

Qualitative Spatial and Temporal Reasoning: Current Status and Future Challenges

Michael Sioutis and Diedrich Wolter

Otto-Friedrich-University Bamberg, Bamberg, Germany
michail.sioutis@uni-bamberg.de, diedrich.wolter@uni-bamberg.de

Abstract

Qualitative Spatial & Temporal Reasoning (QSTR) is a major field of study in Symbolic AI that deals with the representation and reasoning of spatio-temporal information in an abstract, human-like manner. We survey the current status of QSTR from a viewpoint of reasoning approaches, and identify certain future challenges that we think that, once overcome, will allow the field to meet the demands of and adapt to real-world, dynamic, and time-critical applications of highly active areas such as machine learning and data mining.

1 Introduction

In everyday natural language descriptions one typically uses expressions such as *outside*, *left of*, or *north of* to spatially or temporally relate one object with another object or oneself, without resorting to providing the exact metric information about these entities. An AI framework that aims to capture this type of human-like representation and reasoning pertaining to space and time is known in the research community as Qualitative Spatial and Temporal Reasoning, or QSTR for short. Specifically, QSTR is a major field of study in Symbolic AI, and in particular in Knowledge Representation & Reasoning, that deals with the fundamental cognitive concepts of space and time in an abstract, natural manner, ranging from theoretical computer science and logic to practical algorithms and applications [Ligozat, 2011]. More formally, QSTR restricts the vocabulary of rich mathematical theories that deal with spatial and temporal entities to simple qualitative constraint languages, which can be used to form interpretable spatio-temporal constraint networks of disjunctions of *atoms* [Ligozat, 2011], such as the one shown in Figure 1. In a sense, QSTR forms a concise framework that allows for rather inexpensive reasoning about entities located in space and time and, hence, further boosts research and applications in a plethora of areas and domains that include cognitive robotics [Dylla and Wallgrün, 2007], qualitative model generation from video [Dubba *et al.*, 2015], ambient intelligence [Bhatt *et al.*, 2009], visual sensemaking [Suchan *et al.*, 2019], data mining [Moskovitch and Shahar, 2015; Kostakis *et al.*, 2017], and qualitative case-based reasoning

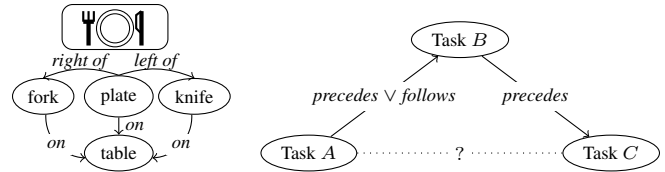


Figure 1: Left: Qualitative abstraction of a spatial configuration. Right: A simplified temporal constraint network of 3 variables, each representing a task; ? denotes complete uncertainty

and learning [Homem *et al.*, 2020]. Here, we exclusively survey reasoning tasks and methods of QSTR beyond what has been covered in previous overviews [Renz and Nebel, 2007], [Cohn and Renz, 2008], and [Ligozat, 2011]. A survey of representation languages, called qualitative calculi in QSTR, is not part of this article as it appears in [Dylla *et al.*, 2017].

In this article we mainly use two representation languages in our examples. The first constraint language to deal with time in a qualitative manner was proposed by Allen in [Allen, 1983], called Interval Algebra. Allen wanted to define a framework for reasoning about time in the context of natural language processing that would be reliable and efficient enough for reasoning about temporal information in a qualitative manner. In particular, Interval Algebra uses intervals on the timeline to represent entities corresponding to actions, events, or tasks. Interval Algebra has become one of the most well-known qualitative constraint languages, due to its use for representing and reasoning about temporal information in various applications. Specifically, notable applications of Interval Algebra involve planning and scheduling [Allen, 1991; Mudrová and Hawes, 2015] natural language processing [Denis and Muller, 2011] workflow [Lu *et al.*, 2006], and intensive care medicine [Ibrahim *et al.*, 2020].

Motivated by the plentiful applications of Interval Algebra, researchers searched for “something like Allen, but for space”. The search turned out to be rather difficult, as Forbus had previously predicted in the poverty conjecture [Forbus *et al.*, 1991], which questions the existence of a universal purely qualitative representation of space. However, with the Region Connection Calculus (RCC) [Randell *et al.*, 1992], an approach appeared that, in the light of its plentiful and diverse applications, can be regarded as almost universal. RCC considers relations between regions in some topological space

that can be derived from a primitive notion of connection. For example, the relation *disconnected* between two regions x and y suggests that none of the points of region x connects with a point of region y , and vice versa. Two fragments of RCC, namely, RCC-8 and RCC-5 (a sublanguage of RCC-8 where no significance is attached to boundaries of regions), have been used in several real-life applications. In particular, Bouzy in [Bouzy, 2001] used RCC-8 in programming the Go game, Lattner et al. in [Lattner et al., 2005] used RCC-5 to set up assistance systems in intelligent vehicles, Heintz et al. in [Heintz and de Leng, 2014] used RCC-8 in the domain of autonomous unmanned aircraft systems (UAS), and Randell et al. in [Randell et al., 2017] used a particular discrete domain counterpart of RCC-8 (called *discrete mereotopology*) to correct segmentation errors for images of human cancer cell line cultures. Other notable applications of RCC involve robot navigation [Fenelon et al., 2013] computer vision [Sridhar et al., 2011], and natural language processing [Kordjamshidi and Moens, 2015].

The remainder of the paper is organized as follows. Section 2 introduces preliminaries on qualitative calculi, and Section 3 describes reasoning tasks in QSTR. Next, in Section 4 we overview the current status of QSTR and list, what we think are, certain shortcomings in the impressive amount of work that has been carried out over the past years. Then, in Section 5 we describe how those challenges could be overcome and what frameworks would need to be developed in order to meet emerging demands in highly active areas such as machine learning and data mining. Finally, in Section 6 we conclude with some final pointers for the future of QSTR.

2 Background

To facilitate discussion, we first recall the formal definition of a *qualitative calculus*, which is a constraint language that is used to represent and reason about qualitative information. Such a calculus is based on a finite set B of *jointly exhaustive and pairwise disjoint* binary relations defined over an infinite domain D (such as a topological space or the real line), called the set of *atoms* [Ligozat, 2011] (in the literature, atoms are also known as *atomic* or *base relations*). A subset of B (item of 2^B) denotes a relation encoding a disjunction of possible atoms, only one of which may hold between two entities (e.g., a region can either *touch* or *contain* another region, both relations cannot hold simultaneously). Hence, 2^B represents the total set of spatial or temporal relations.

Let us revisit the well-known qualitative temporal constraint language of Interval Algebra [Allen, 1983]. The domain D of Interval Algebra is defined to be the set of intervals on the line of rational numbers, i.e., $D = \{x = (x^-, x^+) \in \mathbb{Q} \times \mathbb{Q} \mid x^- < x^+\}$. Each atom can be defined by appropriately constraining the endpoints of two possible intervals, which yields a total of 13 atoms comprising the set $B = \{e, p, pi, m, mi, o, oi, s, si, d, di, f, fi\}$; symbols e, p, m, o, s, d , and f correspond to the atoms *equals*, *precedes*, *meets*, *overlaps*, *starts*, *during*, and *finishes* respectively, with $\cdot i$ denoting the converse of \cdot (note that $ei = e$). For example, d , viz., *during*, is defined as $d = \{(x, y) \in D \times D \mid x^- > y^- \text{ and } x^+ < y^+\}$.

Spatial or temporal information of a qualitative calculus

can be modeled as a *Qualitative Constraint Network* (QCN), which is defined as a network where the vertices correspond to spatial or temporal entities, and the edges are labelled with qualitative spatial or temporal relations respectively. Formally, a QCN can be defined as follows.

Definition 1. A *Qualitative Constraint Network* (QCN) is a tuple (V, C) where:

- $V = \{v_1, \dots, v_n\}$ is a finite set of variables, each representing an entity of an infinite domain D ;
- and C is a mapping $C : V \times V \rightarrow 2^B$ such that $C(v, v) = \{Id\}$ for all $v \in V$, where Id denotes the *identity atom* [Ligozat, 2011] (typically *equals*), and $C(v, v') = (C(v', v))^{-1}$ for all $v, v' \in V$.

We proceed with a gentle example of a QCN.

Example 1. An example of a simplified QCN of Interval Algebra is shown in Figure 1. For instance, formally, the relation between Tasks A and B is $\{p, pi\}$, since the converse of *precedes*, viz., *pi*, is *follows*, and \cdot denotes the whole set of atoms B (hence, each of the atoms in B is possible between Tasks A and C); for simplicity, self-loops corresponding to relation $\{Id\}$ (viz., relation $\{e\}$ in this case) and converse relations of those appearing in the figure, are not depicted. As another example, $x \leq y$ can denote a temporal QCN over \mathbb{Z} .

3 Fundamental QSTR Reasoning Tasks

Given a QCN \mathcal{N} , the literature is particularly interested in its *satisfiability* problem, which is the problem of deciding if there exists a spatial or temporal interpretation of the variables of \mathcal{N} that satisfies its constraints, viz., a *solution* of \mathcal{N} ; for instance, one of the (infinitely many, as the domain is infinite) solutions of the QCN of Figure 1 is

$$(\text{Task } A = 0) \wedge (\text{Task } B = 1) \wedge (\text{Task } C = 2).$$

The satisfiability problem is NP-hard for most qualitative calculi [Dylla et al., 2017]. Further, it lies at the heart of most (if not all) reasoning tasks, as verifying the admissibility of a set of spatio-temporal rules is especially important. While QCNs alone already involve important reasoning tasks, we would like to note that the study of QCNs itself establishes an algorithmic basis for reasoning with more expressive representation languages that build on qualitative primitives too.

Other fundamental reasoning problems that build upon the satisfiability problem include the *minimal labelling* (or *deductive closure*) and the *redundancy* problems [Renz and Nebel, 2007]. The minimal labelling problem is the problem of finding the strongest implied constraints of \mathcal{N} , i.e., determining and maintaining only the atoms in each constraint that are present in a satisfiable *scenario* of \mathcal{N} (also known in the literature as a satisfiable *atomic refinement* or *atomic configuration* of \mathcal{N}); for instance, a satisfiable scenario of the QCN of Figure 1 is (recall that p is *precedes*)

$$(\text{Task } A \{p\} \text{ Task } B) \wedge (\text{Task } B \{p\} \text{ Task } C) \wedge (\text{Task } A \{p\} \text{ Task } C).$$

The unification of all such scenarios yields a minimal network. As noted in [Gottlob, 2012], a minimal network is a quite useful knowledge compilation, since it allows one to answer a number of queries in polynomial time that would otherwise be NP-hard. Indeed, in the context of QSTR, for

instance, one could exploit minimality of a QCN to immediately deduce whether some task A should be scheduled before another task C , or whether some object X could be placed on top of another object Y . For the curious reader, we note that the QCN of Figure 1 is already minimal.

The redundancy problem is the problem of deducing whether a given constraint in \mathcal{N} 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); for instance, the relation $(A \{p\} C)$ is redundant in the aforementioned scenario, and an equivalent set of rules is

$$(\text{Task } A \{p\} \text{Task } B) \wedge (\text{Task } B \{p\} \text{Task } C).$$

The observant reader will notice some similarity between the problems of redundancy and minimality, and indeed in *certain* cases solving the latter allows us to solve the former in polynomial time [Li *et al.*, 2015]. The redundancy problem is particularly important in cases where we want to have the structure of a QCN be as concise and sparse as possible, like in the cases of pattern discovery or search in data mining [Ibrahim *et al.*, 2020; Kostakis *et al.*, 2017].

Naturally, the redundancy problem and the minimal labelling problem are polynomial-time Turing reducible to the satisfiability problem [Golombic and Shamir, 1993].

Finally, variants of the satisfiability problem have been considered. One variant concerns an over-constrained QCN where the task is to compute a spatial or temporal configuration that maximizes the number of satisfied constraints in that QCN; this problem is called the MAX-QCN problem and was recently introduced in [Condotta *et al.*, 2015]. The motivation behind studying the MAX-QCN problem lies in the fact that representing spatio-temporal information may inevitably lead to inconsistencies due to, for example, human error or contradictory data of different sources. This is true for various real-world temporally or spatially annotated data, for which there are ongoing efforts to implement tools for practical tasks such as verification, repair, and visualization.^{1,2} Another variant restricts the domain of some variables to fixed objects, which may result in the satisfiability problem becoming harder to solve [Li *et al.*, 2013]. This variant is relevant for real-world applications that require spatial reasoning with concrete entities [Stock *et al.*, 2015; Kreuzmann and Wolter, 2014]. Other reasoning problems consider the task of computing a solution of a QCN, which can be more difficult than merely deciding existence [Renz, 2002], and spatio-temporal reasoning tasks based on *conceptual neighborhoods* [Freksa, 1992] for qualitative simulation or continuity constraints formulation.

4 Current Status

In this section we review the current status in QSTR with respect to how it can(not) adapt to applications and how (in)accessible it is to the non-expert user.

4.1 Vanilla QSTR Reasoners

Let us first outline how native QSTR methods for deciding satisfiability are constructed. They implement a form of

¹<http://www.timeml.org/timebank/timebank.html>

²<http://gadm.geovocab.org/>

Algorithm 1 Refinement Search

```

1: procedure CHECK(QCN  $\mathcal{N} = (V, C)$ , sub-algebra  $\mathcal{T}$ )
2:    $\mathcal{N} \leftarrow \text{PROPAGATE}(\mathcal{N})$ 
3:   if  $\exists v_i, v_j \in V : C(v_i, v_j) = \{\}$  then
4:     return failure ▷ trigger backtracking/conflict analysis
5:   else if  $\forall v_i, v_j \in V : C(v_i, v_j) \in \mathcal{T}$  then
6:     DECIDE( $\mathcal{N}$ )
7:   else
8:     select  $v_i, v_j \in V$  with  $C(v_i, v_j) \notin \mathcal{T}$  ▷ heuristic search for refinement
9:     select  $r$  as refinement of  $C(v_i, v_j)$  with  $r \in \mathcal{T}$ 
10:    CHECK( $\mathcal{N}$  updated by setting  $C(v_i, v_j)$  to  $r$ ,  $\mathcal{T}$ )
11:   end if
12: end procedure

```

search as shown in Algorithm 1. At the heart of the method, a dedicated decision method DECIDE (line 6) is used that is capable of deciding satisfiability for a QCN that only involves a certain *sub-algebra*, i.e., subset of relations (denoted \mathcal{T} in the algorithm). For several calculi, the well-known path-consistency algorithm can be applied to decide satisfiability for QCNs that only contain atomic relations; the method is called *algebraic closure* in QSTR due to the algebraic relation composition operation being weak for most calculi [Dylla *et al.*, 2017]. Identifying applicable decision procedures and the largest sets of relations over which satisfiability of QCNs can be decided using these procedures is an important task when analyzing calculi. The overall algorithm then performs a refinement search by splitting all disjunctive relations that cannot be handled by DECIDE into relations that *can* be handled, thus refining a relation $s \notin \mathcal{T}$ into $r \subset s, r \in \mathcal{T}$ (lines 8–10). Each refinement step is followed by constraint propagation (line 2) using the algebraic closure method. Upon failure, some form of backtracking or backjumping is performed.

4.2 Choice of Tools Requires Expert Knowledge

Real-world problems involving spatio-temporal information are extremely diverse and do not come with a handbook of which and how QSTR techniques should be applied. Depending on the use case, spatio-temporal information may vary in complexity, size, and nature. As an example, certain real-world temporally and spatially annotated data that appear on the web today are of huge size, involving up to millions of entities, but of very low complexity and density, allowing typically one or, less often, two possible relations to hold between two entities.^{1,2} Such datasets can be tackled with polynomial methods that exploit the sparsity of the underlying constraint graphs quite efficiently [Sioutis *et al.*, 2016]. On the other hand, in other contexts, such as industrial ones, like job shop scheduling or planning, spatio-temporal information involves fewer entities in general, as the information is naturally grounded on a restricted amount of resources; however, there is much more flexibility with regard to the possible relations that can hold between two entities [Sioutis *et al.*, 2020].

Moreover, there is an abundance of native QSTR techniques for tackling many of the problems that exist today [Amaneddine *et al.*, 2013; Sioutis *et al.*, 2015; Long *et al.*, 2016; Sioutis *et al.*, 2018; Gantner *et al.*, 2008; Wolter and Wallgrün, 2012], but no clear strategy of which tool applies best to which dataset. This lack of clarity is further impaired by alternative, non-native tools applicable to the same problems, ranging from Boolean satisfiability (SAT) [Glo-

rian *et al.*, 2018; Condotta *et al.*, 2016; Huang *et al.*, 2013] to Answer Set Programming (ASP) [Baryannis *et al.*, 2020; Janhunen and Sioutis, 2019; Walega *et al.*, 2017].

Currently, and to the best of our knowledge, there has not been a complete answer to the question of how the various native and non-native tools should be chosen and could be combined to tackle QSTR problems.

4.3 Self-adaptive QSTR Techniques Are Missing

So far, the tuning of all previously referenced tools relies upon expert knowledge, as they are not adaptive to the particulars of a given problem instance, which inhibits off-the-shelf application to challenging tasks such as online verification of neural methods [Alirezaie *et al.*, 2019] or data mining [Kostakis *et al.*, 2017]. Every one of the aforementioned tools makes its own assumptions regarding the complexity, size, and nature of the spatio-temporal information involved;³ this may not only affect performance when a tool is to be used in a generic setting, but may even lead to the reasoning process being unsound if its prerequisites are not met.

In the above context, to the best of our knowledge there does not exist any current work that is sufficiently self-adaptive and/or that composes different reasoning techniques in an intelligent manner. One notable exception could be the work of [Glorian *et al.*, 2018], where a CNF encoding is combined with a native QSTR algorithm in a lazy setting for satisfiability checking of spatial networks; however, the domain of spatial regions in that work is assumed to only consist of rectangles, which impacts the overall soundness of the approach if applied to tasks involving arbitrarily shaped regions.

With respect to collaborative frameworks, Sioutis and Janhunen have recently provided a short roadmap in terms of identifying certain structural characteristics in qualitative spatio-temporal constraint networks, *viz.*, *backdoors* and *backbones* [Sioutis and Janhunen, 2019] (cf. [Williams *et al.*, 2003]), which could be used to define combined approaches among SAT, ASP, and native tools and lead to the development of adaptive tools with a better insight into the specifics of real-world and industrial datasets than what is possible today [Sioutis and Janhunen, 2019, Section 4].

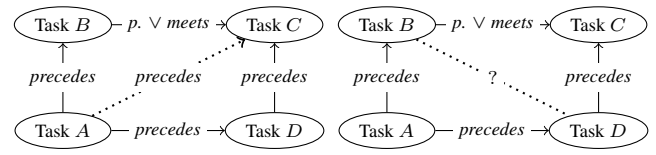
5 Future Challenges

Based on the critical discussion in Section 4, we describe here, what we think are, some viable contributions to QSTR that would allow the field to become more accessible to the non-expert user and better connected to highly active areas such as machine learning and data mining.

5.1 Self-adaptive Techniques for QSTR

In this section, we argue for enhancing existing QSTR techniques with self-adaptive capabilities, and working towards novel self-adaptive approaches based on statistical learning. To this end, we identify the following challenges.

³This is to be expected to some extent for such specialized techniques, and should not negatively reflect the authors' work.



(a) Chord between Tasks A & C (b) Chord between Tasks B & D

Figure 2: Two different ways to triangulate a simplified temporal constraint network of 4 variables and 4 constraints (forming a cycle of length > 3); adding a chord as in (a) allows us to infer a single possibility between Tasks A and C, whereas in (b) all possibilities are maintained between Tasks B and D (*p.* is short for *precedes*)

Relation-aware triangulation Over the past decade, the use of *chordal* graphs has become prevalent when dealing with sparse QCNs, typical of man-made data [Huang *et al.*, 2013; Sioutis *et al.*, 2016; Sioutis *et al.*, 2016; Condotta *et al.*, 2016; Amaneddine *et al.*, 2013; Sioutis *et al.*, 2015]. In short, a graph is chordal if every cycle of length > 3 has a *chord*, which is an edge that is not part of the cycle but connects two vertices of the cycle. Triangulating the underlying (constraint) graph of a QCN instead of completing it allows us to reduce the search space. However, to date, the used triangulation procedures in QSTR do not take at all into account the spatio-temporal relations that label the edges. Let us consider Figure 2 as an example; we can optimally⁴ triangulate the depicted temporal constraint network in two ways, but only one of them is best in terms of reducing the amount of possibilities in the resulting chordal constraint graph. Figuring out the best (or simply a good) triangulation for a QCN becomes even more complicated when taking into account the fact that a resulting triangulation directly affects the pruning capacity of a propagator pertaining to some local consistency; in general, the sparser the triangulation, the weaker the local propagation (i.e., fewer unfeasible atoms are removed) [Sioutis *et al.*, 2016, Section 3.3], though there are certain restricted classes of instances for which the pruning capacity is not affected [Long and Li, 2015]. Therefore, within this context we raise the following questions:

- How much time should we invest in triangulating a given QCN in a relation-aware manner?
- Should we adjust the strength of local propagation as a given QCN is being tackled?
- What lazy strategies of the above could work in practice (triangulating/propagating as needed)?

We think that answering these questions would require the implementation of heuristics and consequent development of intelligent techniques that would address the shortcomings of current tools, *viz.*, their inability to self-adapt and the need of expert knowledge for selecting and tuning them.

Novel techniques based on statistical learning When trying to obtain a satisfiable refinement of a given QCN (e.g., choosing *follows* over *precedes* in the network of Figure 1), it may be beneficial to prefer relations in a given constraint

⁴Obtaining an optimal triangulation is NP-hard [Tarjan and Yannakakis, 1984] in general, but there exist many methods, linear to polynomial in the size of the graph, that yield good approximations.

that individually appear in many local satisfiable atomic configurations, and then use those collectively to construct a complete valid configuration. This leads to a counting-based scheme that forms the basis for a dynamic heuristic. During reasoning, the heuristic would self-adapt to the structure of a network instance based on the counts of local models at hand. A first attempt has been taken in [Sioutis and Wolter, 2020], which shows promising results that motivate further research in statistical methods for QSTR. Specifically, we argue for employing statistical learning via encoding vectors pertaining to statistical regularities of qualitative constraints; these vectors could then be used to guide search or even decide parts of a network in reasoning tasks pertaining to satisfiability checking for example. In the context of the aforementioned work, such a vector would currently comprise only the number of local models, in the way it was explained earlier, but ideally one would want to capture more features of a given constraint, such as the cardinality of its assigned relation, the frequency of repetition of that relation in adjacent constraints, the density of the neighbourhood of the constraint, its splitting factor, i.e., whether its assigned relation belongs to a tractable set of relations or it should be split into two or more subrelations, its proximity to a Horn formula, and its potential search space *coverage* (e.g., relation $\{>\}$ provides more coverage than $\{=\}$ in qualitative point-based reasoning, as the latter restricts two points to being equal and, hence, focuses on a single point), among others. Certain dynamic features may also be explored and recorded, by running some propagators for a limited amount of time on a given instance.

5.2 Self-adaptive Ensemble Solvers for QSTR

In this section, we argue for synthesizing standalone self-adaptive techniques (see previous section) into collaborative and generic ensemble solvers that can self-adapt, by use of satisfiability modulo theories (SMT) and portfolio solving. To this end, we identify the following challenges.

Abstraction-based satisfiability modulo theories Modeling certain QSTR reasoning tasks as Boolean satisfiability (SAT) problems can lead to very efficient implementations in general for relatively small-sized instances, which however do not scale well [Huang *et al.*, 2013; Westphal and Wöflf, 2009]. Some efforts towards resolving the scalability issue have been made by means of abstracting a QCN instance into a SAT instance where a lot of the required encodings (for soundness and completeness) are initially missing and are only added incrementally upon need [Glorian *et al.*, 2018], without fully addressing the blow-up issue though in the generic case (in fact, [Glorian *et al.*, 2018] is only sound when the domain of spatial regions is assumed to only consist of rectangles). On the other hand, native QSTR tools can scale much better in general [Sioutis *et al.*, 2018; Sioutis *et al.*, 2016], but may not be able to keep up with SAT tools for solving some very hard and dense problems [Glorian *et al.*, 2018; Huang *et al.*, 2013; Westphal and Wöflf, 2009]. We think that it is possible to get the best of both worlds via a simple technique where a *hard* part of a QCN instance will be solved by SAT solving, then the resulting (partial) solution will be checked in combination with the rest of the instance by native qualitative spatio-temporal reasoning (a theory-specific

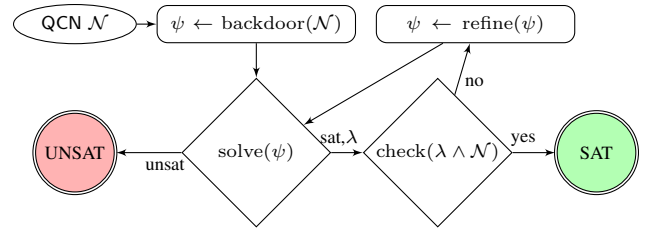


Figure 3: The backdoor (a hard part) of a QCN instance is *solved* by SAT solving, then the resulting (partial) solution is *checked* in combination with the rest of the instance by native qualitative spatio-temporal reasoning, and the procedure repeats iteratively until it converges to a point where a solution is obtained or unsatisfiability is detected; method *refine* suggests that more explicit information is added to counteract a failure in the checking phase (such information would typically involve parts of the QCN not considered before)

solver), and the process will be repeated until it converges, as is typical in lazy SMT approaches; see Figure 3 for details (see also [Sioutis and Janhunen, 2019, Section 4] in this regard). In contrast to SAT or traditional constraint programming, a *hard* part of a QCN instance may be easily recognized due to the domain specific nature of the calculus involved (a finite number of known and well-defined atoms is used). Specifically, Sioutis and Janhunen in [Sioutis and Janhunen, 2019] have introduced a notion of *backdoors* and *backbones* in the context of QCNs that are tied to the use of local consistencies that can be enforced in polynomial time; a backdoor, i.e., a subset of constraints of a given QCN instance, constitutes a hard part of that instance, as deciding its satisfiability allows one to decide the satisfiability of the whole instance. In our experience, and as supported by the literature [Renz and Nebel, 2001, Section 5], such hard parts in QCNs tend to form locally and hence are of relatively small size. Thus, we expect this approach to yield good results and allow further future directions to be defined. Due to the hard-part parameterization and lazy SMT setting, we think that this approach would be more capable of adapting to the particulars of real-world and industrial datasets than the tools available today.

Hierarchical portfolio of solvers The prevalence of multi-core architectures has led a substantial amount of research effort to the development of portfolio solvers, i.e., meta-solvers that run $n > 1$ constituent solvers (which can be instances of a single solver with different settings or several different solvers of the same type) on the same problem in parallel. Such solvers have been extensively used in the SAT community to great success [Balyo *et al.*, 2015; Hamadi *et al.*, 2009; Xu *et al.*, 2008], and in the traditional constraint programming (CP) community to a lesser extent but with good results nevertheless [Amadini *et al.*, 2016]. To the best of our knowledge, there does not exist any similar approach in the field of QSTR. This is unfortunate, because QCNs can be encoded and tackled in different ways, viz., as Boolean satisfiability (SAT) or even traditional (finite-domain) constraint satisfaction problem (CSP) instances [Westphal and Wöflf, 2009], and as Answer Set Programming (ASP) instances too [Baryannis *et al.*, 2020; Janhunen and Sioutis, 2019; Walega *et al.*, 2017], and there should be some intelligent way

of figuring out what the best approach would be for a given task; it is most often the case that “one reasoning tool does not fit all”. To this end, we argue for developing a novel hierarchical portfolio-based approach for tackling QCNs that will not solely rely on native QSTR methods, but will choose among native QSTR, SAT, CSP, and/or ASP techniques and encoding(s) to be used (e.g., there are various QSTR encodings in SAT and ASP in particular). Such approaches that make a series of decisions are referred to as hierarchical models [Hurley *et al.*, 2014]. To make the required decisions, various classification and clustering algorithms could be evaluated on standard CSP and SAT features as described in [Hurley *et al.*, 2014], standard ASP features as presented in [Hoos *et al.*, 2014], and the native QCN features for which we argued in the previous section.

5.3 Facilitating AI/ML with Self-adaptive QSTR

In this section, we argue for reaching out to the fields of data mining and machine learning in terms of facilitating pattern recognition algorithms and enabling *abductive learning* [Zhou, 2019] respectively; abductive learning corresponds to a bidirectional feedback loop between the symbolic framework and the machine learning model [Zhou, 2019, Figure 1]. To this end, we identify the following challenges.

Boosting spatio-temporal pattern recognition algorithms

Temporal intervals naturally appear in medical (and otherwise) data in the form of sequences of labeled events; multiple events may be concurrent, and, in general, any event may start or end independently of another [Kostakis and Papapetrou, 2017]. Abstracting such intervals to qualitative information, e.g., using Interval Algebra [Allen, 1983] or similar calculi, can provide advantages in pattern recognition tasks. For instance, employing an interval sequence-based temporal abstraction on multi-variate time series representing the medical history of patients, leads to substantial improvement in terms of predictive performance against traditional classifiers on real datasets [Moskovitch and Shahar, 2015]. Similar temporal abstractions have been employed for the exploration and classification of renal-damage risk factors in patients with diabetes type II [Klimov *et al.*, 2015], and sepsis prediction, and explanation thereof, using intensive care unit data [Ibrahim *et al.*, 2020]. As another example, when monitoring heart failure patients, one is interested in identifying particular diagnoses/treatment patterns in their medical history, such as whether the standard treatment procedures for heart failure have been applied in the proper order [Asker *et al.*, 2016]. However, despite the important use of (qualitative) temporal intervals in the medical domain, the QSTR community has not been interested much in facilitating physicians and medical researchers by looking into the particulars of the temporal QCNs forming in medical data and providing the required reasoning tools. Specifically, we argue for optimizing such QCNs by removing redundancy and simplifying them, i.e., sparsifying the related temporal constraint graphs. Intuitively, it should then be easier to identify patterns in a sparse, unbiased with respect to implicit facts, graph.

Enabling adaptive learning In [Alirezaie *et al.*, 2019], QSTR acts as a model-based referee upon the output of a

classifier to improve performance of semantic segmentation for satellite imagery data by rejecting impossible interpretations. Drawing inspiration from that work, we argue for working towards a *generic* neuro-symbolic framework that will integrate QSTR and neural methods from a probabilistic perspective. Such an integration is currently identified as an open challenge in the AI community [Raedt *et al.*, 2020, Section 9], yet it is not clear what current and novel QSTR techniques are needed to empower handling spatio-temporal information. Our first take on it is as follows. We think that qualitative spatial and temporal variables could be annotated with the probability-infused output of a classifier, and spatial and temporal relations themselves would carry a probability too. A simplified example of such a neuro-symbolic formula would look as follows (%s denote confidence):

$$X^{(95\% \text{ yolk})} \text{ is contained in }^{(45\% \text{ true})} \text{ or overlaps }^{(55\% \text{ true})} Y^{(90\% \text{ egg})}.$$

In that sense, QSTR would become neurally-enhanced, and probabilities would be used to encode a *bidirectional feedback loop* between the symbolic framework and the machine learning model, known as *abductive learning* [Zhou, 2019, Figure 1]. In a sense, we argue here for using logic to compose concepts learned by machine learning methods, and also allowing learned concepts by machine learning methods to influence that logical composition. To this end, we identify the need for establishing probabilistic QSTR frameworks, as well as dynamic and adaptive algorithms for runtime verification of neural network-based components.

Following up on the last point of integrating QSTR with machine learning, we also point to Knowledge Graph Embeddings (KGEs) as a means to realize the integration [Schockaert and Li, 2018]. KGEs are often inspired by conceptual spaces, which are abstract spatial domains to which QSTR techniques could be applied, yet they may differ from Euclidean spaces previously considered in QSTR and thus require future research. As research strives for learning logic structures richer than relational statements, the aforementioned generalizations of QSTR formulae also call for a better understanding of spatial logics based on QSTR concepts.

6 Conclusion

Qualitative Spatial and Temporal Reasoning is a field of research that is relevant to a wide range of research and application areas—virtually to all tasks that involve processing of spatial or temporal information. As of today, a great variety of representations exist and reasoning methods are sophisticated enough to handle even large problem instances in the light of NP-hard reasoning tasks. Nevertheless, like many mature fields in AI, selection and tuning of tools requires expert knowledge. We have critically surveyed current approaches for solving the fundamental satisfiability problem, and we have discussed future research directions in the field that we think are important to make QSTR serve as an engine that powers even more contemporary AI applications.

Acknowledgments

The work was partially supported by the BMBF project *Dependable Intelligent Systems*. We would also like to thank Dr. Zhiguo Long for fruitful discussions on QSTR challenges.

References

- [Alirezaie *et al.*, 2019] Marjan Alirezaie, Martin Långkvist, Michael Sioutis, and Amy Loutfi. Semantic referee: A neural-symbolic framework for enhancing geospatial semantic segmentation. *Semantic Web*, 10:863–880, 2019.
- [Allen, 1983] James F. Allen. Maintaining Knowledge about Temporal Intervals. *Commun. ACM*, 26:832–843, 1983.
- [Allen, 1991] James F. Allen. Planning as Temporal Reasoning. In *KR*, 1991.
- [Amadini *et al.*, 2016] Roberto Amadini, Maurizio Gabbriellini, and Jacopo Mauro. An Extensive Evaluation of Portfolio Approaches for Constraint Satisfaction Problems. *Int. J. Interact. Multimed. Artif. Intell.*, 3:81–86, 2016.
- [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.
- [Asker *et al.*, 2016] Lars Asker, Henrik Boström, Panagiotis Papatetrou, and Hans E. Persson. Identifying Factors for the Effectiveness of Treatment of Heart Failure: A Registry Study. In *CBMS*, 2016.
- [Balyo *et al.*, 2015] Tomás Balyo, Peter Sanders, and Carsten Sinz. Hordesat: A massively parallel portfolio SAT solver. In *SAT*, 2015.
- [Baryannis *et al.*, 2020] George Baryannis, Ilias Tachmazidis, Sotiris Batsakis, Grigoris Antoniou, Mario Alviano, and Emmanuel Papadakis. A Generalised Approach for Encoding and Reasoning with Qualitative Theories in Answer Set Programming. *Theory Pract. Log. Program.*, 20:687–702, 2020.
- [Bhatt *et al.*, 2009] Mehul Bhatt, Frank Dylla, and Joana Hois. Spatio-terminological Inference for the Design of Ambient Environments. In *COSIT*, 2009.
- [Bouzy, 2001] Bruno Bouzy. Les concepts spatiaux dans la programmation du go. *Rev. d’Intelligence Artif.*, 15:143–172, 2001.
- [Cohn and Renz, 2008] Anthony G. Cohn and Jochen Renz. Qualitative Spatial Representation and Reasoning. In *Handbook of Knowledge Representation and Reasoning*, pages 551–596. Elsevier, 2008.
- [Condotta *et al.*, 2015] Jean-François Condotta, Ali Mensi, Issam Nouaouri, Michael Sioutis, and Lamjed Ben Said. A Practical Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *ICTAI*, 2015.
- [Condotta *et al.*, 2016] Jean-François Condotta, Issam Nouaouri, and Michael Sioutis. A SAT Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks. In *KR*, 2016.
- [Denis and Muller, 2011] Pascal Denis and Philippe Muller. Predicting Globally-Coherent Temporal Structures from Texts via Endpoint Inference and Graph Decomposition. In *IJCAI*, 2011.
- [Dubba *et al.*, 2015] Krishna Sandeep Reddy Dubba, Anthony G. Cohn, David C. Hogg, Mehul Bhatt, and Frank Dylla. Learning Relational Event Models from Video. *J. Artif. Intell. Res.*, 53:41–90, 2015.
- [Dylla and Wallgrün, 2007] Frank Dylla and Jan Oliver Wallgrün. Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control. *J. Intell. Robot. Syst.*, 48:55–78, 2007.
- [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.
- [Fenelon *et al.*, 2013] Valquiria Fenelon, Paulo Eduardo Santos, Hannah M. Dee, and Fábio Gagliardi Cozman. Reasoning about shadows in a mobile robot environment. *Appl. Intell.*, 38:553–565, 2013.
- [Forbus *et al.*, 1991] Kenneth D. Forbus, Paul Nielsen, and Boi Faltings. Qualitative spatial reasoning: The CLOCK project. *Artif. Intell.*, 51:417–471, 1991.
- [Freksa, 1992] Christian Freksa. Temporal Reasoning Based on Semi-Intervals. *Artif. Intell.*, 54:199–227, 1992.
- [Gantner *et al.*, 2008] Zeno Gantner, Matthias Westphal, and Stefan Wöfl. GQR-A Fast Reasoner for Binary Qualitative Constraint Calculi. In *AAAI Workshop on Spatio-Temporal Reasoning*, 2008.
- [Glorian *et al.*, 2018] Gael Glorian, Jean-Marie Lagniez, Valentin Montmirail, and Michael Sioutis. An Incremental SAT-Based Approach to Reason Efficiently on Qualitative Constraint Networks. In *CP*, 2018.
- [Golombic and Shamir, 1993] Martin Charles Golombic and Ron Shamir. Complexity and Algorithms for Reasoning about Time: A Graph-Theoretic Approach. *J. ACM*, 40:1108–1133, 1993.
- [Gottlob, 2012] Georg Gottlob. On minimal constraint networks. *Artif. Intell.*, 191-192:42–60, 2012.
- [Hamadi *et al.*, 2009] Youssef Hamadi, Saïd Jabbour, and Lakhdar Sais. ManySAT: a Parallel SAT Solver. *J. Satisf. Boolean Model. Comput.*, 6:245–262, 2009.
- [Heintz and de Leng, 2014] Fredrik Heintz and Daniel de Leng. Spatio-Temporal Stream Reasoning with Incomplete Spatial Information. In *ECAI*, 2014.
- [Homem *et al.*, 2020] Thiago Pedro Donadon Homem, Paulo Eduardo Santos, Anna Helena Reali Costa, Reinaldo Augusto da Costa Bianchi, and Ramón López de Mántaras. Qualitative case-based reasoning & learning. *Artif. Intell.*, 283:103258, 2020.
- [Hoos *et al.*, 2014] Holger H. Hoos, Marius Lindauer, and Torsten Schaub. claspfolio 2: Advances in Algorithm Selection for Answer Set Programming. *Theory Pract. Log. Program.*, 14:569–585, 2014.
- [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.
- [Hurley *et al.*, 2014] Barry Hurley, Lars Kotthoff, Yuri Malitsky, and Barry O’Sullivan. Proteus: A Hierarchical Portfolio of Solvers and Transformations. In *CPAIOR*, 2014.
- [Ibrahim *et al.*, 2020] Zina Ibrahim, Honghan Wu, and Richard Dobson. Modeling Rare Interactions in Time Series Data Through Qualitative Change: Application to Outcome Prediction in Intensive Care Units. In *ECAI*, 2020.
- [Janhunen and Sioutis, 2019] Tomi Janhunen and Michael Sioutis. Allen’s Interval Algebra Makes the Difference. In *DECLARE*, 2019.
- [Klimov *et al.*, 2015] Denis Klimov, Alexander Shknevsky, and Yuval Shahar. Exploration of patterns predicting renal damage in patients with diabetes type II using a visual temporal analysis laboratory. *J. Am. Medical Informatics Assoc.*, 22:275–289, 2015.
- [Kordjamshidi and Moens, 2015] Parisa Kordjamshidi and Marie-Francine Moens. Global machine learning for spatial ontology population. *J. Web Sem.*, 30:3–21, 2015.

- [Kostakis and Papapetrou, 2017] Orestis Kostakis and Panagiotis Papapetrou. ABIDE: Querying Time-Evolving Sequences of Temporal Intervals. In *IDA*, 2017.
- [Kostakis *et al.*, 2017] Orestis Kostakis, Nikolaj Tatti, and Aristides Gionis. Discovering recurring activity in temporal networks. *Data Min. Knowl. Discov.*, 31:1840–1871, 2017.
- [Kreutzmann and Wolter, 2014] Arne Kreutzmann and Diedrich Wolter. Qualitative Spatial and Temporal Reasoning with AND/OR Linear Programming. In *ECAI*, 2014.
- [Lattner *et al.*, 2005] Andreas D. Lattner, Ingo J. Timm, Martin Lorenz, and Otthein Herzog. Knowledge-based risk assessment for intelligent vehicles. In *KIMAS*, 2005.
- [Li *et al.*, 2013] Sanjiang Li, Weiming Liu, and Shengsheng Wang. Qualitative constraint satisfaction problems: An extended framework with landmarks. *Artif. Intell.*, 201:32–58, 2013.
- [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, 2011] Gérard Ligozat. *Qualitative Spatial and Temporal Reasoning*. ISTE Series. Wiley, 2011.
- [Long and Li, 2015] Zhiguo Long and Sanjiang Li. On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi. In *COSIT*, 2015.
- [Long *et al.*, 2016] Zhiguo Long, Michael Sioutis, and Sanjiang Li. Efficient Path Consistency Algorithm for Large Qualitative Constraint Networks. In *IJCAI*, 2016.
- [Lu *et al.*, 2006] Ruopeng Lu, Shazia Wasim Sadiq, Vineet Padmanabhan, and Guido Governatori. Using a temporal constraint network for business process execution. In *ADC*, 2006.
- [Moskovitch and Shahar, 2015] Robert Moskovitch and Yuval Shahar. Classification of multivariate time series via temporal abstraction and time intervals mining. *Knowl. Inf. Syst.*, 45:35–74, 2015.
- [Mudrová and Hawes, 2015] Lenka Mudrová and Nick Hawes. Task scheduling for mobile robots using interval algebra. In *ICRA*, 2015.
- [Raedt *et al.*, 2020] Luc De Raedt, Sebastijan Dumancic, Robin Manhaeve, and Giuseppe Marra. From Statistical Relational to Neuro-Symbolic Artificial Intelligence. In *IJCAI*, 2020.
- [Randell *et al.*, 1992] David A. Randell, Zhan Cui, and Anthony Cohn. A Spatial Logic Based on Regions and Connection. In *KR*, 1992.
- [Randell *et al.*, 2017] David A. Randell, Antony Galton, Shereen Fouad, Hisham Mehanna, and Gabriel Landini. Mereotopological Correction of Segmentation Errors in Histological Imaging. *J. Imaging*, 3:63, 2017.
- [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 Class. Logics*, 12:469–494, 2002.
- [Schockaert and Li, 2018] Steven Schockaert and Sanjiang Li. Reasoning about betweenness and RCC8 constraints in qualitative conceptual spaces. In *IJCAI*, 2018.
- [Sioutis and Janhunen, 2019] Michael Sioutis and Tomi Janhunen. Towards Leveraging Backdoors in Qualitative Constraint Networks. In *KI*, 2019. Short paper.
- [Sioutis and Wolter, 2020] Michael Sioutis and Diedrich Wolter. Dynamic Branching in Qualitative Constraint Networks via Counting Local Models. In *TIME*, 2020.
- [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.*, 2018] Michael Sioutis, Zhiguo Long, and Sanjiang Li. Leveraging Variable Elimination for Efficiently Reasoning about Qualitative Constraints. *Int. J. Artif. Intell. Tools*, 27:1860001, 2018.
- [Sioutis *et al.*, 2020] Michael Sioutis, Zhiguo Long, and Tomi Janhunen. On Robustness in Qualitative Constraint Networks. In *IJCAI*, 2020.
- [Sridhar *et al.*, 2011] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg. From Video to RCC8: Exploiting a Distance Based Semantics to Stabilise the Interpretation of Mereotopological Relations. In *COSIT*, 2011.
- [Stock *et al.*, 2015] Sebastian Stock, Masoumeh Mansouri, Federico Pecora, and Joachim Hertzberg. Online Task Merging with a Hierarchical Hybrid Task Planner for Mobile Service Robots. In *IROS 2015*, 2015.
- [Suchan *et al.*, 2019] Jakob Suchan, Mehul Bhatt, and Srikrishna Varadarajan. Out of Sight But Not Out of Mind: An Answer Set Programming Based Online Abduction Framework for Visual Sensemaking in Autonomous Driving. In *IJCAI*, 2019.
- [Tarjan and Yannakakis, 1984] Robert Endre Tarjan and Mihalis Yannakakis. Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs. *SIAM J. Comput.*, 13:566–579, 1984.
- [Walega *et al.*, 2017] Przemyslaw Andrzej Walega, Carl P. L. Schultz, and Mehul Bhatt. Non-monotonic spatial reasoning with answer set programming modulo theories. *Theory Pract. Log. Program.*, 17:205–225, 2017.
- [Westphal and Wöflf, 2009] Matthias Westphal and Stefan Wöflf. Qualitative CSP, Finite CSP, and SAT: Comparing Methods for Qualitative Constraint-based Reasoning. In *IJCAI*, 2009.
- [Williams *et al.*, 2003] Ryan Williams, Carla P. Gomes, and Bart Selman. Backdoors To Typical Case Complexity. In *IJCAI*, 2003.
- [Wolter and Wallgrün, 2012] Diedrich Wolter and Jan O. Wallgrün. Qualitative Spatial Reasoning for Applications: New Challenges and the SparQ Toolbox. In *QSTR: Trends and Future Directions*, pages 336–362. IGI Global, 2012.
- [Xu *et al.*, 2008] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based Algorithm Selection for SAT. *J. Artif. Intell. Res.*, 32:565–606, 2008.
- [Zhou, 2019] Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Sci. China Inf. Sci.*, 62:76101:1–76101:3, 2019.