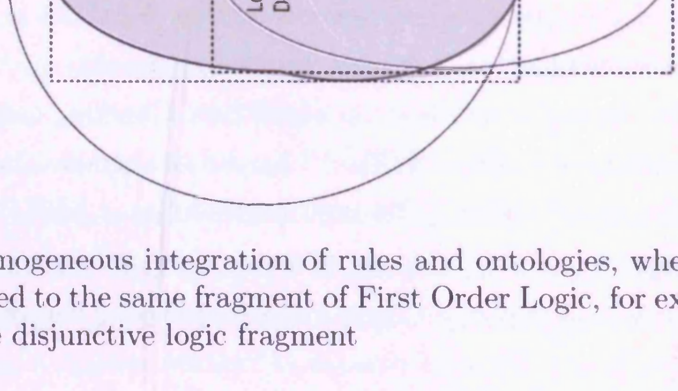


Figure 4.3: Homogeneous integration of rules and ontologies, where both components are mapped to the same fragment of First Order Logic, for example the Horn fragment or the disjunctive logic fragment

Homogeneous approaches (translation / tight integration) revolve around the complete integration of both DL component (ontology language) Σ and rule language Π into a singular language L. Within L no distinction is made between ontology and rule predicates, both languages are now syntactically and semantically identical and can be interpreted under the same reasoning umbrella.

All works based on this approach employ a mapping (typically a recursive mapping) from one language to the other. Mappings from the rule language to ontology language exist, however more common are mappings from the ontology language to the rule language, enabling the use of existing rule engines for reasoning tasks (query answering etc.). The output of the mapping is either too completely combine both languages (expressive union), or to embed one language into the other (intersection). Combining the two languages leads to theoretical and computational problems (as described later in this section). Intersecting the two languages into their common fragment can help to retain decidability and tractability if, for example, their common fragment corresponds to a decidable subset of FOL e.g. the Horn subset. On the flip side, the common fragment may be too restricted for expressive knowledge representation tasks. Figure 4.6.3 illustrates possible homogeneous integration approaches.

One of the first works in this area was by K.Van Belleghem et al. in 1997 [11]. They map $\mathcal{ALCN}$ (although a mapping to a slightly more expressive DL is possible) to open logic programs. Importantly, an open logic program can deal with undefined individuals. This strongly corresponds to the open world assumption of the DL and can therefore deal with incomplete knowledge - a desirable feature when combining the two knowledge paradigms. A recursive mapping between DL and an open logic program is defined. Semantically open logic programs are dealt with by a completion semantics. An open domain is possible by allowing for non-Herbrand interpretations [48].

Description Logic Programs (DLP), with similar name but not to be confused with the previously described Eiter et al.'s Description Logic Programs, first published in 2003 [104], and later revised for the description logic $\mathcal{SHOIN}$(D) underpinning OWL-DL for submission as part of the WonderWeb project in 2004, is a very pragmatic approach to the combination of description logics and rules as motivated by the design and deployment of semantic web services. A DLP is formed from the expressive intersection of OWL-DL and Horn Datalog. The authors first define Description Horn Logic (DHL) as a purely logical ontological language that represents the logical intersection of FOL and definite Horn (def-Horn). A def-Horn knowledge base can be constructed from a DL knowledge base by applying a recursive mapping function $\mathcal{T}$, which takes a DL axiom of the form $C \sqsubseteq D$ (where C is a body class and D is a head class) and maps it into an def-Horn rule of the form $A \leftarrow B$. The mapping preserves meaning between the original ontology and logic program. Additionally the mapping is bidirectional (referred to as DLP-*Fusion*). A Description Logic Program is then a definite equality free logic program (def-LP) syntactically equivalent to a def-Horn knowledge base. Interestingly the def-LP semantics are mildly weaker than the corresponding def-Horn. This is because every def-LP conclusion is a fact, whereas a def-Horn conclusion may be only a partial resolution of the rule (entails another rule), this is known as f-weakening.

Over the past few years a number of extensions to core DLP have emerged. By considering disjunctive logic programs a larger fragment of DL can be mapped into the combined language L [186]. A program for converting OWL to a disjunctive DLP fragment has been developed by the Koan2 Project [187]. In [103] DLP have been implemented in a system called SweetOnto which translates from a subset of OWL to Horn RuleML.

After standardising OWL as the de facto semantic web ontology language, the W3c began work in 2004 on standardising a rule language to augment the knowledge representation abilities of OWL, namely the Semantic Web Rule Language SWRL [129]. SWRL combines (expressive union) decidable unary/binary Horn clauses (Datalog) with the description logic underlying both OWL-DL and the slightly more restricted OWL-Lite. Different from all other homogeneous approaches, SWRL translates Horn clauses syntactically and semantically into the same model theoretic framework as the description logic underpinning OWL. Both rules and ontologies can then be captured by this combined syntax.

The expressive union of the OWL-DL and Horn rules is however not without difficulty. By itself, a Horn rule (unary / binary Datalog) is decidable and the description logic underpinning OWL ($\mathcal{SHOIN}$(D)) is also decidable. However, if combined, the resultant language (extension of OWLs model-theoretic semantics) is no longer decidable. Adding rules to a DL simulates a more expressive DL with role value maps* (R construct from table 3.1), and any DL with role value maps

*A role-value map allows the definition of arbitrary classes from the composition of arbitrary has been proven to be undecidable [230].

Another way of looking at this problem is by reference to the decidability of a DL and the decidability of a LP as shown in [185]. For a DL to remain decidable it has at least a model where the individuals and their properties form a *tree* like structure (referred to as the *tree model property*). Finding a *tree-shaped* model shows the DL knowledge base is satisfiable, and searching for such a model is possible (terminates) in most DL. Rules in a LP on the other hand do not require a *tree-model* but remain decidable as they are restricted to only universal quantification and a lack of negation. Existential quantification in the DL can lead to a possibly infinite number of anonymous individuals being inferred. In an LP where existential quantification is omitted, the reasoning procedure only needs to consider a finite number of individuals, and hence remains decidable. By combining the two without restriction means the LP would interact with the DL and the existence of anonymous individuals, this leads to undecidability.

In practice, SWRL and OWL reasoning is often performed over a subset of their expressive union to maintain decidability. Approaches involve the translation of SWRL into either forward or backward chaining logic programs [67], or by utilising extensions to existing tableaux based description logic reasoners. Full SWRL reasoning, via an iterative reasoning mechanism, has been developed in [93], however it is only tractable for the average case.

Decidability and tractability of SWRL can be obtained by employing DL-safe rules. Indeed in 2004, Motik et al. developed DL-safe rules [185] which combined the DL $\mathcal{SHIQ}$(D) with function free Horn rules (essentially Datalog). The resultant language remained decidable thanks to a restriction whereby each variable in a rule, also occurs in a non DL atom in the rule body - finitely grounding the variable from the Datalog knowledge base.

An alternative to SWRL based on F-Logic [149] has been co-developed by the W3c, named the Web Rule Language WRL [18]. WRL comes in three variants namely core, flight and full. Thanks to its F-Logic like heritage, WRL is a complete ontology language in its own right. WRL is not based on the the same first order semantics as SWRL. Indeed WRL can be used to capture integrity constraints much like OWL-Flight [46] - a subset denoted WRL-flight. The semantics of WRL are defined by two mapping functions. The first maps the WRL conceptual syntax

---

properties

to a more standard logical formalism. The second then maps these logical formulas either directly to Datalog$^{\neg}$ for the core variant, or the Datalog fragment of F-logic (with inequality and locally stratified negation) for the flight and full variants, under the perfect model semantics and well-founded semantics respectively. WRL-full is not restricted in any way (no safety and includes function symbols) and therefore is undecidable. WRL-core is based on the decidable Horn fragment corresponding to a DLP.

Also, in 2004, Heymans et al. defined Conceptual Logic Programs (CLP) [118], a unifying framework for combining and reasoning with rules and ontologies in infinite open and closed (when required) domains. Their work assumes the unique names assumption. CLP extends disjunctive answer set programming to consider open domains. An open domain is useful for handling unnamed individuals, where the program is grounded by a superset of present constants . Their approach can be used to simulate within a disjunctive logic program the notably expressive description logic $\mathcal{ALCHOQ}(\sqcup, \sqcap)$, which is of a similar family of language to $\mathcal{SHOIQ}(\mathrm{D})$ or OWL-DL. However, it does not include inverted roles, data types and role transitivity. Role transitivity and inverted roles are of particular importance as they are required to capture certain aspects of spatial calculi e.g. containment relations. As the authors note, in general, reasoning with open domains is undecidable. To overcome this they syntactically restrict the logic program so as each rule satisfies both the tree model and more general forest model properties.

Conceptual logic programs were extended (extended conceptual logic program or ECLP) in 2005 to include disjunctive rules [119]. ECLP develops upon the work of CLP by allowing rules that break the tree model property. Such rules however maintain the forest model property, as any additional rules must be grounded and not contain variables, and are thus finitely bounded. Such an extension is referred to as ground disjunctive logic programs. Adding only ground rules maintains decidability. More formally an ECLP is a program $Q \cup R$ where Q is a CLP program and R is a finite ground disjunctive logic program. An ECLP has a worst case time complexity of 3-NEXPTIME.

In 2006, motivated by the use of existing mature reasoner engines, Krtzsch et al. studied which fragments (as large as possible) of OWL can semantically be translated into other paradigms [154]. The principle objective of the work is to encode as large a fragment of OWL as possible into the Horn fragment of FOL.

Clearly any resulting Horn logic program is decidable and very tractable. The approach is based on KAON2 OWL*, which transforms OWL to clausal form using a five stage process. The output produces a disjunctive Datalog (Datalog$^{\vee}$) program. The transformation can convert a large fragment of OWL, namely the Horn-$\mathcal{SHIQ}$ fragment into the Horn fragment. The transformation algorithms complexity is exponential in the size of the input. Moreover to maintain sound and complete reasoning, the addition of certain new axioms needs reapplication of the entire transformation algorithm on the whole knowledge base. The authors note that any resulting recursive rules can be dealt with by ensuring the LP engine employs SLG-resolution with tabling, over the more common SLD-resolution as used in typical Prolog implementations. This is also important in reasoning over spatial calculi, as quite a few compositional inferences are recursive.

In 2008 Krotzsch et al. developed Description Logic Rules as a rule based ontology paradigm which can represent a fragment of the DL $\mathcal{SROIQ}$ underpinning OWL 2.0 [155]. It was noted that $\mathcal{SROIQ}$ is highly intractable, and that to regain tractability a sub fragment of $\mathcal{SROIQ}$ must be considered. Two sub fragments are identified, $\mathcal{EL}^{++}$ which is a tractable description logic, and Description Logic Programs (DLP), as described previously in this section.

4.6.4 Ontologies and Rules - Safety Restrictions

As shown previously in this section, the integration of description logics and logic programs (ontologies and rules) without restriction is generally undecidable, as first discovered in the seminal work of KL-ONE [21]. Even the integration of a moderately expressive DL and Horn rules has proven to be undecidable [164]. Therefore the literature identifies several safety conditions which, if adhered to, can help regain the decidability of integration.

CARIN introduced role safety for hybrid approaches. Role safety specifies that at least one variable in each DL role predicate must also occur in a body rule predicate. In the following example, the variable X in the DL role predicate r appears in the body rule predicate a.

$$H(t_1, ..., t_n) \leftarrow a(X, Y) : r(X, Z)$$

*kaon2.semanticweb.org/

Related to this is the idea of DL-safe rules, also just referred to as DL-safety. DL-safety constrains variables in the head of the rule such that they must appear in a non DL body rule predicate. This restricts the scope of the rules to those known individuals that occur in the finite Herbrand base of the transformed program [63], this ensures that the identity of all objects are known [185, 154], and that the set of all facts derivable is finite . DL+log refers to this as weak safety [226]. This is because by finitely bounding the set of individuals in this way, it becomes difficult to deal with existentially introduced individuals, as a infinite amount of these individual could be inferred.

Datalog, or indeed all FOL with function symbols is undecidable. The proof of which follows from reduction of query answering in a logic program to Hilberts Tenth Problem and from the undecidability of the diophantine equations, see for example [44] for a detailed discussion.

4.7 Summary and Conclusions

4.7.1 Overview Of Existing Approaches

We overview in Table 4.1 the representational capabilities and reasoning complexities of the important hybrid and homogeneous approaches to integrating ontologies with rules discussed in the previous sections. As is common e.g. in [44], complexity results are shown for a set of entailed ground atoms A, which are inferred from a set of explicit Datalog facts denoted D_{in}, and a set of Datalog rules P. More formally $D_{in} \cup P \models A$. Complexity is then measured on the following three properties of Datalog programs:

- Data complexity - where P is fixed, and D_{in} and A are variable. Measuring how the change in explicit facts effects the complexity of the program.
- Program complexity - where D_{in} is fixed, and P and A are variable. Measuring how the change in the Datalog program (rules) effects the complexity of the program.
- Combined complexity - where D_{in}, P and A are all variable. Measuring how the change in any part of the system effects the complexity of the program. This can be generalised to the main complexity measure, P $\models$ A

Unless otherwise stated complexity results given in table 4.1 represents results for the combined computational complexity. Complexity results for many classes of logic programs can be either seen in their respective publications, or are shown in [44, 25] (in particular for plain Horn logic programs).

4.7.2 Conclusion

Early hybrid approaches to the combination of description logics and rules were limited in the expressiveness of the DL used, for example $\mathcal{ALC}$ in AL-Log. Moreover, only permitting a class (unary predicates) constraint in a rule prevents the possibility of using roles (properties of binary predicates) from the structural component. Geographic ontologies are particularly rich in role assertions between individuals e.g. qualitative spatial relations. Therefore the omission of relational constraints from rules severely limits the capabilities of the resultant language to represent spatial integrity constraints.

CARIN is a promising hybrid approach that includes both role (binary) and class (unary) constraints. With this capability, the rule language is able to express interpretation constraints over relations and property values from the ontology component. However, as is the case with a number of the early hybrid approaches, CARIN only allows a unidirectional flow of information from structural to rule component. Unidirectional approaches will not trigger new entailments from the structural component based on the outcomes of the rule component, consequently not all possible facts are inferred.

Bidirectional approaches are beneficial when a reasoning synergy is desired. The combined reasoning potential of both rule and structural components is greater than the union of both considered in isolation. Newer hybrid approaches combine expressive DLs with expressive LPs e.g. based on a disjunctive LP and an OWL-DL equivalent description logic. These have, in part, stemmed from the advancements of efficient DL reasoning engines (for example FACT and RACER) and disjunctive rule reasoners e.g. DLV* or SModels[191] etc. Thomas Eiter at al.'s Description Logic Programs is a very promising work that incorporates a bidirectional flow of information, an expressive description logic ($\mathcal{SHID}$(D)) and a rule language that contains negation as failure. However, his language has a less

*http://www.dbai.tuwien.ac.at/proj/dlv/

tractable computational complexity of EXPTIME or NEXPTIME if negation is considered.

From an integrity checking point of view, MKNF is very promising as it allows the local closure of certain predicates using the auto-epistemic operator K, however MKNF is only unidirectional.

Homogeneous approaches are attractive in the sense that they can offer a full reasoning synergy between rule and structural component. Furthermore, if a logic programming approach is adopted, they also make the closed world and unique names assumption suitable for integrity checking tasks, and can be extended with procedural attachments. Homogeneous approaches are also very prominent in practical rule implementations, for example SweetJess and SweetProlog as part of the SweetRules project*. Indeed even the implementation of SWRL within the popular ontology editor Protege has been realised in JESS, a forward firing Rete based production system [193].

Although homogeneous approaches that use disjunctive logic programs can represent inference patterns suitable to represent spatial reasoning rules of the form: $B_1 \wedge B_2 \wedge ... \wedge B_n ... \rightarrow H_1 \vee ... \vee H_m$, we argue against their use for the following two practical reasons. Firstly a disjunctive logic program would need to be evaluated using a bottom up, backward chaining reasoner. A forward chaining implementation strategy is not suitable for disjunctive reasoning as forward chaining approaches are not suitable for concluding partial or incomplete information (as represented by head disjunction in rules). This is because the knowledge base would need to explicitly represent a number of possible model of the worlds, which is expensive with respect to space requirements, and is not catered for in existing forward chaining reasoning engines such as Rete. However, integrity maintenance tasks are more suited to forward chaining approaches that continually monitor the knowledge base for inconsistencies, as opposed to a query answering scenario whereby violated integrity constraints would need to be queried for. Lastly, disjunctive logic programs typically have worse, less tractable computational complexities than LPs that do not contain head disjunction e.g. NEXPTIME [44]. We do not believe that large scale practical applications would benefit from logic programs with large search spaces and thus high computational complexities.

WRL-Flight is based on the Datalog subset of F-Logic which is suitable for

*`http://sweetrules.projects.semwebcentral.org/`

integrity checking tasks. However WRL-Flight uses a complicated perfect model semantics, requires stratification of negation, and is hence less tractable than Description Logic Programs.

Grosof et al's Description Logic Programs (DLP) maintains a solid base for the integration of OWL-DL with a Horn rule language. DLP are often though of as a restricted but yet practical core knowledge representation paradigm with a tractable polynomial data complexity [123, 154, 121]. In addition most existing ontologies do not contain constructs that are not within the Horn fragment of FOL, and hence most are representable in a DLP [266]. Even newer works such as Description Logic Rules use a DLP core to regain tractability of the DL used by OWL 2.0 [155]. Although the semantics of a DLP are still first order and assume an open world and non-unique name assumption [121], by translation into a LP the resultant language could fully exploit the closed world and unique name semantics of the LP (much like Datalog), suitable for integrity checking applications. Furthermore, LP engines are mature and efficient and scale well to large instance bases.

As a result, we argue in favour of using the syntactic Description Logic Programs fragment of OWL-DL as a core knowledge representation paradigm for geographic ontologies. The mapping into a LP allows us also to assume the semantics of a LP using standard closed world and unique name assumptions suitable for integrity checking applications, along with the use of extra-logical procedural attachments and arbitrary Horn rules. However, an LP still does not provide suitable support for spatial datatypes and spatial selection. In the next chapter, a survey of existing work to incorporate spatial information into description logics is investigated, and a new LP geo-ontology paradigm and spatial framework is proposed.

Table 4.1: Tabular view of the main existing approaches to the integration of ontologies and rules

Approach	Type	Year	Computational Complexity	Expressive Fragment
AL-log	Hybrid	1998	Certain sublanguages Polynomial, otherwise NEXPTIME [a]	$\mathcal{ALC}$ plus Horn with unary (class) constraints. Unidirectional flow, safety condition
MKNF	Hybrid	2006		$\mathcal{SHOIN}(\mathcal{D})$ (OWL-DL) plus Datalog$^{\vee,naf}$ and the autoepistemic operator K. Unidirectional flow, DL-safety
DLP (Eiter et al.)	Hybrid	2004	EXPTIME (positive and stratified), NEXPTIME (stratified negation as failure)	$\mathcal{SHIF}(\mathcal{D}) or \mathcal{SHOIN}(\mathcal{D})$ (OWL-DL) plus Horn. Bidirectional flow, structural components as queries rather than pure constraints
AL-Log (Rosati)	Hybrid	1999	$coNEXPTIME^{NP}$	$\mathcal{ALC}$ plus $Datalog^{\neg\vee naf}$ with binary and unary constraints. Unidirectonal flow, safety condition
CARIN	Hybrid	1998	CARIN-MARC co-NP-complete (unary constraints, recursive Horn), Role-safe Horn unary/binary constraints co-NP-complete [25]	$\mathcal{ALCNR}$ plus Horn with unary and binary constraints. Unidirectional flow, Role-safety
Safe-Hybrid KB	Hybrid	2005	Disjunctive $NEXPTIME^{NP} - hard$ or NEXPTIME-complete	Any function free subset of FOL with $Datalog^{\neg\vee}$. Bidirectional flow as structural components in the head of a rule, rule components are permitted in structural definitions. Rule/Role safety from CARIN and AL-Log
SWRL	Homogeneous	2003	Undeciable	Full OWL-DL plus Horn Rules
WRL	Homogeneous	2003	Undecidable for WRL-full with function symbols, otherwise all Polynomial in data complexity and EXPTIME in program complexity	Common FOL fragment for WRL-core, $Datalog^{\neq strat-naf}$ for WRL-flight, Horn plus naf under WFS WRL-Full. Therefore any subset of OWL-DL that will translate into these fragments
CLP[b]	Homogeneous	2004	Reducible to finite answer set programming, hence $\sum_2^p$-complete in data size and $NEXPTIME^{NP}$-complete in program size [61]	$\mathcal{ALCHOQ}(\sqcup,\sqcap)$
ECLP [c]	Homogeneous	2005	time complexity of 3-NEXPTIME	CLP extended with ground disjunctive logic programs
Krtzsch et al.	Homogeneous	2005	Polynomial time complexity using SLG-resoultion	Horn-SHIQ using five step algorithm. Employs SLG-resoultion
DLP (Grosaf et al.)	Homogeneous	2003	Polynomial time complexity (two free variable fragment)	LP intersection of OWL-DL and definite Horn

[a] [illegible]

Chapter 5

General Framework For Combining Geospatial Rules and Ontologies

In the previous chapter we concluded on the best logical base representation paradigm for geographic ontologies on the Semantic Web. In this chapter we explore specific methods to combine description logics with spatial logics. Spatially enabling semantic web technologies empowers reasoners with a form of spatioterminological reasoning as well as spatial reasoning over individual instances of geographic phenomena. The related work, along with the identified limitations of OWL in chapter 4, lead to the description of a new, hybrid geo-ontology framework that combines geo-ontologies with rules, for the primary purpose of maintaining the integrity of geographic information on the Semantic Web.

5.1 Combining Description Logics and Spatial Logics - Existing Approaches

A Description Logic (DL) is a powerful representational tool for describing real world concepts, their attributes and relationships. The key reasoning mechanisms of any DL are checking concept satisfiability and inferring subsumption hierarchies. On the terminological level a DL reasoner will infer concept hierarchies based on concept subsumption. On the level of asserted knowledge (instance level) each individuals type is inferred if not already explicit. Over the past 10 years general concept inclusion has been enriched with spatial reasoning to enable spatioterminological concept hierarchies - both general *is-a* hierarchies along with geographic spatial inclusion hierarchies. Spatial reasoning can then be exploited on two levels, the concept level and the instance level. On the concept level (TBox), spatioterminological reasoning resolves to both concept classification using spatial and terminological inclusion. On the instance level (ABox), spatial reasoning can be intermixed with the axioms in the DL both to derive new information, and to check the consistency of the spatial scene described by such information.

Existing research into combining spatial reasoning with DLs, including how to deal with both quantitative concrete locational information (geometry) and symbolic qualitative spatial information, can be categorized into the following two categories [246]; homogeneous approaches which extend existing description logics with spatial logics, and hybrid approaches which combine existing description logic systems with an existing GIS inside a developed framework*.

5.1.1 HOMOGENEOUS COMBINATION

A Homogeneous combination extends existing description logics (DLs) with spatial concrete domains and qualitative spatial reasoning algorithms. Such an approach provides inherent spatial reasoning for the deduction of, and satisfiability checking of, asserted spatial information.

Spatial reasoning was identified as a key component of DL inference as far back as 2000, when Michael Wessels began work in overcoming the limitations of existing DL based languages to handle composition based role inclusion axioms (complex property composition). This in turn opened the possibility of capturing spatial composition inference patterns, in this case he choose the RCC-8 composition table [270]. A compositional inference role inclusion axiom has the form:

$$\mathrm{S} \circ \mathrm{T} \sqsubseteq R_1 \sqcup \ldots \sqcup R_n$$

Such an axiom can then capture property composition inference patterns, which have the following form in first order logic.

*Note the general approach to integrate ontologies with spatial logics is very similar to the general integration of rules and ontologies

$$\forall x, y, z : \mathrm{S(x,y)} \wedge \mathrm{T(y,z)} \rightarrow (R_1(x,z) \vee ... \vee R_n(x,z)).$$

This is a general extension and can be seen as a way of overcoming a major limitation of a DL. The resulting DL was named $\mathcal{ALC}_{RA\theta}$, and a tableaux calculus for deciding the concept satisfiability problem for this language is presented in [273]. However, as the authors note, such an expressive DL ($\mathcal{ALC}_{RA\theta}$) is undecidable.

Wessels furthered this work in 2002 and developed a DL $\mathcal{ALCI}_{rcc}$ which only includes role axioms as derived from the RCC-8 composition table [271]. The main aim of the work was to investigate concept satisfiability using spatial reasoning. An axiomatization of the RCC-8 composition table is then applied to check the satisfiability (RSAT) of individuals with respect to role box axioms. He also added role disjointness to the language in order to capture the exclusive nature of spatial roles (the eight base spatial relationships are JEPD). Moreover, he notes the need for the DL to handle inverse roles to capture converse ($\smile$) relational inferences*, which completes the RCC-8 network (e.g. the inverse of $NTPP$ or inside is $NTPP^{-1}$ or contains).

Haarslev et al. proposed an extension to $\mathcal{ALCRRP}$(D) DLs to include a concrete spatial domain, thus making spatioterminological reasoning (at the concept level) a reality [113]. Class level (concept level) reasoning can then be performed to infer a subsumption hierarchy based on both concept (terminological) and spatial inclusion. They highlight the need for both spatial and terminological (conceptual) knowledge reasoning intermixed as requirement for a GIS based on DLs.

$\mathcal{ALCRRP}$(D) contains new modelling constructs natively supporting topological spatial relationships (roles) and a concrete spatial domain, allowing complete descriptions of spatial objects and their spatial extents. However, polygons are the only concrete spatial data type that is currently supported. Other point sets from $\mathbb{R}^2$ and $\mathbb{R}^3$ are not supported. Topological relations are in the form of the RCC-8 relations. A three step external algorithm then determines the satisfiability of the concrete domain. The algorithm proceeds by computing topological relations by computational geometry, adding them to the DL to form a constraints network, and then checking the consistency of this network (a verification step) using a classical RCC RSAT algorithm (so for example those described in chapter 2).

In [146], Yarden Katz argues that OWL-DL is an adequate language for the rep-

*Hence the $\mathcal{I}$ in the DL $\mathcal{ALCI}$

resentation of RCC-8 if extended with reflexive roles (reflexive properties). Such work was motivated by the representation of RCC-8 in the modal logic S4, which shares a close correspondence to OWL-DL with reflexive roles. However the representation is unintuitive, that is, regions are expressed as non empty regular closed sets and the RCC-8 relations are sets of concept axioms. Importantly, treating region instances (individuals) as classes and not as individuals limits their interaction with other individuals in the ontology. This is because OWL requires type separation between classes, properties and individuals [202]. Regions as sets of concept axioms does not allow them to be classified along with other individuals in the domain [110]. This leads to a weaker form of spatioterminological reasoning, more of a spatial plus terminological reasoning. Moreover the authors note that a potential pitfall of the approach lies in the inability to use existing reasoners to tractably deal with the proposed encoding.

5.1.2 Hybrid Combination

A hybrid approach is both pragmatic and more readily implementable. A hybrid approach combines the best technology from existing DL systems and existing GISs, in this way maintaining a separation between the semantic ontology store (DL component) from the geometric (locational GIS) store. A certain level of scalability should be preserved thanks to the reuse of highly optimized geo-processing and spatial indexing engines for geometric information.

As part of the Description Logics and Spatial Reasoning DFG grant, Michael Wessels developed an experimental deductive Geographic information System (GIS) [272]. A hybrid software framework was proposed combining three separate (existing) components. Each component of the framework is represented by *substrates.* These substrates offer uniform protocols and reasoning services. These components are: the extensional component (E), intentional component (I) and the query component (Q). The extensional component represents the spatial database or map substrate. The intentional component offers some level of reasoning service, namely the DL component (using a description logic to model an ontology which they term a spatio-thematic concept language). The query component is represented by a hybrid query language capable of information extraction from either the intentional component (with the added vocabulary provided by the reasoning services), or extensional component (geometric store), or both. The languages used for each component of the framework are not predefined, leaving the decision of exact implementation technologies up to the user*. That said, Wessels does present a concrete hybrid query language which can query all substrates transparently, and suggests the use of his previously developed $\mathcal{ALCI}_{RCC}$ DL which integrates support for spatial reasoning (deduction and satisfiability checking) within the intentional component (I).

During real world experimentation with a digital vector map from Hamberg, Wessels highlighted an important issue. When dealing with a dichotomy of spatial relationship representation, both qualitative and quantitative, spatial relations could be pre-computed from the geometry and the geometry could be completely discarded. Such an approach is however not practical as it requires a large number of role assertions in the Abox*. However, such a large number of asserted roles adversely effects the performance of a querying or reasoning engine. He goes on to note that a substantial amount of these roles represent disjointness between objects (as supported by Egenhofers work, where he discovered that 90% of topological relations in a GIS are disjoint relations [137]). He further adds that qualitative spatial reasoning can be used to deduce a percentage of these relationships. That is, explicit spatial relations (roles) in the Abox can be sufficiently edge reduced to only a subset of the total amount, and those left implicit can be derived when needed by spatial reasoning methods. He concludes by saying that on the fly geo-computation of relationships can be expensive but may in the worst case be needed, to reduce storage overheads in the Abox.

Importantly, Wessels did not attempt to add spatial datatypes and spatial index extensions the DL reasoner due to its complexity, and the already proven support for these functions in existing GIS technologies. Instead, he adds an RCC-8 substrate which supports consistency checking and entailment of RCC-8 relations using RCC-8 composition table axioms. He then uses RacerPro to query the entailment from the RCC-8 substrate. As usual, if a given instantiation of an RCC network satisfies the RCC composition table axioms, then the scene is consistent. He emphasizes that dealing with disjunctive base relations can lead to an exponential number of Abox instantiations.

*However the languages are constrained to be a subset of First Order Predicate Logic

*This is quadratic in the number of regions n

The KES-B Project at ESRIN produced a study into the use of semantic web technologies for both the water quality domain and maritime security applications [264]. The project supported two main types of services 1) information retrieval or search services and 2) production services using a workflow system. Their system provides complex geospatial queries relating geometries and attributes. For example, oil spill detection and ship location detection. A spatial reasoning engine is provided within the architecture to define queries between features which are related spatially (and non-spatially), and hence entailments from the spatial reasoning engine are used to enrich the base model. The spatial reasoning engine incorporates a search, fusion and report model. The queries can also involve fuzzy terms or relationships such as near and far. Each query is resolved by a Rete rule based expert system integrated with a GIS feature server. Fuzzy Jess [196] (based on a Fuzzy version of the Rete Algorithm [76]) provides the expressive power required to deal with the fuzzy relations expressed in a query. Of interest, they suggest Rete's scalability is in general acceptable for demanding large scale complex query processing. This is one of the only works that considers the use of Rete for spatial reasoning.

Enriching DLs with spatial understanding and reasoning can be provided by so called *e*-connection [43]. *e*-connection links both spatial and non-spatial knowledge in a way which can be exploited by a small extension to existing OWL-DL reasoning algorithms. For example, a region in the DL may represent a political division (a fiat object), whereas the same region expressed in the spatial knowledge base represents an area division of space - a footprint. The spatial aspect of the object can then be intermixed with the non-spatial aspect and reasoned with.

In 2007 the Swiss Federal Office for the environment for snow and landscape research facilitated spatioterminological reasoning for query answering, that involves both spatial and thematic query expansion [111, 110]. Such queries are common in discovering BIOTYPES that cover and overlap regions of space. For example, a user may request information on endangered butterflies in Birmensdorf and neighbouring villages. They note that two spatial relationships *in* and *neighbouring* are use to link (relate) the concepts of *Birmensdor* and *villages* together. Such queries will be only possible by combining both spatial and terminological reasoning. To achieve this, they developed their own approach which revolves around adding a new RCCBox to the already present TBox and ABox. The RCCBox uses a separate reasoning engine (distinct from the DL reasoner) to derive disconnected regions and to check spatial consistency. A hierarchy of OWL object properties are used to encode all RCC-8 topological spatial relationships, therefore the ABox maintains the spatial relationships from which the RCCBox draws upon during spatial reasoning. Certain computed relationships can be added back to the ABox for faster second time access, however not all spatial relationships are made explicit so as to save storage overheads.

5.1.3 CONCLUSION OF THE INTEGRATION OF DESCRIPTION LOGICS AND SPATIAL LOGICS

In conclusion, both homogeneous and hybrid approaches enable some form of spatioterminological reasoning to infer concept and spatial inclusion hierarchies and check the consistency of spatial instances.

A hybrid approach utilises established research into spatial selection (spatial indexes) and geometric processing in an existing GIS, with well defined semantic knowledge representation in a DL ontology. A hybrid approach is both practical and pragmatic, a view also supported by the UK's mapping agency the Ordnance Survey*. Where such a pragmatic combination of existing techniques could also see a better uptake from the GIS community. The use of both geometric (quantitative) information in the GIS with symbolic (qualitative) spatial information in the DL is a good practical approach for balancing the number of pre-computed explicit role assertions (relations) in the ontology, over the computation of those relations using the functions of the GIS. This is of particular use if the ontology is partially complete with respect to both qualitative and quantitative information. Qualitative relational information can then complement quantitative information, helping to make explicit all implicit spatial relations in the ontology.

However, hybrid approaches have some identified limitations. Wessels's suggests a spatially aware DL language that is undecidable and not compatible with existing reasoning engines. The Swiss Federal Office extend the typical separation of TBox and ABox to include an RCCBox, which is again non standard and requires an new reasoning engine. Indeed, most use some form of spatial reasoner that is external to the main DL reasoning engine, although newer works of Wessels and Moller are

*See `http://owl-workshop.man.ac.uk/acceptedPosition/submission_8.pdf` beginning to add the necessary spatial extensions to the Racer reasoning engine [274].

Homogeneous approaches allow for a total integration of concrete datatypes and a DL, hence providing complete and natural representation of spatial objects without recourse to additional external technologies. The possibility of using a concrete domain satisfiability algorithm for deciding consistency of RCC networks, represented as asserted individuals and roles (relationships) in the Abox, is a useful step toward the maintenance of spatial information in an geographic ontology.

In this area, Haarslev's work is limited in that it only supports the representation of polygons which, although acceptable for reasoning using Region Connection Calculus (which only supports reasoning over regions or polygons), a complete spatial ontology needs to support a general set of spatial data types, such as regions, points, and lines. Moreover, functions to deal with a concrete spatial datatype domain would need to be added into existing DL reasoning engines.

Wessel's DL $\mathcal{ALC}_{RA\theta}$ is undecidable for the full RCC-8. In addition, his subsequent DL, $\mathcal{ALCI}_{rcc}$ requires the use of inverse roles, which are known to cause efficiency issues with DL reasoners that support it [132]. Katz's work to encode RCC directly in OWL-DL is limited in that it does not allow for the representation of other types of classes in the ontology - all classes are treated as regions.

Finally, homogeneous approaches would need better integration with existing mainstream reasoning engines in order to facilitate better uptake within the GIS and Semantic Web community.

All aforementioned approaches (hybrid and homogeneous) do not consider the use of a rule layer. Rules can be a useful tool for application in the geospatial domain [29]. A number of recent contributions do consider the use of rules for application over geospatial ontologies. The Defence Science and Technology Organisation (DSTO) for Australia has developed a prototype semantic information demonstration environment (SIDE) [194]. SIDE incorporates a DL and associated DL reasoner, with rules for topological spatial reasoning. Logical entailments of the DL reasoner are fed to the rule language which generates additional geospatial inferences, hence employing a hybrid approach to the integration of DLs and ontologies. However, the DL reasoner and rule reasoner are separated in this approach.

In the next section we outline the requirements we believe necessary to support geo-ontologies and then introduce a new framework which adds spatial reasoning to existing DLs through the integration of rules and ontologies within a hybrid framework which includes an existing ad-hoc GIS.

5.1.4 Desirable Characteristics of a Geo-ontology Paradigm and Framework

Building upon previous research, we now identify those features we propose necessary as part of a new geospatial rule and ontology maintenance paradigm for the semantic web. In overview, the management of geo-ontologies has two high level requirements:

- Representational Requirements - The ontology paradigm must be expressive enough to represent geographic ontology models.
- Manipulation Requirements - Manipulating geo-ontologies involves the reasoning, search, computation and retrieval of spatial properties and relationships.

With respect to the representational requirements, the language and framework must be able to support a geo-ontology model based on standardised geographic vocabularies and semantics. This should be supported by suitable spatial data types and relationships, and provide a scalable capacity for handling and searching over large geometric data stores using appropriate spatial indexes and geometric computation functions.

In addition to the representation of a geo-ontology, the language should support user-definable spatial and thematic integrity and deduction rules acting over both the individuals in the instance base (Abox), as well as individual geometries (where applicable). Integrity rules will help decide the consistency of the individuals in the geo-ontology using a mixture of qualitative and quantitative information. An integrity rule set also needs to employ constraints based on the application of qualitative spatial calculi. Further to this, deduction rules should be definable that enrich the raw information present in the geo-ontology, with new, inferred information. Another useful extension to the integrity checking feature of the language is to be able to represent default integrity rules and their exceptions, using a form of defeasible reasoning. To enable the expression of integrity rules, the language must assume semantics suitable for integrity related tasks e.g. employing both a closed world and unique name assumption.

With respect to the manipulation requirements two paradigms are possible: quantitative, using computational geometry procedures for structuring and search along with qualitative spatial reasoning techniques. Indeed, both paradigms are complementary and can be used together. Explicit computation and storage of all geometric relations - all relations between pairs of object - leads to an overinflated ABox with large storage overheads [271, 243]. Hence, it is desirable for the framework to be able to mix the use of computed relations with explicit, stored relations. The use of quantitative information brings with it more specific geometric manipulation requirements. That is, to handle geometry correctly, the language or framework must support basic geometric computational and spatial search / selection functions to manipulate the geometry associated with features in the geo-ontology. In order to maintain tractability of the language, the ontology component should be based on a tractable ontology paradigm, which is expressive enough to support the representational requirements outlined above.

Any complete framework should link the various components together seamlessly without requiring the user to have specialized knowledge of each component. Furthermore the framework should present the user with an interface for updating and editing both ontology, rule and geometric components. The user should see the entire system as a singular entity, any separation should only occur during low level storage and computation.

5.2 Geo-ontology Maintenance Framework

This section proposes a new hybrid geo-ontology maintenance framework for the development and management of geospatial ontologies on the Semantic web. The hybrid approach allows the framework to harness existing research for each individual component.

In order to meet the requirements set out in section 5.1.4, the framework is comprised of the integration of a structural ontological component, a relational rule component, along with a location store (a GIS or spatial database), as follows:

(a) Geo-ontology and Spatial Rule Paradigm.

(b) Location Storage System.

(c) Spatial Inference Engine.

(d) Visual Interface.

Concrete framework base: The framework is made up of the following concrete components. For the core geo-ontology and spatial rule paradigm, we choose the already identified Logic Programming (logic program or LP) equivalent of the highly tractable Description Logic Programs (DLP) subset of OWL-DL. A LP is then expressive enough to represent a geo-ontology and error ontology (described in section 5.2.1), both deduction and integrity rules, and is expressive enough to represent the Horn compositional inferences from the generalised RCC composition table in section 2.4.3.2. Furthermore, by assuming LP semantics the language makes a closed world and unique name assumption suitable for integrity checking tasks. A LP can then be run using existing, scalable reasoning engines suitable for reasoning over large instance bases. Lastly, an LP can use the following inference tasks to represent all DL inferences over that subset of their integration:

- Determine whether a ground atom A is entailed from the LP. That is when $LP_{HB} \models A$, where LP_{HB} is the minimal Herbrand model of the LP.

- For a non ground atom A (those which may contain variables), determine all the variable bindings that are ground entailments of the minimal Herbrand model of the LP .

However plain LP alone is not sufficient to capture all the representational requirements described in section 5.1.4. Hence, the base language will be augmented with a form of defeasible reasoning, for which we choose Courteous Logic Programs (CLP)[101], a mixed mode of reasoning to efficiently handle qualitative spatial reasoning, and extra procedural attachments to handle the connection between geo-ontology and location base. This leads to the development of a new language named The Semantic Web Spatial Rule Language (*SWSRL*). The Location Storage System is implemented using an Oracle 10g spatial database. The framework is illustrated in Figure 5.1.

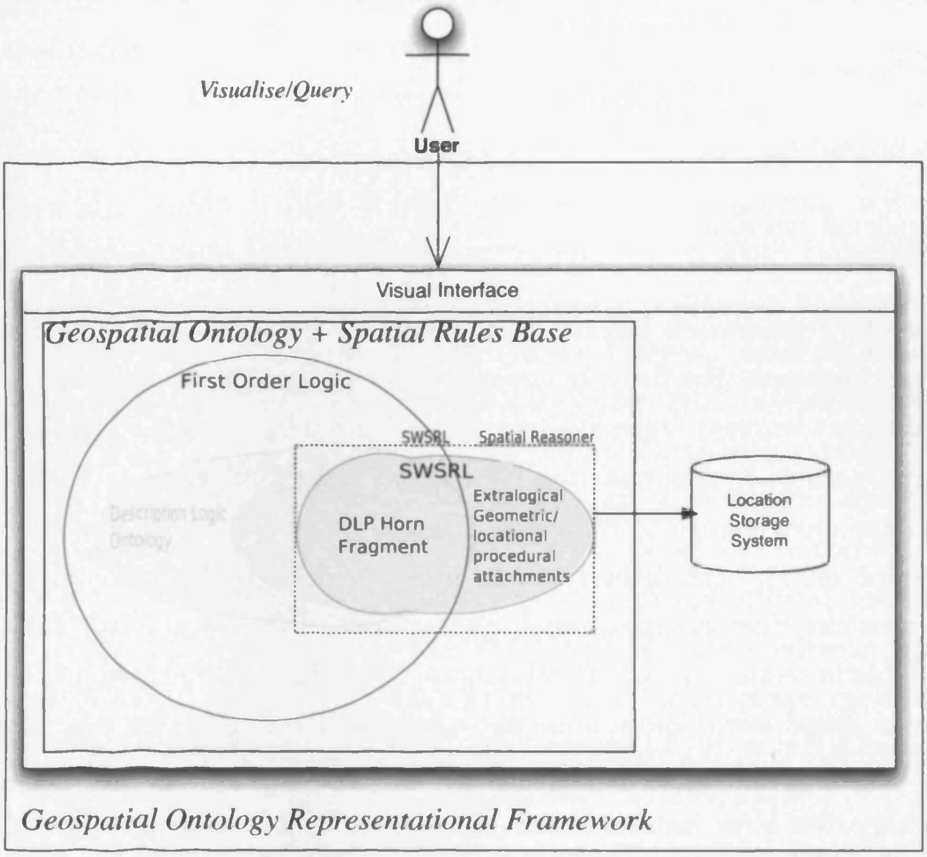


Figure 5.1: Illustration of the complete geo-ontology maintenance framework

The hybrid architecture is of similar vein to other hybrid spatial frameworks already identified in this chapter, but with the following key differences:

- The use of the Description Logic Programs fragment of OWL-DL as a base to represent the geo-ontology, and a logic program to represent accompanying rule sets.
- The use of both forward chaining and backward chaining reasoning modes to facilitate qualitative spatial reasoning and external access to the location base on the fly.
- The use of the rule component, executable in existing reasoning engines, to perform satisfiability checking of topological relations in the geo-ontology, using generalised RCC relations as shown later in chapter 7.

- The addition of error terminology to the geo-ontology to support the representation of errors derived from integrity rules.

In the sections to follow we describe in more detail the geo-ontology model (including the error ontology), the location storage system, and finally the visual interface. We leave the more involved discussion of the the new geo-ontology language *SWSRL* and it associated reasoning engine to later chapters.

5.2.1 Geo-ontology and Error Ontology Model

In this section we consider a typical geographic ontology model. The model is based on the Open Geospatial Consortium (OGC) guidelines for simple geographic features, see [267], and other models commonly used in existing geospatial ontology development e.g. [139, 244]. The terminology of the geo-ontology is relatively plain with regards to the number and type of constructs used. This reflects typical geographic ontology developments which, beyond the complex representation of geometry, are relatively sparse (parsimonious [142]) and fit to purpose. Of note, temporal aspects of the OGC's Reference Model are not used within the model. In the framework, the geometry is modelled and stored within the Location Storage System (LSS) described in Section 5.2.2. Furthermore, at this stage we aim to represent the geo-ontology and later described error ontology in OWL-DL. When we describe the new language *SWSRL* in chapter 6, we show how this ontology can then be mapped into *SWSRL*. This serves to highlight how existing OWL-DL ontologies could be mapped in *SWSRL* and the hybrid framework for later reasoning.

Geo-ontology model: The geo-ontology is shown in Figure 5.2. A geofeature, as a specialisation of a general feature defined in the ISO 19109 standards, is a representation of any geographic phenomenon that exists in space, e.g. a forest, a building or a road. As such, its location and boundary can be specified using a geometric entity of point, line or polygon. Also, as it is located in space, the relationships it exhibits with other geofeatures are of interest, e.g. it may be inside (topological), north of (directional) or near to (proximity) another feature.

The model assumes a predefined set of qualitative spatial relationship properties, including, topological, directional as well as relative proximity and size

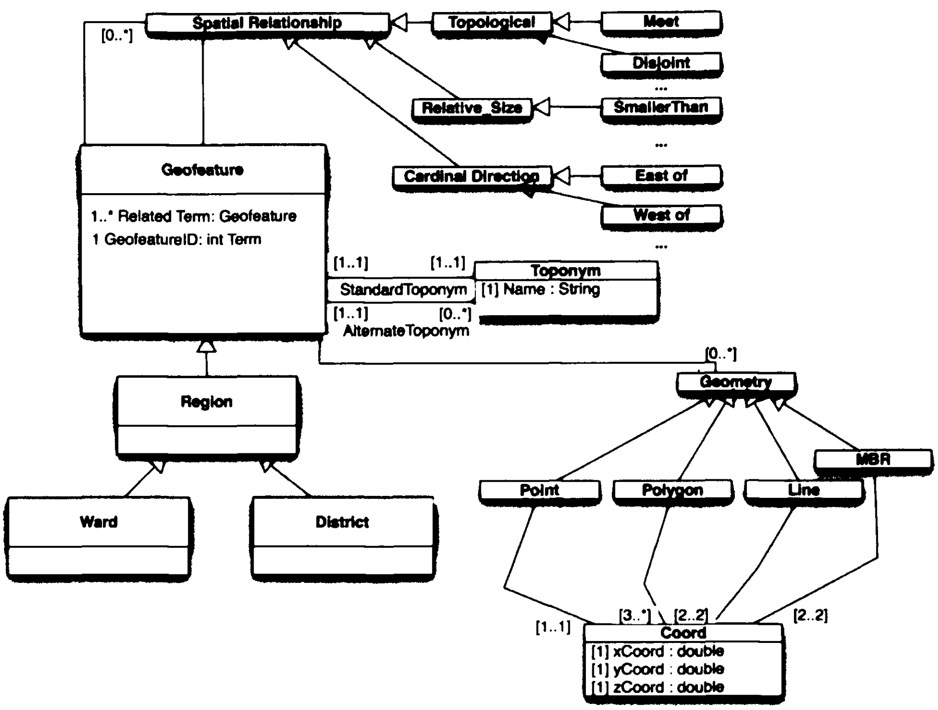


Figure 5.2: The developed OWL-DL geo-ontology model shown as a UML class diagram

relationships. As described in detail in Chapter 2, topological relations describe neighbourhood and incidence and are invariant under scale, rotation and translation. The semantics of the topological relationships have been formalised in [57] and alternatively in [214]. All topological relationships are defined as part of the OGC Filter Encoding Implementation Specification [268]. The topological relationships assumed are: Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects and Contains. These are mapped to and from generalised RCC relations for spatial reasoning as shown later in Chapter 7. Cardinal direction relations describe order in space. The 4 cardinal direction system is assumed here, these are: West, East, North and South. Size from one feature relative to another is also assumed, namely; largerThan, smallerThan and sameSizeAs. Some of these relationships between pairs of geofeatures may be stored explicitly, or can to be computed from the geometric representation using the Location Storage System.

Typically, an extended gazetteer model will specify Broader Term (BT) and Narrower Term (NT) properties [258]. Although these are not explicit in the geo-ontology, the structuring formed from the use of these predicates is implicit within the subsumption hierarchy. Traversing up the hierarchy explores the BT or more general geofeature types, and in reverse traversing down the hierarchy explores the NT or more specific geofeature types.

A Region concept, and any specialisation thereof, is of particular interest as we assume it represents the primitive level of the geo-ontology that conforms to a proper region in RCC. More formally in our context, it is a one piece object in $\mathbb{Z}^2$ with interior (which separates a proper region from an improper one) and exterior separated by a boundary.

5.2.1.1 Error Ontology Usage

Errors mined from the geo-ontology by integrity rules are also stored in the geo-ontology and conform to the error model shown in Figure 5.3. Building an ontology of errors is interesting as it opens up opportunities for reasoning over errors. Some of the possible uses of such an ontology are as follows.

(a) Methods can be developed to correlate statistical measures on most frequent error results.

(b) Provide insight to the types of integrity problems found that may lead to the development of more effective error management procedures.

(c) Comparing error ontologies derived from different geo-ontologies can be used to facilitate their integration.

As part of this thesis, the error ontology is used as an input to the error localisation methods discussed in chapter 8 to identify which relations are inconsistent.

The error ontology shown in figure 5.3 contains two error classes, one represents a positive *error* and the other represents its negation, *notError*, as used to represent the default integrity rules and their exceptions. This is part of the defeasible reasoning extension proposed as part of *SWSRL* and described later in chapter 6. Errors have a number of datatype properties and a link to individuals in the geo-ontology. Errors are instantiated in the error ontology on the conclusion of an integrity rule. More formally, the head of an integrity rule includes the predicate *error* or its negative counterpart *notError*, where both share the same syntax as error records in the model shown in Figure 5.3:

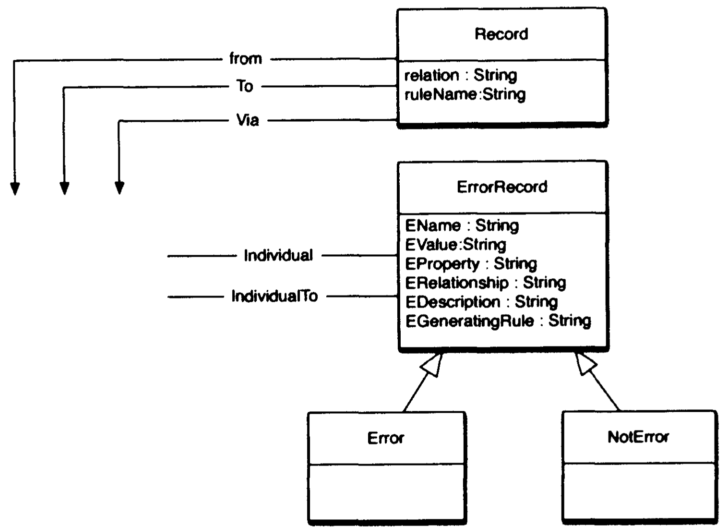


Figure 5.3: The OWL-DL error ontology model.

(not)error(name, individual, Relationship / Property, Value / IndividualTo, Description, Generating Rule)

Integrity rules in *SWSRL* can be used to capture both relative constraints representing the relationships between objects, or absolute constraints reflecting the properties of the object. If the error occurs from a binary relationship between two features, then the *Value* and *Property* properties can be ignored. If the error occurs from a binary relationship between a feature and a literal value (its datatype property), then the *IndividualTo* and *Relationship* properties can be ignored. Errors and their negations *notError* that share the same instantiation of terms are in logical conflict. The conflict is then dealt with using the prioritised defeasible reasoning extension to *SWSRL* discussed in Chapter 8. A further class denoted *Record* is defined which records the results of compositions as instantiated by error localisation rules as shown in Section 8.6.1.

5.2.2 The Locational Storage System

The hybrid representation mode of the geo-ontology architecture influences the design of the geo-ontology such that geometry is not modelled within OWL directly. Typically, spatial information consumes substantial amounts of memory e.g. to explicitly store feature geometries. An example of this was seen during experimentation of converting the Seamless Administrative Boundaries of Europe (SABE) data set into an OWL ontology using the Jena2 toolkit. The base ontology was represented without locational information, but with administrative area identifier, administrative unit name and the topological relationship contained-by. It consisted of 11 classes, 10959 individuals, 2 object properties and 3 datatype properties. During data querying using the Jena2 interface, 16 mb of Java Virutal Machine memory and 2.2 mb of persistent storage (for the output XML/RDF representation of the ontology) was consumed. When all administrative areas were attached with their associated locational information (in this case polygons), the memory footprint increased to 800mb and the size of the XML/RDF output on persistent storage increased to 100mb.

This, along with the other identified limitations of OWL, or indeed all Description Logics, in section 5.1 to store and process geometric information, means that geometric information will be stored external to the ontology component in the (locally or remotely accessible) Location Storage System (LSS). The LSS will then be responsible for the storage, manipulation and processing of locational (geometric) information.

It is useful to make the LSS totally transparent to the end user, such that modifications to the ontology will take place through a common interface, and geometric data is loaded into the LSS and ontological information is loaded into the geo-ontological component automatically. This transparent treatment is not handled within the developed system, and would be the subject of future research.

5.2.2.1 Locational Storage System Implementation

Oracle Spatial 10g is used as the LSS component of the framework. Oracle is an object-relational database that has a set of spatial schemas for the definition and representation of spatial objects. A table (locationBase) will be constructed whereby each row will represent an individual geofeature. Each geofeature will have a Uniform Resource Identifier (URI), which in his case represents an RDFID, as its primary key as type String, and a geometric description of the object of type MDSYS.SDO_GEOMETRY. The RDFID provides a unique reference between features in the geo-ontology and their corresponding locational information represented in the LSS. In this sense the RDFID is also a foreign key in that it refers to the same feature in the geo-ontology. Oracle 10g has numerous spatial operators that provide geoprocessing functions that can compute, for example, the area of a polygon and calculate the geodetic distance between two points.

The exact schema definition of the locationBase table is shown below.

```
CREATE TABLE locationBase (
    rdfID VARCHAR2(30) PRIMARY KEY,
    shape MDSYS.SDO_GEOMETRY
)
```

where the MDSYS_SDO_GEOMETRY object is part of the Oracle Spatial schema, and is defined by:

```
CREATE TYPE sdo_geometry AS OBJECT (
    SDO_GTYPE NUMBER,
    SDO_SRID NUMBER,
    SDO_POINT SDO_POINT_TYPE,
    SDO_ELEM_INFO MDSYS.SDO_ELEM_INFO_ARRAY,
    SDO_ORDINATES MDSYS.SDO_ORDINATE_ARRAY
);
```

The type of the geometry is represented by SDO_GTYPE. A spatial reference system, SDO_SRID, is required to map the coordinates of a feature to a particular coordinate space. A Geodetic (based on the shape of the earth) Coordinate system will be used by the LSS in *SWSRL* based on the Longitude / Latitude (WGS 84) standard - represented as by the identifier 8307 in Oracle 10g.

Various functions of the LSS will be made available through extra logical builtins in the developed rule language, and are hence shown in chapter 8.

Spatial Indexing: It is fundamental within the framework that LSS querying (spatial selection) performance will scale well to the inclusion of large amounts of vector geometry, hence it is important that a proper spatial indexing structure is used. As a result we employ an R-Tree index, where R-Trees are the default spatial index in Oracle, and on average perform better in the computation of distance and spatial relationship queries than the alternative QuadTrees [152]. . .

5.2.3 GEO-ONTOLOGY VISUAL INTERFACE

'The power of the unaided mind is highly overrated. Without external aids, memory, thought, and reasoning are all constrained.' [192]

A Graphical representation can help a user build a mental map of the ontology. Different categories of visualisation for logical Semantic Web ontologies have been studied, namely; SHRIMP views, Attribute explorers, Hyperbolic Tree views, Hierarchical tree views.

Protege's [107] OWL based visualisation deals with the representation of class or terminological structure in a tree-based node-link visualisation. Protege can also be used to view individuals in a simple tree structure. Some, such as OWLFCAView [136], are based on a property-oriented visualisation. It is argued that the property-orientated view is more flexible and scalable than a class-oriented view.

Spectacle's Cluster Map interface[72] emphasises the use of captured semantics to enable smarter visual organisation of class hierarchy and instance membership. Semantics affect the adjacency and distance between clusters. A cluster is a set of points that represent instances, the size of clusters and overlap between different classes can be seen, however no detailed information can be gleaned about individual instances on a smaller scale.

CropCircles [250] depicts an OWL ontology class hierarchy as a tree structure where special emphasis is placed on the tree's topological structure. CropCircles uses containment to represent the parent-child relationship. Other tree views use geometrical containment to depict the class subsumption hierarchy [248, 138]. However, such a depiction can become difficult to comprehend when large hierarchies are represented. A hyperbolic tree view [159] tries to overcome the inefficient organisation of a standard tree view by arranging the hierarchy on a hyperbolic plane.

All aforementioned visualisations are general and make no assumption about the domain or application of the ontology they represent. As a geo-ontology is used to capture the geographic domain, certain assumptions about the data it holds can be made, in particular with regard to the spatial aspect of the data. Thus we propose a visualisation to aid in the representation and maintenance of a geo-ontology, with specific attention to an instance-orientated view, where its organisation is based on the spatial attributes and relationships that exist between geofeatures of the ontology.

Interface Components In this thesis we propose a Geospatial Ontology Management Suite (GMS) which comprises three core views to the geo-ontology and rule sets, these are:

(a) The geo-ontology instance view - this view is an instance-oriented view of geographical features in an $SWSRL$ knowledge base.

(b) The rule authoring view - this view allows for the creation of spatial deduction and integrity rules from $SWSRL$'s set of predefined vocabulary constructs.

(c) The error tracing view - this view shows inconsistencies derived from violated integrity rules, and a trace of violated facts. The error tracing view can be used to aid the rectification of inconsistencies which can be amended directly in the geo-ontology view.

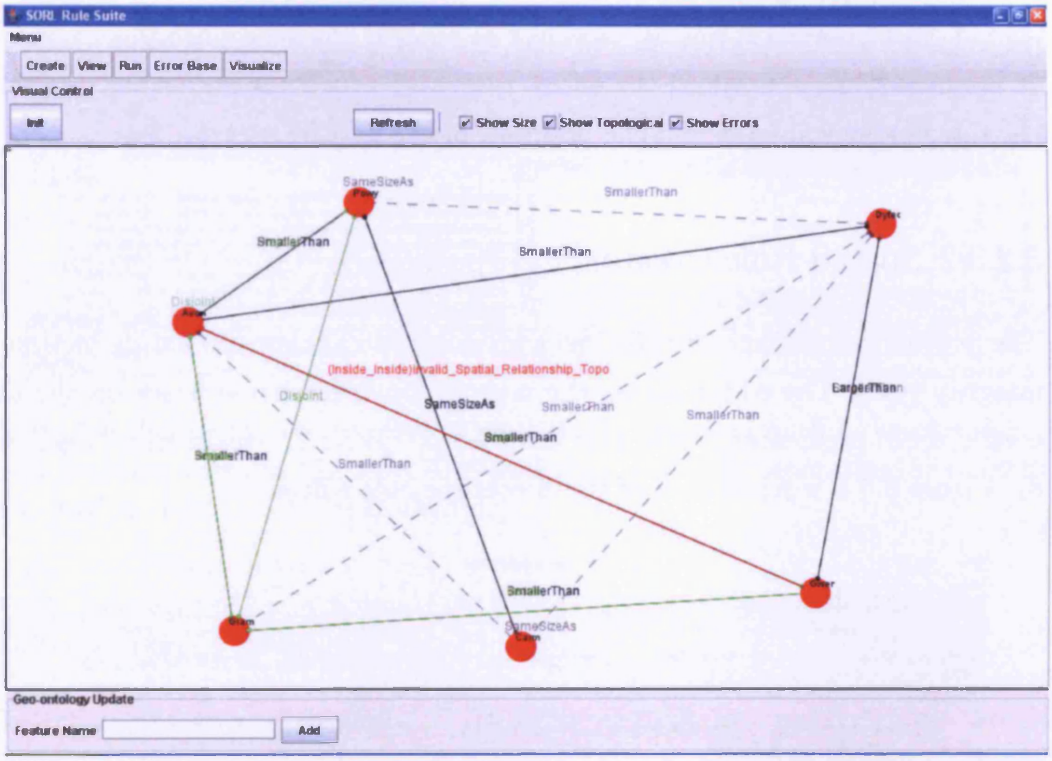
Each view is described in more detail in the sections to follow.

5.2.3.1 Geo-ontology Instance View

Individual instances of geofeatures are depicted by nodes along with their full URI (name). Edges between features are used to represent spatial relationships, as opposed to the typical parent-child (is-a) relationships common in most existing graph or tree visualisations. In particular an edge will represent either a topological, relative size or orientation relationship between two features. The edges are directed as not all spatial relationships are symmetric. Figure 5.4 is a screenshot of the geo-ontology viewer.

Edges and Errors: Edges represent spatial relations or relationship errors between geofeatures. Errors are highlighted by edges drawn in red. Implicit relationships between features as derived by deduction rules are highlighted by edges

Figure 5.4: Geo-ontology Instance Viewer. Showing a sample geo-ontology where geofeature's are connected by relativeSize relations (shown as a green edge), topological relations (shown as a black edge), and one error exists (shown as a red edge). Implicit relations are shown by a edges with a dashed line, explicit relations are shown by edges with continuous lines.



drawn with a dashed line segment. Explicit relationships, those found in the raw information, are highlighted by edges drawn with a continuous line segment.

[redacted]

relationship will be constructed in the raw ontology. The interface will then refresh to represent the new information. This method can also be used to overwrite any relationship that has been highlighted as an error. For example if an error has been found in the relationship *Inside* between the features A and B, the user can rectify this error by replacing the *Inside* relationship by the relationship that should exist between A and B. However of note, if any relationship is overwritten (a nonmonotonic operation) the entire reasoning process is performed again from the raw ontological facts and the addition of the new fact.

5.2.3.2 Spatial Rule Creator

The prototype interface also includes an editor to author spatial deduction and integrity rules. The editor allows the user to construct a rule based on the use of a valid set of predicates (predicates included in $SWSRL$'s vocabulary, see chapter 8). Figure 5.5 is a screenshot of the prototype rule editor.

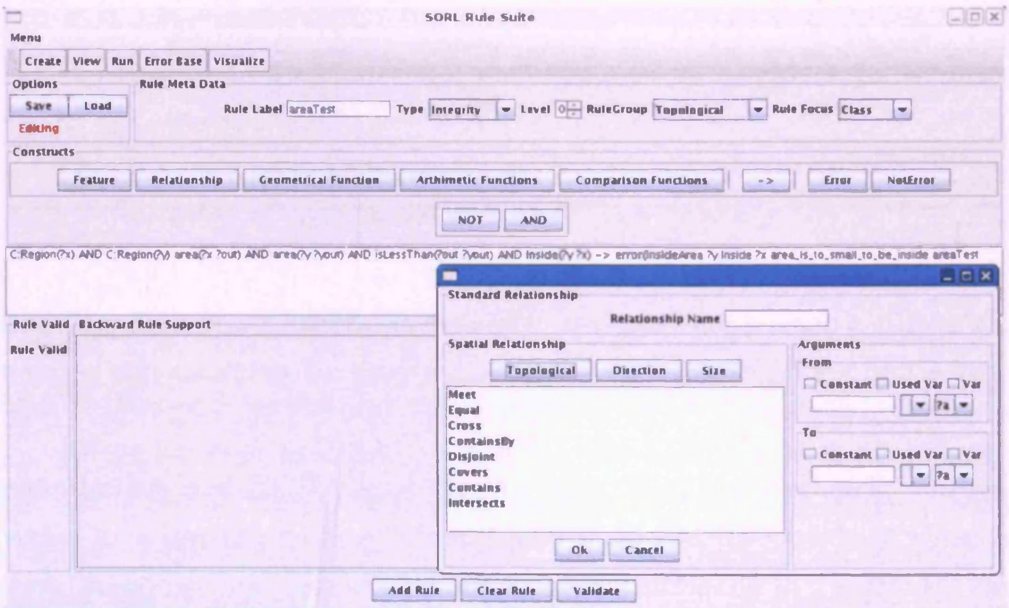


Figure 5.5: Screenshot of the Spatial Rule Editor Viewer

Rule sets can be visualised using the 'rule-tree' view, which is a hierarchical view of rulesets organised by rule metadata (see chapter 6 for a definition of the rule metadata used in $SWSRL$). Figure 5.6 shows the rule-tree view.

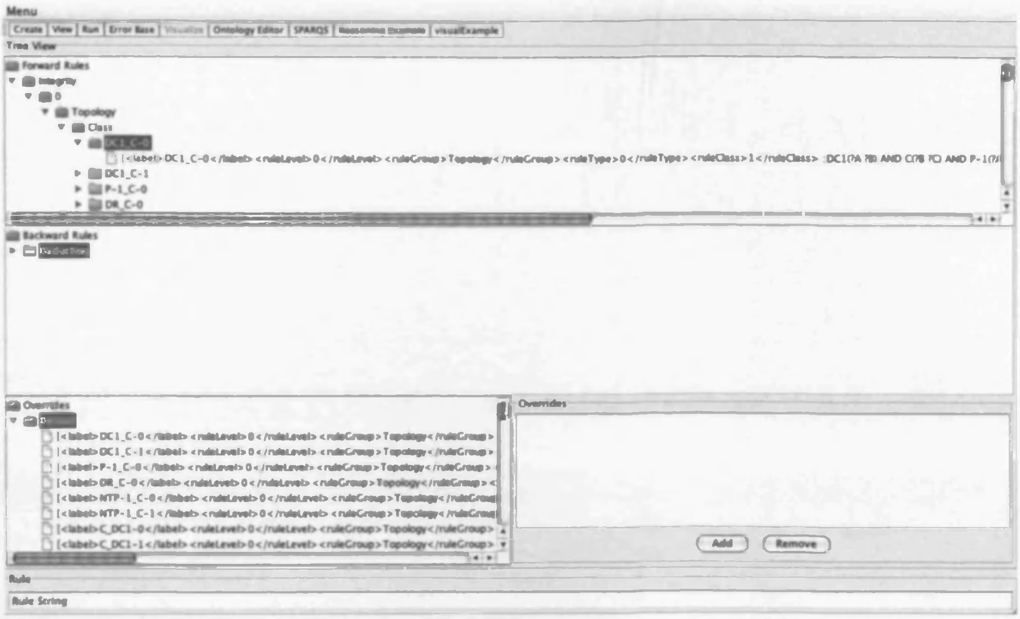


Figure 5.6: Screenshot of the Rule-Tree Viewer. Showing both forward and backward chaining rulesets (forward and backward chaining separation of rulesets is described in detail in the next chapter)

5.2.3.3 Error Tracing View

The error tracing view is used to expose the derivation logs of violated integrity constraints. In particular it is useful for locating the source of inconsistencies in spatial relations. The user can chose from a drop down list of found errors, and the system will return and print the trace of that error - an example of which is shown in Figure 5.7.

5.3 SUMMARY

In this chapter we proposed a framework for maintaining geo-ontologies using a combination of visual interface, ontology, rule and external locational storage systems. An overview of the ontology model, visual interface and location storage system was given in detail. The discussion of *SWSRL* is concluded in Chapters 6, 7 and 8. Chapter 6 describes the new geo-ontology language paradigm. Chapter 7 is a definition of the topological rulesets used in *SWSRL*. Finally Chapter 8 gives a detailed description of *SWSRLs* spatial reasoning engine.

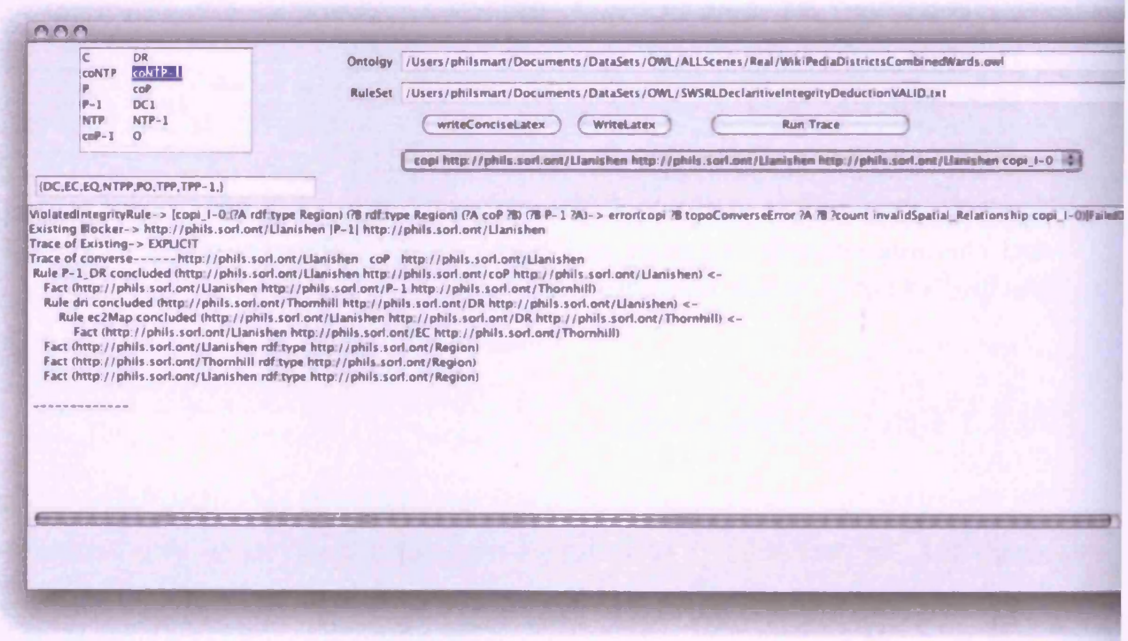


Figure 5.7: Screenshot of the Error Tracing Viewer. Showing an example error trace produced from a violated integrity rule.

Chapter 6

The Semantic Web Spatial Rule Language (SWSRL)

6.1 Introduction

In the previous chapter a hybrid framework was proposed that combined the Logic Program (LP) equivalent of a Description Logic Program (DLP) with a Location Management System (LSS). Suggestions to enrich the core LP ontology for integrity checking in the spatial domain where outlined and a new Semantic Web geo-ontology paradigm was proposed. In this section the new rule language, the Semantic Web Spatial Rule Language (*SWSRL*), is described in detail. The language is designed specifically for geospatial applications and as such has a number of unique features which better suite the distinct characteristics of geospatial data. The rule language, its purpose features, syntax and semantics are described, along with a conversion of the geo-ontology developed in OWL-DL in the previous chapter to an *SWSRL* geo-ontology.

6.2 *SWSRL* Language Overview

The Semantic Web Rule Language (*SWSRL*) is, at its core, a DLP (a Horn Logic Program) ontology extended with deduction and integrity Horn rules that assumes an LP semantics. As suggested in the previous chapter, the following additional features to a plain LP are proposed as part of *SWSRL*:

- Extra-logical procedural attachments for geospatial information processing.
- Defeasible reasoning to manage default integrity rules and their exceptions.
- Spatially oriented rule syntax, for standardising with the syntax and semantics of existing geo-ontology vocabularies.
- Efficient handling of qualitative and quantitative spatial information using interleaved forward and backward reasoning.
- General integrity rules (user defined) as well as those that govern space, utilising previous results in qualitative spatial reasoning and the construction of composition tables for spatial calculi.

Consequently, *SWSRL* is then a combined language capable of encapsulating (representing and reasoning with) the following components:

- An ontology component for representing ontological axioms and instances of the proposed geospatial ontology model as defined in section 5.2.1. The ontology component is henceforth denoted *SWSRL***O**, the Semantic Web Spatial Rule Language **O**ntology.
- Geospatial deduction and integrity rules. The need to efficiently mix qualitative and quantitative reasoning is satisfied by an interleaved extension to the language, where *SWSRL* supports mixing forward and backward rules, described later in section 6.5.2. Consequently, the set of all forward inference rules and backward inference rules are henceforth denoted $GeoR_{fd}$ and $GeoR_{bk}$ respectively.
  - Forward rules are further subdivided into forward deduction and forward integrity rules, where integrity rules have a different syntax than deduction rules. Hence two sets are defined, forward deduction denoted $GeoR_{fdD}$, and forward integrity denoted $GeoR_{fdI}$.
- Geospatial rule metadata, denoted Geo_{meta} for describing geospatial deduction and integrity rules.

As *SWSRL* is designed as a web ontology language it adheres to the following:

- The language uses labels for rules and rule bases, which aids the import / export of rule bases between semantic web applications.
- The language uses URIs to denote the logical vocabulary and knowledge base subsets: predicates, functions, rules and rulebases.

Treating *SWSRL* as the whole logical language with which to fit each component (or subsets of *SWSRL*) above, the logical breakdown of *SWSRL* is then illustrated in Figure 6.1.

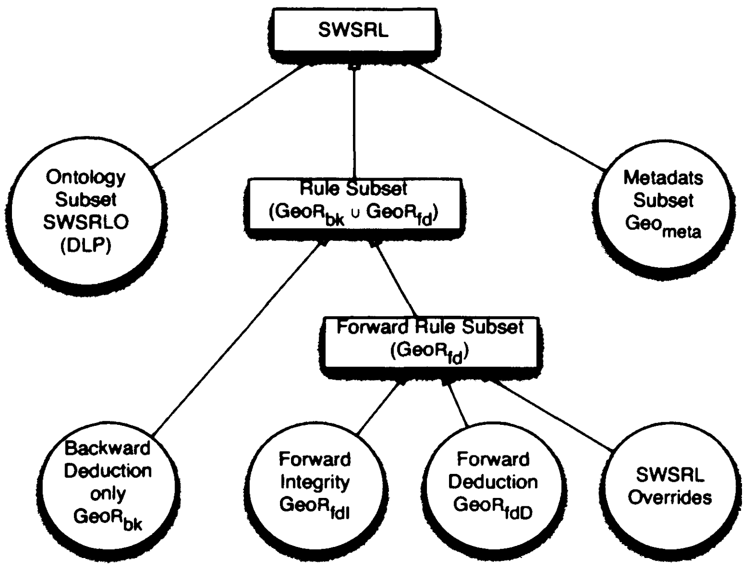


Figure 6.1: Illustration of the various logical fragments of *SWSRL*

6.3 SWSRL Abstract Syntax

A more formal description of *SWSRL* is now given through its abstract syntax. That is, *SWSRL* can be described as the set:

$$SWSRL \equiv SWSRLO \cup GeoR_{fd} \cup GeoR_{bk} \cup Geo_{meta} \cup CLP_{mutex} \tag{6.1}$$

$$GeoR_{fd} = GeoR_{fdI} \cup GeoR_{fdD} \tag{6.2}$$

$$SWSRL_{overrides} \subseteq (GeoR_{fd}) \text{ in which Overrides appears} \tag{6.3}$$

Rules $r \in GeoR_{bk}$ and $r \in GeoR_{fdD}$ are Horn rules (the LP equivalent of

Horn clauses) but extended with procedural attachments (built-ins). Each rule $r \in GeoR_{fdI}$ is a Horn rule again extended with procedural attachments along with a simplified form of Courteous Logic for defeasible reasoning - see section 8.4.1. Functionally, an integrity rule differs from a deduction rule in that the head of an integrity rule only permits a positive or negative error predicate, error(t_1,...,t_n) or its negation ¬error(t_1,...,t_n) respectively, and no other. All rules have a predefined set of spatial metadata tags and a spatially-oriented concrete syntax - see sections 6.6.3 and 6.6.2 respectively.

$SWSRL_{overrides}$ is the subset of $SWSRL$ where an overrides predicate occurs. Override predicates are used to specify priority between integrity rules that conflict, and as such form part of the simplified Courteous Logic extension described in section 6.5.1.

The interleaved interaction between forward and backward system is optional, henceforth both $GeoR_{fd}$ and $GeoR_{bk}$ will be treated separate syntactically. Their interaction is described in section 6.5.2, and a combined semantics is given in section 6.7.2.

Abstract Syntax Restrictions: As identified in Chapter 3, to maintain decidability and tractability of the language, $SWSRL$ adheres to the following expressive restrictions:

- The Datalog restriction, whereby any free variable in the head of the rule must first appear in the body of the rule. In addition only function symbols of arity = 0 (constants) are allowed, as function symbols of arity > 0 cause undecidability of the language.

- $GeoR_{bk}$ is restricted to a definite logic program (Horn rules only), and hence omits any form of negation.

- $GeoR_{fdI}$ permits strong negation but restricts its appearance to head literals only which is then dealt with using the Courteous Logic extension. $GeoR_{fdI}$ omits negation as failure.

- $GeoR_{fdD}$ omits any form of negation, but allows disjunctive head literals which can be transformed using the Lloyd Topor transform into multiple rules with definite head.

When describing the abstract syntax of each of the individual subsets of $SWSRL$ throughout the reminder of this section, the following common notation will be assumed. Let ϕ be a finite set of predicate symbols in $SWSRL$, let Ω be a finite set of builtin predicate symbols in $SWSRL$. Let φ be a finite set of constant symbols and let X be a finite set of variables. An atom is an expression of the form s(t_1,...,t_n) or b(t_1,...,t_m), where s is a predicate symbol $\in \phi$ and has an arity $0 < n \leq 2$, and b is a predicate symbol $\in \Omega$ an has an arity m $\geq$ 0 and t_1,...,t_n are terms. By limiting n in s to have a maximum arity of 2, $SWSRL$ allows only unary or binary predicates (classes or properties). Builtins on the other hand have an arbitrary arity. A term is either a variable where $t_i \in X$ or a constant where $t_i \in \varphi$.

6.3.1 *SWSRLO* SYNTAX

The set of ontological axioms that form the geo-ontology component $SWSRLO$ have a normative Horn rule syntax identical to the core DLP. More formally, an axiom $a \in SWSRLO$ is a logical expression of the form:

$$s_1(t_1, ..., t_n) \wedge ... \wedge s_j(t_1, ..., t_n) \rightarrow h_1(t_1, ..., t_n) \tag{6.4}$$

Where $0 < n \leq 2$ and $j \geq 0$.

6.3.2 $GeoR_{fdD}$ DEDUCTION SYNTAX

The set of all forward deductions rules $GeoR_{fdD}$ in $SWSRL$ extends Horn rules by adding extra-logical builtins, and the following Lloyd Topor And-Or (LTAO) [166] features:

(a) Disjunction is allowed between body atoms

(b) Conjunction is allowed between head atoms

A LTAO transform is used to reduce any rule that contains a disjunction of body predicates into more than one rule with definite head and a conjunction of body literals. As such, Lloyd Topor And-Or provides a natural way of writing more expressive rules*. A rule r where $r \in GeoR_{fdD}$ is a logical expression of the form:

$$s_1(t_1,...,t_n) \wedge / \vee ... \wedge / \vee s_j(t_1,...,t_n) \wedge b_{j+1}(t_1,...,t_v) \wedge / \vee ... \wedge / \vee b_z(t_1,...,t_v) \\ \rightarrow h_1(t_1,...,t_n) \wedge ... \wedge h_i(t_1,...,t_n) \quad (6.5)$$

Where $0 < n \leq 2$, $v \geq 1$, $i \geq 0$, $j \geq 0$, $z \geq 0$.

6.3.3 $GeoR_{fdI}$ Integrity Syntax

The set of all forward integrity rules Geo_{fdI} in $SWSRL$ include classical or more accurately strong negation in the head (denoted $\neg$). Again an integrity rule includes LTAO features and builtins as shown for the set of forward deduction rules. More formally a rule r where $r \in S_{fdI}$ is a logical expression of the form:

$$s_1(t_1,...,t_n) \wedge / \vee ... \wedge / \vee s_j(t_1,...,t_n) \wedge b_{j+1}(t_1,...,t_v) \wedge / \vee ... \wedge / \vee b_z(t_1,...,t_v) \\ \rightarrow (\neg)w_1(t_1,...,t_d) \quad (6.6)$$

Where $0 < n \leq 2$, $v \geq 0$, $d \geq 0$, $i \geq 0$, $j \geq 0$, $z \geq 0$.

In a slight departure from the standard concrete syntax, the concrete syntax of a head predicate, $\mathrm{w}_1(\mathrm{t}_1,...,\mathrm{t}_d)$ / $\neg\mathrm{w}_1(\mathrm{t}_1,...,\mathrm{t}_d)$ of a rule $r \in GeoR_{fdI}$ represents only either $\mathrm{error}(\mathrm{t}_1,...,\mathrm{t}_d)$ or its negation $\neg\mathrm{error}(\mathrm{t}_1,...,\mathrm{t}_d)$, where $\mathrm{t}_1,...,\mathrm{t}_d$ represent the individuals or properties that contribute to the error. Furthermore, as will be shown in section 6.5.1, classical negation is actually removed from the logic of the language altogether, instead represented by a new predicate with prefix *not* e.g. *not*Error and Error. This makes for an easier semantic treatment later on.

6.3.4 $GeoR_{bk}$ Syntax

The set of all backward rules $GeoR_{bk}$ in $SWSRL$ are definite Horn rules extended with builtins - LTAO is not considered for the set of backward rules. More formally, a rule r where $r \in GeoR_{bk}$ is an expression of the form:

$$h(t_1,...,t_n) \leftarrow s_1(t_1,...,t_n) \wedge ... \wedge s_j(t_1,...,t_n) \wedge b_{j+1}(t_1,...,t_v) \wedge ... \wedge b_z(t_1,...,t_v) \quad (6.7)$$

*In reality the re-writing is nothing more than syntactic sugar

Where $0 < n \leq 2$, $v \geq 1$, $j \geq 0$, $z \geq 0$.

6.3.5 CLP_{mutex} SYNTAX

In a Courteous Logic Program (CLP), a mutex defines a mutual exclusion between two literals such that only one can be inferred. This helps to guarantee a consistent answer set. *SWSRL* includes support only for the classical mutex, that is to specify that both an error and its negation (in integrity rules) can not occur at the same time - the concept of a mutex in the context of a CLP is described further in section 6.5.1. The classical mutex has the syntactic form:

$$\bot \leftarrow e_i \wedge \neg e_i \tag{6.8}$$

Where each e_i is an error predicate allowed in the $GeoR_{fdI}$ rule set.

$SWSRL_{overrides}$ **Syntax:** As is common in a CLP, a syntactically reserved binary predicate denoted *Overrides* is used to specify priority between rules that have conflicting head predicates. If no *Overrides* is specified for conflicting rules, neither is inferred - they are treated sceptically. The *Overrides* predicate has the syntactical form:

$$Overrides(i, j) \leftarrow \tag{6.9}$$

where i and j are rule labels or any other allowed rule metadata. A partial order is implied which means that the rule with label i has higher priority than rule with label j.

6.4 THE SEMANTIC WEB SPATIAL RULE LANGUAGE ONTOLOGY - *SWSRLO*

As the Semantic Web Spatial Rule Language Ontology component (*SWSRLO*) is syntactically similar to a DLP, in order to run rules alongside the proposed geo-ontology model in section 5.2.1, the ontology needs to be mapped from its current representation in OWL-DL into the DLP fragment. A DLP can be trivially mapped (using a recursive mapping named DLP-*fusion*) into a traditional logic program with some weakening*. This transformation (DLP-*fusion*) is described in the section to follow.

6.4.1 MAPPING THE OWL-DL GEO-ONTOLOGY

DLP-*fusion*, as defined in [105] using the approach of [19], is a syntactic and semantic preserving bidirectional mapping between a fragment of Description Logic and Horn rules. DL-*fusion* opens the possibility of two different paradigms, namely "rules on-top of ontologies" or "ontologies on-top of rules". We choose to employ the mapping one-way from DL to LP-rules, and hence build "ontologies on-top of rules". Restricting the mapping in this way means that *SWSRLO* will never be reasoned with within existing DL reasoners, which is acceptable due to the limitations of using DL reasoners over large geo-ontology instance bases identified in chapter 4. That said, it is foreseeable that the reverse mapping could be used to map the Horn (DLP like) fragment of *SWSRLO* back into OWL-DL for presentation or compatibility with existing OWL software tools.

Of note the mapping does not consider concrete datatypes, instead it only considers purely logical features such as abstract classes and individuals, hence $\mathcal{SHOIN}$ for OWL-DL as opposed to $\mathcal{SHOIN}(\mathcal{D})$ with concrete datatypes. However using simple concrete datatypes i.e. String, Double and Integer etc. in the resultant language can easily be provided for using procedural attachments in the execution engine or by type assertion [121]. Moreover all spatial datatypes are trivially mapped separately and directly into the location storage system, as an example see Table 6.1.

Mapping Limitations: The mapping does have certain expressive limitations, these are now described as take from [105]. Within OWL-DL certain constructors can occur in either the left hand side (l.h.s) or the right hand side (r.h.s) of an inclusion axiom. For example statement 6.10 shows an inclusion axiom with a disjunctive r.h.s, while statement 6.11 shows an inclusion axiom with a disjunctive

*For example logic programs have a implication operator that does not conform to the classical material implication - instead abiding by its own operational semantics which do not for example allow partial rule resolution

Geofeature Geometry	GeoLMS (Oracle 10g) Table
District(Roath) → Geometry → polygon → Coord(3,13) → Coord(11,13) → Coord(11,21) →Coord(3,21)→Coord(3,13)	INSERT INTO locationBase VALUES('http://cf.ac.swsrl.ont/Roath',MDSYS.SDO_GEOMETRY(2003,8307,null,MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1), MDSYS.SDO_ORDINATE_ARRAY (3,13,11,13,11,21,3,21,3,13)))

Table 6.1: Example geometry mapping into the LSS of the district Roath

l.h.s.

$$C \sqsubseteq D \sqcup E \tag{6.10}$$

$$C \sqcup D \sqsubseteq D \tag{6.11}$$

The mapping between OWL-DL and the definite Horn fragment deals differently with each case and, as a result, two different languages emerge, $\mathcal{L}_h$ for head or r.h.s inclusion axiom mappings and $\mathcal{L}_b$ for body or l.h.s. mappings. This distinction is important, as described in some detail in [105], as not all OWL-DL axioms are directly representable within the definite Horn fragment. For example statement 6.10 corresponds to the following Horn rule (assuming C,D and E are classes from the DL expression 6.10):

$$C(x) \rightarrow D(x) \vee E(x) \tag{6.12}$$

It is easy to see that rule 6.12 is a disjunctive rule and therefore outside the scope of definite Horn. Moreover, the following constructs are outside the scope of definite Horn:

- Functional properties and cardinality restrictions - both require the use of variable equality and inequality which is outside the scope of the Horn fragment. For example, in the functional case equality between the variables y and x is needed to enforce the uniqueness of the property e.g. $\forall x, y, z\ (P(x, y) \wedge P(x, z) \rightarrow y = z)$. However as *SWSRL* assumes a LP implementation, a procedural attachment can be used as a weak test of equality [104]*, which can then be used to represent this restriction.

*weak because it does not hold all the same properties i.e. it is a simple syntactic match on a string value, rather than a true test of object equality using an axiomization of equality

- Disjunction in the r.h.s of the inclusion axiom. For the reasons shown above (see rule 6.12). Disjunction can however be dealt with on the l.h.s of an inclusion axiom as the disjunction can be removed using the Lloyd-Topor transform.

- Universal Restrictions in the l.h.s of the inclusion axioms. For example if contained within the l.h.s. i.e. $\forall P.D \sqsubseteq C$ the equivalent first order logic expression of $(P(x,y) \rightarrow D(y)) \rightarrow C(x)$, can not be translated to a definite Horn clause, as the translation requires the introduction of negation. However a mapping does exist if the restriction occurs in the r.h.s. for example: $C \sqsubseteq \forall P.D \equiv (D(y) \leftarrow P(x,y)) \leftarrow C(x)$, which results in the definite Horn rule: $D(y) \leftarrow C(x) \wedge P(x,y)$.

- Existential restrictions on the r.h.s of an inclusion axiom can not be mapped as this mapping requires an existentially quantified variable in the head of the resulting Horn rule - which is a known source of undecidability. A mapping does exist however when the existential restriction occurs on the l.h.s. of an inclusion axiom for example $\exists P.C \sqsubseteq D$ which maps to the definite Horn rule $P(x,y) \wedge C(y) \rightarrow D(x)$.

- Negation - all negation is omitted (concept/property negation) in particular classical negation and negation as failure is outside the scope of definite Horn. However, a simplified, syntactic form of negation has been added to the defeasible component of *SWSRL*.

Recursive Mapping: DLP-*fusion* is implemented using a recursive mapping function $\mathcal{T}$, which takes a DL axiom in one of the following forms: $(C \sqsubset D)$ or $(S \equiv B)$ or $(\top \sqsubseteq \forall P.D)$ or $(\top \sqsubseteq \forall P^{-}.D)$ or $(a : D)$ or $(< a, b >: P)$ or $(P \sqsubseteq Q)$ or $(P \equiv Q)$ or $(P \equiv Q^{-}))$ or $(P^{+} \sqsubseteq P)$ and outputs a definite Horn rule of the form $A \leftarrow B$. Intuitively this mapping could be (as is the case with *SWSRLs* forward system) mapped onto a rule of the form $B \rightarrow A$ with identical entailments. The complete mapping function $\mathcal{T}$, as taken from [105], is shown in Appendix B.2. As a result, *SWSRLO* can contain the DL axioms shown in table 6.2, as represented after employing the mapping function $\mathcal{T}$ in Horn LP syntax.

The mapping preserves semantic equivalence and is tractable [105]. As an example mapping, take the following simple OWL-DL axioms, where Region, Ward

DL Axiom	*SWSRLO* Horn Rule (LP) Syntax
$C \sqsubseteq D$	$C(x) \rightarrow D(x)$
$A \equiv B$	$A(x) \rightarrow B(x)$ and $B(x) \rightarrow A(x)$
$\top \sqsubseteq \forall P.D$	$P(x,y) \rightarrow D(y)$
$\top \sqsubseteq \forall P^{-}.D$	$P(x,y) \rightarrow D(x)$
$P \sqsubseteq Q$	$P(x,y) \rightarrow Q(x,y)$
$P \equiv Q$	$P(x,y) \rightarrow Q(x,y)$ and $Q(x,y) \rightarrow P(x,y)$
$P \equiv Q^{-}$	$P(x,y) \rightarrow Q(y,x)$ and $Q(y,x) \rightarrow P(x,y)$
$P^{+} \sqsubseteq P$	$P(x,y) \wedge P(y,z) \rightarrow P(x,z)$
$a : C$	$true \rightarrow C(a)$
$< a, b >: P$	$true \rightarrow P(a,b)$

Table 6.2: SWSRLO axioms and their corresponding Horn Rule (or LP) Syntax

and Unitary Authority are classes, and Inside and Contains are Properties (in particular OWL Object Properties).

$$\begin{aligned} Ward \sqcup Unitary\_Authority &\sqsubseteq Region \\ Inside^{+} &\sqsubseteq Inside \\ Contains &\equiv Inside^{-} \end{aligned}$$

These are then mapped using the function $\mathcal{T}$ and the LTAO transformation to the following set of definite Horn rules:

$$\begin{aligned} Ward(x) &\rightarrow Region(x) \\ Unitary\_Authority(x) &\rightarrow Region(x) \\ Inside(x,y) \wedge Inside(y,z) &\rightarrow Inside(x,z) \\ Contains(x,y) &\rightarrow Inside(y,x) \\ Inside(y,x) &\rightarrow Contains(x,y) \end{aligned}$$

The transformation function is also applied to the OWL-DL representation of the error ontology. Consequently, errors are added as facts into *SWSRLO* when derived.

OWL-DL Geo-ontology to *SWSRLO* Concrete Mapping: For the reminder of this work, the axioms of the OWL based geo-ontology defined in Section 5.2 are converted into an *SWSRL* geo-ontology using the mapping function $\mathcal{T}$. This mapping forms the core ontology subset of *SWSRL* namely *SWSRLO*. The transformation can be automated for example using the KOAN2 DLP transformation tool - dlpconvert [187] which produces a DLP in Prolog syntax. For the purpose of this thesis, the geo-ontology model is fixed and hence the mapping of the OWL-TBox of the geo-ontology needs only to be performed once. Once converted, all additional operations are performed over the rule based (or LP based) syntax of *SWSRLO*. Of course if the geo-ontology model was extended, the mapping would need to be re-run in order to generate a new DLP program and ultimately a new *SWSRLO* geo-ontology. The *SWSRLO* geo-ontology is populated by adding individuals directly to the Horn LP fragment of *SWSRLO* as facts, for example: `Region(NS:Cardiff)` and `NS:Disjoint(NS:Wales,NS:England)`, where `NS:` represents a namespace prefix.

As an example of the mapping, we first show a sample of the TBox axioms of the OWL geo-ontology in Table 6.3 (the full set are shown in Appendix B.1).

Axiom Number	DL Syntax
3	$\top \sqsubseteq \leq 1\mathsf{Name}.\top$
5	$\top \sqsubseteq \forall \mathsf{AlternativeName}^{-1}.\textsc{GeoFeature}$
6	$\top \sqsubseteq \forall \mathsf{Name}^{-1}.\textsc{GeoFeature}$
7	$\mathsf{Within} \sqsubseteq \mathsf{Topological}$
8	$\mathsf{WestOf} \sqsubseteq \mathsf{Cardinal\_Direction}$
10	$\mathsf{Topological} \sqsubseteq \mathsf{Spatial\_Relationship}$
11	$\mathsf{CoveredBy} \sqsubseteq \mathsf{Topological}$
20	$\mathsf{Cardinal\_Direction} \sqsubseteq \mathsf{Spatial\_Relationship}$
26	$\textsc{Region} \sqsubseteq \textsc{GeoFeature}$

Table 6.3: Sample geo-ontology axiomisation

***SWSRLO* TBox axioms:** Table 6.4 then shows a sample fragment of the fully converted *SWSRLO* geo-ontology that is formed after applying the mapping function $\mathcal{T}$ - Appendix B.2 lists the full transformed *SWSRLO* geo-ontology.

Axiom Number	LP Horn Syntax
3	Name(x,y) ∧ Name(x,z) → Equal(y,z)
5	AlternativeName^{-1}(x,y) → GEOFEATURE(x)
6	Name^{-1}(x,y) → GEOFEATURE(x)
7	Within(x,y) → Topological(x,y)
8	WestOf(x,y) → Cardinal_Direction(x,y)
10	Topological(x,y) → Spatial_Relationship(x)
11	CoveredBy(x,y) → Topological(x,y)
20	Cardinal_Direction(x,y) → Spatial_Relationship(x)
26	REGION(x) → GEOFEATURE(x)

Table 6.4: *SWSRLO* geo-ontology

One immediate criticism of the above mapping, is that we do not model and map certain spatial relations using the more advanced property types of OWL-DL, namely symmetric ($P(x,y) \rightarrow P(y,x)$), transitive ($P(x,y) \wedge P(y,z) \rightarrow P(x,z)$) and inverse ($P(x,y) \rightarrow Q(y,x), Q(y,x) \rightarrow P(x,y)$). For example, *Within* is a transitive topological spatial relation between regions which would better be represented by the following OWL-DL property: Within$^+$ ⊑ Within, which maps to $\forall x, y, x\ Within(x,y) \wedge Within(y,z) \rightarrow Within(x,z)$. However, properties of these relations are represented when we discuss the mapping of topological compositional inference rules in Chapter 7.

6.5 *SWSRL* General Extensions

In this section we outline the two general extensions to the core *SWSRL* language, namely; a simplified version of Courteous Logic and an interleaved execution mode of inference.

6.5.1 Courteous Logic Programs Extension (CLP^-)

Within the context of integrity rules, it can be useful to specify a form of default (defeasible) reasoning. That is, an integrity violation may be later removed by the addition of new knowledge. To facilitate this feature in *SWSRL* we employ a form of Courteous Logic (see section 3.2.3 for a more in depth discussion of Courteous

Logic Programs).

The full implementation of a Courteous Logic Program (CLP) suffers from a more complicated semantics than most standard ordinary logic programs (e.g. using the Well Founded Semantics [263]) and is not directly representable in ordinary logic programming engines such as Prolog [38] or our proposed language *SWSRLO*. Often a CLP is implemented within an ordinary logic program i.e. Horn logic program, by preprocessing the CLP with a (more complicated) courteous compiler [102]. A courteous compiler compiles away the expressive CLP extensions, leaving a semantically equivalent ordinary logic program [102]. The resulting logic program can then be implemented in most rule engines, for example in JESS (a Rete based rule system) in the open source SweetRules tool suite[106]

Here however, we propose to adapt and simplify such an implementation of the logic by placing some expressive restrictions on the Courteous Logic component, removing the need for a courteous compiler. In all we have made the following expressive restrictions to the full Generalised Courteous Logic [102]:

- Mutual exclusion constraints (conditional or unconditional -'mutex's') are not definable by the user. Only the classical mutex, one for each error predicate *error*, is implicitly allowed within the program (as in a Basic Courteous Logic Program BCLP).

$$\perp \leftarrow error(?X_1, ..., ?X_m) \wedge \neg error(?X_1, ..., ?X_m). \tag{6.13}$$

  Where the arity of *error* is m.

- Classical negation is restricted to integrity rule head atoms only i.e. to infer *error* and its negation $\neg error$. The appearance of strong negation is eliminated in *SWSRL* by assuming new syntactic predicates *notError* and *Error* as discussed in section 6.3.3

SWSRL is therefore reducible to a definite Horn logic program. It does not contain negation as failure and the limited form of classical negation is eliminated. The following features of a CLP have not been removed or restricted:

- Each rule will have a rule label, which is used by the prioritisation predicate *Overrides*.
- The prioritisation predicate *Overrides* can be used as a simple fact or inferred from a rule.
- Additional types of rule meta tags can be used in addition to only a rule label. During reasoning, rule metadata is converted to 0-ary function symbols (constants), which can then be used to infer priorities amongst integrity rules.

6.5.2 Interleaved Execution Extension

SWSRL mixes the execution of forward chaining and backward chaining rule sets. For example, the antecedent in a forward rule can be found not only from raw facts, but by deduction from a backward rule. This is beneficial when the knowledge base is purposefully incomplete (for storage reasons) but where implicit information can be derived on the fly.

Existing Techniques: Interleaving backward and forward rules, sometimes referred to as mixed mode reasoning, is not a widely used technique with only limited examples of such systems. The M.4. system* combines forward and backward reasoning modes, however its predominately a forward firing system, and the backward system uses a different syntax. Indeed the most popular method of invoking a backward rule from within the context of a forward rule is to make an explicit call to a particular backward rule via a reserved predicate. This technique is used within Eclipse† and MIKE [59]. Algernon [41] has a seamless form of mixed mode reasoning, both backward and forward systems share the same syntax. Algernon uses Access Limited Logic [42], where access paths are built form left to right as the rule is run, and any call to the backward system is encoded as an `ask` predicate which triggers the execution of a backward rule set during the course of forward rule evaluation.

Motivation: Storing all possible spatial relations leads to an overinflation of the Abox [271]. We propose interleaving backward and forward reasoning is useful in

*see `http://www.teknowledge.com`

†see `http://www.halye.com`

the spatial domain, for efficiently reasoning over topological in the geo-ontology. Realistic geo-ontologies are potential very large, with thousands of geofeature instances. Further, as each place is related spatially to every other place, even with only topological relations (as we do in this work) stored, the ontology would contain $O(n^2)$ relations - where n is the number of geofeature instances. As an example, the administrative wards of Wales contains a relatively small number (920) of place instances, but a much larger number (846400) of topological relations. Such large numbers of relations increases the size of the ontology substantially. More specifically, using *SWSRLs* geo-ontology model, the Wards of Wales takes 132kb (in XML/RDF syntax) of persistent storage with all 920 administrative wards of Wales stored and no relations. Increasing to 26,900kb when all 920 wards and the 846400 topological relations are stored - roughly a 200 times storage increase.

Spatial reasoning techniques can derive a certain percentage of these relations from a smaller base of explicit relations. By using a forward reasoning system for topological deduction rules, all entailed relations are added back into the ontology. However, querying for topological entailments using the backward system does not add unnecessary information back into the ontology, helping to reduce storage overheads. This is particularly useful when running integrity rules which only aim to test the consistency of topological relations, and can query for necessary topological relations during their execution using interleaved reasoning - no new topological relations need to be added to the ontology for integrity rules to run correctly. In effect, interleaved reasoning can help provide full reasoning support to an *edge reduced* [271] spatial configuration - a form of spatial relationship compression.

Lastly, we propose in-memory reasoning of topological rules over ontologies with large numbers of stored relations can be impractical in systems with a limited memory capacity. We propose that computing these topological relations as and when needed in an interleaved mode of reasoning, will help to decrease the in-memory size of a place ontology, subsequently reducing memory bottlenecks. Of course this will have a negative effect on the speed of reasoning as described in section 8.3.

6.6 *SWSRL* SPATIAL EXTENSIONS

In this section we explore the various spatial extensions incorporated into the definition of *SWSRL*. These extensions affect the syntax and logic of the language, as well as providing a linkage between the language and the Location Storage System (LSS).

6.6.1 SWSRL SPATIAL SYNTAX

A number of efforts have been made by large companies and organizations to standardise the vocabulary of geographic information enabling better sharing and interchange. The Open Geospatial Consortium* have produced a number of standards formally defining geographic data models and their associated functions, along with the definition of the geographic interchange format GML [195].

Most existing rule languages are general and do not place restrictions on the exact syntax of unary and binary predicates they represent. Therefore, under these conditions, the semantics of the rule predicates would need to be agreed upon by each rule author. Without proper agreement on the vocabulary and semantics of allowed predicates, rules from different authors may not use the same ontological facts, or may use the same facts but assuming different meaning, leading to inaccurate or incomplete inferences. For example, the knowledge that object a is geometrically contained by object b could be represented (qualitatively) by a number of different topological spatial relation terms e.g. *inside*, *within* or subtle variants e.g. *in* or *IsWithin*.

An *SWRLO* geo-ontology is a conversion of the OWL-DL geo-ontology. The OWL-DL geo-ontology is based on standard geographic data models, and as such has a well understood vocabulary and semantics. Therefore once converted to *SWSRL*, it is important that any new fact in *SWSRLO* or user defined rule in *SWSRL* also conforms to the same vocabulary and semantics. Subsequently, *SWSRL* ontologies will be syntactically restricted to conform to the same standard geographic model, based on the OGC specification of geographic features and their spatial relationships. In effect, the syntax of the predicates of *SWSRL* must conform to the types of classes and relations allowed in the OWL-DL ontology that

*`http://www.opengeospatial.org`

was mapped into *SWSRL* using the transformation function $\mathcal{T}$ - in this sense the syntax of *SWSRL* is a mirroring of the syntax in the original OWL-DL ontology.

Furthermore, a standard vocabulary for spatial relations is also important for the interleaved reasoning mode described previously. That is, certain spatial relation predicates in the forward system trigger rules in the backward system. Therefore it is import that the semantics of these predicates are understood in the two reasoning modes to deliver desirable and valid inferences.

6.6.2 SWSRL Concrete Syntax

To remain complaint with existing geospatial standards and the OWL-DL geo-ontolgy model in section 5.2.1, the vocabulary of *SWSRL* is taken from the OGC simple features specification*, the ISO 19109 series and the OGC filter specification for SQL. The vocabulary and grammar of the language is defined by *SWSRL*'s concrete syntax which is now described using Backus Naur Form (BNF). All literal values are prefixed by a fully qualified URI, which is shown here using the namespace prefix NS.

6.6.2.1 Facts in *SWSRLO*

All bodyless rules or facts in *SWSRLO* have the form:

→ <Fact>

<Fact> := NS:FeatureType(x) | <thematic-property> | <spatial-property> | builtin(r.

Where x is substituted by a Geofeature and FeatureType is substituted for an actual feature type or class in *SWSRLO*.

6.6.2.2 $GeoR_{fd}$ Rules

The set of forward rules are split (as in their abstract syntax) into forward deduction and forward integrity rules.

*see `http://www.opengis.org/techno/specs.htm`

$$< body > \rightarrow$$ <DeductionHead>

$$< body > \rightarrow$$ <IntegrityHead>

<integrityHead> ::= ¬ <Error> | <Error>

<deductionHead> ::= <atom> | <atom> and <deductionHead>

<body> := <atom> | <atom> and <body>

<atom> = NS:FeatureType(x) | <thematic-property> | <spatial-property> | builtin(r,x...)

Where x is substituted by a Geofeature and FeatureType is substituted for an actual feature type or class in *SWSRLO*.

6.6.2.3 $GeoR_{bk}$ Rules

A backward rule has the form:

$$< Head > \leftarrow$$ <Body>

<Head> ::= <spatial-property >

<Body> ::= NS:FeatureType(x) | NS:FeatureType(x) and <Body> | <spatial-property > | <spatial-property> and <Body>

Where x is substituted by a Geofeature and FeatureType is substituted for an actual feature type or class in *SWSRLO*.

6.6.2.4 Common Constructs

<spatial-property> Spatial properties match the types of spatial property axioms in *SWSRLO*. Topological relations consist of the set of RCC-8 base relations (but using the n-intersection terminology) along with the set of generalised RCC-12 base relations (used for topological spatial reasoning rules in Chapter 7).

<spatial-property> ::= <STP> | <SDP> | <SOP> | <SSP>

<STP> ::= NS:Meets(x,y) | NS:Contains(x,y) | NS:Inside(x,y) | NS:Covers(x,y) | NS:Intersects(x,y) | NS:Equals(x,y) | NS:Disjoint(x,y) | NS:Overlaps(x,y) | NS:CoveredBy(x,y))

| NS:P(x,y) | NS:C(x,y)| NS:DR(x,y)| NS:P-1(x,y)| NS:NTP(x,y)| NS:coNTP(x,y)| NS:coNTP-1(x,y)| NS:O(x,y)| NS:DC(x,y) | NS:NTP-1(x,y)| NS:coP(x,y)| NS:coP-1(x,y)
<SDP> ::= NS:Near(x,y) | NS:Far(x,y)
<SOP> ::= NS:NorthOf(x,y) | NS:SouthOf(x,y) | NS:EastOf(x,y) | NS:WestOf(x,y)
<SSP> ::= NS:LargerThan(x,y) | NS:SmallerThan(x,y)

Where x and y are variables substituted for Geofeatures.

<thematic-property> Thematic properties are geospecific non-spatial properties, for example; population, or geofeature identifier etc.

<thematic-property> ::= NS:P(x,val) | NS:P(x,y)

Where *val* is substituted for a literal value from the concrete datatype domain, P is the name of the thematic property, and x is substituted for a Geofeature e.g Identifier(Cardiff, 1232) or Population(Cardiff,456000).

<Error> The error predicate is the only predicate in SWSRL that can be either positive or strongly negative (¬). A strongly negated error is actually represented syntactically as *notError*, and hence is ignored semantically within the standard logic of *SWSRL* (helping provide a simpler semantics for *SWSRL*). The arguments of the error predicate are similar to those described in [260]. *SWSRL* allows for the implication of either a relationship error between two Geofeatures, or an error between a Geofeature and one of its properties.

<Error>::=NS:error(*name*, *ind*, *rel*/*prop*, *ind2*/*value*, *desc*)

Where, *name* - is the name of the error. *ind* - is the first individual. *ind2* - is the second individual. *value* - is the value of the property. *rel* - is the relationship between the first and second individual. *prop* - is a property of the first individual. *desc* - is a textual description of the error. All values bar *ind* and *ind2*/*value* are constants, and hence must be specified during rule authoring.

The arguments *name*, *ind*, *rel* or *prop*, *ind2* or *value* form the errors signature. Positive and negative errors are in conflict if they share the same ground signature - that is, each variable is substituted for the same value. The succeeding error is decided by the CLP^- feature, see section 6.5.1. The error predicate adds errors to *SWSRLO* as error facts conforming to the simple error ontology model.

Errors are encoded as builtins in *SWSRL* (as they have more than two terms, however only two free variables), these are then reified into many logical facts during execution of the reasoning engine.

Builtins: The complete list of spatial and non-spatial builtins is described in Section 6.6.4.

6.6.3 Geospatial Rule Metadata

Within most rule languages, rules are identified by an alphanumeric rule label. In certain systems (for example SweetRules*) the definition of arbitrary rule labels is allowed. In addition, adding extra tags to rules described in the *de facto* rule interchange format RuleML [4] would be fairly trivial. However, we believe that spatial rules (not facts) should be augmented with a more formal definition of geospatial rule tags for the description, identification and categorisation of designed and implemented rules within large rule sets. Spatially annotating rules allows for the possible construction of a spatial rule ontology, supporting ontological reasoning about rule sets, and supporting development of rule bases. Additionally, geospatial rule metadata serves as input to the default Courteous (CLP^-) reasoning process (for specifying rule priority between locally conflicting rules). All rule tags are treated as 0-ary function symbols (constants) which are preserved during rule instantiation - each possible grounding of a rule containing free variables will share the same tag values. Hence, each metadata tag is represented internally as logical predicates so that it can be used through the reasoning process. All rules, forward or backward, contain rule meta tags, however only forward integrity rules will ever be the scope of prioritized conflict handling - hence priority in the CLP^{-1} component can be derived from all forward integrity rule metadata. All other rule meta tags are included to provide better rule management. A more formal definition of

*see `sweetrules.projects.semwebcentral.org/`

the metadata elements is given in the paragraphs to follow.

Rule Name As used by most rule languages, a rule can have an alphanumeric name associated with it. This helps with rule visualisation and defining priority for integrity rules. Each rule has have a label denoted *label.* The rule label follows the unique name assumption, in that each unique rule will require a unique label to prevent ambiguity. Each grounding of a rule with free variables has the same label.

$$[< label > \textbf{label} < /label >: rule]$$

Rule Type Two rule types can be defined within *SWSRL*. These are deduction and integrity rules. Rule types will use the meta tag *<ruleType> type </ruleType>*:

$$[< ruleType > \textbf{type} < /ruleType > ... < label > ... < /label >: rule]$$

where *type* will be an integer value of either 0 or 1, and 0 represents a deduction rule, and 1 represents an integrity rule.

Rule Level Rules at different levels can be useful in specifying a general prioritization of rule sets rather than specifying prioritization between individual rules. Rule levels are represented by an integer value n where $n \geq 0$. The level $n = 0$ is reserved for space laws - expertly defined integrity / deduction rules that govern the structure of the space. . Rule levels represent a total ordering of rule sets.

$$[< ruleLevel > \textbf{level} < /ruleLevel >< label > ... < /label >: rule]$$

As an example of using rule levels, all level 0 rules can override all level 1 rules. More formally:

$$[< meta - data >: ruleLevel(?A\ 0) \wedge\ ruleLevel?(B\ 1) \rightarrow overrides(A\ B)]$$

As a result, all rules that have a rule level of 0 will override rules that have a level of 1, for those rules that infer conflicting error predicates*.

Rule Group Rules can be grouped together e.g. topological rules, or directional rules etc. The categories for rule groups are influenced by those defined by Cockroft [35] and are shown in Appendix B.6.1. A rule grouping has the meta rule label *<ruleGroup> name </ruleGroup>* where *name* is an alphanumeric rule grouping name.

[< *ruleGroup* > **name** < */ruleGroup* > ... < *label* > ... < */label* >: *rule*]

6.6.4 PROCEDURAL ATTACHMENTS

A logic program is based on 'pure belief' alone. By providing mechanisms that go outside of classical logic additional functionality is gained. Such mechanisms are often provided in the form of external procedural attachments. Many logic engine implementations provide a set of static predefined procedural attachments named builtins. Builtins commonly revolve around simple arithmetic procedures or comparison procedures. Extending *SWSRL* with procedural attachments can lead to a more complicated semantic treatment if the attachments are allowed to affect the logic program in any way. For example a $remove(t_1...t_n)$ procedural attachment which removes knowledge is nonmonotonic, and has adverse side effects on the knowledge base and previously drawn entailments during reasoning (much like the negation as failure operator).

Work in providing a standardized understanding and clean semantic treatment of procedural attachments, has led to the development of 'Situated' Logic Programs [103]. A logic program is situated if it has a rich and tight attachment with external procedures. A situated logic program has two categories of procedural attachments namely sensors and effectors. A sensor is used to test an antecedent's condition - a predicate in the body of a rule. An effector is a predicate that links to an external 'action' procedure in the head of a rule i.e. it runs when the conclusion is drawn. A situated logic program can assign a predicate to an external procedure during run time via a rule.

*Remembering that; Errors conflict iff there exists positive and negative error predicates that have a similar signature, or the same variable bindings error(t_1,...,t_n) $\neg$ error(t_1,...,t_n) respectively in *SWSRLO*

An effector and sensor procedure could range from a simple method call to a remote procedure call from a database. Both operate *independent of inference control* - the time of execution is irrelevant. Effectors are invoked after inferencing has been performed. All standard non-spatial and spatial procedural attachment in *SWSRL* are semantically clean as they do not directly alter information in the ontology, and the time of their execution is irrelevant. Each attachment is encoded by means of builtins and executed in a procedural programming language e.g. Java along with the implementation of the rule engine, see Chapter 8. In this section we outline the different procedural attachments of builtins that are used in *SWSRL*.

6.6.4.1 Comparison Operators

The comparison operators, taken from the OGC Filter Specification [268], evaluate the mathematical comparison of two arguments. All are binary predicates of the form $Comp\_OPP(P_1, V)$ (where $Comp\_OPP$ is substituted by the name of the operator) except PropertyIsBetween which is a ternary builtin of the form $Com\_OPP(P_1, V_1, V_2)$. All return true if they succeed. SWSRLs comparison operators are summarised in table B.4 in Appendix B.7.

6.6.4.2 Arithmetic Operators

SWSRL's arithmetic operators encode fundamental arithmetic operations and are identical to those found in most existing rule languages/engines. All comparison operators are either ternary predicates of the form $Arth\_OPP(V_1, V_2, R)$ or binary predicates of the form $Arth\_OPP(V_1, V_2)$, where $Arth\_OPP$ is substituted by the name of the operator, V_1 and V_2 are values from the set of real numbers $\mathbf{R}$, R is a variable bound to the result of applying the operator to V_1 and V_2. If R is omitted, the predicate becomes a test of truth. *SWSRL*'s comparison operators are summarised in table B.6 in Appendix B.7.

Clearly the language can trivially be extended with any number of binary or unary mathematical operators (for example Square Root etc.).

6.6.4.3 Spatial Operators

SWSRLs spatial operators work over the geometry of features or individuals in *SWSRLO*. Therefore, all spatial operators are computed within the Location Storage System. Geometric computation will not succeed when the feature is not associated with its location or geometry (which is a valid state within the overall *SWSRL* framework). In this case the operator will evaluate as false and thus prevent the rule from firing, avoiding invalid inferences. All spatial operators are either ternary predicates of the form $Sptl\_OPP(Ind_1, Ind_2, R)$ or binary predicates of the form $Sptl\_OPP(Ind_1, R)$, where $Sptl\_OPP$ is substituted by the name of the operator, Ind_1 and Ind_2 are individuals or features from *SWSRLO* and R is a variable bound to the result of applying the operator to Ind_1 and/or Ind_2.

***SWSRL* LSS Builtins** Within *SWSRL* the backward system can make calls via procedural attachments (builtins) to the *LSS* to quantitatively compute (on the fly), topological, directional and relative size relationships between any two features of *SWSRLO* which have associated geometry. All calls to the *LSS* have an *ex* prefix. This mechanism is evaluated later with respect to its ability to effectively and efficiently mix both qualitative and quantitative reasoning. This is beneficial when real world *SWSRLO* ontologies may only be partially complete with respect to explicitly stored qualitative spatial relations or geometric attributes.

Topological: *SWSRL* includes spatial operators to compute topological relationships between two features from *SWSRLO*. Topological spatial operators, shown in Appendix B.7, are based on the terminology of the point-set theoretic semantics of Egenhofers n-intersection model, which then conforms to most modern spatial database implementations which adhere to the OGC Simple Feature Specification for SQL [267]. Topological operators are binary predicates of the form $Topo\_opp(Ind_1, Ind_2)$ where $Topo\_opp$ is substituted by the name of the operator and Ind_1 and Ind_2 are individuals or features from *SWSRLO*. A table of all 8 topological relations is shown in Appendix B.7.

Relative Size: SWSRL also includes operators that can compute the relative size between two features that have an associated area in $\mathbb{R}^2$ (for this we assume polygonal features). Relative size operators are binary predicates of the form $RS\_opp(Ind_1, Ind_2)$ where RS_opp is substituted by the name of the operator and Ind_1 and Ind_2 are individuals or features from *SWSRLO*. The table of *SWSRL* relative size operators is shown in Appendix B.8.

Cardinal Direction: SWSRL includes operators that can compute cardinal direction between two features based on Frank's projection based model [78] that uses a global, west-east/south-north extrinsic frame of reference. To compute the direction all features are generalised (dimensional reduced) to a point - typically based on weighted centroids - and the space around the point (which becomes the primary point) is divided into 8 equal size regions. One of the four Cardinal directions from the set D_4 (6.14) can then be determined between the primary feature and any other reference feature in *SWSRLO*.

$$D_4 = \{N, E, S, W, 0\} \quad \text{where 0 represents the neutral point} \qquad (6.14)$$

Cardinal direction builtins assume that Ind_1 is the reference object and Ind_2 is the primary object. All Cardinal directions are binary predicates of the form $CD_opp(Ind_1, Ind_2)$ where CD_opp is substituted by the name of the operator and Ind_1 and Ind_2 are individuals or features from *SWSRLO*. Cardinal direction operators are shown in Appendix B.9.

6.7 *SWSRL* SEMANTICS

This section completes the description of *SWSRL* by defining the semantics of the language. All of *SWSRL* is treated as a logic program (LP) assuming the closed world and unique name assumptions. An LP has well understood minimal Herbrand model semantics or Datalog semantics.

Backward rules (anything in $GeoR_{bk}$) corresponds to plain Datalog (no disjunction, existential quantification, function symbols or negation). Hence, a simple minimal Hebrand semantics apply. Forward rules, in particular the set of forward integrity rules, have more expressive intent, but can be dealt with using a Herbrand semantics as all its expressive features are simulated i.e. conversion of psuedo classical negation to a new predicate treated in a negative context (a common treatment), and then dealt with using the CLP^{-1} post execution cleanup (whose exact implementation is described procedurally in chapter 8). A post execution cleanup is made possible because the error predicate, the only predicate that can be syntactically negated, is restricted such that is can not be the subject of inference (contained in any rule body), hence during inferencing the language remains monotonic. The interaction between forward and backward rule systems is via extra-logical procedural attachments, in theory the interleaved interaction is considered using an operational fixed point semantics.

We now describe the semantics of *SWSRL* by first considering *SWSRLO*, $GeoR_{fdD}$, $GeoR_{fdI}$ and $GeoR_{bk}$ separately, and then by considering their interleaved interaction.

6.7.1 *SWSRLO* SEMANTICS

Inferencing over the geospatial ontology axioms represented in *SWSRLO* is fairly trivial as they are akin to horn clauses, conforming to a simple declarative Datalog style minimal Herbrand model semantics as opposed to OWL's model theoretic semantics. More formally, Let KB be a *SWSRLO* knowledge base. KB is satisfiable iff KB has a minimal Herbrand Model (denoted M_{kb}). A satisfiable KB entails *SWRLO* ground facts F iff $M_{kb} \models F$.

6.7.2 $GeoR_{fd}$ SEMANTICS

Deduction rules: $GeoR_{fdD}$ are akin to simple Horn clauses and thus conform to standard Datalog minimal Herbrand model semantics (declarative).

Consequently the semantics of forward deduction rules is as follows. Let KB be a $SWSRLO \cup GeoR_{fdD}$ knowledge base. KB is satisfiable iff KB has a minimal Herbrand Model (denoted M_{kb}). A satisfiable KB entails a *SWRLO* ground (variable free) fact F iff $M_{kb} \models F$. Of course in practice as $GeoR_{fdD}$ represents a set of forward production rules, any entailment is permanently added to the set of asserted ontology facts in *SWSRLO*.

Integrity rules: $GeoR_{fdI}$ is assumed to conform to a minimal Herbrand model semantics even though rule heads can contain pseudo-classical negation. However, the negated predicate (Error) is not permitted in the body of any rule $r \in GeoR_{fd}$. Moreover, in the concrete syntax of *SWSRL*, classical negation on error predicates is replaced by two new standard predicates *not*Error and Error. In effect the logical extension of negation is removed, and general rules in *SWSRL* are monotonic - only conflicting Error facts can be removed after inferencing.

Traditionally the integrity ruleset should take the following semantics. Let KB be a $SWSRLO \cup GeoR_{fdI}$ knowledge base. As $GeoR_{fdI}$ can only entail positive or negative errors, KB is only satisfiable iff no ground instantiation of the body of an integrity rule $r \in GeoR_{fdI}$ is true in the minimal Herbrand Model of KB (M_{kb}). Therefore any violated integrity rule renders the KB unsatisfiable.

However, the intended behaviour of *SWSRL* is that forward integrity rules serve to derive error facts in *SWSRLO*. Even though on the first deduction of an error *SWSRLO* should be regarded inconsistent, inferencing in the forward system continues. The reason for this is two fold, firstly all entailed errors are needed by the statistical error localisation techniques described in chapter 8. Secondly, some errors maybe overridden by their negation (the default assumption may not apply), thus only after error conflicts have been resolved using CLP^{-1} can we assume *SWSRLO* is satisfiable (consistent). Hence integrity rules will assume the same semantics as deduction rules. Where, KB is satisfiable iff KB has a minimal Herbrand Model (denoted M_{kb}). A satisfiable KB of $GeoR_{fdI}$ entails an *SWRLO* ground fact (error) E iff $M_{kb} \models E$ (where E is either a positive or negative error). The model M_{kb} may then contain positive and negative versions of the same error, hence M_{kb} is feed into the CLP^{-1} cleanup step, where knowledge of priority in overrides predicates in *SWSRL* are used to remove conflicts. If errors remain after the CLP^{-1} step, *SWSRLO* is unsatisfiable.

Combined Semantics: In practice both forward integrity and deduction rules are run together. Their combination is fairly trivial as both assume the same semantics (after the expressive features of integrity rules have been simulated syntactically). Hence the combined semantics of the forward system is as follows. Let KB be a $SWSRLO \cup GeoR_{fdI} \cup GeoR_{fdD}$ knowledge base, KB is satisfiable iff KB has a minimal Herbrand Model (denoted M_{kb}). A satisfiable KB entails a *SWRLO* ground (variable free) fact F iff $M_{kb} \models F$ (where F could be a positive or negative error). Again as before, M_{kb} is feed into the CLP^{-1} cleanup step where knowledge of priority in overrides predicates in *SWSRL* are used to remove conflicts. If errors remain after the CLP^{-1} step, *SWSRLO* is unsatisfiable.

6.7.3 $GeoR_{bk}$ SEMANTICS

$GeoR_{bk}$ are simple Horn clauses conforming to a declarative Datalog style minimal Herbrand model semantics. Let KB be a $SWSRLO \cup GeoR_{bk}$ knowledge base. KB is satisfiable iff KB has a minimal Herbrand Model (denoted M_{kb}). A satisfiable KB of $GeoR_{bk}$ entails an $SWRLO$ ground facts F iff $M_{kb} \models F$. Of course as $GeoR_{bk}$ represents a set of backward query rules, any fact form entailment will not be added permanently to *SWSRLO*.

6.7.4 $GeoR_{bk}$ AND $GeoR_{fd}$ INTERLEAVED SEMANTICS

In this section the interleaved semantics of *SWSRL* is given based on the operational Emden-Kowalski operator T_p [262]. T_p is a fixed point operator which defines the meaning of a logic program to be the input-output relation which is the least model of the recursive transformation T_p associated with a program P. T_p is defined for *SWSRL* when using both forward ($Geo_R fd$) and backward ($GeoR_{bk}$) rulesets interleaved as follows.

Given the Horn logic program *SWSRL* and the set of ontological atoms (facts) $SWSRLO \subseteq SWRL$, the operator T_p can be defined as:

$$T_p(SWSRLO) =$$
$$\{\alpha | \alpha \leftarrow \beta_1, ..., \beta_n \textit{ is a clause in } GeoR_{fd} \textit{ and} \{\beta_1, ..., \beta_n\} \subseteq SWSRLO\}$$

That is, in the base step (and each step thereafter) facts (α, which can be error predicates from integrity rules) are inferred from clauses in $GeoR_{fd}$ if their body predicates $\{\beta_1, ..., \beta_n\}$ are contained in *SWSRLO*. However, in the interleaved mode, certain body predicates (spatial relations) are either taken directly from *SWSRLO*, or are evaluated from the set of backward rules $GeoR_{bk}$. Hence, the operation is defined as:

$$T_p(SWSRLO) = \{\alpha | \alpha \leftarrow \beta_1, ..., \beta_n \textit{ is a clause in } GeoR_{fd}$$
$$\textit{and } ((\{\beta_1, ..., \beta_n\} \subseteq SWSRLO) \vee (GeoR_{bk} \models \{\beta_1, ..., \beta_n\}))\}$$

Then, the base step, $T_p^0 = \emptyset$ where only explicit facts in $SWSRLO$ are present and no inferences (no α), is iterated as the sequence:

$$T_p^{i+1}(SWSRLO) = T_p(T_p^i(SWSRLO))$$

As T_p is monotone, the sequence $T_p^i(\theta)$ is also, and its union yields the least model (LM) of $SWSRL$:

$$LM(SWSRL) = \bigcup_{i=1}^{\inf} T_p^i(\emptyset)$$

Once a least model of $SWSRL$ has been found, $SWSRLO$ may contain positive and negative versions of the same error predicate. Hence, the knowledge base at this point in fed into the CLP^{-1} where conflicts between error predicates are resolved, if an error remains after this step $SWSRLO$ is unsatisfiable.

In the spatial rule engine, the forward engine is linked to the backward system through a predefined procedural attachment. The exact definition of the linkage, and how to maintain the semantics outlined above is discussed in detail in chapter 8.

6.8 SUMMARY

In this chapter a new spatial rule language was developed based syntactically on a Description Logic Program, but extended to include integrity and deduction rules, extra-logical procedural attachments, a simplified form of defeasible reasoning, interleaved rule execution and assumes standard logic programming semantics. The spatial rule language forms a new geo-ontology maintenance paradigm for the Semantic Web, which could be further extended in the future to include other logical constructs e.g. negation as failure, or other extra-logical procedural attachments, but with a change of semantics. In the next chapter we define a number of deduction and integrity rulesets for representing qualitative topological reasoning rules.

Chapter 7

Qualitative Spatial Reasoning in *SWSRL* with Spatial Rulesets

In this chapter we present the method of Topological Qualitative Spatial Representation and Reasoning (TQSRR) used within *SWSRL*. Qualitative Spatial Representation and Reasoning was introduced in Section 2.4.3. TQSRR is one of *SWSRLs* primary functions which maintains the integrity of any explicit topological qualitative spatial relations within the ontological component *SWSRLO*. Within this context we aim to encode and then reason with topological compositional inferences, such as those pre-computed in the RCC-8 composition table, declaratively using *SWSRL*.

Topological compositional inferences serve to derive new implicit relations from those explicit relations in *SWSRLO*, and to help maintain the topological consistency of regions in *SWSRLO* (in effect deciding Region satisfiability - RSAT). In particular, we aim to achieve closure of a set of relational constraints Θ under composition ($\otimes$) converse ($\smile$) and intersection ($\cap$) by encoding such operations as rules within *SWSRL* - in essence using a declarative representation of the original path-consistency and Revise algorithms of Mackworth and Vilain and Kautzs [169, 265]. Importantly, we aim to encode these compositional inferences within the Horn fragment of FOL, to conform with the semantics of *SWSRL*.

This chapter is organised as follows. Sections 7.1 and 7.2 present an in-depth discussion of how best to represent topological qualitative spatial reasoning within *SWSRL* for the purpose of integrity maintenance and knowledge deduction. Section 7.3 describes a set of base spatial rules encoded in *SWSRL* that performs qualitative spatial reasoning as discussed in section 7.1, along with a description of alternate rulesets (for the same purpose) using both backward and forward reasoning modes and a mixture of qualitative and quantitative relations.

7.1 REPRESENTING COMPOSITIONAL INFERENCES IN *SWSRL*

In this section we first describe how to best represent and reason with compositional inferences from the RCC-8 composition table in *SWSRL*, followed by a detailed discussion of how we perform consistency checking over the set of Regions in *SWSRLO* using a declarative representation of the procedural path-consistency algorithm.

A complete declarative representation of the path-consistency algorithm requires three rule types. Those that perform the closure of a set of relational constraints Θ under composition ($\otimes$) converse ($\smile$) , and refinement (by intersection $\cap$). These three steps are necessary to determine the consistency of each relational constraint in the set Θ.

The set Θ can be regarded as the set of topological relations in *SWSRLO*. Then to emulate the closure of Θ under composition in *SWSRL*, a set of compositional inference rules are required that are a direct representation of composition inferences in the RCC-8 composition table. However, the result of most compositions in table 2.1 produce disjunctive sets of base relations. Hence, in all but a few cases (where most of those involve the identity relation EQ), to represent all 64 possible RCC-8 compositional inferences, *SWSRL* needs to permit head disjunction. Therefore, as previously shown, a direct representation of all 64 compositional inferences in the full RCC-8 composition table (Table 2.1) requires disjunctive rules of the form:

$$R_1(x, y) \wedge R_2(y, z) \rightarrow Rh_1(x, z) \vee ... \vee Rh_n(x, z)$$

Where x, y and z are variables that represent regions in the universe U (or regions from *SWSRLO*), R and Rh are RCC-8 relations, and $n \geq 1$. For example the composition of EC(a,b) and EC(b,c) results in the set $\{DC, EC, PO, TPP, TPP^{-1}, EQ\}$ of disjunctive base relations:

$$EC(a,b) \wedge EC(b,c) \rightarrow DC(a,c) \vee EC(a,c) \vee PO(a,c) \vee TPP(a,c) \vee TPP^{-1}(a,c)$$

In order to deal effectively with disjunctive compositional inferences, either *SWSRL* should allow head disjunction (e.g. $SWSRL^{\vee}$), or we must find a way of dealing with the disjunction within a Horn framework. Adding disjunction to *SWSRL* places the language outside the highly tractable Horn LP fragment employed by *SWSRL* and as such requires more sophisticated semantics and reasoning engine. Hence we aim not to change the expressive capabilities of *SWSRL*. Instead we look at two different ways of partially or completely resolving this issue. The first tries to work with disjunctive inferences in a RCC-8 and a Horn framework (a naive representation), and the second provides a different encoding of the RCC relations based on the work presented by Schockaert [231] as described previously in chapter 2. Both are explained in more detail in the sections to follow.

7.2 Dealing with Disjunctive Compositional Inferences

7.2.1 Naive Representation

As integrity and deduction rules in *SWSRL* fit syntactically within the Horn fragment of FOL, they can not directly represent the head disjunction required to capture all compositional inferences. A naive representation would be to try and only represent those compositions from table 2.1 that resulted in definite inferences. That is to encode only a subset of the full 64 possible compositions that can be sufficiently represented by rules of the form:

$$R_1(x,y) \wedge R_2(y,z) \rightarrow Rh_1(x,z)$$

This would result in the representation of only 25 of the possible 64 compositional inferences (39%). Converse rules ($\smile$) on the other hand are easier to represent, only requiring definite Horn rules of the form:

$$R_1(a,b) \rightarrow Rh_1(b,a)$$

For example, $NTPP(x,y) \rightarrow NTTP^{-1}(y,x)$ and $EC(x,y) \rightarrow EC(y,x)$. Consequently, all converse rules can be directly represented in $SWSRL$.

However, using a smaller subset of compositional inferences is clearly not a complete representation of the RCC-8 composition table, and hence would not generate all possible compositional entailments - we would not achieve a full closure of the set of topological relations in $SWSRLO$ under composition. For example, in a consistency setting, the entailments of such rules could not determine that the following spatial configuration description (shown in figure 7.1) is inconsistent with the set of relational constraints Θ, where $\Theta = \{EC(A,B), NTPP(B,C), DC(A,C)\}$:

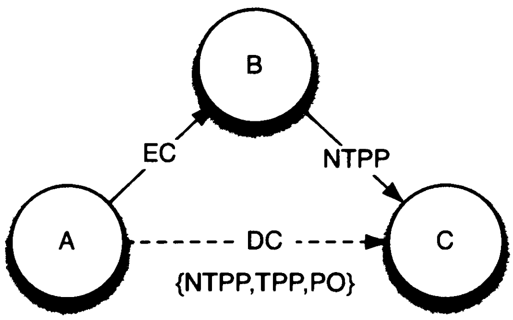


Figure 7.1: Example spatial configuration / constraint network with RCC-8 constraints Θ, where DC(A,C), EC(A,B) and NTPP(B,C) are explicit relations and {NTPP,TPP,PO} is a derived relation

The composition of $EC(A,B)$ with $NTPP(A,B)$ results in the disjunctive RCC-8 relation $\{NTPP, PO, TPP\}$, which does not subsume the relation $\{DC\}$ holding between regions A and C. However, we do not represent this compositional rule, hence this inconsistency would not be found. Evidently a better way to deal with disjunctive compositions is required.

7.2.2 SYNTACTIC RELATION GROUPING

Another possibility to represent disjunction in *SWSRL* involves syntactically grouping sets of disjunctive relations into one corresponding indefinite relation group denoted IDRG_{group}, where group is substituted for an RCC-8 relation that has a cardinality > 1 (an indefinite disjunctive relation set). This is similar to the approaches mentioned, but not thoroughly investigated, in [272, 53]. For example the disjunctive RCC-8 set $\{NTPP, PO, TPP\}$ would be represented as one relation $\text{IDRG}_{NTPP\_PO\_TPP}$. Following this approach requires all possible combinations* of disjunctive sets from the RCC-8 composition table to have a unique grouping (of which there are 11).

Interestingly this approach overcomes the limitations of the naive approach. The composition between regions A and C in figure 7.1 would now be derived, albeit as an IDRG relation. The fact that the existing relation is no longer consistent with the derived relation would need to be determined using procedural code that could intersect the string representation of the IDRG with the existing IDRG relation - such as could be provided by a builtin in an integrity rule.

However, by using IDRG, the language can not still represent complete compositional inferencing. Even though disjunctive relations are inferred and added to *SWSRLO*, these syntactically grouped relations are not further reasoned over by other composition rules. By syntactically grouping disjunctive RCC-8 relations into one generalised relation set we can not guarantee the resulting scene is path-consistent - we will refer to this problem as the RCC-8 Grouping Problem (RGP). RGP can be illustrated by considering the canonical spatial configuration (represented as a spatial network / constraint graph) in figure 7.2. We assume that A,B,C and D are specific regions, and the composition between the Regions A,B and B,C derives the disjunctive RCC-8 relation $\{EC, DC\}$. This disjunctive set under the approach introduced above, is represented as one grouped relation namely, IDRG_{EC_DC}. As a consequence, the inconsistency which should be detected by the compositions show in 7.1 is no longer derived - hence an integrity rule in *SWSRL* would not be tested for the composition between the regions A,C and C,D.

*Combinations and not permutations as the order of the set is not important

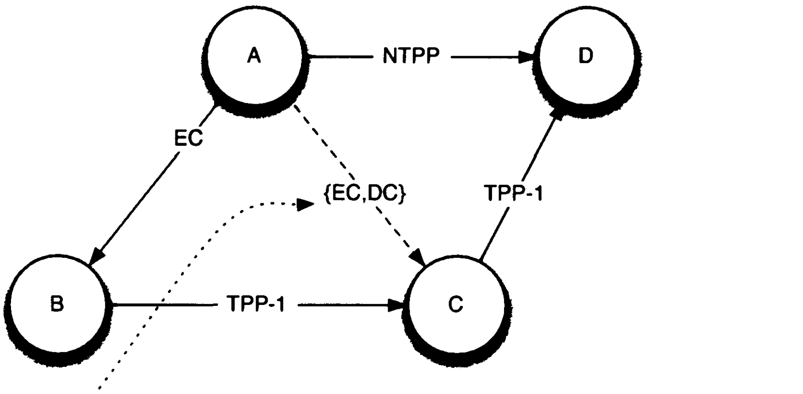


Figure 7.2: Illustration of the RCC-8 grouping problem

$$\begin{aligned}
EC(A,C) \otimes TPP^{-1}(C,D) &\rightarrow \{DC, EC\}(A,D) \\
DC(A,C) \otimes TPP^{-1}(C,D) &\rightarrow \{DC\}(A,D) \\
\{DC\}(A,D) \cup \{DC, EC\}(A,D) &= \{DC, EC\}(A,D) \\
\{DC, EC\}(A,D) \cap \{NTPP\}(A,D) &= \Theta \qquad (7.1)
\end{aligned}$$

In the next section we show how, by considering generalised RCC-8 relations and their associated compositional inferences, the RGP is overcome.

7.2.3 Generalised Topological Relations

In this section we present a representation of generalised RCC-12 relations in *SWSRL*. RCC-12 is discussed in detail in section 2.4.3.2. An overview of the desirable characteristics of RCC-12 for use in *SWSRL* is now given.

Compositional Inferences in RCC-12: Compositional inferences from the RCC-12 composition table (Table 2.3) can be captured natively within the Horn rule component of *SWSRL*. This is possible because, each compositional inference results in a definite inference (rules of the form $R_1(x,y) \wedge R_2(y,c) \rightarrow Rh(x,c)$). Furthermore, sets of RCC-12 relations are conjunctive as opposed to disjunctive, where *SWSRL* can not infer disjunctive relations, but can infer conjunctive sets of relations.

Representing the intersection operator in RCC-12: The execution of integrity checking rules (closing the set of relations under intersection - $\cap$) is order independent in RCC-12. More specifically, the order in which compositional inferences are made does not effect the consistency checking application of integrity rules. This is explained with reference to both integrity and deduction rules (both are defined in detail later in Section 7.3). The declarative semantics of *SWSRL* implies the order in which rules are run is, theoretically, unimportant hence no guarantee is placed on the order of rule execution (from the rule scheduler). When checking the consistency of the composition of a disjunction of RCC-8 base relations, all compositions must first be entailed and joined (by set union) together, before they can be intersected and hence tested for consistency*. This constraint is, however, relaxed when dealing with RCC-12 compositional inferencing. See the example composition below (which relates to the spatial network in figure 7.3):

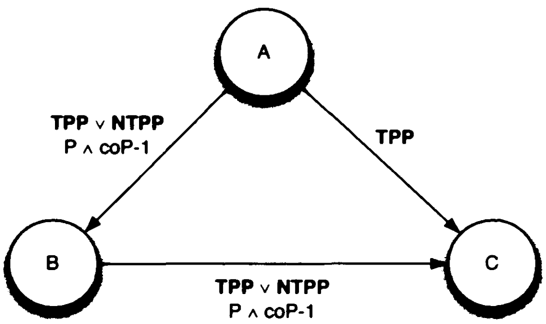


Figure 7.3: Simple spatial network illustrating the order independent reasoning of RCC-12

$$TPP(A,B) \otimes NTPP(B,C) \rightarrow NTPP(A,C) \cup \quad (7.2)$$

$$NTPP(A,B) \otimes TPP(B,C) \rightarrow NTPP(A,C) \cup \quad (7.3)$$

$$NTPP(A,B) \otimes NTPP(B,C) \rightarrow NTPP(A,C) \cup \quad (7.4)$$

$$TPP(A,B) \otimes TPP(B,C) \rightarrow TPP(A,C) \vee NTPP(A,C) \quad (7.5)$$

$$= \{NTPP, TPP\} \quad (7.6)$$

*This is because the composition of non base relations is computed as the union of the composition of the basic relations [221]

Only after the composition of all disjunctions holding between A,B and C, do we obtain the correct RCC-8 relation that holds between A and C (the disjunctive set $\{NTPP, TPP\}$). However as integrity rules are separate from deduction rules in $SWSRL$, if a spatial integrity rule had run after only the conclusion of compositional inference 7.2 (encoded as a deduction rule) a violation would occur. This violation does not however exist, as on the conclusion of compositional inference 7.5 the derived relation in 7.6 becomes consistent with that relation already holding between A and C. An obvious solution to the above would be to encode the complete composition as one compositional inference rule, for example:

$$(TPP(A,B) \vee NTPP(A,B)) \otimes (TPP(A,C) \vee NTPP(A,C)) \rightarrow (NTPP(A,C) \vee TPP(A,C)) \quad (7.7)$$

However this again involves head disjunction, and is outside $SWSRL$.

By considering RCC-12 relations and their corresponding compositional inferences, this problem is alleviated as each relation can be reasoned with independently as now shown:

$$P(A,B) \otimes P(B,C) \rightarrow P(A,C) = \{TPP, NTPP, EQ\} \quad (7.8)$$

$$P(A,B) \otimes coP^{-1}(B,C) \rightarrow coP^{-1}(A,C) = \{PO, NTPP, TPP, EC, DC\} \quad (7.9)$$

$$coP^{-1}(A,B) \otimes P(B,C) \rightarrow coP^{-1}(A,C) = \{PO, NTPP, TPP, EC, DC\} \quad (7.10)$$

Here, the order with which the compositional inferences and integrity rules are run does not matter. The addition of any new RCC-12 compositions (compositional inference 7.8 through to 7.10) only serves to refine (narrow) the set of all corresponding RCC-8 base relations, an inconsistency is only detected once an inconsistent state is reached. Therefore, the system will never reach a state where the addition of new knowledge could serve to validate an already detected inconsistency. Consequently, even though we have already shown that RCC-8 is not directly representable in $SWSRL$, the order irrelevances of reasoning with RCC-12 is again a useful characteristic when dealing with the non-deterministic nature of a declarative rule scheduler.

Relationship Generalisation: One final advantage of generalised relations is that they are helpful in the context of extracted qualitative topological relations from natural language scene descriptions (such as those used on the web). In nearly all cases it is not possible to extract exact knowledge of a regions relationship with another. For example, Roath is within the City of Cardiff (hence a part of, P relation), with no specific knowledge of whether it is a tangential or non-tangential proper part (TPP or $NTPP$ respectively).

In what follows we discuss the representation of RCC-12 in *SWSRL* and provide a complete and concrete realisation of RCC-12 in *SWSRL* for consistency checking, using integrity (intersection rules) and deduction rules (compositional and converse inferences).

7.2.4 USING RCC-12 IN *SWSRL*

Topological relations in *SWSRL* can either be one of the RCC-8 (including those using Egenhofers n-intersection terminology) relations or a conjunction of base RCC-12 relations. However, during spatial reasoning, all relations must be conjunctions of RCC-12 relations and hence RCC-8 relations must be mapped to RCC-12 relations. It is not always possible to map arbitrary distinctions of RCC-8 relations to RCC-12 relations. For example, from table 2.2, it is not possible to map the disjunctive RCC-8 relation $\{DC, EC, EQ\}$ into a conjunctive set of RCC-12 relations. However, If a mapping does exists then, as all RCC-12 relations are in the maximal tractable set $\mathcal{H}_8$, performing path consistency over these relations is sufficient to decide global consistency of the set of topological relations in *SWSRLO*. Table 7.1 shows the mapping between RCC-8 and RCC-12 relations. This mapping is derived, with a few subtle changes, from table 7.2, which defines each RCC relation in terms of the base dyadic relation C (as shown in [214]).

These subtle mapping differences can be enumerated as follows:

(a) Inverse mappings must be considered for RCC-12 base relations described in terms of b to a. For example, $NTPP$ maps to the RCC-12 relations $\neg P(b,a) \wedge NTP(a,b)$, but $P(b,a)$ must be mapped to its inverse from a to b e.g. $P^{-1}(a,b)$.

RCC-8 Relation	Conjunction of RCC-12 Base Relations
EC	$C \wedge DR$
DC	DC
EQ	$P \wedge P^{-1}$
PO	$O \wedge coP \wedge coP^{-1}$
$NTPP$	$NTP \wedge coP^{-1}$
TPP	$P \wedge coP^{-1} \wedge coNTP$
$NTPP^{-1}$	$NTP^{-1} \wedge coP$
TPP^{-1}	$P^{-1} \wedge coP \wedge coNTP^{-1}$

Table 7.1: RCC-8 to RCC-12 mappings

Name	Syntactic Definition
Disconnect from	DC(a,b) iff $\neg C(a,b)$
Part-of	P(a,b) iff $(\forall c \in U)(C(c,a) \Rightarrow C(c,b))$
Proper part-of	PP(a,b) iff $P(a,b) \wedge \neg P(b,a)$
Equal	EQ(a,b) iff $P(a,b) \wedge P(b,a)$
Overlaps	O(a,b) iff $(\exists c \in U)(P(c,a) \wedge P(c,b))$
Discrete from	DR(a,b) iff $\neg O(a,b)$
Partially overlaps	PO(a,b) iff $O(a,b) \wedge \neg P(a,b) \wedge \neg P(b,a)$
Externally connected	EC(a,b) iff $C(a,b) \wedge \neg O(a,b)$
Non-tangential part-of	NTP(a,b) iff $P(a,b) \wedge \neg(\exists c \in U)(EC(c,a) \wedge EC(c,b))$
Tangential proper part-of	TPP(a,b) iff $PP(a,b) \wedge \neg NTP(a,b)$
Non-tangential proper part-of	NTPP(a,b) iff $\neg P(b,a) \wedge NTP(a,b)$

Table 7.2: Topological Relations in RCC, where a,b and c denote regions in the universe of regions U

(b) EC is defined as the conjunction of C and $\neg O$. However $\neg O$ is not one of the RCC-12 base relations. $\neg O$ defines the RCC-12 relation DR, hence EC is mapped to the conjunction $C \wedge DR$ or the RCC-12 relation $\{C, DR\}$.

(c) DC is both in RCC-8 and RCC-12 therefore no mapping is required.

(d) TPP is defined as the conjunction of PP and $\neg NTP$. PP is not an RCC-12 base relation. PP is defined as P and $\neg P^{-1}$, and hence TPP is mapped to the conjunction $P \wedge coP^{-1} \wedge coNTP$.

Clearly intersecting arbitrary numbers and types of RCC-12 relations allows us to map more than just those RCC-8 base relations shown in Table 7.1. For example the relation $\{NTPP, TPP, EQ\}$ can be mapped or generalised directly as P in RCC-12 etc.

Crucially, Table 2.2 shows the reverse mapping of RCC-12 to RCC-8 for $NTPP$ involves only the relation NTP, whereas in Table 7.1 $NTPP$ maps to two RCC-12 relations. This reverse mapping is based on the knowledge that $NTP(x, y) \equiv NTPP(x, y)$, and has been proved in [232] based on euclidean realisable geometries [217](the type of geometries assumed in this thesis, and which is compatible with most GIS's).

Using RCC-12 to overcome the RGP problem: Using the framework of relations based on RCC-12 alleviates the RGP. Consequently, *SWSRL* can entail all compositional inferences, in effect closing a set of relational constraints Θin RCC-12 under composition (⊗). By including converse rules (see Section 7.3) we can also close Θ under converse ($\smile$). The only reasoning task left is to close Θ under intersection (∩), which is achieved using integrity rules as described in section 7.3.2.

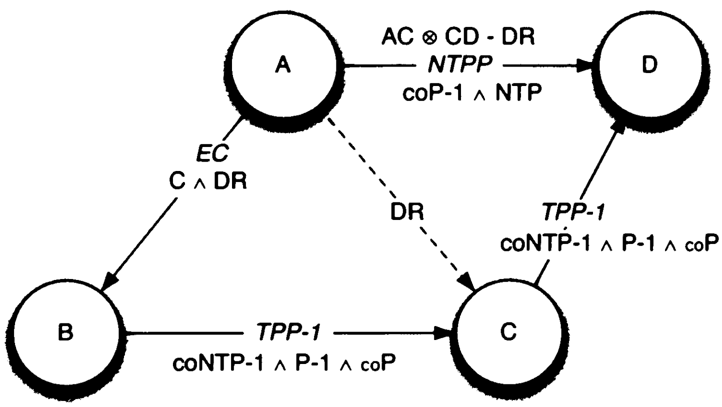


Figure 7.4: Illustration of how to overcome the RCC-8 grouping problem using RCC-12 inferences

As an example, figure 7.4 illustrates a spatial network, where the nodes A,B,C and D are concrete spatial regions and both RCC-8 and their mapped RCC-12 relations are shown on each edge.

DR(A,C) was derived from $\{C, DR\}$(A,B) and $\{coNTP^{-1}, P^{-1}, coP\}$($B$,$C$). $\{coP^{-1}, NTP\}$(A,D) was then determined by composing the derived relation DR(A,C) (found in the previous step) with the explicit relations $\{coNTP^{-1}, P^{-1}, coP\}$($C$,$D$). The disjunctive set of RCC-8 base relations that correspond to the derived RCC-12 relation $\{DR\}$ between A and D, does not intersect with the explicit disjunctive set of RCC-8 base relations corresponding to the RCC-12 relation $\{coP^{-1}, NTP\}$ holding between A and D, as illustrated in the following equations.

$$
\begin{aligned}
DR(A,D) &= \{EC, DC\}(A,D) \\
coP^{-1}(A,D) &= \{PO, NTPP, TPP, EC, DC\}(A,D) \\
NTP(A,D) &= \{NTPP\}(A,D) \\
\{EC, DC\}(A,D) \cap \{PO, NTPP, TPP, EC, DC\}(A,D) & \\
\cap\{NTPP\}(A,D) &= \Theta
\end{aligned} \tag{7.11}
$$

Importantly, the inconsistency between A and D would not have been determined using the relational grouping strategy in Section 7.2.2. That is, the derived disjunctive relation between A and C ($\{EC, DC\}$) would not have triggered a further inference to derive $\{EC, DC\}$ (or DR) between A and D.

7.3 Spatial Rule Sets (Space Laws)

In this section we explore different possible constructions of $SWSRL$ rulesets for the backward deduction (the set $GeoR_{bk} \subset SWSRL$), forward deduction (the set $GeoR_{fdD} \subset SWSRL$) and forward integrity (the set $GeoR_{fdI} \subset SWSRL$) subsets of $SWSRL$. Different combinations of these sets form what we denote *Space Laws*. *Space laws* represent a sound set of predefined rules in $SWSRL$, defined as part of this thesis, to test topological consistency of the relations in $SWSRLO$. Such rules are all level 0, the lowest level ruleset that can not be altered or overridden. Here we note that, as the language is capable of representing arbitrary user-defined rules, this section does not explore all possible types of rulesets that can be constructed in $SWSRL$. Within the scope of this work we only construct *space laws* to reasoning with and maintain topological spatial relationships.

At this stage, *space laws* are constructed for the purpose of deriving new topological relations from those explicit (raw relations) in $SWSRLO$, and to decide the consistency of the topological relations in $SWSRLO$. Furthermore, *space laws* can be used (in the case of $GeoR_{bk}$) to aid user defined rules. As a first step, we aim to closely mimic a typical path-consistency algorithm, to decide if the entire $SWSRLO$ geo-ontology is consistent or not. Localisation of inconsistent relations using statistical techniques is described later in chapter 8. For this, rulesets need to be defined for the following basic tasks:

(a) Derivation of implicit topological relations using composition rules ($\otimes$) and converse rules ($\smile$) - in effect computing the closure of a set of relational constraints under composition and converse.

(b) Deciding the topological consistency of *SWSRLO* (RSAT) requiring, in addition to composition and converse rulesets, a ruleset that simulates the intersection operation ($\cap$) over the relations of *SWSRLO*. Composition and converse rules are represented as derivation rules. Intersection rules are represented as integrity rules and entail errors when an inconsistency is found.

Task a can be implemented in both query mode (backward chaining) or continuous inferencing mode (forward chaining). To continuously monitor *SWSRLO*, a forward chaining approach will be adopted for task b. Task b can also be run using an interleaved mode of reasoning, combining forward chaining integrity rules that perform intersections of RCC-12 relations, with backward chaining deduction rules for querying entailed RCC-12 relations between regions.

There are a number of possible ways to combine forward and backward rulesets to show different features of *SWSRL*. Hence, for the reminder of this thesis the set of *SpaceLaws* can be represented by one of the following combinations of spatial rulesets* :

$$
\begin{aligned}
(1)\quad & SpaceLaws \equiv && F_D^{RCC12}\\
(2)\quad & SpaceLaws \equiv && F_I^{RCC12}\\
(3)\quad & SpaceLaws \equiv && F_{<D,I>}^{RCC12}\\
(4)\quad & SpaceLaws \equiv && FBi_{interleaved}^{RCC12}
\end{aligned}
$$

Where:

- F_D^{RCC12} is the set of all forward deduction rules based on the RCC-12 composition table. This set derives full compositional closure of the generalised topological relations continuously as new topological facts are added

*These possibilities represent an exhaustive list of *space laws*, and as such should not be combined in different ways to $SWSRLO$. Alternatively, $GeoR_{bk}$ is a similar ruleset but using backward deduction rules. $GeoR_{bk}$ can be interleaved with forward rules, and serves only to derive topological relations. $GeoR_{bk}$ is further split into $GeoR_{bk}^{standard}$ which only contains qualitative topological spatial reasoning, and $GeoR_{bk}^{hybrid}$ which additionally adds a call to the LSS to compute topological relations from geometries associated to features.

- F_I^{RCC12} is the set of integrity rules that replicate the intersection operation $\cap$ of the path consistency Revise function, as derived from the RCC-12 composition table. By intersecting RCC-12 relations in $SWSRLO$, this ruleset can maintain the integrity of immediate topological relations in the ontology. Hence, this ruleset can decide the consistency of an $SWSRLO$ geo-ontology that is already closed under composition or converse, either explicitly or by using F_D^{RCC12}.
- $F_{<D,I>}^{RCC12}$ is the complete set of forward integrity and deduction rules. This ruleset guarantees path-consistency and global consistency over the set of topological relations in $SWSRLO$, provided the initial set of RCC-8 relations can be mapped to conjunctive sets of RCC-12 relations and hence are in the maximal tractable subset $\mathcal{H}_8$.
- $FBi_{interleaved}^{RCC12}$ is a ruleset for Interleaved reasoning. This ruleset can either use standard backward deduction rules, or hybrid (which include a link to the LSS) backward deduction rules. Unlike $F_{<D,I>}^{RCC12}$, thanks to the interleaving of backward and forward rules, no deductions are added to $SWSRLO$.

Clearly other user defined rulesets are possible, for example the set of user defined deductions interleaved with backward generalised topological deduction rules (denoted $FBd_{interleaved}$). However such rulesets are not consider *Space Laws* and are given a rule level greater than 0. In the sections to follow, the exact formalisation of each of the four *space law* rulesets is described.

7.3.1 F_D^{RCC12} and $GeoR_{bk}$ - RCC-12 Composition ($\otimes$) and Converse Rules ($\smile$)

As previously described, in order to mimic the Revise and path-consistency algorithms, we need to represent $SWSRL$ rules that capture compositional inferences from the RCC-12 composition table (composition rules $\otimes$), rules that derive converse relations (converse rules $\smile$) and rules that determine whether relations have a valid intersection (intersection rules $\cap$). In this section we look at the representation of composition and converse rules in both forward (F_D^{RCC12}) and backward ($GeoR_{bk}$) rulesets. Here, composition rules are derived directly from the RCC-12 composition Table 2.3. Similarly converse rules are derived directly from Table 7.3. Forward RCC-12 deduction and converse rulesets are henceforth denoted $Fd_{RCC^{12}}$ and $Fd_{RCC^{12}\smile}$ respectively. Backward RCC-12 deduction and converse rulesets are henceforth denoted $B_{RCC^{12}}$ * and $B_{RCC^{12}\smile}$ respectively.

The set $B_{RCC^{12}}$ of backward chaining RCC-12 composition rules can not be reasoned with in an logic programming engine that does not support tabling (i.e. the standard ISO GNU Prolog standard engine) due to the recursive nature of some compositional inferences. Typical forward engine implementations can handle recursive rules natively.

As already discussed, composition and converse rules work over RCC-12 relations. Hence, forward and backward rulesets, denoted $F_{map^{\rightarrow}}$ and $B_{map^{\rightarrow}}$, are needed to map any RCC-8 relations in $SWSRLO$ to a conjunction of RCC-12 base relations. This is shown in table 7.1 e.g. $EC(a,b) \rightarrow C(a,b) \wedge DR(a,b)$.

RCC-12 Relation	Converse RCC-12 Relation
C	C
DR	DR
P	P^{-1}
P^{-1}	P
DC	DC
coP	coP^{-1}
coP^{-1}	coP
O	O
NTP	NTP^{-1}
NTP^{-1}	NTP
coNTP	$coNTP^{-1}$
$coNTP^{-1}$	coNTP

Table 7.3: RCC-12 base relations and their converse

Unlike forward deduction rules, the standard set* $GeoR_{bk}^{standard}$ of all backward

*All backward rules are deduction rules, hence the omission of the d prefix from $B_{RCC^{12}}$

*An extended set of $GeoR_{bk}$ rules that include calls to the LSS is shown in Section 7.3.1.1

deduction rules, can be interleaved into arbitrary forward deduction rules (including user defined) to determine the truth of a base RCC-8 relation. As a result, an additional ruleset, denoted $B_{map^{\leftarrow}}$ (the reverse of $B_{map^{\rightarrow}}$), is needed that contains a reverse mapping from RCC-12 relations to their equivalent RCC-8 base relation. These mappings are described in section 7.2.4 based on the reverse mapping in table 2.2, for example $C \wedge DR$ maps to the RCC-8 base relation EC.

Complete Ruleset: The complete backward ruleset $GeoR_{bk}$ is an amalgamation of the following:

$$GeoR_{bk}^{standard} \equiv B_{map^{\rightarrow}} \cup B_{map^{\leftarrow}} \cup B_{RCC^{12}} \cup B_{RCC^{12\frown}}$$

The ruleset $GeoR_{bk}$ is fixed and can not be augmented with user defined rules. The complete set of forward deduction rules, F_D^{RCC12}, is an amalgamation of the following:

$$F_D^{RCC12} \equiv Fd_{map^{\rightarrow}} \cup Fd_{RCC^{12}} \cup Fd_{RCC^{12\frown}}$$

Example rules for this set are shown in Appendix B.1.1.

7.3.1.1 Hybrid $GeoR_{bk}^{hybrid}$ Rule Set

The hybrid backward ruleset is an extension to the existing $GeoR_{bk}^{standard}$ ruleset to include procedural attachments that call the LSS to compute one of the RCC-8 base relations on demand. In particular this ruleset extends the set of rules $B_{map^{\leftarrow}}$ to include external geo-computation predicates (themselves procedural attachments). The extended set is denoted $B_{map^{\bowtie\leftarrow}}$, where $\bowtie$ represents the joining of 'quantitative' rules with 'qualitative' rules.

As a result hybrid backward deduction rules are the amalgamation of the following four rulesets:

$$GeoR_{bk}^{hybrid} \equiv B_{map^{\rightarrow}} \cup B_{map^{\bowtie\leftarrow}} \cup B_{RCC^{12}} \cup B_{RCC^{12\frown}}$$

Depending on whether both qualitative and quantitative, or just qualitative backward rules are used, the set of all backward rules is equal to $GeoR_{bk}^{hybrid}$ or $GeoR_{bk}^{standard}$

respectively, more formally:

$$GeoR_{bk} \equiv GeoR_{bk}^{hybrid} \vee GeoR_{bk}^{standard}$$

7.3.2 F_I^{RCC12} - RCC-12 INTEGRITY RULES ($\cap$)

Compositional Integrity: Forward integrity rules complement forward deduction rules in that each integrity rule is a replication of a deduction rule, but with two important differences. Firstly, to properly mimic the Revise function, each rule evaluates not only the relations between region pairs $< A, B >$, $< B, C >$, but also the relation(s) between the pair $< A, C >$. That is, for each composition in the composition table, an integrity rule is defined that tests that the relation(s) holding between A and C, would not result in an inconsistency when combined with the composed relations between $< A, B >$ and $< B, C >$. Secondly, integrity rules infer error predicates that describe inconsistencies in $SWSRLO$.

As an example, take the composition : $P^{-1}(A, B) \otimes NTP(B, C)$, which results in the RCC-12 relation $O(A, C)$. If the existing set of RCC-12 relations between the regions A and C do not share any of the same RCC-8 relations as the RCC-12 relation O, then the relations between A and C are inconsistent. In this case, there are two RCC-12 relations that do not share any of the same RCC-8 relations as O, namely DR and DC. Consequently, if either DR or DC holds between regions A and C then the relation(s) are inconsistent. More formally:

$$P^{-1}(a, b) \wedge NTP(b, c) \wedge (DR(a, c) \vee DC(a, c)) \rightarrow error(...) \tag{7.12}$$

Which is converted by LTAO to two separate rules.

$$P^{-1}(a, b) \wedge NTP(b, c) \wedge DR(a, c) \rightarrow error(...) \tag{7.13}$$

$$P^{-1}(a, b) \wedge NTP(b, c) \wedge DC(a, c) \rightarrow error(...) \tag{7.14}$$

The complete set of integrity rules can then be constructed for each composition by using Table 7.4, which shows the correspondence between each RCC-12 relation and their non-intersecting RCC-12 relations.

Converse Integrity: The same principle applies for converse rules. Take for example the converse of the relation $O(A, B)$ which is $O(B, A)$. Neither $DR(B, A)$ or $DC(B, A)$ should then hold between A and C - again as shown in table 7.4.

$$O(b, a) \wedge DR(b, a) \rightarrow error(...) \tag{7.15}$$

$$O(b, a) \wedge DC(b, a) \rightarrow error(...) \tag{7.16}$$

RCC-12 Relation	Non-intersecting Relations
C	DC
coP	$P \vee NTP$
P	$DR \vee coP \vee DC \vee NTP^{-1}$
P^{-1}	$DR \vee coP^{-1} \vee NTP \vee DC$
NTP	$DR \vee coP \vee P^{-1} \vee coNTP \vee DC \vee NTP^{-1}$
O	$DR \vee DC$
NTP^{-1}	$DR \vee P \vee coP^{-1} \vee NTP \vee coNTP^{-1} \vee DC$
$coNTP$	NTP
$coNTP^{-1}$	NTP^{-1}
DR	$P \wedge NTP \vee O \vee P^{-1} \vee NTP^{-1}$
DC	$C \vee P \vee NTP \vee O \vee P^{-1} \vee NTP^{-1}$
coP^{-1}	$P^{-1} \vee NTP^{-1}$

Table 7.4: RCC-12 Relations and Their Non-intersecting Relations

The complete set of forward integrity rules F_I^{RCC12}, is again composed of RCC-8 to RCC-12 mapping rules $F_{map^{\rightarrow}}$, along with two new rulesets; $Fi_{RCC^{12}}$ and $Fi_{RCC^{12\frown}}$. The ruleset $Fi_{RCC^{12}}$ represents the integrity checking equivalent of compositional deduction rules in $Fd_{RCC^{12}}$. Similarly the set $Fi_{RCC^{12\frown}}$ is the integrity checking equivalent of converse deduction rules in $Fd_{RCC^{12\frown}}$. Example rules are shown in Appendix B.1.2.

As a result the full set of RCC-12 integrity rules, denoted F_I^{RCC12}, is then defined as:

$$F_I^{RCC12} \equiv F_{map^{\rightarrow}} \cup Fi_{RCC^{12}} \cup Fi_{RCC^{12\frown}}$$

7.3.3 $F^{RCC12}_{<D,I>}$ - Combined Integrity and Deduction Rule Set

The set F_I^{RCC12} is paired with the set F_D^{RCC12} to guarantee full path-consistency. Effectively closing the set of topological relations in *SWSRLO* under composition ($\otimes$) and converse ($\smile$) using the ruleset F_D^{RCC12}, and intersection ($\cap$) using the ruleset F_I^{RCC12}. Consequently for complete reasoning, RCC-12 topological reasoning rules are compiled into complimentary rule pairings <integrity$_n$,deduction$_n$>, where n is an index to a deduction rule in F_D^{RCC12} and matching integrity rule in F_I^{RCC12}, forming the set $F^{RCC12}_{<D,I>}$:

$$F^{RCC12}_{<D,I>} \equiv F_I^{RCC12} \cup F_D^{RCC12} \tag{7.17}$$

An example integrity deduction rule pairing is shown in Appendix B.1.3.

7.3.4 $FBi^{RCC12}_{interleaved}$ - Forward and Backward Interleaved Rule Set

In *SWSRL* backward rules can be interleaved into the execution of forward rules. That is, body predicates in forward rules can be determined by querying a set of backward rules. For example, take the forward deduction rule:

$$EC(a,b) \wedge EQ(b,c) \rightarrow EC(a,c) \tag{7.18}$$

Then, the spatial relation predicates EC and EQ can be determined from the set of backward rules $GeoR_{bk}$. The following is a small example subset of $GeoR_{bk}$ showing two rules. The first derives that a touches b if b touches a (a converse symmetric relation). The second derives a and c to be equal if a is equal to b, and b is equal to c (by composition):

$$EC(a,b) \leftarrow EC(b,a) \tag{7.19}$$

$$EQ(a,c) \leftarrow EQ(a,b) \wedge EQ(b,c) \tag{7.20}$$

Of course for this to work the ruleset $GeoR_{bk}$ can not be empty. Interleaving arbitrary, user defined forward deduction rules ($F_D^{user} \subseteq GeoR_{fdD}$) or forward integrity ($F_I^{user} \subseteq GeoR_{fdI}$) rules with backward $GeoR_{bk}$ is then achieved by simply combining their sets (how to interleave them in implementation is described in chapter 8). Hence, the set $FBd_{interleaved}$ combines user defined deduction rules with backward deduction rules. Similarly the set $FBi_{interleaved}$ combines user defined integrity rules with backward deduction rules, more formally:

$$FBd_{interleaved} \equiv GeoR_{bk} \cup F_D^{user}$$
$$FBi_{interleaved} \equiv GeoR_{bk} \cup F_I^{user}$$

Interestingly, as the set $GeoR_{bk}$ contains inherent mappings too and from RCC-12 relations used for topological spatial reasoning, both interleaved rulesets do not require forward mapping rules (the set $F_{map\rightarrow}$).

Interleaving RCC-12 forward integrity rules: When interleaving the forward integrity rules set $F_{<D,I>}^{RCC12}$ with the backward ruleset $GeoR_{bk}$, the backward ruleset $GeoR_{bk}$ replaces the forward ruleset F_D^{RCC12}. That is, composition and converse rules are provided by the backward ruleset instead of the forward deduction ruleset, hence leaving the forward integrity ruleset F_I^{RCC12}. In a slight change from interleaving user defined rules, RCC-12 integrity rules are interleaved on RCC-12 relations. For example, assuming the following integrity rule:

$$DC(a,b) \wedge C(b,c) \wedge P^{-1}(a,c) \rightarrow error()$$

Backward calls are made to determine the truth of the RCC-12 predicates DC, C and P^{-1}.

Interleaving forward integrity rules with backward deduction rules means inferences are not added back to $SWSRLO$. That is, path-consistency can be determined over the set of topological relations in $SWSRLO$, without adding all relational entailments back to $SWSRLO$. As a result, the size of the core $SWSRLO$ will not increase - the number of topological relations in $SWSRLO$ remains fixed.

Overall, the interleaved RCC-12 topological integrity ruleset is formed by:

$$FBi_{interleaved}^{RCC12} \equiv GeoR_{bk} \cup F_I^{RCC12} \tag{7.21}$$

Where $GeoR_{bk}$ is either the standard ruleset $GeoR_{bk}^{standard}$, or the hybrid backward ruleset $GeoR_{bk}^{hybrid}$ if both qualitative and quantitative topological reasoning is required.

7.4 SUMMARY

In this chapter we showed how a composition table for generalised topological relations (RCC-12) can be represented directly in $SWSRL$, and how this representation overcame the deficiencies that naive approaches had in representing the original RCC-8 composition table. A complete set of *space laws* was defined which provided an alternative, declarative representation of the path-consistency and Revise algorithms. In the next chapter the spatial rule engine for $SWSRL$ is described in detail.

Chapter 8

SWSRL Spatial Rule Engine

In this chapter we describe the spatial rule engine used to reason with the newly proposed rule language *SWSRL*. This includes a description of the base rule engine implementation, along with all necessary extensions required to conform to the features and semantics of *SWSRL*. We also describe two different error localisation techniques which help to determine the source of inconsistencies in topological relations.

The structure of this chapter is as follows. Section 8.1 motivates and overviews the technical implementation of the rule engine. Section 8.2 then defines a transformation function that transforms axioms and rules in *SWSRL* into an RDF triple representation suitable for the spatial rule engine. Sections 8.3 and 8.4.1 describe the implementation of two prominent features of *SWSRL*, that is dealing with default reasoning assumptions using the CLP^{-1} extension, and the efficient interleaving of forward and backward reasoning modes. Section 8.5 then describes the important procedural attachments in *SWSRL*. Finally section 8.6 describes two techniques that 1) help to determine the source of any topological inconsistency in *SWSRL*, and 2) suggest relations that can hold between any two regions where an inconsistency has been detected.

8.1 Existing Rule Engine Technology

A Rule based paradigm should have an established and sound declarative semantics. A declarative semantics provides the mathematical theory to derive answer sets (logical inferences) from a set of facts and rules, independently of the order in which they are executed and the mode of reasoning used i.e. forward or backward. From the standpoint of rule engines, there are two principle contrasting approaches to the execution of rulesets, namely forward chaining and backward chaining.

Forward chaining approaches are typical of production systems, which are themselves similar to Event Condition Action rules in active databases [176, 203]. Naive execution of a forward chaining production system would check every predicate (denoted patterns in production systems) in the body of every rule against every fact for each iteration of the engine. Iteration then stops when no more facts can be asserted (a global fixed point). This is known as a brute force or *rules finding facts* approach. The naive approach has a general computational complexity per iteration of $O(RF^P)$, where R is the number of rules, P is the average number of patterns or predicates per rule body, and F is the number of facts in the knowledge base. Clearly such an exponential approach would become intractable for large rule sets and knowledge bases. At the end of the 1970's and the early 1980's OPS5 (Official Production System)[75] was developed by Charles Forgy. OPS5 is the forerunner to most modern production rule based expert systems, where its power can be attributed to Forgy's Rete algorithm for efficiently matching, scheduling and executing production rules. Rete employs a rule set and a working memory of facts (the knowledge base). Rete then utilises two characteristics of working memory and rules known as temporal redundancy and structural dependency. Temporal redundancy exploits the knowledge that not many facts will change in working memory between iterations, hence only those rules that match to newly asserted facts in working memory need to be re-checked. Structural dependency allows rules to share pattern matching nodes if they share similar body predicates, this greatly reduces the storage overheads required to store patterns and their list of matching facts (for a more in-depth overview of the Rete algorithm see Appendix A.3). The computational complexity per iteration of the Rete algorithm is $O(RFP)$ [91] linear complexity*, which is a significant improvement over the naive approach.

Backward chaining, a top down approach to reasoning, is the more prevalent of the two and was first developed by J.A Robinson [10] and R. Kowalski and

*Although more in-depth analysis of the computational complexities of Rete, including a framework to compute average case complexity for rule sets and facts, and the complexity of adding, matching and removing tokens from the Rete have been shown in [2] and [279] respectively implemented into the now *de facto* logic programming system PROLOG [38]. Backward chaining is based on SLD † or the improved, to deal with recursive rules, SLD+SLG‡ [249] resolution principles developed using the Warren Abstract Machine (WAM)§ and employed in the XSB logic engine.

Implicit knowledge (entailments) are queried for in a backward chaining system, whereas in a forward chaining system the knowledge base is continually monitored and new facts repeatedly produce new entailments. Within $SWSRL$, both forward and backward reasoning modes are defined. Integrity rules are encoded in a forward chaining system, such that the continual match-resolve-act cycle identifies any new errors or deductions as and when the ontology is updated. By placing integrity rules in the backward system, each integrity violation would have to be queried for, which is less desirable. Deduction rules are encoded in the forward system, apart from the set of RCC-12 topological reasoning rules, which can be represented in either the forward or backward system. If represented in the backward system, the rules can then be interleaved with further deduction or integrity rules in the forward system.

Since the advent of the Semantic Web a number of complete reasoning systems have been developed as shown in Appendix A.5. In this thesis we have chosen Jena2 [175] as a base implementation system as it contain Java versions of both Rete and XSB rule engines, and provides an API for creating, accessing and manipulating RDF ontologies. Hence, Jena2 delivers a suitable foundation from which to base further extensions necessary to reason with $SWSRL$. In overview, these extensions are:

- The Prioritised Conflict Handling Engine (PCHEng), to handle prioritised defeasible integrity rules using the proposed simplified Courteous Logic extension CLP^{-1}.
- A procedural mechanism to allow forward rules to interleave with backward rules.
- Implementation of all necessary procedural attachments or builtins. Each

†Linear Resolution with Selection function for Definite clauses
‡Linear Resolution with Selection Function for General Logic Programs
§Named after its creator David Scott Warren

builtin is registered with Jena2's reasoning engine and implemented as procedures in Java.

- Error localisation methods that help to identify the source of a topological error, and suggest possible rectification.

8.2 *SWSRL* TO RDF GEO-ONTOLOGY CONVERSION

In chapter 6 we introduced and formally defined our geo-ontology paradigm *SWSRL* for representing and reasoning over geo-ontology axioms and rules. Chapter 6 also showed a mapping function $\mathcal{T}$ that can convert OWL ontologies into *SWSRL* ontologies. However, for implementation purposes, it is necessary to convert the *SWSRL* logical syntax into an RDF triple syntax compatible with the Semantic Web enabled spatial rule engine (based in Jena2). This further mapping is now described.

8.2.1 *SWSRL* TO RDF MAPPING

At its core Jena2 works off an RDF graph/triple base ontology representation. As the spatial rule engine is an extension to Jena2 it must use the same RDF triple based rule format. Consequently, a further transformation function $\mathcal{T}_{jena}$ is defined that takes an *SWSRL* set of rules and ontological axioms and transforms them into a semantically equivalent RDF triple based representation. In overview, the transformation function $\mathcal{T}_{jena}$ performs the following:

(a) Transforms forward and backward rules, including ontological facts in *SWSRLO*, into RDF triples.

(b) If a suitable set of backward rules exist, transforms any RCC-8 or RCC-12 predicate into an **ask** builtin for interleaved execution.

(c) Adds dummy nodes to all rules that contain an RCC-8 or RCC-12 predicate in order to maintain semantic cohesion under incremental update as described in section 8.3.

$\mathcal{T}_{jena}$ is a transformation function that maps all constructs in *SWSRL* of the form: $(H \leftarrow B)$, $(B \rightarrow H)$ or $(\rightarrow H)$ into RDF triples compatible with the internal implementation of Jena2's rule engines. More formally, $\mathcal{T}_{jena}$ is defined as:

$$\begin{array}{rcl}
\mathcal{T}_{jena}(GeoR_{fd}(B \rightarrow H)) & \longrightarrow & Rb(B) \rightarrow Rh(H) \\
\mathcal{T}_{jena}(GeoR_{bk}(H \leftarrow B)) & \longrightarrow & Rh(H) \leftarrow Rb(B) \\
\mathcal{T}_{jena}(GeoR_{fd}(\rightarrow H)) & \longrightarrow & Rh_{fact}(H) \\
Rb((B \wedge G)) & \longrightarrow & Rb(B) \wedge Rb(G) \\
 & - & \\
Rb(bt(t_1, ..., t_n)) & \longrightarrow & bt(t_1, ...t_n) \quad \text{(where n} \geq 1) \\
Rb(p(t_1)) & \longrightarrow & triple(t_1 \text{ rdf:type } p) \\
Rb(p(t_1, t_2) & \longrightarrow & triple(t_1, p, t_2) \\
Rb(p_{rsp}(t_1, t_2)) & \longrightarrow & \begin{cases} \text{If GeoR}_{bk} \neq \theta & \text{ask}(t_1, \text{RSP}, t_2) \wedge ((t_1, DC, t_3) \vee (t_1, \\ & (t_1, O, t_3) \vee (t_1, P, t_3) \vee (t_1, coP, t \\ & (t_1, coP^{-1}, t_3) \vee (t_1, NTP, t_3) \vee (t_1, co \\ & \vee(t_1, coNTP^{-1}, t_3) \vee (t_1, DR, t_3) \vee (t_1, N \\ & (t_1, P^{-1}, t_3)) \\ \text{otherwise,} & triple(t_1, p_{rsp}, t_2) \end{cases} \\
Rh_{fact}(p(t_1, t_2) & \longrightarrow & triple(t_1, p, t_2) \\
Rh_{fact}(p(t_1)) & \longrightarrow & triple(t_1 \text{ rdf:type } p) \\
Rh(bt(t_1, ..., t_n)) & \longrightarrow & bt(t_1, ...t_n) \quad \text{(where n} \geq 1) \\
Rh(p(t_1)) & \longrightarrow & triple(t_1 \text{ rdf:type } p) \\
Rh(p(t_1, t_2) & \longrightarrow & triple(t_1, p, t_2)
\end{array}$$

Where B and G are body predicates and H is a head predicate. If B and G are body predicates, so are $(B \wedge G)$, $(p(t_1))$, $(p(t_1, t_2))$,$(bt_{gen}(t_1, t_2))$,$(\texttt{valid}(t_1))$ and $(bt_{rsp}(t_1))$.

Mapping of backward rules in $GeoR_{bk}$ only differs in syntax to the mapping of forward rules from $GeoR_{fd}$. $p(t_1, ..., t_n)$ denotes a logical predicate p, with terms $t_1, ...t_n$ where $n = 1$ for unary predicates or classes and $n = 2$ for binary/relational predicates. A term is either a constant or a variable, and directly maps from *SWSRL* syntax to Jena triple syntax i.e. a variable/constant in *SWSRL* remains a variable/constant in Jena's triple format. As is standard of semantic web languages, within *SWSRL* each ground / constant predicate and term is referenced with a full, expanded URI identifier, or by a qualified namespace and local name e.g. *rdf*: where *rdf* is expanded to `http://www.w3.org/1999/02/22-rdf-syntax-ns#`. Consequently, each term and predicate is a resource that could be references to other imported *SWSRLO* geo-ontologies.

As described, *SWSRLO* is restricted to unary (class) or binary (property) logical predicates - the DL fragment with two free variables. Hence, t_1 and t_2 represent the subject and object of a triple, and p the predicate. Builtins are arbitrary where terms represent arguments to the invoked procedural code (which are always ground when the builtin is evaluated). A *bt* prefix denotes a builtin, and p_{rsp} denotes a reserved spatial predicate, one of the RCC-8 or RCC-12 base relations. The mapping of p_{rsp} differs depending on the existence of a backward rule set. If the backward rule set is empty then the mapping is straight forward, otherwise the **ask** predicate and dummy predicates are added during the mapping - this situation is described in detail in section 8.3.

8.3 INTERLEAVED EXECUTION EXTENSION IMPLEMENTATION

An important feature of *SWSRL* is to employ topological queries evaluated by the backward engine over the geo-ontology component *SWSRLO*, during the course of forward inferencing - interleaved execution of forward and backward systems. More specifically, each RCC-8 or RCC-12 base relation predicate in a forward rule is resolved using the set $GeoR_{bk}$ of backward rules on the fly. Where the set $GeoR_{bk}$ may either be the standard rule set with only qualitative spatial reasoning rules $GeoR_{bk}^{standard}$, or the hybrid rule set with both qualitative spatial reasoning rules and quantitative LSS computation $GeoR_{bk}^{hybrid}$ (see chapter 7).

Jena2 or more specifically Rete, does not inherently provide a means to call a backward rule during the course of rule inferencing. To overcome this, a backward call is added as a builtin. That is, RCC-8 or RCC-12 base relation (henceforth denoted reserved spatial predicates, RSP) are not represented as triple patterns, but are added to the engine as builtins (transparently to the user) during mapping from *SWSRL* to RDF syntax using the transformation function $\mathcal{T}_{jena}$. The builtins are coded as procedural attachments and are registered with Jena2's forward engine. For example the following shows the simple translation between an

SWSRL rule to Jena's RDF triple format for the topological relation *Inside*.

$$\text{NS:Inside(?A ?B) AND ...} \rightarrow \text{Head(?X ?Y)}$$

$$\mapsto$$

$$\text{ask(?A NS:Inside ?B) AND...} \rightarrow \text{(?X Head ?Y)}$$

Once the builtin is called, the backward rule engine is initialised over the current set of explicit and entailed triples (all intentional and extensional triples). To increase the efficiency of retrieving the geometry from the external geometric processor, calls to backward rules must only contain ground variables. Thus backward rules only evaluate one relationship between two geofeatures at a time, and will only either return true or false (they are hence semantically safe in that they do not alter the underlying fact base). For example consider the *SWSRL* rule 8.1, where meta-tags have been omitted for the sake of brevity, and the ask predicate has been added for the original predicate NS:Inside(?A ?B).

[`meta-tags` : Region(?A) AND Region(?B) AND ask(?A, NS:Inside, ?B)) → ...]

(8.1)

`ask(?A,Inside,?B)` now represents a query to the backward rule set. Builtins in Jena2 (and indeed all Rete based engines) are only evaluated after all ground body predicates are satisfied. Then, as the two Region predicates in the rule ensure that the variables $?A$ and $?B$ are bound before the backward query is executed, the backward call will be initialised with only ground variables. Intuitively then, `Inside(?A ?B)` will return either true or false based on whether that relationship can be inferred from *SWSRLO*, or whether it can be it can be determined from the geometry in the LSS when using the $GeoR_{bk}^{hyrbid}$ rule set.

8.3.1 Enabling Incremental Update

One important property of Rete networks is their ability to handle incremental update. That is, when a new fact is added to *SWSRLO*, a token representing the fact is entered into the Rete network and all relevant matching rules are fired.

This property is important to maintain in *SWSRL* as integrity rules should continuously monitor an evolving *SWSRLO* geo-ontology.

However, builtins are represented external to the Rete network. Each time a new topological relation is inferred and added back into the Rete network, builtins within rules are not re-evaluated unless a logical predicate (alpha-node) is rematched to the new fact. Therefore, to guarantee that a newly added relation rematches and hence re-triggers (in the case where the rule has previously been triggered) a rule $r \in GeoR_{fd}$ that contains the **ask** builtin, we need to add new logical predicates to each rule r that matches (as an alpha node) to any newly inserted topological relations. Importantly, we only need to match to topological relations, as these are the only type of relation that could help to satisfy a backward RCC-8 or RCC-12 call.

As an example, consider the following geo-ontology axioms in *SWSRLO*, forward deduction rules in $GeoR_{fd}$, backward rules in $GeoR_{bk}$ and fact update to *SWSRLO* (in Jena2 rule syntax and where NS is the namespace of the current geo-ontology).

$GeoR_{bk}$

$$triple(?A, \text{NS:}Inside, ?B) \leftarrow triple(?A, \text{NS:}Inside, ?C), triple(?C, \text{NS:}Inside, ?B) \tag{8.2}$$

SWSRLO

$$\rightarrow \; triple(\text{NS:}Roath, \text{rdf:type}, \text{NS:Region}) \tag{8.3}$$
$$\rightarrow \; triple(\text{NS:}Cardiff, \text{rdf:type}, \text{NS:Region}) \tag{8.4}$$
$$\rightarrow \; triple(\text{NS:}Wales, \text{rdf:type}, \text{NS:Region}) \tag{8.5}$$
$$\rightarrow \; triple(\text{NS:}Roath, \text{NS:}Inside, \text{NS:}Cardiff) \tag{8.6}$$
$$\tag{8.7}$$

$GeoR_{fd}$

$$triple(?A, \text{rdf:type}, \text{NS:Region}), triple(?B, \text{rdf:type}, \text{NS:Region}), ask(?A, \text{NS:}Inside, ?B)$$
$$\rightarrow \; triple(?A, \text{NS:}Inside, ?B) \tag{8.8}$$

Update to *SWSRLO*

$$\rightarrow triple(\text{NS:}Cardiff, \text{NS:}Inside, \text{NS:}Wales) \tag{8.9}$$

`ask(?A,Inside,?B)` in rule 8.8 is a builtin which queries the rule set $GeoR_{bk}$

in the backward engine (which contains only one rule for simplicity). During execution of the forward ruleset, the **ask** builtin would then be called with only the explicit relational fact Roath Inside Cardiff. Hence as it stands, the rule would only infer the fact Inside(Roath Cardiff). Now, as **ask** is a builtin and not a simple pattern (alpha-node), if the token representing the new fact 8.9 i.e. *triple*(Cardiff, NS:Inside, Wales) is entered at the root of the Rete and propagated to its leaves, it does so without re-matching any of the alpha-node patterns: *triple*(?A, rdf:type,NS:Region), *triple*(?B, rdf:type, NS:Region) in rule 8.8, thus not triggering the re-evaluating of the builtin `ask(?A Inside ?B)`. However, by adding fact 8.9, rule 8.8 should now infer the fact Inside(Roath Cardiff) as well as the new fact Inside(Cardiff Wales) and the implicit fact Inside(Roath Wales), however the current implementation does not.

Adding RETE Dummy Nodes: To alleviate this problem we add dummy nodes to the Rete for rules that have an **ask** predicate. We can categorise the type of change (new facts) which could effect the outcome of an **ask** predicate to only those rules that infer RCC-12 relations *. As a result, we can add those effectual RCC-12 triple patterns into the Rete as *dummy* alpha nodes to each effected rule. That is to say, all possible RCC-12 spatial relationship predicates are added to each rule that contains an **ask** predicate, as any one of these could be used in the process of qualitative spatial reasoning to help conclude the truth of the topological relation in the **ask** predicate. These `dummy` nodes are added as *dummy* predicates to all rules which contain RSPs during the translation from $SWSRL$ to Jena using the transformation function $\mathcal{T}_{jena}$ (see Figure 8.3.1 for an illustration). Importantly, by only adding dummy nodes, and not changing the Rete algorithm directly, the interleaved functionality of $SWSRL$ to be added easily to any existing Rete engine implementation.

More formally, we add a disjunction of all RCC-12 base relations to each rule r that contains an **ask** predicate during the transformation from $SWSRL$ to RDF triples. For example, during translation we add to each rule r the disjunction:

$$C(?A\ ?C) \vee DR(?A\ ?C) \vee DC(?A\ ?C) \vee P(?A\ ?C) \vee coP(?A\ ?C) \vee NTP(?A\ ?C)$$

*Remembering here that RCC-8 relations are mapped to RCC-12 relations for internal operation, hence any new RCC-8 relation creates a new RCC-12 relation

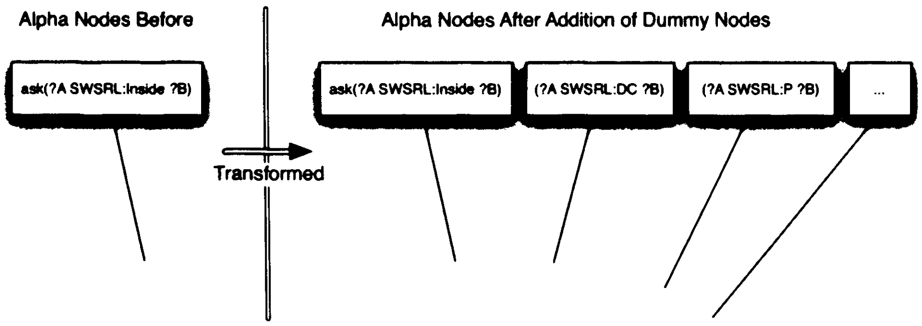


Figure 8.1: Adding dummy alpha nodes to the Rete. The left hand side shows the Rete alpha notes before the addition of dummy nodes, and the right shows the Rete alpha notes after the addition of dummy nodes.

$$\vee coNTP(?A\ ?C) \vee P^{-1}(?A\ ?C) \vee coP^{-1}(?A\ ?C) \vee O(?A\ ?C) \vee NTP^{-1}(?A\ ?C)$$
$$\vee coNTP^{-1}(?A\ ?C) \wedge B \rightarrow H \quad (8.10)$$

Where B represents the original rule body of r, and H represents the original rule head of r. As a result of adding dummy predicates, any rule that includes a RSP builtin will be re-evaluated on the conclusion or addition of any RCC-12 spatial relation in $SWSRLO$. Intuitively this measure, although guaranteeing complete reasoning, has a serious effect on reasoning complexity and in turn reasoning speed. That is, each interleaved rule will re-run each time a new relation is added between any region in $SWSRLO$. Essentially for the set of spatial integrity rules, the intrinsic temporal redundancy characteristic of the Rete algorithm is suppressed, resorting instead to a more classical brute force (data driven) rule execution approach. More formally, for any set of interleaved rules the $O(RFP)$ complexity of using Rete is worsened to a brute force $O(RF^P)$ complexity, where F is a topological relation in $SWSRLO$. In addition as each predicate P is evaluated in the backward engine, a further reasoning overhead (execution of the backward engine) is exhibited for each **ask** predicate P. The overall effects of interleaved reasoning on speed and memory usage are measured empirically and shown later in the results chapter 11.

8.3.2 XSB Backward Engine Implementation

Once called, RSP are executed in Jena2's integrated XSB backward engine. The potential benefits of interleaving a backward engine such as XSB with the forward Rete engine were outlined in section 6.5.2. In overview whereas Rete trades in-memory usage for speed, XSB / Prolog style systems (based on the Warren Abstract Machine - WAM) are known to be good at efficiently handling memory usage during reasoning, as shown for example in [27, 252, 254]. Furthermore, Prolog engines have shown good scalability for RDF applications, reasoning with 3 million triples, loaded from wordnet*, in 2-7 μs with a 237mb in-memory footprint [275]. Hence our hypothesis of using backward reasoning engines for efficient topological reasoning. However of note there are many factors that effect the performance of such implementations e.g. machine hardware, memory caching, garbage collection, along with the choice of ruleset. Consequently to better gauge the performance of the backward rule engine over geo-ontologies using topological reasoning rules, empirical tests are performed and shown later in chapter 11.

XSB is based on both the principles of SLD resolution as well as SLG resolution. SLG resolution allows predicates to be tabled and is known to be sound and search space complete for non-floundering queries[30] - importantly the class of Datalog (definite Horn) programs which *SWSRL* adheres to. Importantly, SLG resolution can compute the transitive closure of recursive predicates without entering an infinite loop. This feature is fundamental to the implementation of *SWSRL* and defined spatial rule sets as all RCC-12 relations are involved in recursive rules, for example: $P \wedge P \rightarrow P$ (where P is the PartOf RCC-12 relation).

8.4 SWSRL Courteous Logic Extension

As proposed in Chapter 6, *SWSRL* will support default integrity constraints and their exceptions using Courteous Logic extensions. Here, we describe the implementation of the simplified form of Courteous Logic CLP^{-1} used in *SWSRL*.

8.4.1 CLP^{-} Implementation

The implementation of Courteous Logic in existing reasoning engines is via the use of a courteous compiler. A courteous compiler transforms expressive Courteous logic extensions leaving a semantically equivalent ordinary logic program [102].

As described in section 6.5.1, the full Courteous logic has been restricted in

*http://wordnet.princeton.edu/

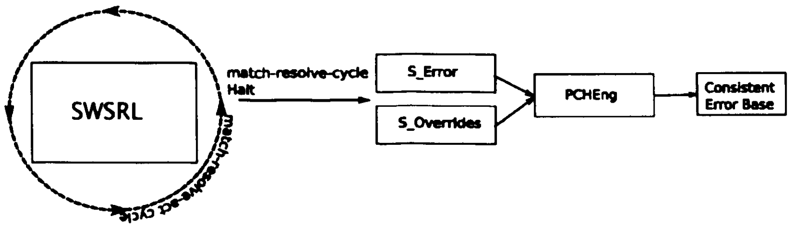


Figure 8.2: *PCHEng* Information Flow, where S_Error represents error predicates in *SWSRL* and S_Overrides represents overrides predicates. The result of the process is a consistent error base.

SWSRL, for simplicity, to only include the classical mutex ($\perp \leftarrow E \wedge \neg E$), and classical negation is only allowed to appear in front of error predicates in the head of integrity rules. This means a positive and negative version of the same error predicate could be inferred by integrity rules (representing default integrity assumptions and individuals exceptions). However after inferencing has finished, any contradicting errors should not exist simultaneously in *SWSRLO*. If such a state does exist, either the positive or negative version should be removed based on which has stated priority defined by an *Overrides* predicate in *SWSRL*. By using these expressive restrictions we are able to alleviate the need for a separate courteous compiler. Instead, we employ a simple algorithmic extension to Jena2 denoted Prioritized Conflict Handling Engine (PCHEng) to perform a post-processing cleanup step that removes any conflict.

PCHEng **Algorithm:** Firstly, as is assumed in the concrete syntax of *SWSRL*, strong negation ($\neg$) is not supported (it is also not supported natively within Rete). Therefore it is assumed the transformed *SWSRL* knowledge base does not contain the appearances of classical negation, only a syntactic equivalent; *error* and *notError* (removing classical negation in this way is a common way of adding a limited form of classical negation in ordinary or definite logic programs [102]). Furthermore, the explicit mutex between *error* and *notError* is assumed.

At the end of inferencing, when Rete's match-resolve-act cycle has halted, all error predicates (*Error* or *notError* denoted S_Error in Figure 8.2) are fed into the Prioritised Conflict Handling Engine (*PCHEng*) along with the *Overrides* sub program of *SWSRL* (denoted S_Overrides in Figure 8.2), where any conflicts are resolved.

To do this, the *PCHEng* removes, pairwise, conflicting error predicates by checking for a relevant *Overrides* predicate with which to resolve the conflict. As is the norm with a CLP, if an *Overrides* predicate can't be found, then both positive and negative versions of the error are removed - treated sceptically. The algorithm is shown below, its computational complexity is of order $O(n^2)$. That is, constructing arrays for overrides triples (Ov) and error predicates (S) takes O(n) - linear. Both main iterations take O(n^2). Intuitively the entire algorithm takes $n^2 + n^2$, leaving an overall complexity of O(n^2).

Algorithm 8.1 PCHEng

```
Let S = array of all error individuals in the error ontology
Let P = array of 2-tuple records representing conflicting errors (error, error) -
  conflict set
Let Ov = array of all overrides predicates
for (i =0; i < sizeof(S);i++) do
  for (int j=0; j < sizeof(S);j++) do
    if (i ≠ j) then
      if (s[i] complementof s[j]) then
        add s[i] and s[j] to P
      end if
    end if
  end for
end for
for (int i=0; i < sizeof(P); i++) do
  Let found = FALSE
  for (int j =0; j < sizeOf(Ov); j++) do
    if (Ov[j] represents priority over P[i]) then
      Remove defeated error triple
      Set found = true
    end if
  end for
  if (found == false) then
    remove both error triples
  end if
end for
```

8.5 *SWSRL* Builtins (Procedural Attachments)

In this section we outline the implementation of *SWSRL*'s procedural attachments (builtins) shown in chapter 6.

8.5.1 LSS SPATIAL BUILTINS

Spatial Relation Determination: The general implementation logic for a LSS spatial relation builtin is shown in Algorithm 8.2. The SQL query used is shown in Appendix B.5. Where <SQL,A,B> is a possible spatial relation SQL query constructed between the Geofeatures A and B, shown in Appendix B.5.

Algorithm 8.2 Determine Spatial Relation

Input: Geofeature A and Geofeature B
Output: True if Geofeature A has the same spatial relation to B as tested
Let boolean hasRelation $\longleftarrow$ false
Let string sqlQuery $\longleftarrow$ <SQL,A,B>
hasRelation $\longleftarrow$ SQLConnection.SQLQuery(<SQL>)
Return hasRelation

Distance and Area Computation: The general implementation logic for a LSS distance and area builtin is shown in Algorithm 8.3. The SQL query used is shown in Appendix B.5. Where <SQL,A,B> is a possible area / distance SQL query constructed between the Geofeatures A and B.

Algorithm 8.3 Determine Spatial Distance / Area

Input: Geofeature A and Geofeature B
Output: The result (in meters) of the computation
Let string sqlQuery $\longleftarrow$ <SQL,A,B>
Let double result $\longleftarrow$ SQLConnection.SQLQuery(<SQL>)
Return result

8.5.2 COMPARISON AND ARITHMETIC BUILTINS

All comparison and arithmetic builtins, shown in Appendix B.4 and B.6, are trivially encoded. As input a comparison or arithmetic builtins takes, as suitable, either two Geofeatures A and B, two literal values L_1 and L_2 or a mixture thereof.

8.6 LOCALISING INCONSISTENCIES

In this section we explore two different error localisation methods. The first attempts to suggest consistent topological relations between two regions where an inconsistency has been detected, named relational confidence. The second tries to trace and locate the source of an inconsistency, named compositional confidence. Both of these methods are included within the spatial rule engine and operate over the output of reasoning on an $SWSRLO$ geo-ontology using the RCC-12 rulesets e.g. $F^{RCC12}_{<D,I>}$. Importantly, these methods will not work when using $SWSRL$ with interleaved rulesets as the results of all compositions need to be stored in working memory.

This section is organised as follows. We first introduce two variations of the *space law* rulesets defined in chapter 7 which enable and aid both statistical techniques relational confidence and compositional confidence, which themselves are presented in the final two sections. The results and evaluation of both techniques on real world $SWSRLO$ geo-ontologies is then shown later in chapter 10.

8.6.1 Modified Spatial Rule Sets

We first define two additional spatial rulesets. The first is necessary for both localisation techniques as it stores, as a vector, the result of every composition and converse rule in F^{RCC12}_{D}. The second ruleset introduces a new procedural attachment, `validTR`, that tries to reduce the effect of error propagation that occurs when reasoning with the set F^{RCC12}_{D}.

8.6.1.1 F^{RCC12}_{D} Error propagation

If the set of topological relations in $SWSRLO$ is inconsistent, during deductive reasoning with the ruleset $F^{RCC12}_{<D,I>}$, those inconsistencies will serve to infer new inconsistent relations. Consequently, inconsistencies or errors are propagated between regions in $SWSRLO$. This situation is illustrated in the spatial network shown in Figure 8.3. A,B,C and D are regions in $SWSRLO$, and raw/explicit topological RCC-12 relations from $SWSRLO$ are shown in bold . An inconsistent relation, DR, between C to A is added which derives the inconsistent relation (with respect to the existing relation between C to B) DR between C and B. Deduction rules then generate a compositional inference between C and D via $< C, B >$ and $< B, D >$. This new relation does not intersect with the existing relation (NTP^{-1}) between C and D, hence leading to a further inconsistency.

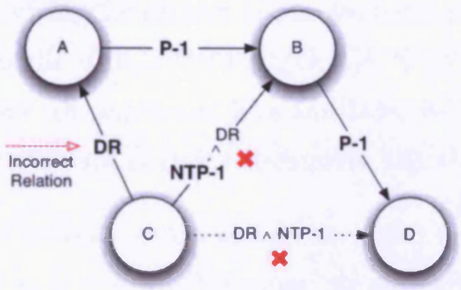


Figure 8.3: Example of error propagation

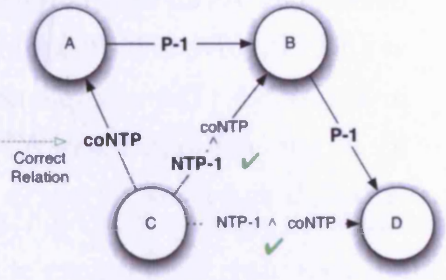


Figure 8.4: Corrected error propagation

The inconsistency could have arisen from the set of conjunctive relations between any of the region pairs <A,B>,<C,A>,<B,D> or <C,B>. In this way, even though an error has been derived between regions A, C and C, D, only the raw relation DR between A and C was originally inconsistent. The second inconsistency was caused by the propagation of the inconsistent relation DR which inferred the relation DR (between C and B), in turn leading to the inconsistent relation DR between C and D. In this case, adding the relation $coNTP$ between C and A is enough to remove the inconsistency, see Figure 8.4.

Preventing Error Propagation: Error propagation can be partly prevented by adding a look-ahead builtin denoted `validTR` (valid topological relation), to the body of all deduction rules in F_D^{RCC12}. The builtin `validTR`, described algorithmically in Appendix C.2, has the form:

```
validTR(<regionA> <relation> <regionB>)
```

Where <regionA> and <regionB> are variables that are grounded by regions in $SWSRLO$. `validTR` then checks that <relation>, which is an RCC-12 relation, can consistently be added to $SWSRLO$ - it is not inconsistent with the current set of RCC-12 relations holding between the regions bound to regionA and regionB. If the deduction is valid, `validTR` evaluates to true, and the relation in the head can be inferred. Otherwise the rule is blocked (the rule body will not evaluate to true) and is hence the head RCC-12 predicate is not inferred.

As a side effect, the look-ahead deduction builtins induces a non-deterministic order dependency on the semantics of $SWSRL$. That is, as the order of execution of rules in F_D^{RCC12} is not guaranteed to be the same each time, different errors will be inferred. As an example consider the unsatisfiable set of relational constraints θ = {NTPP(A,B), NTPP(B,C), DC(A,D), TPP^{-1}(D,C), NTPP(E,A)}- illustrated in Figure 8.5 (The example assumes RCC-8 relations and reasoning for simplicity in presentation), where $NTPP(B,C)$ is the erroneous relation and the nodes A,B,C,D and E are regions in $SWSRLO$.

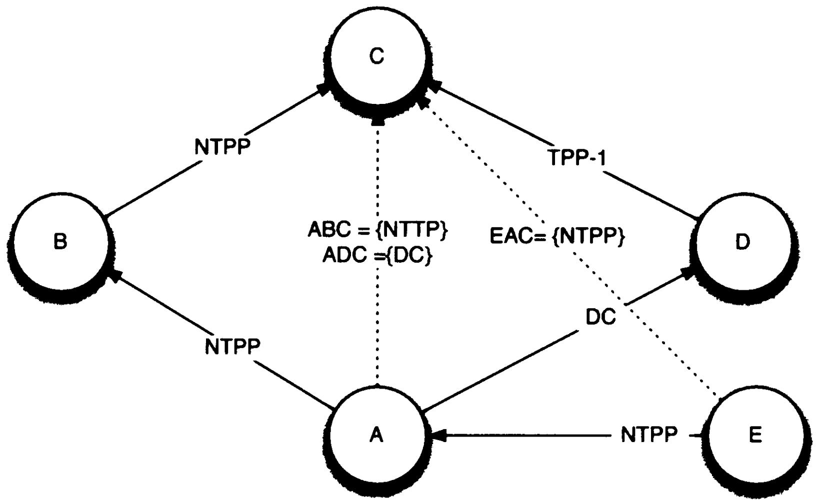


Figure 8.5: Spatial network visualising the relational constraints in θ

Now if the composition between the regions A,B and C is derived first, $NTTP$ is added between the Regions A and C - which in the absence of any other knowledge other than the universal relation holding between A and B, is consistent. If we assume the composition between regions E,A and C is derived second, the erroneous relation $NTTP$ is added between E and C. Then the error relation $NTPP$ between A and C effectively blocks the correct relation being derived between A and C from the composition of A,D and C. Hence in this case, the error relation has been propagated so as to derive an erroneous relation between the regions E and C.

Now in the case where the composition between the regions A,D and C is derived first, the erroneous composition between regions A,B and C is blocked, and hence the correct relation is added between A and C (DC) and E and C (DC). Here an integrity rule would be violated and thus produce an error between the regions A,B and C. Whereas previously an integrity rule would have derived an error between regions A,D and C, moreover an error relation would now exist between the regions E,A and C.

8.6.1.2 $F^{RCC12}_{<D,I>}$ Rule Set With Compositional Counting

The second addition to the *space law* ruleset is now shown. Error localisation methods require information about each possible composition path even if the result of that composition was previously derived. To accommodate this, a new reserved builtin predicate named *record* is added to the head of forward deduction rules (the set F^{RCC12}_{D}). The builtin *record* has five terms and is henceforth defined as:

$$\texttt{record}(\text{from relation to via rule\_name}) \tag{8.11}$$

Where, 'from', 'to' and 'via' are substituted with the regions bound to the variables $?A$, $?B$ and $?C$ from a compositional inference rule $r \in F^{RCC12}_{D}$ and from converse rules $rc \in F^{RCC12\frown}_{D}$. 'Relation' is substituted for the name of the relation predicate in the head of a rule $r \in F^{RCC12}_{D}$ and $rc \in F^{RCC12\frown}_{D}$. 'Rule_name' is substituted for the name of the rule as encapsulated in a rule's `<ruleName></ruleName>` metatag. Grounded `record` predicates are added, during reasoning, to the error ontology subset of $SWSRLO$. Rule 8.12 is then an example $SWSRL$ rule where `record` has been added to the head of the rule.

[`<label>coNTP_P-1</label><ruleLevel>0</ruleLevel><ruleGroup>Topological </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass>` : coNTP(?a ?b) AND P-1(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) → coNTP(?a ?c) AND record(?a coNTP ?c ?b coNTP_P-1)]

(8.12)

Furthermore, to test the maximum effectiveness for both error localisation methods, error propagation needs to be controlled or stopped. A measure to control (but certainly not stop) error propagation has been proposed in the previous section, section 8.6.1.1, using the `validTR` lookahead reserved builtin. Consequently an auxiliary rule set (denoted $F^{RCC12}_{RecordSplit<D,I>}$) is created for testing purposes. Where each rule $r \in F^{RCC12}_{D}$ and $r \in F^{RCC12\frown}_{D}$ is augmented with the builtin `validTR`, and a supporting rule is created where the head only contains the builtin `record` and the body omits the *validTR* builtin. The split is necessary as both error localisation methods need to work on the consequent of compositional and converse inferences even if the deduction is not made to help eliminate error propagation. Rule 8.13 is an example of a compositional inference rule in $F^{RCC12}_{RecordSplit<D,I>}$ that contains the `validTR` builin, and rule 8.14 is an example of the supporting rule to 8.13 which includes the `record` builtin in the rule head, but omitting the `validTR` predicate in the rule body.

[`<label>C_NTP</label><ruleLevel>0</ruleLevel><ruleGroup>Topological </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass>` : C(?a ?b) AND NTP(?b ?c) AND ValidTR(?a O ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(? → O(?a ?c)]

(8.13)

[`<label>C_NTP_record</label><ruleLevel>0</ruleLevel><ruleGroup>Topological </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass>` : C(?a ?b) AND NTP(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) → record(?a O ?c ?b C_NTP)]

(8.14)

8.6.2 Relational Confidence

In this section we present the statistical method relational confidence, that assigns confidence values to RCC-8 base relations between any two regions in *SWSRLO*, where the confidence is the degree to which the system believes the relation should hold. Relational confidence is useful in two scenarios:

(a) It can be used as a mechanism to suggest the relations that should hold between two regions where an inconsistency has been detected, hence suggesting a means of rectifying the error.

(b) Relational confidence is used with a second method, described in the following section, which can help to identify the source of errors even when errors have been further propagated.

As an example of the first case, consider the following set of relational constraints θ = {EC(A,B),NTPP^{-1}(B,C),EC(A,D),TPP^{-1}(D,C),NTPP(A,C)},as illustrated in Figure 8.6, where relations in bold represent explicit (raw) relations (both RCC-8 and their equivalent RCC-12 Relations), and relations in italics represent implicit inferred relations. The system will show not only that θ is unsatisfiable, but it will try to determine which relation should exist between the regions A and C.

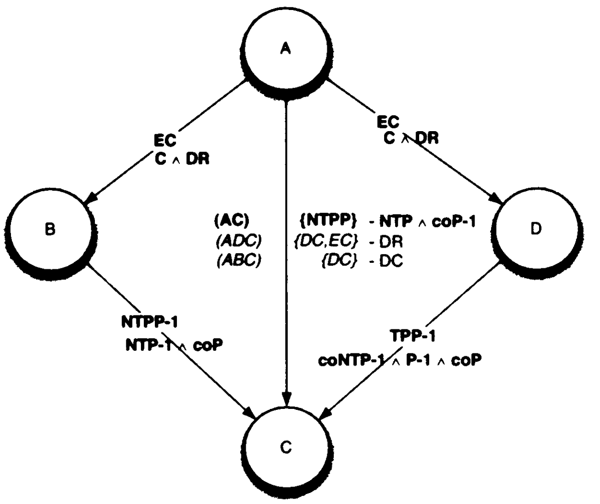


Figure 8.6: Spatial network visualising the relational constraints in θ

Following compositional paths for the region triples $\{A, B, C\}$ and $\{A, D, C\}$ leads to the RCC-12 inferences DR and DC between A and C. These relations, if mapped to their respective set of disjunctive base RCC-8 relations (the sets $\{DC, EC\}$ and $\{DC\}$), contradict the existing relation $NTPP$ between A and C. Hence, as a first step the set of constraints θ are inconsistent. Now from these implicit relations, it is possible to suggest that the relation with the most intersections (DC) should hold between A and C - which would in this case make the scene satisfiable.

Now if we consider further evidence using a fifth and sixth region E and F respectively, and an expanded set of relational constraints θ = {EC(A,B), NTPP^{-1}(B,C), EC(A,D), TPP^{-1}(D,C), NTPP(A,C), NTPP(A,E), DC(A,F), TPP^{-1}(E,C), EC(F,C)} (illustrated in Figure 8.7), the compositional inference between the region triples $\{A, E, C\}$ and $\{A, F, C\}$ gives further evidence that the relation DC holds between A and C. Indeed all bar the original relation agree with the relational constraint $DC(A, C)$.

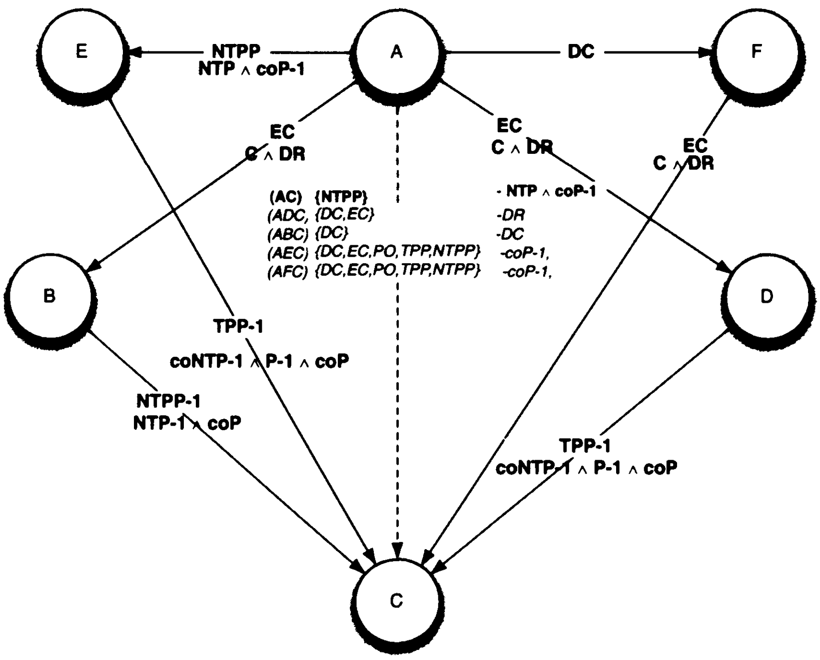


Figure 8.7: Spatial network visualising the relational constraints in the expanded set θ

Method: We now present the general method to compute relational confidence. More formally, Let G be a three dimensional grid* datastructure, which represents a cubic table of elements. The dimensions of the grid are $n \times n \times m$ where n = number of Regions in $SWSRLO$ and $m = n + 2$ (two extra element positions used to hold the existing relations and any derived converse relations).

The 1^{st} axis of the grid (horizontal columns) is indexed using the subscript i. The 2^{nd} axis (vertical rows) is indexed using the subscript j. Finally the 3^{rd} axis (length) is indexed using the subscript k. Each axis is a mirroring of every other and represents a region in $SWSRLO$, where $i, j \in [0, n]$ and $k \in [0, m]$. The region indexed by i can be regarded as the *from* region, the region indexed by

*We use the term grid to refer to a simple matrix without associated matrix functions

j be regarded as the *to* region, and the region indexed by k can be regarded as the *via* region all forming a compositional region triple $\{R_i, R_k, R_j\}$ (where R_i,R_k and R_j are regions in $SWSRLO$). The element in position $G_{0,0,0}$ is null and does not represent an entry of any sort. Each element, where i,j and $k > 0$ in the grid ($G_{i,j,k}$), represents a conjunctive set of RCC-12 relations as determined by the RCC-12 compositional inference from the regions referred to by the indices i.e. R_i, R_k and R_j ($R_{ik} \otimes R_{kj}$). Clearly some entries will be the universal relation if the $SWSRLO$ only contains a partial scene description.

Existing relations (those explicit in $SWSRLO$ between the regions R_i and R_j) are entries in the position $G_{i,j,k}$ where $k = n+1$. For example the existing relation between the 3rd and 4th regions, where $SWSRLO$ contains 6 regions, is stored in the position $G_{3,4,7}$. Similarly, the set of all derived converse relations between the regions R_i and R_j are stored as entries in position $G_{i,j,k}$ where $k = n + 2$ (i.e. $G_{3,4,8}$). An illustration of the grid G is shown in Figure 8.8.

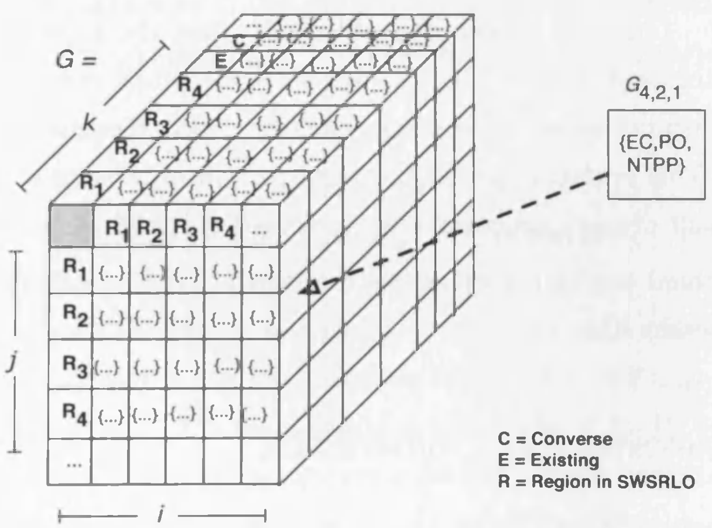


Figure 8.8: Example Grid G of region triples and the results of their compositions

A preprocessing step is required that converts every conjunctive set of RCC-12 relations to their equivalent set of disjunctive RCC-8 base relations which is shown algorithmically in Appendix C.3.

Now that we have defined the relational representation structure and converted each conjunctive set of RCC-12 relations to their disjunctive RCC-8 equivalent, we describe the function to compute relational confidence per region pair R_i and R_j. For each region pair R_i and R_j where $i,j \in [1,n]$, the confidence to which each base RCC-8 should hold is determined by the following: Let $\omega = \{DC, EC, PO, NTPP, TPP, EQ, TPP^{-1}, NTTP^{-1}\}$, then the confidence score for each base RCC-8 relation $r \in \omega$ is then defined as (where the result is a value in [0,1]):

$$conf(r) = \frac{\sum_{k=1}^{m} \begin{cases} 1 & r \cap G_{i,j,k} \neq \theta \\ 0 & \text{otherwise} \end{cases}}{filled(i,j)} \tag{8.15}$$

$$filled(i,j) = \sum_{k=1}^{m} \begin{cases} 1 & G_{i,j,k} \neq \theta \\ 0 & \text{otherwise} \end{cases} \tag{8.16}$$

Clearly if any base RCC-8 relation is assigned a score of 1, then the region pairing R_i and R_j has a consistent RCC-8 relation holding between them, and no further processing is necessary. Otherwise the resultant confidence values are added to a ranked set ψ - ranked in descending order such that the highest ranked relations are toward the beginning of the set. If the result of all consistent compositions produces a disjunctive set of relations, each of these relations are given the same confidence. This is important as the method will only suggest a single relation when the result of assumed consistent compositions produce a definite result.

The relational confidence measures described above are described algorithmically in Appendix C.5.

8.6.3 Compositional Confidence

Compositional confidence builds on the results of relational confidence to try and locate the source RCC-8 relation(s) that rendered *SWSRLO* inconsistent. More specifically, whereas relational confidence serves to help determine the correct relation between two regions, compositional confidence tries to trace the compositions that produced inconsistent relations, hence trying to locate the source inconsistent topological relation.

Compositional confidence works under the assumption that error propagation is reduced using deduction rules that include the `validTR` builtin, and that all compositional inferences are recorded i.e. using the previously defined integrity rule set $F^{RCC12}_{RecordSplit<D,I>}$. Error propagating serves to dilute existing consistent inferences generating new, misrepresented compositional inferences. Hence, the effectiveness of compositional confidence is reliant on reducing error propagation. Moreover, preventing error propagation is an important first step that localises* inconsistent sub-networks of topological relations in $SWSRLO$.

Compositional confidence works off the same grid G as defined in section 8.6.2 an then proceeds by the following steps:

(a) For each region pair (R_i and R_j where $i, j \in [0, n]$), construct the (ranked) ordered set ψ of relation using the relational confidence measure shown in section 8.6.2.

(b) Add the highest ranked relation(s) in ψ to a new set v.

(c) For each composition path (that is, for each index k where $k \in [0, m]$ from the RCC-8 converted grid G) for every region pair R_i and R_j, remove those composition paths from G that do contain any one-of the relations in v (hence leaving suspect compositions). More formally, for each region pair R_i and R_j:

$$\forall k \in [0, m] \begin{cases} \text{set } G_{i,j,k} = \theta & G_{i,j,k} \cap v \neq \theta \\ G_{i,j,k} = G_{i,j,k} & \text{otherwise} \end{cases} \tag{8.17}$$

(d) Again, for each remaining composition path (for each index k of G where $k \in [0, m]$) count the occurrence of each edge that still has an entry in G. That is, where $G_{i,j,k} \neq \theta$, and an edge is either pair $< R_i, R_k >$ or $< R_k, R_j >$. Here we assume that the source error (the topological relation associated to one of the edges $< R_i, R_k >$ or $< R_k, R_j >$) generates the most inconsistent compositional inferences.

As an example of the technique, assume that $SWSRLO$ contains the regions A,B,C and D, and consider the following triples (all elements $G_{i,j,k}$ in G that are non empty sets ordered $\{i, k, j\}$):

{A,C,D}, {A,C,E}, {A,C,B}, {C,D,B},

*Although because the predicate is order depended and the rule scheduler is pseudo-random, this 'help' is not guaranteed

The edge count is then:

$$\{A,C\} = 3,\ \{C,D\} = 2,\ \{C,E\} = 1,\ \{C,B\} = 1,\ \{D,B\} = 1,$$

The edges with the highest count are likely sources of erroneous RCC-8 relations. In this case suggesting that the topological relation associated to the edge {A,C} as the source of the error.

8.7 SUMMARY

In this chapter we described the spatial reasoning engine, based on existing technologies, used to implement the semantic web spatial rule language *SWSRL*. In addition we also showed two error localisation techniques. The first helps to suggest topological relations to overcome inconsistencies. The second tries to identify the source of inconsistencies even when the inconsistency has propagated further inconsistent relations. In the next chapter different *SWSRLO* geo-ontologies are instantiated which are then used for testing the capabilities of *SWSRL* in chapter 10.

Chapter 9

Instantiating Geospatial Ontology Bases

To test the viability of the proposed system to maintain the consistency of real world geospatial information, a suitably instantiated geo-ontology in *SWSRL* must be used. As the maturity of the semantic web grows so will the availability of pre-constructed geospatial ontologies (information in web documents that commit to a defined geospatial ontology). Despite the efforts of Swoogle[*] and SemWebCentral[†] semantic web search engines to discover ontologies, only a few publicly available populated geospatial ontologies exist (this is likely to due licensing issues of most geographic datasets). One such placename ontology using freely available datasets is provided by the Geonames project. However, even though we attempt to construct a basic geo-ontology from Geonames in section 9.3.1, as is common with most gazetteer style information, locational information is recorded as a point reference (using polygon centroids), therefore limiting our ability to determine qualitative spatial containment and overlap relations (Part P and overlap O and their specialisations)[‡].

In order to evaluated the framework, a number of sample geo-ontologies in the proposed language *SWSRLO* are constructed. These are developed from three different sources:

---

[*] http://swoogle.umbc.edu/

[†] http://projects.semwebcentral.org/

[‡] although a basic parent hierarchy is captured using a set of administrative division tags -as strings

- (1) Using a synthetic topological spatial configuration generator, which uses a genetic algorithm to generate consistent topological spatial descriptions, where the number of regions and distributions of each RCC-8 relation can be varied.

- (2) Mining natural language spatial scene descriptions from Wikipedia articles.

- (3) Official geographic datasets that contain rich feature geometries.

9.1 GENERATING SYNTHETIC GEO-ONTOLOGY INSTANCE BASES

Developing a synthetic ontology base is useful to allow fine control over the types of topological relationships and the number of regions in the ontology. Hence here we show how to construct, automatically, a controlled set of topological spatial networks.

The spatial networks constructed must be consistent and valid topological spatial configurations. However, finding a consistent spatial configuration is a combinatorial problem which is exponential in the size of the input (number of regions). More specifically, a full spatial configuration forms a complete and finite directed graph (digraph)*, where the regions represent nodes or vertices V and the topological relationships form a set of edges E between them. A full scene is an undirected graph which allows self-loops. As previously described, topological spatial relations are binary relations R on the the set V of regions where two regions $x, y \in V$ are connected if xRy. Considering the digraph example in Figure 9.1, which is a complete network of regions and topological relations, the search space can be calculated as:

$$size(G) = 8^{n^2} \tag{9.1}$$

There are $n \times n$ possible edges between n regions, and each edge can take one of eight base RCC-8 topological relations. Clearly, this results in a large search space.

*Referred to as a constraint graph or spatial network in previous sections

For small scenes of 20 regions the number of possible topological configurations would be $8^{(20\times 20)} = 1.71e + 361$.

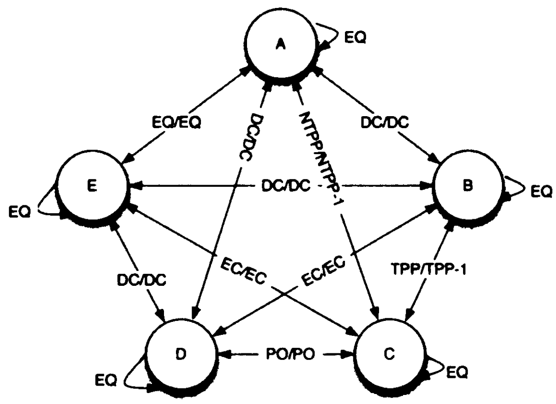


Figure 9.1: Complete undirected graph representation of a spatial configuration/network

Using a-prior knowledge about converse relations and the identity relation EQ, the search space can be constrained by considering only a directed graph without self-loops (so $\overrightarrow{xy}$ but not $\overrightarrow{yx}$ and a relation is not connected to itself). Then if path consistency is enforced over the directed graph, path-consistency of each converse and identity relation is also enforced [219]. As a result, the problem is reduced to a finite simple graph, thus limiting the search space to $(8^{\frac{n.(n-1)}{2}})$ - a significant reduction. The graph representation of a spatial configuration shown in Figure 9.2 is identical to that shown in Figure 9.1 but omitting self-loops and converse relations.

9.1.1 PROPOSED APPROACH

A brute-force approach to solving the problem stated in the previous section is clearly intractable. To highlight this, a random search approach was compared to the chosen approach described below in Appendix A.1, where the random approach never finds a consistent solution in the time allowed. Considering the large space complexity of the problem, it becomes necessary to employ a robust meta-heuristic optimization technique to find a valid spatial configuration in a *reasonable* amount of time. In particular, we employ an adaptive, evolutionary heuristic search approach, a genetic algorithm. Genetic algorithms are:

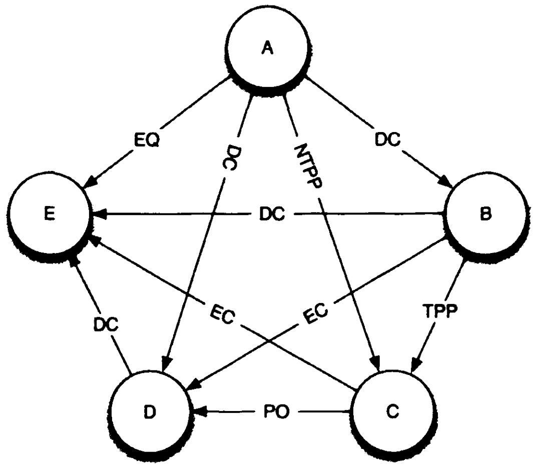


Figure 9.2: Complete simple graph representation of a spatial configuration / network

> "probabilistic search procedures designed to work on large spaces involving states that can be represented by strings" [94]

Genetic algorithms, largely attributed to Jon Holland in 1975 [124], stem from research into human biological reproductive processes and tries to mimic evolutionary biology. As the algorithm runs, each new evolution of potential solutions improves its quality until a suitable solution is found. Genetic algorithms prosper in situations with large, uneven search spaces* containing many possible local optima / minima. A genetic algorithm is comprised of the following core notions or concepts:

- Chromosome : one candidate solution to the problem typically encoded as a string of bits, where each bit in the string has a specified predetermined meaning.

- Population : one generation containing n candidate solutions (chromosomes).

- Fitness : The performance or quality of a candidate solution (chromosome) is assessed by a fitness function. Chromosome selection is determined using fitness of each candidate solution as a guide.

*the search space, or state space represents all possible solutions to the problem

- Selection: produces new solutions to the problem by selecting individual chromosomes which are later reproduced using a reproduction method. There are a number of different selection operators including the more popular *roulette-wheel selection*, *best* selection (fittest chromosome is always selection) and *random* selection.

- Reproduction: reproduction creates new offspring chromosomes and introduces variation into the population in an effort to avoid local optima in the search space. There are two variant inducing operators namely crossover and mutation. Crossover produces offspring (child chromosomes) by mating together two previously selected parent chromosomes. Mutation randomly alters or mutates a chromosomes solution (introduces random variation).

Genetic algorithms have found popular application for the travelling salesman problem [99], evolving computer programs (often termed Genetic Programming) [153], timetabling and scheduling [227]. Genetic algorithms have previously been proposed for solving general constraint satisfaction problems, for example see [40]. More recently the author along with Schockeart [233] employed a mixed genetic algorithm and ant colony optimisation technique to generate a visual interpretation of regions from a mixed set of spatial constraints, i.e. topological, bounding boxes and relative size constraints. However, to the best of our knowledge, no one has used a genetic algorithm to generate arbitrary consistent spatial scenes as presented in this chapter.

9.1.2 Spatial Genetic Algorithm Overview

The genetic algorithm is used to find one topological spatial configuration, out of potentially many, that satisfies a set of RCC-8 topological constraints Θ (as usual constraints of the form xRy where x and y are both regions). While doing so the algorithm tries to satisfy an additional constraint on the ratio of topological relations in the final spatial scene, a set denoted δ, where δ is the set:

$$\delta = \{DC_{\%}, EC_{\%}, TPP_{\%}, NTPP_{\%}, TPP_{\%}^{-1}, NTPP_{\%}^{-1}, PO_{\%}, EQ_{\%}\} \tag{9.2}$$

Each relation value is in the range [0,1]. The ratio of weights are normalised using:

$$relative_{\%}(\delta_i) = \frac{\delta_i}{\sum_{i=1}^{|\delta|} \delta_i} \tag{9.3}$$

Importantly, the relation percentages in δ are soft constraints, and may fluctuate from those originally specified during optimisation in order to obtain a consistent configuration. Therefore these percentages only serve as a guide and do not have to be strictly satisfied.

Implementation logic for the developed genetic algorithm is shown in Appendix C.4. The reminder of this section then describes the framework for the algorithm in more detail.

9.1.3 Problem Encoding - chromosome representation

Each chromosome represents one topological spatial configuration or candidate solution. The set Θ of constraints between all region pairs x and y for that candidate solution is encoded as a table, where the first row and column enumerates each region and the entries in the table corresponds to the relation between those regions. For example see table 9.1.

Region / Region	Region A	Region B	Region C
Region A	EQ	NTPP	DC
Region B	$NTPP^{-1}$	EQ	EC
Region C	DC	EC	EQ

Table 9.1: Sample (consistent) region relation table

The relations in the table are read in a row to column order, for example the first row second column refers to the relation from Region A to Region B. The reflection about the diagonal are converse relations, hence row 2 column 1 represents the converse relation of row 1 column 2. Relations on the diagonal (highlighted in table 9.1) are the identity RCC-8 relation EQ. As described, to narrow the search space, the genetic algorithm only optimises solutions that are non-converse or identity (EQ) relations - that is it considers only the upper triangle matrix (assuming an

LU * decomposition of the table). Converse relations can always be consistently determined and added in linear time using a table look-up.

9.1.4 INITIAL POPULATION

An initial population of 20 chromosomes is created using a pseudo-random assignment of spatial relations. Each of the eight JEPD RCC-8 relations are manually assigned a weighting in [0,1] (creating the set δ in 9.2). The weightings are normalised between 0 and 1 using equation 9.3. For each possible non-converse entry (upper triangle relation) in each chromosome's solution table, a roulette-wheel selection system (described in section 9.1.6) chooses one of the RCC-8 relations guided by the weightings in δ. Hence, higher weighted relations are more often chosen than those lower weighted relations. As a result, each candidate solution should follow a similar ratio and distribution of topological spatial relations to those specified in δ.

It is possible to envisage a more intelligent initialisation method whereby certain expert knowledge is used to produce a better set of initial chromosomes. For example it is possible, algorithmically, to consistently and deterministically assign a *proper part* (PP) hierarchy (assuming NTPP or TPP have a weighting greater than 0 in δ) thus producing an initial partially-consistent configuration. Indeed similar has been used in [233] to generate partially consistent visualizations of topological constraints.

9.1.5 FITNESS FUNCTION

The set of RCC-8 constraints Θ are evaluated using a path-consistency algorithm. Path consistency is determined over the table of region relations for each chromosome in the population using a standard $O(n^3)$ *path-consistency* algorithm such as those described in section 2.4.3. The exact fitness function used is outlined in Appendix C.1.

As the genetic algorithm only ever generates definite topological relations (only one RCC-8 base relations holds between any two regions x and y), consistency checking is deterministic, no backtracking is required and hence we can guarantee

*lower/upper

that the scene is globally consistent [189]. The fitness score assigned to a chromosome is a count of the number of constraints that are violated. More formally:

$$fitness(\Theta) = \sum_{i=1}^{|n|}\sum_{j=1}^{|n|}\sum_{k=1}^{|n|} \begin{cases} \text{error=error+1} & (R_{ij} \cap (R_{ik} \otimes R_{kj}) = \theta) \\ \text{error=error} & \text{otherwise,} \end{cases} \tag{9.4}$$

Where R_{ij} represents the topological relation between regions assigned to the variables i and j. Importantly, more than one path (region triples i, k, j) between the regions i and j may indicate inconsistency, where each still increases the error count.

9.1.6 Selection Function

The genetic algorithm uses a vanilla form of *roulette-wheel* selection in addition to a *best* (in this case the top two) selection policy. The top two chromosomes are copied directly into the new generation without reproduction (crossover or mutation). All other chromosomes are selected using *roulette-wheel* selection, reproducing new child chromosomes using the crossover and mutation functions described below.

Crossover A crossover point is chosen at random between 1 and the number of regions in the scene (single point crossover). The crossover point corresponds to a table column which is the origin of the swap. For example assuming the following chromosomes c_1 (table 9.2) and c_2 (table 9.3) and a crossover point of 2, the offspring o_1 (table 9.4) and o_2 (table 9.5) are produced from the crossover function. Note that only the upper triangle is swapped, non-converse relations are at this stage are empty.

	A	B	C	D
A	EQ	EC	PO	DC
B	–	EQ	EC	DC
C	–	–	EQ	NTPP
D	–	–	–	EQ

Table 9.2: Chromosome c_1

	A	B	C	D
A	EQ	NTPP	NTPP	PO
B	–	EQ	PO	TPP
C	–	–	EQ	DC
D	–	–	–	EQ

Table 9.3: Chromosome c_2

Mutation As with crossover, mutation is classical in the sense of being based on a simple randomised mutation function. The function is summarised in Appendix

	A	B	C	D
A	EQ	EC	NTPP	PO
B	–	EQ	PO	TPP
C	–	–	EQ	DC
D	–	–	–	EQ

Table 9.4: Offspring o_1

	A	B	C	D
A	EQ	NTPP	PO	DC
B	–	EQ	EC	DC
C	–	–	EQ	NTPP
D	–	–	–	EQ

Table 9.5: Offspring o_2

C.4.2(where a mutation rate of 2% was used as it was found to work best in practice).

9.1.7 Termination

Two termination strategies are employed. The first is to return a spatial configuration when any candidate solution (chromosome) has a fitness of 0 (consistent). Alternatively, if an inconsistent scene is required, the genetic algorithm can terminate when a pre-specified fitness score is met. On termination the lower triangle matrix (converse relations to the upper triangle matrix) are added to the final solution using lookup to an RCC-8 converse relation table.

Results The genetic algorithm was able to produce instantiated *SWSRLO* geo-ontologies from 2 regions up to 25 regions using all 8 RCC base relations, and up to 200 regions when restricted to using only the RCC-8 base relations EC, $NTPP$, $NTPP^{-1}$ and DC.

9.2 Instantiating *SWSRLO* from the Web

In this section we explore a Wikipedia based web mining technique to extract qualitative spatial relations and regions with which to instantiate different *SWSRLO* geo-ontologies.

We do not attempt to acquire spatial footprints for places from web sources for the following reason. It is often not possible to properly ground and assign a spatial footprint to all candidate places on the web using freely available gazetteer sources (Geonames, GNIS etc), as these sources are incomplete. Secondly, most free gazetteer sources used for grounding only contain point referenced footprints. Point based geometry is to coarse to determine containment, overlap and equality relations important in constructing a rich geo-ontology. Consequently we rely on official data sets for spatial footprints. Natural language content is however a good source of spatial expressions from which to extract qualitative spatial relationships. For example, from the spatial expression; *Cardiff is bordered to the east by the city of Newport*, it is possible to extract that Newport is east of Cardiff, and Cardiff and Newport are adjacent.

Automating the extraction of geographic place information from web pages requires a level of machine understanding using natural language processing techniques (NLP). Recently within the field of Geographic Information Retrieval (GIR) NLP techniques are being used to locate geographic information from web sites (including social web sites*), and add that information to suitable models / ontologies. Automatic construction of geographic gazetteers (helping to overcome the manual effort required to construct geographic ontologies) is an active research domain [23, 205, 259]. These techniques use a variety of free resources, for example classical gazetteer resources such as GNIS†, GNS‡, Geonames§, and newer social websites such as Wikipedia, Geograph and Flickr.

For the purpose of this thesis, we aim to construct *SWSRLO* geo-ontologies from Wikipedia articles. Articles are mined for information about place and their spatial relations using a variety of standard NLP techniques along with a number of Wikipedia specific heuristics. The section to follow describes these approaches in more detail.

9.2.1 MINING WIKIPEDIA

Wikipedia is a massive, user contributed, hyper-linked corpus of semi structured text content and, importantly for this work, Wikipedia contains around 1,000,000 georeferenced articles [205]. Much recent research has focused on the extraction of information from Wikipedia, for example [150, 49]. Due to the inherent semi-structure of each Wikipedia article, providing effective parsing methods for knowledge extraction is easier than unstructured web content. In particular, Wikipedia has a number of unique characteristics useful to help locate and disambiguate ge-

*Extraction of content from social websites is sometimes referred to as crowd sourcing [95]
†`geonames.usgs.gov/pls/gnispublic/`
‡`earth-info.nga.mil/gns/html/index.html`
§`www.geonames.org`

ographic information (information about places). These are exploited to differing degrees in a number of works [205, 198, 17, 199]:

- Page titles provide entity names and subject identification - each article is unambiguous in the sense that it only contains information about one subject, and that subject is identified by the page title.
- *Info boxes* often contain concise geospatial information about the place (assuming the subject of the article is geographic) including a point based latitude and longitude coordinate. Such *info boxes* form a minimal gazetteer entry [17].
- Page categories help to identify candidate places and the feature type of place entries i.e. is it a country, district etc,
- Wikipedia articles are, unlike arbitrary web pages, more tightly focused around the article subject, and hence are more likely to contain relevant co-occurring place names. This helps to build robust co-occurrence models to aid the disambiguation process.
- Often the first sentence of an article contains a geographic concept e.g. *is a neighbourhood*, *is a district*. These help to determine if the content is geographic or not [148].

Place information can also be extracted from the free text content of each Wikipedia article using standard NLP techniques as described in section ??. Of course not all Wikipedia articles are standardised to include all the above types of information, and hence any mining technique will have variable success depending on the richness of the mined article.

Much of the interest surrounding place name extraction from Wikipedia has led to the development of Placeopedia* and Wikimapia†. Placeopedia aims to link wikipedia articles to their spatial locations - providing a further grounding to Wikipedia articles - although the tie-up is performed by users, and the accuracy of user contributed relationships, and indeed the accuracy of all wikipedia articles is suspect in some cases [92]. Wikimapia is an on-line editable map where users

*http:/www.placeopedia.com/

†http://wikimapia.org/

can describe places by associating a minimal bounding rectangle (drawn during place entry) with a place name. However, as noted in [205] their approach suffers from poor place type categorisation and they lack stringent control over entries such that some entries are not entirely geographic.

In the following sections we refer to the process of finding a place name candidate from Wikipedia as partial place name resolution - a slight modification of the term used in [163]. A fully resolved place name candidate would be one which is known to exist (typically from a gazetteer source), has been disambiguated and has been grounded (assigned a spatial footprint). A partially resolved place name candidate is one which is known to exist and is contextually unambiguous, but has not been assigned its spatial footprint. Note that in our approach trying to extract and ground places, fully resolving them, is not necessary, as we are only interested in qualitative spatial relations. Consequently, we only need to ground and disambiguate place names to the extent that, in the context of all minded places and relations, each same named place refers to the same location on the earth e.g. Cardiff in Wales as opposed to Cardiff in Australia. Otherwise the wrong relations may be attached to the wrong place e.g. Wales in Australia is inside the UK. We also limit the task of disambiguating place name candidates by limiting the scope of the geographical area of web documents mined to that of the area of Glamorgan and Cardiff in the United Kingdom.

9.2.1.1 Wikipedia Extraction Technique

We use Wikipedia to extract qualitative spatial relations for two geo-ontologies. The first is about places in Cardiff districts and electoral Wards, and the second augments these with information about the slightly larger area of the County of Glamorgan. Districts are smaller unofficial areas which, in Wales, are either used in place of wards (but are the same geographical area) or refer to parts of wards. Wikipedia articles about wards and districts are often tightly packed with rich spatial information. The extract shown below from the Cathays electoral ward Wikipedia article, shows that containment relations and adjacency relations are clearly evident from sentences which start with *consists of* and *It is bounded by* respectively*. In order to extract spatial sentences of the from described above,

*Clearly there are also a significant number of cardinal direction relations

Example Wikipedia Extract for Roath Electoral Ward: spatial phrases are shown in **bold**, place names are shown in *italics*
The *Cathays* electoral ward of *Cardiff* **consists of** some or all of the following areas: *Blackweir*, *Cardiff city centre*, *Cathays*, *Cathays Park* and *Maindy* in the the parliamentary constituency of *Cardiff Central*. **It is bounded** by *Gabalfa* and *Heath* to the north; *Plasnewydd* and *Adamsdown* to the east; *Butetown* to the south; and *Riverside* to the west. The *River Taff* forms its western boundary to where it meets the South Wales Main Line, the South Wales Main Line forms the southern boundary to where it meets the Valley Lines northbound branch, this railway line then forms the eastern boundary as far as the *A48 road*

a number of heuristics, as shown in Table 9.6, have been developed. Heuristics 1-3 match to spatial sentences that begin with either, *falls within*, *bounded by* or *it covers*, and end with a full stop (a complete sentence). Heuristic 4 again matches to a spatial sentence which begins with *consists of* but ends with either the spatial preposition *in* or a full stop. Curtailing a sentence when reaching the word *in* is important to avoid extracting sentences that describe both contains and part of (P^{-1}) relations. For example (as taken from Table 9.2.1.1),"The Cathays electoral ward of Cardiff consists of ... and Maindy **in** the the parliamentary constituency of Cardiff Central", where the relation after the spatial preposition *in* is a containment relation (P). Heuristic 5 represents a common and often successful assumption in named entity recognition (NER) tasks. It states that any proper noun is a candidate place name (a proper noun in the English language is either a named place, a named person or more generally a *thing*). Determining spatial

Wiki Heuristic No.	Pattern	Extracted Relations
1	$falls\ within[\backslash.]^*$	P - part of
2	$bounded\ by[\backslash.]^*$	EC - adjacent
3	$It\ covers[\backslash.]^*$	P^{-1} - contains
4	$consists\ of((?!\backslash s\ in \backslash s).)^*$	P^{-1} - contains
5	$(([A-Z]\{1\}[a-z]+\backslash s^*)\|([A-Z]\{2\}))$	proper noun pattern, placenames

Table 9.6: Wikipedia specific heuristics (as regular expressions)

relations from free text content is then a three step process. Step one extracts sentences from each Cardiff district / ward and Glamorgan county Wikipedia article using heuristics 1-4. Step two extracts candidate place names using the proper noun regular expression 5, and then assigns each candidate place name a relationship with the subject place name of the wikipedia article.. The last step attempts to partially resolve each candidate place name, and any un-resolvable candidate place name is removed.

In addition to spatial information contained in free text content, we exploit neighbourhood tables for ward and district pages. A neighbourhood table is a 3x3 table that contains the articles subject place name (ward or district) in the centre of the table, surrounded by other districts or wards, where their location in the table indicates a cardinal direction from the subject place, and their entry in the table indicates an adjacency relation. An example 3x3 neighbourhood table is shown in Figure 9.3. Each entry in the table is a strong place name candidate. That

Heath	Roath Park	Penylan
Cathays	Roath	Tremorfa
City centre	Adamsdown	Splott

Figure 9.3: Example Roath district 3x3 neighbourhood table taken from Wikipedia

is, we assume the table only contains place names, albeit apart from erroneous or null table entries which must be filtered (e.g. a blank entry, or an entry that represents a cardinal direction i.e. NW). From each neighbourhood table we glean both adjacency and cardinal direction relations (of which only adjacency relations are used in *SWSRLO*). Unfortunately, 3x3 neighbourhood tables are rare and, as such, are not included as standard across all Wikipedia ward and district articles. Luckily for the purpose of generating test scenes, such tables are ubiquitous across wards and districts within Cardiff.

Before we describe the complete extraction process, we first construct a place name list (denoted Pl), from two Wikipedia sources, to form a minimal gazetteer (minimal in that the list does not contain any locational information) with which to partially resolve candidate place names:

(a) Both ward and district and county category pages contain an easily parseable table of ward names and district names. Each of these names, along with Cardiff (the containing region) and Glamorgan, are added to *Pl*.

(b) Place names obtained from the filtered neighbourhood tables (filtered for null entries) are added to *Pl*. This works under the assumption that we can assume neighbourhood tables to only contain valid place names.

The resultant list *Pl* is, in effect, a small list of place names compared to existing gazetteers with over 6 million place names. Hence, the problem of partially resolving place names becomes easier. The complete Wikipedia spatial relation extraction process is as follows:

(a) Filter all Wikipedia articles, leaving only articles from the categories; Cardiff Electoral Wards and Districts and the Glamorgan County pages. By initially filtering Wikipedia articles, we reduce the polysemy count of place names (ambiguity) to 1, and hence all place names are unambiguous. These document sets are denoted C_w, C_d and C_g respectively. The complete document set C is then formed from the union of ward, district document sets with the Glamorgan county page: $C = C_w \cup C_d \cup C_g$.

(b) Remove any wiki markup from each article in each document set. For example removing links to other Wikipedia articles or redirects ([[<page-name>]] and [[<alt-page-name> — <page-name>]]), and removing headings and subheadings ("<heading>").

(c) For each article a_1,...,a_n in C_d (articles in C_w and C_g do not contain neighbourhood tables), extract the 3x3 neighbourhood table, then:

  (i) Remove null entries and cardinal direction string entries.

  (ii) Use the middle table entry as the subject place name.

  (iii) Relate each remaining table entry candidate place name to the subject place name assuming the adjacency relation (*EC*) and add to a triple of the form {subject-place name, *EC*, candidate-place name}.

  (iv) Add each triple to the set *Rels*.

(d) For each article a_1,...,a_n in C use heuristics 1-4 and do:

  (i) Use the heuristic to extract a spatial sentence, *Sp*.

(ii) Extract place name candidates from Sp using the proper noun heuristic 5.

(iii) Take the subject place name from the page title and add each extracted place name candidate to a triple of the form {subject-place name,relation,candidate-place name} (and add it to *Rels*), where relation is an RCC relation that is appropriate for the used heuristic.

(e) For each triple in *Rels*, partially resolve the candidate place name. If the candidate place name can not be resolved the tuple is removed from *Rels*.

Spatial relations about wards extracted from Wikipedia can be matched (by exact name match) to the ward level of polygons extracted in section 9.3 - allowing for mixed quantitative and qualitative reasoning through the hybrid architecture of *SWSRL*.

9.2.2 RESULTS

Running the complete extraction process over the set of Wikipedia Cardiff District and Wards Wikipedia articles results in 198 spatial relations (in particular P, P^{-1}, and EC RCC relations) between 74 distinct wards and districts. Adding information about Glamorgan county increases this set to 79 regions, and 220 relations.

9.3 INSTANTIATING GEO-ONTOLOGIES FROM OFFICIAL DATASETS

To test the proposed hybrid architecture, the LSS needs to be populated with the geometry of some, or all, of the regions in *SWSRLO*. To do this we use two versions of an official administrative hierarchy. The first represents the regions that form part of Cardiff, and the second represents administrative regions in South Wales along with the boundary of Wales. The Cardiff hierarchy is constructed from the administrative subdivisions; Wards, Civic Parishes, and Unitary Authorities from the Ordnance Surveys 2001 Census Administrative Super Output wards/parishes (CASWA/CASPAR)*. For the area of South Wales, only the Unitary Authorities are used. The boundaries of Wales are made up of the outlines of all regions that together form the country. The formal definitions of these subdivisions are shown in Appendix A.2.

Wards and Communities, which are at the same hierarchical level, were chosen for the Cardiff geo-ontology because Communities and Wards can cross boundaries, and therefore provide overlap relationships. The Unitary Authority, in this case Cardiff, provides a part-of hierarchy useful for extracting containment relations (TPP,TPP^{-1},$NTPP$,$NTPP^{-1}$ RCC-8 base relations). As only Unitary Authorities are used for the area of South Wales, we sacrifice most containment and overlap relations, but introduce a much larger number of regions.

Each dataset contains polygonal geometry along with basic naming attributes which are added directly to the LSS. The three datasets, which are supplied in ESRI's Shapefile vector data format, are converted to an Oracle Spatial database using the GeoTools* Java library. Each administrative subdivision (ward etc.) in the shapefile is converted into a tuple {RDF:ID,Shape}, where RDF:ID represents the name of the administrative subdivision and Shape stores its geometry. Maps of the administrative subdivisions of Cardiff, South Wales and the boundaries of Wales stored in the LSS are shown in Figures 9.4 and 9.5 respectively.

Qualitative topological relationships that correspond to the geometric representation of the administrative subdivisions are determined using Oracle's spatial `SDO_Relate` operator, and then added to $SWSRLO$. More specifically, the procedure for converting the each offical dataset is as follows:

(a) For each administrative subdivision in the LSS, determine its topological relation with every other administrative subdivision in the LSS including itself, using the following Oracle Spatial SQL query:

```
SELECT c_b.RDFID, c_d.RDFID, SDO_GEOM.RELATE(c_b.shape, 'RELATION',
 c_d.shape, 0.005)  FROM locationBase c_b, locationBase c_d
```

(b) Attach the qualitative topological relations to the corresponding (identical RDF:ID's) administrative subdivision in $SWSRLO$.

*©Crown Copyright/database right 2008. An Ordnance Survey/EDINA supplied service
*http://geotools.codehaus.org/

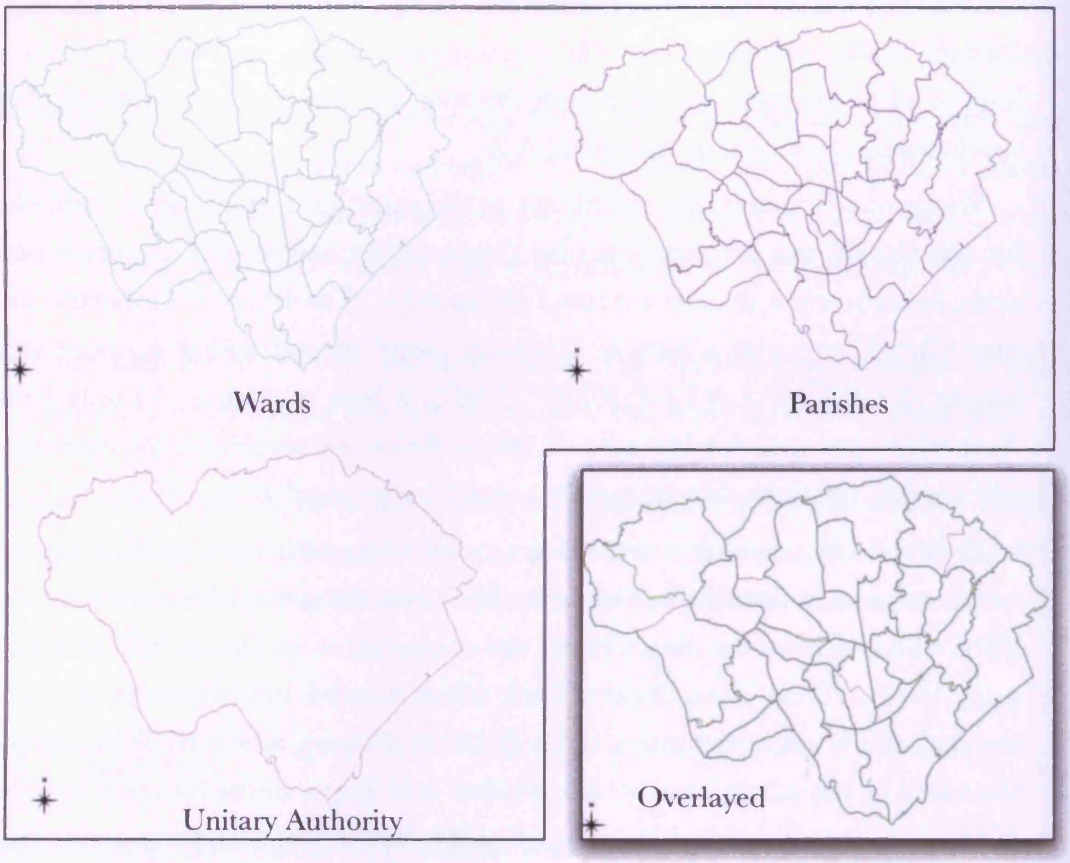


Figure 9.4: A map of Cardiff Parishes, Wards and Unitary Authorities

9.3.1 COMBINING GEONAMES

Geonames[†] is an online, evolving, placename (toponym) ontology, that contains over 8 million geographical names[‡], their location and other attributes. Geonames is used here because it is more likely to contain erroneous locational information and hence topological relations compared to the Ordnance Survey data sets and thus provides a solid test case for *SWSRL*.

The Geonames toponym ontology is freely available and its model is illustrated in Appendix A.4. An OWL/RDF version of the ontology exists which we convert to an *SWSRLO* geo-ontology. Importantly, we do not convert the entire ontology, instead only concentrating on administrative entries (those with feature code of

[†]http://www.geonames.org
[‡]http://www.geonames.org/about.html

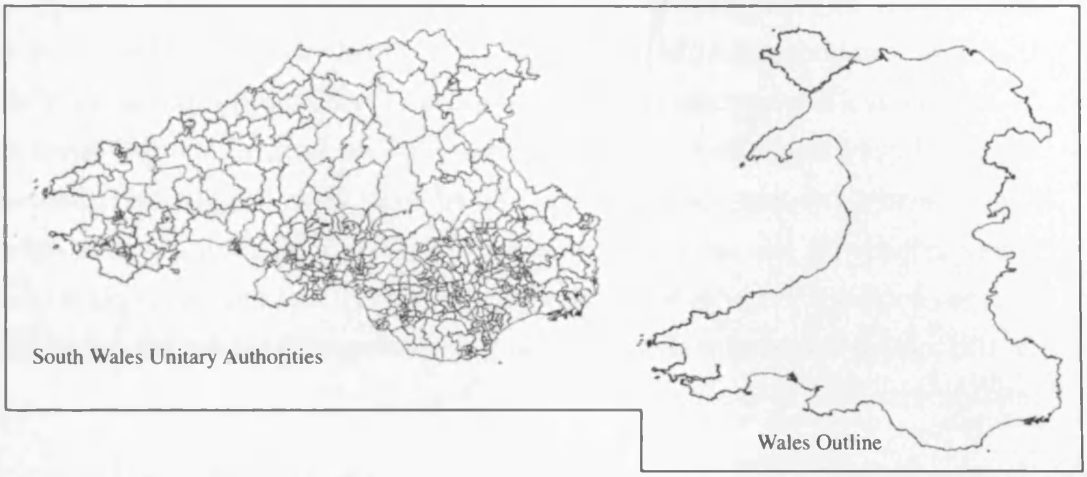


Figure 9.5: A map of South Wales Unitary Authorities and the oultine of Wales

A.ADM1 in Geonames) that fall within a rough bounding box of south Wales - in total 15 features. Furthermore, we extract the geometry for each toponym (latitude and longitude field) and store it inside the LSS.

In order to detect errors within the Geonames ontology it is combined with the Wikipedia and official boundary of Wales geo-ontologies, as described in section 9.4.

9.4 Combining Synthetic, Wikipedia, and Official Datasets

Wikipedia and the official administrative subdivisions of Cardiff are integrated into combined *SWSRLO* ontologies as well as left as individual *SWSRLO* geo-ontologies. Wikipedia, Geonames and the official boundary of Wales are also combined into a single *SWSRLO* geo-ontology. The official administrative areas of South Wales are left as a standalone geo-ontology for the purpose of large scale testing. Synthetic geo-ontologies can be derived as an when needed using the spatial genetic algorithm, and are left as individual *SWSRLO* geo-ontologies.

Combining Wikipedia and the Administrative Subdivisions of Cardiff: Individuals from the official administrative subdivision geo-ontology are matched, syntactically by name, to those individuals from the Wikipedia geo-ontology. Furthermore, combining the official administrative subdivision geo-ontology with the Wikipedia geo-ontology requires consideration of conflicting topological relations (hence already suggesting inconsistencies in one or both of the geo-ontologies). Consequently, two geo-ontologies are created from their combination, one which places priority on the topological relations in the official administrative subdivision geo-ontology (hence keeping the official relation), and one which place priority on the topological relation in Wikipedia geo-ontology (hence replacing the official relation with that of the Wikipedia geo-ontology).

Combining Geonames, Wikipedia and the Administrative Boundary of Wales: To combine the Geonames, Wikipedia and and administrative boundary of Wales geo-ontologies, similar features that occur in all three sources need to be matched to create a single, unified entity to reason over.

Geonames individuals that have Wales as a parent are linked via a part-of topological relation (P) to the official boundary of Wales. Individuals and their relations from Wikipedia including the County of Glamorgan geo-ontology are then matched by standard name (with some manual filtering) to those individuals from the combined Geonames and Administrative Boundary geo-ontology.

9.5 OVERVIEW OF ALL *SWSRLO* GEO-ONTOLOGIES

Table 9.7 shows *SWSRLO* geo-ontologies that has been generated from the techniques described in this chapter, as then used in Chapters 10 and 11 to follow.

Number	*SWSRLO* Geo-ontology	Number of Regions	Number of Qualitative Topological Relations	No. of Regions with Polygonal Geometries
1	SyntheticGA (SGA)	1 to 100	1^2-100^2	0
2	Official Administrative Subdivision of Cardiff (OAS)	62	3844	62
3	Wikipedia geo-ontology (WikiGeo)	74	198	0
4a	Wikipedia geo-ontology including Glamorgan entries (WikiGeoGlam)	79	220	0
4	Geonames and WikiGeoGlam and Boundary of Wales (Geonames-WikieGeo-Wales)	95	235	1 (Boundary of Wales) 15 (Geonames)
5	OAS+WikiGeo(Primary)	96	3936 (18 WikiGeo relations replaced conflicting relations in OAS)	62
6	Official Unitary Authorities of South Wales (OAS-large)	537	323761	537

Table 9.7: All generated *SWSRLO* geo-ontologies

Chapter 10

Application and Results

This chapter evaluates the developed Semantic Web Spatial Rule Language *SWSRL* by showing its application over the real world instantiated geo-ontologies created in Chapter 9 . In particular this chapter will demonstrate and evaluate the following prominent components of the geo-ontology maintenance framework:

- Employing *SWSRL* over an *SWSRLO* geo-ontology for the purpose of integrity maintenance.
- Employing *SWSRL* over an *SWSRLO* geo-ontology for the purpose of deduction.
- Employing combined quantitative and qualitative reasoning in deduction and integrity rules in *SWSRL* over *SWSRLO* geo-ontologies, thus demonstrating the proposed hybrid geo-ontology maintenance framework.
- Testing the accuracy of relational confidence and compositional confidence error localisation techniques.

In overview this Chapter is broken down into four major evaluation tasks. Further empirical testing of the language and reasoning engine is given in the next chapter. These four major evaluation areas are split into the following sections (where the name an number of each geo-ontology used in each evaluation is shown in table 9.7):

- Section 10.1 determines topological scene consistency over a complete topological scene description in the OAS(2) geo-ontology using both deduction and integrity rules. The results are improved by considering the `validTR` builtin which helps eliminate error propagation and hence provides a better indication of the source of the inconsistency (unsatisfiable topological constraint / relation).

- Section 10.2 evaluates full integrity maintenance reasoning over real world WikiGeo(3) and OAS-WikiGeo(5) geo-ontologies using forward reasoning. These tests also evaluate the relational confidence and compositional confidence methods in their ability to automatically locate errors and suggest new relations to rectify errors.

- Section 10.3 evaluates full integrity maintenance reasoning over the real world Geonames-WikiGeo-Wales(4) geo-ontology using interleaved reasoning which includes both qualitative relations and quantitative geometry in the LSS.

- Section 10.4 shows a topo-semantic integrity rule, with a view to motivate future work in this area.

10.1 Determining Inconsistencies from the OAS *SWSRLO* geo-ontology

Test Number: 1
Test Ontology:(2)
Rule Set(s): $F^{RCC12}_{<D,I>}$
Purpose: Deciding consistency of a definite set (only RCC-8 base relations) of topological relations in *SWSRLO* using deduction and integrity rules

Importantly, the system must be able to first determine if a complete or partial scene description is either consistent or inconsistent. That is, testing whether the set of topological relations that exist inherently in *SWSRLO* is satisfiable or not. As a first step we will consider only determining consistency for complete scene descriptions that only have definite or base RCC-8 relations between regions. In a later step we will consider the same technique to determine consistency for partial scene descriptions.

10.1 Determining Inconsistencies from the OAS *SWSRLO* geo-ontology

This test employs the OAS geo-ontology(2), shown before reasoning in the geo-ontology instance view (GIV) in Figure 10.1. An inconsistent relation $NTPP^{-1}$ is explicitly added between Radyr and Cardiff, overwriting the existing relation $NTPP$, as illustrated in Figures 10.2 and 10.3.

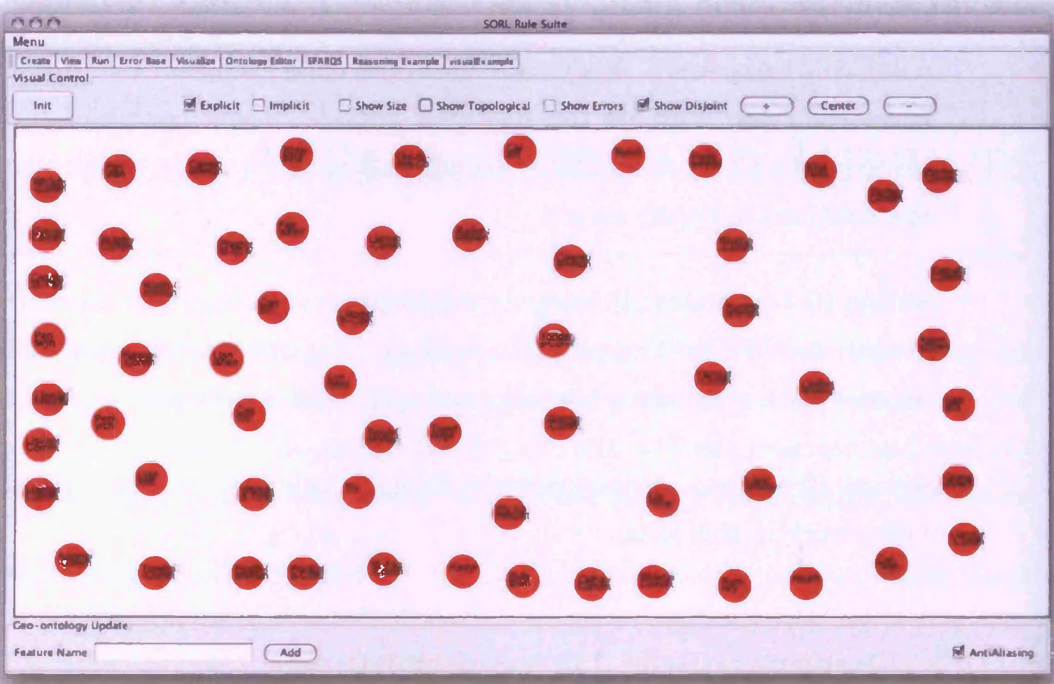


Figure 10.1: The geo-ontology instance viewer showing the Wards, Parishes and Unitary Authority of Cardiff, where topological relations (edges) are not shown

Figure 10.4 then illustrates (in the GIV) the errors created by the insertion of the inconsistent relation between Radyr and Cardiff after execution of the$F^{RCC12}_{<D,I>}$ ruleset, along with the errors found as a result of the propagation of that inconsistency to other relations in the network. In total there are 9594 relational inconsistencies detected

Clearly employing deduction rules in addition to integrity rules (as is the case with the $F^{RCC12}_{<D,I>}$ ruleset) leads to error propagation. That is, one erroneous relation in *SWSRLO* will propagate around the geo-ontology, thus generating inconsistencies between most connected (directly or indirectly) regions. Consequently, at this stage we only aim to determine if an inconsistency exists, as opposed to showing the likely source of the error - although in these example cases the erroneous

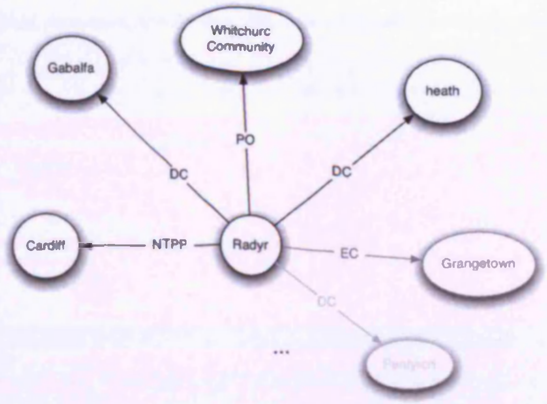


Figure 10.2: An illustration of a subnetwork of the initially consistent *SWSRLO* geo-ontology

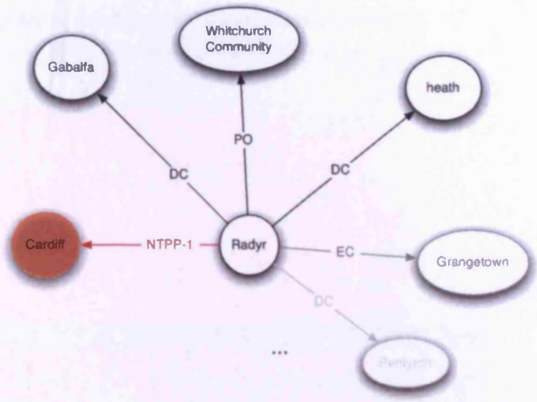


Figure 10.3: An illustration of a subnetwork of the now inconsistent *SWSRLO* geo-ontology

relation is clearly the one manually inserted.

10.1.1 LOCALISING THE ERROR - *validTR* BUILTIN TESTING

Test Number: 2

Test Ontology:(2)

Rule Set(s): F_I^{RCC12} and F_D^{RCC12} with `validTR`

Purpose: Deciding consistency and reducing error propagation

One way to stop errors from propagating in *SWSRLO* is to remove the deduction ruleset F_D^{RCC12} and only run the integrity ruleset F_I^{RCC12}. As a result, only checking the consistency of existing relations (closing the set Θ of relational constraints under intersection only). For the OAS geo-ontology, this is sufficient to decide consistency as every region is related to every other by a definite relation [189].

The result of applying only F_I^{RCC12} integrity rules to the OAS *SWSRLO* geo-ontology is shown in Figure 10.5. In this case it is clear that all inconsistencies are either between the regions Radyr and Cardiff, or are produced from immediate compositions of it. Being able to determine the exact source of an inconsistency (as in this scenario) is a best case for integrity checking rules.

Assuming that most geo-ontologies are not complete, full reasoning is required

Figure 10.4: Geo-ontology instance viewer visualising all inconsistent relations after the insertion of the erroneous relation inserted between Radyr and Cardiff

using both integrity and deduction rules (the set $F^{RCC12}_{<D,I>}$). In this case the `validTR` predicate is used (as defined in Chapter 8) within the ruleset F^{RCC12}_{D} to help minimize the effects of error propagation. Figure 10.6 depicts the number of invalid relations found from the inconsistent OAS *SWSRLO* geo-ontology when using deduction rules that contain the `validTR` predicate. The number of errors detected has reduced from 9594 in Figure 10.4, to 363 in Figure 10.6. This is an important result and will help increase the accuracy of both compositional confidence and relational confidence methods evaluated later in this chapter.

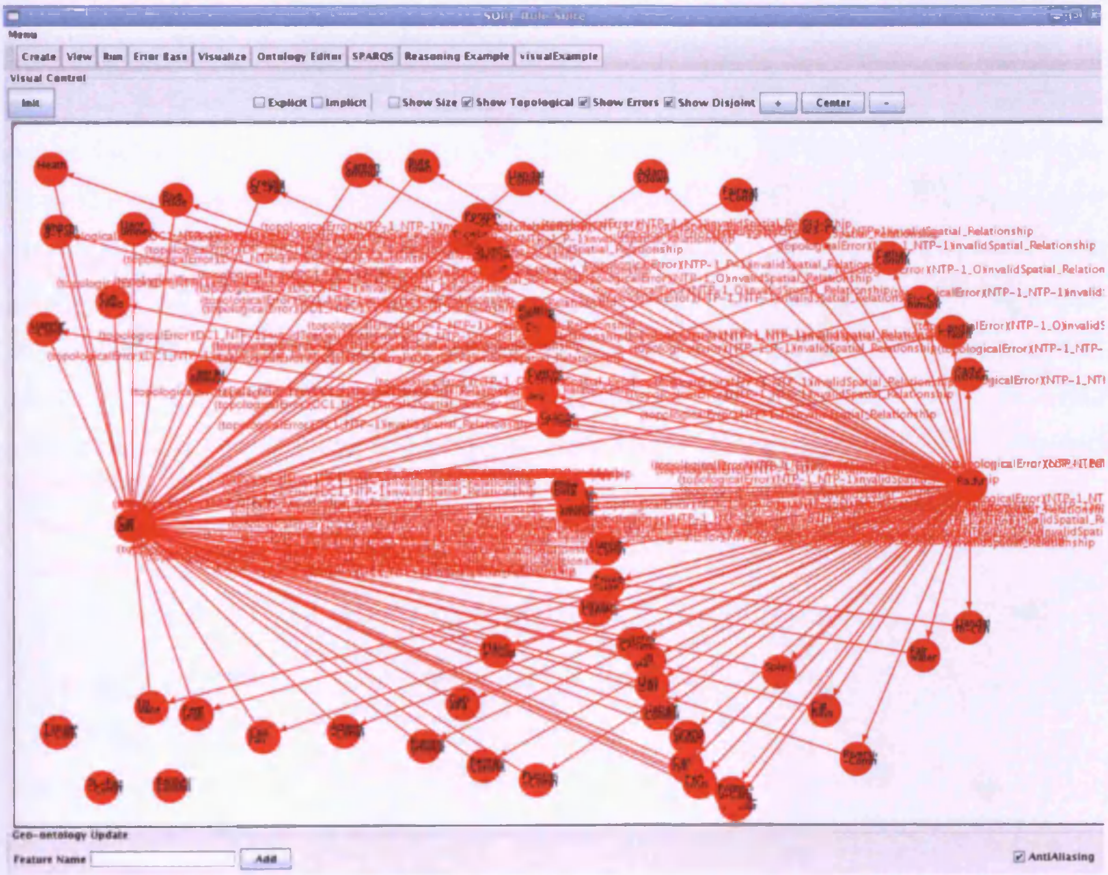


Figure 10.5: Geo-ontology Instance View visualising the inconsistent relations between Radyr and Cardiff and any immediately connected regions

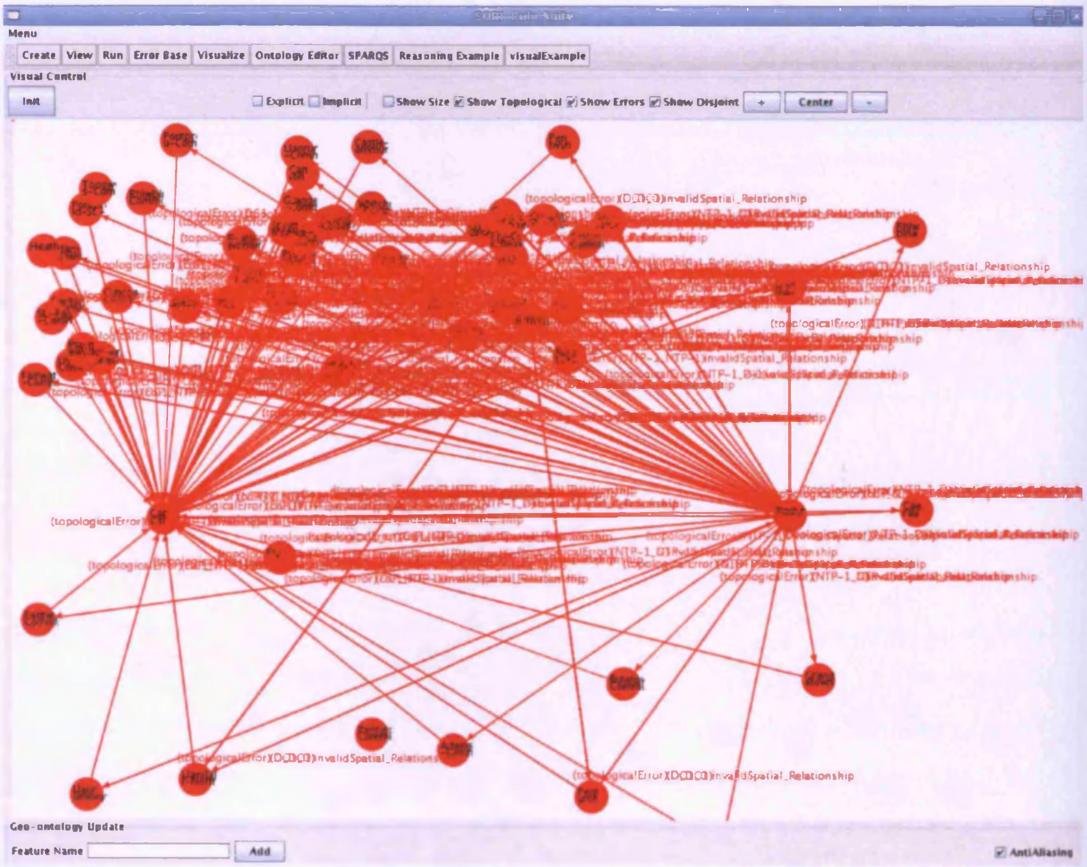


Figure 10.6: Geo-ontology Instance View visualising the inconsistent relations between Radyr and Cardiff when reducing error propagation using look ahead deduction rules

10.2 REALISTIC APPLICATION TESTING

In this section we test the application of topological spatial reasoning, relational confidence and compositional confidence to identify and locate inconsistencies in the real world *SWSRLO* geo-ontologies WikiGeo(3) and OAS+WikiGeo(Primary) (5).

For each geo-ontology we show inconsistencies (simply termed errors) derived during reasoning from the deduction and integrity ruleset $F^{RCC12}_{RecordSplit<D,I>}$. In addition, to determine the exact source of an error, a derivation trace of each evaluated fact in each violated integrity rule is examined, to highlight the raw facts from the place ontology that led to the derivation of each error. A complete trace of a violated integrity rule is shown in Table 10.1 as an example. Concise versions of these tables are shown for each tested geo-ontology* and each error is manually investigated and marked as inconsistent using knowledge of official boundary relations. The results of error localisation using both relational confidence and compositional confidence, which do not use this trace, are shown and evaluated against the results of the manual investigation using these error traces.

10.2.1 TESTING THE WIKIPEDIA GEO-ONTOLOGY

Test Number: 3
Test Ontology:(3)
Rule Set(s): $F^{RCC12}_{RecordSplit<D,I>}$ with `validTR`
Purpose: Deciding consistency of a real world geo-ontology and locating the source of any inconsistencies

In this section we test the integrity of the WikiGeo(3) *SWSRLO* geo-ontology using both integrity and deduction rules which includes the `validTR` and `record` builtin.

Violations: Table 10.2 shows the raw relations in the WikiGeo go-ontology (found using the previously described rule tracing method) that triggered and hence violated certain integrity rules in $F^{RCC12}_{RecordSplit<D,I>}$ - the exact rules violated

*These tables are automatically produced by the system and are shown in the Error Tracing View of the visual interface

Error Information	Details
Error	ns:Llandaff ns:Llandaff ns:Butetown (Rule:P_DC1-3)
Rule	[P_DC1-3:(?A P ?B) (?B DC1 ?C) (?A C ?C) (?A rdf:type Region) (?B rdf:type Region) (?C rdf:type Region)→ error(P_DC1 ?A topologicalError ?C ?B ?count invalidSpatial_Relationship P_DC1-3)]
FailedRelation	C
Existing Blocking Relaiton	ns:Llandaff :C: ns:Butetown
Trace of Existing	Rule ci concluded (ns:Llandaff ns:C ns:Butetown) ← Rule ec1Map concluded (ns:Butetown ns:C ns:Llandaff) ← **Fact** (ns:Butetown ns:EC ns:Llandaff)
A to B (Composition)	ns:Llandaff P ns:Llandaff Rule eq1Map concluded (ns:Llandaff ns:P ns:Llandaff) ← **Fact** (ns:Llandaff ns:EQ ns:Llandaff)
B to C (Composition)	ns:Llandaff DC1 ns:Butetown Rule tpp3 concluded (ns:Llandaff ns:DC1 ns:Butetown) ← **Fact** (ns:Llandaff ns:DC ns:Butetown)

Table 10.1: Example error trace of a single topological inconsistency in the Wiki-Geo geo-ontology

No	From	Relation	To	Correct Relation
1	ns:Llanishen	ns:P-1	ns:Llanishen	✔
2	ns:Llanishen	ns:P-1	ns:Thornhill	✔
3	ns:Llanishen	ns:EC	ns:Thornhill	×
4	ns:Culverhouse-Cross	P	ns:Ely	✔
5	ns:Ely	ns:EC	ns:Culverhouse-Cross	×
6	ns:Llanishen	ns:EC	ns:Birchgrove	×
7	ns:Llanishen	ns:P-1	ns:Birchgrove	✔
			Total Incorrect Relations	3

Table 10.2: Automatically determined topological inconsistencies in the WikiGeo geo-ontology

can be seen in their full rule traces but not in Table 10.2 for the sake of brevity. The fourth column in the table shows if the relation is known, by manual evaluation, to be correct or incorrect.

Table 10.3 shows the results of the compositional confidence technique which tries to automatically locate the source of these errors within the geo-ontology. The technique suggests 3 edges (region to region pairs) that are possible sources

From	To	Count	Rank
ns:Llanishen	ns:Thornhill	2	2
ns:Llanishen	ns:Birchgrove	2	2
ns:Ely	ns:Culverhouse-Cross	1	3

Table 10.3: Compositional confidence counts for errors in the combined WikiGeo geo-ontology

of inconsistent relations. These 3 edges are between the same regions that were identified manually by tracing the derivation logs shown in Table 10.2. Therefore the accuracy is shown here to be 100%. Importantly, our technique was able to do this without the aid of a derivation trace, which is a feature of Jena2 and is not always available.

The incorrect relation (EC) between Ely and Culverhouse Cross is ranked lower than the other two inconsistencies. This can be largely attributed to the fact that this incorrect relation does not propagate very far in the network causing further inconsistencies, a reflection of the lack of connectedness of the region Culverhouse-Cross with other regions - where Culverhouse Cross is only connected to Ely in the raw data. Table 10.4 shows the source of each inconsistent relation from their originating Wikipedia articles. All 3 errors are results of direct conflicts between topological relations in different articles.

A manual evaluation of all 198 topological relations from the Wikipedia geo-ontology determined that, in total, there existed 16 topological inconsistencies. Therefore only 18.75% of all inconsistencies where automatically found during reasoning. There are a possible 3600 topological relations between all 60 Wards and Districts in WikiGeo (n^2 where n is the number of Wards and Districts). With only 5.5% (198) of these relations extracted from Wikipedia, the relatively low precision (18.75%) in detecting errors is understandable. It is likely that if other topological relations where added, many more of these inconsistencies would be found - such case is tested in the section to follow by combining the OAS and WikiGeo geo-ontologies.

Relational Confidence: Table 10.5 shows the ranking of relations produced using relational confidence between the 7 different regions with detected inconsistencies shown in Table 10.2. For the 3 known error sources (3, 5 and 6) determined

No	Source	Actual
3a (P^{-1})	**Llanishen**: *"it **covers** all of the geographical areas of Llanishen, Birchgrove, and **Thornhill**"*	P^{-1}
3b (EC)	Llanishen is adjacent to Thornhill as found on the Llanishen 3x3 table	P^{-1}
5a (EC)	**ely**, *"is bounded by Fairwater, and Gabalfa to the north-west; Caerau, to the south; **Culverhouse Cross** to the west;"*	P^{-1}
5b (P)	**culverhouse cross**, *"falls within the southwestern tip of the **Ely**, war"*	P^{-1}
6a (P^{-1})	Llanishen *"it **covers** all of the geographical areas of Llanishen, **Birchgrove**, and Thornhill,"*	P^{-1}
6b (EC)	Llanishen is adjacent to Birchgrove, as found on the Llanishen 3x3 neighbourhood table.	P^{-1}

Table 10.4: Actual topological inconsistencies in Wikipedia articles for Cardiff Districts and Wards

manually and agreed with using the compositional confidence technique, the correct relation was ranked 1st 33% of the time, 2nd 33% of the time, and >2nd 33% of the time. Consequently, relational confidence has not been completely successful in suggesting possible relations to replace inconsistent relations, with only 66% ranked within the top 2 possibilities. Relational confidence requires evidence from consistent inferences, hence as the number of explicit relations in the geo-ontology increases, so should the accuracy of the ranking.

For those detected inconsistencies that where actually correct relations (1, 2, 4 and 7), the explicit relation is the relation that should hold. However, relational confidence only ranked the correct relation 1st in 25% of the cases, 2nd in 25% of cases, 3rd in 25% of cases and 4th in the remaining 25% of cases. Again this should improve when more relations are present in the geo-ontology. If we remove the relations from table 10.5 that correspond to known inconsistencies identified in table 10.3, then the correct relation for error 3 is now ranked 2nd, similarly the correct relation for error 6 is now ranked 1st and the relation for error 5 is still ranked 1st but with less ambiguity. This means that by omitting those known inconsistencies, the correct relation is ranked 1st in 42.85% of cases, 2nd in 28.57% of cases and >2nd in the remaining 28.58% of cases.

No.	Relation	Ranking	Pos.
1	TPP-1	0.875	1
1	PO	0.75	2
1	EC	0.75	2
1	TPP	0.625	3
1	EQ	0.5	4
1	DC	0.5	4
1	NTPP-1	0.125	5
2	EC	0.888889	1
2	TPP-1	0.77778	2
2	PO	0.66666667	3
2	NTPP-1	0.666666667	3
2	DC	0.66666666667	3
2	EQ	0.444444	4
2	TPP	0.444444	4
3	EC	0.888889	1
3	TPP-1	0.77778	2
3	PO	0.66666667	3
3	NTPP-1	0.666666667	3
3	DC	0.66666666667	4
3	EQ	0.444444	5
3	TPP	0.444444	5
4	PO	0.875	1
4	DC	0.875	1
4	TPP	0.875	1
4	NTPP	0.875	1
4	EQ	0.625	2
4	EC	0.625	2
4	TPP-1	0.625	2

No.	Relation	Ranking	Pos.
5	NTPP-1	0.8888889	1
5	EC	0.8888889	1
5	TPP-1	0.8888889	1
5	PO	0.77777778	2
5	EQ	0.7777777778	2
5	DC	0.7777778	2
5	TPP	0.77777778	2
6	EC	0.88888889	1
6	NTPP-1	0.77777777778	2
6	TPP-1	0.777777778	2
6	PO	0.6666667	3
6	DC	0.66666667	3
6	EQ	0.44444444	4
6	TPP	0.44444444	4
7	EC	0.88888889	1
7	NTPP-1	0.77777778	2
7	TPP-1	0.77777778	2
7	PO	0.666666667	3
7	DC	0.66666667	3
7	EQ	0.4444444	4
7	TPP	0.4444444	4

Table 10.5: Relational confidence measure for errors in the WikiGeo geo-ontology- correct relations, determined manually, are highlighted for each error

10.2.2 Testing the Combined Official and Wikipedia Geo-ontologies

Test Number: 4

Test Ontology:(5)

Rule Set(s): $F^{RCC12}_{RecordSplit<D,I>}$ with `validTR`

Purpose: Deciding consistency of a real world geo-ontology

The previous application of $SWSRL$ using the $F^{RCC12}_{RecordSplit<D,I>}$ ruleset over the WikiGeo geo-ontology found 18.75% of all errors. In this section we perform the same experiment over the combined OAS+WikiGeo(Primary)(5) geo-ontology, where the existing relations from the WikiGeo geo-ontology are added into (replacing any conflicting) the official administrative subdivisions of Cardiff geo-ontology. Consequently, the number of regions in $SWSRLO$ rises from 60 to 96, and the number of qualitative relations rises from 198 to 3936 - a 42% coverage of topological relations between regions.

Violations: Table 10.6 shows the raw relations in the OAS+WikiGeo(Primary) geo-ontology (found using the previously described rule tracing method) that violated integrity rules in $F^{RCC12}_{RecordSplit<D,I>}$. The fourth column in the table shows if the relation is known, by manual evaluation, to be a correct or incorrect relation. As with the previous experiment, the source of each error is determined automatically by the compositional confidence technique as shown in Table 10.7. For this test, compositional confidence identifies the same 7 edges (region region pairs) as those manually identified. However it also identifies a further 7 edges that correspond to relations that are known to be consistent. Each of these 7 were converse edges to those 7 source inconsistencies, suggesting that converse inferences had propagated errors onto their inverse edge without being blocked by the `validTR` builtin.

As opposed to the previous result which saw only 18.75% of the known errors found, by combining WikiGeo with the OAS geo-ontology, 43.75% of known errors are now found.

Relational Confidence: Tables 10.8 and 10.9 show the ranked relations produced using relational confidence between the 23 different regions with detected (by integrity rules) inconsistencies. With the extra relations gained from the addition of the OAS geo-ontology to the WikiGeo geo-ontology, the overall accuracy of the relational confidence techniques has improved dramatically. This time, for all derived inconsistencies that are known to be correct relations (errors: 2,3,5,7,8,10,11,12,14,15,16,17,19,21,22,23) 100% of them were ranked 1st. This shows that although these relations triggered an integrity rule, the evidence in the ontology stills suggest these to be the correct relation - agreeing with the manual identification.

For each known inconsistent error relation (errors: 1,4,6,9,13,18,20) the correct

Error No	From	Relation	To	Correct
1	ns:Ely	ns:EC	ns:Culverhouse-Cross	×
2	ns:Culverhouse-Cross	P	ns:Ely	✔
3	ns:Ely	ns:EQ	ns:Ely	✔
4	ns:Ely	ns:EC	ns:Gabalfa	×
5	ns:Gabalfa	ns:DC	ns:Ely	✔
6	ns:Ely	ns:DC	ns:Canton	×
7	ns:Canton	ns:EC	ns:Ely	✔
8	ns:Llandaff-North	ns:EQ	ns:Llandaff-North	✔
9	ns:Llandaff-North	ns:EC	ns:Rhiwbina	×
10	ns:Rhiwbina	ns:DC	ns:Llandaff-North	✔
11	ns:Canton	ns:EQ	ns:Canton	✔
12	ns:Llandaff	ns:DC	ns:Butetown	✔
13	ns:Butetown	ns:EC	ns:Llandaff	×
14	ns:Butetown	ns:EQ	ns:Butetown	✔
15	ns:Llandaff	ns:EQ	ns:Llandaff	✔
16	ns:Gabalfa	ns:EQ	ns:Gabalfa	✔
17	ns:Butetown	ns:EC	ns:Splott	✔
18	ns:Splott	ns:PO	ns:Butetown	×
19	ns:Splott	ns:EQ	ns:Splott	✔
20	ns:Grangetown	ns:PO	ns:Butetown	×
21	ns:Butetown	ns:EC	ns:Grangetown	✔
22	ns:Grangetown	ns:EQ	ns:Grangetown	✔
23	ns:Rhiwbina	ns:EQ	ns:Rhiwbina	✔
			Number of Incorrect Relations	7

Table 10.6: Automatically determined topological inconsistencies in the OAS+WikiGeo(Primary) geo-ontology

relation was always ranked 2nd. Choosing the second ranked relation in each of these cases does make the geo-ontology consistent. Hence, as before, by excluding those relations identified by compositional confidence to be inconsistent, 100% of now top ranked relations are the correct relation.

From	To	Count
ns:Grangetown	ns:Butetown	2
ns:Butetown	ns:Grangetown	2
ns:Canton	ns:Ely	2
ns:Ely	ns:Canton	2
ns:Splott	ns:Butetown	2
ns:Butetown	ns:Splott	2
ns:Rhiwbina	ns:Llandaff-North	3
ns:Llandaff-North	ns:Rhiwbina	3
ns:Llandaff	ns:Butetown	2
ns:Butetown	ns:Llandaff	2
ns:Gabalfa	ns:Ely	2
ns:Ely	ns:Gabalfa	2
ns:Culverhouse-Cross	ns:Ely	2
ns:Ely	Culverhouse-Cross	2

Table 10.7: Compositional Confidence counts for errors in the combined OAS+WikiGeo(Primary) geo-ontology

Error No.	Relation	Ranking	Pos.
1	EC	0.9	1
1	NTPP-1	0.8	2
1	TPP-1	0.8	2
1	PO	0.7	3
1	EQ	0.7	3
1	DC	0.7	3
1	TPP	0.7	3
2	EC	0.888888889	1
2	TPP	0.888888889	1
2	NTPP	0.888888889	1
2	PO	0.777777778	2
2	DC	0.777777778	2
2	EQ	0.555555556	3
2	TPP-1	0.555555556	3
3	PO	0.727272727	1
3	EC	0.727272727	1
3	EQ	0.727272727	1
3	DC	0.636363636	2
3	TPP	0.636363636	2
3	TPP-1	0.545454545	3
3	NTPP	0.181818182	4
3	NTPP-1	0.090909091	5
4	EC	0.96	1
4	DC	0.88	2
4	PO	0.84	3
4	TPP	0.52	3
4	NTPP	0.48	5
4	NTPP-1	0.36	6
4	TPP-1	0.36	6
4	EQ	0.04	7
5	DC	0.961538462	1
5	EC	0.884615385	1
5	PO	0.846153846	2
5	NTPP-1	0.461538462	3
5	TPP-1	0.461538462	3
5	TPP	0.384615385	4
5	NTPP	0.384615385	4
6	DC	0.95	1
6	EC	0.85	2
6	PO	0.8	3
6	TPP	0.55	4
6	TPP-1	0.45	5
6	NTPP	0.35	6
6	NTPP-1	0.25	7
6	EQ	0.2	8
7	EC	0.947368421	1
7	DC	0.842105263	2
7	PO	0.789473684	3
7	TPP-1	0.578947368	4
7	TPP	0.421052632	5
7	NTPP-1	0.368421053	6
7	EQ	0.210526316	7
7	NTPP	0.210526316	7

Error No.	Relation	Ranking	Pos.
8	EQ	0.923076923	1
8	PO	0.769230769	2
8	EC	0.769230769	2
8	DC	0.769230769	2
8	TPP-1	0.769230769	2
8	TPP	0.692307692	3
8	NTPP-1	0.076923077	4
9	EC	0.956521739	1
9	DC	0.869565217	2
9	PO	0.826086957	3
9	TPP-1	0.52173913	4
9	TPP	0.47826087	5
9	NTPP-1	0.347826087	6
9	NTPP	0.304347826	7
9	EQ	0.173913043	8
10	DC	0.956521739	1
10	EC	0.869565217	2
10	PO	0.826086957	3
10	TPP	0.52173913	4
10	TPP-1	0.47826087	5
10	NTPP	0.347826087	6
10	NTPP-1	0.304347826	7
10	EQ	0.173913043	8
11	EQ	0.923076923	1
11	PO	0.769230769	2
11	EC	0.769230769	2
11	DC	0.769230769	2
11	TPP-1	0.769230769	2
11	TPP	0.692307692	3
11	NTPP-1	0.076923077	4
12	DC	0.962962963	1
12	EC	0.888888889	2
12	PO	0.851851852	3
12	TPP-1	0.481481481	4
12	NTPP-1	0.444444444	5
12	TPP	0.407407407	6
12	NTPP	0.37037037	7
12	EQ	0.037037037	8
13	EC	0.962962963	1
13	DC	0.888888889	2
13	PO	0.851851852	3
13	TPP	0.481481481	4
13	NTPP	0.444444444	5
13	TPP-1	0.407407407	6
13	NTPP-1	0.37037037	7
13	EQ	0.037037037	8

Table 10.8: Relational confidence measure for errors in the combined OAS+WikiGeo(Primary) geo-ontology - correct relations are highlighted for each error. This table is continued in Table 10.9

Error No.	Relation	Ranking	Pos.
14	PO	0.785714286	1
14	EC	0.785714286	1
14	EQ	0.785714286	1
14	DC	0.785714286	1
14	TPP	0.714285714	2
14	TPP-1	0.642857143	2
14	NTPP	0.142857143	3
14	NTPP-1	0.071428571	4
15	EQ	0.916666667	1
15	PO	0.75	2
15	EC	0.75	2
15	DC	0.75	2
15	TPP	0.75	2
15	TPP-1	0.666666667	3
15	NTPP	0.083333333	4
16	EQ	0.888888889	1
16	PO	0.666666667	2
16	EC	0.666666667	2
16	DC	0.666666667	2
16	TPP	0.666666667	2
16	TPP-1	0.555555556	3
16	NTPP	0.111111111	4
17	EC	0.95	1
17	PO	0.85	2
17	DC	0.8	3
17	TPP-1	0.5	4
17	TPP	0.45	5
17	NTPP-1	0.35	6
17	NTPP	0.3	7
17	EQ	0.15	8
18	PO	0.947368421	1
18	EC	0.842105263	2
18	DC	0.789473684	3
18	TPP-1	0.473684211	4
18	TPP	0.473684211	4
18	NTPP-1	0.315789474	5
18	NTPP	0.315789474	5
18	EQ	0.157894737	6
19	EQ	0.909090909	1
19	PO	0.727272727	2
19	EC	0.727272727	2
19	DC	0.727272727	2
19	TPP-1	0.727272727	2
19	TPP	0.636363636	3
19	NTPP-1	0.090909091	4
20	PO	0.933333333	1
20	EC	0.8	2
20	DC	0.733333333	3
20	TPP	0.6	4
20	TPP-1	0.4	5
20	NTPP	0.333333333	6
20	EQ	0.266666667	7
20	NTPP-1	0.133333333	8
21	EC	0.9375	1
21	PO	0.8125	2
21	DC	0.75	3
21	TPP-1	0.625	4
21	NTPP-1	0.375	5
21	TPP	0.375	5
21	EQ	0.25	6
21	NTPP	0.125	7
22	EQ	0.888888889	1
22	PO	0.666666667	2
22	EC	0.666666667	2
22	DC	.666666667	2
22	TPP-1	0.666666667	2
22	TPP	0.555555556	3
22	NTPP-1	0.111111111	4
23	EQ	0.909090909	1
23	PO	0.727272727	2
23	EC	0.727272727	2
23	DC	0.727272727	2
23	TPP	0.727272727	2
23	TPP-1	0.636363636	3
23	NTPP-1	0.60131313	4
23	NTPP	0.090909091	5

Table 10.9: Relational confidence measure for errors in the combined OAS+WikiGeo(Primary) geo-ontology - correct relations are highlighted for each error

10.3 Hybrid Framework Application Testing

Test Number: 5
Test Ontology:(4)
Rule Set(s): $FBi^{RCC12}_{interleaved}$
Purpose: Deciding consistency of qualitative and quantitative relations in real world geo-ontologies using the hybrid framework

In this section we evaluate how well the hybrid architecture of $SWSRL$ can maintain the topological integrity of the combination of three heterogeneous data sources which mix qualitative and quantitative information, namely; Geonames, Wikipedia and the official administrative boundaries of Wales (which together form the Geonames-WikiGeo-Wales $SWSRLO$ geo-ontology). For this test we use the $FBi^{RCC12}_{interleaved}$ ruleset. This ruleset, as with any interleaved ruleset, can not use either `record` or `valid` builtins, hence error propagation can not be prevented, inference results can not be recorded and thus both relational confidence and compositional confidence methods can not be used. As a consequence, the only way to try to localise and detect the source of errors is to remove all relations from the Geonames-WikiGeo-Wales geo-ontology, and add them back incrementally one relation at a time. When the first error is detected, the newly inserted relation is inspected manually.

The purpose of this test is not to find, as with previous tests, all errors in the geo-ontology, but instead to show the benefits of reasoning over sources with qualitative relations as well as sources with geometric information.

South Glamorgan Violation: An integrity rule was violated during the insertion of the relation P^{-1}(Wales, Glamorgan). Reasoning was stopped and the error was investigated further. The Geonames entry for South Glamorgan, an administrative subdivision of Wales, was found to have a conflict between its geometric and qualitative representation. That is, the geometric representation of South Glamorgan from Geonames is disjoint (DC) from the geometric representation of Wales from the official boundary source. However, the qualitative relations extracted and inserted from Wikepdia suggest that Wales contains The Vale of Glamorgan, and the Vale of Glamorgan is Part of South Glamorgan, hence concluding that

South Glamorgan is either Overlapping, Containing, Equal to, Inside or Covered by Wales, more formally (shown in a logical syntax using RCC-12 relations):

$$P^{-1}(\text{Wales,Vale-of-Glamorgan}) \wedge P(\text{Vale-of-Glamorgan, South-Glamorgan}) \\ \rightarrow O(\text{Wales, South-Glamorgan})$$

Where the RCC-12 relation O is the disjunctive set of RCC-8 relations:

$O \equiv \{PO \vee TPP \vee NTPP \vee EQ \vee NTPP^{-1} \vee TPP^{-1}\}(\text{Wales, South-Glamorgan}\}$

Consequently, South Glamorgan can not be disjoint from Wales, as identified by the following integrity rule (in simplified syntax):

$$P^{-1}(A,B) \wedge P(B,C) \wedge DC(A,C) \rightarrow error(A,C) \quad (10.1)$$

Where, as this example uses the interleaved ruleset, each of the topological predicates in rule (10.1) are determined using the entire set of compositional and converse deduction rules in the backward system. The disjointness relation DC was determined from its computation in the LSS, when run from inside the backward system using the procedural attachment `exDisjoint(A,B)`.

In actuality, the geometric representation of South-Glamorgan from Geonames is in the sea (illustrated in Figure 10.3), hence the conflict.

This example shows the synergy that can be achieved between quantitative and qualitative information. Furthermore, the success of the hybrid architecture shows that spatial relations between regions can be computed on the fly during reasoning, and does not have to be pre-computed and stored in the ontology.

10.4 TOPO-SEMANTIC INTEGRITY RULES IN *SWSRLO*

Test Number: 6
Test Ontology:(2)
Rule Set(s): F_I^{RCC12} plus user defined constraint
Purpose: Detecting topo-semantic errors

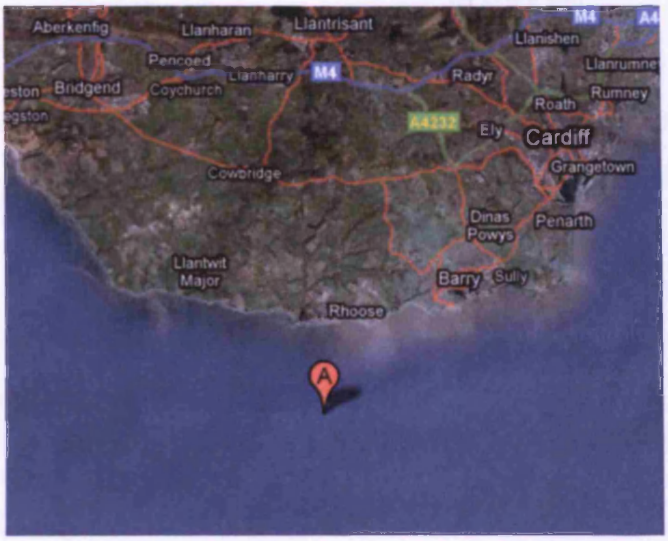


Figure 10.7: Geonames South Glamorgan geometric error

In this section we show, briefly, how feature type semantics of geographic regions, represented by axioms within *SWSRLO*, can be formalised as integrity constraints and used to detect geometric errors present in official data sets.

The topological relations in the OAS geo-ontology are consistent with the integrity ruleset $F^{RCC12}_{<D,I>}$. However, on closer inspection, the geometry of the Pontprennau Community Parish and the Unitary Authority of Cardiff overlap as shown in Figure 10.8. Web searches reveal that the Unitary Authority of Cardiff contains the Pontprennau Community Parish, hence implying the converse relation that Pontprennau Community Parish is either inside or covered by the Unitary Authority of Cardiff (in this case it is coveredBy). Indeed, a parish should always be contained in a Unitary Authority as shown by the administrative hierarchy in Appendix A.2. By adding a new integrity constraint to *SWSRL* that Parishes and Unitary Authorities do not overlap (which can be assumed a general rule), we can detect the geometric inconsistency that is evident between the Pontprennau Community Parish and the Unitary Authority of Cardiff.

Consequently, the following rule was added to *SWSRL*, where Unitary Authorities and Parishes are model as subclasses of Region in the OAS geo-ontology so as to differentiate between them.

```
[ <label>parishOverlapUAError</label><ruleLevel>0</ruleLevel><ruleGroup>
```

`Topo-Semantic</ruleGroup><ruleType>0</ruleType><ruleClass>1</ruleClass>` :
C:Parish(?x) AND C:UnitaryAuthority(?y) AND PO(?x ?y) →
error(parishOverlapUAError ?x Overlaps ?y NA 0 parishes_and_UA_do_not_overlap parishOverlapUAError)]

(10.2)

Using rule 10.2, 14 errors where detected. Indeed the geometric representation of all parishes that should be covered by Cardiff are slightly inaccurate - as an examaple see Figure 10.8.

Figure 10.8: Geometric inconsistency between Cardiff Unitary Authority, shown in blue, and the and Pontprennau Community Parish, shown in orange. The image to the right is a close-up of the image on the left, highlighting the geometric inaccuracy

This evaluation shows the generality of the language and reasoning engine, and motivates future work in defining additional types of integrity constraints in *SWSRL*.

Chapter 11

Empirical Results - Efficiency Testing

In the previous chapter we looked at the application of *SWSRL* over geo-ontologies constructed from real world data sources. In this chapter we consider the efficiency and scalability of *SWSRL*'s reasoning engines from four different perspectives:

(a) An analysis of the relative performance of deduction only topological spatial reasoning in the forward and backward engines, when treated separately.

(b) An analysis of the efficiency of the engine when considering integrity and deduction reasoning in both the forward engine, and the interleaved engine over both synthetic and real world *SWSRLO* geo-ontologies.

(c) The efficiency of the forward engine under incremental topological updates to an *SWSRLO* geo-ontology.

(d) How the distribution of topological relations in *SWSRLO* effects the number of inferred topological relations.

Testing Methodology: Empirical testing is performed over both real world and synthetic geo-ontologies constructed in Chapter 9. Synthetic ontologies are used where the exact type and distribution of topological relations is important, otherwise one of the real world geo-ontologies is employed. Furthermore, the real world geo-ontologies are needed when testing the hybrid mode, as these tests require geometric information associated to each geofeature to be present in the LSS.

Hardware and Software: All tests are run on a 2.00 ghz core2 duo Intel processor, with 2GB of RAM, running on OSX 10.5 (Leopard) from Apple Inc. and Java v1.6 from Sun. For tests that measure reasoning performance, memory usage and execution times were recorded for three separate runs and averaged.

11.1 Deduction and Integrity Rule Execution Times and Memory Usage

In this section we analyse the efficiency in terms of reasoning time and memory usage for the main mode of inference in *SWSRL* (forward reasoning with Rete) using the topological deduction ruleset F_D^{RCC12}, and the topological integrity and deduction ruleset $F_{<D,I>}^{RCC12}$.

Firstly we optimise the representation of the forward ruleset F_D^{RCC12} in Rete. Then, using the optimised form of these rules, we test the scalability of the Rete engine to reason with large *SWSRLO* geo-ontologies. Each result is compared to a Java implementation of the well known and efficient Vilain and Kautzs path consistency [265] and Mackworth's Revise [169] algorithms as a base-line, as detailed in Appendix C.1.

11.1.1 Improving the Efficiency of F_D^{RCC12} Rules

Test Number: 7
Test Ontology: (1)
Rule Set(s): F_D^{RCC12}
Purpose: Improving the efficiency of topological spatial reasoning rules in Rete

Topological reasoning in *SWSRL* should only operate over features of type Region in *SWSRLO*, hence each topological reasoning rule needs to include type checking body predicates. It is well known that rule execution using a Rete based production engine benefits from a *sensible* ordering of rule predicates (fact patterns) in a rule body, for example see [261]. The ordering of predicates in the rule body of forward qualitative spatial reasoning rules in F_D^{RCC12} can greatly effect the efficiency of rule execution. In particular, the memory footprint of the network can be reduced and the speed of inference can be increased by considering a post-order (rule 11.2) checking of individual types (checking if the individuals are members of the class Region) as opposed to pre-order (rule 11.1) type checking.

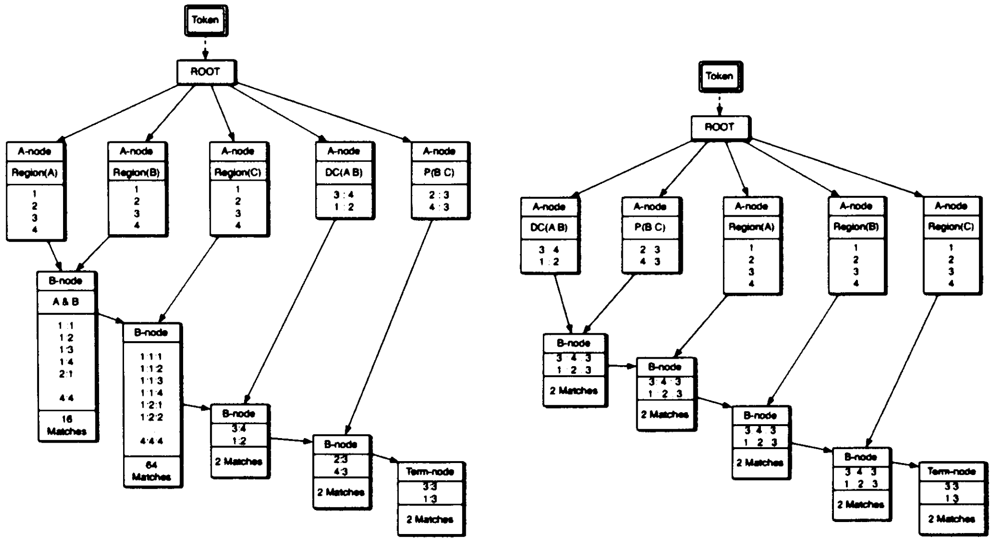
$$Region(?a) \wedge Region(?b) \wedge Region(?c) \wedge R_1(?a,?b) \wedge R_2(?b,?c) \rightarrow R_3(?a?c) \quad (11.1)$$

$$R_1(?a,?b) \wedge R_2(?b,?c) \wedge Region(?a) \wedge Region(?b) \wedge Region(?c) \rightarrow R_3(?a?c) \quad (11.2)$$

For efficient rule execution, the most general predicates must be ordered toward the end of the rule body and the most specific toward the start of the rule body. Consider the following example, where R_1 is substituted with the generalised topological relation DC, R_2 is substituted with the relation P and R_3 is substituted with the relation coP. We also assume the following ontological facts in *SWSRLO* : $\rightarrow$ Region(1), $\rightarrow$ Region(2), $\rightarrow$ Region(3), $\rightarrow$ Region(4), $\rightarrow$ DC(1,2), $\rightarrow$ DC(3,4), $\rightarrow$ P(2,3) and finally $\rightarrow$ P(4,3).

The Rete is constructed separately for each rule as illustrated in Figures 11.1(a) and 11.1(b). Rete uses a-nodes which are one input pattern matching nodes, and b-nodes (beta) which are two input nodes that join common variables from two a-nodes. The terminal node then projects the variables from the rule body to those found in the rule's head. In the pre-order case, a total of 16+64+2+2+2 = 86 elements need to be stored in the Rete. In the post-order case this is dramatically reduced to a total of 2+2+2+2+2 = 10 elements (11% of the storage cost). As the number of region individuals in the ontology increases, so do the benefits from using the post-ordering. Indeed, the pre-ordering has a $O(n^3)$ data complexity for each rule r, on the number of regions stored in the beta (join) nodes, whereas the post-ordering is affected only by the number of topological relations between regions, a worse case of $O(n)$ data complexity for any one rule. A naive grounding of each rule requires n groundings, where n is the number of relations in *SWSRLO*. This results in a $O(n^2)$ data complexity in the number of regions n, much like the data complexity of the original path-consistency algorithm. However, Rete exploits structural similarity between rules (matched facts in alpha nodes are shared between rules) hence the overall complexity depends on the particular construction of Jena2's Rete discrimination network.

The result of different orderings is shown empirically in Figures 11.2 and 11.3,



(a) The Rete using a pre-ordering of type checking predicates

(b) The Rete using a post-ordering of type checking predicates

Figure 11.1: Different Rete discrimination networks

where Figure 11.2 highlights the in-memory usage of the three approaches, and Figure 11.3 highlights the execution time of the three approaches. The experiment was conducted on 2 through to 25 region scenes generated using the genetic algorithm, where all regions have a definite topological relation to itself and every other region (a full spatial configuration). All 89 deduction rules where run to fixed-point (all entailments where generated). The post-ordering of type checking predicates (the class predicate `Region`) was significantly faster than the pre-ordering of type checking predicates, and came closer to the baseline path-consistency algorithmic approach (which does not type check, instead assuming the network is between regions only). Furthermore, the memory footprint of the Rete was, as predicted, significantly lower with the post-ordering approach see Figure 11.2. As a result, all predefined rules are encoded using a post-ordering of type checking predicates.

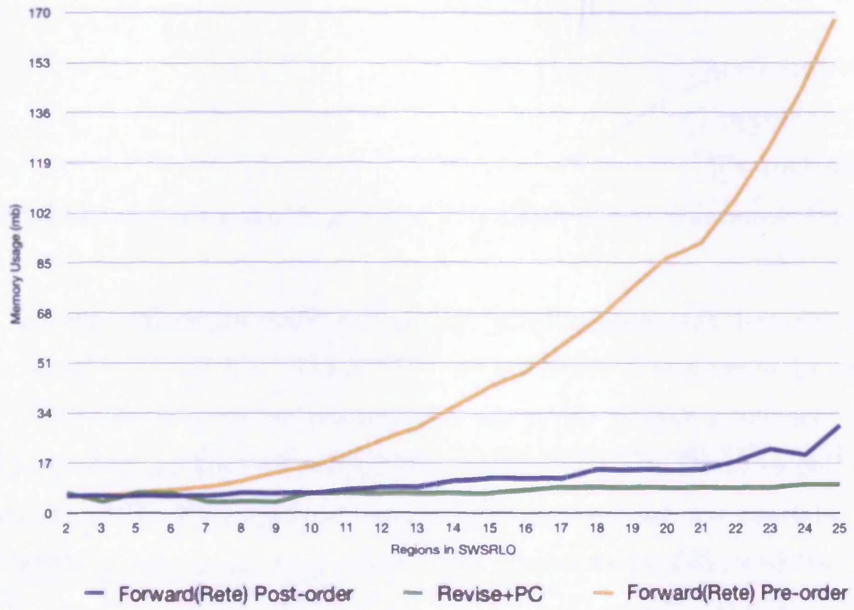


Figure 11.2: Pre-order v Post-order v Baseline (Revise+PC) region checking memory usage

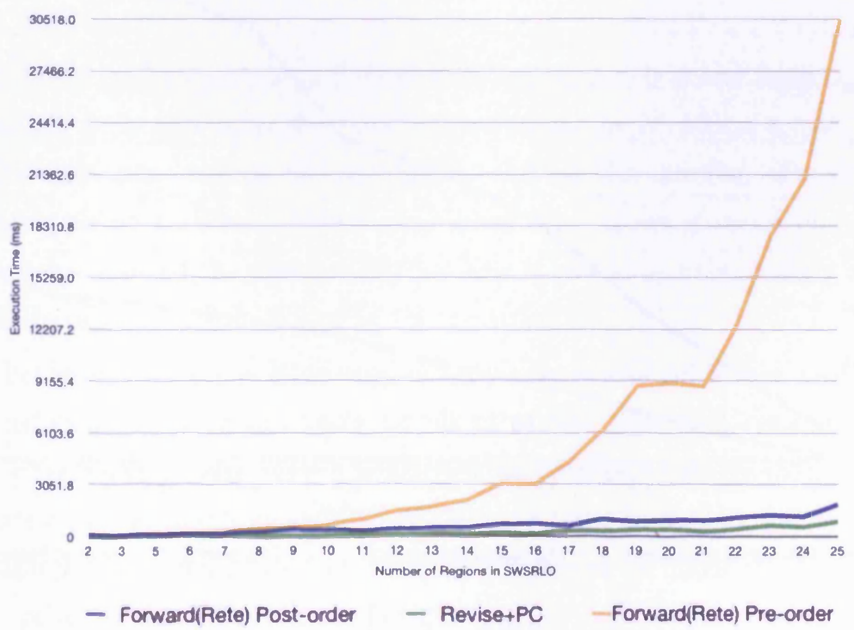


Figure 11.3: Pre-order v Post-order v Baseline (Revise+PC) execution time

11.1.2 Deduction using F_D^{RCC12}

Test Number: 8

Test Ontology:(1)

Rule Set(s): F_D^{RCC12}

Purpose: Scalability of the deductive topological reasoning in the forward engine

Here we test the scalability of using the Rete algorithm against increasing numbers of regions and relations in $SWSRLO$. To test the scalability of the forward reasoning engine using the forward deduction ruleset F_D^{RCC12}, synthetic geo-ontologies of 50 and 100 regions are used with varying percentages of topological relations i.e. for the 100 region geo-ontology, 50% of the relations relates to $(50^2) * 0.5 = 1250$ relations etc. The in-memory usage and execution times for reasoning using the forward engine and the base-line Revise path-consistency (PC) algorithm are recorded, plotted and shown in Graphs 11.4 and 11.5 respectively.

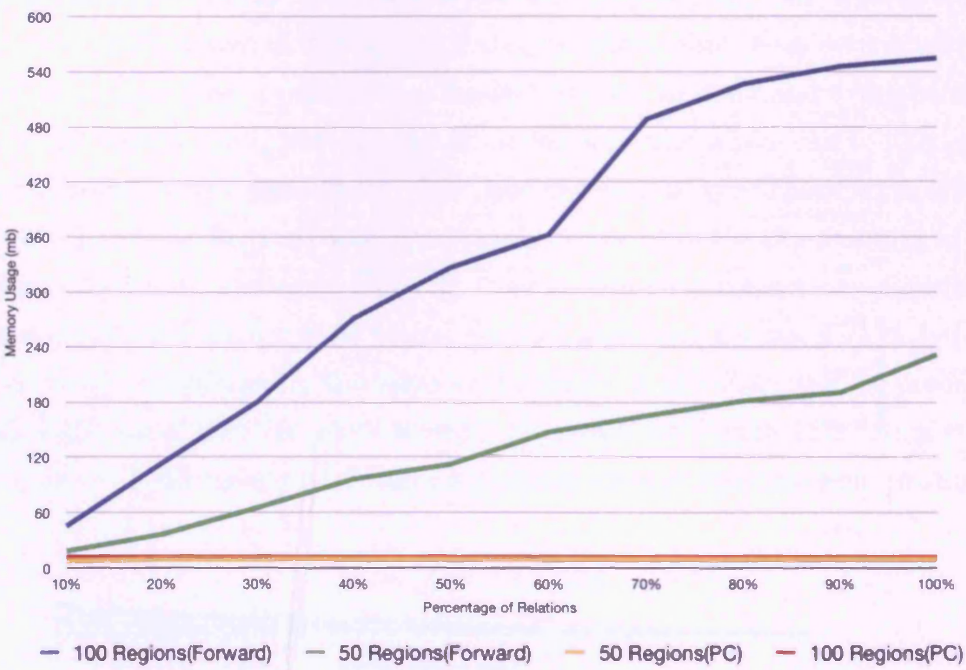


Figure 11.4: F_D^{RCC12} memory usage for different numbers of regions (50 and 100) and relations (from 10% to 100%)

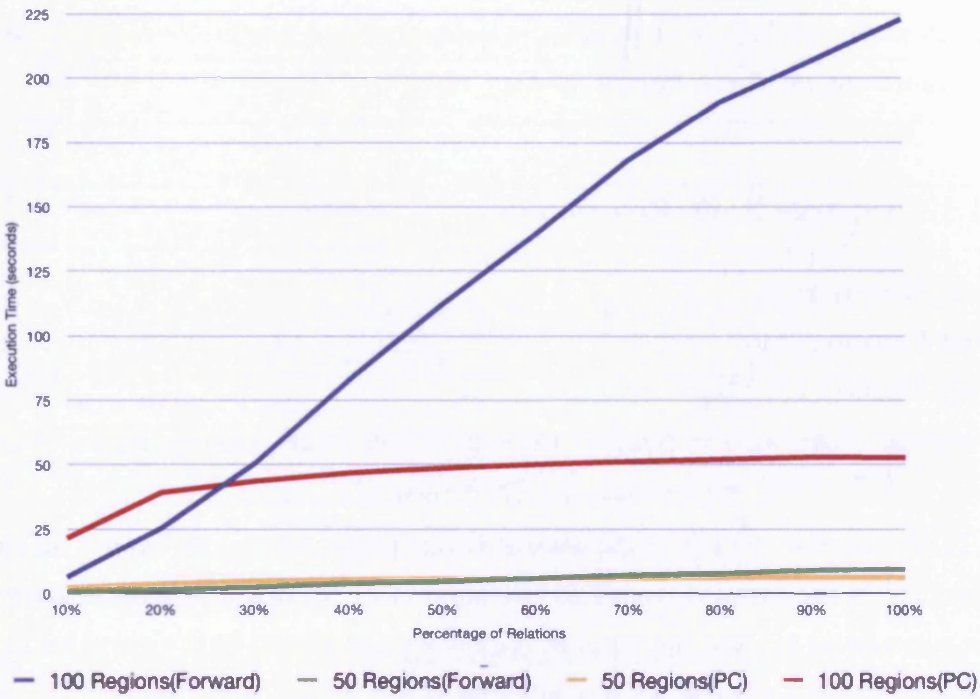


Figure 11.5: F_D^{RCC12} execution times for different numbers of regions (50 and 100) and relations (from 10% to 100%)

As is shown in Figure 11.5, the baseline PC algorithm is slower for small numbers of relations, as it still tries all possible combinations of region triples A, B and C for the first run - which is quadratic based on the number of regions in the ontology n ($O(n^2)$). The baseline algorithm then shows a small increase in its runtime behaviour as the number of relations it checks is only slightly more than the number of relations it checks on the first run.

On the other hand, the Rete engine only runs those rules that have matched body predicates. Hence for lower numbers of relations, this results in a lower number of rule executions. The execution time increases linearly with the number of relations in the ontology, which is an expected result that follows from the computational complexity of Rete ($O(RFP)$) which is linear as the number of facts F (relations here) increases. For 100 regions, and 100% of relations (10,000 relations), the Rete engine takes 223 seconds, whereas the PC algorithm takes 53 seconds. It is foreseeable that as the number of regions increases, the difference between the Rete engine and the PC algorithm will also increase substantially.

That said, taking only 223 seconds to perform full first time reasoning over a 100 region place ontology is acceptable in comparison to the limits reached using the backward engine shown later in section 11.2.

11.1.2.1 Large Scale Reasoning

Test Number: 9
Test Ontology:(6)
Rule Set(s): $F^{RCC12}_{<D,I>}$
Purpose: Scalability of the forward engine over larger geo-ontologies

In this section we try to give a real world pragmatic upper bound on the performance of the forward engine as the number of regions in a geo-ontology varies (as opposed to varying the number of relations as shown in the previous section). Here the largest *SWSRLO* geo-ontology (OAS-Large(6)) is used, where regions and their associated relations are added in 9 steps of 25, starting at 37 regions and finishing with 237 regions. Figures 11.6 and 11.7 show the performance of the forward Rete engine using the forward integrity and deduction ruleset $F^{RCC12}_{<D,I>}$, compared to the base-line PC algorithm.

It is clear in Figure 11.6 that the execution time of the Rete approach is polynomial in the number of regions in the ontology. This is again an expected trend as Rete is known to be polynomial on the number of objects (regions here) in working memory (|WM|) [76]. Indeed the PC approach is also polynomial ($O(n^3)$), but its more concise practical implementation (working off a primitive array of relations, as opposed to the overheads of constructing and reasoning with large Rete discrimination networks) makes it more scalable during real world testing.

In terms of real world scalability, reasoning with a 237 region scene that contains 56,169 relations using the $F^{RCC12}_{<D,I>}$ ruleset is time consuming, completing in 56 minutes. This proves much less tractable than the Vilain and Kautzs procedural PC approach which completes in 5 minutes. Furthermore, a 2gb memory overhead is hit when reasoning with ontologies with more than 200 regions and over 40,000 relations - see Figure 11.7. The in-memory usage shows an almost linear increase with the number of regions, which is similar (but over a larger amount of regions) with the results in section 11.1.1 after the post-order optimisation of predicates

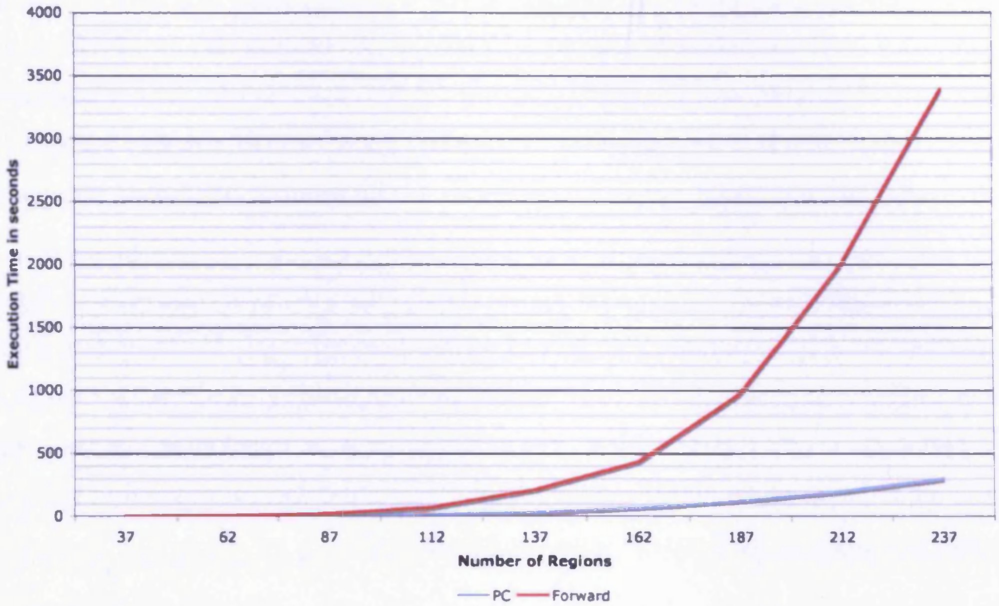


Figure 11.6: Execution times for the declarative forward ruleset $F^{RCC12}_{<D,I>}$ and procedural PC approaches, where the number of regions increases from 37 to 237 in 25 region increments

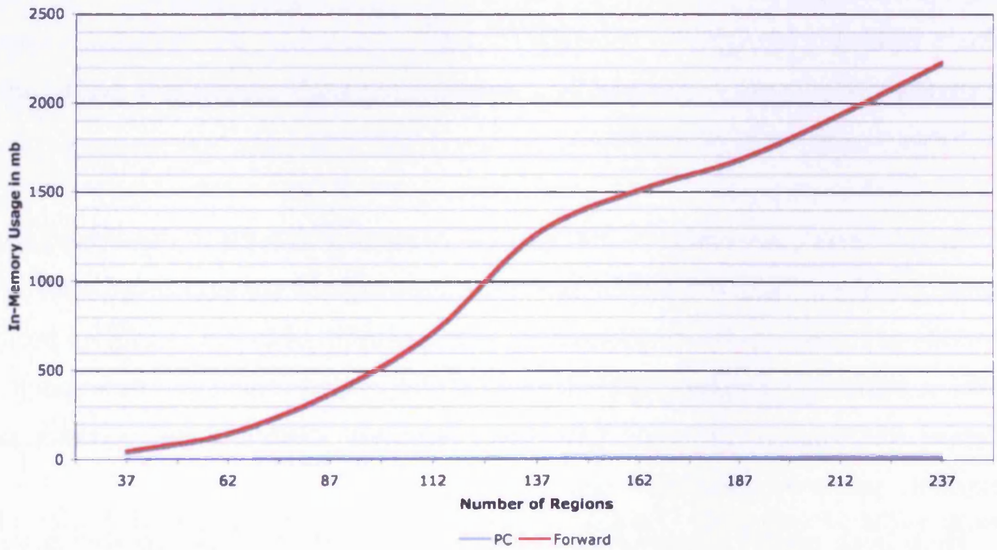


Figure 11.7: In-memory overheads for the declarative forward ruleset $F^{RCC12}_{<D,I>}$ and procedural PC approaches, where the number of regions increases from 37 to 237 in 25 region increments

in the ruleset. The in-memory usage of the PC algorithm is known to be strictly quadratic in the number of regions n ($O(n^2)$). As seen with execution speed, the Java PC algorithm has no additional implementation overheads compared to the construction of a full Rete network, hence the lower overall memory usage - a more obvious quadratic trend would be expected as the number of regions increases past 237.

Web ontologies of more realistic size may contain millions of regions. For example there are 6 million places in the Geonames geo-ontology. Under these conditions it is unlikely that this declarative approach to reasoning would scale well, if at all, to topological qualitative spatial reasoning over such ontologies. In the next section we show in more detail reasons why topological spatial reasoning rules do not scale well using a declarative reasoning approach.

11.2 Direct Comparison of Backward and Forward Engines for Topological Reasoning

Test Number: 10
Test Ontology:(2)
Rule Set(s): $GeoR_{bk}^{standard}$, $GeoR_{bk}^{hybrid}$,F_{D}^{RCC12}
Purpose: Performance comparison of reasoning with topological deduction rules in forward and backward engines

In this section we compare and analyse reasoning in both the forward and backward system in more depth, before later showing the scalability of both systems combined in the interleaved reasoning mode of *SWSRL*. All testing is performed over a partially complete OAS(2) geo-ontology, where 50% of the explicit topological relations in the scene have been removed. Hence some relations are not explicit, and need inferring.

Here both modes of reasoning (forward and backward) are treated separately and their performance in terms of real world reasoning time and memory usage is analysed. In addition, we also count the number of logical inferences (or number of rule evaluations) needed to either derive all topological entailments (closing the scene under composition and converse) in the forward system, or performing a single query to the backward system. One query to the backward system can require the evaluation of all topological inferences to determine the answer to the query, hence a single query to the backward system can be comparable to finding and storing all entailments using the forward system.

Counting the number of logical inferences per query: Figure 11.8 shows the number of rule evaluations used to answer queries in the backward system. In total we performed 8 queries (for each RCC-8 base relation) between the regions `(Gabalfa Cathays)`, `(Llandaff Heath)` and `(Butetown Grangetown)` - a total of 24 queries.

Although at least one of these queries *should* succeed for each of the regions, for the purpose of this test we have chosen queries that can not be answered using only qualitative reasoning (remembering that the geo-ontology only has 50% of all possible qualitative relations). In this way we can show the worst case of querying in the backward system, where all possible inference paths (branches of the SLD or SLG resolution tree in XSB) are searched and fail. For each query to the backward system, a base-line (shown with a black line denoted FD-Complete) number of rule evaluations taken to find all topological entailments using the forward system is shown. Furthermore, overlaid on the second y-axis is the amount of memory used during reasoning.

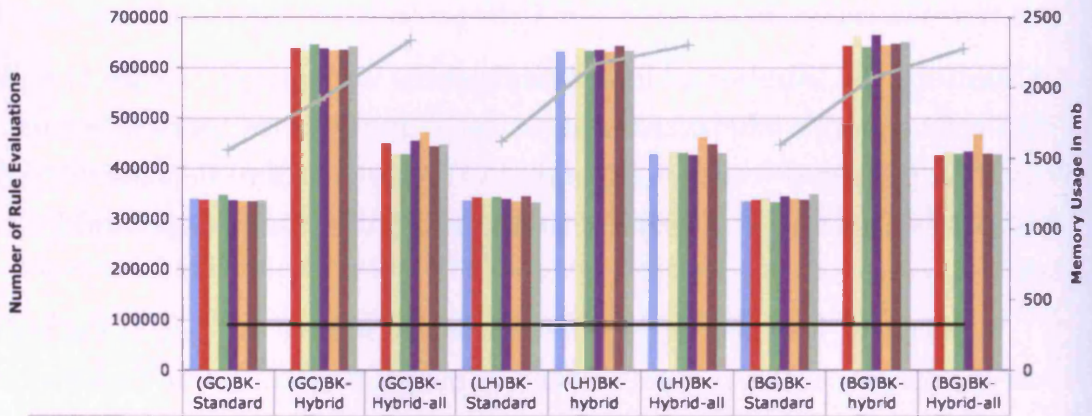
Queries are answered in the backward system using three different reasoning modes, namely; BK-Standard, BK-Hybrid and BK-Hybrid-all, these are:

(a) BK-Standard : using the ruleset $GeoR_{bk}^{standard}$, which only includes qualitative topological inference rules. This ruleset is almost* a direct representation of the F_D^{RCC12} ruleset used in the forward system (FD-Complete) as a base-line measure. Here, all RCC-12 predicates (topological relations) are tabled in XSB to avoid entering an infinite loop.

(b) BK-Hybrid : which uses the $GeoR_{bk}^{hybrid}$ ruleset, that also includes external calls to the LSS to compute topological relations. Importantly here, all RCC-12 predicates are tabled as before, however the LSS predicates are not tabled as these predicates are not involved in any recursive rules.

*It also contains mapping to and from RCC-12 predicates

(c) BK-Hybrid-all: using the same ruleset as BK-Hybrid ($GeoR_{bk}^{hybrid}$) but tabling all predicates in the rule engine, which then tables results computed from the LSS.



	(GC)BK-Standard	(GC)BK-Hybrid	(GC)BK-Hybrid-all	(LH)BK-Standard	(LH)BK-hybrid	(LH)BK-Hybrid-all	(BG)BK-Standard	(BG)BK-hybrid	(BG)BK-Hybrid-all
EC	339657	0	2	336176	633479	429471	334416	8	2
DC	337303	638154	449591	342618	0	2	337522	642734	425362
PO	338154	634140	428273	341197	638940	432707	340441	662704	433267
EQ	346619	646052	428137	343919	633331	430647	332480	640728	428567
NTPP	336439	637682	454990	339488	635763	426780	344429	664923	433176
TPP	334080	633778	471768	334700	631918	462076	339547	644526	467362
TPP-1	334428	635538	444381	345075	643922	448057	337839	646870	429024
NTPP-1	336221	641781	447693	332442	633450	430348	348560	650483	428271
Mem Usage	1558.875	1908.125	2336.75	1621.25	2168.25	2300.125	1593.75	2068.25	2275.125
FD Complete	90162	90162	90162	90162	90162	90162	90162	90162	90162

Figure 11.8: Number of rule evaluations overlaid with memory usage for three queries to the backward system using BK-Standard, BK-Hybrid and BK-Hybrid-all modes

As is immediately clear, for unsatisfiable or failed queries e.g. `ns:EC(ns:Gabalfa Cathays)`, the number of rule evaluations for all reasoning modes (even using only the BK-Standard mode which uses an almost identical ruleset to the forward mode) involves a much greater number of rule evaluations than the forward system does to find all possible entailments for the given scene. For example, the failed query `ns:dc(ns:Gabalfa Cathays)` required only 90,162 rules to fire to determine whether the fact `ns:EC(ns:Gabalfa Cathays)` is entailed using the forward system. The backward system requires 337,303 rule evaluations (73% more) in the BK-Standard mode, 638,154 evaluations (86% more) in the BK-Hybrid mode and 449,591 evaluations (80% more) in the BK-Hybrid-all mode.

This is an expected behaviour of the backward system which works in a top down manner. For an usatisfiable query all relevant branches of the constructed

SLD or SLG trees in the XSB backward engine have to be traversed, including branches that result in a fail. As is clear from the large number of rule evaluations, compositional inferences are highly interconnected inducing a large search space. On the other hand, the forward system (thanks to Rete) works in a bottom up manor and only fires rules that can be satisfied from explicit or newly inferred facts. As a result, branches which need to be explored in the backward system that result in failure, would not trigger the execution of a rule in the forward system.

It is also clear that the BK-Hybrid mode required more rule evaluations, across all queries, than the BK-Hybrid-all mode. This is because in the BK-Hybrid-all mode calls to the LSS to compute the topological relations between two regions are tabled when first computed. Subsequent consumers of those facts in other branches of the SLG tree are then evaluated against the tabled result. In the BK-Hybrid mode duplicate queries to the LSS are re-computed. However tabling all facts including the results of LSS computation increases the size of the table and hence increases in-memory usage, as is again clear from the in-memory usage trend shown in Figure 11.8.

Satisfiable queries to either hybrid modes, for example the query `ns:EC(ns:Gabalfa ns:Cathays)`, are found directly from the LSS within a few rule evaluations. Such behaviour is not theoretically guaranteed as the rule scheduler could have evaluated more qualitative rules before executing the procedural attachment that calls the LSS. However within Jena's XSB implementation, rules with procedural attachments are often executed and evaluated before the rules without procedural attachments. Hence the computation and satisfaction of these relations in the LSS before other logical rules have been evaluated.

A closer inspection of tabling in BK-Hybrid and BK-Hybrid-all modes: Figure 11.9 shows how the number of rule evaluations for each rule in the ruleset $GeoR_{bk}^{standard}$ or $GeoR_{bk}^{hybrid}$ varies for the query `ns:DC(ns:Gabalfa ns:Cathays)`. This highlights a few important points. Firstly, it is clear that the BK-Standard mode, using the $GeoR_{bk}^{standard}$ ruleset, does not employ rules that call the LSS. Secondly, BK-Hybrid-all uses the minimum number of calls to the LSS for each pair of evaluated regions - repeated calls are taken directly from the table. On the other hand, the BK-Hybrid mode does not store these results which contributes to the greater number of rule evaluations for rules with LSS calls.

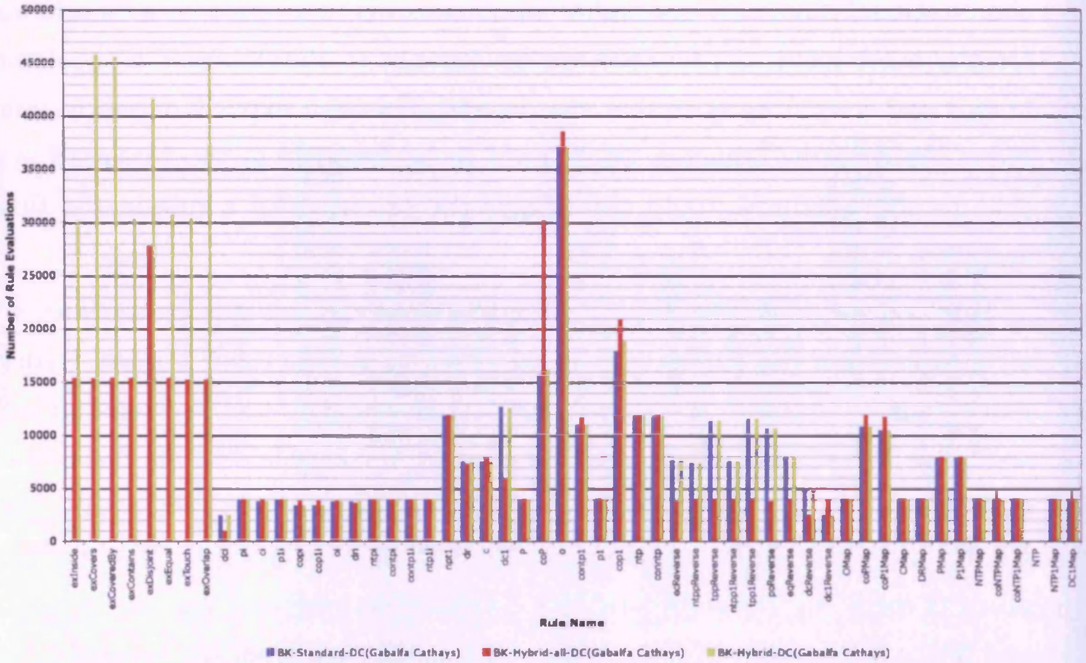


Figure 11.9: Number of rule evaluations per rule name for the query ns:DC(ns:Gabalfa ns:Cathays) to the backward system using BK-Standard, BK-Hybrid and BK-Hybrid-all modes

Evaluating satisfiable queries: Figure 11.10 shows the number of rule evaluations for the query `ns:DC(Penylan Lisvane)`, which is known to be entailed from, but is not explicit in, the geo-ontology. Again all three modes of the backward system are used. In addition, on the secondary y-axis the base-line number of rule evaluations using the forward system is shown, along with the accumulative total rule evaluations for the BK-Standard mode. Both BK-Hybrid and BK-Hybrid-all modes require only 1-2 rules before the relation is computed and found from the LSS, hence the omission of accumulative totals for these modes.

Of importance here, the query which is satisfiable still requires 73% more rule evaluations in the BK-Standard mode than the forward system does (with almost identical ruleset) to find all topological entailments from the geo-ontology.

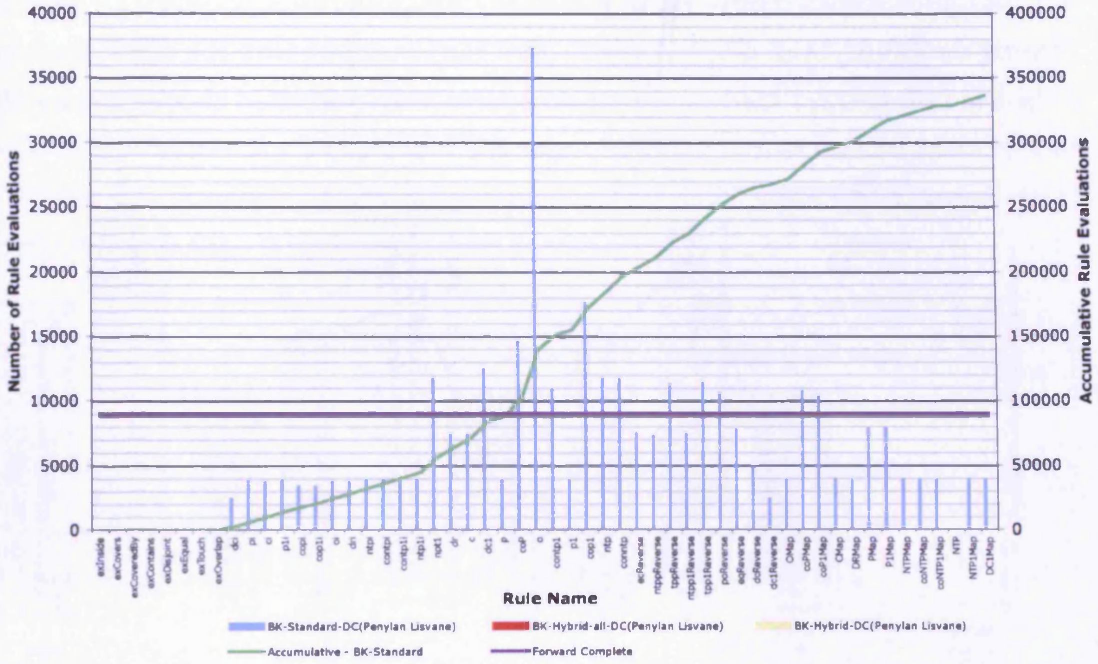


Figure 11.10: Number of rule evaluations per rule name for the query ns:DC(ns:Penylan ns:Lisvane) to the backward system using BK-Standard, BK-Hybrid and BK-Hybrid-all modes

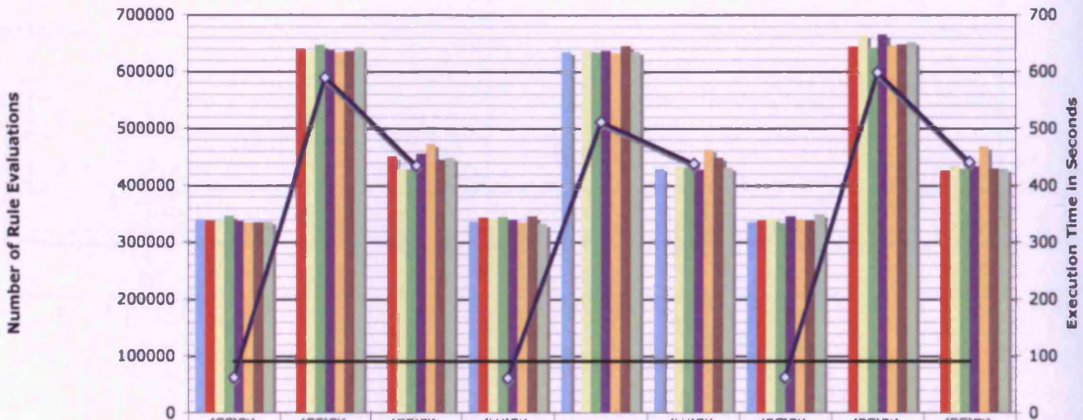
Reasoning time: Figure 11.11 is similar to Figure 11.8, only this time overlaid on the secondary y-axis is the average execution time for performing the 8 queries to the backward system, for each of the three different region pairs.

Importantly, BK-Hybrid and BK-Hybrid-all results are normalised so as to remove the overheads involved in accessing the remote Oracle server using the Java DB interface (JDBC). More specifically, they are normalised as follows:

$$\text{NormalisedTime} = \text{EXETime} - (\text{LSSCalls} * \text{aveTO}) \tag{11.3}$$

Where aveTO is the average time taken to send and receive, but not perform, queries to the Oracle System. This was measured by taking the average time for Oracle to compute topological relations between regions in the OAS geo-ontology, away from the total time for the accessing, querying and receiving a result from Oracle. EXETime is the total recorded execution time for the query and LSSCalls is the number of external calls to Oracle recorded during evaluation of the query.

In an expected reverse from in-memory usage shown in Figure 11.8, reasoning with the BK-Hybrid-all mode was faster than reasoning with the BK-Hybrid mode as the LSS was not used to re-compute previously computed topological relations.



	(GC)BK-Standard	(GC)BK-Hybrid	(GC)BK-Hybrid-all	(LH)BK-Standard	(LH)BK-hybrid	(LH)BK-Hybrid-all	(BG)BK-Standard	(BG)BK-hybrid	(BG)BK-Hybrid-all
EC	339657	0	2	336176	633479	429471	334416	8	2
DC	337303	638154	449591	342618	0	2	337522	642734	425362
PO	338154	634140	428273	341197	638940	432707	340441	662704	433267
EQ	346619	646052	428137	343919	633331	430647	332480	640728	428567
NTPP	336439	637682	454990	339488	635763	426780	344429	664923	433176
TPP	334080	633778	471768	334700	631918	462076	339547	644526	467362
TPP-1	334428	635538	444381	345075	643922	448057	337839	646870	429024
NTPP-1	336221	641781	447693	332442	633450	430348	348560	650483	428271
FD Complete	90162	90162	90162	90162	90162	90162	90162	90162	90162
Execution Time(s)	61.49875	588.8569269	433.955838	60.350625	510.1637649	436.9308505	61.793875	597.6098165	440.1760776

Figure 11.11: Number of rule evaluations overlaid with execution time for three queries to the backward system using BK-Standard, BK-Hybrid and BK-Hybrid-all modes

Mode	Average Execution Time (seconds)	Satisfiable
BK-Standard	61.21441667	no
BK-Hybrid	646.3249308	no
BK-Hybrid-all	523.6658467	no

Table 11.1: Average execution times for unsatisfiable queries to BK-Standard, BK-Hybrid and BK-Hybrid-all modes

Table 11.1 shows the average execution times, from the result in Figure 11.11, for failed or unsatisfiable queries*. As expected BK-Standard proves on average to be the fastest. Using this as a base-line, the BK-Hybrid mode takes 90%

*Hence here we exclude the times of those queries that succeeded in the BK-Hybrid and BK-Hybrid-all modes. No query to the BK-Standard succeed, so these times are identical to those shown in Figure 11.11

longer and the BK-Hybrid-all mode takes 88% longer to find queries that do not succeed. Both hybrid modes show a significant increase in execution time over the qualitative only BK-Standard mode. This can be attributed to the fact that both hybrid modes try to evaluate the same qualitative topological inferences as with the BK-Standard mode, but in addition to large numbers of quantitative computation using the LSS.

Query	Mode	Execution Time (seconds)	Explicit
ns:DC(ns:Penylan ns:Lisvane)	BK-Standard	58.702	no
ns:DC(ns:Penylan ns:Lisvane)	BK-Hybrid	0.119	no
ns:DC(ns:Penylan ns:Lisvane)	BK-Hybrid-all	0.05	no
ns:DC(ns:Penylan ns:Lisvane)	BK-Standard	0.033	yes
ns:DC(ns:Lisvane ns:Penylan)	BK-Hybrid	0.06	yes
ns:DC(ns:Lisvane ns:Penylan)	BK-Hybrid-all	0.048	yes

Table 11.2: Execution times for example satisfiable queries to BK-Standard, BK-Hybrid and BK-Hybrid-all modes

For examples of satisfiable queries shown in Table 11.2 (taken from the result in Figure 11.10), BK-hybrid and BK-Hybrid-all show significant improvements in execution time, taking between 30-120 ms to answer an entailed query as these are typically computed within the first few rule evaluations using the LSS. All modes take between 6 to 48 ms for a query that is directly answerable from its corresponding raw fact in the ontology. The BK-Standard still requires 58.7 seconds to successfully answer the entailed query `ns:DC(ns:Penylan ns:Lisvane)`. However this is likely to improve if a smaller number of rule evaluations (shorter inference chains) are required to answer the query.

Summary: Jena2's XSB implementation uses a Single Stack Scheduling Strategy which is known to be poor in memory usage [251, 81]. This is in part responsible for the much larger memory overheads when reasoning with the backward engine than reasoning in the forward engine. The use of more efficient XSB engines is left to future work. However even if the backward engine could be improved, the number of rule evaluations in the backward system and hence the number of tabled results or subgoals, would still be greater than the total number of rule evaluations in the forward system. This suggests that queries to the backward system would never be as efficient as finding *all* entailments using the forward engine. This leaves us to believe that topological (or indeed all spatial) compositional inferences are more efficiently served, in terms of executing time and in-memory usage, using a bottom up data driven forward engine such as Rete.

With respect to the different backward reasoning modes, as the premise of interleaving forward and backward reasoning was to reduce stored fact and memory overheads, where the use of the LSS is needed, the BK-Hybrid mode is chosen over the BK-Hybrid-all mode. In effect sacrificing reasoning time performance for a slight reduction of in-memory usage.

11.3 Scalability of Interleaved Reasoning

Test Number: 10
Test Ontology:(2)
Rule Set(s): $F^{RCC12}_{<D,I>}$,$FBi^{RCC12}_{interleaved}$
Purpose: Testing the scalability of interleaved reasoning in the combined XSB and Rete engines

In the previous section we compared both forward and backward reasoning modes for query answering on individual queries. Here we test the scalability of $SWSRL$ in interleaved mode where the forward integrity ruleset is interleaved with the backward deduction ruleset using the combined ruleset $FBi^{RCC12}_{interleaved}$. As with the previous comparison, we again use the OAS(2) $SWSRLO$ geo-ontology.

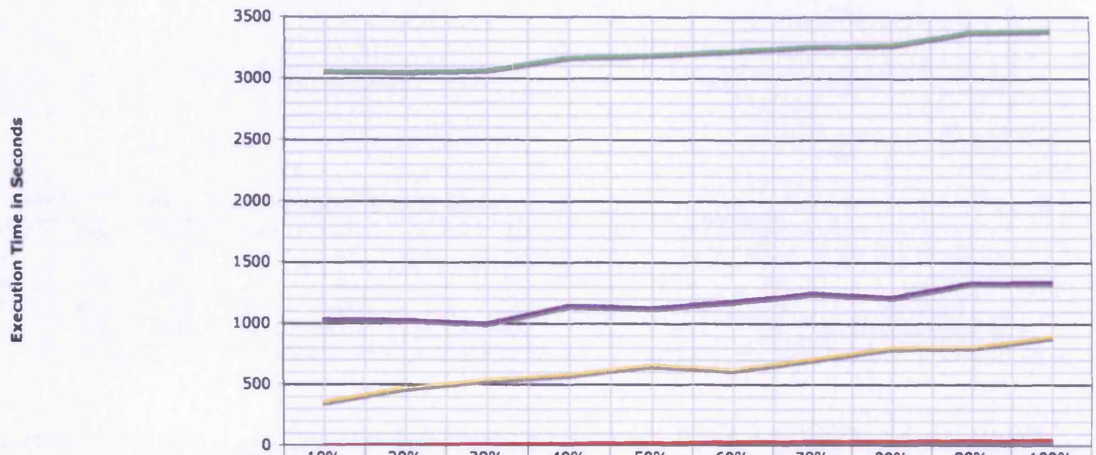
Figures 11.12 and 11.13 show how the time and memory used to reason with the OAS ontology changes as the number of relations in the ontology varies from 10% (577 relations) to 100% (5776 relations) but the number of regions remains fixed. The tests were run using the following modes and rulesets:

- Forward Only : Rete on its own with the forward firing integrity and deduction ruleset$F^{RCC12}_{<D,I>}$.

- Interleaved : Rete interleaved with XSB using the forward integrity rules and backward deduction ruleset $FBi^{RCC12}_{interleaved}$, where the backward ruleset $GeoR^{standard}_{bk}$ used does not include calls to the LSS.

- Interleaved-hybrid : Rete interleaved with XSB using the forward integrity rules and backward deduction ruleset $FBi^{RCC12}_{interleaved}$, where the backward ruleset $GeoR^{hybrid}_{bk}$ used includes calls to the LSS.

- PC : the base-line path-consistency algorithm.



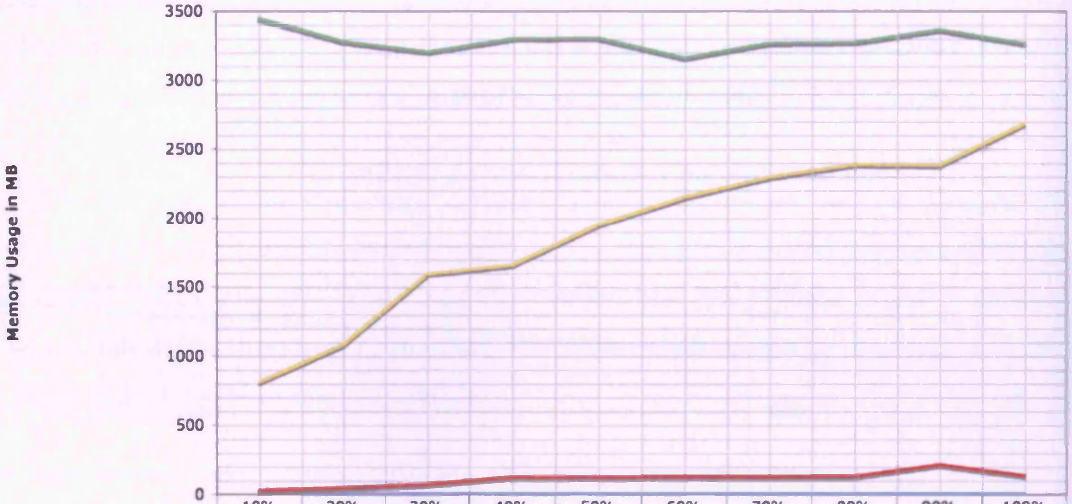
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
PC	3.448	5.52	7.016	8.169	8.843	9.177	9.539	9.75	9.698	9.635
Forward Only	1.341	4.014	8.802	14.211	20.69	25.886	31.715	35.847	40.039	42.581
Interleaved	353.97	470.212	534.963	576.715	656.774	618.983	706.711	801.229	807.545	890.342
Interleaved-hybrid	3058.811	3050.817	3064.167	3167.462	3188.772	3224.727	3261.998	3272.036	3375.91	3387.786
Interleaved-hybrid Normalised	1032.884	1026.773	995.5087	1144.045	1125.542	1179.167	1244.775	1212.884	1335.078	1337.156

Percentage of Relations

Figure 11.12: $F^{RCC12}_{<D,I>}$ v both standard and hybrid $FBi^{RCC12}_{interleaved}$ rulesets v baseline (Revise+PC) execution time for different numbers of relations

In-line with the results shown per query in section 11.2, the interleaved-hybrid mode is the most expensive in terms of execution time and in-memory usage. Indeed the interleaved-hybrid mode is, on average, 45% slower than the interleaved mode, 98% slower than the Forward Only mode and 99% slower than the PC baseline.

Interleaved was the second worst performing in both in-memory and execution time testing, again an expected result. Interleaved is faster than interleaved-hybrid as no calls are being made to the LSS, however both interleaved and interleaved-hybrid are significantly slower than the Forward mode. This is expected as each relation in the body of each forward integrity rule is resolved against the set of backward deduction rules each time an integrity rule is tested.



	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
PC	10	10	11	11	11	11	11	11	11	11
Forward	28	47	73	125	126	129	128	130	216	136
Interleaved	813	1083	1594	1660	1955	2154	2298	2391	2386	2687
Interleaved-Hyrbid	3447	3280	3206	3304	3306	3161	3269	3275	3367	3266

Percentage of Relations

Figure 11.13: $F^{RCC12}_{<D,I>}$ v both standard and hybrid $FBi^{RCC12}_{interleaved}$ rulesets v baseline (Revise+PC) in-memory usage for different numbers of relations

Interestingly the total time of reasoning in the interleaved or interleaved-hybrid mode, which requires 46,624,765 queries to the backward engine, is only 10 times and 2 times slower than performing one query to either BK-Standard or BK-Hybrid respectively (which use the same rulesets). This can again be attributed to tabling, where the first few queries help to build the table of evaluated goals (grounded predicates), subsequent queries to the backward system then find answers to queries directly from the table. This is illustrated in Figure 11.14 (that uses a logarithmic scale for clarity), where the first query evaluates and hence stores results (answers to goals and subgoals) of substantially more rules than subsequent queries - subsequent queries are taken directly from the table. If results were not tabled, each query to the backward system would take the full time for evaluation (around 646 seconds for the interleaved-hybrid mode and 61 seconds for the interleaved mode, as shown in Table 11.1). Taking the OAS geo-ontology with 50% of possible topological relations, then considering there are 46,624,765 calls to the backward engine from the set of interleaved forward integrity rules, the estimated time for reasoning with the interleaved-hybrid mode without tabling

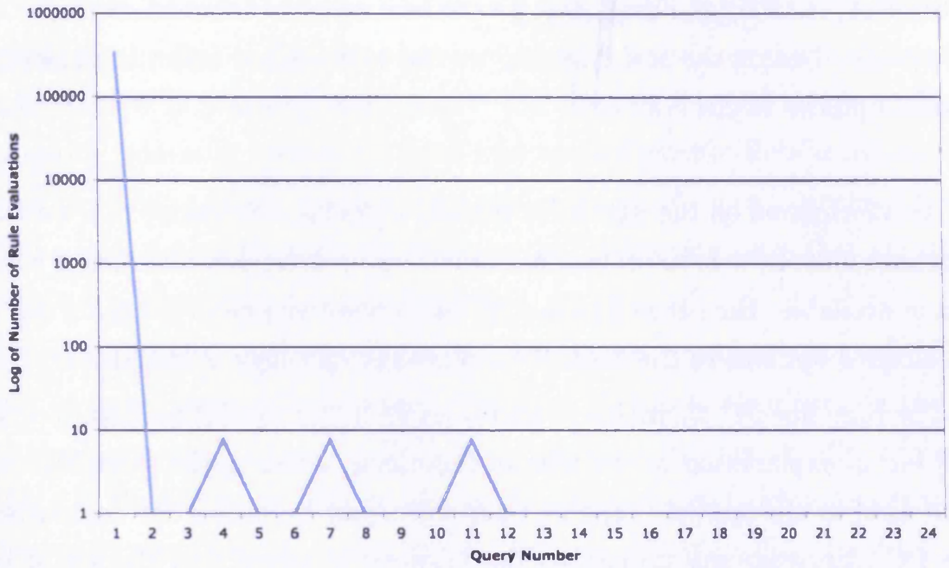


Figure 11.14: Number of rule evaluations per query to the backward system

would be 979 years, and 92 years for the interleaved mode.

As with previous results, tabling combined with the highly interconnected nature of topological reasoning (leading to a large number of necessary rule evaluations), and the memory inefficient nature of Jena2's XSB implementation mean that the in-memory usage of interleaved and interleaved-hybrid reasoning is far greater than using the forward engine alone.

As a last point, because all geometry for the OAS geo-ontology is stored in the LSS, when using interleaved-hybrid, no matter how many relations are stored explicitly in *SWSRLO*, all relations can be evaluated or computed by the LSS and tabled. Hence, even for small numbers of explicit relations in *SWSRLO*, the same number of calls are being evaluated and tabled in the LSS, therefore the in-memory usage remains almost constant as the number of explicit relations increases.

11.4 Forward Rule Engine Performance Under Incremental Updates

Test Number: 11

Test Ontology:(2)

Rule Set(s): $F^{RCC12}_{<D,I>}$,$FBi^{RCC12}_{interleaved}$

Purpose: Testing the scalability of the forward engine under incremental topological updates to *SWSRLO*

As information on the web is continually evolving, the engine of a web ontology language must be able to process new information incrementally as and when it becomes available. Here the efficiency of the forward engine in handling incremental topological updates to the OAS *SWSRLO* geo-ontology is tested.

The base-line PC algorithm is also suitable to process incremental updates (see [89] for an explanation in the temporal domain, which is the same algorithm 2.2 used here in the spatial domain), and will again be tested for comparison. For the PC algorithm any update to the network is added into the set of relations denoted $\mathcal{Q}$ in algorithm 2.2, and changes to the network are propagated until either an inconsistency is found or not. Figure 11.15 shows the time taken for the Rete based spatial reasoning engine using the $F^{RCC12}_{<D,I>}$ ruleset, and the incremental version of base-line PC algorithm to process 3460 topological relational updates - shown per update.

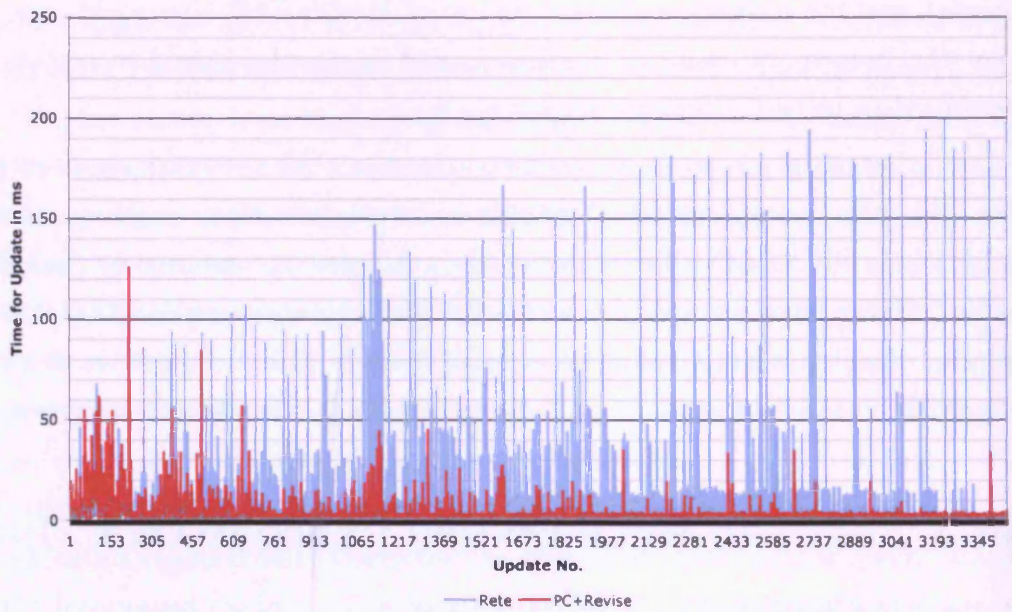


Figure 11.15: Comparison of time take to insert new relations into SWSRL (using the Rete engine) and into the incremental PC+Revise engine

As shown in Figure 11.15 the Rete based forward engine compares well to the base-line PC algorithm. The Rete network handles incremental updates natively, a feature known as temporal redundancy, and does not require the re-application of the entire reasoning process for each update. Effectively Rete maintains state information about existing compositions and converse relations in working memory (or in the Rete network), and only integrity and deduction rules that are effected by the change in working memory are fired.

On average the PC algorithm takes 18 ms to insert, propagate and test the consistency of newly inserted topological relations. The Rete algorithm on the other hand takes on average 53 ms. A PC algorithm which can not handle incremental updates e.g. Dechter et al. [47], or a brute-force rule engine (such as the case of the interleaved forward and backward mode), requires the re-application of the entire reasoning process for each update. For the case of the OAS test geo-ontology here, this would take 10 seconds per update average for the Dechter algorithm and an 8 minute average per update for *SWSRL* in interleaved mode.

Also of note, the Rete engine starts to performs worse as the number of facts already in working memory increases. This can be attributed to the fact that as more relations are added, the Rete network increases in size, uses more memory, and takes longer in general to reason with compared to the PC algorithm, a trend shown in previous results, for example see Figures 11.6 and 11.7.

11.5 EVALUATING THE EFFECTS OF TOPOLOGICAL RELATIONSHIP DISTRIBUTION ON REASONING

Test Number: 12
Test Ontology:(1)(2)
Rule Set(s): F_D^{RCC12}
Purpose: Evaluating the inference potential of forward deduction topological reasoning rules under different distributions of explicit RCC-8 relations

It is known that around 90% of all topological relations in geographic information systems represent disjointness between two regions [137]. Many of these relations can be derived / inferred on the fly by using qualitative spatial reasoning rules, or by computing them when needed from quantitative geometric representations of regions. As a consequence, spatial scenes can be compressed or edge reduced [274] to only contain a smaller subset of explicit topological relations from which some (a lossy compression), or all (a lossless compression), can be derived.

Here we investigate the potential for spatial scene compression over topological relations in an *SWSRLO* geo-ontology using the qualitative spatial ruleset F_D^{RCC12}. Testing is performed over two different types of *SWSRLO* geo-ontologies. A synthetic geo-ontology containing 25 regions, and the OAS+WikiGeo(Primary) *SWSRLO* geo-ontology. Testing with synthetically generated ontologies allows us to control the distribution of topological relations, and then measure how the number of inferred topological relations differs when the number and type of raw topological relations in the ontology change. Table 11.3 shows the relative distribution of topological relations in the real world OAS(2) geo-ontology. As this distribution is representative of real world geo-ontologies, it is used as a template for generating syntheticGA geo-ontologies - although other types of distributions will be tested. The exact distribution of 9 different synthetic geo-ontologies generated is shown in Figure 11.16. Of note, reasoning is performed using RCC-12 compositional inferences (in the set F_D^{RCC12}), the conjunctive set of RCC-12 relations between regions are then converted back to a disjunctive set of RCC-8 relations after reasoning for the analysis which is presented here.

Table 11.3: Relative distribution of topological relations for Wards and Parishes in Cardiff

Inside	Contains	CoveredBy	Disjoint	Overlap	Covers	Equal	Meet
0.88%	0.88%	0.34%	77.78%	9.68%	0.34%	1.61%	8.37%

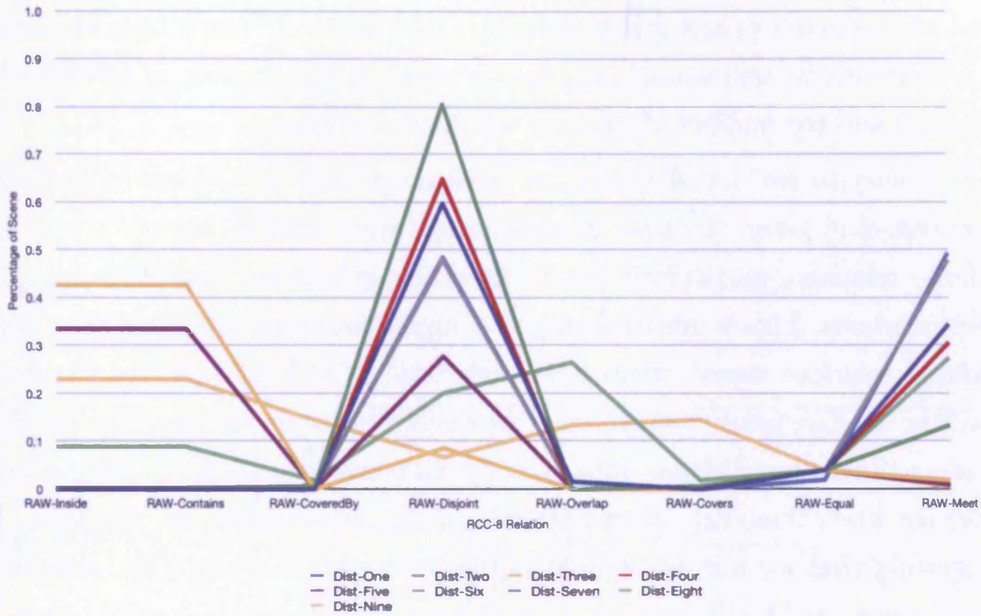


Figure 11.16: RCC-8 distributions for nine topologically edge reduced *SWSRLO* geo-ontologies

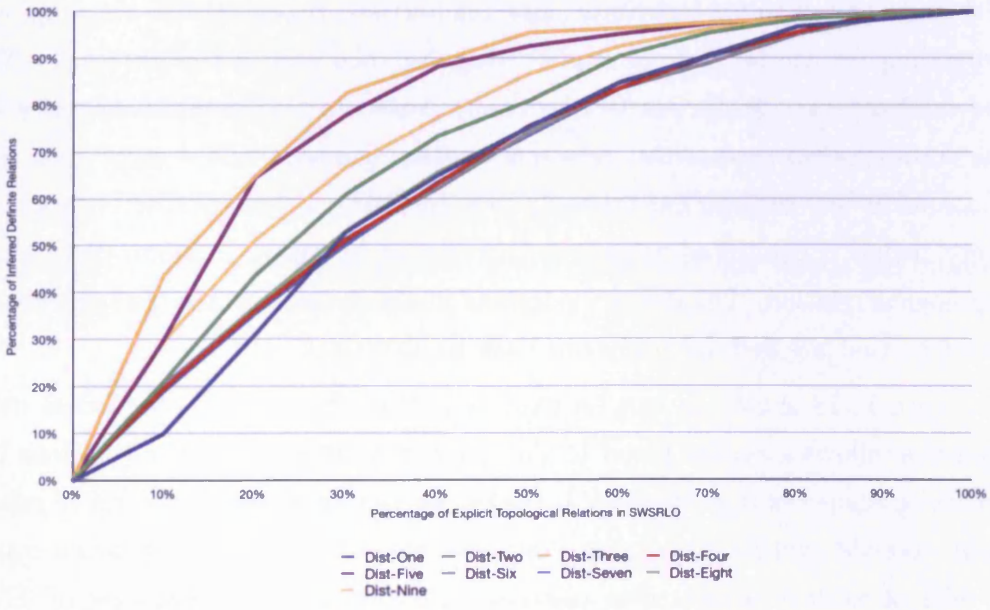


Figure 11.17: Definite relations produced after inference for nine topologically edge reduced *SWSRLO* geo-ontologies

Figure 11.17 reinforces the idea that the number of inferred relations is dependent on the numbers and types of explicit relations in the ontology. Overall there is a clear inverse polynomial correlation between the number of raw topological relations and the number of inferred topological relations.

It is easy to see that distributions with larger amounts of inside and contains relations, and lower amounts of disjointness and meet relations, produce more definite relations, particularly when the ontology has between 10% and 70% explicit relations. This is unsurprising, as compositions only involving the inside and contains relations always produce a single base RCC-8 relation (see Figure 2.1 in Chapter 2). Compositions involving the disjoint relation on the other hand, often produce indefinite relations. Interestingly, all converge to a 100% definite relation coverage when there are around 90-98% of raw relations in the ontology. Hence suggesting that for a realistic distribution of topological relations, a lossy scene compression would still require a large number of explicit relations to be contained in the ontology.

Real World Testing: Testing over the real world OAS(2) geo-ontology shows what can be expected in a true to life setting. Figure 11.18 shows how the number of definite and definite+indefinite relations between regions in the ontology varies, depending on the number of explicit definite relations in the ontology. Where the *total coverage* line refers to how many region to region relations there are in the ontology that are not the universal relation (any one of the eight base RCC-8 relations or any disjunction thereof). For instance, if the coverage is 100% then every region is connected to every other region by either a definite or indefinite topological relation. The *definite relations* line is the percentage of region to region relations that are definite (only one base RCC-8 relation).

Figure 11.18 shows, as can be expected, that the number of inferred definite relations follows a similar trend to the number of inferred definite relations in the synthetic scene testing. However it converges slower to 100% of definite relations than most the synthetic examples - only reaching a 100% definite relation coverage at 96% of explicit relations, as opposed to a slightly lower best case of 90% for the synthetic scene. Here we can see that a relatively low number of raw relations are needed to get at least a narrowing of possible relations (a subset of all eight topological relations) between two regions.

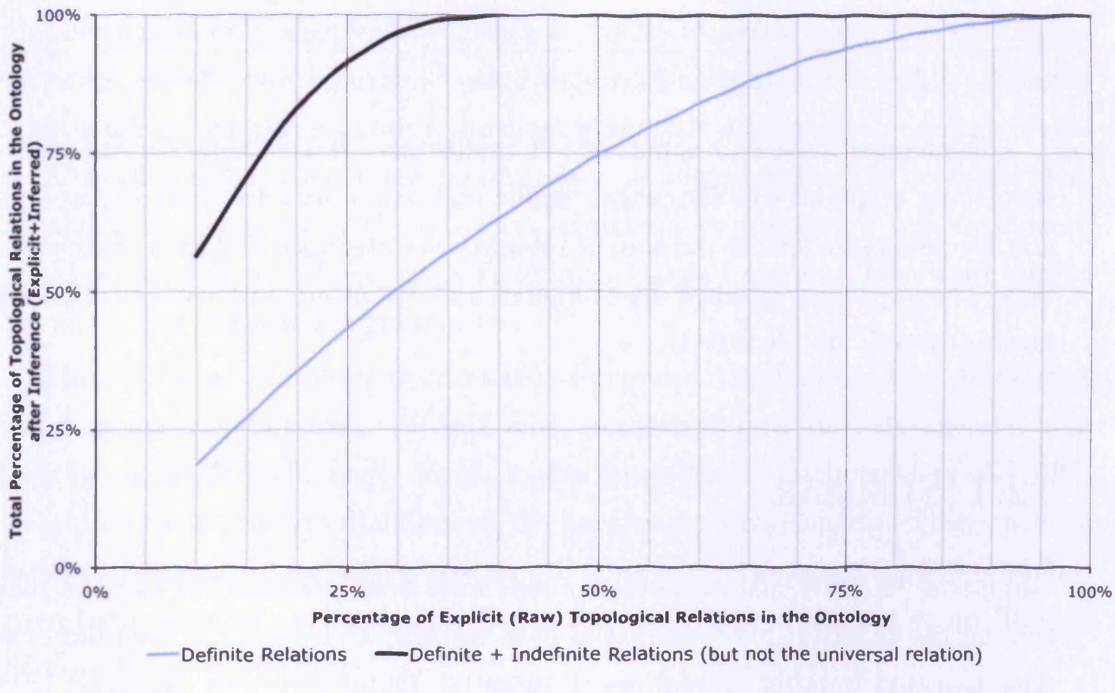


Figure 11.18: The number of derived topological relations versus the number of stored topological relations

CHAPTER 12

CONCLUSION

This chapter presents a summary of this thesis. It concludes on the work of developing a spatial rule language, engine and framework for semantic web geographic ontologies for the purpose of integrity maintenance and knowledge deduction. The important findings are evaluated and discussed, and avenues of possible future research are identified.

12.1 SUMMARY

Geospatial integrity maintenance methods have been presented to assist in maintaining the consistency of geospatial data models, and these methods have been demonstrated working on real-world datasets. Ontologies were discussed briefly from a philosophical context and then in more depth from a computer science context. Geographic ontologies were compared to general ontologies, and their defining characteristics were discussed.

Logical knowledge representation paradigms based on Description Logics and Logic Programs were analysed. The various families of Description Logics, each a different expressive subset of First Order Logic (FOL), constitute a popular knowledge representation language powerful in reasoning about terminological structures. Logic programs are a way of capturing deductive reasoning axioms, or rules, as declarative logical statements, and can incorporate some elements of procedural programming. Most Logic Programs are syntactically equivalent to the Horn fragment of FOL, and can complement Description Logics by adding powerful relational reasoning (reasoning about binary relations).

State of the art semantic web technologies were explored and assessed for their potential to represent, reason with, and maintain geospatial ontologies. The current Web Ontology Language (OWL) was found to be limited with respect to handling the unique requirements of geospatial information, and as an integrity checking language. Various existing approaches to overcome this problem were surveyed, including limited forms of domain closure using autoepistemic and syntactic constructs, through to the integration of Description Logic ontologies with rules or Logic Programs. Different approaches to the integration of spatial logics and ontologies were also reviewed, with a conclusion being made that hybrid approaches, which revolved around using existing Geographic Information Systems with ontology components, were the most pragmatic and applicable to this work. As a result of this survey, a new geo-ontology paradigm, named the Semantic Web Spatial Rule Language ($SWSRL$), and maintenance framework was developed, based on the syntactic fragment of Description Logic Programs, but firmly within the semantics of Logic Programs.

The hybrid geo-ontology maintenance framework combines the newly defined ontology and rule language $SWSRL$ with an external locational storage and processing engine (Oracle 10g). Extra logical procedural attachments in $SWSRL$ were used to call certain functions of the locational storage engine. These include ways to determine qualitative spatial relations between regions in the ontology from their geometric representation, useful in providing a mixture of qualitative and quantitative reasoning.

An OWL-DL OGC and ISO 1901 complaint geo-ontology was converted to, and hence represented in, $SWSRL$. $SWSRL$ as a paradigm is capable of employing 'ontologies on top of rules' or, more specifically, integrates into one language both ontological statements as axioms and facts, with additional user defined rules. Rules are either intended to derived new implicit information from the geo-ontology (deduction rules), or to detect inconsistencies in the geo-ontology (integrity rules).

Qualitative spatial representation and reasoning with topological relations was investigated. How to decide consistency of networks of topological relations using spatial calculi and procedural path-consistency methods was explored. From the literature, tractable subsets of region connection calculus (RCC) topological relations were investigated, where deciding path-consistency for these subsets is enough to decide global consistency in a tractable (polynomial) amount of time. An analysis of how to represent and reason with the relations of RCC within *SWSRL* was carried out. A semantic preserving, alternative syntactic representation of the RCC relations and composition tables was chosen from the literature, as it can be captured fully and natively within *SWSRL*.

Two methods were developed to help localise the source of topological inconsistencies in *SWSRL*. The first is able to suggest a topological relation that can replace an inconsistent relation. The second is able to help locate the exact source of inconsistencies, even if the inconsistency had propagated to further relations in *SWSRLO*.

A number of different *SWSRLO* geo-ontologies were instantiated for testing and evaluation. The administrative geography of Cardiff, Glamorgan and Wales was used to show the applicability of the approach over existing real world geo-ontologies. These where augmented with qualitative topological information about regions in Cardiff and Glamorgan mined from Wikipedia articles. These two sources allowed testing of the hybrid framework in its ability to mix qualitative and quantitative spatial reasoning. A genetic algorithm was developed to instantiate synthetic geo-ontologies with differing numbers of relations and regions in order to perform empirical scalability testing. The results and major findings of this thesis are discussed in the next section.

12.2 Results and Major findings

In this section we review the major findings of this thesis before, in the next section, highlighting future directions for the work developed in this thesis.

The Semantic Web Spatial Rule Language *SWSRL*: After investigation, a Description Logic Program (DLP) was deemed the most appropriate foundation on which to base the new geo-ontology paradigm *SWSRL*. A DLP has a number of significant advantages for representing geographic information. Firstly, OWL-DL ontologies can be mapped into a DLP where, once mapped, a DLP can represent triangular knowledge (property chaining) on top of ontology axioms, which is necessary for spatial reasoning rules. Secondly, a DLP is a Horn logic program that has a polynomial data complexity and EXPTIME combined complexity, which makes it very pragmatic and more tractable than most existing approaches. Of note, a DLP is only a subset of OWL-DL, hence a DLP can not represent, verbatim, the proposed geo-ontology mode. Importantly however, this subset was still suitable to represent all interesting features of the geo-ontology model that OWL-DL could capture, apart from functional and cardinality constraints.

the base DLP was then extended to represent all requirements of the proposed spatial ontology paradigm. The syntactic correspondence between a DLP and logic program (LP) was exploited to create a new language based on the semantics of a LP (and hence Datalog) named the Semantic Web Spatial Rule Language ($SWSRL$). LP semantics are based on the closed world and unique name assumptions which are required for integrity checking tasks. Moreover, LP's have mature implementation engines (of which we chose XSB and a Rete based production system), which are known to scale better to larger instance bases than DL reasoners. Lastly, a LP can include extra-logical procedural attachments, allowing linking with the location storage system LSS in the hybrid framework. It also overcame the limitations of not being able to capture functional and cardinality constraints, as these can be represented as procedural attachments in the LP engine. The base LP Horn implementation in $SWSRL$ was then further extended to deal with default integrity rules, a mixture of qualitative and quantitative reasoning, and spatial data representation and processing in a hybrid framework.

A declarative form of path-consistency was represented directly in $SWSRL$. It was important that, unlike other DL + spatial logic techniques which relied on external procedural path-consistency algorithms, we encoded the path-consistency techniques directly as declarative integrity and deduction rules in $SWSRL$. This was possible thanks to a discovered recently developed topological composition table based on the Region Connection Calculus, which allowed the reasoning engine to effectively close the set of topological relations in the ontology subset of $SWSRL$ ($SWSRLO$) under composition, converse and intersection. In addition if the relations in $SWSRLO$ were in the maximal tractable subset $\mathcal{H}^8$ then reasoning with both integrity and deduction rules decided not only path-consistency but global consistency. Then, deciding consistency of topological relations can be achieved alongside the execution of other integrity and deduction rules including all entailments from the core ontology axioms (now represented as rules) e.g. subsumption reasoning. Of course, should $SWSRLO$ prove to be inconsistent with respect to topological relations, the result of any rule that employs topological relations should be ignored until the inconsistencies in *SWSRLO* are rectified.

Hybrid Framework: The hybrid framework combined a Location Storage System (*LSS*) with the newly defined rule language *SWSRL* and a visual interface. Procedural attachments in the reasoning engine proved successful in linking *SWSRL* to the underlying implementation of the *LSS* (an Oracle spatial database). Beneficially, all spatial operations present in Oracle could then be used during reasoning. This included the computation of topological relations, which was employed within the interleaved mode of *SWSRL* to great effect to detect inconsistencies in the combined Geonames and Wikipedia geo-ontology.

The visual interface offered an instance-oriented view of geofeatures in the *SWSRLO*. Errors were easier to identify through the interface as opposed to looking directly at textual output of the reasoning engine. However, a better way to visualise large number of instances should be considered for future work.

Application Testing: Application testing was important in highlighting the benefit of employing spatial consistency checking over both single source *SWSRLO* geo-ontologies i.e. the WikiGeo geo-ontology, as well as *SWSRLO* geo-ontologies from multiple, diverse sources i.e. the OAS+WikiGeo(Primary) geo-ontology. *SWSRL* was capable of finding 18.75% of known (manually tagged) errors in WikiGeo, and 43.75% of known errors in OAS+WikiGeo(Primary). Although a seemingly low number of errors where identified in the WikiGeo geo-ontology, this result reflects the fact that only 5.5% of all possible topological relations were known. Hence, working with only limited raw knowledge, this result is still acceptable . The ratio of detected errors increased significantly when more explicit topological relations were available using the OAS+WikiGeo(Primary) geo-ontology. This not only shows the increased ability of the language, but the importance of treating information linked (manually or automatically) from diverse sources.

In addition to treating qualitative information from multiple sources, there is a need to combine partial qualitative information ubiquitous in free text content on the web, with incomplete quantitative information becoming more common from free geo-data sources such as Geonames. Such use case was served by the hybrid implementation architecture of *SWSRL*, mixing qualitative information in

SWSRLO with quantitative information in the *LSS*. The architecture was successful at determining inconsistencies between quantitative information in Geonames and qualitative information in Wikipedia. However both relational and composition confidence techniques could not be employed in this mode. Consequently, determining the exact source of inconsistencies is more difficult - where incremental updates where made to the ontology until an inconsistency was detected - and techniques to extend this approach should be investigated. Furthermore, a larger scale evaluation was not possible due to the poor scalability of the backward engine.

Application testing showed the benefits of employing both relational confidence and compositional confidence techniques. Compositional confidence was found to be very accurate at detecting the source of inconsistent topological relations (100% of inconsistent relations where found in each case). Relational confidence was found to be moderately accurate when dealing with *SWSRLO* geo-ontologies with the low numbers of topological relations, where the correct relation was only identified in 57.14% of cases in the WikiGeo ontology - a result obtained by excluding those relations identified by the compositional confidence measure to be inconsistent. However this improved significantly when 43.75% of all possible topological relations where present in the OAS+WikiGeo(Primary) *SWSRLO* geo-ontology. The correct relation was then identified and ranked 1st in 100% of cases. At the very least it was shown that relational confidence can be used as a guide to help a geo-ontology user in rectifying inconsistencies by selecting relations in order of ranking - for which relations ranked toward the top have been shown here to be more likely to solve such inconsistencies

Real World Geo-ontologies: As real world geographic ontologies become more evident and well used e.g. DBPedia and Geonames, the importance of maintaining the consistency of the information they hold increases. It has been shown in this thesis that inconsistencies are apparent even in regulated information resource such as Wikipedia or Geonames. This suggests that users should expect a low confidence in the accuracy of retrieved web based geographic information (in particular for topological information which was tested in this thesis) . However currently the user is not told of such inaccuracies by existing search engines, instead having to discover the accuracy of information for themselves.

From the perspective of maintaining consistency of spatial relations, existing resources are often based on minimal qualitative spatial information e.g. often incomplete parent hierarchies, and only store point based locational references to places. As a result building qualitative relation bases is non-trivial, for example where only approximations of topological relations can be obtained using point based data i.e. finding neighbours using Voronio diagrams. Further, as no spatial extent is provided, size relations are impossible to determine and proximity and directional relations can again only be approximated. Hence an effort must be made to wrap natural language descriptions of place, which often contain qualitative descriptions of size, directional and topological relations, in a machine readable format such as RDF, OWL or directly in *SWSRLO* where the consistency of such information can be checked.

***SWSRL* Spatial Reasoning Engine Implementation:** The empirical evaluation showed the relative performance of reasoning with topological rule sets in *SWSRL* using forward and backward engines on their own and when used interleaved together. The forward reasoning engine which employs the Rete algorithm clearly outperforms the XSB backward reasoning engine. For example, when considering only qualitative deduction topological reasoning rules in the OAS *SWSRLO* geo-ontology, the forward engine can compute all topological inferences in *SWSRLO* using on average around only 25-27% of the number of rule evaluations that the backward engine uses to answer the same query (which has the effect of inferring a large part of the scene, but not necessarily all). Indeed it was shown by rule evaluation counting that in the worst case the backward engine requires significantly more rule evaluations than the forward engine to find the same inferences. Queries to the backward engine often need to evaluate branches of the resolution tree that result in failure. The forward engine on the other hand only runs read-to-fire (ground-able) rules that have essentially already succeeded. This suggests that qualitative spatial reasoning is better served in a forward chaining reasoning engine as opposed to a top down reasoning engine such as XSB.

The premise that interleaving forward and backward reasoning modes would lead to a more efficient use of main memory did not hold up in practice. More specifically, within the XSB engine, all RCC-12 deduction rules needed to be tabled and, as queries need to explore a very large search space (due to the complexity of the rule set and large data complexity of the numbers of relations $O(n^2)$) then a large number of predicates need to be evaluated and tabled. As an example, even for the small* OAS *SWSRLO* geo-ontology, the failed query *EC* took over 330,000 rule evaluations. This combined with the memory inefficient implementation of Jena2's reasoning engine (using a single stack scheduling strategy) led to a much higher increase in in-memory usage over the forward engine. Consequently the interleaved approach did conserve in-memory usage as first hypothesised. That said, in interleaved mode, RCC-12 inferences are not added back to *SWSRLO* during reasoning, hence the number of topological relations in the core geo-ontology does not increase thus helping to reduce persistent storage costs.

As already described, the use of qualitative information in *SWSRLO* in addition to quantitative information in the *LSS* highlight inconsistencies between multiple sources. However the use of the *LSS* within the hybrid rule set lead to an expected increase in both in-memory usage and execution time compared to the interleaved only ruleset which did not contain calls to the *LSS*. This can be attributed to the fact that the qualitative representation of topological relations computed from geometry in the *LSS* were being stored in the XSB table.

In terms of scalability, despite the forward Rete engine having a polynomial execution time in the number of regions (more accurately objects) in *SWSRLO*, this did not lead to tractability for our application. In practice, the Rete based forward engine took 222 seconds to reason with the topological integrity and deduction rule set over 237 regions. Hence, considering this is only a small percentage of the number of regions in a more complete geo-ontology i.e. 0.03% of the number of places in Geonames, it is unlikely that without further optimization the current approach would scale to this level. However the main limitations of the forward approach came not from execution time, but from in-memory overheads, where a 2gb in-memory usage was hit for the same 237 regions. This was exemplified in the backward system which had a much worse in-memory overhead and is substantially slower, taking a predicated 92 years (with tabling) to perform the same set of queries as could be determined after all deductions have been made in the forward engine.

As a final benefit of using the forward engine only, Rete's temporal redundancy characteristic proved suitable for geo-ontologies that undergo continuous update.

*Small in comparison to a more complete geo-ontology containing millions of individuals

Rete stores knowledge of existing RCC-12 inferences so that when a new topological relation is introduced, only affected rules are triggered. This is in contrast to the backward engine or indeed any non-incremental path-consistency algorithm, which would require the re-application of the entire ruleset or path-consistency algorithm each time a topological relations is updated in $SWSRLO$.

A notable difference in the percentage of inferred topological relations to the number of explicit topological relations was found between $SWSRLO$ with the same numbers of regions, but different distributions of topological relations. It was shown that the typical distribution of topological relations (for example as found in the official OAS geo-ontology) did not allow for the biggest gains in inferred definite relations. This can be explained by considering the high percentage of meet and disjoint relations typically found between regions in real world scenarios. When these relations are composed with other relations, the resultant relation is typically a large set of possible relations i.e. the result is vague.

Final Conclusion: In relation to the initial hypothesis of this thesis we found the following. Current ontology languages, as they stand, are not capable of representing spatial information from the web or from official geographic data authorities. Fundamentally, they could not store or process the geometric component of geographical features, did not have spatial reasoning capabilities, and could not maintain the integrity of spatial information. A new language was successfully developed that included spatial representation and reasoning capabilities through the incorporation and integration of state of the art spatial calculi, a homogenous logical ontology language, and an Ad-Hoc Geographic Information System. The language could deal efficiently with geometric information, and could successfully combine and reason with spatial relations computed from quantitative geometry as well as those stored symbolically (qualitatively) in the ontology language. The thesis showed the applicability of the language for maintaining the integrity of, importantly, realistic geographic ontologies that encapsulated both existing web information as well as information from official data authorities. In all, the language proved successful in locating spatial inconsistencies and suggesting how best to rectify those inconsistencies over realistic, although small in scale, geo-ontologies. Despite theoretical tractability of the language, real world testing showed that current reasoning engines combined with the complexity of the employed spatial calculi, and the sheer scale of geographic information lead to severe limitations in the size of geographic ontology that could be reasoned with. Indeed, it is unlikely that without further research and modification, the current language could be employed to manage the evolution of large ontology bases as was first proposed.

12.3 Future Work

The work presented in this thesis leads to a number of future research questions. These can be split into the following high level categories.

(a) Extending the set of *space laws*.

(b) Increasing the scalability of the spatial reasoning engine.

(c) Increasing the expressivity and hence reasoning potential of the spatial rule language.

(d) Extending the user interface of both geo-ontology and rule components.

(e) Real world exploitation.

(f) Instantiating geo-ontologies from the web, or other, sources.

Integrity constraints and space laws: For this thesis, the set of *space laws* was restricted to the pairing of deduction and integrity rules that maintained the topological consistency of regions in *SWSRLO*, using the generalised RCC composition table. However, one area of future work would be to investigate the use of additional *space law* rule sets to maintain other aspects of geo-ontologies. For example, adding additional spatial relation constraints based on relative size, directional and proximity. Such spatial relation constraints could also be combined together, for example see [236]. However, to use other spatial relations in *SWSRL* would involve the development of new composition tables, where compositional inferences are syntactically inside the Horn fragment of FOL. Also, in order to guarantee global consistency using path-consistency, these relations must be within an identified maximal tractable fragment, for example as shown for direction in [241].

Furthermore, in this thesis, bar a few exceptions shown in Chapter 10, we do not perform a full evaluation of integrity rules that utilise thematic only, or spatio-thematic aspects of geographic information. For example, integrity rules could be constructed using knowledge about a specific feature type rather than abstract regions e.g. bridges must be connected at both ends to a land mass, or buildings do not intersect rivers.

Scalability: As highlighted during empirical testing (Chapter 11), the spatial rule reasoner does not scale well to large more realistic geo-ontologies in excess of 200 regions, particularly if the interleaved integrity rule set is used. For this reason, future work could be centred around increasing the scalability of the language to work over larger *SWSRLO* geo-ontologies, and with a better understanding of the theoretical complexities of interleaving both forward and backward engines.

One approach would be to investigate the possibility of segmenting regions and topological relations in *SWSRLO*, such that in-memory reasoning can be performed over only subsets of relations at any one time. Similarly, further work could investigate the benefits (memory overheads) and trade-offs (speed of reasoning) of using the spatial reasoning engine over persistent stores, as opposed to in-memory reasoning only.

Increasing the scalability of the interleaved reasoning mode, in particular how to overcome the performance disadvantage of adding dummy nodes, needs further investigation. Dummy nodes were added to allow incremental updates to trigger integrity rules in the interleaved ruleset. This was achieved whilst not to changing the basic Rete algorithm, thus maintaining compatibility with any Rete based rule engine. However adding dummy nodes significantly increased the time complexity of the program. Hence, a better way to integrate the two engines could be investigated.

Language Expressivity: Presently, the core of *SWSRL* purposely sits within the highly tractable, but not so expressive Horn fragment of FOL. It is possible to envisage a layering of the language to use increasingly more expressive logical constructs. One significant example of this would be to use a more expressive logic fragment that can represent the closed world, non-monotoic negation as failure construct. This would then allow for the expression of typical database style schema constraints. That is, where an integrity rule is violated on the absence of information, as opposed to integrity rules in this thesis which only fire when a positive example of the constraint is found.

User interface: The current working system offers a prototype visual interface to author rulesets in *SWSRL* and to view geographic features in *SWSRLO*. Such an interface could be improved in the following ways. Firstly, extending the error tracing view to include options for the semi-automatic rectification of errors in the geo-ontology, using the relational and compositional confidence measures. Secondly, the 'rule-tree' based view of a rule set could be expanded to show interdependencies between rule predicates (which usually form directed acyclic graphs DAG's). Lastly, the rule editor could be based on a visual rule workflow, whereby a user can visually chain together relations, properties and classes to form either deduction or integrity rules.

Real world exploitation: With the increasing paradigm shift to the web of knowledge (as opposed to knowledge in traditional databases), it is important that information added to geographic (or place) ontologies from web sources does not contradict existing, known, geographic knowledge. Hence, in order to show the feasibility of the language and engine under real world conditions (and indeed semantic web technologies in general) over the web of knowledge, the system could be integrated as a backend to existing geographic web sources. For example, geographic information from Wikipedia articles are parsed into RDF triples by DBpedia *, and by adding these to *SWSRL*, the integrity of the information could be maintained and reasoned with - where currently the information is only stored and queried. A step toward this was presented in this thesis, where an *SWSRLO* geo-ontology was constructed from mined qualitative information from Wikipedia articles about regions in Cardiff, and found to be inconsistent.

Geo-ontology instantiation: During this thesis, two general methods for instantiating geo-ontologies was presented. The first produced synthetic geo-ontologies using a genetic algorithm, and the second used web mining techniques to extract qualitative information from web sources. Both these methods could be used as part of further research. The genetic algorithm could be employed for other spatial reasoning tasks. For example to help generate two dimensional visual interpretations of a set of spatial constraints - work which has been started in [233].

Extension of the web mining techniques could be used to help instantiate, on the fly, geo-ontologies from web 2.0 sources. Certain existing techniques are being developed in this way [205], however these concentrate on acquiring point referenced locations of place, limiting the ability to extract rich qualitative information.

*DBPedia also convert non-geographic information, see `http://dbpedia.org/About`

Other areas: Lastly as a point of future work, the concept of an error ontology and rule meta-data could be taken further. Rule meta-data, along with the representation of individual rule predicates, allows for the possible construction of a rule ontology supporting ontological reasoning about rule sets e.g testing if rules are consistent with respect to other rules - supporting rule set development. Finally, further investigation could be carried out on the use of the error ontology, with the aim of improving the developed error localisation techniques.

APPENDIX A

APPENDIX A

A.1 GENETIC ALGORITHM V RANDOM SEARCH FOR SPATIAL SCENE GENERATION

In this section we illustrate, empirically, how effective the Genetic Algorithm (GA) is compared to a baseline random search (RS) method. Both techniques will try to generate a consistent configuration of topological relations between 20 regions, using an even set δ of weightings for each RCC-8 relation (see section 9.1.2 for a definition of δ):

$$\delta = \{0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125\} \tag{A.1}$$

The GA has 20 possible solutions per generation - a population size of 20. For consistency, the RS is based on the same framework as the GA, using the idea of a population and generations. However, each generation has 20 solutions based on a random selection of RCC-8 base relations. Each new generation is then produced by generating 20 new random solutions.

Both the GA and RS were run twice. The first run of the GA performed worse than the second run, taking 25731 generations to find a consistent scenario (see Figure A.1). Consequently, to make it a fair test, the RS was run twice for the same amount of generations as the worst GA run. The results are shown in Figure A.1.

Clearly the GA is superior to the RS. Unsurprisingly, both RS runs do not find a

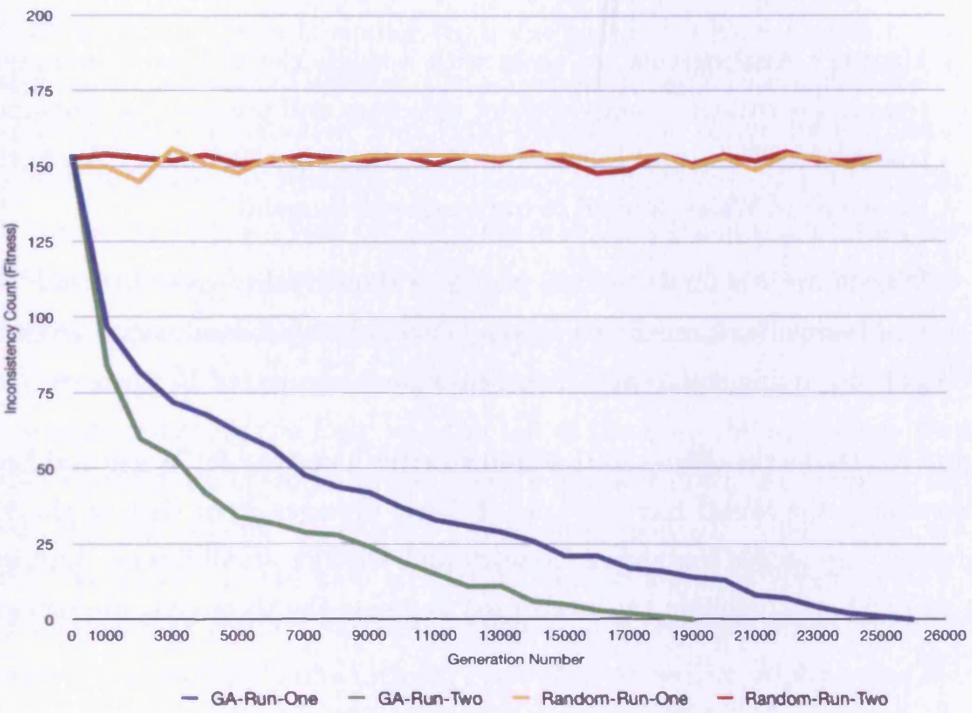


Figure A.1: GA v Random Search Comparison

consistent solution. The GA on the other hand, converges to a consistent solution in 25731 generations for run one, and 18391 generations for run two. It is unlikely that the RS would find a consistent solution in a tractable amount of time, hence justifying the use of the more intelligent GA.

A.2 Ordnance Survey Official Administrative Hierarchy

The formal definition of the subdivisions of the Ordnance Survey Super Output wards/parishes (CASWA/CASPAR) is shown below (description taken from the official Ordnance Survey definition*):

- **Civil Parishes** are often the lowest level of local government. The Civil Parish was abolished in Wales and Scotland in 1974 and 1975 respectively,

*See - http://www.ordnancesurvey.co.uk/oswebsite/ontology/v1/AdministrativeGeography.htm instead being renamed as communities.

- **Unitary Authorities** are areas with a single tier of local government (replacing the two-tier county-district structure still prevalent in England). Introduced in Wales and England in 1996, there are 22 Unitary Authorities in the whole of Wales, and 46 in only parts of England.

- **Wards** are electoral districts within a municipality used in local politics. Wards are often named after through fares, parishes, landmarks, geographical features and in some cases historical figures connected to the area.

Figure A.2 shows the official 2001 administrative hierarchy for Wales and England. Interestingly the Welsh hierarchy, as of 2001, diverges from that of the English hierarchy - no longer containing Country and District subdivisions. Importantly, the shaded boxes highlight the levels used to create the three geo-ontologies in this section.

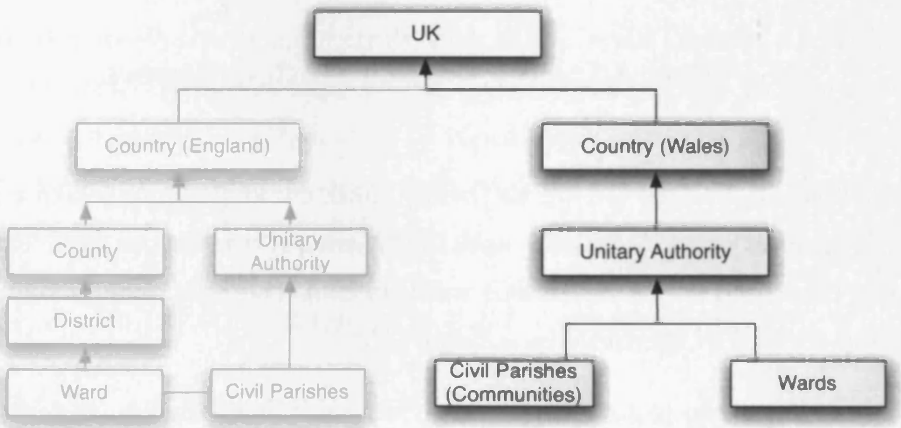


Figure A.2: 2001 Administrative Hierarchy of England and Wales

A.3 Overview of the Rete Algorithm

The Rete algorithm was developed by Charles Forgey as part of his PhD thesis. Rete is latin for 'net' and thus network. Rete has many advantages over a simple exhaustive forward inferencing approach. Rete's efficiency stems from the idea that the knowledge base is similar from one match-resolve-act cycle to the next - temporal redundancy. Therefore the system only needs to evaluate the changes to working memory. Another important optimization to the Rete is the idea of structural dependency, whereby redundancy is eliminated by node sharing. As such, exploiting the fact that certain rules will share patterns, and they need not be evaluated more than once, provided the partial matches are stored.

Rete is implemented by using a discrimination network of alpha (1-input) and beta (2-input)nodes. Discrimination refers to the discriminate filtering of facts (or patterns) entering the Rete. At the top of the Rete the nodes may contain a number of matches, however as the data propagates down the network, facts are sieved, and the amount of matches per node is reduced. Alpha nodes are used to test conditions - in the case of RDF triples, an alpha node represents a triple pattern, that may contain variables. Such a node tests the input triple and if it succeeds it is stored (in alpha memory) and then passed on. Alpha nodes therefore form the pattern network of the Rete.

A beta node is used as a join node. The beta node integrates the facts (triples) from left and right inputs, whereby left and right inputs are both alpha nodes. The joins are then propagated further down the Rete when suitable joins are found.

At the bottom of the Rete is a terminal node or p-node (production node). Once the token has propagated all the way down to the terminal node, the rule's left hand side has completely matched, and the ready to fire rule and is placed onto an agenda. Conflict resolution is then used to determine the order of which rules are to be executed on the agenda. Conflict resolution is implemented outside of the Rete, and can be different per implementation. The order rules are executed can be as simple as random selection, to selection based on rule priority - salience.

Once completed the rule's head or right hand side is inserted into working memory, which will again propagate through the Rete, and maybe trigger another set of rules to be placed onto the agenda. This cycle (match-resolve-conflict) continues until no more rules left hand sides have matched and the system is halted. In classical logic programming this point is often referred to as the least fixed point of the program.

A.4 GEONAMES TOPONYM ONTOLOGY MODEL

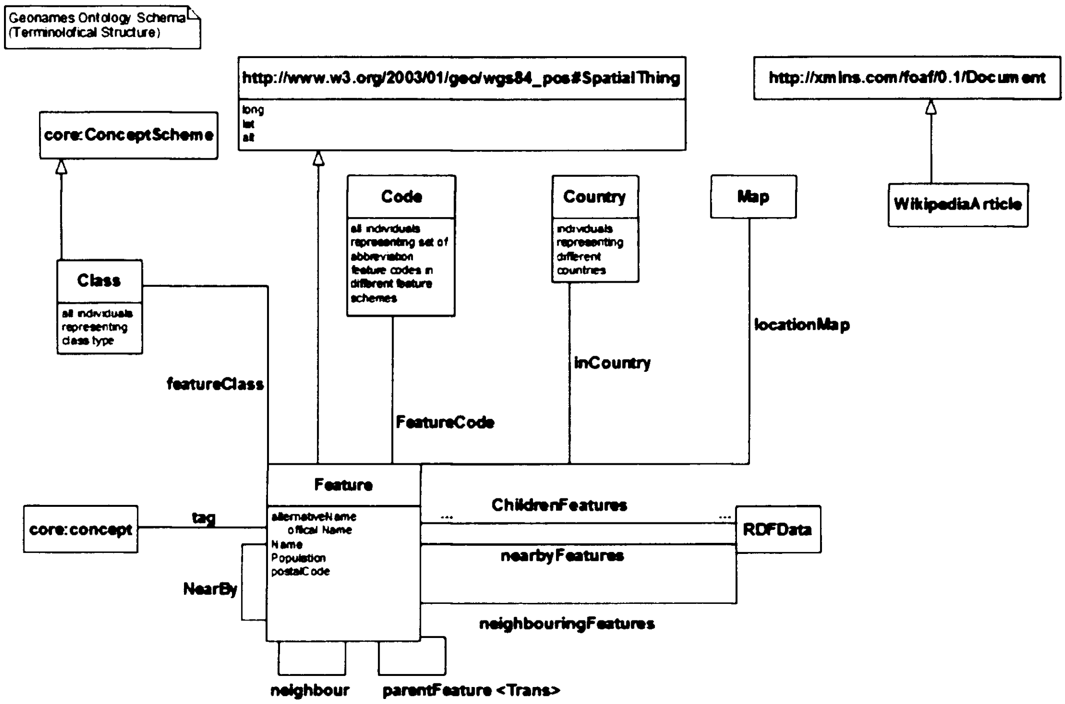


Figure A.3: Geonames Toponym Ontology Model

A.5 SEMANTIC WEB REASONING SYSTEMS

Much progress has been made in the past few years in the storage and reasoning/querying of semantic web ontologies. Table A.1 illustrates and compares eight prominent semantic web ontology systems.

Table A.1: Semantic Web Systems Comparison

System	Ontology Standard	Memory Model	Persistent Model	Querying	Reasoning
Jena2	RDF to OWL-DL	√	√	RDF triple queries using either SPARQL and or a custom API	RDF(S), OWL-Lite, near complete OWL-DL and partial OWL-Full inbuilt forward and backward reasoners
Instance Store	Partial OWL, designed for large instance data (Abox information) storage and retrieval	√	×	Query by inference lookup	OWL TBox reasoning via link to existing DL reasoner (via a DIG interface). Efficient Abox Reasoning by combining DL reasoners with Database queries
Redland	RDF triple storage	√	√	RDQL and SPARQL RDF querying	NA
OpenRDF.org Sesame	RDF(S), via some enhancements supports OWL-DL	√ [a]	√	B-Tree indexing of RDF triples. Seasame RDF Query Language (SeRQL), supports numerous advanced features e.g. RDF graph transformation, RDF Schema support and XML Schema datatype support	RDF(S) plus custom axioms and rules (OWL-DL also possible)
Kowari	RDF triple based	√	×	interactive Tucana Query Language (iTQL). RDF query language with AVL index's and a 64 but wide data struture to improve query speed.	OWL Class property inheritance only
KOAN2	Datalog (Horn Subsets)	√ [b]	√	Simple query language under development	Datalog Reasoning engine (hence Horn like reasoning)
Oracle10g	RDF triples	×	√	Indexed Storage and efficient RDF querying (using a newly defined custom Oracle Datatype). Millions of triples can be queried in the order of seconds [31]	RDF(S) and user defined rules
rdfDB	RDF triple storage	×	√	Highly scalable triple based querying	RDF(S)

[a] Using a Storage and Inference Layer (SAIL) API, any implementation storage format is possible

[b] Can bound to any storage implementation through a suitable interface

APPENDIX B

APPENDIX B - *SWSRL* LANGUAGE CONSTRUCTS AND RULESETS

B.1 SPATIAL RULE SETS EXAMPLES

B.1.1 $GeoR_{bk}^{standard}$ AND F_D^{RCC12} EXAMPLES

Example rules from the rulesets $Fd_{RCC^{12}}$ and $GeoR_{bk}^{standard}$ are shown below.

[<label>DR_0</label><ruleLevel>0</ruleLevel><ruleGroup>Topological</ruleGroup
<ruleType>1</ruleType><ruleClass>1</ruleClass><exeNumber>1
</exeNumber><backGroup>TopologicalDeduction</backGroup> : coP-1(?a ?c) ← DR(?a ?b) AND O(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c)]

(B.1)

[<label>copli</label><ruleLevel>0</ruleLevel><ruleGroup>TopologicalConverse
</ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass><exeNumber>1</exeNu
<backGroup>TopologicalMapping</backGroup> : coP(?B ?A) ← coP-1(?A ?B) AND C:Region(?A) AND C:Region(?B)]

(B.2)

[<label>P_P</label><ruleLevel>0</ruleLevel><ruleGroup>Topology</ruleGroup> <ruleType>1</ruleType><ruleClass>1</ruleClass> : P(?a ?b) AND P(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) → P(?a ?c)]

(B.3)

[<label>ntpi</label><ruleLevel>0</ruleLevel><ruleGroup>TopologicalConverse </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass> : NTP(?a ?b) → NTP-1(?b ?a)]

(B.4)

[<label>dri</label><ruleLevel>0</ruleLevel><ruleGroup>TopologicalConverse </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass> : DR(?a ?b) → DR(?b ?a)]

(B.5)

[<label>ec1Map</label><ruleLevel>0</ruleLevel><ruleGroup>TopologicalMappings </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass> : EC(?a ?b) AND C:Region(?a) AND C:Region(?b) → C(?a ?b)]

(B.6)

[<label>ec2Map</label><ruleLevel>0</ruleLevel><ruleGroup>TopologicalMappings </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass> : EC(?a ?b) AND C:Region(?a) AND C:Region(?b) → DR(?a ?b)]

(B.7)

B.1.2 Forward Integrity Rule Examples

Example rules from F_I^{RCC12} that detect an inconsistency based on the composition of C and DR are shown in B.8 and B.9.

[<label>C_DC1_I-1-</label><ruleLevel>0</ruleLevel><ruleGroup>Topological </ruleGroup><ruleType>0</ruleType><ruleClass>1</ruleClass> : C(?a ?b) AND DR(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) AND P(?a ?c) → error(C_DR ?a topologicalError ?c ?b ?count invalidSpatial_Relationship C_DR)]

(B.8)

[<label>C_DC1_I-2-</label><ruleLevel>0</ruleLevel><ruleGroup>Topological </ruleGroup><ruleType>0</ruleType><ruleClass>1</ruleClass> : C(?a ?b) AND DR(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) AND NTP(?a ?c) → error(C_DR ?a topologicalError ?c ?b ?count invalidSpatial_Relationship C_DR)]

(B.9)

B.1.3 Combined Forward Integrity Deduction Ruleset

As an example of a forward integrity deduction rule pairing, consider the compositional inference in B.10 represented as the integrity, deduction pairing in B.11 and B.12.

$$C(a\ b) \wedge DC(b\ c) \rightarrow coP(a\ c) \qquad (B.10)$$

[<label>C_DC</label><ruleLevel>0</ruleLevel><ruleGroup>Topological </ruleGroup><ruleType>1</ruleType><ruleClass>1</ruleClass> : C(?a ?b) AND DC1(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) → coP(?a ?c)]

(B.11)

[<label>C_DC_I</label><ruleLevel>0</ruleLevel><ruleGroup>Topological

`</ruleGroup><ruleType>0</ruleType><ruleClass>1</ruleClass>` : C(?a ?b) AND DC(?b ?c) AND C:Region(?a) AND C:Region(?b) AND C:Region(?c) AND P(?a ?c)) → error(C_DC ?a topologicalError ?c ?b ?count invalidSpatial_Relationship C_DC)]

(B.12)

B.2 DLP Transformation Function $\mathcal{T}$

The DLP transformation function $\mathcal{T}$ from [105] is shown in this section. C represents a class from the language $\mathcal{L}_b$, D represents a class from the language $\mathcal{L}_h$ and S and B represent classes from the language $\mathcal{L}$. R, P and Q are properties, A is an atomic class name and x and y are variables. In the languages $\mathcal{L}_h$, $\mathcal{L}_b$ and $\mathcal{L}$, A is an atomic named class and C and D are classes. The language $\mathcal{L}_h$ then represents the valid constructs: if C and D are classes, then $C \sqcup D$ is also a class that can occur in the l.h.s of inclusion axioms. The language $\mathcal{L}_b$ then represents the valid constructs: if D and C are classes and R is a property then $C \sqcup D$ and $\exists R.C$ are classes that can occur in the r.h.s of inclusion axioms. The language $\mathcal{L}$ represents the intersection of both $\mathcal{L}_b$ and $\mathcal{L}_h$ and contains the valid constructs: if S and B are classes, then $S \sqcap B$ is also a class. As highlighted above, the distinction in $\mathcal{L}_b$ and $\mathcal{L}_h$ and $\mathcal{L}$ exists so as to avoid mapping those constructs that form invalid definite Horn rules.

$$
\begin{aligned}
\mathcal{T}(C \sqsubseteq D) &\longrightarrow Th(D,y) \leftarrow Tb(C,y)\\
Th(A,x) &\longrightarrow A(x)\\
Th((C \sqcap D),x) &\longrightarrow Th(C,x) \wedge Th(D,x)\\
Th((\forall R.C),x) &\longrightarrow Th(C,y) \leftarrow R(x,y)\\
Tb(A,x) &\longrightarrow A(x)\\
Tb((C \sqcap D),x) &\longrightarrow Tb(C,x) \wedge Tb(D,x)\\
Tb((C \sqcup D),x) &\longrightarrow Tb(C,x) \vee Tb(D,x)\\
Tb((\exists R.C),x) &\longrightarrow R(x,y) \wedge Tb(C,y)
\end{aligned}
$$

$$\mathcal{T}(\top \sqsubseteq \forall P.D) \longrightarrow Th(D,y) \leftarrow P(x,y)$$
$$\mathcal{T}(\top \sqsubseteq \forall P^{-}.D) \longrightarrow Th(D,x) \leftarrow (Px,y)$$
$$\mathcal{T}(a : D) \longrightarrow Th(D,a)$$
$$\mathcal{T}(< a,b >: P) \longrightarrow P(a,b)$$

$$\mathcal{T}(P \sqsubseteq Q) \longrightarrow Q(x,y) \leftarrow P(x,y)$$
$$\mathcal{T}(P \equiv Q) \longrightarrow \left\{ \begin{array}{l} Q(x,y) \leftarrow P(x,y) \\ P(x,y) \leftarrow Q(x,y) \end{array} \right]$$
$$\mathcal{T}(P \equiv Q^{-}) \longrightarrow \left\{ \begin{array}{l} Q(x,y) \leftarrow P(y,x) \\ P(y,x) \leftarrow Q(x,y) \end{array} \right]$$
$$\mathcal{T}(P^{+} \sqsubseteq P) \longrightarrow P(x,z) \leftarrow P(x,y) \wedge P(y,z)$$
$$\mathcal{T}(S \equiv B) \longrightarrow \left\{ \begin{array}{l} \mathcal{T}(S \sqsubseteq B) \\ \mathcal{T}(B \subseteq S) \end{array} \right]$$

In addition to the function $\mathcal{T}$ above, the following rule simplification transformations are made to rewrite any rules that have the form:

$$(H \wedge H^{'}) \leftarrow B \; to \; (H \leftarrow B \; and \; H^{'} \leftarrow B) \quad \text{Lloyd Topor transformation}$$
$$H \leftarrow (B \vee B^{'}) \; to \; (H \leftarrow B \; and \; H \leftarrow B^{'}) \quad \text{Lloyd Topor transformation}$$
$$(H \leftarrow H^{'}) \leftarrow B \; to \; H \leftarrow (B \wedge H^{'}) \quad \text{UniversalRestrictions}$$

B.3 DLP GEO-ONTOLOGY AXIOMISATION

Table B.1: Full Geo-ontology Axiomisation

Axiom Number	DL Syntax
1	$\top \sqsubseteq \forall$AlternativeName.xsd:string
2	$\top \sqsubseteq \forall$Name.xsd:string
3	$\top \sqsubseteq \leq 1$Name.$\top$
4	$\top \sqsubseteq \forall$RelatedTerm^{-1}.GEOFEATURE
5	$\top \sqsubseteq \forall$AlternativeName^{-1}.GEOFEATURE
6	$\top \sqsubseteq \forall$Name^{-1}.GEOFEATURE

7	$\mathsf{Within} \sqsubseteq \mathsf{Topological}$
8	$\mathsf{WestOf} \sqsubseteq \mathsf{Cardinal\_Direction}$
9	$\mathsf{NorthOf} \sqsubseteq \mathsf{Cardinal\_Direction}$
10	$\mathsf{Topological} \sqsubseteq \mathsf{Spatial\_Relationship}$
11	$\mathsf{CoveredBy} \sqsubseteq \mathsf{Topological}$
12	$\mathsf{EastOf} \sqsubseteq \mathsf{Cardinal\_Direction}$
13	$\mathsf{SouthOf} \sqsubseteq \mathsf{Cardinal\_Direction}$
14	$\mathsf{EqualTo} \sqsubseteq \mathsf{Relative\_Size}$
15	$\mathsf{Meets} \sqsubseteq \mathsf{Topological}$
16	$\mathsf{Disjoint} \sqsubseteq \mathsf{Topological}$
17	$\mathsf{Covers} \sqsubseteq \mathsf{Topological}$
18	$\mathsf{Equal} \sqsubseteq \mathsf{Topological}$
19	$\mathsf{SmallerThan} \sqsubseteq \mathsf{Relative\_Size}$
20	$\mathsf{Cardinal\_Direction} \sqsubseteq \mathsf{Spatial\_Relationship}$
21	$\mathsf{LargerThan} \sqsubseteq \mathsf{Relative\_Size}$
22	$\mathsf{Overlaps} \sqsubseteq \mathsf{Topological}$
23	$\mathsf{Relative\_Size} \sqsubseteq \mathsf{Spatial\_Relationship}$
24	$\mathsf{Contains} \sqsubseteq \mathsf{Topological}$
25	$\top \sqsubseteq \forall \mathsf{RelatedTerm}.\textsc{GeoFeature}$
26	$\textsc{Region} \sqsubseteq \textsc{GeoFeature}$
27	$\top \sqsubseteq \forall \mathsf{Spatial\_Relationship}^{-1}.\textsc{GeoFeature}$
28	$\top \sqsubseteq \forall \mathsf{Spatial\_Relationship}.\textsc{GeoFeature}$
29	$\textsc{notError} \sqsubseteq \textsc{ErrorRecord}$
30	$\textsc{Error} \sqsubseteq \textsc{ErrorRecord}$
31	$\top \sqsubseteq \forall \mathsf{IndividualFrom}^{-1}.\textsc{ErrorRecord}$
32	$\top \sqsubseteq \forall \mathsf{Individual}^{-1}.\textsc{ErrorRecord}$
33	$\top \sqsubseteq \forall \mathsf{Individual}.\textsc{GeoFeature}$
34	$\top \sqsubseteq \forall \mathsf{IndividualFrom}.\textsc{GeoFeature}$
35	$\top \sqsubseteq \forall \mathsf{Relationship}^{-1}.\textsc{ErrorRecord}$
36	$\top \sqsubseteq \forall \mathsf{Relationship}.\mathsf{xsd{:}string}$
37	$\top \sqsubseteq \forall \mathsf{EName}.\mathsf{xsd{:}string}$
38	$\top \sqsubseteq \forall \mathsf{EName}^{-1}.\textsc{ErrorRecord}$
39	$\top \sqsubseteq \forall \mathsf{EValue}.\mathsf{xsd{:}string}$
40	$\top \sqsubseteq \forall \mathsf{EValue}^{-1}.\textsc{ErrorRecord}$
41	$\top \sqsubseteq \forall \mathsf{EGeneratingRule}.\mathsf{xsd{:}string}$
42	$\top \sqsubseteq \forall \mathsf{EGeneratingRule}^{-1}.\textsc{ErrorRecord}$

43	$\top \sqsubseteq \forall$EProperty.xsd:string
44	$\top \sqsubseteq \forall$EProperty^{-1}.ERRORRECORD
45	$\top \sqsubseteq \forall$EDescription.xsd:string
46	$\top \sqsubseteq \forall$EDescription^{-1}.ERRORRECORD
47	$\top \sqsubseteq$ RECORD
48	$\top \sqsubseteq \forall$Relation.xsd:string
49	$\top \sqsubseteq \forall$Relation^{-1}.RECORD
50	$\top \sqsubseteq \forall$RuleName.xsd:string
51	$\top \sqsubseteq \forall$RuleName^{-1}.RECORD
52	$\top \sqsubseteq \forall$From.GEOFEATURE
53	$\top \sqsubseteq \forall$From^{-1}.RECORD
54	$\top \sqsubseteq \forall$To.GEOFEATURE
55	$\top \sqsubseteq \forall$To^{-1}.RECORD
56	$\top \sqsubseteq \forall$Via.GEOFEATURE
573	$\top \sqsubseteq \forall$Via^{-1}.RECORD

B.4 TRANSFORMED *SWSRLO* GEO-ONTOLOGY USING $\mathcal{T}$

Table B.2: *SWSRLO* Geo-ontology

Axiom Number	LP Horn Syntax
1	AlternativeName(x,y) $\rightarrow$ xsd:String(y)
2	Name(x,y) $\rightarrow$ xsd:String(y)
3	Name(x,y) $\wedge$ Name(x,z) $\rightarrow$ Equal(y,z)
4	RelatedTerm(x,y) $\rightarrow$ GEOFEATURE(x)
5	AlternativeName(x,y) $\rightarrow$ GEOFEATURE(x)
6	Name(x,y) $\rightarrow$ GEOFEATURE(x)
7	Within(x,y) $\rightarrow$ Topological(x,y)
8	WestOf(x,y) $\rightarrow$ Cardinal_Direction(x,y)
9	NorthOf(x,y) $\rightarrow$ Cardinal_Direction(x,y)
10	Topological(x,y) $\rightarrow$ Spatial_Relationship(x)
11	CoveredBy(x,y) $\rightarrow$ Topological(x,y)
12	EastOf(x,y) $\rightarrow$ Cardinal_Direction(x,y)
13	SouthOf(x,y) $\rightarrow$ Cardinal_Direction(x,y)
14	EqualTo(x,y) $\rightarrow$ Relative_Size(x,y)

15	Meets(x,y)	→	Topological(x,y)
16	Disjoint(x,y)	→	Topological(x,y)
17	Covers(x,y)	→	Topological(x,y)
18	Equal(x,y)	→	Topological(x,y)
19	SmallerThan(x,y)	→	Relative_Size(x,y)
20	Cardinal_Direction(x,y)	→	Spatial_Relationship(x)
21	LargerThan(x,y)	→	Relative_Size(x,y)
22	Overlaps(x,y)	→	Topological(x,y)
23	Cardinal_Direction(x,y)	→	Spatial_Relationship(x)
24	Contains(x,y)	→	Topological(x,y)
25	RelatedTerm(x,y)	→	Geofeature(y)
26	Region(x)	→	Geofeature(x)
27	Spatial_Relationship(x,y)	→	Geofeature(y)
28	Spatial_Relationship(x,y)	→	Geofeature(x)
29	NotError(x)	→	ErrorRecord(x)
30	Error(x)	→	ErrorRecord(x)
31	IndividualFrom(x,y)	→	ErrorRecord(x)
32	Individual(x,y)	→	ErrorRecord(x)
33	IndividualFrom(x,y)	→	Geofeature(y)
34	Individual(x,y)	→	Geofeature(y)
35	ERelationship(x,y)	→	xsd:String(y)
36	ERelationship(x,y)	→	ErrorRecord(x)
37	EName(x,y)	→	xsd:String(y)
38	EName(x,y)	→	ErrorRecord(x)
39	EVaue(x,y)	→	xsd:String(y)
40	EValue(x,y)	→	ErrorRecord(x)
41	EGeneratingRule(x,y)	→	xsd:String(y)
42	EGeneratingRule(x,y)	→	ErrorRecord(x)
43	EProperty(x,y)	→	xsd:String(y)
44	EProperty(x,y)	→	ErrorRecord(x)
45	EDescription(x,y)	→	xsd:String(y)
46	EDescription(x,y)	→	ErrorRecord(x)
47	Relation(x,y)	→	Record(x)
48	RuleName(x,y)	→	Record(x)
49	RuleName(x,y)	→	xsd:String(y)
50	Relation(x,y)	→	xsd:String(y)

51	From(x,y) → RECORD(x)
52	From(x,y) → GEOFEATURE(y)
53	To(x,y) → RECORD(x)
54	To(x,y) → GEOFEATURE(y)
55	Via(x,y) → RECORD(x)
56	Via(x,y) → GEOFEATURE(y)

B.5 Procedural Attachments - SQL Queries

Assuming the GeoLSS contains one table named LocationBase, and $< ind1 >$ and $< ind2 >$ are two input Geofeatures.

Distance: SELECT SDO_GEOM.SDO_DISTANCE(loce.shape, loca.shape, 0.005,unit FROM locationBase loce, locationBase loca WHERE loce.rdfID = '$< ind1 >$ AND loca.rdfID = '$< ind2 >$

Area: SELECT SDO_GEOM.SDO_AREA(loce.shape, 0.005,unit=KM) FROM locationBase loce WHERE loce.rdfID = $< ind1 >$

exRCC-8 Spatial Relation: SELECT c_b.rdfID, c_d.rdfID, SDO_GEOM.RELATE(c '$< SRel >$', c_d.shape, 0.005) FROM locationBase c_b, locationBase c_d WHERE c_b.rdfID = $< ind1 >$ AND c_d.rdfID =$< ind2 >$

Where <SRel> is one of, Contains, Covers, Inside, Disjoint, Equal, Touch, Overlapbyintersect or CoveredBy.

B.6 Spatial Rule Metadata

B.6.1 Rule Groups

B.7 *SWSRL* Procedural Attachments

B.8 Standard Operators

Table B.3: Rule Groups

Rule Group	Semantics
Topological	Detecting inconsistency between topological relations without considering geofeature semantics, i.e. if any geofeature a contains a geofeature b, b can not also contain a
Topo-Semantic	Using the semantics of geofeature types to detected inconsistencies in topological relations e.g. a lake can not be inside a motorway
Directional	Detecting inconsistency between directional relations without considering geofeature semantics, i.e. if any geofeature a is to the west of a geofeature b, b can not be south of a
Directional-Semantic	Using the semantics of geofeature types to detected inconsistencies in directional relations
Relative Size	Detecting inconsistency between relative size relations without considering geofeature semantics, i.e. if any geofeature a is larger than a geofeature b, b can not be the same size as a
Size-Semantic	Using the semantics of geofeature types to detected inconsistencies in relative size relations e.g. a ward is not bigger than its containing district
User Defined	Any other user defined rule
Geometric	Maintaining the integrity of geofeature geometry e.g. a polygon must have 3 or more vertices

Table B.4: SWSRL Comparison Operators

Built-in	Arguments	Semantics
PropertyIsEqualTo	(P_1, V)	Returns true iff $P_1 = V$ where P_1 and $V \in \mathbf{R}$
PropertyIsNotEqualTo	(P_1, V)	Returns true iff $P_1 \neq V$ where P_1 and $V \in \mathbf{R}$
PropertyIsLessThan	(P_1, V)	Returns true iff $P_1 < V$ where P_1 and $V \in \mathbf{R}$
hline PropertyIsGreaterThan	(P_1, V)	Returns true iff $P_1 > V$ where P_1 and $V \in \mathbf{R}$
PropertyIsLessThanOrEqualTo	(P_1, V)	Returns true iff $P_1 \leq V$ where P_1 and $V \in \mathbf{R}$
PropertyIsGreaterThanOrEqualTo	(P_1, V)	Returns true iff $P_1 \geq V$ where P_1 and $V \in \mathbf{R}$
PropertyIsBetween	(P_1, V_1, V_2)	Returns true iff $V_1 < P_1 < V_2$ where P_1, V_1 and $V_2 \in \mathbf{R}$

Table B.5: SWSRL Arithmetic Operators

Built-in	Arguments	Semantics
Add	(V_1, V_2, R)	Add V_1 to V_2 and bind the result to R
Divide	(V_1, V_2, R)	Divide V_1 by V_2 and bind the result to R
Subtract	(V_1, V_2, R)	Subtract V_1 by V_2 and bind the result to R
Equals	(V_1, V_2, R)	Test the mathematical equality of V_1 to V_2 return true/false
Modulo	(V_1, V_2, R)	Compute V_1 modulo V_2 (the reminder of their division) and bind the result to R
Multiply	(V_1, V_2, R)	Multiply V_1 to V_2 and bind the result to R

B.9 SPATIAL OPERATORS

Table B.6: SWSRL Spatial Operators

Built-in	Arguments	Semantics
Area	(Ind_1, R)	binds to R the geodetic area in m^2 of the polygon feature Ind_1
Distance	(Ind_1, Ind_2, R)	binds to R the geodetic distance in m between the centroid of the Ind_1 and Ind_2
Perimeter	(Ind_1, R)	binds to R the geodetic distance in m the perimeter of the polygon feature Ind_1

Table B.7: SWSRL External Topological Operators

Backward Builtin	Arguments	Description
exAdjacent	(Ind_1, Ind_2)	Returns true if Ind_1 and Ind_2 are adjacent
exTouches	(Ind_1, Ind_2)	Returns true if Ind_1 and Ind_2 are touching
exDisjoint	(Ind_1, Ind_2)	Returns true if Ind_1 and Ind_2 are disjoint
exCrosses	(Ind_1, Ind_2)	Returns true Ind_1 and Ind_2 cross
exIntersects	(Ind_1, Ind_2)	Returns true if Ind_1 and Ind_2 intersect
exEqual	(Ind_1, Ind_2)	Returns true if Ind_1 and Ind_2 are equal
exContains	(Ind_1, Ind_2)	Returns true if Ind_1 contains Ind_2
exOverlaps	(Ind_1, Ind_2)	Returns true if Ind_1 and Ind_2 overlap
exWithin	(Ind_1, Ind_2)	Returns true if Ind_1 is within Ind_2

Table B.8: SWSRL External Relative Size Operators

Backward Builtin	Arguments	Semantics
exSameSizeAs	(Ind_1, Ind_2)	Returns true iff the area of the polygonal geometry of Ind_1 in $\mathbb{R}^2$ is identical to the area of the polygonal geometry of Ind_2 in $\mathbb{R}^2$ ($\text{area}(Ind_1) \equiv \text{area}(Ind_2)$)
exLargerThan	(Ind_1, Ind_2)	Returns true iff the area of the polygonal geometry of Ind_1 in $\mathbb{R}^2$ is larger than the area of the polygonal geometry of Ind_2 in $\mathbb{R}^2$ ($\text{area}(Ind_1) > \text{area}(Ind_2)$)
exSmallerThan	(Ind_1, Ind_2)	Returns true iff the area of the polygonal geometry of Ind_1 in $\mathbb{R}^2$ is smaller than the area of the polygonal geometry of Ind_2 in $\mathbb{R}^2$ ($\text{area}(Ind_1) < \text{area}(Ind_2)$)

Table B.9: SWSRL External Cardinal Direction Operators

Backward Builtin	Arguments	Description
exNorthOf	(Ind_1, Ind_2)	Returns true if Ind_1 is to the North of Ind_2
exSouthOf	(Ind_1, Ind_2)	Returns true if Ind_1 is to the South of Ind_2
exWestOf	(Ind_1, Ind_2)	Returns true if Ind_1 is to the West of Ind_2
exEastOf	(Ind_1, Ind_2)	Returns true if Ind_1 is to the East of Ind_2

APPENDIX C

APPENDIX C - ALGORITHMS

C.1 VILAIN AND KAUTZ'S BASELINE PATH CONSISTENCY ALGORITHM

The baseline path-consistency algorithm is a practical realisation, in Java, of the standard Vilain and Kautzs path-consistency [265] and Mackworth's revise [169] algorithm, but tailored to work over the RCC-12 relations. Both algorithms are shown in this section.

Key differences between the implemented algorithm, denoted VKPC4J, and the original algorithm are now discussed. VKPC4J works over RCC-12 relations and uses the RCC-12 composition table. Construction of the set of edges $\mathcal{Q}$ (procedure `generateInitialSet`) takes $O(n^2)$ where n is the number of regions in $SWSRLO$. As with the original algorithm, the main body of the Path-Consistency algorithm (procedure `PathConsistency`) is dependent on the size of $\mathcal{Q}$ ($|\mathcal{Q}|$) times the number of possible compositions with a third region k. In other words the number of possible triples of regions i,j and k.

The Revise algorithm is based on Allen's Revise algorithm [3]. Certain improvements to the original Revise algorithm where shown in [16], however we only consider the original implementation, as it more closely mimics how each composition is performed using $SWSRL$. That is, we compose every relation between regions i and j (R_{ij}) with every relation between i and k (R_{jk}), which has the complexity $|R_{ij}| \times |R_{jk}|$. The HashMap composition table has a O(1) constant time complexity, and so does not add to the overall time complexity of VKPC4J.

Algorithm C.1 PathConsistency()

```
Let n[] = 2D array of relations between regions in SWSRLO
Q = generateInitialSetQ(n.length)
boolean consistent = true
int errorCount =0
while Q.size() > 0 do
  edge< i, j > e = Q.get(0)
  Q.remove(0)
  for k = 1 to k = n.length − 1 do
    if k! = e < i > && k! = e < j > then
      boolean ikConsistent, ikRevised = REVISE(e< i >,e< j >,k)
      boolean kjConsistent, kjRevised = REVISE(k,e< i >,e< j >)
      if ikRevised = true then
        if ikConsistent = false then
          consistent = false
          errorCount = errorCount +1
        else
          if Q.contains(e< i >, k) then
            Q.add(new edge(e< i >,k))
          end if
        end if

      end if
      if kjRevised =true then
        if kjConsistent = false then
          consistent = false
          errorCount = errorCount +1
        else
          if Q.contains(k,e< j >) then
            Q.add(new edge(k,e< j >))
          end if
        end if
      end if
    end if
  end for
end while
return consistent, errorCount
```

Algorithm C.2 generateInitialSetQ(int length)

```
Let Q = new List of edge< i, j >
for i = 1 to i < length do
   for j = 1 to j < length do
      if i < j then
         Q.add(new edge(i,j))
      end if
   end for
end for
return Q
```

Algorithm C.3 REVISE(i,k,j)

```
String ik[] = convertRCC12toRCC8(n[i],n[j])
String kj[] = convertRCC12toRCC8(n[j],n[k])
String compositions[] = getCompositions(ik[], kj[])
boolean revised = addTo(i,k,compositions[])
boolean consistent = checkConsistentRCC12(extractRCC12toRCC8(n[i],n[k])
converse = getConverse(compositions)
addTo(k,i,converse[])
return consistent, revised
```

Algorithm C.4 addTo(i,k,compositions[])

```
boolean revised = false
String ik[] = convertRCC12toRCC8(n[i],n[k])
String[] existingAndNewRelations
for all newRelation in compositions do
   boolean isExisting = false
   for all existingRelation in ik do
      if newRelation = existingRelation then
         isExisting = true
      end if
   end for
   if isExisting = false then
      n[i][k] = n[i][k] + newRelation
   end if
end for
return revised
```

Algorithm C.5 getCompositions(ik[], kj[])

```
Let HashCompositions = HashMap of RCC-12 compositions i.e.Key = "p-1:p-" value = "p-1"
String[] compositions
for i = 0 to i = ik.length − 1 do
  for j = 0 to j = kj.length − 1 do
    compositions.add(HashCompositions.get(ik[i]+":"+kj[j])
  end for
end for
return compositions
```

Algorithm C.6 getConverse(compositions[])

```
Let HashConverse = HashMap of RCC-12 converse relations i.e.Key = "p-1" value = "p"
String[] converse
for i = 0 to i = compositions.length − 1 do
  compositions.add(HashCompositions.get(compositions[i])
end for
return converse
```

Algorithm C.7 checkConsistentRCC12(existing[])

```
if existing.length = 0 then
  return true
end if
TreeSet allRelations
boolean first = true
for all relations r in existing do
  RCC8 = convertRCC12toRCC8(r)
  if first = true then
    allRelations.addAll(RCC8)
    first = false
  else
    HashSet current
    current.addAll(RCC8)
    allRelations.retainAll(current)
  end if
end for
boolean consistent = false
if allRelations.length > 0 then
  consistent = true
end if
return consistent
```

Importantly, the algorithm can be run in both deduction only mode, or deduction and integrity mode. Deduction only mode solely adds RCC-12 relations to each pair of regions. Deduction and integrity mode additionally uses procedure `checkConsistentRCC12`, which mimics operation C.1, to check the consistency of the relations between pairs of regions.

$$A_r C \;=\; A_r C \cap (A_r B \otimes B_r C) \qquad \text{(C.1)}$$

`checkConsistentRCC12` creates a new set from the intersection of RCC-8 disjunctive relations (converted from the conjunctive set of RCC-12 inputted to the procedure) using a Java TreeSet. The addAll method has a O(log(n)) time complexity. The retainAll is an intersection operation over the set, which intersects the current contents of the TreeSet, with the RCC-8 relations corresponding to each RCC-12 relation r in `existing`. retainAll has a complexity of O(max(m,n)) where m is the size of the current TreeSet and n is the size of the given set. Hence this method does not add significantly to the overall time complexity of the algorithm.

The procedure `convertRCC8toRCC12`, which is not shown, converts a set of RCC12 relations to RCC8 relations in linear time.

The procedure `addTo` takes a time of O($e \times n$). Where e is the number of existing relations between regions i and k, and n is the number of new relations inputted to the method (in the variable `compositions[]`).

In overview, VKPC4J has a similar overall space complexity of O(n^2) (based on the initial size of $\mathcal{Q}$) as the original Vilain and Kautzs path-consistency and Allens Revise algorithms. The time complexity is fundamentally similar to the original O(n^3), but will in reality be slightly worse due to the overhead of running the concrete realisation of operations `addTo` and `checkConsistencyRCC12`, and the new procedure `convertRCC8toRCC12`, in Java.

C.2 Intersection Builtin validTR

The `validTR` predicate represents a call to a builtin (procedural attachment) that, in overview, checks if an RCC-12 relation can be added between two regions without causing a topological inconsistency.

validTR: the builtin `validTR` returns true if, when intersecting (refining) the set of generalised RCC-12 relations between the two input regions with the input generalised relations, the result is not the empt set - hence an inconsistency is not detected. The builtin accepts three arguments, two Geofeatures in *SWSRLO* and one generalised relationship. In essence, `validTR`, is a replacement for the core revision method of Allen's REVISE function for performing path-consistency:$R_{ac} = R_{ac} \cap (R_{ab} \otimes R_{bc})$. However the procedure is clean semantically, and does not replace the relation R_{ac} with a refined relation $R_{ac}^{'}$ (as with REVISE), instead only reporting whether it is inconsistent or not.

Algorithm C.8 validTR

1: **Input:** Geofeature A, Geofeature B and a single Generalised Relation Gr
2: **Output:** True if the relations between A and B are consistent, false otherwise
3: Let set Gr_{rcc8} = convertGeneralisedToRCC8(Gr)
4: Let set AB = conjunction of RCC-12 relations between geofeatures A and B
5: Let set AB_{rcc8} = convertedGeneralisedToRCC8(AB)
6: Let boolean hasIntersection = false
7: **for** (i=0 ; i < sizeOf(AB_{rcc8}) ;i=i+1) **do**
8:   **for** (j=0 ; j < sizeOf(Gr_{rcc8}) ;j=i+1) **do**
9:     **if** (Gr_{rcc8}[i] equals AB_{rcc8}[i]) **then**
10:       Let hasIntersection ⟵ true
11:     **end if**
12:   **end for**
13: **end for**
14: **Return** hasIntersection

convertGeneralisedToRCC8() is a support procedure that converts a conjunctive set of RCC-12 relations into a disjunctive set of RCC-12 relations, described in C.3.

C.3 VALIDTR SUPPORT PROCEDURE

Where RCC12toRCC8 is a table lookup function that maps RCC-12 relations to a corresponding set of RCC-8 relations.

Algorithm C.9 convertGeneralisedToRCC8()

```
Input: conjunctive set of generalised relations Gr (can contain on one element)
Output: A set of RCC-8 relations that corresponds to the conjunction set of
  RCC-12 relations Gr
Let set Revised ⟵ θ
Let Array of sets RCC8 ⟵ θ
for each element E of Gr do
  Let set E_rcc8 ⟵ RCC12toRCC8(E)
  Add to RCC8 the set E_rcc8
end for
for (i=0 ; i < sizeOf(RCC8) ;i=i+1 ) do
  Let R_rcc8 ⟵ RCC8[i]
  for each element E of R_rcc8 do
    Let allIntersects ⟵ true
    for (j=0 ; j < sizeOf(RCC8) ;j=j+1 ) do
      if (i ≠ j) then
        Let R2_rcc8 ⟵ RCC8[j]
        for each element E2 of R2_rcc8 do
          if (E ≠ E2) then
            Let allIntersects ⟵ false
          end if
        end for
      end if
    end for
    if (allIntersects = true) then
      Add to set Revise element E
    end if
  end for
end for
Return Revise
```

C.4 Spatial Genetic Algorithms

C.4.1 Complete Algorithm

Algorithm C.10 Spatial Genetic Algorithm

Generate initial population *Pop* of candidate solutions *cs*
for all $cs \in Pop$ **do**
  Calculate fitness of *cs*
end for
for (i =0; i < $\frac{sizeof(Pop)}{2}$-2;i=i+2) **do**
  choose chromosome c_1 using roulette-wheel selection
  choose chromosome c_2 using roulette-wheel selection
  offspring o_1 = cross c_1 with c_2 using random single point crossover
  offspring o_2 = cross c_2 with c_1 using random single point crossover
  copy o_1 and o_2 into the new population Pop_{new}
end for
choose two best chromosomes bc_1 and bc_2 from *Pop* and add to Pop_{new}
for all $cs \in Pop_{new}$ **do**
  mutate *cs* at a rate of 2%
end for

C.4.2 Mutation Algorithm

Algorithm C.11 Mutation

for all $cs \in Pop_{new}$ **do**
  let solution[][] represent a candidate solution table for *cs* {encoded in an array, where i is a row index, j is a column index in solution[i][j] and array indexes start a 1}
  for ($i = 1$; $i <$ solution.length; $i++$) **do**
    for ($j = 1$; $j < i - 1$; $j++$) **do**
      let randNum = a random number in [0,1]
      if (randNum ≤ 0.02) **then**
        let RCC_{new} = a quasi-random RCC-8 relation chosen using weightings in δ
        set solution[i][j] = RCC_{new}
      end if
    end for
  end for
end for

C.4.3 FITNESS FUNCTION ALGORITHM

Algorithm C.12 Fitness Function

```
let errorCount = 0
for i = 0; i < |n|; i++ do
    for j = 0; j < |n|; j++ do
        for k = 0; k < |n|; k++ do
            R_ij = R_ij ∩ (R_ik ⊗ R_kj)
            if R_ij = θ then
                errorCount++
            end if
        end for
    end for
end for
```

C.5 RELATIONAL CONFIDENCE ALGORITHM

Algorithm C.13 Relational Confidence

```
Create Array fromToRecords[]
Create Array groupedFromTo[]
for (w =0; w < sizeof(groupedFromTo);w++) do
  Set fromToGroup[] = groupedFromTo[w]
  Let fromToRecordsRCC8[] = new Array
  Let count = 0
  for (j =0; j < sizeof(fromToGroup);j++) do
    Let RCC8[] = convertGeneralisedToRCC8(fromToGroup[j])
    Set fromToRecordsRCC8[count].from = fromToGroup[i].from
    Set fromToRecordsRCC8[count].to = fromToGroup[i].to
    Set fromToRecordsRCC8[count].via = fromToGroup[i].via
    Set fromToRecordsRCC8[count].RCC8Relations = RCC8[]
    count++
  end for
  for (i =0; i < sizeof(fromToRecordsRCC8);i++) do
    for (j =0; j < sizeof(fromToRecordsRCC8);j++) do
      if (i ≠ j) then
        Let RCC8A[] = fromtToRecordsRCC8[i].RCC8Relations
        Let RCC8B[] = formToRecordsRCC8[j].RCC8Relations
        for (s =0; s < sizeof(RCC8A);s++) do
          for (p =0; p < sizeof(RCC8B);p++) do
            if (RCC8A[s].getRelation == RCC8B[p].getRelation) then
              Set RCC8A[s].matching ++
            end if
          end for
        end for
        Set fromToRecordsRCC8[i].RCC8Relations = RCC8A
      end if
    end for
  end for
  RankRelations(fromToRecordsRCC8)
  ConvertRelationCountToPercentage(fromToRecordsRCC8,
  sizeOf(fromToGroup))
  writeFromToRelationalConfidence(fromToRecordsRCC8)
end for
```

References

[1] Alia I. Abdelmoty, Philip D. Smart, Christopher B. Jones, Gaihua. Fu, and David. Finch. *A critical evaluation of ontology languages for geographic information retrieval on the Internet.*, volume 16 of *Journal of Visual Languages & Computing*, pages 331–358. 2005.

[2] L. Albert and F. Fages. Average case complexity analysis of the RETE multi-pattern match algorithm. In Timo Lepistö and Arto Salomaa, editors, *Proceedings of the 15th International Colloquium on Automata, Languages and Programming, ICALP'88 (Tampere, Finland, July 11-15, 1988)*, volume 317 of *LNCS*, pages 18–37. Springer-Verlag, Berlin-Heidelberg-New York-London-Paris-Tokyo-Hong Kong, 1988.

[3] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832–843, 1981.

[4] Grigoris Antoniou and Harold Boley, editors. *Rules and Rule Markup Languages for the Semantic Web: Third International Workshop, RULEML 2004, Hiroshima, Japan, November 8, 2004, Proceedings*, volume 3323 of *Lecture Notes in Computer Science*, New York, NY, USA, 2004. Springer-Verlag Inc.

[5] Grigoris Antoniou, Carlos Viegas Damásio, Benjamin Grosof, Ian Horrocks, Michael Kifer, Jan Maluszyński, and Peter F. Patel-Schneider. Combining Rules and Ontologies, a survey. Deliverable, 2005.

[6] Chandrabose Aravindan, Jürgen Dix, and Ilkka Niemelä. DisLoP: Towards a Disjunctive Logic Programming System. In J. Dix, U. Furbach, and A. Nerode, editors, *Logic Programming and Non-Monotonic Reasoning, Proceedings of the Fourth International Conference*, LNAI 1265, pages 342–353, Berlin, June 1997. Springer.

[7] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings*

of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005, pages 364-369. Professional Book Center, 2005.

[8] Franz Baader and Bernhard Hollunder. Priorities on defaults with prerequisites, and their application in treating specificity in terminological default logic. In *Journal of Automated Reasoning*, volume Volume 15, pages 41–68. Springer Netherlands, February 1995.

[9] Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics as ontology languages for the semantic web. In *Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence*, volume 2605/2005, pages 228–248. Springer-Verlag, 2003.

[10] L. Bachmair and H. Ganzinger. Resolution theorem proving. In A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier Science, 2001.

[11] Kristof Van Belleghem, Marc Denecker, and Danny De Schreye. A strong correspondence between description logics and open logic programming. In *International Conference on Logic Programming*, pages 346–360, 1997.

[12] Brandon Bennett. Spatial reasoning with propositional logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR94)*, pages 51–62. Morgan Kaufmann, 1994.

[13] Brandon Bennett, Hans Guesgen, Frank Anger, Gerard Ligozat, and Rita Rodriguez. Determining consistency of topological relations. *Constraints*, 3:213–225, 1998.

[14] Tim Berners-Lee, Mark Fischetti, and Michael Dertouzos. *Weaving the Web*. HarperCollins, San Francisco, California, 1999.

[15] Stefano Berretti and Alberto Del Bimbo. Modeling spatial relationships between 3d objects. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 119–122, Washington, DC, USA, 2006. IEEE Computer Society.

[16] Christian Bessiere. A simple way to improve path consistency processing in interval algebra networks. In *AAAI/IAAI, Vol. 1*, pages 375–380, 1996.

[17] Abhiji Bhole, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenić. Extracting named entities and relating them over time based on wikipedia. volume 22, pages 20–26, 2007.

[18] Harold Boley, Jos de Bruijn, Dieter Fensel, Pascal Hitzler, Michael Kifer, Reto Krummenacher, Holger Lausen, Axel Polleres, and Rudi Studer. Web rule language (wrl). W3C Member Submission, 09 September, 2005.

[19] A. Borgida. On the relative expressiveness of description logics and predicate logics. *Artificial Intelligence*, 82(1–2):353–367, 1996.

[20] R. J. Brachman and H. J. Levesque. The tractability of subsumption in frame-based description languages. In R. J. Brachman, editor, *Proceedings of the National Conference on Artificial Intelligence*, pages 34–37, Austin, Texas, August 1984. William Kaufmann.

[21] Ronald J. Brachman, Alexander Borgida, Deborah L. Mcguinness, Peter F. Patel-schneider, and Lori Alperin Resnick. The classic knowledge representation system, or, kl-one: The next generation. In *Preprints of the Workshop on Formal Aspects of Semantic Networks, Two Harbors*, pages 1036–1043. MorganKaufman, 1989.

[22] Dan Brickley and Ramanathan V. Guha. RDF vocabulary description language 1.0: RDF schema. World Wide Web Consortium, Recommendation http://www.w3.org/TR/2004/REC-rdf-schema-20040210, February 2004.

[23] Davide Buscaldi, Paolo Rosso, and Piedachu Peris. Inferring geographical ontologies from multiple resources for geographical information retrieval. In Ross Purves and Chris Jones, editors, *Proceedings of the 3rd ACM Workshop On Geographic Information Retrieval, Seattle, WA*. Department of Geography, University of Zurich, August 10 2006.

[24] Orkut Buyukkokten, Junghoo Cho, Hector Garcia-Molina, Luis Gravano, and Narayanan Shivakumar. Exploiting geographical location information of web pages. In *In Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*, pages 91–96, 1999.

[25] Marco Cadoli, Luigi Palopoli, and Maurizio Lenzerini. Datalog and description logics: Expressive power. In *Database Programming Languages*, volume 1369/1998, pages 281–298. Springer Berlin / Heidelberg, 1998.

[26] Diego Calvanese, Giuseppe Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The DL-lite family. *Journal Automatic Reasoning*, 39(3):385–429, 2007.

[27] Luis F. Castro, Vitor S. Costa, and Ricardo Lopes. On the cache performance of Prolog systems. In *Colloquium on Implementation of Constraint and LOgic*

Programming Systems, volume 344 of *CW Reports*, pages 74–85, Leuven, Belgium, July 31 2002.

[28] C.E. Shannon. A symbolic analysis of relay and switching circuits. *Trans. AIEE*, 67:713–723, 1938.

[29] Harry Chen, Stephane Fellah, and Yaser A. Bishr. Rules for geospatial semantic web applications. In *Rule Languages for Interoperability, Washington, D.C., USA*. W3C, 2005.

[30] Weidong Chen and David S. Warren. Tabled evaluation with delaying for general logic programs. *Journal of the ACM*, 43(1):20–74, January 1996.

[31] Eugene Inseok Chong, Souripriya Das, George Eadon, and Jagannathan Srinivasan. An efficient SQL-based RDF querying scheme. In Klemens Böhm, Christian S. Jensen, Laura M. Haas, Martin L. Kersten, Per-Åke Larson, and Beng Chin Ooi, editors, *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*, pages 1216–1227. ACM, 2005.

[32] Keith L. Clark. Negation as failure. In H. Gallaire and Jack Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[33] Bowman L. Clarke. A calculus of individuals based on connection. *Notre Dame Journal of Formal Logic*, 22(3):204–218, 1981.

[34] Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *SSD '93: Proceedings of the Third International Symposium on Advances in Spatial Databases*, pages 277–295, London, UK, 1993. Springer-Verlag.

[35] Sophie Cockcroft. A taxonomy of spatial data integrity constraints. *Geoinformatica*, 1(4):327–343, 1997.

[36] Sophie Cockcroft. User defined spatial business rules: storage, management and implementation a pipe network case study. In *Presented at the 10th Colloquium of the Spatial Information Research Centre,*, pages 73–81, November, 1998.

[37] A G Cohn and S M Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundam. Inf.*, 46(1-2):1–29, 2001.

[38] A. Colmerauer and P. Roussel. The birth of prolog. In *Second History of Programming Languages Conference*, volume 28, pages 37–52, march 1993.

[39] Helen Couclelis. People manipulate objects (but cultivate fields): Beyond the raster-vector debate in gis. In *Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, pages 65–77, London, UK, 1992. Springer-Verlag.

[40] Bart G. W. Craenen, A. E. Eiben, and Jano I. van Hemert. Comparing evolutionary algorithms on binary constraint satisfaction problems. *IEEE Trans. Evolutionary Computation*, 7(5):424–444, 2003.

[41] J. M. Crawford and B. J. Kuipers. Algernon a tractable system for knowledge-representation. *SIGART Bull.*, 2(3):35–44, 1991.

[42] James M. Crawford. *Access-Limited Logic—A Language for Knowledge Representation.* Ph.D. dissertation, Department of Computer Science, University of Texas at Austin, Austin, Texas, 1990. Also Technical Report AI90–141, Artificial Intelligence Laboratory, University of Texas at Austin.

[43] Cuenca, Bijan Parsia, and Evren Sirin. Combining owl ontologies using e-connections. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1):40–59, January 2006.

[44] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. In *CCC '97: Proceedings of the 12th Annual IEEE Conference on Computational Complexity*, page 82, Washington, DC, USA, 1997. IEEE Computer Society.

[45] Government Spatial Data, Amarnath Gupta, Ashraf Memon, Joshua Tran, Rajiv P. Bharadwaja, and Ilya Zaslavsky. Information mediation across heterogeneous. In *National Conference on Digital Government Research (May 20-22, 2002, Los Angeles, CA). Digital Government Research Center*, pages 107–113, 2002.

[46] Jos de Bruijn, Rubén Lara, Axel Polleres, and Dieter Fensel. Owl dl vs. owl flight: conceptual modeling and reasoning for the semantic web. In *Proceedings of the 14th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2005. ACM Press.

[47] Rina Dechter, Itat Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1–3):61–95, 1991.

[48] Marc Denecker and Danny De Schreye. Justification semantics: a unifying framework for the semantics of logic programs. In *Proceedings of the second international*

workshop on Logic programming and non-monotonic reasoning, pages 365–379, Cambridge, MA, USA, 1993. MIT Press.

[49] Ludovic Denoyer and Patrick Gallinari. The wikipedia XML corpus. In Norbert Fuhr, Mounia Lalmas, and Andrew Trotman, editors, *INEX*, volume 4518 of *Lecture Notes in Computer Science*, pages 12–19. Springer, 2006.

[50] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. AL-log: Integrating datalog and description logics. *Intelligent Information Systems*, 10(3):227–252, 1998.

[51] Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Description logics of minimal knowledge and negation as failure. *ACM Transactions Computing*, 3(2):177–225, 2002.

[52] Nick Drummond and Rob Shearer. The open world assumption or sometimes its nice to know what we don't know. Internet: http://www.cs.man.ac.uk/ drummond/presentations/OWA.pdf, 2006.

[53] Matt Duckham, Jenny Lingham, Keith T. Mason, and Michael F. Worboys. Qualitative reasoning about consistency in geographic information. *Inf. Sci.*, 176(6):601–627, 2006.

[54] M. J. Egenhofer and D. Mark. Naive Geography. In Andrew U. Frank and Werner Kuhn, editors, *Spatial Information Theory: a theoretical basis for GIS*, number 988 in Lecture Notes in Computer Science, pages 1–15. Springer, Berlin/Heidelberg/New York, 1995. International Conference COSIT'95.

[55] Max Egenhofer and Robert Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems 5*, pages 161–174, 1991.

[56] Max J. Egenhofer. Toward the semantic geospatial web. In *Proceedings of the tenth ACM international symposium on Advances in geographic information systems*, pages 1–4. ACM Press, 2002.

[57] Max J. Egenhofer, David M. Mark, and John Herring. The 9-intersection: Formalism and its use for natural-language spatial predicates. NCGIA Technical Report :http://citeseer.ist.psu.edu/397820.html; http://www.ncgia.ucsb.edu/Publications/Tech_Reports/94/94-1.PDF, February 1994.

[58] Andrew Eisenberg and Jim Melton. SQL: 1999, formerly known as SQL 3. *SIGMOD Record*, 28(1):131–138, 1999.

[59] M. Eisenstadt and M. Brayshaw. A knowledge engineering toolkit: The first of a two-part series presenting a knowledge-engineering toolkit for building expert systems. *Byte Magazine*, 15(10):268 - 282, October 1990.

[60] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Adding disjunction to datalog. In *Proceedings of the 13th Symposium on Principles of Database Systems*, pages 267–278, New York, NY, USA, May 1994. ACM Press.

[61] Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive datalog. *ACM Transactions Database Systems*, 22(3):364–418, 1997.

[62] Thomas Eiter, Giovambattista Ianni, Axel Polleres, Roman Schindlauer, and Hans Tompits. Reasoning with rules and ontologies. In *Reasoning Web, Second International Summer School 2006, Lisbon, Portugal*, volume 4126 of *LNCS*, pages 93–127. Springer, 2006.

[63] Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits. Combining answer set programming with description logics for the semantic web. In *KR2004*, pages 141–151. AAAI Press, Menlo Park, California, 2004.

[64] B.A. El-Geresy and A.I. Abdelmoty. Towards a general theory for modelling qualitative space. In *International Journal on Artificial Intelligence Tools, IJAIT*, volume 11, pages 347–367. World Scientific Publishing, 2002.

[65] B.A. El-Geresy and A.I. Abdelmoty. SPARQS: a Qualitative Spatial Reasoning Engine. In *Journal of knowledge-based Systems*, volume 17, pages 89–102. Elsevier, 2004.

[66] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems.* Benjamin/Cummings, Redwood City, 2 edition, 1994.

[67] Henrik Eriksson. Using jesstab to integrate protégé and jess. *IEEE Intelligent Systems*, 18(2):43–50, 2003.

[68] Christian Fahrner, Thomas Marx, and Stephan Philippi. Integration of integrity constraints into object-oriented database schema according to ODMG-93. Fachberichte Informatik 9–95, Universität Koblenz-Landau, 1995.

[69] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

[70] Richard Fikes and Tom Kehler. The role of frame-based representation in reasoning. *Communications of the ACM*, 28(9):904–920, September 1985.

[71] L. De Floriani, P. Marzano, and E. Puppo. Spatial queries and data models. In I. Campari A. U. Frank and U. Formentini, editors, *Information Theory: a Theoretical Basis for GIS*, pages 113–138. Springer-Verlag, Lecture Notes in Computer Science, N.716, 1992.

[72] Christiaan Fluit, Marta Sabou, and Frank van Harmelen. Ontology-based information visualization. In *Visualizing the Semantic Web*, pages 36–48. 2002.

[73] Frederico T. Fonseca and Max J. Egenhofer. Ontology-driven geographic information systems. In *ACM-GIS '99, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems, November 2-6, 1999, Kansas City, USA*, pages 14–19, 1999.

[74] K. D Forbus. Qualitative spatial reasoning framework and frontiers. In J. Narayanan and B. Chandrasekaran, editors, *Diagrammatic Reasoning*, pages 183–202. AAAI Press, 1995.

[75] C. L. Forgy. Ops5 reference manual. Technical Report CMU-CS-81-135, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA, 1981.

[76] Charles Forgy. Rete: A fast algorithm for the many patterns/many objects match problem. *Artificial Intelligence*, 19(1):17–37, 1982.

[77] Andrew U. Frank. The use of geographical information systems: The user interface is the system. In Hilary M. Medyckyj-Scott, David Hearnshaw, editor, *Human Factors in Geographical Information Systems*, pages 3–14. Belhaven Press, London and Florida, 1993.

[78] Andrew U. Frank. Qualitative spatial reasoning: Cardinal directions as an example. *International Journal of Geographical Information Science*, 10(3):269–290, 1996.

[79] Andrew U. Frank. Tiers of ontology and consistency constraints in geographic information systems. *International Journal of Geographical Information Science*, (7):667–678, 2001.

[80] A.U. Frank. Qualitative spatial reasoning about distances and directions in geographic space. In *Journal of Visual Languages and Computing*, volume 639, pages 3:343–371. Lecture Notes In Computer Science, Springer, 1992.

[81] Juliana Freire, Terrance Swift, and David S. Warren. A formal framework to model scheduling in tabled evaluations. Technical report, 1997.

[82] Christian Freksa. Using orientation information for qualitative spatial reasoning. *Lecture Notes in Computer Science*, 639:162–178, 1992.

[83] Mark Gahegan. Proximity operators for qualitative spatial reasoning. In *COSIT, Lecture Notes in Computer Science*, volume 988, pages 31–44, 1995.

[84] M. Gelfond. On stratified autoepistemic theories. In *Proceedings of AAAI-87*, pages 207–211, Los Altos, CA., 1987. American Association for Artificial Intelligence, Morgan Kaufmann.

[85] M. Gelfond. Autoepistemic logic and formalization of commonsense reasoning. In M. Reinfrank, J. de Kleer, M. L. Ginsberg, and E. Sandewall, editors, *Non-monotonic reasoning*, pages 176–186. Berlin: Springer-Verlag, 1989. Lecture Notes in Artificial Intelligence, 346.

[86] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Proceedings of the Fifth International Conference on Logic Programming (ICLP)*, pages 1070–1080, 1988.

[87] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.

[88] Michael Gelfond, Halina Przymusinska, and Teodor Przymusinski. On the relationship between cwa, minimal model and minimal herbrand model semantics. *International Journal of Intelligent Systems*, 5:549–564, 1995.

[89] Alfonso Gerevini. Incremental tractable reasoning about qualitative temporal constraints. In Georg Gottlob and Toby Walsh, editors, *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 1283–1288. Morgan Kaufmann, 2003.

[90] Alfonso Gerevini and Jochen Renz. Combining topological and qualitative size constraints for spatial reasoning. In *4th International Conference on Principles and Practice of Constraint Programming (CP98)*, volume 1520 of *Lecture Notes in Computer Science*, pages 220–234, October 1998.

[91] J. Giarratano and G. Riley. *Expert Systems: Principles and Programming.* PWS Publishing Company, Boston, 2nd edition, 1994.

[92] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, December 2005.

[93] Christine Golbreich. Combining rule and ontology reasoners for the semantic web. In *Rules and Rule Markup Languages for the Semantic Web*, volume 3323/2004, pages 6–22. Springer Berlin / Heidelberg, October 2004.

[94] David E. Goldberg and John H. Holland. Genetic algorithms and machine learning. In *Machine Learning*, volume 3, pages 95–99. Springer Netherlands, January 1988.

[95] Michael Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, September 2007.

[96] R. Goyal and M.J. Egenhofer. The direction-relation matrix: A representation for directions relations between extended spatial objects. In *The Annual Assembly and the Summer Retreat of University Consortium for Geographic Information Systems Science*, 1997.

[97] Roop Goyal, Max Egenhofer, In C. Jensen, M. Schneider, B. Seeger, V. Tsotras (eds, Roop K. Goyal, and Max J. Egenhofer. Similarity in cardinal directions. In *In Proceedings of the Seventh International Symposium on Spatial and Temporal Databases*, pages 36–55. Springer-Verlag, 2001.

[98] Bernardo Cuenca Grau and Boris Motik. OWL 1.1 web ontology language: Model-theoretic semantics. World Wide Web Consortium, Working Draft:http://www.w3.org/TR/2008/WD-owl11-semantics-20080108, January 2008.

[99] John J. Grefenstette, Rajeev Gopal, Brian J. Rosmaita, and Dirk Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 160–168, Mahwah, NJ, USA, 1985. Lawrence Erlbaum Associates, Inc.

[100] Stephan Grimm and Boris Motik. Closed World Reasoning in the Semantic Web through Epistemic Operators. In Bernardo Cuenca Grau, Ian Horrocks, Bijan Parsia, and Peter Patel-Schneider, editors, *Proc. of the Workshop on OWL: Experiences and Directions (OWLED 2005)*, Galway, Ireland, November 11–12 2005.

[101] Benjamin N. Grosof. Prioritized conflict handling for logic programs. In *ILPS '97: Proceedings of the 1997 international symposium on Logic programming*, pages 197–211. MIT Press, 1997.

[102] Benjamin N. Grosof. DIPLOMAT: Compiling prioritized default rules into ordinary logic programs, for E-commerce applications. In *AAAI/IAAI*, pages 912–913, June 23 1999.

[103] Benjamin N. Grosof. Representing e-commerce rules via situated courteous logic programs in ruleML. *Electronic Commerce Research and Applications*, 3(1):2–20, 2004.

[104] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: combining logic programs with description logic. In *Proceedings of the twelfth international conference on World Wide Web*, pages 48–57. ACM Press, 2003.

[105] Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: combining logic programs with description logic. In *WWW*, pages 48–57, 2003.

[106] Benjamin N. Grosof, D. Mahesh, and Timothy W. Finin. Sweetjess: Translating damlruleml to jess. *RuleML 2002, Proceedings of the International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, 14 June 2002, Sardinia, Italy (In conjunction with the First International Semantic Web Conference ISWC2002 and hosted by SIG2 of the OntoWeb Network*, 60:23–45, August 08 2002.

[107] W. Grosso, H. Eriksson, R. Fergerson, J. Gennari, S. Tu, and M. Musen. Knowledge modeling at the millennium (the design and evolution of protege-2000). In *Proceedings of the 12 th International Workshop on Knowledge Acquisition, Modeling and Mangement (KAW'99)*, 1999.

[108] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

[109] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5,6):907–928, 1995.

[110] Rolf Grutter and Bettina Bauer-Messmer. Combining OWL with RCC for spatioterminological reasoning on environmental data. In Christine Golbreich, Aditya

Kalyanpur, and Bijan Parsia, editors, *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, June 6-7, 2007*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[111] Rolf Grütter and Bettina Bauer-Messmer. Towards spatial reasoning in the semantic web: A hybrid knowledge representation system architecture. *The European Information Society. Leading the Way with Geo-Information. Part 8. LNGC*, pages 349–364, December 2007.

[112] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, 43(5/6):625–640, 1995.

[113] Volker Haarslev, Carsten Lutz, and Ralf Moller. Foundations of spatioterminological reasoning with description logics. In *Principles of Knowledge Representation and Reasoning*, pages 112–123, 1998.

[114] Volker Haarslev and Ralf Muller. Racer system description. In *Proceedings of the First International Joint Conference on Automated Reasoning*, volume 2083, pages 701–706. Springer-Verlag, 2001.

[115] Jenny Harding. Geo-ontology concepts and issues. *Report of a workshop on Geo-ontology*, pages 16–17, 2003.

[116] Glen Hart and Catherine Dolbear. *What's So Special about Spatial?*, volume 10.1007/978-1-84628-827-2, pages 39–44 39–44 39–44. Springer London, the geospatial web how geobrowsers, social software and the web 2.0 are shaping the network society edition, February 2007.

[117] Patrick Hayes. RDF semantics. World Wide Web Consortium, Recommendation: http://www.w3.org/TR/2004/REC-rdf-mt-20040210, February 2004.

[118] Stijn Heymans, Davy Van Nieuwenborgh, and Dirk Vermeir. Semantic web reasoning with conceptual logic programs. In *RuleML 2004*, pages 113–127, 2004.

[119] Stijn Heymans, Davy Van Nieuwenborgh, and Dirk Vermeir. Nonmonotonic ontological and rule-based reasoning with extended conceptual logic programs. In *ESWC 2005*, pages 392–407, 2005.

[120] Linda L. Hill. Core elements of digital gazetteers: Placenames, categories, and footprints. In *ECDL '00: Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 280–290, London, UK, 2000. Springer-Verlag.

[121] P. Hitzler, P. Haase, M. Krtzsch, Y. Sure, and R. Studer. DLP isn't so bad after all. In *Proceedings of the Workshop OWL - Experiences and Directions. CEUR Workshop Proceedings*, volume 188, November 2005.

[122] Pascal Hitzler, Jürgen Angele, Boris Motik, and Rudi Studer. Bridging the paradigm gap with rules for OWL. In *Rule Languages for Interoperability*, pages 27–31. W3C, 2005.

[123] Pascal Hitzler, York Sure, and Rudi Studer. Description logic programs: A practical choice for the modelling of ontologies. In François Bry, François Fages, Massimo Marchiori, and Hans-Jürgen Ohlbach, editors, *Principles and Practices of Semantic Web Reasoning*, volume 05371 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2006.

[124] John H. Holland. *Adaptation in natural and artificial systems.* MIT Press, Cambridge, MA, USA, 1992.

[125] Ian Horrocks. Introduction to semantic web, daml+oil, description logics, and possible combinations with axioms and rules: http://www.cs.man.ac.uk/ horrocks/slides/dagstuhlp070202.pd, 2009.

[126] Ian Horrocks. OWL Rules, OK? In *W3C Workshop on Rule Languages for Interoperability, Washington, USA*, pages 27–28, April 2005.

[127] Ian Horrocks and Peter F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 of *Lecture Notes in Computer Science*, pages 17–29. Springer, 2003.

[128] Ian Horrocks and Peter F. Patel-Schneider. Three theses of representation in the semantic web. In *Proceedings of the twelfth international conference on World Wide Web*, pages 39–47. ACM Press, 2003.

[129] Ian Horrocks, Peter F. Patel-Schneider, Harold Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. Internet Report, May 2004. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.

[130] Ian Horrocks and Ulrike Sattler. Decidabiltiy of $\mathcal{SHIQ}$ with complex role inclusion axioms. *Artificial Intelligence*, 160(1–2):79–104, 2004.

[131] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Reasoning with individuals for the description logic shiq. In *Proceedings of the 17th International Conference on Automated Deduction*, pages 482–496. Springer-Verlag, 2000.

[132] Ian Horrocks, Ulrike Sattler, and Frank Wolter, editors. *Proceedings of the 2005 International Workshop on Description Logics (DL2005), Edinburgh, Scotland, UK, July 26-28, 2005*, volume 147 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.

[133] Ian R. Horrocks. Using an expressive description logic: FaCT or fiction? In Anthony G. Cohn, Lenhart Schubert, and Stuart C. Shapiro, editors, *KR'98: Principles of Knowledge Representation and Reasoning*, pages 636–645. Morgan Kaufmann, San Francisco, California, 1998.

[134] G.J. Hunter. Management issues in gis: Accuracy and data quality. In *Conference on managing Geographic Information Systems for success*, pages 95–101, 1996.

[135] Ullrich Hustadt, Boris Motik, and Ulrike Sattler. Data complexity of reasoning in very expressive description logics. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pages 466–471. Professional Book Center, 2005.

[136] Guoqian Jiang, Harold R. Solbrig, James D. Buntrock, and Christopher G. Chute. OWLFCAView Tab: A visualization and modeling tool for composite expressions of snomed ct using formal concept analysis (fca). In *In: Proceedings of the ninth international Protégé conference*, July 2006.

[137] Florence J.III and Max. J. Egenhofer. Distribution of topological relations in geographic datasets. In *Proceedings of 1996 ASPRS/ACSM Annual Convention and Exposition, Baltimore*, pages 314–324, 1996.

[138] B. Johnson and Ben Shneiderman. Tree maps: A space-filling approach to the visualization of hierarchical information structures. In *IEEE Visualization, San Diego, California, U.S.A.*, pages 284–291, 1991.

[139] Christopher B. Jones, Alia I. Abdelmoty, David Finch, Gaihua Fu, and Subodh Vaid. The SPIRIT spatial search engine: Architecture, ontologies and spatial indexing. In Max J. Egenhofer, Christian Freksa, and Harvey J. Miller, editors, *Geographic Information Science, Third International Conference, GIScience 2004*,

Adelphi, MD, USA, October 20-23, 2004, Proceedings, volume 3234 of *Lecture Notes in Computer Science*, pages 125–139. Springer, 2004.

[140] Christopher B. Jones, Alia I. Abdelmoty, and Gaihua Fu. Maintaining ontologies for geographical information retrieval on the web. volume 2888, pages 934–951. LNCS, Springer Berlin / Heidelberg, 2003.

[141] Christopher B. Jones, Harith Alani, and Douglas Tudhope. Geographical information retrieval with ontologies of place. In *COSIT 2001: Proceedings of the International Conference on Spatial Information Theory*, pages 322–335, London, UK, 2001. Springer-Verlag.

[142] Christopher B. Jones, Harith Alani, and Douglas Tudhope. Geographical information retrieval with ontologies of place. In Daniel R. Montello, editor, *COSIT*, volume 2205 of *Lecture Notes in Computer Science*, pages 322–335. Springer, 2001.

[143] Christopher B. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M. van Kreveld, and R. Weibel. Spatial information retrieval and geographical ontologies an overview of the spirit project. In *SIGIR '02*, pages 387–388, New York, NY, USA, 2002. ACM.

[144] Peter Jonsson and Thomas Drakengren. A complete classification of tractability in rcc-5. *Journal of Artificial Intelligence Research*, 6:211–221, 1997.

[145] Vipul Kashyap and Amit Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In Michael P. Papazoglou and Gunter Schlageter, editors, *Cooperative Information Systems*, pages 139–178. Academic Press, San Diego, 1998.

[146] Yarden Katz and Bernardo Cuenca Grau. Representing qualitative spatial information in OWL-DL. In Bernardo Cuenca Grau, Ian Horrocks, Bijan Parsia, and Peter F. Patel-Schneider, editors, *OWLED, OWLWorkshop*, volume 188, 2005.

[147] Yarden Katz and Bijan Parsia. Towards a nonmonotonic extension to owl. In Bernardo Cuenca Grau, Ian Horrocks, Bijan Parsia, and Peter F. Patel-Schneider, editors, *OWLED*, volume 188 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.

[148] Jun'ichi Kazama and Kentaro Torisawa. Exploiting wikipedia as external knowledge for named entity recognition. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707, 2007.

[149] Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, July 1995.

[150] D. Kinzler. Wikisense - mining the wiki. In *Proceedings of Wikimania 05*, 2005.

[151] Matthias Knorr, Jose Alferes, and Pascal Hitzler. A well-founded semantics for hybrid mknf knowledge bases. In Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris, editors, *Proceedings of the 2007 International Workshop on Description Logics (DL-2007), Brixen-Bressanone, Italy, June 2007*, volume Vol-250 of *CEUR Workshop Proceedings*, JUN 2007.

[152] Ravi Kanth V Kothuri, Siva Ravada, and Daniel Abugov. Quadtree and r-tree indexes in oracle spatial: a comparison using gis data. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 546–557, New York, NY, USA, 2002. ACM.

[153] John R. Koza. *Genetic programming II: automatic discovery of reusable programs.* MIT Press, Cambridge, MA, USA, 1994.

[154] Markus Krötzsch, Pascal Hitzler, Denny Vrandecic, and Michael Sintek. How to reason with OWL in a logic programming system. In Thomas Eiter, Enrico Franconi, Ralph Hodgson, and Susie Stephens, editors, *RuleML*, pages 17–28. IEEE Computer Society, 2006.

[155] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Description logic rules. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI-08)*, pages 80–84.

[156] Vipin Kumar. Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, 13:32–44, 1992.

[157] Gabriel M. Kuper and Wenfei Fan. A unified constraint model for XML. In *Computer Networks (Amsterdam, Netherlands: 1999)*, volume 39, pages 489–505, March 23 2001.

[158] Peter Ladkin and Alexander Reinefeld. Effective solution of qualitative interval constraint problems. *Artifical Intelligence*, 57(1):105–124, 1992.

[159] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI*, pages 401–408, 1995.

[160] Ray R. Larson. Geographic information retrieval and spatial browsing. *GIS and Libraries: Patrons, Maps and Spatial Information*, pages 81–124, April 1996.

[161] O. Lassila and R. Swick. Resource description framework (RDF) model and syntax specification: http://www.w3.org/ tr/wd-rdf-syntax.

[162] Jochen L. Leidner. Toponym resolution: A first large-scale comparative evaluation. Research Report EDI-INF-RR-0839, 2006.

[163] Jochen L. Leidner. *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names.* PhD thesis, School of Informatics, Edinburgh, Scotland, UK, June 2007.

[164] Alon Y. Levy and Marie-Christine Rousset. Combining horn rules and description logics in carin. *Artificial Intelligence*, 104(1-2):165–209, 1998.

[165] Sanjiang Li. On topological consistency and realization. *Constraints*, 11(1):31–51, 2006.

[166] J. W. Lloyd and R. W. Topor. Making prolog more expressive. *Journal of Logic Programming*, 1(3):225–240, 1984.

[167] P.A. Longley. *Geographic Information Systems: Principles, Techniques, Applications and Management.* Number 0470244216. John Wiley and Sons, January 1999.

[168] Jildou Louwsma, Sisi Zlatanova, Ron Lammeren, and Peter Oosterom. Specifying and implementing constraints in gis–with examples from a geo-virtual reality system. *Geoinformatica*, 10(4):531–550, 2006.

[169] A. Mackworth. Consistency in networks of constraints. *Artificial Intelligence*, 8(1):99–118, 1977.

[170] Alan K. Mackworth and Eugene C. Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artifical Intelligence*, 25(1):65–74, 1985.

[171] D.F. Marble. The extended data dictionary: A critical element in building viable spatial databases. In *11th annual ESRI user conference*, 1990.

[172] Victor W. Marek and Miroslaw Truszczynski. Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-year*

Perspective, pages 375–398. Springer-Verlag, September 18 1998. Comment: 21 pages.

[173] D. M. Mark, B. Smith, M. Egenhofer, and S. C. Stephen Hirtle. Ontological foundations for geographic information science. In McMaster and L. Usery, editors, *Research Challenges in Geographic Information Science*, pages 335–350. Boca Raton, FL: CRC Press., 2004.

[174] Bruno Martins, Mário J. Silva, Sérgio Freitas, and Ana Paula Afonso. Handling locations in search engine queries. In Ross Purves and Chris Jones, editors, *Proceedings of the 3rd ACM Workshop On Geographic Information Retrieval, GIR 2006, Seattle, WA, USA, August 10, 2006*. Department of Geography, University of Zurich, 2006.

[175] Brian McBride. Jena: A semantic web toolkit. Internet, November 2002. http://dsonline.computer.org/0211/f/wp6jena.htm.

[176] D. R. McCarthy and U. Dayal. The architecture of an active data base management system. In *19 ACM SIGMOD Conf. on the Management of Data, (Portland OR), SIGMOD RECORD*, volume 18, June 1989.

[177] J. McCarthy. Circumscription—A form of non-monotonic reasoning. *Artificial Intelligence*, 13(1 & 2):27–39, 1980.

[178] M. McCloskey. Naive theories of motion. In D. Gentner and A. Stevens, editors, *Mental Models*, pages 299–324. Erlbaum, Hillsdale, NJ, 1983.

[179] Jérôme Mengin. Prioritized conflict resolution for default reasoning. In *ECAI*, pages 376–380, 1994.

[180] Jack Minker. On indefinite databases and the closed world assumption. In *Proceedings of the Sixth Conference on Automated Deduction*, volume 138, pages 292–308, Berlin, 1988. Springer-Verlag.

[181] Jack Minker. Overview of disjunctive logic programming. *Annals of Mathematics and Artificial Intelligence*, 12(1-2):1–24, 1994.

[182] Marvin Minsky. A framework for representing knowledge. In John Haugeland, editor, *Mind Design*, pages 95–128. The MIT Press, Cambridge, MA, 1981.

[183] U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing. *Information Sciences*, 7:95–132, 1974.

[184] Boris Motik, Ian Horrocks, Riccardo Rosati, and Ulrike Sattler. Can OWL and logic programming live together happily ever after? In *International Semantic Web Conference*, volume 4273, pages 501–514. Springer, 2006.

[185] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. *Web Sematics*, 3(1):41–60, 2005.

[186] Boris Motik and Raphael Volz. Optimizing query answering in description logics using disjunctive deductive databases. In *10th International Workshop on Knowledge Representation meets Databases*, volume 79, 2003.

[187] Boris Motik, Denny Vrandecic, Pascal Hitzler, York Sure, and Rudi Studer. dlp-convert - converting OWL DLP statements to logic programs. system available at http://owltools.ontoware.org/. In *European Semantic Web Conference 2005 Demos and Posters*, 2005.

[188] Daniele Nardi and Ronald J. Brachman. An introduction to description logics. In Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation and Applications*, pages 1–40. Cambridge University Press, 2003.

[189] B. Nebel. Computational properties of qualitative spatial reasoning: first results. In *Proceedings of the 19th German Conference on Artificial Intelligence*, pages 233–244, 1994.

[190] Allen Newell. The knowledge level. *Artificial Intelligence*, 18(1):82–127, 1982.

[191] Ilkka Niemela, Patrik Simons, and Tommi Syrjanen. Smodels: A system for answer set programming. *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning, April 9-11*, 2000.

[192] Donald A. Norman. Cognition in the head and in the world: An introduction to the special issue on situated action. *Cognitive Science*, 17(1):1–6, January-March 1993.

[193] Martin O'connor, Holger Knublauch, Samson Tu, and Mark Musen. Writing rules for the semantic web using swrl and jess. In *8th International Protege Conference, Protege with Rules Workshop*, 2005.

[194] Damian O'Dea, Sean Geoghegan, and Chris Ekins. Dealing with geospatial information in the semantic web. In *AOW '05: Proceedings of the 2005 Australasian*

Ontology Workshop, pages 69–73, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[195] Open Geospatial Consortium. OpenGIS geography markup language (GML) encoding specification. OGC 07-036, Version 3.2.1, August 2007.

[196] R Orchard. Fuzzy reasoning in jess: The fuzzy j toolkit and fuzzy jess. In *Third International Conference on Enterprise Information Systems (ICEIS 2001), Setubal, Portugal*, pages 433–542, July 2001.

[197] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter. Data complexity of query answering in expressive description logics via tableaux. *Journal Automatic Reasoning*, 41(1):61–98, 2008.

[198] Simon E. Overell and Stefan M. Rüger. Identifying and grounding descriptions of places. In Ross Purves and Chris Jones, editors, *Proceedings of the 3rd ACM Workshop On Geographic Information Retrieval, GIR 2006, Seattle, WA, USA, August 10, 2006.* Department of Geography, University of Zurich, 2006.

[199] Simon E. Overell and Stefan M. Rüger. Geographic co-occurrence as a tool for gir. In *Proceedings of the 4th ACM Workshop On Geographic Information Retrieval, GIR 2007*, pages 71–76. ACM, 2007.

[200] Simon E. Overell and Stefan M. Rüger. Using co-occurrence models for placename disambiguation. *International Journal of Geographical Information Science*, 22(3):265–287, 2008.

[201] Dimitris Papadias, Yannis Theodoridis, and Timos K. Sellis. The retrieval of direction relations using r-trees. In *DEXA '94: Proceedings of the 5th International Conference on Database and Expert Systems Applications*, pages 173–182, London, UK, 1994. Springer-Verlag.

[202] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. Owl web ontology language semantics and abstract syntax section 5. rdf-compatible model-theoretic semantics: http://www.w3.org/tr/owl-semantics/rdfs.html#built_in_vocabulary, December 2004.

[203] Norman W. Paton and Oscar Diaz. Active database systems. *ACM Computing Surveys*, 31(1):63–103, 1999.

[204] B. Pernici, I. Mirbel, and N. R. Brisaboa. Constraints in spatio-temporal databases: A proposal of classification: http://citeseer.ist.psu.edu/427798.html, June 04 1998.

[205] Adrian Popescu, Gregory Grefenstette, and Pierre-Alain Moëllic. Gazetiki: automatic creation of a geographical gazetteer. In *ACM/IEEE Joint Conference on Digital Libraries*, pages 85–93. ACM, 2008.

[206] Alessandro Provetti and Tran Cao Son, editors. *Answer Set Programming, Towards Efficient and Scalable Knowledge Representation and Reasoning, Proceedings of the 1st Intl. ASP'01 Workshop, Stanford, March 26-28, 2001*, 2001.

[207] Eric Prud'Hommeaux and Andy Seaborne. SPARQL query language for RDF. World Wide Web Consortium, Recommendation http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115, January 2008.

[208] T. C. Przymusinski. *On the Semantics of Stratified Deductive Databases and Logic Programs.* Los Altos, CA: Morgan-Kaufman, 1987.

[209] T. C. Przymusinski. Every logic program has a natural stratification and an iterated least fixed point model. In *PODS '89: Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 11–21, New York, NY, USA, 1989. ACM.

[210] Teodor C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Computing*, 9:401–424, 1991.

[211] D. Pullar and M. Egenhofer. Toward formal definitions of topological relations among spatial objects. In *Proceedings of the 3rd International Symposium on Spatial Data Handling*, pages 165–176, Sydney, Australia, 1988.

[212] Raghu Ramakrishnan and Jeffrey D. Ullman. A survey of research on deductive database systems. *Journal of Logic Programming*, 23:125–149, 1993.

[213] D. A. Randell, Anthony G. Cohn, and Z. Cui. Computing transitivity tables: A challenge for automated theorem provers. In Deepak Kapur, editor, *11th International Conference on Automated Deduction*, LNAI 607, pages 786–790, Saratoga Springs, New York, USA, June 15–18, 1992. Springer-Verlag.

[214] D. A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proceedings 3rd International Conference on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.

[215] Ray Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

[216] Raymond Reiter. Nonmonotonic reasoning. In Howard E. Shrobe, editor, *Exploring Artificial Intelligence*, pages 439–482. Morgan Kaufmann, San Mateo, California, 1988.

[217] J. Renz. A Canonical Model of the Region Connection Calculus. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning, Proceedings of the 6th International Conference (KR98)*, pages 330–341, San Francisco, 1998. Morgan Kaufmann Publishers.

[218] Jochen Renz. A canonical model of the region connection calculus. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 6th International Conference (KR-98*, pages 330–341. Morgan Kaufmann, 1997.

[219] Jochen Renz. Maximal tractable fragments of the region connection calculus: A complete analysis. In Dean Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1)*, pages 448–455, S.F., July 31–August 6 1999. Morgan Kaufmann Publishers.

[220] Jochen Renz. *Qualitative Spatial Reasoning with Topological Information*, volume 2293 of *Lecture Notes in Computer Science*. Springer, 2002.

[221] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artificial Intelligence*, 108:69–123, 1997.

[222] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.

[223] Panos Rondogiannis and William W. Wadge. Minimum model semantics for logic programs with negation-as-failure. *ACM Trans. Comput. Logic*, 6(2):441–467, 2005.

[224] Riccardo Rosati. Towards expressive KR systems integrating datalog and description logics: preliminary report. In *Description Logics 99*, pages 160–164, 1999.

[225] Riccardo Rosati. Semantic and computational advantages of the safe integration of ontologies and rules. In François Fages and Sylvain Soliman, editors, *Principles and Practice of Semantic Web Reasoning, Third International Workshop, PPSWR 2005, Dagstuhl Castle, Germany, September 11-16, 2005, Proceedings*, volume 3703 of *Lecture Notes in Computer Science*, pages 50–64. Springer, 2005.

[226] Riccardo Rosati. DL+log: Tight integration of description logics and disjunctive datalog. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 68–78. AAAI Press, 2006.

[227] Peter Ross, Dave Corne, and Hsiao-Lan Fang. Genetic algorithms for timetabling and scheduling problems. In S. Stols, editor, *Expert 94. (Pretoria, South Africa*, 1994.

[228] Mark Sanderson and Janet Kohler. Analyzing geographic queries. In *Workshop on Geographic Information Retrieval SIGIR*, 2004.

[229] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[230] Manfred Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In *Knowledge Representation*, 55860-032-9, pages 421–431, 1989.

[231] S. Schockaert. *Reasoning about Fuzzy Temporal and Spatial Information from the Web.* PhD thesis, Ghent University, 2008.

[232] Steven Schockaert, Martine De Cock, Chris Cornelis, and Etienne E. Kerre. Fuzzy region connection calculus: Representing vague topological information. *Int. J. Approx. Reasoning*, 48(1):314–331, 2008.

[233] Steven Schockaert and Philip D. Smart. Evolutionary strategies for expressive spatial reasoning. In *Workshop on Soft Methods for Statistical and Fuzzy Spatial Information processing, Workshop at SMPS 2008*, pages 26–39, 2008.

[234] Andy Seaborne. RDQL - A Query Language for RDF (Member Submission) : http://www.w3.org/submission/2004/subm-rdql-20040109. Technical report, Hewlett-Packard, January 2004.

[235] Sylvie Servigne, Thierry Ubeda, Alain Puricelli, and Robert Laurini. A methodology for spatial consistency improvement of geographic databases. *GeoInformatica*, 4(1):7–34, 2000.

[236] Jayant Sharma. *Integrated spatial reasoning in geographic information systems:– Combining topology and direction.* PhD thesis, in Spatial Information Science and Engineering–University of Maine, 1996.

[237] Dennis Shasha and Gottfried Vossen, editors. *Information Systems: Databases: Their Creation, Management and Utilization*, volume 34. Elsevier Science Publishers Ltd., 2009.

[238] J. Shepherdson. Negation as failure. *Journal Logic Programming*, 1(1), 1984.

[239] "Mário J. Silva, Bruno Martins, Marcirio Silveira Chaves, Ana Paula Afonso, and Nuno Cardoso". Adding geographic scopes to web resources. *Computers, Environment and Urban Systems*, 30(4):378–399, 2006.

[240] Moninder Singh. Efficient path consistency algorithms for constraint satisfaction problems: http://citeseer.ist.psu.edu/144040.html. Technical report, February 05 1995.

[241] Spiros Skiadopoulos and Manolis Koubarakis. Consistency checking for qualitative spatial reasoning with cardinal directions. In Pascal Van Hentenryck, editor, *Principles and Practice of Constraint Programming - CP 2002, 8th International Conference, CP 2002, Ithaca, NY, USA, September 9-13, 2002, Proceedings*, volume 2470 of *Lecture Notes in Computer Science*, pages 341–355. Springer, 2002.

[242] Spiros Skiadopoulos and Nikos Sarkas. A family of directional relation models for extended objects. *IEEE Trans. on Knowl. and Data Eng.*, 19(8):1116–1130, 2007. Member-Sellis,, Timos and Member-Koubarakis,, Manolis.

[243] Philip D. Smart, Alia I. Abdelmoty, Baher A. El-Geresy, and Christopher B. Jones. A framework for combining rules and geo-ontologies. In *Web Rule Reasoning Systems - RR2007*, volume 4524, pages 133–147, 2007.

[244] Terence R. Smith and James Frew. Alexandria digital library. *Communications of ACM*, 38(4):61–62, 1995.

[245] John F. Sowa, editor. *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. 1558600884. Morgan Kaufmann, San Mateo, CA, 1991.

[246] Stefano Spaccapietra, Nadine Cullot, Christine Parent, and Christelle Vangenot. On spatial ontologies. Internet, 2004. http://lbdwww.epfl.ch/ stefano/geoinfolast.pdf.

[247] J. G. Stell and M. F. Worboys. The algebraic structure of sets of regions. In *Spatial Information Theory, International Conference COSIT'97, Proceedings, volume 1329 of Lecture Notes in Computer Science*, pages 163–174. Springer-Verlag.

[248] Margaret-Anne D. Storey, Natasha F. Noy, Mark A. Musen, Casey Best, Ray W. Fergerson, and Neil Ernst. Jambalaya: an interactive environment for exploring ontologies. In *IUI*, pages 239–239, 2002.

[249] Terrance Swift and David Scott Warren. An abstract machine for SLG resolution: Definite programs. In *SLP*, pages 633–652, 1994.

[250] Bijan Parsia Taowei David Wang. Cropcircles: Topology sensitive visualization of owl class hierarchies. In *5th International Semantic Web Conference*. 2006.

[251] Juliana Freire Terrance, Juliana Freire, Terrance Swift, and David S. Warren. Beyond depth-first: Improving tabled logic programs through alternative scheduling strategies. In *Journal of Functional and Logic Programming*, pages 243–258. Springer-Verlag, 1996.

[252] Evan Tick. *Memory performance of prolog architectures*. 0-89839-254-8. Kluwer Academic Publishers, Norwell, MA, USA, 1988.

[253] Stephan Tobies. Complexity results and practical algorithms for logics in knowledge representation. *CoRR*, cs.LO/0106031, 2001. informal publication.

[254] H. Touati and A. Despain. An Empirical Study of the Warren Abstract Machine. In *International Symposium on Logic Programming*, pages 114–124. San Francisco, IEEE Computer Society, August 1987.

[255] Andre Trudel and Haiyi Zhang. Exploiting the relationship between ia networks and finite domain csps. In *TIME '05: Proceedings of the 12th International Symposium on Temporal Representation and Reasoning*, pages 177–179, Washington, DC, USA, 2005. IEEE Computer Society.

[256] Miroslaw Truszczynski. Computing large and small stable models. *TPLP*, 2(1):1–23, 2002.

[257] Nectaria Tryfona and Max Egenhofer. Multi-resolution spatial databases: Consistency among networks. In *Integrity in Databases-Sixth International Workshop on Foundations of Models and Langauges for Data and Objects*, pages 119–132, September 1996.

[258] D. Tudhope, H. Alani, and C. Jones. Augmenting thesaurus relationships: possibilities for retrieval. In *Journal of Digital Information Vol. 1(8)*, 2001.

[259] Florian A. Twaroch, Christopher B. Jones, and Alia I. Abdelmoty. Acquisition of a vernacular gazetteer from web sources. In *LOCWEB '08: Proceedings of the first international workshop on Location and the web*, pages 61–64, New York, NY, USA, 2008. ACM.

[260] Thierry Ubeda and Max J. Egenhofer. Topological error correcting in gis. In Michel Scholl and Agnès Voisard, editors, *Advances in Spatial Databases, 5th International Symposium, SSD'97, Berlin, Germany, July 15-18, 1997, Proceedings*, volume 1262 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 1997.

[261] Murat Osman Ünalir, Tugba Özacar, and Övünç Öztürk. Reordering query and rule patterns for query answering in a rete-based inference engine. In Mike Dean, Yuanbo Guo, Woochun Jun, Roland Kaschek, Shonali Krishnaswamy, Zhengxiang Pan, and Quan Z. Sheng, editors, *Web Information Systems Engineering - WISE 2005 Workshops, WISE 2005 International Workshops, New York, NY, USA, November 20-22, 2005, Proceedings*, volume 3807 of *Lecture Notes in Computer Science*, pages 255–265. Springer, 2005.

[262] van Emden and Kowalski. The semantics of predicate logic as a programming language. *JACM: Journal of the ACM*, 23, 1976.

[263] Allen van Gelder, Kenneth A. Ross, and John S. Schilpf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–662, 1991.

[264] J Varas, J. Busto, and R. Torguet. Eo domain specific knowledge enabled services (kes-b). In H. Lacoste and L. Ouwehand, editors, *ESA-EUSC 2004 - Theory and Applications of Knowledge-Driven Image Information Mining with Focus on Earth Observation (ESA SP-553)*. Published on CDROM, March 2004.

[265] Marc Vilain and Henry Kautz. Constraint propagation algorithms for temporal reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 377–382, Los Altos, California, 1986. Morgan Kaufmann.

[266] Raphael Volz. *Web ontology reasoning with logic databases.* PhD thesis, Institut für Angewandte Informatik und Formale Beschreibungsverfahren, 2004.

[267] P Vretanos. Open geospatial consortium filter encoding implementation specification. In *Technical report. Retrieved 2008, from: http://www.opengis.org/techno/specs.htm.*, 2005.

[268] Panagiotis A. Vretanos. Opengis filter encoding implementation specification :http://portal.opengeospatial.org/files/?artifact_id=8340. Open Geospatial Consortium Inc., May 2005.

[269] Gerd Wagner. Logic programming with strong negation and inexact predicates. *Journal of Logic and Computation*, 1(6):835–859, December 1991.

[270] M. Wessel. Obstacles on the way to spatial reasoning with description logics – undecidability of ALCRA. technical report fbi–hh–m–297/00, university of hamburg, october 2000. 2000.

[271] Michael Wessel. On spatial reasoning with description logics. In *Proceedings of the International Workshop on Description Logics 2002 (DL2002), number 53 in CEUR-WS*, pages 156–163, 2002.

[272] Michael Wessel. Some practical issues in building a hybrid deductive geographic information system with a DL component. In *Knowledge Representation Meets Databases*, volume 79, 2003.

[273] Michael Wessel, Volker Haarslev, and Ralf Möller. Alc_ra - alc with role axioms. In *Proceedings of the International Workshop in Description Logics 2000 (DL2000), number 33 in CEUR-WS*, pages 21–30, 2000.

[274] Michael Wessels and Ralf Moller. A flexible dl-based architecture for deductive information systems. In *IJCAR-06 Workshop on Empirically Succesful Computerized Reasoning (ESCoR)*, pages 92–111, 2006.

[275] Jan Wielemaker, Guus Schreiber, and Bob Wielinga. Prolog-based infrastructure for rdf: Scalability and performance. In *The SemanticWeb - ISWC 2003*, pages 644–658. Springer-Verlag, 2003.

[276] S. Winter. Ontology – buzzword or paradigm shift in gi science. *International Journal of Geographical Information Science*, (7):587–590, October 2001.

[277] M. F. Worboys and M. Duckham. *GIS: A Computing Perspective, Second Edition.* CRC Press, 2005.

[278] Michael F Worboys. Nearness relations in environmental space. *International Journal of Geographical Information Science*, 15:2001, 2001.

[279] Ian Wright and James Marshall. The execution kernel of rc++: Rete*, a faster rete with treat as a special case. *International Journal of Intelligent Games and Simulation*, 2(1):36–48, February 2003.

[280] S. Zlatanova. On 3d topological relationships. In *DEXA '00: Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, page 913, Washington, DC, USA, 2000. IEEE Computer Society.