In: Proceedings of Qualitative Reasoning (QR2019)

Towards Parallelisation of Qualitative Spatial and Temporal Reasoning

Michael Sioutis[1]*, **Diedrich Wolter**[2]
[1]Aalto University, Finland
[2]University of Bamberg, Germany
michael.sioutis@aalto.fi, diedrich.wolter@uni-bamberg.de

Abstract

This summary paper is concerned with algorithmic aspects of qualitative reasoning (QR), in particular qualitative spatial and temporal reasoning (QSTR). Fundamental reasoning techniques in this area are constraint propagators and algorithms for deciding core reasoning problems such as the satisfiability of constraint networks. Developing new algorithms for solving these tasks that exploit parallel computing hardware such as multicore processors or GPUs can be expected to lead to improved reasoning performance. We are motivated to study parallelisation of qualitative reasoning algorithms in order to determine these improvements. In the presence of machine learning applications, which often come with special hardware that supports parallelisation, efficient QR implementations that exploit such hardware can have important applications, for example online verification of AI systems performed in real time.

1 Introduction

Machine learning now plays an important role in the development of autonomous systems. Systems using learnt models introduce specific challenges to the software and system engineering process as, for example, classic verification is not applicable to neural network based components, but only runtime verification can be applied [Taylor *et al.*, 2003]. Given a model describing a computational system, its working environment, and interactions between the computational system and the working environment, runtime verification is applied to check whether actions proposed by the computational system are admissible with respect to the model. Inadmissible actions are inhibited to avoid execution of potentially dangerous actions. In the light of how fragile software based on neural networks is with respect to slight variation or manipulation of the input data—consider the infamous one-pixel attacks to image classifiers [Su *et al.*, 2019], developing means to supervise learning-based systems by a reliable model and runtime verification seems to be a necessity. Qualitative Reasoning presents itself as a good candidate to build models of cyber-physical systems, since qualitative approaches have been demonstrated to be capable of describing complex systems and environments, whilst offering highly efficient methods for the associated reasoning tasks. For example, QR has been considered and eventually applied in space engineering [Forbus, 1990; Williams and Nayak, 1996; Bresina *et al.*, 1999] in which interactions between technical systems like planetary rovers and their environment are important to model.

Aside from the need to develop qualitative representations suitable for modelling intended behaviour of autonomous software, empowering reasoning algorithms to meet hard real-time requirements of runtime verification is of central relevance. With our investigations on how qualitative reasoning algorithms can be parallelised we aim to contribute towards meeting this requirement. Moreover, by parallelising algorithms we take a first step towards benefiting from hardware specialised to application of neural network models, namely, massively parallel processing units like GPUs.

In this paper we consider qualitative reasoning with spatial and temporal relations [Ligozat, 2013] and describe some basic properties that are helpful for developing parallelised versions of existing algorithms.

2 Qualitative Spatial and Temporal Reasoning

One of the most prominent ways of performing qualitative spatial and temporal reasoning is via the use of constraint-based frameworks, such as a *binary qualitative constraint language* [Dylla *et al.*, 2017]. A binary qualitative constraint language is based on a finite set $\mathcal{B}$ of *jointly exhaustive and pairwise disjoint* relations, called the set of *base relations* [Ligozat and Renz, 2004], that is defined over an infinite domain D. These base relations represent definite knowledge between two entities with respect to the level of granularity provided by the domain D; indefinite knowledge can be specified by a union of possible base relations, and is represented by the set containing them. The set $\mathcal{B}$ contains the identity relation Id, and is closed under the *converse* operation ($^{-1}$). The total set of relations $2^{\mathcal{B}}$ is equipped with the usual set-theoretic operations of union and intersection, the converse operation, and the *weak composition* operation denoted by $\diamond$ [Ligozat and Renz, 2004]. For all $r \in 2^{\mathcal{B}}$,

*Contact Author

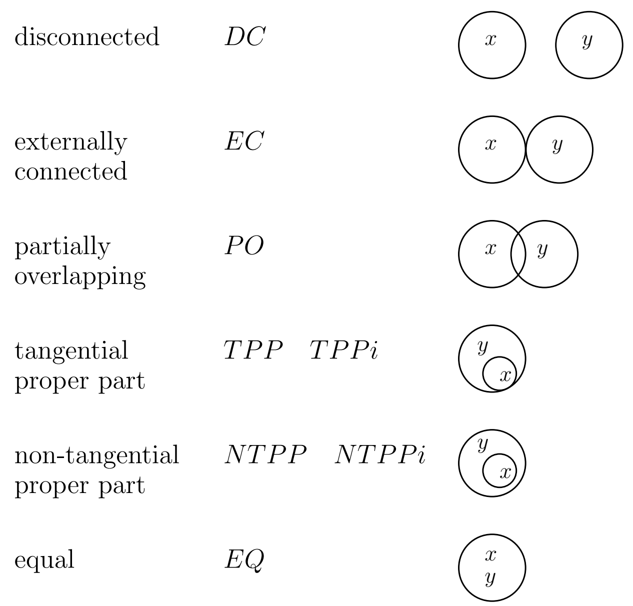

Figure 1: The base relations of RCC8, $\cdot i$ denotes the converse of $\cdot$

$r^{-1} = \bigcup\{b^{-1} \mid b \in r\}$. The weak composition ($\diamond$) of two base relations $b, b' \in \mathcal{B}$ is defined as the smallest (i.e., strongest) relation $r \in 2^{\mathcal{B}}$ that includes $b \circ b'$, or, formally, $b \diamond b' = \{b'' \in \mathcal{B} \mid b'' \cap (b \circ b') \neq \emptyset\}$, where $b \circ b' = \{(x, y) \in \mathsf{D} \times \mathsf{D} \mid \exists z \in \mathsf{D}$ such that $(x, z) \in b \wedge (z, y) \in b'\}$ is the (true) composition of b and b'. For all $r, r' \in 2^{\mathcal{B}}$, $r \diamond r' = \bigcup\{b \diamond b' \mid b \in r, b' \in r'\}$.

As an illustration, let us consider the Region Connection Calculus (RCC), which is a first-order theory for representing and reasoning about mereotopological information [Randell *et al.*, 1992]. The domain D of RCC comprises all possible non-empty regular subsets of some topological space. These subsets do not have to be internally connected and do not have a particular dimension, but are commonly required to be regular *closed* [Renz, 2002]. In particular, a fragment of RCC, called RCC8, makes use of the topological relations *disconnected* (DC), *externally connected* (EC), *equal* (EQ), *partially overlapping* (PO), *tangential proper part* (TPP), *tangential proper part inverse* ($TPPi$), *non-tangential proper part* ($NTPP$), and *non-tangential proper part inverse* ($NTPPi$) to encode knowledge about the spatial relations between regions in some topological space. These spatial relations constitute the set of base relations $\mathcal{B} = \{EQ, DC, EC, PO, TPP, TPPi, NTPP, NTPPi\}$ of RCC8, where each base relation of RCC8 represents a particular topological configuration of two regions in some topological space, and EQ is the identity relation Id of RCC8. The base relations of RCC8 are depicted in Figure 1.

Other notable and well-known qualitative spatial and temporal constraint languages include Point Algebra [Vilain and Kautz, 1986], Cardinal Direction Calculus [Ligozat, 1998; Frank, 1991], Block Algebra [Balbiani *et al.*, 2002], and Cardinal Direction Calculus for extended objects [Goyal and Egenhofer, 2000; Skiadopoulos and Koubarakis, 2005; Liu *et al.*, 2010].

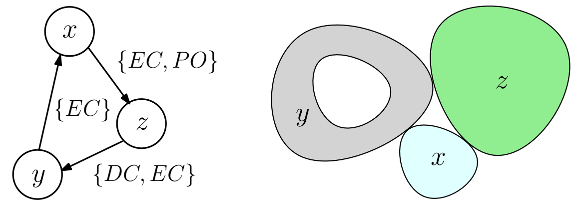

Figure 2: A QCN of RCC8 along with a solution

The problem of representing and reasoning about a given qualitative spatial or temporal constraint language can be modeled as a *qualitative constraint network* (QCN), defined in the following manner:

Definition 1. *A* qualitative constraint network (QCN) *is a tuple* (V, C) *where:* $V = \{v_1, \ldots, v_n\}$ *is a non-empty finite set of variables, each representing an entity of an infinite domain* D*; and* C *is a mapping* $C : V \times V \to 2^{\mathcal{B}}$.

It is typically required that for any given QCN $\mathcal{N} = (V, C)$ it holds that $C(v, v) = \{\mathsf{Id}\}$ for all $v \in V$ and $C(v, v') = (C(v', v))^{-1}$ for all $v, v' \in V$. An example of a QCN of RCC8 is shown in Figure 2; for clarity, neither converse relations nor Id loops are mentioned or shown in the figure.

Given a QCN $\mathcal{N} = (V, C)$, a *solution* of $\mathcal{N}$ is a mapping $\sigma : V \to \mathsf{D}$ such that $\forall (u, v) \in V \times V$, $\exists b \in C(u, v)$ so that $(\sigma(u), \sigma(v)) \in b$ (see again Figure 2).

The fundamental reasoning problems associated with a given QCN $\mathcal{N}$ are the problems of *satisfiability checking*, *minimal labeling* (or *deductive closure*), and *redundancy* (or *entailment*) [Renz and Nebel, 2007]. In particular, the satisfiability checking problem is the problem of deciding if there exists a spatial or temporal valuation of the variables of $\mathcal{N}$ that satisfies its constraints, such a valuation being called a *solution* of $\mathcal{N}$ (as defined earlier), the minimal labeling problem is the problem of finding the strongest implied constraints of $\mathcal{N}$, and the redundancy problem is the problem of determining if a given constraint is entailed by the rest of $\mathcal{N}$ (that constraint being called redundant, as its removal does not change the solution set of the QCN). In general, for most well-known qualitative constraint languages the satisfiability checking problem is NP-hard [Dylla *et al.*, 2013]. Further, the redundancy problem, the minimal labeling problem, and the satisfiability checking problem are equivalent under polynomial Turing reductions [Golumbic and Shamir, 1993].

Many of the published works that study the aforementioned reasoning problems, use *partial $\diamond$-consistency*[1] as a means to define practical algorithms for efficiently tackling them [Amaneddine *et al.*, 2013; Sioutis *et al.*, 2015; Sioutis *et al.*, 2016; Li *et al.*, 2015; Renz and Nebel, 2001; Nebel, 1997; Huang *et al.*, 2013]. Given a QCN $\mathcal{N}$ and a graph G, partial $\diamond$-consistency with respect to G, denoted by $\overset{\diamond}{G}$-consistency, entails consistency for all triples of variables in $\mathcal{N}$ that correspond to three-vertex cycles (triangles) in G. We note that if G is complete, $\overset{\diamond}{G}$-consistency becomes

[1]Note that $\diamond$-consistency can be interpreted as *weak* path-consistency, i.e., path-consistency where the (true) composition of relations is replaced by *weak composition* [Renz and Ligozat, 2005]

identical to $\diamond$-*consistency* [Renz and Ligozat, 2005]. Hence, $\diamond$-consistency is a special case of $\overset{\diamond}{G}$-consistency.

Definition 2. *Given a* QCN $\mathcal{N} = (V, C)$ *and a graph* $G = (V, E)$*,* $\mathcal{N}$ *is said to be* $\overset{\diamond}{G}$-consistent *iff* $\forall\{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$ *we have that* $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$*.*

Specifically, $\overset{\diamond}{G}$-consistency can be directly utilized to decide the satisfiability of QCNs that are defined over particular subclasses of qualitative spatial or temporal relations, called *tractable* classes of relations (cf. [Renz and Nebel, 2007]), it can be applied on a QCN as a preprocessing step to remove impossible, unfeasible, base relations and simplify the problem, and it can also be incorporated in a look-ahead subprocedure in backtracking algorithms whenever the use of such algorithms is appropriate (e.g., in the general case where a QCN is not defined over a tractable class of relations).

3 Parallelising QSTR

In order to efficiently solve a given QCN using some form of parallelism, one would ideally combine the two aspects that characterize a QCN, namely, the structure of its constraint graph and the semantics of the relations over which the QCN is defined. Specifically, the constraint graph of a given QCN is a graph whose set of vertices is the set of variables of the QCN and whose set of edges corresponds to the set of non-trivial constraints of the QCN (i.e., constraints that involve relations other than the universal relation $\mathcal{B}$). In addition, the semantics of the relations over which the QCN is defined refers to the transitivity rules that are implied by the weak composition operation among the different relations of the considered qualitative constraint language.

3.1 Overview of the state of the art

With regard to structural properties of the constraint graphs of the studied QCNs that can be leveraged to boost efficiency of the reasoning process, to the best of our knowledge there does not exist any *practical* published research besides the works that utilize tree decompositions for QCNs [Huang *et al.*, 2013; Sioutis *et al.*, 2017]. The exploitation of tree decomposition became possible due to some generalized theoretical results of [Huang, 2012], which in turn build upon results relating to concrete domains for description logics; in particular, Lutz and Milićič [2007] identify the property of ω-admissibility, which includes a *patchwork property* that grants satisfiability of a complete QCN, given that the QCN can be decomposed into overlapping sub-QCNs (patches) that are individually satisfiable and agree on their overlap. While the paper shows ω-admissibility to hold for RCC8 and Allen's Interval Algebra, the property does not hold for several other constraint languages. In a first step towards parallelization, Sioutis et al. in [Sioutis *et al.*, 2017] go as far as identifying the biconnected components of the constraint graph of a given QCN in order to acquire a particular tree decomposition where the nodes (corresponding to partitions of the original QCN) can be solved independently of one another, in parallel. However, this approach is almost entirely graph-based and does not take into account the semantics of the relations of a QCN (or, to be exact, it only exploits the fact that the identity relation can relate a spatial or temporal entity with itself, which is usually always the case for QCNs). On the other hand, a purely relation-based decomposition technique appears in [Broxvall, 2002], where a QCN is partitioned into smaller QCNs based on a calculus-dependent set of *partitioning relations* that can be treated independently of one another. Although the aforementioned technique is very elegant in its conception, it was noted in [Broxvall, 2002] that useful candidates of this kind of decomposition can be difficult to identify, especially when the size of the set of partitioning relations is small (as is the case with Interval Algebra and RCC8), thus deeming the technique impractical for efficient reasoning with qualitative constraint languages. Finally, an interesting parallel, distributed, implementation of techniques associated with QCNs that addresses the challenge of reasoning over large-scale qualitative spatial and temporal datasets appears in [Mantle *et al.*, 2019]. That implementation builds upon the Apache Spark framework and is tailored for handling constraint networks consisting of millions of relations that cannot be tackled with other state-of-the-art approaches. The downside of the approach described in [Mantle *et al.*, 2019] is that it suffers from the massive startup costs that are associated with distributed parallel applications, the need to move fairly big amounts of data across a network, and uneven workloads. Due to the aforementioned nature of the approach, it can exhibit poor performance for real-world datasets, because such datasets typically present some internal structural properties that the authors' distributed implementation is unable to exploit (see Table 5 in [Mantle *et al.*, 2019]).

3.2 Roadmap to go beyond the state of the art

In light of the previous discussion, we could make the reasonable argument that the area of interdependently exploring the semantics and the structure of the constraint graphs of the studied QCNs in an effort to obtain new decomposability and theoretical tractability properties towards parallelization is almost uncharted territory in QSTR. Therefore, the field is open to lay our own foundations in accordance with the ideas that are presented in what follows.

A fist idea is to do research on structural properties for QCNs, similar to *backbones* and *backdoors* [Williams *et al.*, 2003; Monasson *et al.*, 1999] or *broken triangles* [Cooper *et al.*, 2016] in traditional constraint programming for example, that could allow for exploiting even further the use of tree decompositions in QSTR [Huang *et al.*, 2013; Sioutis *et al.*, 2017] and enable parallelization in a fruitful manner as a consequence (cf. [Sioutis *et al.*, 2017]). Towards this direction, a technical communication is to be presented [Sioutis and Janhunen, 2019], where the notions of backbones and backdoors are defined for QCNs. In particular, a backbone in a given QCN represents the part of it that can only map to a single qualitative configuration (e.g., the relationship between two regions can only be such that one is contained inside the other), and a backdoor in a given QCN represents its intractable part for some local consistency (i.e., utilizing that consistency alone does not allow one to decide the satisfiability of the QCN). These notions relate to our goal of using parallelism in QSTR in the following manner: A potential backbone in a QCN can allow for partitioning its constraint graph into simpler to solve instances, perhaps even independently of one another, in parallel; for example, we can imagine the case where a QCN can be viewed as two overlapping instances whose common constraints form a local backbone. In the same vein, exposing a (strong) backdoor in a given QCN with respect to some local consistency, allows for defining search space splitting approaches, i.e., approaches based on dividing the search space of the QCN into disjoint subspaces to be explored in parallel [Martins *et al.*, 2012].

A second idea is to exploit properties of certain subclasses of qualitative spatial or temporal relations, such as the *distributive* classes of relations [Long and Li, 2015], in order to establish distributive frameworks for reasoning about QCNs. Specifically, distributive classes of relations allow for achieving (weak) global consistency of a QCN via a simple application of $\overset{\diamond}{G}$-consistency [Long and Li, 2015], and this property can be important for the following reason: a given QCN may be refined over such a class of relations, and a tree decomposition of its constraint graph can be devised and used as a basis to solve certain brances in parallel. This idea is in line with what has been done in quantitative temporal reasoning [Boerkoel and Durfee, 2013] and traditional (finite-domain) constraint programming [Kong *et al.*, 2018].

A third idea is to utilize parallel computing hardware, such as multicore processors or GPUs, which are readily available today. While parallelisation techniques discussed so far can directly be applied to multicore architectures, exploitation of highly parallel GPUs is more challenging since these systems have been designed for single-instruction multiple data floating point operations required for computer graphics. With the advent of software using neural networks, GPU hardware becomes particularly useful and is sometime called 'AI hardware'. As such software is in need for formal methods for supervision as discussed in the beginning, exploitation of 'AI hardware' for reasoning is highly relevant. In context of model checking in linear temporal logic (LTL), research already exists that explores means to exploit GPUs [Barnat *et al.*, 2009; Barnat *et al.*, 2012]. A basic idea underlying exploitation of GPUs is to rewrite core operations as vector-matrix multiplications that the GPU can carry out in parallel. For the refinement operation

$$C(x,z) \leftarrow C(x,z) \cap C(x,y) \diamond C(y,z), \quad x,y,z \in V \quad (1)$$

required to test $\overset{\diamond}{G}$-consistency according to Definition 2, such an approach may be possible. However, it remains a challenge to identify means of reducing the overhead of setting up data structures on which the GPU can operate. Furthermore, breadth first search has been shown to be adaptable to GPU implementations [Wu *et al.*, 2014]. Since search is central to QSTR [Renz and Nebel, 2007], exploitation of GPUs may present a viable research direction. The gap between uninformed search to sophisticated search methods using backjumping and conflict analysis that are common in constraint solving is however large and effectiveness of GPUs remains to be shown. While preliminary studies in constraint solving could identify advantages when tackling certain global constraints [Campeotto *et al.*, 2014], the possible impact on QSTR remains unknown.

4 Conclusions

The ability to perform online verification using qualitative system models presents itself as a promising approach to avoid dangerous malfunctioning of systems based on learnt models that cannot be analyzed for functional safety. Exploitation of parallel computing hardware available in modern systems is a natural step in order to meet real-time constraints. While the topic of parallelization is by no means new, there is little work in the area of qualitative reasoning, in particular qualitative spatial and temporal reasoning so far. This summary paper surveys principle approaches to exploit parallelization in qualitative spatial and temporal reasoning. We identify three main approaches: adaption of constraint-based reasoning techniques to qualitative spatial and temporal reasoning, decompositions of constraint networks in 'well-behaved' sub-algebras, and rewriting core operations as vector-matrix computations. Each of the three approaches suggest a rich field of research. In future work we aim to analyze their respective contributions to advancing efficiency of qualitative reasoning and analyze how these techniques can be integrated.

References

[Amaneddine *et al.*, 2013] Nouhad Amaneddine, Jean-François Condotta, and Michael Sioutis. Efficient Approach to Solve the Minimal Labeling Problem of Temporal and Spatial Qualitative Constraints. In *IJCAI*, 2013.

[Balbiani *et al.*, 2002] Philippe Balbiani, Jean-François Condotta, and Luis Fariñas del Cerro. Tractability Results in the Block Algebra. *J. Log. Comput.*, 12:885–909, 2002.

[Barnat *et al.*, 2009] Jiří Barnat, Luboš Brim, Milan Češka, and Tomáš Lamr. CUDA accelerated LTL Model Checking. Technical Report IMU-RS-2009-05, Faculty of Informatics, Masaryk University, FI MU Report Series, 2009.

[Barnat *et al.*, 2012] Jiří Barnat, Petr Bauch, Luboš Brim, and Milan Češka. Designing fast LTL model checking algorithms for many-core GPUs. *J. Parallel Distrib. Comput.*, 72:1083 – 1097, 2012.

[Boerkoel and Durfee, 2013] James C. Boerkoel and Edmund H. Durfee. Distributed Reasoning for Multiagent Simple Temporal Problems. *J. Artif. Intell. Res.*, 47:95–156, 2013.

[Bresina *et al.*, 1999] L. J. Bresina, K. Golden, E. D. Smith, and R. Washington. Increased Flexibility and Robustness of Mars Rovers. In *i-SAIRAS*, 1999.

[Broxvall, 2002] Mathias Broxvall. Constraint Satisfaction on Infinite Domains: Composing Domains and Decomposing Constraints. In *KR*, 2002.

[Campeotto *et al.*, 2014] Federico Campeotto, Alessandro Palù, Agostino Dovier, Ferdinando Fioretto, and Enrico Pontelli. Exploring the Use of GPUs in Constraint Solving. In *PADL*, 2014.

[Cooper *et al.*, 2016] Martin C. Cooper, Achref El Mouelhi, Cyril Terrioux, and Bruno Zanuttini. On Broken Triangles. In *IJCAI*, 2016.

[Dylla *et al.*, 2013] Frank Dylla, Till Mossakowski, Thomas Schneider, and Diedrich Wolter. Algebraic Properties of Qualitative Spatio-Temporal Calculi. In *COSIT*, 2013.

[Dylla *et al.*, 2017] Frank Dylla, Jae Hee Lee, Till Mossakowski, Thomas Schneider, André van Delden, Jasper van de Ven, and Diedrich Wolter. A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties. *ACM Comput. Surv.*, 50:7:1–7:39, 2017.

[Forbus, 1990] Kenneth D. Forbus. Qualitative Models for Space System Engineering – Final report. Technical report, NASA, June 1990.

[Frank, 1991] Andrew U. Frank. Qualitative Spatial Reasoning with Cardinal Directions. In *ÖGAI*, 1991.

[Golumbic and Shamir, 1993] Martin Charles Golumbic and Ron Shamir. Complexity and Algorithms for Reasoning about Time: A Graph-Theoretic Approach. *J. ACM*, 40:1108–1133, 1993.

[Goyal and Egenhofer, 2000] Roop K. Goyal and Max J. Egenhofer. Consistent Queries over Cardinal Directions across Different Levels of Detail. In *DEXA Workshop*, 2000.

[Huang *et al.*, 2013] Jinbo Huang, Jason Jingshi Li, and Jochen Renz. Decomposition and tractability in qualitative spatial and temporal reasoning. *Artif. Intell.*, 195:140–164, 2013.

[Huang, 2012] Jinbo Huang. Compactness and its implications for qualitative spatial and temporal reasoning. In *KR*, 2012.

[Kong *et al.*, 2018] Shufeng Kong, Jae Hee Lee, and Sanjiang Li. A new distributed algorithm for efficient generalized arc-consistency propagation. *Auton. Agents Multi-Agent Syst.*, 32:569–601, 2018.

[Li *et al.*, 2015] Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both. On redundant topological constraints. *Artif. Intell.*, 225:51–76, 2015.

[Ligozat and Renz, 2004] Gérard Ligozat and Jochen Renz. What Is a Qualitative Calculus? A General Framework. In *PRICAI*, 2004.

[Ligozat, 1998] Gérard Ligozat. Reasoning about cardinal directions. *J. Vis. Lang. Comput.*, 9:23–44, 1998.

[Ligozat, 2013] G. Ligozat. *Qualitative Spatial and Temporal Reasoning*. Wiley, 2013.

[Liu *et al.*, 2010] Weiming Liu, Xiaotong Zhang, Sanjiang Li, and Mingsheng Ying. Reasoning about cardinal directions between extended objects. *Artif. Intell.*, 174:951–983, 2010.

[Long and Li, 2015] Zhiguo Long and Sanjiang Li. On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi. In *COSIT*, 2015.

[Lutz and Milićič, 2007] Carsten Lutz and Maja Milićič. A Tableau Algorithm for Description Logics with Concrete Domains and General TBoxes. *J. Autom. Reason.*, 38:227–259, 2007.

[Mantle *et al.*, 2019] Matthew Mantle, Sotirios Batsakis, and Grigoris Antoniou. Large scale distributed spatio-temporal reasoning using real-world knowledge graphs. *Knowl.-Based Syst.*, 163, 2019.

[Martins *et al.*, 2012] Ruben Martins, Vasco M. Manquinho, and Inês Lynce. An overview of parallel SAT solving. *Constraints*, 17:304–347, 2012.

[Monasson *et al.*, 1999] Remi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400:133–137, 1999.

[Nebel, 1997] Bernhard Nebel. Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class. *Constraints*, 1:175–190, 1997.

[Randell *et al.*, 1992] David A. Randell, Zhan Cui, and Anthony Cohn. A Spatial Logic Based on Regions & Connection. In *KR*, 1992.

[Renz and Ligozat, 2005] Jochen Renz and Gérard Ligozat. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *CP*, 2005.

[Renz and Nebel, 2001] Jochen Renz and Bernhard Nebel. Efficient Methods for Qualitative Spatial Reasoning. *J. Artif. Intell. Res.*, 15:289–318, 2001.

[Renz and Nebel, 2007] Jochen Renz and Bernhard Nebel. Qualitative Spatial Reasoning Using Constraint Calculi. In *Handbook of Spatial Logics*, pages 161–215. Springer, 2007.

[Renz, 2002] Jochen Renz. A Canonical Model of the Region Connection Calculus. *J. Appl. Non-Classical Logics*, 12:469–494, 2002.

[Sioutis and Janhunen, 2019] Michael Sioutis and Tomi Janhunen. Towards Leveraging Backdoors in Qualitative Constraint Networks. In *KI*, 2019. To appear.

[Sioutis *et al.*, 2015] Michael Sioutis, Sanjiang Li, and Jean-François Condotta. Efficiently Characterizing Non-Redundant Constraints in Large Real World Qualitative Spatial Networks. In *IJCAI*, 2015.

[Sioutis *et al.*, 2016] Michael Sioutis, Jean-François Condotta, and Manolis Koubarakis. An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks. *Int. J. Artif. Intell. Tools*, 25:1–33, 2016.

[Sioutis *et al.*, 2017] Michael Sioutis, Yakoub Salhi, and Jean-François Condotta. Studying the use and effect of graph decomposition in qualitative spatial and temporal reasoning. *Knowl. Eng. Rev.*, 32:e4, 2017.

[Skiadopoulos and Koubarakis, 2005] Spiros Skiadopoulos and Manolis Koubarakis. On the consistency of cardinal direction constraints. *AIJ*, 163:91–135, 2005.

[Su *et al.*, 2019] J. Su, D. V. Vargas, and K. Sakurai. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*, 2019. Early Access.

[Taylor *et al.*, 2003] Brian J Taylor, Marjorie A Darrah, and Christina D Moats. Verification and validation of neural networks: a sampling of research in progress. In *Intelligent Computing: Theory and Applications*, 2003.

[Vilain and Kautz, 1986] Marc B. Vilain and Henry A. Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *AAAI*, 1986.

[Williams and Nayak, 1996] Brian C. Williams and P. Pandurang Nayak. A model-based approach to reactive self-configuring systems. In *AAAI*, 1996.

[Williams *et al.*, 2003] Ryan Williams, Carla P. Gomes, and Bart Selman. Backdoors To Typical Case Complexity. In *IJCAI*, 2003.

[Wu *et al.*, 2014] Z. Wu, Y. Liu, Y. Liang, and J. Sun. GPU Accelerated Counterexample Generation in LTL Model Checking. In *ICFEM*, 2014.