# Semantic Compression with Region Calculi in Nested Hierarchical Grids

Joseph Zalewski
Pascal Hitzler
Department of Computer Science, Kansas State University
Manhattan, Kansas, USA

Krzysztof Janowicz
Geography Department, University of California, Santa Barbara
Santa Barbara, California, USA

## ABSTRACT

We propose the combining of region connection calculi with nested hierarchical grids for representing spatial region data in the context of knowledge graphs, thereby avoiding reliance on vector representations. We present a resulting region calculus, and provide qualitative and formal evidence that this representation can be favorable with large data volumes in the context of knowledge graphs; in particular we study means of efficiently choosing which triples to store to minimize space requirements when data is represented this way, and we provide an algorithm for finding the smallest possible set of triples for this purpose including an asymptotic measure of the size of this set for a special case. We prove that a known constraint calculus is adequate for the reconstruction of all triples describing a region from such a pruned representation, but problematic for reasoning with hierarchical grids in general.

## CCS CONCEPTS

• **Information systems → Geographic information systems**; • **Theory of computation → Logic**; • **Mathematics of computing** → *Trees*.

## KEYWORDS

RCC5, hierarchical grids, knowledge graphs

**ACM Reference Format:**
Joseph Zalewski, Pascal Hitzler, and Krzysztof Janowicz. 2021. Semantic Compression with Region Calculi in Nested Hierarchical Grids. In *29th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '21), November 2–5, 2021, Beijing, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3474717.3483965

## 1 INTRODUCTION

In traditional geographic information systems (GIS), geographic data, such as the locations of region boundaries, are stored using one of a variety of techniques based on coordinate geometry. Vertices of polygons, etc. are points in a continuous space, represented by their real-number coordinates in some coordinate system. An alternative to this is the use of so-called hierarchical grids, where "space" is subdivided into hierarchically-nested "cells", whose exact geometries can be easily computed due to their regular nature. Hierarchical grids are already long in use in GIS, being used as index structures to speed up lookup of points and objects, stored as coordinates [12]. Recently hierarchical grid systems have been employed very successfully by companies such as Google [2] and Uber [3] to structure large quantities of their internal data. While in these applications there is a strong emphasis on indexing and efficient look-up using the index, we see another advantage to hierarchical grids that has not yet been systematically explored, namely that they lend themselves naturally to a context in which *knowledge graphs* are used for data integration and management.

Knowledge graphs are an approach to structuring data (or metadata[1]) in form of a labeled and typed graph, together with a type logic that is often referred to as a knowledge graph *schema* or an *ontology* [5]. Knowledge graphs have recently seen significant uptake by industry, with visible success [9]. The World Wide Web Consortium (W3C) has developed standards for knowledge graphs and their schemas – the Web Ontology Language OWL and the Resource Description Framework RDF – as well as the SPARQL querying language and other relevant standards, that are widely used [6]. The schema, if expressed in OWL, consists of a set of logical formulas that can be used for deductive inference if desired.

We argue that hierarchical grids are a natural choice for representing information about spatial regions, for many contexts: Each grid cell thus becomes a node in the knowledge graph, with relations between the cells (or relations between cells and features or information of interest) represented naturally by labelled graph edges. Collections of cells can be used to approximate regions of interest (e.g., by representing them with a suitable cover), thus trading some representational precision for increased querying and data processing speed. In a data integration context – in which knowledge graphs are prominently used – a chosen hierarchical grid can serve as the central integration anchor for spatial data originating from different formats, thus providing a uniform representation that can be tapped into, e.g. by visualization tools and geographical information systems.

Furthermore, type logics that provide schema information for knowledge graphs can naturally be used to capture logic-based calculi about spatial relations between regions, such as variants of the Region Connection Calculus RCC [10]. The formal logic of the region calculus and the formal logic of the knowledge graph schema then naturally combine and can be utilized for joint logical inferencing, i.e., for deducing knowledge that arises as necessary

*SIGSPATIAL '21, November 2–5, 2021, Beijing, China*

ACM ISBN 978-1-4503-8664-7/21/11.
https://doi.org/10.1145/3474717.3483965

[1] In a knowledge graph context, the boundary between metadata and data is – deliberately – not crisp. But what is referred to as "data" in a knowledge graph context would often be called "metadata" in different contexts.

logical consequences from the data and type logic of the knowledge graph, and can for example be used for querying for logically implied, but not explicitly encoded, information.

Another interesting aspect of the combined region and type logic is that it can be utilized for what has been called *semantic compression* in the context of (RDF) knowledge graphs [7]. It refers to the idea of using logical deduction rules to compress a knowledge graph without loss of information. In some situations, for example, addition of a single suitable logical formula to the type logic can make a very large number of node-edge-node graph triples redundant in the sense that they can now be removed, while at the same time the new logical formula makes it possible to re-generate the removed triples as needed.

This paper is a short paper that serves as an extended abstract to the contributions in the extended technical report [15] that contains all formal definitions, results and proofs referred to herein.

## 2 REGION CONNECTION CALCULUS ON THE GRID

We assume that the reader is familiar with basic set-theoretic topology, see e.g. [8] and also with the basics of formal logic, see e.g. [14].

We provide a topological definition of a hierarchical grid, which in particular applies to the square, quadtree, etc. grids often used in practice, e.g. the Google S2 grid [2]. Note that it does not apply to grid systems like Uber's H3 [3] in which child cells may not be fully contained in their parent.

*Definition 2.1.* Let a *nested hierarchical grid* be a pair $(A, \mathsf{cells}_A)$ where $A$ is a topological space, and $\mathsf{cells}_A$ is a tree with root $A$ and in which every node $N$ is a nonempty topological space, and if it has children, it has finitely many, but at least two, and the children $N_i$ of $N$ are regular closed subspaces of $N$ such that $\bigcup_i N_i = N$ and no two $N_i$ share an open subset. Additionally, for this paper we will require $A$ to be a Baire space, i.e., such that countable unions of degenerate subspaces are degenerate, which is a very mild condition given usual application scenarios for grids. In fact we really only need the condition that a finite union of degenerate sets is degenerate, but so many common spaces are Baire spaces that the distinction is not too important. For a tree $T$, we will use $|T|$ to denote the set of all nodes of $T$.

For the region calculus, we will focus on RCC5 (background in e.g. [13]) which is a logic of relations between regions with five predicates, $\mathsf{EQ}, \mathsf{PP}, \mathsf{PP}^{-1}, \mathsf{DR}, \mathsf{PO}$, which can be read, "equal, proper part of, properly containing, not significantly overlapping, partially overlapping". We use a topological semantics for RCC5, but there are several different semantics possible which give the same consequences. RCC5 is a popular "constraint calculus" used in GIS database systems [13]. We provide a slightly stronger variant of RCC5, which takes into account a priori *all* information about the structure of a hierarchical grid, not just the part of that information which is expressible in RCC5, by explicitly including a hierarchical grid in the logic's signature. We call this variant "RCC5-G."

It is usual in many geodatabase systems to use not RCC5 but the more powerful calculus RCC8. Both arise from the system RCC introduced in the paper [10]. RCC8 also has both a topological semantics and an a priori one (the original, from [10]); a discussion of topological semantics for RCC8 can be found in [11]. We use RCC5 here not just to simplify presentation, but because we believe that for our present purposes, RCC8 is actually unnecessary. Recall the approach to geometries which motivates this paper: geometries are to be considered only insofar as they can be captured by relationships to a fixed grid. The principal difference between the two RCC constraint formalisms is RCC8's concern with boundaries – it differentiates, for example, between the true disconnectedness relation and the "external connection" relation, in which two regions overlap, but in a degenerate set (with empty interior). RCC5 considers these situations to be indistinguishable (they can both provide the semantics for the predicate DR). While in traditional geometry representation schemes, the additional information about boundary relationships can be useful, our position is that for grid representation it is not. For, real-world data (locations of boundaries) should be thought of as sampled from a continuous distribution, and so it is vanishingly unlikely that a real-world boundary will ever exactly coincide with the finitely many artificial boundaries of our grid cells. Even real-world boundaries defined as straight lines, such as latitudes, will not usually line up with hierarchical grid cells, unless the grid is specifically planned out to make this happen, which they often are not. Whenever it seems to happen in real data, exact coincidence of boundaries should by default be attributed to insufficient decimal precision, rather than assumed to have real meaning. This assumption is convenient for us, as RCC8 is not nearly as compatible with a hierarchical organization of space as RCC5 is, and obtaining results for it like those in the following sections is much harder.

### 2.1 Inheritance

While in this paper we will concentrate on the use of RCC5-G to reduce sets of relations between the grid and a single region, we will briefly mention a more standard use of such calculi – to store information about properties of regions that are "upward-inherited" or "downward-inherited." By downward-inherited we mean a property which, if possessed by a region $R$, also characterizes all regions containing $R$. Such a property can be represented by the collection of all maximal regions having it, and checked by checking whether a region $R$ is $\mathsf{EQ}$ or $\mathsf{PP}$ to one of these – that is, whether a satisfiable constraint network exists in which an edge from $R$ to one such region is labeled $\{\mathsf{EQ}, \mathsf{PP}\}$. Often there will only be one maximal region needed, as for the property "completely covered by water." Common types of upward-inherited properties involve "containing a feature", such as a particular city, or containing some part of a distributed entity, such as "water". These can be represented by a $\{\mathsf{PP}^{-1}, \mathsf{EQ}\}$ (in the first case) or $\{\mathsf{PO}, \mathsf{PP}^{-1}, \mathsf{EQ}\}$ (in the second) relation to a region, i.e. the region occupied by the city, or the region covered by water.

## 3 SEMANTIC COMPRESSION

We have already argued in the introduction that the use of hierarchical grids together with knowledge graphs, as described herein, provides some advantages in some circumstances. It is important to note, however, that that there are always trade-offs, and that a particular representational form (such as using a hierarchical grid) is advantageous in some use cases, and not so in others. Our approach provides *additional flexibility* in making a choice for representing spatial information in the context of knowledge graphs.

Using a hierarchical grid as described is an approximation for spatial representation that is constrained by the pre-defined grid cells. As such, it comes at the loss of some precision. However, it also comes with some advantages. One of them is representational simplicity. Rather than representing each region with, say, a polygon in the graph, the spatial representation of the regions becomes normalized as a selection of cells that have some specified region-connection relationship to the region. By taking the hierarchical structure (and corresponding logical axiomatization) into account, it is in fact *not* necessary to flag all such cells, as covering a region inherits upwards and containment within a region inherits downwards. We can thus arrive at a *semantically compressed* representation.

In a similar vein, not only region representation can be understood as semantically compressed, but relevant features of such a region can likewise be represented in compressed form by making use of the logic from Section 2, in particular upward and downward inheritance as discussed. E.g., if a cell is known to fully fall within a region with arid climate, then we know that arid climate also applies for all its sub-cells. In particular in the context of knowledge graphs, where information pertaining to many different regions may be abundant, this type of reasoning over the grid may result in a cleaner representation of content.

Another possible advantage of using hierarchical grids for knowledge graphs with spatial content is for the information integration process itself; indeed knowledge graphs excel as a tool for information integration from heterogeneous sources. Using a hierarchical grid, spatial information from a data source can be *normalized* by expressing it approximately using cells, thus providing a convenient format for the integrated representation, while at the same time providing a simplified logic for reasoning about spatial relations and inheritance of features as just discussed. Once cast into this form, it is no longer necessary to compute region intersections etc. from, say, vector representations, or to deal with the complexities of a region calculus on arbitrarily shaped regions: Instead we have arrived at a compressed representation with a much simpler logic.

We consider how to arrive at a compressed representation, in terms of cells, of a single region, about which we know as much as our grid-based representation of geometry can tell us, in a vacuum, so to speak – our only option to record information about it is by RCC5 relations directly with cells, not with other regions. See the Further Work section for other similar problems we may want to solve. The *full-knowledge single region compression* problem is to find a small set of formulas $B$ (in RCC5-G) which is logically equivalent to the set of *all* RCC5-G formulas describing a given region (without reference to any other non-cell regions). Now in general the effectiveness of compression is hard to neatly quantify in a provable way, but in this case we can get a very nice fact – that there is an optimal solution, up to a constant discrepancy. This optimal solution is intuitively, not to say trivially, obvious to anyone who can visualize a square grid: cells which are fully inside or fully outside the region $R$ ought to be conglomerated together as much as possible in $B$, since decomposing them into smaller cells adds no further information about where $R$ is. Hierarchical grid libraries often contain functions to perform this kind of compression (see e.g. "compact" in H3). Note that without loss of generality we can consider only formulas of the type $P(d, R)$, since every formula $P(R, d)$ is equivalent to one of this form. Indeed, thinking and writing about the correctness of the optimal solution is very cumbersome if we keep using this predicate notation, with all its superscripts and arbitrary ordering of arguments. In the technical report we introduce a different formalism, that of *tree label logics*, which sheds more light on the idea behind the correctness proof, and will be seen to easily generalize to certain other logics with more expressive power than RCC5-G. Using this, we are able to provide a naive algorithm to solve the full-knowledge single-region compression problem, and prove that the set of formulas returned is minimal in size up to a constant discrepancy, so long as the hierarchical grid has a constant-bounded branching factor.

## 3.1 Size of Compressed Sets

It is of interest to us to know, even though this compression is near-optimal, how much the size of the region description is actually reduced by using it. There is a clean answer to this question for rectangular regions that are exactly unions of cells in a square grid of finite depth: measuring the region's perimeter $p$ as the number of *minimal* cells in contact with its boundary, the minimum number of cells needed to exactly cover the region is $\Theta(p)$ (much better than the naive bound $O(p^2)$ achieved by covering with minimal cells.)

It should be noted that this theorem does not imply that there is, for every regular rectangle $R$ of perimeter $p$, a set of $\Theta(p)$ RCC5-G formulas describing $R$, because depending on its position, many formulas (with predicate DR) may be needed to describe the place where $R$ *is not*. However, as long as $R$ is contained in a cell not too much larger than itself, such a set will exist.

# 4 SEMANTIC DECOMPRESSION

The RCC5 *composition table* can be found in, e.g., [13] (and we reproduce it in the technical report.) It gives, for each two RCC5 relations $P, Q$, the largest set $P \circ Q$ of RCC5 relations $R$ such that $R^I$ intersects $P^I \circ Q^I$ in some interpretation $I$.

Composition tables like this are commonly used with RCC5, RCC8 and similar systems, typically in the context of *constraint networks* (see [4]). An RCC5 constraint network $C$ is a directed graph in which each edge is labeled by a set of RCC5 relations. A network is *atomic* if all edges are labeled by a singleton. A network is *path-consistent* if, whenever $R \in C(x, z)$, there are $P \in C(x, y)$ and $Q \in C(y, z)$ such that $R$ is in the composition set for $P$ and $Q$; and furthermore no edge is labeled by an empty set. (This is a *formal* notion of path consistency; there are also semantic notions.) In most applications of binary constraint calculi and composition tables, reasoning tasks are focused on finding path-consistent, often atomic, networks $C'$ that are consistent with a given network $C$, in the sense that $C'(x, y) \subseteq C(x, y)$ everywhere. A path-consistent network is usually used as a proxy for a network that is satisfiable with respect to some semantics. In our case, there are natural semantics for constraint networks derived from RCC5 and RCC5-G formulas.

The use of the RCC5 composition table for reasoning *with grids* is not in general very powerful. It is in a precise way not complete with respect to the RCC5-G semantics.

The essential problem is that binary relations alone cannot readily capture the idea that the children of $c$ cover $c$, together but not individually. This “problem" cannot be easily avoided. However, the RCC5 composition table is in a certain way complete for the specific purpose we would put it to in this paper, that is, to decompress a representation of a region which has been compressed. Deriving a constraint network $N$ from our near-optimal compression formulas in a natural way, we can prove that there is exactly one path-consistent atomic network $N'$ such that for all $x, y$, $N'(x, y) \subseteq N(x, y)$. Computing this network can be accomplished using well-developed standard tools.

## 5 AN APPLICATION

In the process of building a knowledge graph with a hierarchical grid, it is typically necessary to “triplify" a large amount of data from existing sources, such as database tables containing vector geometries.[2] If additionally the knowledge graph is to use a hierarchical grid to orient its objects in space, as we have claimed may be desirable, then a large number of geometric operations must be done “comparing" grid cells against vector geometries. This process can be made more efficient by strategically choosing which cells to try, rather than computing every spatial relation between every cell and every region. Indeed our compression algorithm does this naturally; it computes enough spatial relations to describe a region as much as spatial relations with the grid can, but since it accesses spatial relations mainly in a top-down-decomposing manner, it does not need access to nearly all of them in general. If it is still considered desirable to include all cell-region relations in the knowledge graph, all others can be computed quickly by logical inference without any more expensive geometric operations.

## 6 CONCLUSIONS AND FUTURE WORK

In addition to compressing the full cell descriptions of single regions in a vacuum, we may consider some more general types of compression problems:

1. Partial Knowledge. Instead of all information that can be captured by RCC5-G about a region, we may sometimes have incomplete information. How well can this be compressed?
2. Multiple Regions. We may have several regions and know RCC5 relations between them, not just between the regions and grid cells. It may be possible to compress this set of relations more thoroughly than can be done when we must disregard the relations between regions.
3. More Expressive Logic. While we advocate against using RCC8, there are other more expressive logics that could be worthwhile to reason about spatial data stored by cell representation. For example, replace the qualitative RCC5 relations with quantitative ones, like “cell $d$ is 60% covered by region $R$". (This kind of relation was popularized in [1].)

In addition, there is of course also more empirical work to be done to substantiate the added value of our approach in application settings.

*Acknowledgement.* This work was supported by the National Science Foundation (NSF) under award OIA-2033521 "KnowWhere-Graph: Enriching and Linking Cross-Domain Knowledge Graphs using Spatially-Explicit AI Technologies."

## REFERENCES

[1] Max J Egenhofer and Matthew P Dube. 2009. Topological Relations from Metric Refinements. In *Proc. of the 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems.* https://doi.org/10.1145/1653771.1653796

[2] Jesse Rosenstock et al. [n.d.]. S2 Geometry Library. https://github.com/google/s2geometry

[3] Zacharias Knudsen et al. [n.d.]. H3: A Hexagonal Hierarchical Geospatial Indexing System. https://github.com/uber/h3

[4] Zeno Gantner, Matthias Westphal, and Stefan Woelfl. 2008. GQR – A Fast Reasoner for Binary Qualitative Constraint Calculi. (2008).

[5] Pascal Hitzler. 2021. A Review of the Semantic Web Field. *Commun. ACM* 64, 2 (Jan. 2021), 76–83. https://doi.org/10.1145/3397512

[6] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. 2010. *Foundations of Semantic Web Technologies.* Chapman and Hall/CRC. https://doi.org/10.1201/9781420090512

[7] Amit Krishna Joshi, Pascal Hitzler, and Guozhu Dong. 2013. Logical Linked Data Compression. In *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 7882)*, Philipp Cimiano, Óscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph (Eds.). Springer, 170–184. https://doi.org/10.1007/978-3-642-38288-8_12

[8] James R Munkres. 1975. *Topology: A First Course.* Prentice-Hall Inc.

[9] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM* 62, 8 (2019), 36–43. https://doi.org/10.1145/3331166

[10] David A Randell, Zhan Cui, and Anthony G Cohn. 1992. A Spatial Logic Based on Regions and Connection. In *Proc. of the 3rd Int. Conf. on Knowledge Representation and Reasoning.*

[11] Jochen Renz. 2002. A Canonical Model of the Region Connection Calculus. *Journal of Applied Non-Classical Logics* 12, 3-4 (2002), 469–494.

[12] Hanan Samet. 2005. *Foundations of Multidimensional and Metric Data Structures.* Morgan Kaufmann. https://doi.org/doi/book/10.5555/1076819

[13] Stephen Schockaert and Sanjiang Li. 2013. Combining RCC5 Relations with Betweenness Information. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.*

[14] Uwe Schöning. 2008. *Logic for Computer Scientists.* Birkhäuser Boston.

[15] Joseph Zalewski, Pascal Hitzler, and Krzysztof Janowicz. 2021. *Semantic Compression with Region Calculi in Nested Hierarchical Grids (Technical Report).* Technical Report. Manhattan, Kansas, USA. https://daselab.cs.ksu.edu/publications/semantic-compression-region-calculi-nested-hierarchical-grids-technical-report

[2] A "triple" is a node-edge-node piece of the knowledge graph.