



Deposited via The University of Leeds.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/id/eprint/104356/>

Version: Accepted Version

Article:

Du, H and Alechina, N (2016) Qualitative Spatial Logics for Buffered Geometries. *Journal of Artificial Intelligence Research*, 56. pp. 693-745. ISSN: 1943-5037

<https://doi.org/10.1613/jair.5140>

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

Qualitative Spatial Logics for Buffered Geometries

Heshan Du

University of Leeds, UK

H.DU@LEEDS.AC.UK

Natasha Alechina

University of Nottingham, UK

NATASHA.ALECHINA@NOTTINGHAM.AC.UK

Abstract

This paper describes a series of new qualitative spatial logics for checking consistency of *sameAs* and *partOf* matches between spatial objects from different geospatial datasets, especially from crowd-sourced datasets. Since geometries in crowd-sourced data are usually not very accurate or precise, we buffer geometries by a margin of error or a level of tolerance $\sigma \in \mathbb{R}_{\geq 0}$, and define spatial relations for buffered geometries. The spatial logics formalize the notions of ‘buffered equal’ (intuitively corresponding to ‘possibly sameAs’), ‘buffered part of’ (‘possibly partOf’), ‘near’ (‘possibly connected’) and ‘far’ (‘definitely disconnected’). A sound and complete axiomatisation of each logic is provided with respect to models based on metric spaces. For each of the logics, the satisfiability problem is shown to be NP-complete. Finally, we briefly describe how the logics are used in a system for generating and debugging matches between spatial objects, and report positive experimental evaluation results for the system.

1. Introduction

The motivation for our work on qualitative spatial logics comes from the needs of integrating disparate geospatial datasets, especially crowd-sourced geospatial datasets. Crowd-sourced data involves non-specialists in data collection, sharing and maintenance. Compared to authoritative geospatial data, which is collected by surveyors or other geodata professionals, crowd-sourced data is less accurate and less well structured, but often provides richer user-based information and reflects real world changes more quickly at a much lower cost (Jackson, Rahemtulla, & Morley, 2010). It is in the interests of national mapping agencies, government organisations, and all other users of geospatial data to be able to integrate and use different geospatial data synergistically.

Geospatial data matching refers to the problem of establishing correspondences (matches) between spatial objects represented in different geospatial datasets. It is an essential step for data comparison, data integration or enrichment, change detection and data update. Matching authoritative geospatial data and crowd-sourced geospatial data is a non-trivial task. Geometry representations of the same location or place in different datasets are usually not exactly the same. Objects are also sometimes represented at different levels of granularity. For example, consider geometries of objects in Nottingham city centre given by Ordnance Survey of Great Britain (OSGB) (2012) and by OpenStreetMap (OSM) (2012) in Figure 1. The position and shape of the Prezzo Ristorante are represented differently in OSGB data (dotted) and OSM data (solid). The Victoria Shopping Centre is represented as a whole in OSM, and as several shops in OSGB.

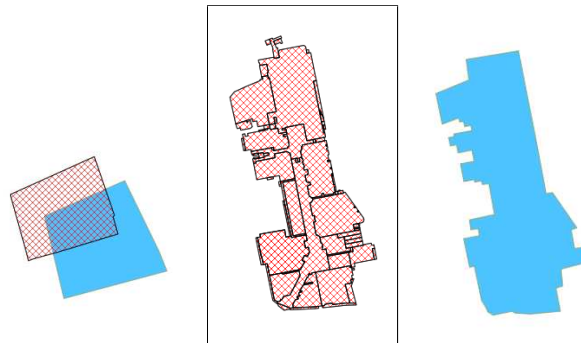


Figure 1: Prezzo Ristorante and Victoria Shopping Centre represented in OSGB (dotted) and OSM (solid)

In order to integrate the datasets, we need to determine which objects are the same (represent the same entity) and sometimes which objects in one dataset are parts of objects in the other dataset (as in the example of Victoria Shopping Centre). The statements representing these two types of relations are referred to as *sameAs* matches and *partOf* matches respectively. One way to produce such matches is to use locations and geometries of objects, although of course we also use any lexical labels associated with the objects, such as names of restaurants etc. In our previous work (Du, Alechina, Jackson, & Hart, 2016), we present a method which generates matches using both location and lexical information about spatial objects. As generated matches may contain errors, they are seen as retractable assumptions and require further validation and checking. One way is to use logical reasoning to check the consistency of matches with respect to statements in input datasets. By using description logic reasoning, the correctness of matches can be checked with respect to classification information. For example, it is wrong to state that spatial objects a and b are the same, if a is a *Bank* and b is a *Clinic*, because the concepts *Bank* and *Clinic* are disjoint, containing no common elements. However, this is not sufficient for validating matches between spatial objects¹. For example, two spatial objects which are close to each other in one dataset cannot be matched to two spatial objects which are far away apart in the other dataset, no matter whether they are of the same type or not. Therefore, spatial reasoning is required to validate matches with regard to location information, in addition to description logic reasoning.

Spatial logic studies relations between geometrical structures and spatial languages describing them (Aiello, Pratt-Hartmann, & van Benthem, 2007). There are a variety of spatial relations, such as topological connectedness of regions, relations based on distances, relations for expressing orientations or directions, etc. In a spatial logic, spatial relations are represented in a formal language, such as first order logic or its fragments, and inter-

1. There are works (Lutz & Milicic, 2007) on extending description logics with concrete domains or constraint systems, such as the region connection calculus (RCC) (Randell, Cui, & Cohn, 1992) and Allen’s Interval Algebra (Allen, 1983). The description logic reasoner Pellet (Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007) was extended to PelletSpatial (Stocker & Sirin, 2009), which supports qualitative spatial reasoning in RCC. However, later we will show that it is not appropriate to use RCC in our application.

preted over some structures based on geometrical spaces, such as topological spaces, metric spaces and Euclidean spaces. In the field of qualitative spatial reasoning, several spatial formalisms have been developed for representing and reasoning about topological relations, such as the Region Connection Calculus (RCC) (Randell et al., 1992), the 9-intersection model (Egenhofer & Franzosa, 1991) and their extensions (Clementini & Felice, 1997; Roy & Stell, 2001; Schockaert, Cock, Cornelis, & Kerre, 2008b, 2008a; Schockaert, Cock, & Kerre, 2009). In addition, there are formalisms for representing and reasoning about directional relations (Frank, 1991, 1996; Ligozat, 1998; Balbiani, Condotta, & del Cerro, 1999; Goyal & Egenhofer, 2001; Skiadopoulos & Koubarakis, 2004), as well as relative or absolute distances (Zimmermann, 1995; Clementini, Felice, & Hernández, 1997; Wolter & Zakharyashev, 2003, 2005). Recent comprehensive surveys on qualitative spatial representations and reasoning are provided by Cohn and Renz (2008) and Chen, Cohn, Liu, Wang, OuYang, and Yu (2015).

Qualitative spatial reasoning has been shown to be applicable to geospatial data (Bennett, 1996; Bennett, Cohn, & Isli, 1997; Guesgen & Albrecht, 2000; Mallenby, 2007; Mallenby & Bennett, 2007; Li, Liu, & Wang, 2013), where location information of spatial objects comes from a single data source. The application described in this paper is different, as location representations about the same spatial object come from different sources and are usually not exactly the same. Rather than treating all the differences in geometric representations as logical contradictions, we would tolerate slight geometric differences and only treat qualitatively defined ‘large’ differences as logical contradictions used for detecting wrong matches. More specifically, after establishing matches between two sets of spatial objects, if the set of matches gives rise to a contradiction, then some match must be wrong and should be retracted. In addition, we would provide explanations to help users understand why a contradiction exists and why some matches are wrong. In the following, we assess the appropriateness of several existing spatial formalisms for these purposes.

The Region Connection Calculus (RCC) (Randell et al., 1992) is a first order formalism based on regions and the connection relation C , which is axiomatised to be reflexive and symmetric. Two regions x, y are connected (i.e. $C(x, y)$ holds), if their closures share a point. Based on the connection relation, several spatial relations are defined for regions. Among them, eight jointly exhaustive and pairwise disjoint (JEPD) relations are identified: DC (Disconnected), EC (Externally Connected), PO (Partially Overlap), TPP (Tangential Proper Part), $NTPP$ (Non-Tangential Proper Part), $TPPi$ (Inverse Tangential Proper Part), $NTPPi$ (Inverse Non-Tangential Proper Part) and EQ (Equal). They are referred to as RCC8, which is well-known in the field of qualitative spatial reasoning.

The 9-intersection model is developed by Egenhofer and Franzosa (1991) and Egenhofer and Herring (1991) based on the point-set interpretation of geometries. By comparing the nine intersections between interiors, boundaries and exteriors of point-sets, it identifies 2^9 mutually exclusive topological relations. The 9-intersection model provides a comprehensive formal categorization of binary topological relations between points, lines and regions. Only a small number of these 2^9 relations are realisable in a particular space (Egenhofer & Herring, 1991). Restricting point-sets to simple regions (regions homeomorphic to disks), the 512 relations collapse to the RCC8 relations.

For the described application, we found it difficult to use spatial formalisms such as the Region Connection Calculus (Randell et al., 1992) and the 9-intersection model (Egenhofer

& Franzosa, 1991), since they presuppose accurate geometries or regions with sharp boundaries and define spatial relations based on the connection relation. This is too strict for crowd-sourced geospatial data. As shown in Figure 2, a_1 is *sameAs* a_2 , both representing a Prezzo Ristorante; b_1 is *sameAs* b_2 , both referring to a Blue Bell Inn. Though the *sameAs* matches are correct, a topological inconsistency still exists, since a_1 and b_1 are disconnected (*DC*), a_2 and b_2 are externally connected (*EC*), and the spatial relations *DC* and *EC* are disjoint. Therefore, relations based on connection are too strict for crowd-sourced geospatial data which is possibly inaccurate and may contain errors.

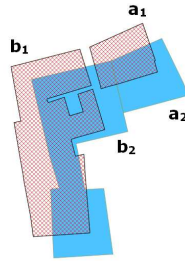


Figure 2: In OSGB data, the Prezzo Ristorante (a_1) and the Blue Bell Inn (b_1) are disconnected, whilst in OSM data, they (a_2 and b_2) are externally connected.

The egg-yolk theory is independently developed by Lehmann and Cohn (1994), Cohn and Gotts (1996b, 1996a), and Roy and Stell (2001) extending the RCC theory and by Clementini and Felice (1996, 1997) extending the 9-intersection model, in order to represent and reason about regions with indeterminate boundaries. In this theory, a region with an indeterminate boundary (an indeterminate region) is represented by a pair of regions, an ‘egg’ and a ‘yolk’, which are the maximum extension and the minimum extension of the indeterminate region respectively (similar to the upper approximation and lower approximation in the rough set theory, Pawlak, Polkowski, & Skowron, 2007). The yolk is not empty and it is always a proper part of the egg. The egg-yolk theory presupposes the existence of a core part of a region and a more vague part. For the described application, the same location can be represented using two disconnected polygons from an authoritative geospatial dataset and a crowd-sourced geospatial dataset respectively. In this case, we could not define a certain inner region for any of the disconnected polygons, otherwise, it is inconsistent to treat them as different representations for the same location.

We are aware that there are several approaches (Fine, 1975; Zadeh, 1975; Smith, 2008) to representing vague concepts and relations, which have been adopted to extend classical theories such as RCC and the 9-intersection model. A main approach is to assign a degree of truth or a degree of membership to concepts and relations. For example, in the fuzzy region connection calculus (fuzzy RCC) (Schockaert et al., 2008b, 2008a, 2009), the connection relation C is defined as a reflexive and symmetric fuzzy relation. For regions a, b , $C(a, b)$ denotes the degree to which a and b are connected. Using C as the primitive relation, every RCC relation R can be redefined to calculate the degree to which R holds. The fuzzy RCC or other similar formalisms may be applied in the case shown in Figure 2. For example, with an appropriate membership function for the relation *EC*, $EC(a_1, b_1) = 0.8$

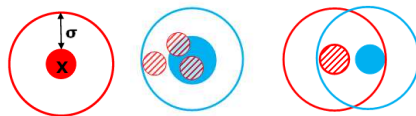


Figure 3: Buffering the geometry X by a distance σ ; three dashed circles are buffered part of (BPT) the solid circle; the dashed circle and the solid circle are buffered equal (BEQ)

and $EC(a_2, b_2) = 1$, then no contradiction will arise. We did not adopt this approach in the matching problem, mainly because there is no good way to define the degree of membership, and it is difficult to generate user-friendly explanations for why the matches are wrong if the underlying reasoning is numerical and relatively obscure.

The logic $MS(M)$ was proposed and developed by Sturm, Suzuki, Wolter, and Zakharyashev (2000), Kutz, Sturm, Suzuki, Wolter, and Zakharyashev (2002), Kutz, Wolter, Sturm, Suzuki, and Zakharyashev (2003), Wolter and Zakharyashev (2003, 2005), and Kutz (2007) for reasoning about distances. The logic $MS(M)$ makes it possible to define concepts such as ‘an object within the distance of 100 meters from a School’. M in $MS(M)$ is a parameter set. A typical example of M is $\mathbb{Q}_{\geq 0}$. The satisfiability problem for a finite set of $MS(\mathbb{Q}_{\geq 0})$ formulas in a metric space is EXPTIME-complete (Wolter & Zakharyashev, 2003). $MS(M)$ was not developed for the problem of geospatial data matching. However, after we designed the logics introduced in this paper, we discovered that they form a proper fragment of $MS(\mathbb{Q}_{\geq 0})$. To detect problematic matches, we also reason about distances between objects, but this reasoning is of a more restricted and qualitative kind. The complexity of the satisfiability problem of our logics is NP-complete, which makes them somewhat more suitable for automatic debugging of matches than the full $MS(\mathbb{Q}_{\geq 0})$. The syntax and semantics of $MS(M)$ and the proofs for the ‘proper fragment relations’ are provided later in this paper (see Section 3).

In this paper, we present a series of new qualitative spatial logics developed for validating matches between spatial objects: a logic of NEAR and FAR for buffered points (LNF) (Du, Alechina, Stock, & Jackson, 2013), a logic of NEAR and FAR for buffered geometries (LNFS) and a logic of part and whole for buffered geometries (LBPT) (Du & Alechina, 2014a, 2014b). The notion of buffer (ISO Technical Committee 211, 2003) is used to model the uncertainty in geometry representations, tolerating slight differences up to a margin of error or a level of tolerance $\sigma \in \mathbb{R}_{\geq 0}$. As shown in Figure 3, the buffer of a geometry X is a geometry which contains exactly all the points within σ distance from X . The buffer of X is denoted as $buffer(X, \sigma)$. For a geometry X which is possibly represented inaccurately within the margin of error σ in one dataset, its corresponding representation in the other dataset is assumed to be somewhere within $buffer(X, \sigma)$.

The spatial logics involve four spatial relations BufferedPartOf (BPT), BufferedEqual (BEQ), NEAR and FAR. They formalize the notions of ‘possibly partOf’, ‘possibly sameAs’, ‘possibly connected’ (given a possible displacement by σ) and ‘definitely disconnected’ (even if displaced by σ) respectively. A geometry X is BufferedPartOf a geometry X' , if X is within $buffer(X', \sigma)$; two geometries are BufferedEqual, if they are BufferedPartOf

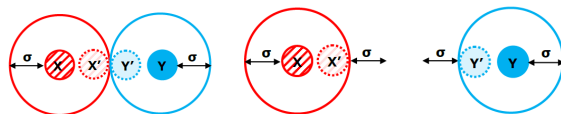


Figure 4: NEAR and FAR

each other (Figure 3). We assume that two geometries X and X' from two different datasets may correspond to the same object if they are *BufferedEqual*. The parameter σ captures the margin of error in representation of geometries. Two geometries X, Y are *NEAR*, if the corresponding geometries X', Y' in the other dataset could be connected, i.e. $distance(X, Y) \in [0, 2\sigma]$ (Figure 4). Clearly, if $FAR(X, Y)$ holds, then $NEAR(X, Y)$ should be false for X and Y from the same dataset. In addition, we want to exclude the possibility that $NEAR(X', Y')$ may hold for X', Y' (corresponding to X, Y respectively) in the other dataset. Therefore we define $FAR(X, Y)$ as $distance(X, Y) \in (4\sigma, +\infty)$ (Figure 4). It is possible that two geometries X, Y are not *NEAR* and not *FAR*, this is, $distance(X, Y) \in (2\sigma, 4\sigma]$.

The way of defining *BEQ*, *NEAR* and *FAR* is similar to that for defining distance relations between points by Moratz and Wallgrün (2012), where each point is assigned one or more reference distances. The distance relations between two points X, Y are defined by comparing the distance between X, Y to the reference distances of X and those of Y . As different points can have different reference distances for indicating nearness, distance relations may not be symmetric. Differing from the work by Moratz and Wallgrün (2012), the relations we defined are not only for points but also for general geometries, and every geometry has the same reference distances (σ , 2σ and 4σ), which leads to the symmetric definitions of *BEQ*, *NEAR* and *FAR*. We provide sound and complete sets of axioms to support reasoning about *BEQ*, *BPT*, *NEAR* and *FAR* relations (see Section 4). This reasoning is useful for verifying matches between spatial representations from different sources.

As explained in our previous work (Du et al., 2013), though the relations are named *NEAR* and *FAR*, we do not attempt to model human notions of nearness or proximity, which is influenced by several factors, such as absolute distance, relative distance, frame of reference, object size, travelling costs and reachability, travelling distance and attractiveness of objects (Guesgen & Albrecht, 2000). In this work, we provide a strict mathematical definition for the calculation of whether two objects are to be considered as being *NEAR* or *FAR*, based on a margin of error σ . While this makes our approach less likely to be suitable for the simulation of human notions of nearness, it provides a useful tool for verifying consistency of matches. The following arguments are formalized for checking consistency of *sameAs* and *partOf* matches: if spatial objects a_1, b_1 are *sameAs* or *partOf* spatial objects a_2, b_2 respectively, a_1, b_1 are *NEAR*, a_2, b_2 are *FAR*, then a contradiction exists.

The rest of this paper is structured as follows. Section 2, Section 3 and Section 4 provide an introduction to the new spatial logics: their syntax and semantics, their relationships with the logic $MS(M)$, their axioms and theorems. Section 5 and Section 6 present the proofs of the soundness, completeness, decidability and complexity theorems for LBPT, as the proofs for LNF and LNFS are similar and LBPT is more expressive than LNF and LNFS. Section 7 describes how LBPT is used for debugging matches between objects from

different geospatial datasets. Section 8 discusses the generality and limitations of the spatial logics. Section 9 concludes the paper.

2. Syntax and Semantics

The language $L(LNF)$ is defined as

$$\phi, \psi := BEQ(a, b) \mid NEAR(a, b) \mid FAR(a, b) \mid \neg\phi \mid \phi \wedge \psi$$

where a, b are individual names. $\phi \rightarrow \psi \equiv_{def} \neg(\phi \wedge \neg\psi)$. The language $L(LNFS)$ is exactly the same as $L(LNF)$. The language $L(LBPT)$ is almost the same as $L(LNF)$ and $L(LNFS)$, except that it has BPT instead of BEQ as a predicate. $L(LBPT)$ is defined as

$$\phi, \psi := BPT(a, b) \mid NEAR(a, b) \mid FAR(a, b) \mid \neg\phi \mid \phi \wedge \psi.$$

$L(LNF)$, $L(LNFS)$ and $L(LBPT)$ are all interpreted over models based on a metric space. Every individual name involved in an LNF formula is mapped to a point, whilst each of those involved in an LNFS/LBPT formula is mapped to an arbitrary geometry or a non-empty set of points.

Definition 1 (Metric Space) A metric space is a pair (Δ, d) , where Δ is a non-empty set (of points) and d is a metric on Δ , i.e. a function $d : \Delta \times \Delta \rightarrow \mathbb{R}_{\geq 0}$, such that for any $x, y, z \in \Delta$, the following axioms are satisfied:

1. identity of indiscernibles: $d(x, y) = 0$ iff $x = y$;
2. symmetry: $d(x, y) = d(y, x)$;
3. triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

Definition 2 (Metric Model of LNF) A metric model M of LNF is a tuple (Δ, d, I, σ) , where (Δ, d) is a metric space, I is an interpretation function which maps each individual name to an element in Δ , and $\sigma \in \mathbb{R}_{\geq 0}$ is a margin of error. The notion of $M \models \phi$ (ϕ is true in the model M) is defined as follows:

$$\begin{aligned} M \models BEQ(a, b) & \text{ iff } d(I(a), I(b)) \in [0, \sigma]; \\ M \models NEAR(a, b) & \text{ iff } d(I(a), I(b)) \in [0, 2\sigma]; \\ M \models FAR(a, b) & \text{ iff } d(I(a), I(b)) \in (4\sigma, \infty); \\ M \models \neg\phi & \text{ iff } M \not\models \phi; \\ M \models \phi \wedge \psi & \text{ iff } M \models \phi \text{ and } M \models \psi, \end{aligned}$$

where a, b are individual names, ϕ, ψ are formulas in $L(LNF)$.

Definition 3 (Metric Model of LNFS/LBPT) A metric model M of LNFS/LBPT is a tuple (Δ, d, I, σ) , where (Δ, d) is a metric space, I is an interpretation function which maps each individual name to a non-empty set of elements in Δ , and $\sigma \in \mathbb{R}_{\geq 0}$ is a margin of error. The notion of $M \models \phi$ (ϕ is true in the model M) is defined as follows:

$M \models BPT(a, b)$ iff $\forall p_a \in I(a) \exists p_b \in I(b) : d(p_a, p_b) \in [0, \sigma]$;

$M \models NEAR(a, b)$ iff $\exists p_a \in I(a) \exists p_b \in I(b) : d(p_a, p_b) \in [0, 2\sigma]$;

$M \models FAR(a, b)$ iff $\forall p_a \in I(a) \forall p_b \in I(b) : d(p_a, p_b) \in (4\sigma, \infty)$;

$M \models \neg\phi$ iff $M \not\models \phi$;

$M \models \phi \wedge \psi$ iff $M \models \phi$ and $M \models \psi$,

where a, b are individual names, ϕ, ψ are formulas in $L(LNFS)/L(LBPT)$. $BEQ(a, b)$ is defined as $BPT(a, b) \wedge BPT(b, a)$.

The notions of validity and satisfiability in metric models are standard. A formula is satisfiable if it is true in some metric model. A formula ϕ is valid ($\models \phi$) if it is true in all metric models (hence if its negation is not satisfiable). The logic LNF/LNFS/LBPT is the set of all valid formulas in the language $L(LNF)/L(LNFS)/L(LBPT)$ respectively.

3. Relationship with the logic $MS(M)$

The logic $MS(M)$, as well as its variations, was developed by Sturm et al. (2000), Kutz et al. (2002, 2003), Wolter and Zakharyashev (2003, 2005), and Kutz (2007) for reasoning about distances.

$MS(M)$ is a family of logics defined relative to the parameter set $M \subseteq \mathbb{Q}_{\geq 0}$. M is subject to the following two conditions: if $a, b \in M$ and $a + b \leq r$, then $a + b \in M$, where $r = \sup M$ if M is bounded, otherwise $r = \infty$; if $a, b \in M$ and $a - b > 0$, then $a - b \in M$.

The alphabet of $MS(M)$ consists of

- an infinite list of region variables X_1, X_2, \dots ;
- an infinite list of location constants c_1, c_2, \dots ;
- a set constant $\{c_i\}$ for every location constant c_i ;
- binary distance (δ), equality (\doteq) and membership (\in) predicates;
- the boolean operators \sqcap, \neg (and their derivatives \sqcup, \top and \perp);
- two distance quantifiers $\exists^{<a}, \exists^{\leq a}$ and their duals $\forall^{<a}, \forall^{\leq a}$, for every $a \in M$;
- two universal quantifiers \exists and \forall .

$MS(M)$ terms are defined as:

$$s, t := X_i \mid \{c_i\} \mid \top \mid \perp \mid \neg s \mid s \sqcap t \mid \exists^{<a} s \mid \exists^{\leq a} s \mid \exists s.$$

In addition to standard description logic concept constructions, $MS(M)$ can define a concept of objects which are at a distance less than a from instances of some other concept s : $\exists^{<a} s$, and similarly for a distance at most a . $\forall^{<a} s$ and $\forall^{\leq a} s$ are defined as $\neg \exists^{<a} (\neg s)$ and $\neg \exists^{\leq a} (\neg s)$ respectively.

$MS(M)$ formulas are defined as

$$\phi, \psi := c \in s \mid s \doteq t \mid \delta(c_1, c_2) < a \mid \delta(c_1, c_2) \leq a \mid \neg\phi \mid \phi \wedge \psi.$$

Further, $s \sqsubseteq t$ is an abbreviation for $(s \sqcap t) \doteq s$ and $s \not\sqsubseteq t$ is an abbreviation for $\neg(s \doteq t)$. $\delta(c_1, c_2) > a$ and $\delta(c_1, c_2) \geq a$ are defined as $\neg(\delta(c_1, c_2) \leq a)$ and $\neg(\delta(c_1, c_2) < a)$ respectively.

An $MS(M)$ -model B is a structure of the form:

$$B = \langle W, d, X_1^B, X_2^B, \dots, c_1^B, c_2^B, \dots \rangle$$

where $\langle W, d \rangle$ is a metric space (Definition 1), each X_i^B is a subset of W , and each c_i^B is an element of W . The value of any other $MS(M)$ -term in B is computed inductively as follows:

- $\top^B = W, \perp^B = \emptyset;$
- $\{c_i\}^B = \{c_i^B\};$
- $(\neg s)^B = W - s^B;$
- $(s_1 \sqcap s_2)^B = s_1^B \cap s_2^B;$
- $(\exists^{<a} s)^B = \{x \in W \mid \exists y \in s^B : d(x, y) < a\};$
- $(\exists^{\leq a} s)^B = \{x \in W \mid \exists y \in s^B : d(x, y) \leq a\};$
- $(\exists s)^B = \{x \in W \mid \exists y \in s^B\}.$

$\forall^{<a}, \forall^{\leq a}$ and \forall are dual to $\exists^{<a}, \exists^{\leq a}$ and \exists respectively. For instance,

$$(\forall^{<a} s)^B = \{x \in W \mid \forall y \in W : (d(x, y) < a \rightarrow y \in s^B)\}.$$

The truth condition of $B \models \phi$, where ϕ is an $MS(M)$ -formula, is defined as follows:

- $B \models c \in s$ iff $c^B \in s^B;$
- $B \models s_1 \doteq s_2$ iff $s_1^B = s_2^B;$
- $B \models \delta(k, l) < a$ iff $d(k^B, l^B) < a;$
- $B \models \delta(k, l) \leq a$ iff $d(k^B, l^B) \leq a;$
- $B \models \neg\phi$ iff $B \not\models \phi;$
- $B \models \phi \wedge \psi$ iff $B \models \phi$ and $B \models \psi.$

A set of $MS(M)$ formulas Σ is satisfiable, if there exists an $MS(M)$ -model B such that $B \models \phi$ for every $\phi \in \Sigma$. This is denoted as $B \models \Sigma$.

It is proved below that LNF/LNFS/LBPT are proper fragments of the logic $MS(\mathbb{Q}_{\geq 0})$. Strictly speaking, this only holds when $\sigma \in \mathbb{Q}_{\geq 0}$, but later we will show that a finite set of LNF/LNFS/LBPT formulas is satisfiable where $\sigma \in \mathbb{R}_{\geq 0}$, if it is satisfiable when $\sigma = 1$. In other words, σ acts as a scaling factor (see the proof of Lemma 43).

Lemma 1 *For individual names a, b , the $MS(M)$ formula $\{a\} \sqsubseteq \neg\{b\}$ is not expressible in LNF.*

Proof. Let M_1, M_2 be metric models². $M_1 = (\Delta_1, d, I_1, \sigma)$, $M_2 = (\Delta_2, d, I_2, \sigma)$.

In M_1 , $\Delta_1 = \{o_1, o_2\}$, $d(o_1, o_2) = \sigma$. $I_1(a) = o_1$, $I_1(b) = o_2$. For any individual name x differing from a, b , $I_1(x) = o_1$.

In M_2 , $\Delta_2 = \{o\}$. $I_2(a) = o$, $I_2(b) = o$. For any individual name x differing from a, b , $I_2(x) = o$. For any individual name y , $I_i(\{y\}) = \{I_i(y)\}$, $i \in \{1, 2\}$.

By the definitions of M_1, M_2 , for any individual names x, y , $d(I_1(x), I_1(y)) \in [0, \sigma]$, $d(I_2(x), I_2(y)) = 0$. If ϕ is an atomic LNF formula about x, y , then by Definition 2, $M_1 \models \phi$ iff $M_2 \models \phi$. By an easy induction on logical connectives, for any LNF formula ϕ , $M_1 \models \phi$ iff $M_2 \models \phi$.

Since $I_1(\{a\}) = \{o_1\}$, $I_1(\{b\}) = \{o_2\}$ and $I_2(\{a\}) = I_2(\{b\}) = \{o\}$, by the truth definition of $MS(M)$ formulas, $M_1 \models (\{a\} \sqsubseteq \neg\{b\})$, $M_2 \not\models (\{a\} \sqsubseteq \neg\{b\})$. Hence, $\{a\} \sqsubseteq \neg\{b\}$ is not equivalent to any LNF formula. \square

Lemma 2 *The logic LNF is a proper fragment of the logic $MS(\mathbb{Q}_{\geq 0})$.*

Proof. Every atomic LNF formula is expressible in $MS(\mathbb{Q}_{\geq 0})$:

- $BEQ(a, b) \equiv (\delta(a, b) \geq 0) \wedge (\delta(a, b) \leq \sigma)$;
- $NEAR(a, b) \equiv (\delta(a, b) \geq 0) \wedge (\delta(a, b) \leq 2\sigma)$;
- $FAR(a, b) \equiv (\delta(a, b) > 4\sigma)$.

This means that all LNF formulas can be expressed in a fragment of $MS(\mathbb{Q}_{\geq 0})$ (the image of LNF under the translation above) which only contains location constants, binary distance predicate and boolean connectives \neg, \wedge . By Lemma 1, LNF is a proper fragment of $MS(M)$. \square

Lemma 3 *For individual names a, b , the $MS(M)$ formula $a \sqsubseteq \neg b$ is not expressible in LNFS/LBPT.*

Proof. Let M_1, M_2 be metric models³. $M_1 = (\Delta_1, d, I_1, \sigma)$, $M_2 = (\Delta_2, d, I_2, \sigma)$.

In M_1 , $\Delta_1 = \{o_1, o_2\}$, $d(o_1, o_2) = \sigma$. $I_1(a) = \{o_1\}$, $I_1(b) = \{o_2\}$. For any individual name x differing from a, b , $I_1(x) = \{o_1\}$.

In M_2 , $\Delta_2 = \{o\}$. $I_2(a) = \{o\}$, $I_2(b) = \{o\}$. For any individual name x differing from a, b , $I_2(x) = \{o\}$.

If ϕ is an atomic LNFS/LBPT formula about x, y , then by Definition 3, $M_1 \models \phi$ iff $M_2 \models \phi$. By an easy induction on logical connectives, for any LNFS/LBPT formula ϕ , $M_1 \models \phi$ iff $M_2 \models \phi$.

2. Note that we can construct models in a one-dimensional or two-dimensional Euclidean space in a similar way and prove the lemma.

3. Note that we can construct models in a one-dimensional or two-dimensional Euclidean space in a similar way and prove the lemma.

By the truth definition of $MS(M)$ formulas, $M_1 \models (a \sqsubseteq \neg b)$ and $M_2 \not\models (a \sqsubseteq \neg b)$. Hence, $a \sqsubseteq \neg b$ is not equivalent to any LNFS/LBPT formula. \square

Lemma 4 *The logic LNFS/LBPT is a proper fragment of $MS(\mathbb{Q}_{\geq 0})$.*

Proof. Every atomic LNFS/LBPT formula is expressible in $MS(\mathbb{Q}_{\geq 0})$:

- (For LNFS) $BEQ(a, b)$ iff $(a \sqsubseteq (\exists^{\leq \sigma} b)) \wedge (b \sqsubseteq (\exists^{\leq \sigma} a))$;
- (For LBPT) $BPT(a, b)$ iff $(a \sqsubseteq (\exists^{\leq \sigma} b))$;
- $NEAR(a, b)$ iff $(a \sqcap (\exists^{\leq 2\sigma} b) \neq \perp)$;
- $FAR(a, b)$ iff $(a \sqcap (\exists^{\leq 4\sigma} b) \doteq \perp)$.

Note that the formulas on the right belong to a fragment of $MS(\mathbb{Q}_{\geq 0})$ which is the image of LNFS/LBPT under the translation above.

The correctness of translation of $BEQ(a, b)$ and $BPT(a, b)$ into $MS(\mathbb{Q}_{\geq 0})$ follows directly from the truth definition of BEQ and BPT (Definition 3). To show that the translation of $NEAR$ and FAR are correct, consider that the truth definition of $NEAR(a, b)$ is equivalent to $0 \leq d_{min}(a, b) \leq 2\sigma$ and $FAR(a, b)$ to $d_{min}(a, b) > 4\sigma$, where $d_{min}(a, b) = \inf\{d(p_a, p_b) \mid p_a \in I(a), p_b \in I(b)\}$. It was shown by Wolter and Zakharyashev (2005) that $d_{min}(a, b) \leq m$ iff $a \sqcap (\exists^{\leq m} b) \neq \perp$. This makes the translation of the formulas have the same truth conditions as defined in Definition 3. By Lemma 3, LNFS/LBPT is a proper fragment of $MS(\mathbb{Q}_{\geq 0})$. \square

Wolter and Zakharyashev (2003) proved that the satisfiability problem for a finite set of $MS(\mathbb{Q}_{\geq 0})$ formulas in a metric space is EXPTIME-complete, which provides an upper bound on the complexity of the satisfiability problems of LNF, LNFS and LBPT in a metric space.

Kutz et al. (2002) and Kutz (2007) gave axioms or inference rules connecting $MS(M)$ terms (e.g. $\forall^{\leq 0} s \rightarrow s$) for $MS(M)$ and its variants. However, the axiomatisation we are going to present is for LNF, LNFS and LBPT formulas (corresponding to $MS(M)$ formulas rather than $MS(M)$ terms).

4. Axioms and Theorems

This section presents a sound and complete axiomatisation for the logic LNF/LNFS/LBPT respectively. The axiomatic systems have been used as a basis for a rule-based reasoner described later in Section 7⁴.

The following calculus (which we will also refer to as LNF) is sound and complete for LNF:

Axiom 0 All tautologies of classical propositional logic

4. It is important to have a complete axiomatisation. Otherwise, the reasoner can not detect all the LNF/LNFS/LBPT inconsistencies caused by problematic matches.

Axiom 1 $BEQ(a, a)$;

Axiom 2 $BEQ(a, b) \rightarrow BEQ(b, a)$;

Axiom 3 $NEAR(a, b) \rightarrow NEAR(b, a)$;

Axiom 4 $FAR(a, b) \rightarrow FAR(b, a)$;

Axiom 5 $BEQ(a, b) \wedge BEQ(b, c) \rightarrow NEAR(c, a)$;

Axiom 6 $BEQ(a, b) \wedge NEAR(b, c) \wedge BEQ(c, d) \rightarrow \neg FAR(d, a)$;

Axiom 7 $NEAR(a, b) \wedge NEAR(b, c) \rightarrow \neg FAR(c, a)$;

MP Modus ponens: $\phi, \phi \rightarrow \psi \vdash \psi$.

The following calculus (which we will also refer to as LNFS) is sound and complete for LNFS:

Axiom 0 All tautologies of classical propositional logic

Axiom 1 $BEQ(a, a)$;

Axiom 2 $BEQ(a, b) \rightarrow BEQ(b, a)$;

Axiom 3 $NEAR(a, b) \rightarrow NEAR(b, a)$;

Axiom 4 $FAR(a, b) \rightarrow FAR(b, a)$;

Axiom 5 $BEQ(a, b) \wedge BEQ(b, c) \rightarrow NEAR(c, a)$;

Axiom 6 $BEQ(a, b) \wedge NEAR(b, c) \wedge BEQ(c, d) \rightarrow \neg FAR(d, a)$;

Axiom 8 $NEAR(a, b) \wedge BEQ(b, c) \wedge BEQ(c, d) \rightarrow \neg FAR(d, a)$;

MP Modus ponens: $\phi, \phi \rightarrow \psi \vdash \psi$.

Axiom 7 of the calculus LNF only holds for points, but *not* for arbitrary geometries, because two points within the same line or polygon can be far from each other. Axiom 7 is replaced by Axiom 8 in LNFS. All other axioms in LNFS are the same as those in LNF.

The following calculus (which we will also refer to as LBPT) is sound and complete for LBPT:

Axiom 0 All tautologies of classical propositional logic

Axiom 3 $NEAR(a, b) \rightarrow NEAR(b, a)$;

Axiom 4 $FAR(a, b) \rightarrow FAR(b, a)$;

Axiom 9 $BPT(a, a)$;

Axiom 10 $BPT(a, b) \wedge BPT(b, c) \rightarrow NEAR(c, a)$;

Axiom 11 $BPT(b, a) \wedge BPT(b, c) \rightarrow NEAR(c, a)$;

Axiom 12 $BPT(b, a) \wedge NEAR(b, c) \wedge BPT(c, d) \rightarrow \neg FAR(d, a)$;

Axiom 13 $NEAR(a, b) \wedge BPT(b, c) \wedge BPT(c, d) \rightarrow \neg FAR(d, a)$;

MP Modus ponens: $\phi, \phi \rightarrow \psi \vdash \psi$.

The calculus LBPT is similar to the calculus LNFS. Changing predicates from *BEQ* to *BPT*, LNFS Axioms 1, 6, 8 are replaced by Axioms 9, 12, 13 respectively in LBPT. Since *BPT* is not symmetric, LNFS Axiom 2 does not have a corresponding axiom in LBPT, and LNFS Axiom 5 is replaced by two LBPT axioms, Axiom 10 and Axiom 11.

The notion of derivability $\Gamma \vdash \phi$ in LNF/LNFS/LBPT calculus is standard. A formula ϕ is derivable if $\vdash \phi$. A set Γ is LNF/LNFS/LBPT-inconsistent if for some formula ϕ it derives both ϕ and $\neg\phi$.

We proved the following theorems for LNF, LNFS and LBPT.

Theorem 1 (Soundness and Completeness) *The LNF/LNFS/LBPT calculus is sound and complete for metric models, namely that*

$$\vdash \phi \Leftrightarrow \models \phi$$

(every derivable formula is valid and every valid formula is derivable).

Theorem 2 (Decidability and Complexity) *The satisfiability problem for a finite set of LNF/LNFS/LBPT formulas in a metric space is NP-complete.*

In the following sections, we give proofs of the results above for the case of LBPT. The proofs for LNF and LNFS are similar. For LBPT, we have the following derivable formulas, which we will refer to as facts in the completeness proof:

Fact 14 $BPT(a, b) \rightarrow NEAR(a, b)$;

Fact 15 $NEAR(a, b) \rightarrow \neg FAR(a, b)$;

Fact 16 $NEAR(a, b) \wedge BPT(b, c) \rightarrow \neg FAR(c, a)$;

Fact 17 $BPT(a, b) \rightarrow \neg FAR(a, b)$;

Fact 18 $BPT(a, b) \wedge BPT(b, c) \rightarrow \neg FAR(c, a)$;

Fact 19 $BPT(b, a) \wedge BPT(b, c) \rightarrow \neg FAR(c, a)$;

Fact 20 $BPT(a, b) \wedge BPT(b, c) \wedge BPT(c, d) \rightarrow \neg FAR(d, a)$;

Fact 21 $BPT(b, a) \wedge BPT(b, c) \wedge BPT(c, d) \rightarrow \neg FAR(d, a)$;

Fact 22 $BPT(a, b) \wedge BPT(b, c) \wedge BPT(c, d) \wedge BPT(d, e) \rightarrow \neg FAR(e, a)$;

Fact 23 $BPT(b, a) \wedge BPT(b, c) \wedge BPT(c, d) \wedge BPT(d, e) \rightarrow \neg FAR(e, a)$;

Fact 24 $BPT(b, a) \wedge BPT(c, b) \wedge BPT(c, d) \wedge BPT(d, e) \rightarrow \neg FAR(e, a)$.

As shown by Facts 17-24, a chain of at most four *BPT*s implies the negation of *FAR*, because *FAR* is defined as being $> 4\sigma$ distance away in Definition 3.

5. Soundness and Completeness of LBPT

This section shows that the LBPT calculus is sound and complete for metric models. Though several definitions and lemmas have been presented in our previous work (Du et al., 2013; Du & Alechina, 2014b), the proofs presented here are more complete, structured, accurate (small errors are corrected) and simplified.

The proof of soundness (every LBPT derivable formula is valid: $\vdash \phi \Rightarrow \models \phi$) is by an easy induction on the length of the derivation of ϕ . Axioms 3, 4, 9-13 are valid (by the truth definition of *BPT*, *NEAR* and *FAR*) and modus ponens preserves validity.

In the rest of this section, we prove completeness (every LBPT valid formula is derivable):

$$\models \phi \Rightarrow \vdash \phi$$

We will actually prove that *if a finite set of LBPT formulas Σ is consistent, then there is a metric model satisfying it*. Any finite set of formulas Σ can be rewritten as a formula ψ which is the conjunction of all formulas in Σ . Σ is consistent, iff ψ is consistent ($\not\vdash \neg\psi$). If there is a metric model M satisfying Σ , then M satisfies ψ , thus $\not\models \neg\psi$. Therefore, if we show that if Σ is consistent, then there exists a metric model satisfying it, then we show that if $\not\vdash \neg\psi$, then $\not\models \neg\psi$. This shows that $\not\vdash \phi \Rightarrow \not\models \phi$ and by contraposition we get completeness.

The completeness theorem is proved by constructing a metric model for a maximal consistent set (Definition 4) of any finite consistent set of LBPT formulas (Lemma 5).

Definition 4 (MCS) *A set of formulas Γ in the language $L(\text{LBPT})$ is maximal consistent, if Γ is consistent, and any set of LBPT formulas over the same set of individual names properly containing Γ is inconsistent. If Γ is a maximal consistent set of formulas, then we call it an MCS.*

Proposition 1 (Properties of MCSs) *If Γ is an MCS, then,*

- Γ is closed under modus ponens: if $\phi, \phi \rightarrow \psi \in \Gamma$, then $\psi \in \Gamma$;
- if ϕ is derivable, then $\phi \in \Gamma$;
- for all formulas ϕ : $\phi \in \Gamma$ or $\neg\phi \in \Gamma$;
- for all formulas ϕ, ψ : $\phi \wedge \psi \in \Gamma$ iff $\phi \in \Gamma$ and $\psi \in \Gamma$;
- for all formulas ϕ, ψ : $\phi \vee \psi \in \Gamma$ iff $\phi \in \Gamma$ or $\psi \in \Gamma$.

Lemma 5 (Lindenbaum's Lemma) *If Σ is a consistent set of formulas in the language $L(\text{LBPT})$, then there is an MCS Σ^+ over the same set of individual names such that $\Sigma \subseteq \Sigma^+$.*

Let $\phi_0, \phi_1, \phi_2, \dots$ be an enumeration of LBPT formulas over the same set of individual names as that in Σ . Σ^+ can be defined as follows:

- $\Sigma_0 = \Sigma$;
- $\Sigma_{n+1} = \Sigma_n \cup \{\phi_n\}$, if it is consistent, otherwise, $\Sigma_{n+1} = \Sigma_n \cup \{\neg\phi_n\}$;

- $\Sigma^+ = \bigcup_{n \geq 0} \Sigma_n$.

For a finite consistent set of formulas Σ , we construct a metric model satisfying a maximal consistent set Σ^+ , which contains Σ and is over the same set of individual names as Σ , as follows. Firstly, we equivalently transform Σ^+ to $B(\Sigma^+)$, which is a set of basic quantified formulas. Then we construct a set of distance constraints $D(\Sigma^+)$ from $B(\Sigma^+)$. A key concept here is path-consistency for a set of distance constraints.

Definition 5 (Non-negative Interval) *An interval h is non-negative, if $h \subseteq [0, +\infty)$.*

Definition 6 (Distance Constraint, Distance Range) *A distance constraint is a statement of the form $d(p, q) \in g$, where p, q are constants representing points, $d(p, q)$ stands for the distance between p, q , and g is a non-negative interval, which stands for the distance range for p, q .*

Definition 7 (Composition) *If d_1, d_2 are non-negative real numbers, then the composition of $\{d_1\}$ and $\{d_2\}$ is defined as: $\{d_1\} \circ \{d_2\} = [|d_1 - d_2|, d_1 + d_2]$ ⁵. If g_1, g_2 are non-negative intervals, then their composition is an interval which is the union of all $\{d_1\} \circ \{d_2\}$, where $d_1 \in g_1, d_2 \in g_2$, this is,*

$$g_1 \circ g_2 = \bigcup_{d_1 \in g_1, d_2 \in g_2} \{d_1\} \circ \{d_2\}.$$

It is assumed that a set of distance constraints D contains at most one distance range for each pair of constants p, q involved in D , and D is closed under symmetry, i.e. if $d(p, q) \in g$ is in D , then $d(q, p) \in g$ is in D .

Definition 8 (Path-Consistency) *For a set of distance constraints D , for every pair of different constants p, q involved in D , their distance range is strengthened by successively applying the following operation until a fixed point is reached:*

$$\forall s : g(p, q) \leftarrow g(p, q) \cap (g(p, s) \circ g(s, q))$$

where s is a constant in D , $s \neq p$, $s \neq q$, and $g(p, q)$ denotes the distance range for p, q . This process is called enforcing path-consistency on D . If at a fixed point, for every pair of constants p, q , $g(p, q) \neq \emptyset$, then D is called path-consistent.

In this paper, we say an interval is referred to in the process of enforcing path-consistency on D , if it occurs in D or is involved in the enforcement of the operation $g(p, q) \leftarrow g(p, q) \cap (g(p, s) \circ g(s, q))$. In other words, it is used as $g(p, q)$, $g(p, s)$ or $g(s, q)$. A distance constraint appears in the process of enforcing path-consistency on D , if its distance range (an interval) is referred to in the process of enforcing path-consistency on D .

The way of enforcing path-consistency on a set of distance constraints defined above is almost the same as that of enforcing path-consistency on a binary constraint satisfaction problem (CSP) (Renz & Nebel, 2007; van Beek, 1992), except that the operation $\forall s : g(p, q) \leftarrow g(p, q) \cap (g(p, s) \circ g(s, q))$ (\circ is the composition operator for non-negative intervals, Definition 7) is applied instead of $\forall k : R_{ij} \leftarrow R_{ij} \cap (R_{ik} \circ R_{kj})$ (\circ is the composition operator

5. Based on $d(x, z) \leq d(x, y) + d(y, z)$ (Property 3 of Definition 1).

for relations). The time complexity of the path-consistency algorithm for CSP is $O(n^3)$ (van Beek, 1992; Mackworth & Freuder, 1985), where n is the number of variables involved in the input set of binary constraints. The path-consistency algorithm for CSP can be adapted easily for enforcing path-consistency on a set of distance constraints. The time complexity of the resulting path-consistency algorithm is also $O(n^3)$, where n is the number of constants involved in the input set of distance constraints. Later in this paper, we will show that the process of enforcing path-consistency on $D(\Sigma^+)$ terminates, and a fixed point can be reached in $O(n^3)$ (see Lemma 33).

After constructing a set of distance constraints $D(\Sigma^+)$ from Σ^+ , we prove the Metric Model Lemma, Metric Space Lemma and Path-Consistency Lemma which are stated below. The notion of path-consistency acts as a bridge between the lemmas.

Lemma 6 (Metric Model Lemma) *Let Σ be a finite consistent set of formulas, and Σ^+ be an MCS which contains Σ and is over the same set of individual names as Σ . If a metric space satisfies $D(\Sigma^+)$, then it can be extended to a metric model satisfying Σ^+ .*

Lemma 7 (Metric Space Lemma) *Let Σ be a finite consistent set of formulas, and Σ^+ be an MCS which contains Σ and is over the same set of individual names as Σ . If $D(\Sigma^+)$ is path-consistent, then there is a metric space (Δ, d) such that all distance constraints in $D(\Sigma^+)$ are satisfied.*

Lemma 8 (Path-Consistency Lemma) *Let Σ be a finite consistent set of formulas, and Σ^+ be an MCS which contains Σ and is over the same set of individual names as Σ . Then, $D(\Sigma^+)$ is path-consistent.*

Using these three lemmas, we prove the completeness of LBPT: if a finite set of formulas Σ is LBPT-consistent, then there exists a metric model satisfying it.

Proof. If Σ is LBPT-consistent, by the Lindenbaum's Lemma (Lemma 5), there is an MCS Σ^+ over the same set of individual names such that $\Sigma \subseteq \Sigma^+$. By the Path-Consistency Lemma (Lemma 8) and the Metric Space Lemma (Lemma 7), there is a metric space (Δ, d) such that all distance constraints in $D(\Sigma^+)$ are satisfied. By the Metric Model Lemma (Lemma 6), the metric space can be extended to a metric model M satisfying Σ^+ . Since $\Sigma \subseteq \Sigma^+$, M satisfies Σ . \square

The detailed proofs of the Metric Model Lemma, Metric Space Lemma and Path-Consistency Lemma are provided in Section 5.1, Section 5.2 and Section 5.3 respectively. Note that, in this paper, Σ^+ denotes an MCS which contains a given finite consistent set of formulas Σ and is over the same set of individual names as Σ .

5.1 Metric Model Lemma

This section shows how to construct a set of distance constraints $D(\Sigma^+)$ from Σ^+ , and presents the proof of the Metric Model Lemma.

By the definition and properties of MCSs (Definition 4 and Proposition 1), the following lemma holds.

Lemma 9 *If Σ^+ is an MCS, then for any pair of individual names a, b occurring in Σ , exactly one of the following cases holds in Σ^+ :*

1. $case(a, b) = BPT(a, b) \wedge BPT(b, a)$;
2. $case(a, b) = BPT(a, b) \wedge \neg BPT(b, a)$;
3. $case(a, b) = \neg BPT(a, b) \wedge BPT(b, a)$;
4. $case(a, b) = \neg BPT(a, b) \wedge \neg BPT(b, a) \wedge NEAR(a, b)$;
5. $case(a, b) = \neg NEAR(a, b) \wedge \neg FAR(a, b)$;
6. $case(a, b) = FAR(a, b)$,

where $case(a, b)$ denotes the formula which holds between a, b in each case.

Lemma 9 is proved using LBPT axioms and facts (such as Axiom 3, Facts 14, 15) in the same way as proving the lemma for LNF (see Du et al., 2013). The full proof of Lemma 9 is provided in Appendix A.

The construction of a set of distance constraints $D(\Sigma^+)$ from Σ^+ has two main steps:

Step 1 For every pair of individual names a, b occurring in Σ , we translate $case(a, b)$ into a set of first order formulas which is equi-satisfiable to $case(a, b)$. The union of all such sets of first order formulas is $B(\Sigma^+)$ (hence, $B(\Sigma^+)$ and Σ are equi-satisfiable.). This step is described by Definition 9 and Definition 10.

Step 2 We construct a set of distance constraints $D(\Sigma^+)$ from $B(\Sigma^+)$. This step is described by Definitions 11-13.

For LBPT formulas, there are first order formulas corresponding to their truth definition in Definition 3. We use formulas of the form $d(p, q) \in g$ as abbreviations of their equivalent first order formulas. For example, $d(p, q) \in [0, \sigma]$ abbreviates $d(p, q) \geq 0 \wedge d(p, q) \leq \sigma$. Observe that⁶

- $BPT(a, b)$ and $\forall p_a \in a \exists p_b \in b : d(p_a, p_b) \in [0, \sigma]$ are equi-satisfiable ;
- $NEAR(a, b)$ and $\exists p_a \in a \exists p_b \in b : d(p_a, p_b) \in [0, 2\sigma]$ are equi-satisfiable;
- $FAR(a, b)$ and $\forall p_a \in a \forall p_b \in b : d(p_a, p_b) \in (4\sigma, \infty)$ are equi-satisfiable.

Definition 9 (Basic Quantified Formula) *We refer to the first order formulas of the following forms as basic quantified formulas:*

- $\forall p_a \in a \forall p_b \in b : d(p_a, p_b) \in g$;
- $\exists p_a \in a \exists p_b \in b : d(p_a, p_b) \in g$;

6. Note that by $\forall p_a \in a \exists p_b \in b : d(p_a, p_b) \in [0, \sigma]$, we are actually quantifying over a metric space. In such sense, it is more precise to say, for example, $BPT(a, b)$ is satisfiable in a metric model, iff $\forall p_a \in a \exists p_b \in b : d(p_a, p_b) \in [0, \sigma]$ is satisfiable over a metric space.

- $\forall p_a \in a \exists p_b \in b : d(p_a, p_b) \in g$;
- $\exists p_a \in a \forall p_b \in b : d(p_a, p_b) \in g$,

where g is a non-negative interval. The abbreviations of these four forms are defined as $\forall(a, b, g)$, $\exists(a, b, g)$, $\chi(a, b, g)$ and $\xi(a, b, g)$ respectively. In other words, for example, $\forall(a, b, g) \equiv (\forall p_a \in a \forall p_b \in b : d(p_a, p_b) \in g)$.

Now we translate the formula in each case listed in Lemma 9 into basic quantified formulas, which will be used to count the number of points needed for interpreting individual names occurring in Σ later.

Definition 10 ($\mathbf{B}(\Sigma^+)$) For an MCS Σ^+ over the same set of individual names as Σ , its corresponding set of basic quantified formulas $B(\Sigma^+)$ is constructed as follows. For every pair of individual names a, b , we translate $\text{case}(a, b)$ into basic quantified formulas:

- $\text{translate}(BPT(a, b) \wedge BPT(b, a)) = \{\chi(a, b, [0, \sigma]), \chi(b, a, [0, \sigma])\}$;
- $\text{translate}(BPT(a, b) \wedge \neg BPT(b, a)) = \{\chi(a, b, [0, \sigma]), \xi(b, a, (\sigma, \infty))\}$;
- $\text{translate}(\neg BPT(a, b) \wedge BPT(b, a)) = \{\xi(a, b, (\sigma, \infty)), \chi(b, a, [0, \sigma])\}$;
- $\text{translate}(\neg BPT(a, b) \wedge \neg BPT(b, a) \wedge NEAR(a, b)) = \{\xi(a, b, (\sigma, \infty)), \xi(b, a, (\sigma, \infty)), \exists(a, b, [0, 2\sigma]), \exists(b, a, [0, 2\sigma])\}$;
- $\text{translate}(\neg NEAR(a, b) \wedge \neg FAR(a, b)) = \{\forall(a, b, (2\sigma, \infty)), \forall(b, a, (2\sigma, \infty)), \exists(a, b, [0, 4\sigma]), \exists(b, a, [0, 4\sigma])\}$;
- $\text{translate}(FAR(a, b)) = \{\forall(a, b, (4\sigma, \infty)), \forall(b, a, (4\sigma, \infty))\}$,

where $\sigma \in \mathbb{R}_{\geq 0}$ is a fixed margin of error. Let $\text{names}(\Sigma)$ be the set of individual names occurring in Σ . Then,

$$B(\Sigma^+) = \bigcup_{a \in \text{names}(\Sigma), b \in \text{names}(\Sigma)} \text{translate}(\text{case}(a, b)).$$

In the following, for a set of basic quantified formulas $B(\Sigma^+)$, we construct a set of distance constraints $D(\Sigma^+)$, and then show that if there is a metric space satisfying $D(\Sigma^+)$, then it can be extended to a model of Σ^+ . In other words, we are constructing a metric over a set of points used to interpret individual names.

The number of points needed for interpreting each individual name depends on the numbers of different forms of formulas in $B(\Sigma^+)$. For any individual name a , let us predict how many particular constants in $\text{points}(a)$ (points assigned to an individual name a) can be specified by the finite set of formulas about a in $B(\Sigma^+)$. $\text{points}(a)$ contains at least one constant. If a formula in $B(\Sigma^+)$ says ‘there exists a constant in $\text{points}(a)$ ’, then this constant is a particular constant within $\text{points}(a)$. For any pair of different individual names a, b , if both $\exists(a, b, g)$ and $\exists(b, a, g)$ are in $B(\Sigma^+)$, we only count one of them; if $\chi(a, b, g)$ is in $B(\Sigma^+)$, we map all the constants in $\text{points}(a)$ to the same constant in $\text{points}(b)$. By Lemma 9 and Definition 10, in $B(\Sigma^+)$, for any pair of different individual names a, b and $R \in \{\exists, \xi, \chi\}$ we never have $R(a, b, g_1)$ and $R(a, b, g_2)$, where $g_1 \neq g_2$, at the same time. The cardinality of $\text{points}(a)$ is specified as follows.

Definition 11 ($num(a, B(\Sigma^+))$ and $points(a)$) Let $names(\Sigma)$ be the set of individual names occurring in Σ ⁷. For any individual name $a \in names(\Sigma)$,

$$\begin{aligned} num(\exists a, B(\Sigma^+)) &= |\{b \in names(\Sigma) \mid \exists g : \exists(a, b, g) \in B(\Sigma^+)\}| \\ num(\xi a, B(\Sigma^+)) &= |\{b \in names(\Sigma) \mid \exists g : \xi(a, b, g) \in B(\Sigma^+)\}| \\ num(\chi a, B(\Sigma^+)) &= |\{b \in names(\Sigma) \mid \exists g : \chi(b, a, g) \in B(\Sigma^+)\}| \end{aligned}$$

Then $num(a, B(\Sigma^+)) = num(\exists a, B(\Sigma^+)) + num(\xi a, B(\Sigma^+)) + num(\chi a, B(\Sigma^+))$. $points(a)$ is a set of constants $\{p_a^1, \dots, p_a^n\}$, where $n = num(a, B(\Sigma^+))$.

Definition 12 (Witness for a formula) A witness for a formula $\exists(a, b, g)$ is a pair of constants $p_a \in points(a)$, $p_b \in points(b)$ such that $d(p_a, p_b) \in g$. A witness for a formula $\xi(a, b, g)$ or $\chi(b, a, g)$ is a constant $p_a \in points(a)$, such that for any constant $p_b \in points(b)$, $d(p_a, p_b) \in g$. A constant is clean for a formula, if it is not a witness for any other formula.

Definition 13 ($D(\Sigma^+)$) Let $B(\Sigma^+)$ be the corresponding set of basic quantified formulas of an MCS Σ^+ . For every individual name a in Σ , we assign a fixed set of new constants $points(a)$ to it. We construct a set of distance constraints $D(\Sigma^+)$ as follows, by iterating through the basic quantified formulas in $B(\Sigma^+)$ and eliminating quantifiers on new constants. Initially, $D(\Sigma^+) = \{\}$. For every individual name a in Σ , for every constant $p_a \in points(a)$, we add $d(p_a, p_a) \in \{0\}$ to $D(\Sigma^+)$. For every pair of different individual names a, b , if

- $\exists(a, b, g) \in B(\Sigma^+)$, then we take clean constants $p_a \in points(a)$, $p_b \in points(b)$, and add $d(p_a, p_b) = d(p_b, p_a) \in g$ to $D(\Sigma^+)$, so p_a, p_b become a witness for $\exists(a, b, g)$;
- $\xi(a, b, g) \in B(\Sigma^+)$, then we take a clean constant $p_a \in points(a)$, for every $p_b \in points(b)$, we add $d(p_a, p_b) = d(p_b, p_a) \in g$ to $D(\Sigma^+)$, so p_a becomes a witness for $\xi(a, b, g)$;
- $\xi(b, a, g) \in B(\Sigma^+)$, then we take a clean constant $p_b \in points(b)$, for every $p_a \in points(a)$, we add $d(p_a, p_b) = d(p_b, p_a) \in g$ to $D(\Sigma^+)$, so p_b becomes a witness for $\xi(b, a, g)$;
- $\chi(a, b, g) \in B(\Sigma^+)$, then we take a clean constant $p_b \in points(b)$, for every $p_a \in points(a)$, we add $d(p_a, p_b) = d(p_b, p_a) \in g$ to $D(\Sigma^+)$, so p_b becomes a witness for $\chi(a, b, g)$;
- $\chi(b, a, g) \in B(\Sigma^+)$, then we take a clean constant $p_a \in points(a)$, for every $p_b \in points(b)$, we add $d(p_a, p_b) = d(p_b, p_a) \in g$ to $D(\Sigma^+)$, so p_a becomes a witness for $\chi(b, a, g)$;
- $\forall(a, b, g) \in B(\Sigma^+)$, then for every pair of constants $p_a \in points(a)$, $p_b \in points(b)$, we add $d(p_a, p_b) = d(p_b, p_a) \in g$ to $D(\Sigma^+)$.

For every pair of different constants p, q in $D(\Sigma^+)$, we add $d(p, q) = d(q, p) \in [0, \infty)$ to $D(\Sigma^+)$, then repeatedly replace $d(p, q) = d(q, p) \in g_1$ and $d(p, q) = d(q, p) \in g_2$ with $d(p, q) = d(q, p) \in (g_1 \cap g_2)$, until there is only one distance range for each pair of p, q in $D(\Sigma^+)$.

⁷ By the definition of Σ^+ , Σ^+ contains the same set of individual names as Σ .

In Definition 13, for every pair of different individual names a, b , we check whether $\chi(a, b, g) \in B(\Sigma^+)$ holds and check whether $\chi(b, a, g) \in B(\Sigma^+)$ holds, because it is possible only one of them holds. For the same reason, we check $\xi(a, b, g) \in B(\Sigma^+)$ and $\xi(b, a, g) \in B(\Sigma^+)$ separately. By Definition 10, $\exists(a, b, g) \in B(\Sigma^+)$ iff $\exists(b, a, g) \in B(\Sigma^+)$. Hence we only need to check any one of them. We only check whether $\forall(a, b, g) \in B(\Sigma^+)$ holds, as $\forall(a, b, g) \in B(\Sigma^+)$ iff $\forall(b, a, g) \in B(\Sigma^+)$.

Lemma 10 *When constructing $D(\Sigma^+)$, for any individual name a , the number of clean constants needed from $points(a)$ is no larger than $num(a, B(\Sigma^+))$.*

Proof. By Definition 10, for any individual name a , $\chi(a, a, [0, \sigma])$ is in $B(\Sigma^+)$. By Definition 11, $num(a, B(\Sigma^+)) \geq 1$.

If a is not involved in any formula of the form $\exists(a, b, g)$, $\xi(a, b, g)$ or $\chi(b, a, g)$, for any other individual name b , then by Definition 11, $num(a, B(\Sigma^+)) = 1$. By Definition 13, we need no clean constants from $points(a)$.

Otherwise, by Lemma 9 and Definition 10, in $B(\Sigma^+)$, for any pair of different individual names a, b and $R \in \{\exists, \xi, \chi\}$, we never have $R(a, b, g_1)$ and $R(a, b, g_2)$, where $g_1 \neq g_2$, at the same time. By Definition 13, for each $\exists(a, b, g) \in B(\Sigma^+)$, we take one clean constant from $points(a)$, so $num(\exists a, B(\Sigma^+))$ clean constants are needed in total for all formulas of this form. Similarly, $num(\xi a, B(\Sigma^+))$ and $(num(\chi a, B(\Sigma^+)) - 1)$ clean constants are needed for formulas of forms $\xi(a, b, g)$ and $\chi(b, a, g)$ respectively, where a, b are *different* individual names. We do not need any other clean constant from $points(a)$ for formulas in other forms. By Definition 11, $num(a, B(\Sigma^+))$ is enough. \square

$D(\Sigma^+)$ and $B(\Sigma^+)$ are not equi-satisfiable because of the way we assign witnesses for χ formulas. More specifically, for any pair of different individual names a, b , if $\chi(a, b, g)$ is in $B(\Sigma^+)$, we map all the constants in $points(a)$ to the same constant in $points(b)$. In other words, if $B'(\Sigma^+)$ is the set of formulas resulting from replacing every $\chi(a, b, g)$ in $B(\Sigma^+)$ with $\xi(b, a, g)$, then $D(\Sigma^+)$ and $B'(\Sigma^+)$ are equi-satisfiable. Since every individual name is interpreted as a *non-empty* set of constants, if a model satisfies $\xi(b, a, g)$, then it satisfies $\chi(a, b, g)$, but not vice versa. Hence, constructing $D(\Sigma^+)$ for $B'(\Sigma^+)$ rather than $B(\Sigma^+)$ imposes stronger restrictions (i.e. $\chi(a, b, g)$ in $B(\Sigma^+)$ is replaced with $\xi(b, a, g)$ in $B'(\Sigma^+)$) on the metric space compared to that required by Σ^+ . However, later we will show that if Σ^+ is consistent, then $D(\Sigma^+)$ can be satisfied in a metric space by proving the Metric Space Lemma and Path-Consistency Lemma in the following sections.

Before proving the Metric Model Lemma, let us look at some important properties of $D(\Sigma^+)$, as shown by Lemmas 11-13. The proof of Lemma 11 is provided in Appendix A. Lemma 12 follows from the proof of Lemma 11.

Lemma 11 *For any distance range g occurring in $D(\Sigma^+)$,*

$$g \in \{\{0\}, [0, \sigma], (\sigma, \infty), [0, 2\sigma], (2\sigma, \infty), (2\sigma, 4\sigma], (4\sigma, \infty), [0, \infty)\}.$$

Lemma 12 *If $p \in points(a)$, $q \in points(b)$, and $a \neq b$, then $d(p, q) \in \{0\}$ is not in $D(\Sigma^+)$.*

Lemma 13 *The number of constants in $D(\Sigma^+)$ is finite.*

Proof. It is assumed that Σ is a finite consistent set of formulas over n (a finite number) individual names. By Lemma 9 and Definition 10, $B(\Sigma^+)$ contains at most $f = (n + 2n(n - 1))$ formulas over n individual names. By Definition 11, for any individual name a , $\text{num}(a, B(\Sigma^+)) \leq f$. By Definition 13, the number of constants in $D(\Sigma^+)$ is at most nf . \square

The Metric Model Lemma is proved as follows.

Lemma 14 *If a metric model satisfies $B(\Sigma^+)$, then it satisfies Σ^+ .*

Proof. The lemma follows from two observations. First, by Lemma 9, Σ^+ is entailed by the set $C(\Sigma^+) = \{\text{case}(a, b) : a \in \text{names}(\Sigma^+), b \in \text{names}(\Sigma^+)\}$. Second, by Definition 10, $B(\Sigma^+)$ is a translation of truth conditions of $C(\Sigma^+)$ into first order logic. If a metric model satisfies $B(\Sigma^+)$, then it satisfies $C(\Sigma^+)$, and hence it satisfies Σ^+ . \square

Lemma 6 (Metric Model Lemma) *Let Σ be a finite consistent set of formulas, and Σ^+ be an MCS which contains Σ and is over the same set of individual names as Σ . If a metric space satisfies $D(\Sigma^+)$, then it can be extended to a metric model satisfying Σ^+ .*

Proof. Suppose a metric space satisfies $D(\Sigma^+)$. We extend it to a metric model M by interpreting every individual name a occurring in Σ^+ as $\text{points}(a)$, a 's corresponding set of constants of size $\text{num}(a, B(\Sigma^+))$ (Definition 11 and Definition 13). By Definition 13, any formula of the form $\chi(a, a, [0, \sigma])$ is satisfied by M . For any pair of different individual names, every \exists , ξ or χ formula has a witness, and all \forall formulas are also satisfied by M . Therefore, M is a metric model of $B(\Sigma^+)$. By Lemma 14, M is a metric model of Σ^+ . \square

5.2 Metric Space Lemma

As the process of enforcing path-consistency (Definition 8) involves the application of the composition operator \circ (Definition 7), we present several lemmas in Section 5.2.1 to demonstrate the main calculation rules of \circ and the properties of intervals obtained from composition. In Section 5.2.2, we characterize distance constraints in $D(\Sigma^+)$ and those appearing in the process of enforcing path-consistency on $D(\Sigma^+)$. Using the definitions and lemmas introduced in Section 5.2.1 and Section 5.2.2, the Metric Space Lemma is proved in Section 5.2.3.

5.2.1 THE COMPOSITION OPERATOR

In this section, we present several lemmas to show the main calculation rules of the composition operator \circ and the properties of intervals obtained from composition. These lemmas are important for understanding several proofs in later sections.

Lemmas 15-16 follow from Definition 7.

Lemma 15 *Let g_1, g_2 be non-negative intervals. If $d_3 \in g_1 \circ g_2$, then there exist $d_1 \in g_1$, $d_2 \in g_2$ such that $d_3 \in [|d_1 - d_2|, d_1 + d_2]$.*

Lemma 16 (Calculation of Composition) *If (m, n) , (s, t) , (m, ∞) , (s, ∞) , $\{l\}$, $\{r\}$ are non-negative non-empty intervals, H_1 , H_2 , H are non-negative intervals, then the following calculation rules hold:*

1. $\{l\} \circ \{r\} = [l - r, l + r]$, if $l \geq r$;
2. $\{l\} \circ (s, t) = (s - l, t + l)$, if $s \geq l$;
3. $\{l\} \circ (s, t) = [0, t + l)$, if $l \in (s, t)$;
4. $\{l\} \circ (s, t) = (l - t, t + l)$, if $t \leq l$;
5. $\{l\} \circ (s, +\infty) = (s - l, +\infty)$, if $s \geq l$;
6. $\{l\} \circ (s, +\infty) = [0, +\infty)$, if $s < l$;
7. $(m, n) \circ (s, t) = (s - n, t + n)$, if $s \geq n$;
8. $(m, n) \circ (s, t) = [0, t + n)$, if $(m, n) \cap (s, t) \neq \emptyset$;
9. $(m, n) \circ (s, +\infty) = (s - n, +\infty)$, if $s \geq n$;
10. $(m, n) \circ (s, +\infty) = [0, +\infty)$, if $s < n$;
11. $(m, +\infty) \circ (s, +\infty) = [0, +\infty)$;
12. $H_1 \circ \emptyset = \emptyset$;
13. $H_1 \circ H_2 = H_2 \circ H_1$;
14. $(H_1 \cup H_2) \circ H = (H_1 \circ H) \cup (H_2 \circ H)$;
15. $(\bigcup_k H_k) \circ H = \bigcup_k (H_k \circ H)$, where $k \in \mathbb{N}_{>0}$;
16. $(H_1 \cap H_2) \circ H = (H_1 \circ H) \cap (H_2 \circ H)$, if $(H_1 \cap H_2) \neq \emptyset$;
17. $(H_1 \circ H_2) \circ H = H_1 \circ (H_2 \circ H)$.

In Lemma 16, Rule 14 is a special case of Rule 15, where $k = 2$. Rule 16 states that the composition operation is distributive over non-empty intersections of intervals. It is necessary to require $H_1 \cap H_2 \neq \emptyset$, otherwise the property may not hold. For example, if $H_1 = [0, 1]$, $H_2 = [2, 3]$, $H = [1, 2]$, then $(H_1 \cap H_2) \circ H = \emptyset$ whilst $(H_1 \circ H) \cap (H_2 \circ H) = [0, 3] \cap [0, 5] \neq \emptyset$. A similar property is defined by Li, Long, Liu, Duckham, and Both (2015) for RCC relations. The proofs for the last three calculation rules are provided in Appendix A, whilst others are more obvious.

For an interval h of the form (l, u) , $[l, u)$, $(l, u]$ or $[l, u]$, we call l the greatest lower bound of h , represented as $glb(h)$, and u the least upper bound of h , represented as $lub(h)$. Below we show some interesting properties regarding the composition of intervals and their greatest lower/least upper bounds.

Lemma 17 *For any non-negative non-empty intervals g, h , the following properties hold:*

1. $\text{lub}(g \circ h) = \text{lub}(g) + \text{lub}(h)$;
2. $\text{glb}(g \circ h) \leq \max(\text{glb}(g), \text{glb}(h))$.

Proof. Follows from Lemma 16. \square

A non-empty interval h is right-closed, iff $h = [x, y]$ or $h = (x, y]$. h is right-open, iff $h = [x, y)$ or $h = (x, y)$. h is right-infinite, iff $h = [x, \infty)$ or $h = (x, \infty)$. h is left-closed, iff $h = [x, y]$ or $h = [x, y)$. h is left-open, iff $h = (x, y]$ or $h = (x, y)$.

Lemma 18 *Let g_1, g_2, g_3 be non-negative non-empty right-closed intervals, if $g_1 \subseteq g_2 \circ g_3$, then $\text{lub}(g_1) \leq \text{lub}(g_2) + \text{lub}(g_3)$.*

Proof. Suppose $g_1 \subseteq g_2 \circ g_3$. Since $\text{lub}(g_1) \in g_1$, $\text{lub}(g_1) \in g_2 \circ g_3$. By Lemma 15, there exist $d_2 \in g_2$, $d_3 \in g_3$, such that $\text{lub}(g_1) \leq d_2 + d_3$. Since $d_2 \leq \text{lub}(g_2)$, $d_3 \leq \text{lub}(g_3)$, $\text{lub}(g_1) \leq \text{lub}(g_2) + \text{lub}(g_3)$. \square

Lemma 19 *Let g_1, g_2, g_3 be non-negative non-empty intervals, $g_1 \subseteq g_2 \circ g_3$. If g_1 is right-infinite, then g_2 or g_3 is right-infinite.*

Proof. Suppose g_1 is right-infinite. Since $g_1 \subseteq g_2 \circ g_3$, $g_2 \circ g_3$ is right-infinite. By Definition 7 and Lemma 16, g_2 or g_3 is right-infinite. \square

5.2.2 DISTANCE CONSTRAINTS IN $D(\Sigma^+)$ AND $DS(\Sigma^+)$

In this section, we characterize the distance constraints which appear in the process of enforcing path-consistency on $D(\Sigma^+)$ in two main steps:

Step 1 We characterize the intervals involved in $D(\Sigma^+)$, as well as the composition of those intervals. This step is described by Definition 14 and Lemmas 20-24.

Step 2 We introduce the notion of $DS(\Sigma^+)$ as a set containing all distance constraints appearing in the process of enforcing path-consistency on $D(\Sigma^+)$, and characterize the distance constraints in $DS(\Sigma^+)$. This step is described by Definitions 15-17 and Lemmas 25-31.

Definition 14 (Primitive, Composite, Definable Intervals) *Let h be a non-negative interval. h is primitive, if h is one of $[0, \sigma]$, (σ, ∞) , $[0, 2\sigma]$, $(2\sigma, \infty)$, $(2\sigma, 4\sigma]$, $(4\sigma, \infty)$, $[0, \infty)$. h is composite, if it can be obtained as the composition of at least two primitive intervals. h is definable, if it is primitive or composite.*

Lemma 20 *For any non-negative interval h , $h \circ \{0\} = h$.*

Proof. Follows from Definition 7. \square

Since Lemma 20 holds, we call $\{0\}$ an identity interval.

Lemma 21 *If an interval occurs in $D(\Sigma^+)$, then it is an identity interval or a primitive interval.*

Proof. Follows from Definition 14 and Lemma 11. \square

Lemma 22 *If h is a definable interval, then $h \neq \emptyset$.*

Proof. Follows from Definition 14 and Definition 7. \square

Lemma 23 *If an interval h is definable, then the following properties hold:*

1. $glb(h) = n\sigma$, $n \in \{0, 1, 2, 3, 4\}$;
2. $lub(h) = +\infty$ or $lub(h) = m\sigma$, $m \in \mathbb{N}_{>0}$.

Proof. Let us prove by induction on the structure of h .

Base case: h is primitive. By Definition 14, $n \in \{0, 1, 2, 4\}$, $lub(h) = +\infty$ or $m \in \{1, 2, 4\}$.

Inductive step: Suppose Properties 1, 2 hold for any interval h_t which can be obtained as the composition of t primitive intervals, where $t \in \mathbb{N}_{>0}$ (induction hypothesis). We will show Properties 1, 2 hold for any interval h_{t+1} which can be obtained as the composition of $(t+1)$ primitive intervals.

For any h_{t+1} , there exist an h_t and a primitive interval h_p such that $h_{t+1} = h_t \circ h_p$. By induction hypothesis, $glb(h_t) = n_t\sigma$, $n_t \in \{0, 1, 2, 3, 4\}$; $lub(h_t) = +\infty$ or $lub(h_t) = m_t\sigma$, $m_t \in \mathbb{N}_{>0}$. From the base case, $glb(h_p) = n_p\sigma$, $n_p \in \{0, 1, 2, 4\}$; $lub(h_p) = +\infty$ or $lub(h_p) = m_p\sigma$, $m_p \in \{1, 2, 4\}$. By Lemma 17, $lub(h_{t+1}) = lub(h_t) + lub(h_p)$. Thus, Property 2 holds. By Lemma 16, if

- $lub(h_t) < glb(h_p)$, then $glb(h_{t+1}) = glb(h_p) - lub(h_t)$;
- $lub(h_p) < glb(h_t)$, then $glb(h_{t+1}) = glb(h_t) - lub(h_p)$;
- otherwise, $glb(h_{t+1}) = 0$.

Since $m_t > 0$ and $m_p > 0$, for $glb(h_{t+1}) = n_{t+1}\sigma$, $n_{t+1} < 4$. In each case, $n_{t+1} \in \{0, 1, 2, 3\}$ (Property 1 holds). \square

Lemma 24 *If h is an identity or definable interval, then:*

1. $lub(h) = 0$, iff $h = \{0\}$;
2. $lub(h) = \sigma$, iff $h = [0, \sigma]$;
3. $glb(h) = 4\sigma$, iff $h = (4\sigma, \infty)$.

Proof. Follows from Lemma 17, Lemma 23 and its proof. \square

Now we start to characterize the distance constraints which appear in the process of enforcing path-consistency on $D(\Sigma^+)$.

Definition 15 ($DS(\Sigma^+)$) $DS(\Sigma^+)$ is a minimal set of distance constraints such that the following holds:

- Any distance constraint in $D(\Sigma^+)$ is in $DS(\Sigma^+)$;
- If distance constraints $d(p, q) \in h$ and $d(q, s) \in g$ are in $DS(\Sigma^+)$, then $d(p, s) \in h \circ g$ is in $DS(\Sigma^+)$;
- If distance constraints $d(p, q) \in h$ and $d(p, q) \in g$ are in $DS(\Sigma^+)$, then $d(p, q) \in h \cap g$ is in $DS(\Sigma^+)$,

where p, q, s are constants in $D(\Sigma^+)$.

In the definition above, $DS(\Sigma^+)$ is required to be minimal, such that any interval involved in $DS(\Sigma^+)$ is either in $D(\Sigma^+)$ or is obtained by applying composition or intersection operations on intervals in $D(\Sigma^+)$. For generality, we do not restrict p, q, s to be different constants. For example, it is possible $p = q$.

Lemma 25 *If a distance constraint appears in the process of enforcing path-consistency on $D(\Sigma^+)$, then it is in $DS(\Sigma^+)$.*

Proof. Follows from Definition 8 (path-consistency) and Definition 15. \square

$DS(\Sigma^+)$ covers all the distance constraints appearing in the process of enforcing path-consistency on $D(\Sigma^+)$. However, *not* every distance constraint in $DS(\Sigma^+)$ necessarily appears in the process of enforcing path-consistency on $D(\Sigma^+)$. For example, if $D(\Sigma^+)$ contains exactly one distance constraint $d(p, p) \in [0, \sigma]$, then by Definition 15, $d(p, p) \in [0, 2\sigma]$ is in $DS(\Sigma^+)$ (so is $d(p, p) \in [0, n\sigma]$, for any $n \in \mathbb{N}_{>0}$), but by Definition 8, $d(p, p) \in [0, 2\sigma]$ does not appear in the process of enforcing path-consistency. It is easy to see that $DS(\Sigma^+)$ is an infinite set.

The concept of $DS(\Sigma^+)$ is similar to the concept of distributive subalgebra defined by Li et al. (2015), as the composition operation distributes over non-empty intersections of intervals involved in $DS(\Sigma^+)$ (Rule 16 in Lemma 16). However, in our work, the composition operation is defined for intervals rather than relations.

Lemma 26 *If a distance constraint $d(p, q) \in h$ is in $DS(\Sigma^+)$, then h is a non-negative interval.*

Proof. If a distance constraint $d(p, q) \in h$ is in $D(\Sigma^+)$, by Lemma 11, h is a non-negative interval. By Definitions 5, 7 and the definition of intersection, applying composition or intersection on non-negative intervals, we obtain non-negative intervals. By Definition 15, h is a non-negative interval. \square

Differing from the previous version (Du & Alechina, 2014b), the following definitions and lemmas are restricted to non-empty intervals.

Recall that a non-empty interval h is right-closed, iff $h = [x, y]$ or $h = (x, y]$. h is right-open, iff $h = [x, y)$ or $h = (x, y)$. h is right-infinite, iff $h = [x, \infty)$ or $h = (x, \infty)$. h is left-closed, iff $h = [x, y]$ or $h = [x, y)$. h is left-open, iff $h = (x, y]$ or $h = (x, y)$.

Lemma 27 *If a distance constraint $d(p, q) \in h$ is in $DS(\Sigma^+)$ and $h \neq \emptyset$, then h is either right-infinite or right-closed.*

Proof. Let n denote the total number of times of applying composition or intersection to obtain h , $n \geq 0$. We prove by induction on n .

Base case: $n = 0$, then $d(p, q) \in h$ is in $D(\Sigma^+)$. By Lemma 11, h is either right-infinite or right-closed. Inductive step: Suppose the statement holds for any non-empty h which can be obtained by applying composition or intersection no more than n times (induction hypothesis). We will show it also holds for any non-empty h which can be obtained by applying composition or intersection $(n + 1)$ times.

- If the last step to obtain h is intersection, then by Definition 15, there exist non-empty h_1, h_2 such that $h = h_1 \cap h_2$. By induction hypothesis, for each h_i , $i \in \{1, 2\}$, h_i is either right-infinite or right-closed. By intersection rules, h is either right-infinite or right-closed.
- If the last step to obtain h is composition, then by Definition 15, there exist non-empty h_1, h_2 such that $h = h_1 \circ h_2$. By induction hypothesis, for each h_i , $i \in \{1, 2\}$, h_i is either right-infinite or right-closed. By composition rules (Lemma 16), h is either right-infinite or right-closed.

□

Lemma 28 *For a distance constraint $d(p, q) \in h$ in $DS(\Sigma^+)$ and $h \neq \emptyset$, if $glb(h) \neq 0$, then h is left-open.*

Proof. Let n denote the total number of times of applying composition or intersection to obtain h , $n \geq 0$. We prove by induction on n .

Base case: $n = 0$, then $d(p, q) \in h$ is in $D(\Sigma^+)$. By Lemma 11, if $glb(h) \neq 0$, then h is left-open. Inductive step: Suppose the statement holds for any non-empty h which can be obtained by applying composition or intersection no more than n times (induction hypothesis). We will show it also holds for any non-empty h which can be obtained by applying composition or intersection $(n + 1)$ times.

- If the last step to obtain h is intersection, then by Definition 15, there exist non-empty h_1, h_2 such that $h = h_1 \cap h_2$. By induction hypothesis, for each h_i , $i \in \{1, 2\}$, if $glb(h_i) \neq 0$, then h_i is left-open. By intersection rules, if $glb(h) \neq 0$, then h is left-open.
- If the last step to obtain h is composition, then by Definition 15, there exist non-empty h_1, h_2 such that $h = h_1 \circ h_2$. If $glb(h) \neq 0$, then by composition rules (Lemma 16), $h_1 \cap h_2 = \emptyset$. Suppose $lub(h_1) \leq glb(h_2)$, then $glb(h) = glb(h_2) - lub(h_1)$. By Lemma 26 and $glb(h) \neq 0$, we have $glb(h) > 0$, thus $glb(h_2) > lub(h_1)$. By Lemma 26, $lub(h_1) \geq 0$, thus $glb(h_2) > 0$. By induction hypothesis, h_2 is left-open. By composition rules (Lemma 16), h is left-open. Similarly, this also holds if $lub(h_2) \leq glb(h_1)$.

□

For a distance constraint $d(p, q) \in h$ in $DS(\Sigma^+)$, by Definition 15, h is obtained by applying the composition and/or intersection operations $n \geq 0$ times on intervals occurring in $D(\Sigma^+)$. As applying the intersection operation does not generate any new bound, the greatest lower/least upper bound (and its openness) of h must be the same as that of an interval in $D(\Sigma^+)$ or the composition of intervals in $D(\Sigma^+)$. We formalize this rationale as the concepts of ‘Left-Definable’ and ‘Right-Definable’ to characterize the distance constraints in $DS(\Sigma^+)$. Later, we will show that every distance constraint $d(p, q) \in h$ ($h \neq \emptyset$) in $DS(\Sigma^+)$ is left-definable and right-definable. ‘Left-Definable’ and ‘Right-Definable’ are key concepts for proving the Path-Consistency Lemma, as they establish the correspondences between a distance constraint in $DS(\Sigma^+)$ and a sequence of distance constraints in $D(\Sigma^+)$.

If a non-empty interval h is left-open, then its greatest lower bound is represented as $glb^-(h)$. If h is left-closed, then its greatest lower bound is represented as $glb^+(h)$. If h is right-open, then its least upper bound is represented as $lub^-(h)$. If h is right-closed, then its least upper bound is represented as $lub^+(h)$.

Definition 16 (Left-Definable) *A distance constraint $d(p_1, p_n) \in h_s$ ($n > 1$) is left-definable, iff $h_s \neq \emptyset$ and there exists a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$, such that for $m = h_1 \circ \dots \circ h_{n-1}$, the following holds:*

1. If h_s is left-open, then m is left-open and $glb^-(m) = glb^-(h_s)$;
2. If h_s is left-closed, then m is left-closed and $glb^+(m) = glb^+(h_s)$;
3. $h_s \subseteq m$.

Definition 17 (Right-Definable) *A distance constraint $d(p_1, p_n) \in h_s$ ($n > 1$) is right-definable, iff $h_s \neq \emptyset$ and there exists a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$, such that for $m = h_1 \circ \dots \circ h_{n-1}$, the following holds:*

1. If h_s is right-open, then m is right-open and $lub^-(m) = lub^-(h_s)$;
2. If h_s is right-closed, then m is right-closed and $lub^+(m) = lub^+(h_s)$;
3. $h_s \subseteq m$.

It is important to distinguish the definition of left-definable/right-definable distance constraints (Definitions 16 and 17) from Definition 14 (Definable Intervals). For example, if distance constraints $d(p_1, p_2) \in \{0\}$ and $d(p_2, p_3) \in \{0\}$ are in $D(\Sigma^+)$, then $d(p_1, p_3) \in \{0\}$ is left-definable and right-definable, but $\{0\}$ is not a definable interval. If distance constraints $d(p_1, p_2) \in [0, \sigma]$ and $d(p_2, p_3) \in (4\sigma, \infty)$ are in $D(\Sigma^+)$, then $d(p_1, p_3) \in (3\sigma, 5\sigma]$ is left-definable, but $(3\sigma, 5\sigma]$ is not a definable interval.

Lemma 29 *Let h, g be non-negative intervals. If distance constraints $d(p, q) \in h$ and $d(q, s) \in g$ are left-definable and right-definable, then $d(p, s) \in h \circ g$ is left-definable and right-definable.*

Proof. Since $d(p, q) \in h$ and $d(q, s) \in g$ are right-definable, then by Definition 17, $h \neq \emptyset$, $g \neq \emptyset$. By Definition 7, $h \circ g \neq \emptyset$. By Definition 17, in $D(\Sigma^+)$, there exist a sequence of distance constraints $d(p, x_2) \in h_1, \dots, d(x_{n-1}, q) \in h_{n-1}$ for $d(p, q) \in h$ and a sequence of distance constraints $d(q, y_2) \in g_1, \dots, d(y_{t-1}, s) \in g_{t-1}$ for $d(q, s) \in g$ respectively satisfying the three properties. Let us take the union of the two sequences as a new one, this is, $d(p, x_2) \in h_1, \dots, d(x_{n-1}, q) \in h_{n-1}, d(q, y_2) \in g_1, \dots, d(y_{t-1}, s) \in g_{t-1}$. By composition rules (Lemma 16), the new sequence satisfies the properties in Definition 17 for $d(p, s) \in h \circ g$. Hence, $d(p, s) \in h \circ g$ is right-definable.

By composition rules (Lemma 16), if $h \cap g \neq \emptyset$, then $glb^+(h \circ g) = 0$. We can use the same new sequence above. Let $m_1 = (h_1 \circ \dots \circ h_{n-1})$, $m_2 = (g_1 \circ \dots \circ g_{t-1})$. By Definition 17, $h \subseteq m_1$, $g \subseteq m_2$. Then $m_1 \cap m_2 \neq \emptyset$, therefore, $glb^+(m_1 \circ m_2) = 0$. By Definition 7, $h \circ g \subseteq m_1 \circ m_2$. By Definition 16, $d(p, s) \in h \circ g$ is left-definable.

If $h \cap g = \emptyset$, let us suppose $glb(h) \geq lub(g)$. Since $d(p, q) \in h$ is left-definable and $d(q, s) \in g$ is right-definable, by Definitions 16 and 17 respectively, in $D(\Sigma^+)$, there exist a sequence of distance constraints for $d(p, q) \in h$ and a sequence of distance constraints for $d(q, s) \in g$, satisfying the corresponding properties. Then by composition rules (Lemma 16), the union of the two sequences satisfies the properties in Definition 16 for $d(p, s) \in h \circ g$. Hence, $d(p, s) \in h \circ g$ is left-definable. Similarly, we can show $d(p, s) \in h \circ g$ is left-definable, if $glb(g) \geq lub(h)$. \square

Lemma 30 *Let h, g be non-negative intervals. If distance constraints $d(p, q) \in h$ and $d(p, q) \in g$ are left-definable and right-definable, $h \cap g \neq \emptyset$, then $d(p, q) \in h \cap g$ is left-definable and right-definable.*

Proof. As applying intersections does not generate any new bound and $h \cap g \neq \emptyset$, the left/right bound of $h \cap g$ is the same as that of h or g . If the left bound of $h \cap g$ is the same as that of h , then by Definition 16, the same sequence used for showing $d(p, q) \in h$ is left-definable can be used to show $d(p, q) \in h \cap g$ is left-definable. Other cases are similar. \square

Lemma 31 *If a distance constraint $d(p, q) \in h$ is in $DS(\Sigma^+)$ and $h \neq \emptyset$, then it is left-definable and right-definable.*

Proof. Let n denote the total number of times of applying composition or intersection to obtain h , $n \geq 0$. We prove by induction on n .

Base case: $n = 0$, then $d(p, q) \in h$ is in $D(\Sigma^+)$. By Definitions 16 and 17, $d(p, q) \in h$ is left-definable and right-definable.

Inductive step: Suppose the statement holds for any non-empty h which can be obtained by applying composition or intersection no more than n times (induction hypothesis). We will show it also holds for any non-empty h which can be obtained by applying composition or intersection $(n + 1)$ times. By Definition 15, the last operation to obtain h is either composition or intersection. In the former case, there exist $d(p, s) \in g_1$ and $d(s, q) \in g_2$ in $DS(\Sigma^+)$, such that $g_1 \circ g_2 = h$. As $h \neq \emptyset$, by Definition 7, $g_i \neq \emptyset$, $i \in \{1, 2\}$. Since g_1 and g_2 are obtained by applying composition or intersection no more than n times, then by induction hypothesis, $d(p, s) \in g_1$ and $d(s, q) \in g_2$ are left-definable and right-definable. By

Lemma 29, $d(p, q) \in h$ is left-definable and right-definable. In the latter case, there exist $d(p, q) \in g_1$ and $d(p, q) \in g_2$ in $DS(\Sigma^+)$, such that $g_1 \cap g_2 = h$. As $h \neq \emptyset$, by intersection rules, $g_i \neq \emptyset$, $i \in \{1, 2\}$. By induction hypothesis, $d(p, q) \in g_1$ and $d(p, q) \in g_2$ are left-definable and right-definable. By Lemma 30, $d(p, q) \in h$ is left-definable and right-definable. \square

For generality, we do not exclude the possibility that $d(p, q) \in \emptyset$ is in $DS(\Sigma^+)$. However, it follows from the proof of the Path-Consistency Lemma in Section 5.3 that $d(p, q) \in \emptyset$ is not in $DS(\Sigma^+)$. Alternatively, for a direct proof, see Lemmas 45 and 46 in Appendix C.

5.2.3 PROVING THE METRIC SPACE LEMMA

In the following, we show there is a metric space satisfying $D(\Sigma^+)$, if $D(\Sigma^+)$ is path-consistent (Metric Space Lemma). Firstly, we show that the process of enforcing patch-consistency on $D(\Sigma^+)$ terminates. By Lemma 13, the number of constants in $D(\Sigma^+)$ is finite. Let us suppose the number of constants in $D(\Sigma^+)$ is $t \in \mathbb{N}_{>0}$.

Lemma 32 *Let $t \in \mathbb{N}_{>0}$ be the number of constants in $D(\Sigma^+)$. For any non-empty right-closed interval h referred to in the process of enforcing path-consistency on $D(\Sigma^+)$, $\text{lub}^+(h) \leq 4\sigma t$.*

Proof. If a non-empty right-closed interval h occurs in $D(\Sigma^+)$, then by Lemma 11, $\text{lub}^+(h) \leq 4\sigma \leq 4\sigma t$.

Otherwise, it is generated from the application of composition and/or intersection operators by Definition 8. Composition creates larger least upper bounds (Lemma 17), whilst intersection does not. Since h is right-closed, $\text{lub}^+(h)$ is obtained by composing right-closed intervals only (Lemma 16). Over t constants, the longest chain involves $(t - 1)$ intervals. $\text{lub}^+(h)$ is maximal if we use all of these $(t - 1)$ intervals and the least upper bound of each interval is 4σ . Thus, $\text{lub}^+(h) \leq 4\sigma t$. \square

Lemma 33 *Let $t \in \mathbb{N}_{>0}$ be the number of constants in $D(\Sigma^+)$. Enforcing path-consistency on $D(\Sigma^+)$, a fixed point can be reached in $O(t^3)$.*

Proof. By Definition 8, Lemmas 23 and the fact that intersection does not generate new bounds, for any interval s appearing in the process of enforcing path-consistency on $D(\Sigma^+)$, the following properties hold:

1. $\text{glb}(s) = n\sigma, n \in \{0, 1, 2, 3, 4\}$;
2. $\text{lub}(s) = +\infty$ or $\text{lub}(s) = m\sigma, m \in \mathbb{N}_{>0}$.

For any interval h appearing in $D(\Sigma^+)$, by enforcing path-consistency (Definition 8), h can only become an $h' \subseteq h$. By Lemma 11, $h \neq \emptyset$. By Lemma 27, h is either right-closed or right-infinite, h' is \emptyset , right-closed or right-infinite.

- If h is right-closed, then $h' = \emptyset$ or h' is right-closed. If h' is right-closed, then by Lemma 11, $\text{lub}(h') \leq \text{lub}(h) \leq 4\sigma$. By Properties 1, 2, there are finitely many possibilities for h' .

- If h is right-infinite, then h' is \emptyset , right-closed or right-infinite.
 - If h' is right-closed, then by Lemma 32, $\text{lub}(h') \leq 4t\sigma$. By Properties 1, 2, there are finitely many possibilities for h' .
 - If h' is right-infinite, then by Property 1, there are finitely many possibilities for its greatest lower bound, thus for h' .

Since in each case, there are finitely many possibilities for h' , a fixed point is always reached.

Suppose the widest non-negative interval $[0, \infty)$ appears in the process of enforcing path-consistency on $D(\Sigma^+)$. In the worst case, firstly, $[0, \infty)$ is strengthened to $[0, u]$, where $u \leq 4t\sigma$ (by Lemma 32), then $[0, u]$ is strengthened by σ each time. Hence, $[0, \infty)$ can be strengthened at most $(4t + 1)$ times. Over t constants, by Definition 13, there are $O(t^2)$ distance constraints in $D(\Sigma^+)$. For any interval h appearing in $D(\Sigma^+)$, $h \subseteq [0, \infty)$, hence h can be strengthened at most $(4t + 1)$ times. Therefore, the total time of strengthening all the distance constraints is $O(t^3)$. \square

The following lemma shows how to construct a metric space from $D(\Sigma^+)$. It is used to prove the Metric Space Lemma.

Lemma 34 *Let $t \in \mathbb{N}_{>0}$ be the number of constants in $D(\Sigma^+)$, $D^f(\Sigma^+)$ be a fixed point of enforcing path consistency on $D(\Sigma^+)$. If $D(\Sigma^+)$ is path-consistent, $D_s(\Sigma^+)$ is obtained from $D^f(\Sigma^+)$ by replacing every right-infinite interval with $\{5t\sigma\}$, every right-closed interval h with $\{\text{lub}(h)\}$, then $D_s(\Sigma^+)$ is path-consistent.*

Proof. Suppose $D(\Sigma^+)$ is path-consistent. By Lemma 25, $D^f(\Sigma^+) \subseteq DS(\Sigma^+)$. By Definition 8, for any interval h appearing in $D^f(\Sigma^+)$, $h \neq \emptyset$. By Lemma 27, h is either right-infinite or right-closed. To prove $D_s(\Sigma^+)$ is path-consistent, we only need to show that for any three distance ranges, $\{n_{pq}\}, \{n_{qs}\}, \{n_{ps}\}$ in $D_s(\Sigma^+)$ over three constants p, q, s , we have

1. $n_{pq} \leq n_{qs} + n_{ps}$;
2. $n_{qs} \leq n_{pq} + n_{ps}$;
3. $n_{ps} \leq n_{pq} + n_{qs}$.

Let h_{pq}, h_{qs}, h_{ps} denote the corresponding distance ranges of $\{n_{pq}\}, \{n_{qs}\}, \{n_{ps}\}$ respectively in $D^f(\Sigma^+)$, by Definition 8, we have

- $h_{pq} \subseteq h_{qs} \circ h_{ps}$;
- $h_{qs} \subseteq h_{pq} \circ h_{ps}$;
- $h_{ps} \subseteq h_{pq} \circ h_{qs}$.

We prove $D_s(\Sigma^+)$ is path-consistent by cases:

- If every h_i ($i \in \{pq, qs, ps\}$) is right-closed, then $n_i = \text{lub}(h_i)$. By Lemma 18, 1-3 hold.

- Otherwise, not all of them are right-closed. By Lemma 19, at least two of them are right-infinite.
 - If all of them are right-infinite, then $n_i = 5t\sigma$. Since $5t\sigma \leq 5t\sigma + 5t\sigma$, 1-3 hold.
 - Otherwise, only one of them is right-closed. Let h_{pq} be right-closed. Then, $n_{pq} = \text{lub}(h_{pq})$, $n_{qs} = 5t\sigma$, $n_{ps} = 5t\sigma$. By Lemma 32 and $\sigma \in \mathbb{R}_{\geq 0}$, $\text{lub}(h_{pq}) \leq 4t\sigma < 5t\sigma$. By Lemma 26, $\text{lub}(h_{pq}) \geq 0$. Since $\text{lub}(h_{pq}) < 5t\sigma + 5t\sigma$ and $5t\sigma \leq 5t\sigma + \text{lub}(h_{pq})$, 1-3 hold.

□

Lemma 7 (Metric Space Lemma) *Let Σ be a finite consistent set of formulas, and Σ^+ be an MCS which contains Σ and is over the same set of individual names as Σ . If $D(\Sigma^+)$ is path-consistent, then there is a metric space (Δ, d) such that all distance constraints in $D(\Sigma^+)$ are satisfied.*

Proof. Suppose $D(\Sigma^+)$ is path-consistent. Let Δ be the set of constants in $D(\Sigma^+)$, which is used to interpret individual names occurring in Σ , as shown in Definition 13. If $\Delta = \emptyset$, then it is trivial. Let us assume $\Delta \neq \emptyset$. The number of constants in Δ is denoted by $t \in \mathbb{N}_{>0}$. By Lemma 33, a fixed point $D^f(\Sigma^+)$ can be reached by enforcing path-consistency on $D(\Sigma^+)$. Let $D_s(\Sigma^+)$ be a set of distance constraints obtained from $D^f(\Sigma^+)$ by replacing every right-infinite interval with $\{5t\sigma\}$, every right-closed interval h with $\{\text{lub}(h)\}$. Since every distance constraint in $D_s(\Sigma^+)$ is of the form $d(p, q) \in \{r\}$, where $r \in \mathbb{R}_{\geq 0}$, and $d(p, q) \in \{r\}$ is equivalent to $d(p, q) = r$, a metric (distance function) is defined over Δ . By Definition 13 and Lemma 34, for any pair of constants x, y , if $x = y$, then $d(x, y) = 0$ holds in $D_s(\Sigma^+)$; if $x \neq y$, then $d(x, y) \geq \sigma > 0$ holds in $D_s(\Sigma^+)$. Thus, we have $d(x, y) = 0$ iff $x = y$ in $D_s(\Sigma^+)$. By Definitions 13 and 8, for any pair of constants x, y , $d(x, y) = d(y, x)$ holds in $D_s(\Sigma^+)$. By Lemma 34, $D_s(\Sigma^+)$ is path-consistent. Thus, for any constants x, y, z , $d(x, z) \leq d(x, y) + d(y, z)$ holds in $D_s(\Sigma^+)$. By Definition 1, the (Δ, d) of $D_s(\Sigma^+)$ is a metric space such that all distance constraints in $D(\Sigma^+)$ are satisfied. □

5.3 Path-Consistency Lemma

This section proves the Path-Consistency Lemma by contradiction, supposing that $D(\Sigma^+)$ is not path-consistent. We examine every case where the first \emptyset interval is obtained by enforcing path-consistency. In each case, we show that \perp is derivable from the corresponding LBPT formulas in Σ^+ using LBPT axioms. This contradicts the assumption that Σ^+ is consistent. Lemma 35 is used to generate all possible cases and make sure no duplicated ones are generated. By using Lemma 35, the proof of the Path-Consistency Lemma is largely simplified, compared to the previous version (Du & Alechina, 2014b).

Lemma 35 *Let g, h be non-negative intervals. $g \cap h = \emptyset$ iff $(g \circ h) \cap \{0\} = \emptyset$.*

Proof. If $g \cap h \neq \emptyset$, then by Definition 7, $0 \in (g \circ h)$.

If $0 \in (g \circ h)$, then by Lemma 15, there exist $d_1 \in g$, $d_2 \in h$ such that $0 \in [|d_1 - d_2|, d_1 + d_2]$. Thus, $d_1 = d_2$. Therefore, $g \cap h \neq \emptyset$.

Since $g \cap h \neq \emptyset$ iff $0 \in (g \circ h)$, by contraposition we get $g \cap h = \emptyset$ iff $(g \circ h) \cap \{0\} = \emptyset$. \square

Knowing a least upper bound or a greatest lower bound of a definable interval h , Lemmas 36-42 show all possible ways in which h can be obtained as the composition of primitive intervals. Lemma 36 and Lemma 39 are proved below. Proofs of the other lemmas are similar and omitted.

Lemma 36 *If an interval h is definable, $\text{lub}(h) = 2\sigma$, then h is a primitive interval $[0, 2\sigma]$ or h is obtained as the composition of two $[0, \sigma]$.*

Proof. If h is primitive, then by Definition 14, $h = [0, 2\sigma]$.

If h is composite, then by Definition 14, there exist two definable intervals g_1, g_2 such that $g_1 \circ g_2 = h$. By Lemma 17, $\text{lub}(g_1) + \text{lub}(g_2) = 2\sigma$. By Lemma 23, $\text{lub}(g_1) \geq \sigma$, $\text{lub}(g_2) \geq \sigma$, thus $\text{lub}(g_1) = \sigma$, $\text{lub}(g_2) = \sigma$. By Lemma 24, $h = [0, \sigma] \circ [0, \sigma]$. \square

Lemma 37 *If an interval h is definable, $\text{lub}(h) = 3\sigma$, then h is obtained as the composition of $[0, \sigma]$ and $[0, 2\sigma]$ or as the composition of three $[0, \sigma]$.*

Lemma 38 *If an interval h is definable, $\text{lub}(h) = 4\sigma$, then h is a primitive interval $(2\sigma, 4\sigma]$, or h is obtained as the composition of two $[0, 2\sigma]$, as the composition of two $[0, \sigma]$ and one $[0, 2\sigma]$ or as the composition of four $[0, \sigma]$.*

Lemma 39 *If an interval h is definable, $\text{glb}(h) = 3\sigma$, then h is obtained as the composition of $[0, \sigma]$ and $(4\sigma, \infty)$.*

Proof. By Definition 14, h cannot be primitive.

Since h is composite, then by Definition 14, there exist two definable intervals g_1, g_2 such that $g_1 \circ g_2 = h$. $g_1 \cap g_2 = \emptyset$, otherwise, by Lemma 16, $\text{glb}(h) = 0$.

Without loss of generality, let us suppose $\text{lub}(g_1) \leq \text{glb}(g_2)$. By Lemma 16, $\text{glb}(g_2) - \text{lub}(g_1) = 3\sigma$. By Lemma 23, $\text{glb}(g_2) \leq 4\sigma$, $\text{lub}(g_1) \geq \sigma$, thus $\text{glb}(g_2) = 4\sigma$, $\text{lub}(g_1) = \sigma$. By Lemma 24, h is obtained as the composition of $[0, \sigma]$ and $(4\sigma, \infty)$. \square

Lemma 40 *If an interval h is definable, $\text{glb}(h) = 2\sigma$, then h is a primitive interval $(2\sigma, \infty)$ or $(2\sigma, 4\sigma]$, or h is obtained as the composition of $[0, 2\sigma]$ and $(4\sigma, \infty)$ or as the composition of two $[0, \sigma]$ and one $(4\sigma, \infty)$.*

Lemma 41 *If an interval h is definable, $\text{glb}(h) = \sigma$, then h is a primitive interval (σ, ∞) , or h is obtained as the composition of $[0, \sigma]$ and $(2\sigma, \infty)$, as the composition of $[0, \sigma]$ and $(2\sigma, 4\sigma]$, as the composition of one $[0, \sigma]$, one $[0, 2\sigma]$ and one $(4\sigma, \infty)$, or as the composition of three $[0, \sigma]$ and one $(4\sigma, \infty)$.*

Lemma 42 *If an interval h is definable and left-open, $\text{glb}(h) = 0$, then h is obtained in exactly the following ways:*

- as the composition of $[0, \sigma]$ and (σ, ∞) ;

- as the composition of $[0, 2\sigma]$ and $(2\sigma, \infty)$;
- as the composition of two $[0, \sigma]$ and one $(2\sigma, \infty)$;
- as the composition of $[0, 2\sigma]$ and $(2\sigma, 4\sigma]$;
- as the composition of two $[0, \sigma]$ and one $(2\sigma, 4\sigma]$;
- as the composition of $(2\sigma, 4\sigma]$ and $(4\sigma, \infty)$;
- as the composition of two $[0, 2\sigma]$ and one $(4\sigma, \infty)$;
- as the composition of two $[0, \sigma]$, one $[0, 2\sigma]$ and one $(4\sigma, \infty)$;
- as the composition of four $[0, \sigma]$ and one $(4\sigma, \infty)$.

In our previous work (Du et al., 2013; Du & Alechina, 2014b), we presented a slightly different way to prove the Path-Consistency Lemma for LNF and LBPT respectively: the first empty interval is obtained using the strengthening operator, this is, $g_1 \cap (g_2 \circ g_3) = \emptyset$ and $g_i \neq \emptyset$, where $i \in \{1, 2, 3\}$. g_i may be $\{0\}$ or a primitive interval, or it can be written as $x_i \cap (y_i \circ z_i)$, where each of x_i, y_i, z_i may not be an identity or primitive interval also. In the latter case, since $g_i = x_i \cap (y_i \circ z_i) \neq \emptyset$, x_i, y_i, z_i are all not empty. Since the composition operation is distributive over non-empty intersections of intervals (Rule 16 in Lemma 16), we use Rule 16 repeatedly to rewrite $g_1 \cap (g_2 \circ g_3)$ until every interval is an identity or primitive interval. The final form is $h_1 \cap \dots \cap h_n = \emptyset$, $n > 1$, where h_x ($0 < x \leq n$) is $\{0\}$ or a definable interval. Thus there exist two intervals h_i, h_j ($0 < i \leq n$, $0 < j \leq n$, $i \neq j$) such that $h_i \cap h_j = \emptyset$. Then we look at all the different combinations such that $\text{lub}(h_i) \leq \text{glb}(h_j)$. There are exactly 15 such combinations. In this paper, the proof of the Path-Consistency Lemma is largely simplified. It shows that it is sufficient to examine 5 rather than 15 combinations.

Lemma 8 (Path-Consistency Lemma) *Let Σ be a finite consistent set of formulas, and Σ^+ be an MCS which contains Σ and is over the same set of individual names as Σ . Then, $D(\Sigma^+)$ is path-consistent.*

Proof. Suppose $D(\Sigma^+)$ is not path-consistent. Then by Definitions 8, 15 and Lemma 25, $d(p, q) \in \emptyset$ is in $DS(\Sigma^+)$, for some constants p, q . By Lemma 11, for any distance range g occurring in $D(\Sigma^+)$, $g \neq \emptyset$. By Definitions 15, 7, and intersection rules, the last operation to obtain the first \emptyset interval is intersection. By Definition 15, there exist $d(p, q) \in g_1$ and $d(p, q) \in g_2$ in $DS(\Sigma^+)$, $g_1 \neq \emptyset$, $g_2 \neq \emptyset$, and $g_1 \cap g_2 = \emptyset$. By Lemma 26, g_1, g_2 are non-negative intervals. By Lemma 35, $g_1 \cap g_2 = \emptyset$ iff $(g_1 \circ g_2) \cap \{0\} = \emptyset$.

By Definition 13 and Definition 15, $d(q, p) \in g_2$ is in $DS(\Sigma^+)$. Since $d(p, q) \in g_1$ is in $DS(\Sigma^+)$, by Definition 15, $d(p, p) \in (g_1 \circ g_2)$ is in $DS(\Sigma^+)$. By Definition 7, $g_1 \circ g_2 \neq \emptyset$. By Lemma 31, $d(p, p) \in (g_1 \circ g_2)$ is left-definable and right-definable. Let $h = g_1 \circ g_2$. Since $d(p, p) \in h$ is left-definable, then by Definition 16, there exists a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$, such that $p = p_1 = p_n$ and for $h' = h_1 \circ \dots \circ h_{n-1}$, h and h' have the same greatest lower bound (including both value and

openness) and $h \subseteq h'$. By Definition 14, Lemmas 21 and 20, h' is an identity or definable interval. By Lemma 23, $glb(h') \in \{0, \sigma, 2\sigma, 3\sigma, 4\sigma\}$. Therefore, $(g_1 \circ g_2) \cap \{0\} = \emptyset$ iff one of the following holds:

- $glb(h) \in \{\sigma, 2\sigma, 3\sigma, 4\sigma\}$;
- h is left-open and $glb^-(h) = 0$.

We will check whether \perp can be derived in every case using axioms (or derivable facts). By Axiom 3 and Axiom 4, *NEAR* and *FAR* are both symmetric.

1. $glb(h) = \sigma$: we look at all the different ways where h' is obtained from a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$ such that $p = p_1 = p_n$ and $h' = h_1 \circ \dots \circ h_{n-1}$ (see Definition 16). As every h_i is $\{0\}$ or a primitive interval (by Lemma 21), Lemma 41 specifies all the different ways to obtain h' :
 - (a) h' is a primitive interval $(\sigma, +\infty)$: by Definition 16, $d(p_1, p_n) \in (\sigma, +\infty)$ is in $D(\Sigma^+)$ and $n = 2$. As $p = p_1 = p_n$, $d(p, p) \in (\sigma, +\infty)$ is in $D(\Sigma^+)$. Suppose $p \in points(a)$ for an individual name a in Σ . By the proof of Lemma 11, $(\sigma, +\infty)$ only can come from formulas of the form $\xi(x, y, (\sigma, \infty))$, where x, y are individual names. By Definition 10, $\xi(x, y, (\sigma, \infty))$ only can come from $\neg BPT(x, y)$. Since $d(p, p) \in (\sigma, +\infty)$ is in $D(\Sigma^+)$ and $p \in points(a)$, $\neg BPT(a, a) \in \Sigma^+$. By Axiom 9, $\neg BPT(a, a) \rightarrow \perp$.
 - (b) h' is obtained as the composition of $[0, \sigma]$ and $(2\sigma, \infty)$ or as the composition of $[0, \sigma]$ and $(2\sigma, 4\sigma]$:
by the proof of Lemma 11 and Definition 10, $BPT(a, b) \in \Sigma^+$ or $BPT(b, a) \in \Sigma^+$, $\neg NEAR(a, b) \in \Sigma^+$ and $\neg NEAR(b, a) \in \Sigma^+$.
By Fact 14, $BPT(x_1, x_2) \wedge \neg NEAR(x_1, x_2) \rightarrow \perp$, $\{x_1, x_2\} = \{a, b\}$.
 - (c) h' is obtained as the composition of one $[0, \sigma]$, one $[0, 2\sigma]$ and one $(4\sigma, \infty)$:
by the proof of Lemma 11 and Definition 10, $BPT(a, b) \in \Sigma^+$ or $BPT(b, a) \in \Sigma^+$, $NEAR(b, c) \in \Sigma^+$, $NEAR(c, b) \in \Sigma^+$, $FAR(c, a) \in \Sigma^+$, $FAR(a, c) \in \Sigma^+$. By Fact 16, $BPT(x_2, x_1) \wedge NEAR(x_2, x_3) \wedge FAR(x_3, x_1) \rightarrow \perp$, $\{x_1, x_2, x_3\} = \{a, b, c\}$.
 - (d) h' is obtained as the composition of three $[0, \sigma]$ and one $(4\sigma, +\infty)$:
by the proof of Lemma 11 and Definition 10, we have three *BPT* and one *FAR* over four individual names a, b, c, d . *BPT* refers to either $BPT(x, y)$ or $BPT(y, x)$. Some cases (for example, in Σ^+ , we have $BPT(a, b)$, $BPT(c, b)$, $BPT(d, c)$ and $FAR(a, d)$) are not valid, because different constants will be taken from the same $points(b)$, for an individual name b (by Definition 13). As a consequence, in an invalid case, a sequence consisting of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$, $p = p_1 = p_n$) cannot exist in $D(\Sigma^+)$. We only need to consider all valid cases, which are listed below.
 - i. $BPT(x_1, x_2), BPT(x_2, x_3), BPT(x_3, x_4), FAR(x_4, x_1)$, where $\{x_1, x_2, x_3, x_4\} = \{a, b, c, d\}$. By Fact 20, $BPT(x_1, x_2) \wedge BPT(x_2, x_3) \wedge BPT(x_3, x_4) \wedge FAR(x_4, x_1) \rightarrow \perp$.
 - ii. $BPT(x_2, x_1), BPT(x_2, x_3), BPT(x_3, x_4), FAR(x_4, x_1)$, where $\{x_1, x_2, x_3, x_4\} = \{a, b, c, d\}$. By Fact 21, $BPT(x_2, x_1) \wedge BPT(x_2, x_3) \wedge BPT(x_3, x_4) \wedge FAR(x_4, x_1) \rightarrow \perp$.

Cases 2-5 below use similar arguments. In the following proof, BPT refers to either $BPT(x, y)$ or $BPT(y, x)$ (whichever makes the corresponding case valid). $NEAR$ and FAR are symmetric, thus the order of x, y does not matter.

2. $glb(h) = 2\sigma$: by Definition 16 and Lemma 21, Lemma 40 specifies all the different ways to obtain h' from a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$:
 - (a) h' is a primitive interval $(2\sigma, \infty)$ or $(2\sigma, 4\sigma]$:
 $\neg NEAR(a, a)$, using Axiom 9 and Fact 14.
 - (b) h' is obtained as the composition of $[0, 2\sigma]$ and $(4\sigma, +\infty)$:
one $NEAR$ and one FAR , using Fact 15.
 - (c) h' is obtained as the composition of two $[0, \sigma]$ and one $(4\sigma, +\infty)$:
two BPT and one FAR , using Facts 18 and 19.
3. $glb(h) = 3\sigma$: by Definition 16 and Lemma 21, Lemma 39 specifies all the ways to obtain h' from a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$. By Lemma 39, h' is obtained as the composition of $[0, \sigma]$ and $(4\sigma, +\infty)$.
one BPT and one FAR , using Fact 17.
4. $glb(h) = 4\sigma$: by Definition 16 and Lemma 21, Lemma 24 specifies all the ways to obtain h' from a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$. By Lemma 24, $h' = (4\sigma, +\infty)$. $FAR(a, a)$, using Axiom 9 and Fact 17.
5. $glb^-(h) = 0$: by Definition 16 and Lemma 21, Lemma 42 specifies all the ways to obtain h' from a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($0 < i < n$) in $D(\Sigma^+)$:
 - (a) h' is obtained as the composition of $[0, \sigma]$ and (σ, ∞) : by Definition 13, ensuring no different constants taken from the same $points(x)$,
 $BPT(x_1, x_2) \in \Sigma^+$ and $\neg BPT(x_1, x_2) \in \Sigma^+$, $\{x_1, x_2\} = \{a, b\}$.
 $BPT(x_1, x_2) \wedge \neg BPT(x_1, x_2) \rightarrow \perp$.
 - (b) h' is obtained as the composition of $[0, 2\sigma]$ and $(2\sigma, \infty)$ or as the composition of $[0, 2\sigma]$ and $(2\sigma, 4\sigma]$:
one $NEAR$ and one $\neg NEAR$, using Axiom 3.
 - (c) h' is obtained as the composition of two $[0, \sigma]$ and one $(2\sigma, \infty)$ or as the composition of two $[0, \sigma]$ and one $(2\sigma, 4\sigma]$:
two BPT and one $\neg NEAR$, using Axioms 10 and 11.
 - (d) h' is obtained as the composition of $(2\sigma, 4\sigma]$ and $(4\sigma, \infty)$:
one $\neg FAR$ and one FAR , using Axiom 4.
 - (e) h' is obtained as the composition of two $[0, 2\sigma]$ and one $(4\sigma, \infty)$:
two $NEAR$ and one FAR .
This case is invalid. By Definition 16, $D(\Sigma^+)$ contains $d(p_a, p_b) \in [0, 2\sigma]$, $d(p_b, p_c) \in [0, 2\sigma]$ and $d(p_a, p_c) \in (4\sigma, \infty)$, where $p_a \in points(a)$, $p_b \in points(b)$, $p_c \in points(c)$, for individual names a, b, c . By Definitions 10 and 13, $d(p_a, p_b) \in [0, 2\sigma]$ and $d(p_b, p_c) \in [0, 2\sigma]$ cannot come from $NEAR(a, b)$ and $NEAR(b, c)$ in Σ^+ (by

the proof of Lemma 11, it is clear that they cannot come from other formulas as well), because two different constants will be taken from $\mathit{points}(b)$ as witnesses for $\exists(a, b, [0, 2\sigma])$ and $\exists(b, c, [0, 2\sigma])$ respectively.

- (f) h' is obtained as the composition of two $[0, \sigma]$, one $[0, 2\sigma]$ and one $(4\sigma, \infty)$: two BPT , one $NEAR$ and one FAR , using Axioms 12 and 13.
- (g) h' is obtained as the composition of four $[0, \sigma]$ and one $(4\sigma, \infty)$: four BPT and one FAR , using Facts 22-24.

In each valid case, \perp is derivable using the corresponding axioms or facts, which contradicts the assumption that Σ^+ is consistent. Therefore, $D(\Sigma^+)$ is path-consistent. \square

There is an alternative way to prove the Path-Consistency Lemma, which we believe is longer and more complicated than the one presented in this paper, but since it may provide additional intuitions to the reader, we sketch it in Appendix B.

6. Decidability and Complexity of LBPT

In this section, we establish the complexity of the LBPT satisfiability problem. The complexity of the LNF/LNFS satisfiability problem can be established in a similar way. The complexity of these satisfiability problems is important, as it is related to the complexity of the problem of finding inconsistencies, which is the basis of our approach to debugging matches between geospatial datasets.

Definition 18 (Size of a Formula) *The size of a LBPT formula $s(\phi)$ is defined as follows:*

- $s(BPT(a, b)) = 3$, $s(NEAR(a, b)) = 3$, $s(FAR(a, b)) = 3$;
- $s(\neg\phi) = 1 + s(\phi)$;
- $s(\phi \wedge \psi) = 1 + s(\phi) + s(\psi)$,

where a, b are individual names, ϕ, ψ are formulas in $L(LBPT)$.

As a set of LBPT formulas S and the conjunction of all formulas in S are equi-satisfiable, the combined size of LBPT formulas in a set S is defined as the size of the conjunction of all formulas in S .

Next we prove Theorem 2 for LBPT: the satisfiability problem for a finite set of LBPT formulas in a metric space is NP-complete.

Proof. NP-hardness of the LBPT satisfiability problem follows from NP-hardness of the satisfiability problem for propositional logic, which is included in LBPT.

To prove that the LBPT satisfiability problem is in NP, we show that if a finite set of LBPT formulas Σ is satisfiable, then we can guess a metric model for Σ and verify that this model satisfies Σ , both in time polynomial in the combined size of formulas in Σ .

Suppose Σ is a finite set of LBPT formulas, and the number of individual names in Σ is n . The completeness proof shows that, if Σ is satisfiable, it is satisfiable in a metric model

M of size which is polynomially bounded by the number of individual names in Σ . To recap the construction of the metric model for Σ , first we construct $B(\Sigma^+)$, the corresponding set of basic quantified formulas from an MCS Σ^+ containing Σ , and then construct a model for $B(\Sigma^+)$. By Definition 10, the number of formulas in $B(\Sigma^+)$ is at most $f = (n + 2n(n - 1))$. By Definitions 11 and 13, to every individual name a in Σ , we assign a fixed set of new constants, $points(a) = \{p_a^1, \dots, p_a^x\}$, where $x = num(a, B(\Sigma^+))$. Since $x \leq f$, the number of constants in M is at most $t = nf$. By Lemma 34 and proofs of the Metric Space Lemma, in such a model M , every value assigned by the distance function is of the form $m\sigma$, $m \in \mathbb{N}_{\geq 0}$, $m \leq 5t$.

We guess a metric model M like this. Let s be the combined size of formulas in Σ . Then $n < s$. To every individual name a in Σ , we assign $\{p_a^1, \dots, p_a^x\}$, where $x < 2s^2$. This results in a set of constants Δ , the size of which is $< 2s^3$. To every pair of constants p, q in Δ , we assign $m\sigma$ to $d(p, q)$, where $m \in \mathbb{N}_{\geq 0}$, $m < 10s^3$. To verify that (Δ, d) is a metric space, by Definition 1, it is in $O(s^9)$.

To verify that M satisfies Σ , we need to verify that it satisfies the conjunction of all formulas in Σ . For any $R(a, b)$, where $R \in \{BPT, NEAR, FAR\}$, a, b are individual names, to verify that $R(a, b)$ is satisfied, it takes time which is polynomial in $|points(a)| \times |points(b)|$, thus it is in $O(s^4)$. Hence, verifying that M satisfies Σ can be done in $O(s^5)$. \square

In Section 3, we mentioned that σ acts as a scaling factor of a metric model. This is stated as Lemma 43 and follows from the proofs of the completeness theorem and Theorem 2. The proof for LBPT is provided below. The proof for LNF/LNFS is similar.

Lemma 43 *A finite set of LNF/LNFS/LBPT formulas is satisfiable in a metric model where $\sigma \in \mathbb{R}_{\geq 0}$, iff it is satisfiable in a metric model when $\sigma = 1$.*

Proof.[for LBPT] Suppose Σ is a finite set of LBPT formulas, the number of individual names in Σ is n and the combined size of formulas in Σ is s . By Definition 18, $n < s$. The completeness proof shows that, if Σ is satisfiable, it is satisfiable in a metric model $M = (\Delta, d, I, \sigma)$ constructed as shown in Section 5.1 and Section 5.2. By Definition 13, to every individual name a in Σ , we assign $\{p_a^1, \dots, p_a^x\}$, where $x = num(a, B(\Sigma^+)) < 2s^2$. This results in a set of constants Δ , the size of which is $< 2s^3$. To every constant p in Δ , we assign $d(p, p) = 0$. By Definition 13 and Lemma 34, to every pair of different constants p, q in Δ , we assign $m\sigma$ to $d(p, q)$, where $m \in \mathbb{N}_{> 0}$, $m < 10s^3$. As M is a metric model, by Definition 1 and the proof of the Metric Space Lemma, for any $x, y, z \in \Delta$, we have

1. $d(x, y) = 0$, iff $x = y$;
2. $d(x, y) = m_{xy}\sigma$, iff $d(y, x) = m_{xy}\sigma$;
3. if $d(x, z) = m_{xz}\sigma$, $d(x, y) = m_{xy}\sigma$, $d(y, z) = m_{yz}\sigma$, then $m_{xz}\sigma \leq m_{xy}\sigma + m_{yz}\sigma$.

If M satisfies Σ , by Definition 3, the following holds:

$$M \models BPT(a, b) \text{ iff } \forall p_a \in I(a) \exists p_b \in I(b) : d(p_a, p_b) \in [0, \sigma];$$

$$M \models NEAR(a, b) \text{ iff } \exists p_a \in I(a) \exists p_b \in I(b) : d(p_a, p_b) \in [0, 2\sigma];$$

$M \models FAR(a, b)$ iff $\forall p_a \in I(a) \forall p_b \in I(b) : d(p_a, p_b) \in (4\sigma, \infty)$;

$M \models \neg\phi$ iff $M \not\models \phi$;

$M \models \phi \wedge \psi$ iff $M \models \phi$ and $M \models \psi$,

where a, b are individual names, ϕ, ψ are formulas in $L(LBPT)$.

By setting $\sigma = 1$, (Δ, d) is still a metric space, as the following holds for any $x, y, z \in \Delta$:

1. $d(x, y) = 0$ iff $x = y$;
2. $d(x, y) = m_{xy}$, iff $d(y, x) = m_{xy}$;
3. if $d(x, z) = m_{xz}$, $d(x, y) = m_{xy}$, $d(y, z) = m_{yz}$, then $m_{xz} \leq m_{xy} + m_{yz}$.

By setting $\sigma = 1$, the definitions of *BPT*, *NEAR* and *FAR* change accordingly as well. One can easily see that M still satisfies Σ after replacing every σ by 1.

Similar, if we have a metric model with $\sigma = 1$, we can obtain a metric model with $\sigma \in \mathbb{R}_{\geq 0}$ by multiplying every distance value m by σ . One can easily see that M still satisfies Σ after multiplying all distance values in M by σ , and multiplying all greatest lower bounds and least upper bounds of intervals in the truth definitions of *BPT*, *NEAR* and *FAR* by σ . \square

7. Validating Matches using Spatial Logic

The spatial logics *LNF*, *LNFS* and *LBPT* can be used to verify consistency of *sameAs* and *partOf* matches between spatial objects from different geospatial datasets. If every spatial object has a point geometry, then we apply *LNF*, otherwise, we apply *LNFS* or *LBPT*. *LBPT* reasoning has been used together with description logic reasoning in a geospatial data matching system MatchMaps (Du, Nguyen, Alechina, Logan, Jackson, & Goodwin, 2015; Du, 2015). *LBPT* reasoning and description logic reasoning complement each other in the sense that *LBPT* reasoning verifies matches regarding spatial information whilst description logic reasoning verifies matches regarding classification information, the unique name assumption and a stronger version of it. In the following, we describe how *LBPT* is used for debugging matches.

A dedicated *LBPT* reasoner integrated with an assumption-based truth maintenance system (ATMS) (de Kleer, 1986) was developed as a part of MatchMaps. It implements *LBPT* axioms and the definition $BEQ(a, b) \leftrightarrow BPT(a, b) \wedge BPT(b, a)$ as a set of inference rules. For efficiency reasons, there is no one-to-one correspondence between rules and axioms. To speed up matching and avoid cycles, facts such as *NEAR*(a, b) are stored for only one order of a and b , and symmetry axioms are removed. Each of the remaining axioms involving any symmetric relation gives rise to several rules, to compensate for the removal of symmetry. For example, the axiom

$$BPT(b, a) \wedge NEAR(b, c) \wedge BPT(c, d) \rightarrow \neg FAR(d, a)$$

also gives rise to a rule corresponding to the following implication:

$$BPT(b, a) \wedge NEAR(c, b) \wedge BPT(c, d) \rightarrow \neg FAR(d, a)$$

(with $NEAR(c, b)$ instead of $NEAR(b, c)$). However the set of rules is trivially equivalent to the set of axioms.

Possible matches of the form $sameAs(a, b)$ and $partOf(a, b)$ (a is $partOf$ b) are generated as *assumptions*, to be withdrawn if they are involved in a derivation of a contradiction in description logic or in LBPT. In order to apply the LBPT reasoning, $sameAs(a, b)$ is replaced by $BEQ(a, b)$, and $partOf(a, b)$ is replaced by $BPT(a, b)$. These substitutions do not affect correctness of the matching results of MatchMaps, as MatchMaps adopts the definitions of $sameAs$ and $partOf$ such that $sameAs(a, b)$ entails $BEQ(a, b)$ and $partOf(a, b)$ entails $BPT(a, b)$. $NEAR(a, b)$ and $FAR(a, b)$ facts are generated for those objects a, b in the same dataset that are involved in some matches across the two datasets (there is an object c in the other dataset such that $BEQ(a, c)$, $BPT(a, c)$ or $BPT(c, a)$ holds, and similarly for b).

The LBPT reasoner derives new formulas by applying inference rules to previously derived formulas, and the ATMS maintains dependencies between derived consequences and a set of *assumptions* (corresponding to possible matches). In particular, it maintains all minimal sets of assumptions responsible for the derivation of \perp (false), referred to as *nogoods* in ATMS terminology. Such minimal sets of assumptions responsible for a contradiction are used to decide which matches are wrong and should be withdrawn.

In experiments, the LBPT reasoner with an ATMS was used to validate matches between spatial objects from OSM data (building layer) and OSGB MasterMap data (Address Layer and Topology Layer) (Du et al., 2015). The study areas are in city centres of Nottingham UK and Southampton UK. The Nottingham data was obtained in 2012, and the Southampton data in 2013. The numbers of spatial objects in the case studies are shown in Table 1.

	OSM spatial objects	OSGB spatial objects
Nottingham	281	13204
Southampton	2130	7678

Table 1: Data used for Evaluation

The initial matches are generated by the matching method implemented in MatchMaps. The detailed matching method is provided by Du et al. (2016). The method consists of two main steps: matching geometries and matching spatial objects. A spatial object in a geospatial dataset has an ID, location information (coordinates and geometry) and meaningful labels, such as names or types, and represents an object in the real world. A geometry here refers to a point, a line or a polygon, which is used to represent location information in geospatial datasets.

The geometry matching requires a level of tolerance, as some difference in the geometry representation of a spatial object is to be expected in different datasets. After discussing with domain experts in geospatial science, we decided to apply the same level of tolerance for the matching method and the spatial logic used in MatchMaps. In the experiments described by Du et al. (2015), the level of tolerance for the geometry matching was set to be 20 meters, based on the ‘published estimate’ about the positional accuracy of OSM data. The OSM positional accuracy was estimated to be about 20 meters in UK (Haklay, 2010). In our more recent work (Du et al., 2016), we analysed how the level of tolerance affects

the precision and recall of matching results for the same geographic area in Nottingham (the same data as shown in the first row of Table 1) using 12 different levels of tolerance within a range of 1 to 80 meters. It shows that, for the Nottingham case, 20 meters is a good estimate, though it is not the optimal value.

Following the first step of the matching method, we first aggregate adjacent single geometries, such as those of shops within a shopping center, then establish correspondences between aggregated geometries using geometry matching. In the second step, we match spatial objects located on these corresponding aggregated geometries by comparing the similarity of names and types of spatial objects in several different cases (one-to-one, many-to-one and many-to-many). The most difficult case is when there is a match between two aggregated geometries which contain objects $\{a_1, \dots, a_n\}$ in one dataset and objects $\{b_1, \dots, b_k\}$ in the other dataset (many-to-many matching case). When we cannot decide the exact matches automatically using names and types of objects, we generate all matches which are possibly correct between the objects in the two sets: for each pair a_i, b_j with similar labels, we generate $sameAs(a_i, b_j)$, $partOf(a_i, b_j)$, $partOf(b_j, a_i)$. We apply reasoning in LBPT and description logic to verify consistency of these matches. The use of description logic reasoning is described by Du (2015).

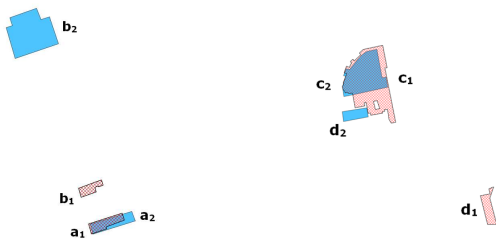


Figure 5: Examples of using LNFS and LBPT for validating matches

If a minimal set of statements involved in a contradiction contains more than one retractable assumption, a domain expert is needed to decide the correctness of the retractable assumptions and remove the wrong one(s) to restore consistency. Location information is visualized and provided to help domain experts make such decisions. As shown in Figure 5, a_1, b_1, c_1, d_1 (dotted) are from OSGB data and a_2, b_2, c_2, d_2 (solid) are from OSM data. In the left example, by LNFS Axiom 6 (or by LBPT Axiom 12 and $BEQ(a, b) \leftrightarrow BPT(a, b) \wedge BPT(b, a)$), a minimal set of statements for deriving an inconsistency consists of $BEQ(a_1, a_2)$, $BEQ(b_1, b_2)$, $NEAR(a_1, b_1)$, $FAR(a_2, b_2)$. It is clear that $BEQ(b_1, b_2)$ is wrong, as $NEAR(a_1, b_1)$ and $FAR(a_2, b_2)$ are facts. In the right example, $BPT(d_2, d_1)$ is wrong, because it contradicts $BPT(c_2, c_1)$, $NEAR(c_2, d_2)$, $FAR(c_1, d_1)$ by LBPT Axiom 12. As a consequence, the *sameAs* and *partOf* matches corresponding to $BEQ(b_1, b_2)$ and $BPT(d_2, d_1)$ respectively are also incorrect.

Table 2 shows the numbers of nogoods generated by the LBPT reasoner with an ATMS. As mentioned earlier, nogoods are justifications for ‘false’: the minimal sets of statements from which a contradiction is derivable. The number of interactions is the number of times users are asked to take actions or use strategies to resolve problems (a strategy is a heuristic which allows users to retract all ‘similar’ statements at a time, for example,

	nogoods	retracted BEQ/BPT	retracted <i>sameAs/partOf</i>	interactions
Nottingham	172	31	1325	3
Southampton	268	114	488	7

Table 2: LBPT Verification of Matches

retracting *partOf*(o, x) for any x differing from an object o). As a result of LBPT reasoning and removal of BEQ and BPT assumptions, we withdraw 1325 *sameAs/partOf* assumptions for the Nottingham case and 488 *sameAs/partOf* assumptions for the Southampton case. With the LBPT validation of matches, MatchMaps achieved high precision ($\geq 90\%$) and recall ($\geq 84\%$) for both Nottingham and Southampton cases.

As described in our previous work (Du, Alechina, Hart, & Jackson, 2015), MatchMaps was used by 12 experts from the University of Nottingham and Ordnance Survey of Great Britain to match about 100 buildings and places in Southampton. A graphical user interface of MatchMaps is provided allowing users to take different types of actions to remove wrong matches. The number of actions and the decision time of users are recorded. The precision and recall of the matching results are compared to those obtained without using any user-involved verification. Experimental results showed that by using reasoning in LBPT and description logic, the precision and recall of matches generated by MatchMaps were improved on average by 9% and 8% respectively. The human effort is also reduced, in the sense that the decision time required is much less than that of a fully manual matching process.

8. Discussion

The spatial logics LNF, LNFS and LBPT are generally applicable to reason with spatial objects whose locations are represented at different levels of accuracy or granularity in different datasets. Locations of spatial objects can be represented using vector data (coordinates) or raster data (images). Sometimes, for spatial objects in different datasets, measuring whether their locations are buffered equal directly is difficult or impossible, for example, when locations are represented as images without knowing their coordinates. In such cases, spatial objects may be matched by comparing shapes in images or using lexical information. No matter how the matches between spatial objects are generated, the LNF/LNFS/LBPT reasoning could be used to verify consistency of matches with regard to relative locations (*NEAR/FAR* facts) between spatial objects in the same dataset, which are often reliable and easy to capture.

Another potential application of these logics is in matching non-georeferenced volunteered spatial information or sketch data (Egenhofer, 1997; Kocczynski, 2006; Wallgrün, Wolter, & Richter, 2010). Instead of being created by surveying or other mapping techniques, sketch data is often created by a person from memory or by schematizing authoritative geospatial data. A sketch map cannot provide precise metric information such as the exact distance or size of a spatial object, but it roughly shows several kinds of qualitative relations (e.g. nearness and directions) between spatial objects. In the work by Wallgrün et al. (2010), qualitative spatial reasoning (based on the dipole relation algebra presented in Moratz, Renz, & Wolter, 2000 for checking connectivity and the cardinal direction calculus

presented in Ligozat, 1998) is used for the task of matching a sketch map of a road network to a larger geo-referenced data set, for example, from OpenStreetMap. Endpoints and junctions of roads are extracted and their relative directions are represented and checked by spatial reasoning. The spatial logic LNF can be applied similarly to check the relative distances between endpoints or junctions of roads. *NEAR/FAR* relations between these points indicate the length of roads. For the task of matching a sketch map of polygonal objects (e.g. buildings and places), the logic LNFS/LBPT can be applied. Suppose users draw a sketch map of buildings and estimate the distances between some buildings as being *NEAR* or *FAR* regarding an agreed level of tolerance σ . The *NEAR* and *FAR* relations between buildings in a geo-referenced map can be calculated automatically. A mapping between the sketch map and a geo-referenced map can be checked by reasoning in the logic LNFS/LBPT. For example, two buildings which are specified as being *FAR* in the sketch map cannot be matched to two buildings which are *NEAR* in the geo-referenced map.

The main limitation of the new spatial logics is that they require a level of tolerance σ and when using the logics, the value of σ is the same for spatial objects of different sizes and types (such as buildings, roads, rivers and lakes). For example, the margin of error used for cities should be larger than that for buildings. Ideally, the value of σ should vary by the size and type of the spatial object being checked. This motivates the development of new spatial logics to reason about the sizes and types of spatial objects, in addition to their relative locations.

In this paper, the theorems were all proved with respect to a metric space. However, models based on a metric space may not be realisable in a 2D Euclidean space, which is more realistic for geospatial data. Suppose there are four points p_i , where $i \in \{1, 2, 3, 4\}$. For each point p_i , $d(p_i, p_i) = 0$. For any pair of them, $d(p_i, p_j) = d(p_j, p_i) = 1$. It is clear that there is a metric space satisfying all the distance constraints, but there is no such 2D Euclidean space. Wolter and Zakharyashev (2003, 2005) proved that the satisfiability problem for a finite set of $MS(Q_{\geq 0})$ formulas in a 2D Euclidean space \mathbb{R}^2 is undecidable, whilst its proper fragments may be decidable. We proved that the satisfiability problem for a finite set of LNF formulas in a 2D Euclidean space is decidable in PSPACE (Du, 2015), but whether the satisfiability problem for a finite set of LNFS/LBPT formulas in a 2D Euclidean space is decidable is still unknown. It also remains open that whether the LNF/LNFS/LBPT calculus is complete for models based on a 2D Euclidean space. If not, a theoretical challenge is to design logics which are complete for 2D Euclidean spaces, and hence provide more accurate debugging of matches than the logics of metric spaces.

Finally, the use of description logic and the new spatial logics may not be able to detect all the wrong matches. For example, for spatial objects X, Y in one dataset and X', Y' in the other dataset, if $\text{sameAs}(X, X')$ is correct, Y is *NEAR* and to the south of X , and Y' is *NEAR* and to the north of X' , then $\text{sameAs}(Y, Y')$ is wrong but cannot be detected. To deal with this, we could extend the logics with existing spatial formalisms for reasoning about directional relations (Frank, 1991, 1996; Ligozat, 1998; Balbiani et al., 1999; Goyal & Egenhofer, 2001; Skiadopoulou & Koubarakis, 2004).

9. Conclusion and Future Work

We presented a series of new qualitative spatial logics LNF, LNFS and LBPT for validating matches between spatial objects, especially in crowd-sourced geospatial data. For models based on a metric space, a sound and complete axiomatisation is provided and corresponding theorems are proved for each logic. The LNF, LNFS and LBPT satisfiability problems in a metric space are all NP-complete. An LBPT reasoner with an ATMS was implemented and used as a part of MatchMaps. Experimental results show that the LBPT reasoner can be used to verify consistency of matches with respect to location information and detect obvious logical errors effectively. As future work, we will investigate whether the LNF/LNFS/LBPT calculus is complete for models based on a 2D Euclidean space and develop new spatial logics (e.g. for reasoning about directional relations and object sizes in addition to distances) to provide more accurate debugging of matches.

Acknowledgments

We would like to thank the anonymous reviewers who provided excellent comments that helped us improve the paper.

Appendix A. Proofs

Lemma 9 *If Σ^+ is an MCS, then for any pair of individual names a, b occurring in Σ , exactly one of the following cases holds in Σ^+ :*

1. $case(a, b) = BPT(a, b) \wedge BPT(b, a)$;
2. $case(a, b) = BPT(a, b) \wedge \neg BPT(b, a)$;
3. $case(a, b) = \neg BPT(a, b) \wedge BPT(b, a)$;
4. $case(a, b) = \neg BPT(a, b) \wedge \neg BPT(b, a) \wedge NEAR(a, b)$;
5. $case(a, b) = \neg NEAR(a, b) \wedge \neg FAR(a, b)$;
6. $case(a, b) = FAR(a, b)$,

where $case(a, b)$ denotes the formula which holds between a, b in each case.

Proof. For any pair of individual names a, b occurring in Σ^+ , we have:

$$\begin{aligned} \vdash & (B \wedge B^{-1} \wedge N \wedge F) \vee (B \wedge B^{-1} \wedge N \wedge \neg F) \vee (B \wedge B^{-1} \wedge \neg N \wedge F) \vee (B \wedge B^{-1} \wedge \neg N \wedge \neg F) \\ & \vee (B \wedge \neg B^{-1} \wedge N \wedge F) \vee (B \wedge \neg B^{-1} \wedge N \wedge \neg F) \vee (B \wedge \neg B^{-1} \wedge \neg N \wedge F) \vee (B \wedge \neg B^{-1} \wedge \neg N \wedge \neg F) \\ & \vee (\neg B \wedge B^{-1} \wedge N \wedge F) \vee (\neg B \wedge B^{-1} \wedge N \wedge \neg F) \vee (\neg B \wedge B^{-1} \wedge \neg N \wedge F) \vee (\neg B \wedge B^{-1} \wedge \neg N \wedge \neg F) \\ & \vee (\neg B \wedge \neg B^{-1} \wedge N \wedge F) \vee (\neg B \wedge \neg B^{-1} \wedge N \wedge \neg F) \vee (\neg B \wedge \neg B^{-1} \wedge \neg N \wedge F) \vee (\neg B \wedge \neg B^{-1} \wedge \neg N \wedge \neg F) \end{aligned}$$

where B, B^{-1}, N, F stand for $BPT(a, b), BPT(b, a), NEAR(a, b), FAR(a, b)$ respectively. From Table 3, we have

$$\vdash (B \wedge B^{-1}) \vee (B \wedge \neg B^{-1}) \vee (\neg B \wedge B^{-1}) \vee (\neg B \wedge \neg B^{-1} \wedge N) \vee (\neg N \wedge \neg F) \vee F.$$

B	B^{-1}	N	F	Prime Implicant	Axiom/Fact used
1	1	1	1	\perp	Fact 15
1	1	1	0	$B \wedge B^{-1}$	Facts 14, 15, Axiom 3
1	1	0	1	\perp	Fact 14, Axiom 3
1	1	0	0	\perp	Fact 14, Axiom 3
1	0	1	1	\perp	Fact 15
1	0	1	0	$B \wedge \neg B^{-1}$	Facts 14, 15, Axiom 3
1	0	0	1	\perp	Fact 14, Axiom 3
1	0	0	0	\perp	Fact 14, Axiom 3
0	1	1	1	\perp	Fact 15
0	1	1	0	$\neg B \wedge B^{-1}$	Facts 14, 15, Axiom 3
0	1	0	1	\perp	Fact 14, Axiom 3
0	1	0	0	\perp	Fact 14, Axiom 3
0	0	1	1	\perp	Fact 15
0	0	1	0	$\neg B \wedge \neg B^{-1} \wedge N$	Fact 15
0	0	0	1	F	Facts 14, 15, Axiom 3
0	0	0	0	$\neg N \wedge \neg F$	Fact 14, Axiom 3

Table 3: truth table, 1 stands for ‘true’, 0 stands for ‘false’

□

Lemma 11 *For any distance range g occurring in $D(\Sigma^+)$,*

$$g \in \{\{0\}, [0, \sigma], (\sigma, \infty), [0, 2\sigma], (2\sigma, \infty), (2\sigma, 4\sigma], (4\sigma, \infty), [0, \infty)\}.$$

Proof. Suppose p, q are constants and $d(p, q) \in g$ is in $D(\Sigma^+)$. Let us look at g in all different cases:

- $p = q$: by Definition 13, $g = \{0\}$.
- $p \neq q$:
 - $p \in \text{points}(a), q \in \text{points}(a)$, for some individual name a :
by Definition 13, $g = [0, \infty)$.
 - $p \in \text{points}(a), q \in \text{points}(b)$, for different individual names a, b :
by Lemma 9 and Definition 10, exactly one of following cases holds:
 - C1** $\{\chi(a, b, [0, \sigma]), \chi(b, a, [0, \sigma])\} \subseteq B(\Sigma^+)$
 - C2** $\{\chi(a, b, [0, \sigma]), \xi(b, a, (\sigma, \infty))\} \subseteq B(\Sigma^+)$
 - C3** $\{\xi(a, b, (\sigma, \infty)), \chi(b, a, [0, \sigma])\} \subseteq B(\Sigma^+)$
 - C4** $\{\xi(a, b, (\sigma, \infty)), \xi(b, a, (\sigma, \infty)), \exists(a, b, [0, 2\sigma]), \exists(b, a, [0, 2\sigma])\} \subseteq B(\Sigma^+)$
 - C5** $\{\forall(a, b, (2\sigma, \infty)), \forall(b, a, (2\sigma, \infty)), \exists(a, b, [0, 4\sigma]), \exists(b, a, [0, 4\sigma])\} \subseteq B(\Sigma^+)$
 - C6** $\{\forall(a, b, (4\sigma, \infty)), \forall(b, a, (4\sigma, \infty))\} \subseteq B(\Sigma^+)$

In **C1**:

- * if exactly one of p, q is a witness of $\chi(a, b, [0, \sigma])$ or $\chi(b, a, [0, \sigma])$, then by Definition 13, in the construction process, $d(p, q) \in [0, \sigma]$ will be added to

$D(\Sigma^+)$, and then $d(p, q) \in [0, +\infty)$ will be added to $D(\Sigma^+)$. Since $[0, \sigma] \cap [0, +\infty) = [0, \sigma]$, $g = [0, \sigma]$.

- * else if p is a witness of $\chi(b, a, [0, \sigma])$ and q is a witness of $\chi(a, b, [0, \sigma])$, then by Definition 13, in the construction process, $d(p, q) \in [0, \sigma]$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \sigma]$ will be added to $D(\Sigma^+)$ again, and then $d(p, q) \in [0, +\infty)$ will be added to $D(\Sigma^+)$. Since $[0, \sigma] \cap [0, \sigma] \cap [0, +\infty) = [0, \sigma]$, $g = [0, \sigma]$.
- * else, by Definition 13, $g = [0, +\infty)$.

In **C2**:

- * if q is a witness of $\chi(a, b, [0, \sigma])$, then by Definition 13, in the construction process, $d(p, q) \in [0, \sigma]$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $[0, \sigma] \cap [0, \infty) = [0, \sigma]$, $g = [0, \sigma]$.
- * else if q is a witness of $\xi(b, a, (\sigma, \infty))$, then by Definition 13, in the construction process, $d(p, q) \in (\sigma, \infty)$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $(\sigma, \infty) \cap [0, \infty) = (\sigma, \infty)$, $g = (\sigma, \infty)$.
- * else, by Definition 13, $g = [0, +\infty)$.

In **C3**:

- * if p is a witness of $\xi(a, b, (\sigma, \infty))$, then by Definition 13, in the construction process, $d(p, q) \in (\sigma, \infty)$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $(\sigma, \infty) \cap [0, \infty) = (\sigma, \infty)$, $g = (\sigma, \infty)$.
- * else if p is a witness of $\chi(b, a, [0, \sigma])$, then by Definition 13, in the construction process, $d(p, q) \in [0, \sigma]$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $[0, \sigma] \cap [0, \infty) = [0, \sigma]$, $g = [0, \sigma]$.
- * else, by Definition 13, $g = [0, +\infty)$.

In **C4**:

- * if the pair p, q is a witness of $\exists(a, b, [0, 2\sigma])$, then by Definition 13, in the construction process, $d(p, q) \in [0, 2\sigma]$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $[0, 2\sigma] \cap [0, \infty) = [0, 2\sigma]$, $g = [0, 2\sigma]$.
- * else if exactly one of p, q is a witness of $\xi(a, b, (\sigma, \infty))$ or $\xi(b, a, (\sigma, \infty))$, then by Definition 13, in the construction process, $d(p, q) \in (\sigma, \infty)$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $(\sigma, \infty) \cap [0, \infty) = (\sigma, \infty)$, $g = (\sigma, \infty)$.
- * else if p is a witness of $\xi(a, b, (\sigma, \infty))$ and q is a witness of $\xi(b, a, (\sigma, \infty))$, then by Definition 13, in the construction process, $d(p, q) \in (\sigma, \infty)$ will be added to $D(\Sigma^+)$, and then $d(p, q) \in (\sigma, \infty)$ will be added to $D(\Sigma^+)$ again, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $(\sigma, \infty) \cap (\sigma, \infty) \cap [0, \infty) = (\sigma, \infty)$, $g = (\sigma, \infty)$.
- * else, by Definition 13, $g = [0, +\infty)$.

In **C5**:

- * if the pair p, q is a witness of $\exists(a, b, [0, 4\sigma])$, then by Definition 13, in the construction process, $d(p, q) \in [0, 4\sigma]$ will be added to $D(\Sigma^+)$, and then,

$d(p, q) \in (2\sigma, \infty)$ will be added to satisfy the \forall formulas, and then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $[0, 4\sigma] \cap (2\sigma, \infty) \cap [0, \infty) = (2\sigma, 4\sigma]$, $g = (2\sigma, 4\sigma]$.

* else, by Definition 13, $d(p, q) \in (2\sigma, \infty)$ will be added to satisfy the \forall formulas, then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $(2\sigma, \infty) \cap [0, \infty) = (2\sigma, \infty)$, $g = (2\sigma, \infty)$.

In **C6**, by Definition 13, $d(p, q) \in (4\sigma, \infty)$ will be added, then $d(p, q) \in [0, \infty)$ will be added to $D(\Sigma^+)$. Since $(4\sigma, \infty) \cap [0, \infty) = (4\sigma, \infty)$, $g = (4\sigma, \infty)$.

Therefore, $g \in \{\{0\}, [0, \sigma], (\sigma, \infty), [0, 2\sigma], (2\sigma, \infty), (2\sigma, 4\sigma], (4\sigma, \infty), [0, \infty)\}$. \square

Lemma 16 (Calculation of Composition) *If (m, n) , (s, t) , (m, ∞) , (s, ∞) , $\{l\}$, $\{r\}$ are non-negative non-empty intervals, H_1 , H_2 , H are non-negative intervals, then the following calculation rules hold:*

1. $\{l\} \circ \{r\} = [l - r, l + r]$, if $l \geq r$;
2. $\{l\} \circ (s, t) = (s - l, t + l)$, if $s \geq l$;
3. $\{l\} \circ (s, t) = [0, t + l]$, if $l \in (s, t)$;
4. $\{l\} \circ (s, t) = (l - t, t + l)$, if $t \leq l$;
5. $\{l\} \circ (s, +\infty) = (s - l, +\infty)$, if $s \geq l$;
6. $\{l\} \circ (s, +\infty) = [0, +\infty)$, if $s < l$;
7. $(m, n) \circ (s, t) = (s - n, t + n)$, if $s \geq n$;
8. $(m, n) \circ (s, t) = [0, t + n)$, if $(m, n) \cap (s, t) \neq \emptyset$;
9. $(m, n) \circ (s, +\infty) = (s - n, +\infty)$, if $s \geq n$;
10. $(m, n) \circ (s, +\infty) = [0, +\infty)$, if $s < n$;
11. $(m, +\infty) \circ (s, +\infty) = [0, +\infty)$;
12. $H_1 \circ \emptyset = \emptyset$;
13. $H_1 \circ H_2 = H_2 \circ H_1$;
14. $(H_1 \cup H_2) \circ H = (H_1 \circ H) \cup (H_2 \circ H)$;
15. $(\bigcup_k H_k) \circ H = \bigcup_k (H_k \circ H)$, where $k \in \mathbb{N}_{>0}$;
16. $(H_1 \cap H_2) \circ H = (H_1 \circ H) \cap (H_2 \circ H)$, if $(H_1 \cap H_2) \neq \emptyset$;
17. $(H_1 \circ H_2) \circ H = H_1 \circ (H_2 \circ H)$.

Proof.[for Rule 15] Suppose $d \in (\bigcup_k H_k) \circ H$. By Lemma 15, there exist $d_1 \in \bigcup_k H_k$ and $d_2 \in H$ such that $d \in \{d_1\} \circ \{d_2\}$. Since $d_1 \in \bigcup_k H_k$, then there exists a $k \in \mathbb{N}_{>0}$ such that $d_1 \in H_k$. Since $d_1 \in H_k$ for some k , $d_2 \in H$, then by Definition 7, $d \in H_k \circ H$, for some k . Therefore $d \in \bigcup_k (H_k \circ H)$.

Suppose $d \in \bigcup_k (H_k \circ H)$. Then, there exists a $k \in \mathbb{N}_{>0}$ such that $d \in H_k \circ H$. Since $H_k \subseteq \bigcup_k H_k$, by Definition 7, $d \in (\bigcup_k H_k) \circ H$. \square

Proof.[for Rule 16] Suppose $H_1 \cap H_2 \neq \emptyset$. Then, $H_i \neq \emptyset$, $i \in \{1, 2\}$. Since H_1, H_2, H are non-negative intervals, by the intersection rules and Definition 7, $(H_1 \cap H_2) \circ H$ and $(H_1 \circ H) \cap (H_2 \circ H)$ are non-negative intervals. Let $L = (H_1 \cap H_2) \circ H$, $R = (H_1 \circ H) \cap (H_2 \circ H)$. If $H = \emptyset$, then by Rule 12, $L = R = \emptyset$; otherwise, we show $L = R$ by cases:

- $H_1 \supseteq H_2$ or $H_2 \supseteq H_1$: When $H_1 \supseteq H_2$, then Definition 7, $H_1 \circ H \supseteq H_2 \circ H$. $L = H_2 \circ H = R$. When $H_2 \supseteq H_1$, similarly, $L = H_1 \circ H = R$.
- $H_1 \not\supseteq H_2$ and $H_2 \not\supseteq H_1$: Without loss of generality, let us suppose $glb(H_1) \leq glb(H_2) \leq lub(H_1) \leq lub(H_2)$. Then, $glb(H_1 \cap H_2) = glb(H_2)$, $lub(H_1 \cap H_2) = lub(H_1)$.

To prove $L = R$, it is sufficient to show the following properties hold:

1. $lub(L) = lub(R)$;
2. $glb(L) = glb(R)$;
3. $lub(L) \in L$ iff $lub(R) \in R$;
4. $glb(L) \in L$ iff $glb(R) \in R$.

By Rules 1-14 and intersection rules, $lub(L) = lub(H_1 \cap H_2) + lub(H) = lub(H_1) + lub(H)$. $lub(R) = \min(lub(H_1) + lub(H), lub(H_2) + lub(H)) = lub(H_1) + lub(H)$. Thus, $lub(L) = lub(R)$ (Property 1 holds).

If $lub(H_1) \in H_1$ and $lub(H) \in H$, then by Rules 1-14 and intersection rules, $lub(L) \in L$ and $lub(R) \in R$; otherwise, $lub(L) \notin L$ and $lub(R) \notin R$. Thus, Property 3 holds.

We prove Property 2 and Property 4 by cases:

- $H \cap H_1 = \emptyset$ and $H \cap H_2 = \emptyset$:
 - * $lub(H) \leq glb(H_1)$:
 - $glb(L) = glb(H_1 \cap H_2) - lub(H) = glb(H_2) - lub(H)$.
 - $glb(R) = \max(glb(H_1) - lub(H), glb(H_2) - lub(H)) = glb(H_2) - lub(H)$.
 - Thus, $glb(L) = glb(R)$ (Property 2 holds).
 - If $glb(H_2) \in H_2$ and $lub(H) \in H$, then by Rules 1-14 and intersection rules, $glb(L) \in L$ and $glb(R) \in R$; otherwise, $glb(L) \notin L$ and $glb(R) \notin R$. Thus, Property 4 holds.
 - * $glb(H) \geq lub(H_2)$:
 - $glb(L) = glb(H) - lub(H_1 \cap H_2) = glb(H) - lub(H_1)$.
 - $glb(R) = \max(glb(H) - lub(H_1), glb(H) - lub(H_2)) = glb(H) - lub(H_1)$.
 - Similar to the case above, it is clear that Property 2 and Property 4 hold.
- $H \cap H_1 \neq \emptyset$ and $H \cap H_2 = \emptyset$: then, $H \cap (H_1 \cap H_2) = \emptyset$.
 - $glb(L) = glb(H_1 \cap H_2) - lub(H) = glb(H_2) - lub(H)$.

$$glb(R) = max(0, glb(H_2) - lub(H)) = glb(H_2) - lub(H).$$

Similar to the cases above, it is clear that Property 2 and Property 4 hold.

– $H \cap H_1 = \emptyset$ and $H \cap H_2 \neq \emptyset$: then, $H \cap (H_1 \cap H_2) = \emptyset$.

$$glb(L) = glb(H) - lub(H_1 \cap H_2) = glb(H) - lub(H_1).$$

$$glb(R) = max(glb(H) - lub(H_1), 0) = glb(H) - lub(H_1).$$

Similar to the cases above, it is clear that Property 2 and Property 4 hold.

– $H \cap H_1 \neq \emptyset$ and $H \cap H_2 \neq \emptyset$:

since H_1, H_2, H are intervals and $H_1 \cap H_2 \neq \emptyset$, $H \cap (H_1 \cap H_2) \neq \emptyset$.

$$glb(L) = 0.$$

$$glb(R) = max(0, 0) = 0.$$

By Rules 1-14, $glb(L) \in L$ and $glb(R) \in R$.

It is clear that Property 2 and Property 4 hold.

In every case, Properties 1-4 hold.

Therefore, $L = R$. \square

Proof.[for Rule 17] Let $L = (H_1 \circ H_2) \circ H$, $R = H_1 \circ (H_2 \circ H)$.

By Definition 7, $L = (\bigcup_{d_1 \in H_1, d_2 \in H_2} \{d_1\} \circ \{d_2\}) \circ H$.

By Rule 15, $L = \bigcup_{d_1 \in H_1, d_2 \in H_2} (\{d_1\} \circ \{d_2\}) \circ H$.

By Rule 13, $(\{d_1\} \circ \{d_2\}) \circ H = H \circ (\{d_1\} \circ \{d_2\})$.

By Rule 15, $H \circ (\{d_1\} \circ \{d_2\}) = \bigcup_{d \in H} (\{d\} \circ (\{d_1\} \circ \{d_2\}))$.

By Rule 13, $L = \bigcup_{d_1 \in H_1, d_2 \in H_2, d \in H} (\{d_1\} \circ \{d_2\}) \circ \{d\}$.

Similarly, $R = \bigcup_{d_1 \in H_1, d_2 \in H_2, d \in H} (\{d_1\} \circ (\{d_2\} \circ \{d\}))$.

To prove $L = R$, it is sufficient to show

$$(\{d_1\} \circ \{d_2\}) \circ \{d\} = \{d_1\} \circ (\{d_2\} \circ \{d\}).$$

Let $l = (\{d_1\} \circ \{d_2\}) \circ \{d\}$, then $l = [|d_1 - d_2|, d_1 + d_2] \circ \{d\}$;

$r = \{d_1\} \circ (\{d_2\} \circ \{d\})$, then $r = \{d_1\} \circ [|d_2 - d|, d_2 + d]$.

We prove $l = r$ by cases:

- $d \in [|d_1 - d_2|, d_1 + d_2]$: By Definition 7, $l = [0, d_1 + d_2 + d]$.

$$d_1 + d_2 \geq d, d_2 + d \geq d_1, d_1 + d \geq d_2.$$

Thus, $d_1 \in [|d_2 - d|, d_2 + d]$. By Definition 7, $r = [0, d_1 + d_2 + d]$.

- $d \notin [|d_1 - d_2|, d_1 + d_2]$:

– $d > d_1 + d_2$: By Definition 7, $l = [d - d_1 - d_2, d_1 + d_2 + d]$.

$$d_1 < d - d_2 = |d_2 - d|.$$

By Definition 7, $r = [d - d_2 - d_1, d_1 + d_2 + d]$.

– $d < |d_1 - d_2|$: By Definition 7, $l = [|d_1 - d_2| - d, d_1 + d_2 + d]$.

* $d_1 \geq d_2$: $d < d_1 - d_2$, this is, $d_1 > d_2 + d$.

By Definition 7, $r = [d_1 - d_2 - d, d_1 + d_2 + d]$.

* $d_1 < d_2$: $d < d_2 - d_1$, this is, $d_1 < d_2 - d$.

By Definition 7, $r = [d_2 - d - d_1, d_1 + d_2 + d]$.

In each case, $l = r$. Therefore, $L = R$. \square

Appendix B. Alternative Proof of the Path-Consistency Lemma

In this appendix, we would like to provide a sketch of an alternative proof idea of the Path-Consistency Lemma, since it may appeal to some of the readers more than the proof presented in Section 5.3. The alternative proof uses Lemma 44.

Lemma 44 *If a distance constraint $d(p, q) \in h$ is in $DS(\Sigma^+)$ and $h \neq \emptyset$, then there exist $d(p, q) \in m_1$ and $d(p, q) \in m_2$ in $DS(\Sigma^+)$ such that $h = m_1 \cap m_2$, m_1 and m_2 are both identity or definable intervals.*

Proof. By Lemma 31, $d(p, q) \in h$ is left-definable and right-definable. By Definition 16, there exists a sequence of distance constraints $d(p_i, p_{i+1}) \in h_i$ ($p_1 = p, p_n = q, 0 < i < n$) in $D(\Sigma^+)$, such that for $m_1 = h_1 \circ \dots \circ h_{n-1}$, $h \subseteq m_1$, h and m_1 have the same greatest lower bound (both value and openness). By Definition 15, $d(p, q) \in m_1$ is in $DS(\Sigma^+)$. By Lemma 21 and Definition 14, m_1 is an identity or definable interval. Similarly, by Definition 17, there exists an m_2 such that $h \subseteq m_2$, h and m_2 have the same least upper bound (both value and openness), $d(p, q) \in m_2$ is in $DS(\Sigma^+)$, m_2 is an identity or definable interval. By intersection rules, $h = m_1 \cap m_2$. \square

Proof.[sketch of the alternative proof of the Path-Consistency Lemma] Suppose $D(\Sigma^+)$ is not path-consistent. Then there exist $d(p, q) \in g_1$ and $d(p, q) \in g_2$ in $DS(\Sigma^+)$, $g_1 \neq \emptyset$, $g_2 \neq \emptyset$, and $g_1 \cap g_2 = \emptyset$. By Lemma 44, there exist $d(p, q) \in m_1$ and $d(p, q) \in s_1$ in $DS(\Sigma^+)$ such that $g_1 = m_1 \cap s_1$, m_1 and s_1 are both identity or definable intervals. Similarly, we have $g_2 = m_2 \cap s_2$, where m_2 and s_2 are both identity or definable intervals. $g_1 \cap g_2 = \emptyset$ holds iff one of the following holds: $m_1 \cap m_2 = \emptyset$, $m_1 \cap s_2 = \emptyset$, $s_1 \cap m_2 = \emptyset$, $s_1 \cap s_2 = \emptyset$. Without loss of generality, let us suppose $m_1 \cap m_2 = \emptyset$. By Lemma 35, $m_1 \cap m_2 = \emptyset$ iff $(m_1 \circ m_2) \cap \{0\} = \emptyset$. Let $m = m_1 \circ m_2$. Then m is an identity or definable interval. Since $d(p, q) \in m_1$ and $d(p, q) \in m_2$ are in $DS(\Sigma^+)$, $m_1 \neq \emptyset$, $m_2 \neq \emptyset$, then by Lemma 31, $d(p, q) \in m_1$ and $d(p, q) \in m_2$ are left-definable and right-definable. Since $d(p, q) \in m_2$, we have $d(q, p) \in m_2$. By Lemma 29, $d(p, p) \in m$ is left-definable and right-definable. The rest of the proof is almost the same as the proof of the Path-Consistency Lemma (starting from ‘By Lemma 23, $glb(h') \in \{0, \sigma, 2\sigma, 3\sigma, 4\sigma\}$ ’) presented in Section 5.3. We discuss different ways to obtain m given its greatest lower bound (the role of m is similar to h') and check whether \perp can be derived in every valid case. \square

Appendix C. Consequences of the Path-Consistency Lemma

In this appendix, we state explicitly some implications of the Path-Consistency Lemma.

Lemma 45 *Let Σ be a finite consistent set of formulas. If a distance constraint $d(p, q) \in h$ is in $DS(\Sigma^+)$, then $h \neq \emptyset$.*

Proof. Follows immediately from the proof of the Path-Consistency Lemma. \square

Lemma 46 *Let Σ be a finite consistent set of formulas. If a distance constraint $d(p, p) \in h$ is in $DS(\Sigma^+)$, then $0 \in h$.*

Proof. Suppose a distance constraint $d(p, p) \in h$ is in $DS(\Sigma^+)$ and $0 \notin h$. By Definition 13 and Definition 15, $d(p, p) \in \{0\}$ is in $D(\Sigma^+)$. By Definition 15, $d(p, p) \in (h \cap \{0\}) = \emptyset$. This contradicts the fact that $d(p, p) \in \emptyset$ is not in $DS(\Sigma^+)$ (by Lemma 45). Therefore, $0 \in h$. \square

References

- Aiello, M., Pratt-Hartmann, I., & van Benthem, J. (Eds.). (2007). *Handbook of Spatial Logics*. Springer.
- Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11), 832–843.
- Balbiani, P., Condotta, J., & del Cerro, L. F. (1999). A New Tractable Subclass of the Rectangle Algebra. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 442–447.
- Bennett, B. (1996). The Application of Qualitative Spatial Reasoning to GIS. In *Proceedings of the 1st International Conference on GeoComputation*, Vol. I, pp. 44–47.
- Bennett, B., Cohn, A. G., & Isli, A. (1997). A Logical Approach to Incorporating Qualitative Spatial Reasoning into GIS (Extended Abstract). In *Proceedings of the 3rd International Conference on Spatial Information Theory*, Vol. 1329 of *Lecture Notes in Computer Science*, pp. 503–504. Springer.
- Chen, J., Cohn, A. G., Liu, D., Wang, S., OuYang, J., & Yu, Q. (2015). A survey of qualitative spatial representations. *The Knowledge Engineering Review*, 30(1), 106–136.
- Clementini, E., & Felice, P. D. (1996). An algebraic model for spatial objects with indeterminate boundaries. In *Proceedings of the GISDATA specialist meeting on Geographic Objects with Undeterminate Boundaries*, pp. 155–169.
- Clementini, E., & Felice, P. D. (1997). Approximate Topological Relations. *International Journal of Approximate Reasoning*, 16(2), 173–204.
- Clementini, E., Felice, P. D., & Hernández, D. (1997). Qualitative Representation of Positional Information. *Artificial Intelligence*, 95(2), 317–356.
- Cohn, A. G., & Gotts, N. M. (1996a). Representing Spatial Vagueness: A Mereological Approach. In *Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 230–241.
- Cohn, A. G., & Gotts, N. M. (1996b). The ‘Egg-Yolk’ Representation of Regions with Indeterminate Boundaries. In *Proceedings of the GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries*, pp. 171–187.
- Cohn, A. G., & Renz, J. (2008). Qualitative Spatial Representation and Reasoning. In *Handbook of Knowledge Representation*, pp. 551–596. Elsevier.
- de Kleer, J. (1986). An assumption-based TMS. *Artificial Intelligence*, 28(2), 127–162.
- Du, H. (2015). *Matching Disparate Geospatial Datasets and Validating Matches using Spatial Logic*. Ph.D. thesis, School of Computer Science, University of Nottingham, UK.

- Du, H., & Alechina, N. (2014a). A Logic of Part and Whole for Buffered Geometries. In *Proceedings of the 21st European Conference on Artificial Intelligence*, pp. 997–998.
- Du, H., & Alechina, N. (2014b). A Logic of Part and Whole for Buffered Geometries. In *Proceedings of the 7th European Starting AI Researcher Symposium*, pp. 91–100.
- Du, H., Alechina, N., Hart, G., & Jackson, M. (2015). A Tool for Matching Crowd-sourced and Authoritative Geospatial Data. In *Proceedings of the International Conference on Military Communications and Information Systems*, pp. 1–8. IEEE.
- Du, H., Alechina, N., Jackson, M., & Hart, G. (2016). A Method for Matching Crowd-sourced and Authoritative Geospatial Data. *Transactions in GIS*. <http://dx.doi.org/10.1111/tgis.12210>.
- Du, H., Alechina, N., Stock, K., & Jackson, M. (2013). The Logic of NEAR and FAR. In *Proceedings of the 11th International Conference on Spatial Information Theory*, Vol. 8116 of *Lecture Notes in Computer Science*, pp. 475–494. Springer.
- Du, H., Nguyen, H., Alechina, N., Logan, B., Jackson, M., & Goodwin, J. (2015). Using Qualitative Spatial Logic for Validating Crowd-Sourced Geospatial Data. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (the 27th Conference on Innovative Applications of Artificial Intelligence)*, pp. 3948–3953.
- Egenhofer, M. J. (1997). Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, 8(4), 403–424.
- Egenhofer, M. J., & Franzosa, R. D. (1991). Point Set Topological Spatial Relations. *International Journal of Geographical Information Systems*, 5(2), 161–174.
- Egenhofer, M. J., & Herring, J. R. (1991). Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. Tech. rep., University of Maine.
- Fine, K. (1975). Vagueness, truth and logic. *Synthèse*, 30, 263–300.
- Frank, A. U. (1991). Qualitative Spatial Reasoning with Cardinal Directions. In *Proceedings of the 7th Austrian Conference on Artificial Intelligence*, pp. 157–167.
- Frank, A. U. (1996). Qualitative Spatial Reasoning: Cardinal Directions as an Example. *International Journal of Geographical Information Science*, 10(3), 269–290.
- Goyal, R. K., & Egenhofer, M. J. (2001). Similarity of Cardinal Directions. In Jensen, C. S., Schneider, M., Seeger, B., & Tsotras, V. J. (Eds.), *Advances in Spatial and Temporal Databases*, Vol. 2121 of *Lecture Notes in Computer Science*, pp. 36–55. Springer.
- Guesgen, H. W., & Albrecht, J. (2000). Imprecise reasoning in geographic information systems. *Fuzzy Sets and Systems*, 113(1), 121–131.
- Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design*, 37(4), 682–703.
- ISO Technical Committee 211 (2003). ISO 19107:2003 Geographic information – Spatial schema. Tech. rep., International Organization for Standardization (TC 211).

- Jackson, M., Rahemtulla, H., & Morley, J. (2010). The synergistic use of authenticated and crowd-Sourced data for emergency response. In *Proceedings of the 2nd International Workshop on validation of GeoInformation products for crisis management*, pp. 91–99.
- Kopczynski, M. (2006). Efficient spatial queries with sketches. In *Proceedings of the ISPRS Technical Commission II Symposium*, pp. 19–24.
- Kutz, O. (2007). Notes on Logics of Metric Spaces. *Studia Logica*, 85(1), 75–104.
- Kutz, O., Sturm, H., Suzuki, N., Wolter, F., & Zakharyashev, M. (2002). Axiomatizing Distance Logics. *Journal of Applied Non-Classical Logics*, 12(3-4), 425–440.
- Kutz, O., Wolter, F., Sturm, H., Suzuki, N., & Zakharyashev, M. (2003). Logics of metric spaces. *ACM Transactions on Computational Logic*, 4(2), 260–294.
- Lehmann, F., & Cohn, A. G. (1994). The EGG/YOLK Reliability Hierarchy: Semantic Data Integration Using Sorts with Prototypes. In *Proceedings of the 3rd International Conference on Information and Knowledge Management*, pp. 272–279.
- Li, S., Liu, W., & Wang, S. (2013). Qualitative constraint satisfaction problems: An extended framework with landmarks. *Artificial Intelligence*, 201, 32–58.
- Li, S., Long, Z., Liu, W., Duckham, M., & Both, A. (2015). On redundant topological constraints. *Artificial Intelligence*, 225, 51–76.
- Ligozat, G. . (1998). Reasoning about Cardinal Directions. *Journal of Visual Languages & Computing*, 9(1), 23–44.
- Lutz, C., & Milicic, M. (2007). A Tableau Algorithm for Description Logics with Concrete Domains and General TBoxes. *Journal of Automated Reasoning*, 38(1-3), 227–259.
- Mackworth, A. K., & Freuder, E. C. (1985). The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems. *Artificial Intelligence*, 25(1), 65–74.
- Mallenby, D. (2007). Grounding a Geographic Ontology on Geographic Data. In *AAAI Spring Symposium - Logical Formalizations of Commonsense Reasoning*, pp. 101–106.
- Mallenby, D., & Bennett, B. (2007). Applying Spatial Reasoning to Topographical Data with a Grounded Ontology. In *Proceedings of the 2nd International Conference GeoSpatial Semantics*, No. 4853 in Lecture Notes in Computer Science, pp. 210–227. Springer.
- Moratz, R., Renz, J., & Wolter, D. (2000). Qualitative Spatial Reasoning about Line Segments. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pp. 234–238.
- Moratz, R., & Wallgrün, J. O. (2012). Spatial reasoning with augmented points: Extending cardinal directions with local distances. *Journal of Spatial Information Science*, 5(1), 1–30.
- OpenStreetMap (2012). The Free Wiki World Map. <http://www.openstreetmap.org>.
- Ordnance Survey (2012). Ordnance Survey. <http://www.ordnancesurvey.co.uk>.
- Pawlak, Z., Polkowski, L., & Skowron, A. (2007). Rough Set Theory. In *Wiley Encyclopedia of Computer Science and Engineering*. John Wiley & Sons, Inc.

- Randell, D. A., Cui, Z., & Cohn, A. G. (1992). A Spatial Logic based on Regions and Connection. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pp. 165–176.
- Renz, J., & Nebel, B. (2007). Qualitative Spatial Reasoning Using Constraint Calculi. In Aiello, M., Pratt-Hartmann, I., & van Benthem, J. (Eds.), *Handbook of Spatial Logics*, pp. 161–215. Springer.
- Roy, A. J., & Stell, J. G. (2001). Spatial Relations between Indeterminate Regions. *International Journal of Approximate Reasoning*, 27(3), 205–234.
- Schockaert, S., Cock, M. D., Cornelis, C., & Kerre, E. E. (2008a). Fuzzy region connection calculus: An interpretation based on closeness. *International Journal of Approximate Reasoning*, 48(1), 332–347.
- Schockaert, S., Cock, M. D., Cornelis, C., & Kerre, E. E. (2008b). Fuzzy region connection calculus: Representing vague topological information. *International Journal of Approximate Reasoning*, 48(1), 314–331.
- Schockaert, S., Cock, M. D., & Kerre, E. E. (2009). Spatial reasoning in a fuzzy region connection calculus. *Artificial Intelligence*, 173(2), 258–298.
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2), 51–53.
- Skiadopoulos, S., & Koubarakis, M. (2004). Composing cardinal direction relations. *Artificial Intelligence*, 152(2), 143–171.
- Smith, N. J. (2008). *Vagueness and Degrees of Truth*. Oxford University Press.
- Stocker, M., & Sirin, E. (2009). PelletSpatial: A Hybrid RCC-8 and RDF/OWL Reasoning and Query Engine. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions*.
- Sturm, H., Suzuki, N., Wolter, F., & Zakharyashev, M. (2000). Semi-qualitative Reasoning about Distances: A Preliminary Report. In *Proceedings of the Logics in Artificial Intelligence, European Workshop, JELIA*, pp. 37–56.
- van Beek, P. (1992). Reasoning About Qualitative Temporal Information. *Artificial Intelligence*, 58(1-3), 297–326.
- Wallgrün, J. O., Wolter, D., & Richter, K. (2010). Qualitative matching of spatial information. In *Proceedings of the 18th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems*, pp. 300–309.
- Wolter, F., & Zakharyashev, M. (2003). Reasoning about Distances. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 1275–1282.
- Wolter, F., & Zakharyashev, M. (2005). A logic for metric and topology. *Journal of Symbolic Logic*, 70(3), 795–828.
- Zadeh, L. A. (1975). Fuzzy logic and approximate reasoning. *Synthese*, 30(3-4), 407–428.
- Zimmermann, K. (1995). Measuring without Measures: The Delta-Calculus. In *Proceedings of the 2nd International Conference on Spatial Information Theory*, pp. 59–67.