Implicit Constraints for Qualitative Spatial and Temporal Reasoning

Jochen Renz
Research School of Computer Science
The Australian National University
Canberra, ACT, Australia
jochen.renz@anu.edu.au

Abstract

Qualitative information about spatial or temporal entities is represented by specifying qualitative relations between these entities. It is then possible to apply qualitative reasoning methods for tasks such as checking consistency of the given information, deriving previously unknown information or answering queries. Depending on the kind of information that is represented, qualitative reasoning methods might lead to incorrect results, and it is a topic of ongoing research efforts to determine when and why this occurs. In this paper we present two possible explanations for this behaviour: (1) the existence of implicit entities that we do not explicitly represent; (2) the existence of implicit constraints that have to be satisfied, but which are not explicitly represented. We show that both of these can lead to undetected inconsistencies. By making these implicit entities and constraints explicit, and by including them in the qualitative representation, we are able to solve problems that could not be solved qualitatively before. We present different examples of implicit entities and implicit constraints and an algorithm for solving them.

Introduction

A qualitative representation of spatial or temporal information typically consists of a number of given spatial or temporal entities and given qualitative relations between these entities. Qualitative reasoning about such a representation includes tasks such as deciding whether the given information is consistent, inferring previously unknown information from the given information, or answering queries about the given entities and relations. The dominant approach for qualitative spatial and temporal reasoning over the past 20+ years has been to use a qualitative spatial or temporal calculus (Cohn and Renz 2008). Typically[1], this is defined as follows:

1. We take an infinite domain $\mathcal{D}$ that contains all possible spatial or temporal entities of a particular type (usually points, intervals or regions in a one, two, or three dimensional space);
2. We define a finite set $\mathcal{B}$ of pairwise disjoint and jointly exhaustive binary[2] base relations over $\mathcal{D}$, i.e., between any pair of values of $\mathcal{D}$ exactly one base relation holds;
3. We determine the converse relation of each base relation and the composition $\circ$ between any pair of base relations (defined as $R \circ S = \{(x, z) | \exists y : (x, y) \in R \text{ and } (y, z) \in S\}$), and only accept sets of base relations for which their powerset $2^{\mathcal{B}}$ is closed under composition, intersection, union and converse of relations.

Using these relations, spatial and temporal information can be represented qualitatively by specifying a set Θ of constraints of the form xRy, where x, y are variables over $\mathcal{D}$ and $R \in 2^{\mathcal{B}}$. Taking relations of $2^{\mathcal{B}}$ instead of only relations of $\mathcal{B}$ allows us to specify indefinite information as a union of possible base relations when it is not known which one holds.

The most common reasoning problem is to determine consistency of Θ, i.e., decide whether there is an instantiation of each variable with values from $\mathcal{D}$ such that all constraints are satisfied. Qualitative reasoning can then be done by applying the so-called *path-consistency* operation (Mackworth 1977), which takes a triple of variables x, y, z and their constraints $xRy, ySz, xTz \in \Theta$ and replaces xTz with the new constraint $x(R \circ S) \cap Tz$. This operation removes base relations that cannot hold between x and z. If we remove all base relations of a constraint, then Θ is inconsistent. Otherwise, we apply this operation to all triples of Θ repeatedly until no base relations are removed any more, in which case the resulting set Θ' is *path-consistent*.

The limit of the path-consistency operation is reached when all constraints consist of only one base relation. We call this an *atomic* set of constraints. If there are still constraints with multiple base relations left, we can try path-consistency again by selecting each of the base relations separately.[3] This is usually done in a backtracking manner for

[1] We will later discuss that this is not always the case.

[2] Here we restrict our exposition to binary relations, but there have also been calculi of larger arity in the literature.

[3] There is a significant amount of work on identifying tractable subsets of qualitative calculi for which path-consistency decides consistency (Renz 2007). Once such a tractable subset has been identified, it is not necessary to test each base relation separately, but rather unions of base relations that are part of the tractable subset. This can considerably increase the efficiency of reasoning, but has no immediate relevance to this paper.

all the different constraints (Van Beek and Manchak 1996). But if we only have an atomic set of constraints left, it is commonly accepted that nothing more can be done using qualitative reasoning alone.

It is, therefore, considered to be one of the most important questions in the area of qualitative spatial and temporal reasoning to determine if and when path-consistency decides consistency for atomic sets of constraints of a given calculus. If we can prove that it does, then all is good and we can do reasoning within the calculus using path-consistency and backtracking. However, in cases where we can show that path-consistency does not decide consistency for the base relations of a calculus, this is often considered a set-back and a demonstration of the limits of qualitative reasoning.

In this paper we show that qualitative reasoning is much more powerful than using only path-consistency, and that even in cases where qualitative reasoning seems to fail, there is still much more qualitative reasoning that can be done. In particular, we demonstrate that one of the main shortcomings of the current standard way of qualitative representation and reasoning is that much of the readily available qualitative information is not being used (Section 3). There is an abundance of implicit information available, in the form of implicit spatial or temporal entities and in the form of implicit qualitative constraints. We demonstrate that ignoring this implicit information is the main reason why current qualitative reasoning methods fail in some cases (Section 4). We show that by making this implicit information explicit, we can successfully apply qualitative reasoning methods to cases that seemed impossible to solve using qualitative reasoning alone. We also introduce a new qualitative reasoning method that goes beyond the traditional path-consistency method and which is not based on composition in the traditional sense (Section 5).

In the following section we give some more details on qualitative spatial and temporal reasoning and present additional background information.

Qualitative Spatial & Temporal Reasoning

One of the main motivations of qualitative spatial and temporal reasoning is the ability to represent information about spatial or temporal entities without knowing all the exact details about them. For example, we do not need to know the exact shape and the exact location of an object in order to represent information about it. It is usually enough to know that the phone is on the desk in the study, rather than knowing its exact coordinates and those of the desk. Dealing with such everyday qualitative information requires qualitative reasoning and, therefore, both qualitative representation and qualitative reasoning are considered to be similar to the way humans usually deal with spatial and temporal information. The lack of exact details and the inherent vagueness of qualitative information makes it very hard to represent this kind of information in a quantitative way, using for example a coordinate system. In the introduction we mentioned that sometimes the limit of qualitative reasoning seems to be reached and that other methods have to be used. But due to this difficulty of adequately representing qualitative information in an equivalent quantitative way, there are often no

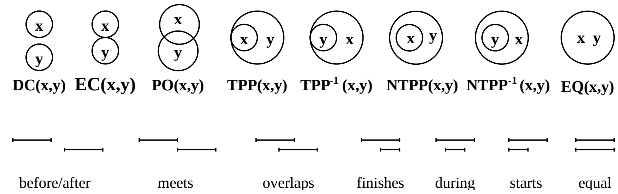

Figure 1: The eight base relations of RCC8 (top) and eight of the 13 base relations of the Interval Algebra (bottom). The remaining six base relations are the converse relations of these.

alternative methods available and qualitative reasoning is all we can do.

Some well-known examples

The three best known and most widely studied qualitative spatial and temporal calculi are the *Point Algebra (PA)* (Vilain et al. 1990), the *Interval Algebra (IA)* (Allen 1983), and the *Region Connection Calculus RCC8* (Randell et al. 1992) (see Figure 1). The domain of the PA are the points on a directed infinite line, its base relations are $\{<, =, >\}$. The domain of the IA are convex intervals on a directed infinite line. Its base relations represent 13 different possibilities how two intervals on a directed line can be topologically related: before (b), after (bi), meets (m), met-by (mi), overlaps (o), overlapped-by (oi), starts (s), started-by (si), during (d), contains (di), finishes (f), finished-by (fi), and equal (eq). The domain of RCC8 is the (regular) closed regions in an n-dimensional space. The 8 base relations are the different possibilities how two region can be related topologically: disconnected (DC), externally connected (EC), partially overlap (PO), tangential proper part (TPP), non-tangential proper part (NTPP), the converses TPPI and NTPPI, and equal (EQ).

The IA is closely related to the PA as intervals can be represented as a set of two endpoints and all the IA base relations can be expressed as simple combinations of PA relations over the four endpoints. For both of these calculi, it is straightforward to compute the compositions between any two base relations since all involved entities can be enumerated. For both calculi, path-consistency decides consistency for atomic sets of constraints. RCC8 is much more difficult than IA and PA as there is no obvious way of formally representing the shape of an arbitrary region. This makes it very difficult (if not impossible) to compute the compositions of the RCC8 base relations. By identifying a counterexample, it was found that RCC8 is actually not closed under composition, which violates point 3 in the introduction. As a consequence, the concept of *weak composition* was introduced, which is the smallest relation contained in $2^{\mathcal{B}}$ that contains the actual composition of two base relations (Ligozat and Renz 2004). Obviously, any calculus is closed under weak composition.

Algebraic closure and how it can fail

One consequence of using weak composition instead of composition in the definition of a calculus is that path-consistency does not work any more as it requires composition. Path-consistency was, therefore, replaced by the concept of *algebraic closure*, which is the same operation as path-consistency, but uses weak composition instead of composition (Ligozat and Renz 2004). For cases where the actual composition can be computed, weak composition is equivalent to composition, and hence, weak composition and algebraic closure is always used nowadays. Even though RCC8 uses weak composition instead of composition, it has been shown that algebraic closure decides consistency for atomic sets of constraints.

This raises the question of what difference it makes whether we know the composition or only the weak composition of a calculus. It has been shown that the reason for whether algebraic closure decides consistency for atomic sets of constraint does not depend on whether we have composition or weak composition, but whether a calculus is *closed under constraints* (Renz and Ligozat 2005). A set of relations is not closed under constraints, if it is possible to enforce non-overlapping subatomic relations. A hypothetical example of this would be if a set $\mathcal{S}$ of RCC8 constraints enforces that a certain constraint $xECy \in \mathcal{S}$ only has solutions where x connects to y along a line, and if another set $\mathcal{S}'$ enforces that $xECy$ only has a solution where x connects to y at a single point. By considering $\mathcal{S} \cap \mathcal{S}'$ we would still have $xECy$ but it cannot be instantiated as it cannot connect only along a line and only along a point at the same time. Here, connecting along a line, and connecting at a point are non-overlapping subatomic relations of the base relation EC. In such a case, algebraic closure cannot detect the inconsistency in general. But since algebraic closure decides consistency for atomic sets of RCC8 constraints, our hypothetical example is not possible for RCC8.

In this paper we will introduce an additional qualitative method that has the potential to detect inconsistencies generated by non-overlapping subatomic relations.

Shortcomings of Current Representations – Implicit Entities and Implicit Constraints

In this section we demonstrate the shortcomings of existing qualitative spatial and temporal representations. For this we identify implicit entities and implicit constraints.

Implicit spatial and temporal entities

We begin by formally defining the concept of an implicit entity.

Definition 1 (Implicit, explicit and conditional entities) *Given a set Θ of spatial or temporal constraints xRy, where $x, y \in \mathcal{V}$ are variables over a domain $\mathcal{D}$, and $R \in 2^{\mathcal{B}}$ is a spatial or temporal relation over a set of base relations $\mathcal{B}$. We assume that $\mathcal{D} \subseteq \mathcal{R}^n$, i.e., it is not a set of symbols, but a set of entities in an n-dimensional Euclidean space $\mathcal{R}^n$.*

For a given consistent instantiation of Θ, we call each entity in $\mathcal{D}$ that is explicitly referred to by a variable in Θ an explicit entity*, and refer to the set of all explicit entities as $\mathcal{E} \subseteq \mathcal{D}$. An* implicit entity *is any entity that can be derived from elements of $\mathcal{E}$ in a clearly defined way, for example by union, intersection, set difference, complement, convex hull, or other one-to-one, or many-to-one functions. We refer to the set of all implicit entities as $\mathcal{I} \subseteq \mathcal{D}^*$, where $\mathcal{D}^*$ is the closure of $\mathcal{D}$ under the functions we use to derive $\mathcal{I}$. We note that $\mathcal{D}^*$ can contain elements of lower dimension than $\mathcal{D}$.*

To make implicit entities explicit *means that we extend the set of* explicit variables $\mathcal{V}$ *to $\mathcal{V}^*$ by adding* implicit variables *$\overline{v} \in \mathcal{V}_{\mathcal{I}}$ that refer to particular implicit entities. Variables in $\mathcal{V}_{\mathcal{I}}$ are dependent variables of $\mathcal{V}$, since their instantiation is determined by instantiations of variables of $\mathcal{V}$ using a clearly defined one-to-one or many-to-one function. An implicit variable that may be instantiated with the empty set $\emptyset$ is called a* conditional variable *(written as $\widehat{z}$) and the corresponding entity a* conditional entity.

Conditional entities can be interesting if they provide a meaningful qualitative distinction, such as cases where the conditional entity exists vs. cases where it does not exist.

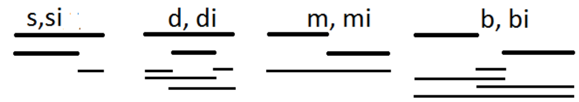

Figure 2: Some IA relations and their implicit entities, depicted using thin lines.

Let us look at some examples for implicit entities. We start with the Interval Algebra and the base relation $starts$. Whenever the constraint xsy is satisfied, there must be two explicit entities $x = [x_s, x_e]$ and $y = [y_s, y_e]$, where the startpoints x_s of x and y_s of y are the same, while the endpoint x_e of x is inside y before the endpoint y_e of y. But there is also an implicit entity $\overline{z}$ which is the interval $[x_e, y_e]$ from when x ends to when y ends (see Fig. 2, left). This interval might never be used and never be referred to explicitly, but there is no denying the fact that this interval must exist whenever xsy is satisfied.

There are similar implicit intervals for the other 12 base relations of the IA. In general, every IA base relation xBy uses between two and four distinct points out of the four endpoints x_s, x_e, y_s, y_e of x and y. Any two of these points define an interval. There are 6 intervals $(3 * 4/2)$ for the base relations with 4 distinct endpoints, 3 interval $(3 * 2/2)$ for the base relations with 3 distinct endpoints, and 1 interval $(2 * 1/2)$ for the equal relation with 2 distinct endpoints (see Fig. 2 for some examples)

Proposition 2 (Implicit entities of the Interval Algebra) *Given an atomic set Θ over the Interval Algebra. Whenever a constraint $xRy \in \Theta$ is consistently instantiated with two intervals $[x_s, x_e]$ and $[y_s, y_e]$, the four endpoints x_s, x_e, y_s, y_e induce implicit intervals as follows:*

- *If R is one of b, bi, o, oi, d, di, then there are 4 distinct endpoints that induce 4 unique implicit intervals.*
- *If R is one of m, mi, s, si, f, fi, then there are 3 distinct endpoints that induce 1 unique implicit interval.*
- *If $R = eq$, there are only two distinct endpoints and no implicit intervals.*

Definition 3 *Given an atomic set Θ over the Interval Algebra and its variables $\mathcal{V}$. We introduce a fresh implicit variable $\overline{v}$ for each implicit interval induced by Θ as specified* *in Proposition 2. We define $\mathcal{V}_{IA}$ as the set that consists of $\mathcal{V}$ and all implicit variables $\overline{v}$.*

Lemma 4 *Given an atomic set Θ over the Interval Algebra with variables $\mathcal{V}$ representing intervals. Let E be the set of all endpoints corresponding to variables $v \in \mathcal{V}$, i.e., $E = \{v_s, v_e | v = [v_s, v_e] \in \mathcal{V}\}$. $\mathcal{V}_{IA}$ contains a variable referring to each pair of endpoints $e_i, e_j \in E$.*

Proof Sketch. Each $e \in E$ belongs to one variable in $\mathcal{V}$. Any two variables $x, y \in \mathcal{V}$ form an atomic constraint in Θ and there is a variable corresponding to each pair of endpoints from x and y either in $\mathcal{V}_{IA}$ or on $\mathcal{V}$. ∎

We can also obtain implicit entities for RCC8. For all explicit entities x, we can define the boundary δx as an implicit entity of x. For the constraint $xPOy$ we get four implicit entities[4] that exist whenever $xPOy$ is satisfied: $\overline{z_1} = x \cap y$, $\overline{z_2} = x \setminus y$, $\overline{z_3} = y \setminus x$, and $\overline{z_4} = x \cup y$. For the PO relation there is an interesting conditional enitity that can be useful in distinguishing cases of PO relations. It is the intersection of the boundary of x with the boundary of y: $\widehat{z_5} = \delta x \cap \delta y$ and allows us to distinguish between cases where it exists and cases where it does not exist. For the TPP and TPPi relations, the intersection of the boundaries of x and y is an implicit entity as there must always be a non-empty intersection, i.e., $\overline{z_1} = \delta x \cap \delta y$. The second implicit entity is the set difference between x and y, i.e., $\overline{z_2} = x \setminus y$ for TPPi and $\overline{z_2} = y \setminus x$ for TPP. For the EC relation, the intersection of the boundaries of x and y is also an implicit entity, but since x and y only intersect at the boundaries, it is equivalent to $\overline{z_1} = x \cap y$. The second implicit entity of EC is $\overline{z_2} = x \cup y$. For the DC relation, the union is the only implicit entity, while for NTPP and NTPPi the set differences are the only implicit entities.

Note that some of these implicit entities of RCC8 are actually used in the definition of the 9-intersection model (Egenhofer and Franzosa 1991). Egenhofer and Franzosa considered the 9 possible intersections of the boundaries, interiors and exteriors of two regions and defined relations according to whether these intersections are empty or non-empty, leading to 2^9 potential relations. In the end, they grouped these relations together to form 8 different base relations, similar to the RCC8 relations. To the best of our knowledge, the intersections corresponding to our implicit entities were not used in the way and for the purpose we are proposing in this paper.

Unless there is an explicit entity that happens to be equivalent to an implicit entity, these implicit entities are never considered when we do composition based qualitative reasoning in the way it has been done in the past 20+ years. It is possible that while using qualitative reasoning, one or more of these explicit entities become empty. This leads to a contradiction which may be undetected by the existing qualitative reasoning methods. However, the more likely case is that constraints that must hold for implicit entities are violated and that this leads to undetected contradictions.

[4]Note: In order to make our formalism easier to understand, we are slightly abusing notation by writing statements such as $x \cap y$ where x and y are variables over a domain, while $\cap$ is only defined for domain values and not for variables.

Implicit relations and implicit constraints

Given the existence of implicit entities, it is clear that there must be constraints between implicit entities, and between implicit and explicit entities that are never considered by the traditional qualitative reasoning methods. However, there are also additional constraints between explicit entities that have to be satisfied whenever a given constraint or a given set of constraints is consistent. These additional constraints are usually a consequence of the fundamental properties of space and time and must hold for whatever spatial or temporal domains we use. These constraints can be so simple and obvious that we might forget to make them explicit. An example of implicit constraints has been used by (Gerevini and Renz 2002): If a region x is contained in a region y, then x must be smaller than y. $x < y$ is an implicit constraint that must be satisfied whenever $xTPPy$ or $xNTPPy$ are satisfied.

Definition 5 (implicit and conditional constraints) *Given a set Θ of spatial or temporal constraints over a domain $\mathcal{D} \subseteq \mathcal{R}^n$ and over a set of spatial or temporal relations $2^{\mathcal{B}}$ as above. Given a set $\mathcal{V}^*$ of variables over $\mathcal{D}^*$, referring to the implicit and explicit entities of Θ, where $\mathcal{D}^*$ is the closure of $\mathcal{D}$ under the relevant transformations. An* implicit constraint *$R(x_1, \ldots, x_n)$, where $x_1, \ldots, x_n \in \mathcal{V}^*$ and $R \subseteq \mathcal{D}^* \times^n \mathcal{D}^*$ is a constraint between implicit entities, between explicit entities, or between implicit and explicit entities that is not part of Θ, but that has to be satisfied whenever Θ is consistent. A relation $R \subseteq \mathcal{D}^* \times^n \mathcal{D}^*$ is called an* implicit relation *if it is used as part of an implicit constraint. A* conditional constraint *is similar to an implicit constraint, but does not have to be satisfied whenever Θ is consistent.*

We give some examples for implicit constraints that use the implicit entities we defined above. For the IA relation $starts$ and the constraint xsy, we discussed that there is one implicit entity, the interval $\overline{z}$ between the endpoint of x and the endpoint of y. Some obvious implicit constraints in this case are $xm\overline{z}$ and $\overline{z}fy$. Another implicit constraint is $x \cup \overline{z} = y$. Since we know that x and $\overline{z}$ do not overlap, we also have the implicit constraint $duration(x) + duration(\overline{z}) = duration(y)$, or in short $x + \overline{z} = y$.

It is of course one of the main features of the IA that we do not consider the duration of intervals and that the IA base relations they satisfy are independent of their durations. But independent of this, and independent of the actual durations of the intervals and whether we know them or not, these implicit constraints have to be satisfied whenever the constraint xsy is consistent. They represent some of the fundamental properties of time and space, and we often forget about them in our qualitative representations. Similar implicit constraints hold for other IA relations.

Definition 6 (Implicit constraints of the Interval Algebra) *Given a set Θ of atomic constraints over the Interval Algebra and the corresponding set of explicit and implicit variables $\mathcal{V}_{IA}$. Each variable in $\mathcal{V}_D = \mathcal{V}_{IA} \setminus \mathcal{V}$ is a* dependent variable *of two variables in $\mathcal{V}$. The IA base relations between two variables $x, y \in \mathcal{V}$ and all their common* *dependent variables are clearly defined. We define Θ_{IA} as the set of all (explicit and implicit) IA constraints between any two variables in $\mathcal{V}$ and their common dependent variables. Note that Θ_{IA} includes Θ.*

In addition to IA constraints, we can also obtain implicit size constraints (IS) of type $x+yRz$, with $x, y, z \in \mathcal{V}_{IA}$ and R a relative size relation of the set $\{<, >, =\}$. We obtain such an IS constraint whenever $xmy \in \mathcal{V}_{IA}$ and the relations between x and z and between y and z in $\mathcal{V}_{IA}$ are the following:

1. *If xdz and (ydz or yfz) in Θ_{IA}, then $x + y < z$ holds.*
2. *If xsz and ydz in Θ_{IA}, then $x + y < z$ holds.*
3. *If xsz and yfz in Θ_{IA}, then $x + y = z$ holds.*
4. *If zsx, zfx, zdx, zsy, zfy, or zdy in Θ_{IA}, then $x+y > z$ holds.*
5. *If xoz and zoy or $zfiy$ in Θ_{IA}, then $x + y > z$ holds.*
6. *If xsz and zoy in Θ_{IA}, then $x + y > z$ holds.*

We define Θ_{IS} as the set of all these implicit size constraints.

The same kinds of implicit constraints also hold for RCC8. For the constraint $xPOy$, for example, using the notations we introduced after Lemma 4, we get the implicit constraint $\overline{z_1} \cup \overline{z_2} = x$, or similarly $volume(\overline{z_1}) + volume(\overline{z_2}) = volume(x)$ (simplified just $\overline{z_1} + \overline{z_2} = x$). In addition, we get the implicit constraints $\overline{z_1} + \overline{z_3} = y$, $x + \overline{z_2} = \overline{z_4}$ and $y + \overline{z_1} = \overline{z_4}$. Note that for RCC8, we have some implicit entities that are of lower dimension, and we assume that any entity of lower dimension has a volume of 0 when compared with entities of higher dimension. We can specify similar constraints for the boundary of regions. For example, the sum of all disjoint pieces of the boundary cannot be larger than the whole boundary. An example for a conditional constraint is "$\overline{z_1}$ is a point" or "$\overline{z_1}$ is a line" for the RCC8 relation EC, as discussed in the previous section.

While we can specify implicit entities and implicit constraints for the IA and for RCC8, we cannot give an example where they actually make a difference. This is because for IA and for RCC8, algebraic closure decides consistency for atomic sets of constraints and they are both closed under constraints. We believe that this is the main reason why implicit entities and implicit constraints have been ignored in the past.

In the next section we analyse some calculi for which algebraic closure does not decide consistency for atomic sets of constraints. It turns out that implicit entities and implicit constraints have a significant effect on these calculi.

When Implicit Entities and Implicit Constraints Matter

We have seen in the previous section that implicit entities and implicit constraints can be defined, but they do not have an effect for calculi where algebraic closure decides consistency. However, there are many calculi for which this property is not satisfied. Some of these calculi are relatively simple extensions or modifications of the "big two", the IA and RCC8. In this section we have a closer look at two of these calculi and try to understand why qualitative reasoning fails for these calculi and how this relates to implicit entities and implicit constraints. We start with a simple case.

The Containment Algebra

Our first example is the Containment Algebra (Ladkin and Maddux 1994) which consists of 5 base relations and is isomorphic to a subalgebra of IA. The 5 base relations are $equal(=)$, $contains(c)$, $contained\text{-}in(ci)$, $nonempty\text{-}intersection(n)$, and $apart(a)$. They correspond to unions of IA relations as follows:

$$= \quad \equiv \quad \{eq\} \qquad (1)$$

$$c \quad \equiv \quad \{s, f, d\} \qquad (2)$$

$$ci \quad \equiv \quad \{si, fi, di\} \qquad (3)$$

$$n \quad \equiv \quad \{o, oi\} \qquad (4)$$

$$a \quad \equiv \quad \{b, bi, m, mi\} \qquad (5)$$

The containment algebra does not distinguish between the direction of intervals and does not consider the endpoint of intervals when comparing them. As such it is very similar to RCC5, the subalgebra of RCC8 that does not consider the boundary of regions. While algebraic closure decides consistency for atomic sets of RCC 5 constraints, it does not decide consistency for the containment algebra, as the following example demonstrates.

Example 7 *Given the set of constraints in the containment algebra $\Theta = \{xny, ynz, xaz, wny, wax, waz\}$. Since x and z both have a non-empty intersection with y, they must overlap y from two different sides. w also overlaps y but is apart from x and z. This is impossible as y only has two different sides from which it can be overlapped. This inconsistency cannot be detected using algebraic closure.*

This example is consistent for RCC5 since we do not have the restriction that two entities can only overlap from two sides. For the Containment Algebra, we get the same implicit entities as for the Interval Algebra, which allows us to distinguish the two sides of an interval. By adding the implicit entities and the corresponding implicit constraints, we can solve instances of the Containment Algebra. But since algebraic closure decides consistency for the ORD-Horn subset of IA (Nebel and Bürckert 1999) and since all containment algebra relations are in ORD-Horn, we can also detect inconsistency of the given example by converting it into IA relations and running algebraic closure on the transformed set.

INDU – Qualitative meets Quantitative

The INDU calculus has been introduced by (Pujari et al. 1999) and its complexity has been analysed by (Balbiani et al. 2006). INDU is a straightforward and fairly simple extension of the IA. In addition to the 13 IA base relation, INDU also considers the relative durations of intervals. Each interval can have a shorter, longer, or equal duration with respect to any of the other intervals, which can be represented using the standard point algebra relations. INDU consists of 25 base relations: for each of the IA base relations b, bi, m, mi, o, oi there will be 3 INDU base relations of the form $b_<, b_=, b_>$, for the other 7 IA base relations the relative size of the intervals is fixed to $<$ for s, f, d, to $>$ for si, fi, di and to $=$ for eq. In addition, we have to consider the implicit entities and implicit constraints that we already have for the IA.

Definition 8 (Implicit INDU entities and constraints) *Given a set of INDU constraints Θ, and the corresponding sets of IA constraints Θ^I and the corresponding set of PA constraints Θ^P. Θ has the same implicit entities and the same implicit IA and implicit size constraints as Θ^I. In addition to the existing PA constraints Θ^P, we also get implicit size constraints for Θ. All implicit and explicit size constraints Θ_{PA} we get for Θ are the following:*

- *If $aRb \in \Theta^P$, then add aRb to Θ_{PA}*
- *If $a + b < c$ or $a + b = c$ in Θ_{IS}, then add $a < c$ and $b < c$ to Θ_{PA}*

The next lemma follows immediately from this definition.

Lemma 9 *Given a set of INDU constraints Θ. Θ is consistent if and only if $\Theta_{IA} \cup \Theta_{IS} \cup \Theta_{PA}$ is consistent.*

So INDU is essentially a combination of the IA with the PA. The PA is one of the simplest algebras one can consider and, intuitively, a combination of IA with PA should not be too much harder than the IA alone. However, consider the simple example of three intervals x, y, z where the INDU constraints $xs_<z$, $xm_>y$, and $yf_<z$ hold. In this example, the relative duration of x is larger than that of y, so the relative duration of x must be larger than half of the duration of z. We could similarly enforce that the duration of x is less than half of z:

Example 10 *Given six variables $x_1, x_2, y_1, y_2, z_1, z_2$ and the following inconsistent INDU constraints: (1) $x_1s_<z_1$, $\mathbf{x_1m_>y_1}$, $y_1f_<z_1$ enforce that x_1 is larger than y_1. (2) $x_2s_<z_2$, $\mathbf{x_2m_<y_2}$, and $y_2f_<z_2$ enforce that x_2 is smaller than y_2, plus (3) $x_1b_=x_2$ and $z_1b_=z_2$ that connect the previous constraints. As a simple implementation shows, algebraic closure does not detect that these constraints are inconsistent.*

This is an example that shows that INDU is not closed under constraints and we cannot detect this contradiction using the standard qualitative reasoning methods. But INDU is even more expressive. In fact, we can express all standard arithmetic operations over the rational numbers in INDU and solve equations over the rational numbers by deciding consistency of a set of INDU constraints.

INDU Arithmetic

Addition and subtraction can already be expressed in the IA by using a triple of constraints xsz, xmy, and yfz. We get $x + y = z$ and $z - x = y$ without even knowing the lengths of the intervals. These are the implicit constraints we introduced earlier. Multiplication $y = n * x$ can be obtained by concatenating n intervals with length x, i.e, $x_1m_=x_2m_= \dots m_=x_n$, The interval y with $x_1s_<y$ and $x_nf_<y$ is the result. Division $y = x/m$ works equally by dividing interval x into m intervals with equal length y: $y_1m_=y_sm_= \dots m_=y_m$, $y_1s_<x$, $y_mf_<x$.

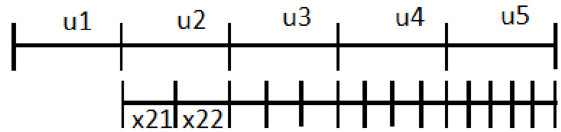

Figure 3: Constructing rational numbers using INDU

These operations alone are not very useful as we do not know the length of any of the intervals. Also, we need to be able to enumerate a certain number of intervals. However, we can recursively generate intervals of any length ℓ where ℓ is a rational number and u_1 is an interval of unit length 1 using the following algorithm.

Algorithm 11 (Construction of rational numbers) *$\mathcal{U}$ is the set of all unit length intervals, $\mathcal{N}$ the set of all integer length intervals, $\mathcal{I}$ is the set of all intervals, $\mathcal{C}$ is the set of INDU constraints over $\mathcal{I}$, and u_1 is the unit interval (see Fig. 3). k_{max} is the maximum integer length up to which we generate intervals.*

1. $\mathcal{U} = \{u_1\}$, $\mathcal{N} = \{n_1\}$, $\mathcal{I} = \{u_1, n_1\}$, $\mathcal{C} = \{u_1 = n_1\}$
2. $k := 2$
3. While $k < k_{max}$ do
4. *We introduce two fresh intervals u_k and n_k as follows:*
$\mathcal{U} := \mathcal{U} \cup \{u_k\}$, $\mathcal{N} := \mathcal{N} \cup \{n_k\}$, $\mathcal{I} := \mathcal{I} \cup \{u_k, n_k\}$
$\mathcal{C} := \mathcal{C} \cup \{u_{k-1}m_=u_k, u_1s_<n_k, u_kf_<n_k\}$,
5. *For each interval $u_i \in \mathcal{U}$, we add two fresh intervals x_{ki} and y_{ki} to $\mathcal{I}$*
and add the following constraints to $\mathcal{C}$:
a. if $i = 1$, then we add $x_{ki}s_<u_k$ and $y_{ki} = x_{ki}$,
b. if $1 < i \leq k$, add $x_{ki-1}m_=x_{ki}$, $y_{ki}s_<u_k$, and $x_{ki}f_<y_{ki}$,
c. if $i = k$, then we add $x_{ki}f_<u_k$.
6. *k:=k+1*

After running this procedure, $\mathcal{N}$ contains an interval with length k for every integer $k < k_{max}$, in particular $length(n_k) = k$. $\mathcal{I}$ contains an interval of length ℓ for every rational number $\ell = n/m$, where n and m are integers and $n < m$. Then the interval $y_{m,n}$ has length ℓ. If $n > m$, n and m are integers and $n/m = k + n'/m$, where k and n' are integers and $n' < m$, then we can get an interval of length ℓ by adding n_k and $y_{m,n'}$. Note that the procedure works without being able to count, it works purely symbolically by comparing elements of sets. As such we can construct the concept of rational numbers from INDU relations.

If we repeat this procedure up to a given maximum k-value $k = k_{max} - 1$, then we denote the resulting sets of intervals and constraints as $\mathcal{N}_k$, $\mathcal{I}_k$, $\mathcal{U}_k$ and $\mathcal{C}_k$. In this case the longest interval we get has length k and the smallest interval length $1/k$. While this procedure generates all rational numbers systematically, we do not need all these intervals if we only want to generate a certain rational number $\ell = n/m$ represented by the interval $y_{m,n}$. We only need the unit intervals in $\mathcal{U}_m$ plus the intervals x_{mi} for $1 \leq i \leq m$. We call this set of intervals $\mathcal{G}^m$, the intervals that generate rational numbers over m. The corresponding set of constraints is called $\mathcal{G}^m_{\mathcal{C}}$.

All intervals in $\mathcal{I}$ are to the right of u_1, so we can now represent any equation over the rational numbers using the space to the left of u_1. In order to have an interval x of rational length ℓ, we can write the constraint $xb_= y_\ell$ where y_ℓ is the interval in our above defined structure that has length ℓ. We can combine these intervals using the addition, subtraction, multiplication and division constraints we defined above and express any equation over the rational numbers.

Example 12 *Consider the equation $1/3+1/7 = 1/2$, which is obviously wrong. The corresponding set Θ of INDU constraints should be recognized as inconsistent. Θ contains the constraints $xs_< z$, $x\{m_<, m_=, m_>\}y$, and $yf_< z$, where x corresponds to $1/3$, y to $1/7$, and z to $1/2$. This can be represented as $xb_= y_{3,1}$, $yb_= y_{7,1}$ and $zb_= y_{2,1}$. In addition, we have to add the consistent sets of constraints $\mathcal{G}_\mathcal{C}^3$, $\mathcal{G}_\mathcal{C}^7$, and $\mathcal{G}_\mathcal{C}^2$ to Θ, as otherwise we cannot enforce the desired lengths of the intervals. It should be clear from example 10 that even though Θ is inconsistent, the traditional INDU reasoning method will not detect the inconsistency.*

As shown in Example 10, algebraic closure does not decide consistency for atomic INDU constraints. This is not very surprising, since INDU is so expressive, that it even allows us to do arithmetic calculation over the rational numbers. However, despite this expressivity, and despite the unavailability of algebraic closure as a sufficient qualitative reasoning method, we can still solve INDU instances using qualitative reasoning alone. In the next section we show how we can make use of implicit entities and implicit constraints and solve INDU using a simple qualitative reasoning algorithm that is similar to algebraic closure.

Reasoning with Implicit Constraints

In the previous section we demonstrated the expressiveness of INDU and showed that even simple and obvious inconsistencies cannot be detected by the standard qualitative reasoning methods. In this section we show how implicit entities and implicit constraints can be used to decide consistency of INDU instances in a purely qualitative way.

Example 13 *We first look at the initial INDU example 10. If we consider the implicit constraints (1) $x_1+y_1 = z_1$ and (2) $x_2 + y_2 = z_2$ together with the explicit constraints $x_1 = x_2$ and $z_1 = z_2$, then we can replace all x_1 with x_2 and all z_1 with z_2 and get (3) $x_2 + y_1 = z_2$. By combining (2) and (3) we get $y_1 = y_2$ and can replace all y_1 with y_2. The explicit constraint $x_1 < y_1$ turns into $x_2 < y_2$ which is a contradiction to the explicit constraint $x_2 > y_2$. Note that we did not do any actual arithmetic calculation, we only detect and replace symbols that are equal. For this simple example we did not need implicit entities, only implicit constraints.*

The next example is more complex and demonstrates how we can use implicit constraints and implicit entities for solving arithmetic equations.

Example 14 *Solving example 12 is more complicated. We first determine all implicit entities which effectively means we introduce an interval between any two endpoints of intervals in Θ. We then add to Θ all the applicable implicit constraints for all implicit and explicit intervals. This includes normal INDU constraints, but also constraints of type $a + b = c$. In particular, we get $x + y = z$, but we also get this kind of equations for the three intervals $y_{3,1}, y_{2,1}$ and $y_{7,1}$. For example for $y_{3,1}$ we have the intervals $x_{3,1}, x_{3,2}, x_{3,3}, u_3$, plus the interval $x^2_{3,1}$ that is formed by $x_{3,1}$ and $x_{3,2}$ and the interval $x^2_{3,2}$ formed by $x_{3,2}$ and $x_{3,3}$. We know that $y_{3,1} = x_{3,1}$ and $xb_= y_{3,1}$. The equations for these intervals are $x_{3,1}+x_{3,2} = x^2_{3,1}$, $x_{3,2}+x_{3,3} = x^2_{3,2}$, $x^2_{3,1} + x_{3,3} = u_3$, $x^2_{3,2} + x_{3,1} = u_3$. For $y_{7,1}$ we get intervals up to $x^7_{7,1}$, and get for example $x^7_{7,1} + x_{7,7} = u_7$. Using the following transformations, we obtain that $x + y = z$ is inconsistent: 1. We know that $x + x = x^2_{3,1}$, $y + y = x^2_{7,1}$, and $z + z = u_2$ and from our assumption $x + y = z$ we get (a) $x^2_{3,1} + x^2_{7,1} = u_2$. 2. We know that $x^2_{3,1} + x_{3,3} = u_2$ and $x^2_{7,1} + y^5_{7,3} = u_2$ and using (a) we get (b) $x_{3,3} = x^2_{7,1}$ and (c) $x^2_{3,1} = y^5_{7,3}$. 3. We know that $x^2_{7,1} + x^2_{7,1} = x^4_{7,1}$ and using (b) get (d) $x_{3,3} + x_{3,3} = x^4_{7,1}$. 4. From $x_{3,3} + x_{3,3} = x^2_{3,1}$ and (c) and (d) we get (e) $x^4_{7,1} = x^5_{7,3}$ which is a contradiction, since $x^5_{7,3} = x^5_{7,1}$ which is larger than $x^4_{7,1}$.*

Again, we did not do any arithmetic calculation, but only detected symbols that are equal by comparing $a+b = c$ with $a+d = c$ in which case we derive that $b = d$. In addition we used the following rule: if $a + b = c$, $d + e = f$, $a + d = g$, $b + e = h$, and $f + g = i$, then $g + h = i$. We also replace intervals that have the same length.

Before presenting a qualitative reasoning algorithm for these types of implicit constraints, we prove how adding the implicit entities and the implicit constraints affect a given set of constraints.

Lemma 15 *Given an atomic set Θ of IA constraints and the corresponding set of (non-atomic) constraints Θ_{IA}. After applying the algebraic closure algorithm to Θ_{IA}, the resulting set Θ'_{IA} is atomic.*

Proof Sketch. The set of endpoints used for instantiating variables in Θ_{IA} is exactly the same as for those of Θ since each implicit entity is defined via two explicit entities. Whenever the exact base relation between an explicit and an implicit interval is not known, we can take an explicit interval with the same endpoint as the implicit interval and resolve the ambiguity with respect to this endpoint via this triple of intervals. The algebraic closure algorithm applies this method recursively until all base relations have been derived between all intervals. Once all relations between implicit and explicit intervals have been derived, algebraic closure will derive base relations between pairs of implicit intervals as well. ■

We can now add the implicit size constraints Θ_{IS} to Θ_{IA}.

Lemma 16 *Given an atomic set of IA constraints Θ, and the corresponding sets Θ_{IA} and Θ_{IS}. Algebraic closure decides consistency of $\Theta_{IA} \cup \Theta_{IS}$.*

The previous lemma is trivial given that algebraic closure decides IA atomic relations. We now outline the proof of our main results that forms the basis for solving INDU in a qualitative way. This result makes use of the fact that INDU is equivalent to IA plus a set of relative size constraints.

Theorem 17 *Given an atomic set of INDU constraints Θ and the corresponding set of IA constraints Θ_{IA}, the corresponding set of PA constraints Θ_{PA} and the corresponding set of implicit size constraints Θ_{IS}. If Θ is algebraically closed, then Θ is consistent if and only if $\Theta_{PA} \cup \Theta_{IS}$ is consistent.*

Proof Sketch. From the previous lemma we know that the implicit size constraints do not affect the interval constraints and we assume that the implicit PA constraints do not affect the IA constraints either. This is guaranteed by the algebraic closure of Θ, otherwise it would be trivial to detect Θ as inconsistent. If $\Theta_{PA} \cup \Theta_{IS}$ is inconsistent, it is clear that Θ cannot be consistent. If $\Theta_{PA} \cup \Theta_{IS}$ is consistent, we can construct a consistent instantiation of Θ as follows:

1. Since Θ_{IA} is consistent, we can compute a canonical solution θ for it using only integers such that each successive integer belongs to at least one endpoint. It is clear that some of the PA relations on the durations might not be satisfied.
2. We can arbitrarily increase or decrease the distance between two conscutive endpoints a and b without affecting consistency of Θ_{IA} and by the previous lemma also without affecting consistency of Θ_{IS}. This changes the length of intervals that include both a and b without affecting the length of other intervals.
3. Since we consider all implicit intervals, each interval between the consecutive endpoints of θ has been considered. If $\Theta_{PA} \cup \Theta_{IS}$ is consistent, and therefore Θ_{PA} is consistent, there is an instantiation of durations to all intervals between consecutive endpoints in θ that satisfies $\Theta_{PA} \cup \Theta_{IS}$. Likewise, all intervals consisting of multiple of those intervals will satisfy $\Theta_{PA} \cup \Theta_{IS}$ as well.
4. We can now adjust the duration of each interval between consecutive endpoints in θ to the values that satisfy $\Theta_{PA} \cup \Theta_{IS}$. Since changing the length of intervals in θ does not affect Θ_{IA}, this will also satisfy Θ_{IA} and is therefore a consistent instantiation of Θ.

■

We now present a qualitative algorithm for deciding whether $\Theta_{PA} \cup \Theta_{IS}$ is consistent. Note that it is straightforward to solve this in polynomial time using either standard linear programming methods (for example using Khachian's linear programming algorithm (Khachian 1979)) or using the Horn method presented by Balbiani et al (Balbiani et al. 2006) who proved that deciding atomic INDU relations is tractable. But this is not the point of our paper. We want to show that it can be solved purely qualitatively by making the implicit entities and implicit constraints explicit. We will only use constraints of type $a + bRc$ (from Θ_{IS}) and aRb (from Θ_{PA}), that is, we have a system of **l**inear **i**nequalities with at most **3** variables per inequality, where all variables are non-negative and all coefficients are 1. Most importantly, and this is what makes the method qualitative, we will only use known implicit and explicit entities and will not do any arithmetic calculation.

Algorithm 18 (LI3-consistency) *Given an a-closed atomic set of INDU constraints Θ over the variables $\mathcal{V}$ and the corresponding a-closed sets of implicit and explicit constraints Θ_{IA}, Θ_{IS} and Θ_{PA} over the implicit and explicit variables $\mathcal{V}_I$. We set $\Sigma = \Theta_{IS}$, $\sigma = \Theta_{PA}$, and $\mathcal{V}_W = \mathcal{V}_I$ and complete both Σ and σ, i.e., we add $a + b\{<,>,=\}c$ to Σ for all triples $a, b, c \in \mathcal{V}_W$ that are not yet in Σ, and add $a\{<,>,=\}b$ to σ for all pairs $a, b \in \mathcal{V}_W$ that are not yet in σ.*

1. *For all $a, b \in \mathcal{V}_W$ do: If $a = b \in \sigma$, then σ-add($a = b$);*
2. *For all $a, b, c \in \mathcal{V}_W$ do:*
   If $a > c, a = c, b > c$, or $b = c$ in σ, then Σ-add($a + b > c$);
3. *Change := true;*
4. *While Change = true do:*
5. *Change := false;*
6. *For all $a, b, c, d \in \mathcal{V}_W$ do:*
   - *i. If $(a + b = c), (a + bRd) \in \Sigma$, then σ-add(cRd);*
   - *ii. If $(a + b < c), (a + b > d) \in \Sigma$, then σ-add($c > d$);*
   - *iii. If $(a + b = c), (a + dRc) \in \Sigma$, then σ-add(dRb);*
   - *iv. If $(a + b < c), (a + d > c) \in \Sigma$, then σ-add($b < d$);*
   - *v. If $(a + b = c) \in \Sigma$ and $dRb \in \sigma$, then Σ-add($a + dRc$);*
   - *vi. If $(a + b = c) \in \Sigma$ and $cRd \in \sigma$, then Σ-add($a + bRd$);*
   - *vii. If $(a + bRc) \in \Sigma$ and $dRb \in \sigma$, then Σ-add($a + dRc$);*
   - *viii. If $(a + bRc) \in \Sigma$ and $cRd \in \sigma$, then Σ-add($a + bRd$);*
7. *Return "LI3-consistent";*

The different add *functions are defined as follows:*

- *σ-add(aRb):*
  - *i. When aRb is added to σ, we intersect it with the existing constraint $aSb \in \sigma$. If $T = R \cap S = \emptyset$, then return "inconsistent";*
  - *ii. If $R = \{=\}$, we remove b from $\mathcal{V}_W$ and consecutively remove each occurrence bRc or cRb of σ for all $c \in \mathcal{V}_W$ and respectively add aRc or cRa to σ. Likewise, we consecutively remove each $b + cRd$, $c + bRd$, or $c + dRb$ from Σ for all $c, d \in \mathcal{V}_W$ and respectively add $a + cRd$, $c + aRd$, or $c + dRa$ to Σ; Change := true; Return;*
  - *iii. If $T = R \cap S \neq S$, then replace aSb with aTb and $bS^{-1}a$ with $bT^{-1}a$ in σ; Change := true;*
  - *iv. If $T \subset \{>,=\}$, then Σ-add($a + c > b$) for all $c \in \mathcal{V}_W \setminus \{a, b\}$;*
  - *v. If $T = \{<\}$, then Σ-add($b + c > a$) for all $c \in \mathcal{V}_W \setminus \{a, b\}$;*
- *Σ-add($a + bRc$):*
  - *i. When $a + bRc$ is added to Σ, we intersect it with the existing constraint $a + bSc \in \Sigma$. If $R \cap S = \emptyset$, we return "inconsistent".*
  - *ii. If $T = R \cap S \neq S$, then we replace $a + bSc$ with $a + bTc$ in Σ; Change := true;*

iii. If $T \subset \{<, =\}$, then σ-add($a < c$) and σ-add($b < c$);

The LI3-algorithm works similar to the algebraic closure algorithm and computes the relations $a + bRc$ (and aSb) for all triples (and pairs) of implicit and explicit entities a, b, c until no further changes can be made and no inconsistency occurs. It is purely qualitative and operates only on the existing entities.

Since Θ is atomic, each constraint in Σ and σ can be changed at most once (from $R = \{<, >, =\}$ to either $\{<\}, \{>\}$, or $\{=\}$. Therefore, the algorithm terminates after at most n^3 loops, where $n = |\mathcal{V}_I|$. By using a weighted queue of changed triples similar to the algebraic closure algorithm, the performance of the algorithm can be improved.

Theorem 19 *The LI3-Consistency Algorithm decides consistency of sets of atomic INDU constraints.*

Proof Sketch. We prove by induction over the number n of different endpoints that Algorithm 18 computes the strongest constraint $a + bRc$ for each triple of intervals $a, b, c \in \mathcal{V}_I$ that is entailed by Θ. *Base case:* The algorithm includes all possible inferences for four intervals, i.e., for the base case of up to $n \leq 8$ endpoints, which occurs if four explicit intervals have no endpoint in common. In cases where the explicit intervals have some points in common, we can transform any situation with up to 8 different endpoints to an equivalent situation where those endpoints are taken up by the 4 explicit intervals. This is because we only look at atomic constraints and always know all IA base relations between all explicit and implicit intervals as of Lemma 15.

Induction step: We now assume that the strongest constraints are obtained for $n = k$ different endpoints and show that they are also inferred for $n = k + 1$ different endpoints. For a new endpoint, we get k new implicit or explicit intervals between the new endpoint and the other k endpoints. Since we know all the basic IA relations, we can derive the size relations between the new intervals. Any other unknown size relations can be obtained by using at most four intervals, which is covered by the base case. Since we know the strongest implied size constraint for any k endpoints, it follows that the newly derived constraints will also be the strongest.

If the algorithm terminates and returns "inconsistent", then clearly Θ cannot be consistent. If it returns "LI3-consistent", then we have obtained the strongest implicit size constraints that can be inferred from Θ. Therefore, we have a partial order on the sizes of intervals in $\mathcal{V}_W$ according to σ, and can now assign values starting from the smallest intervals and fix values of larger intervals according to the constraints in Σ. None of these assignments will contradict $\sigma \cup \Sigma$ as otherwise they would not be the strongest constraints. It follows that $\Theta_{IS} \cup \Theta_{PA}$ is consistent, and because of Theorem 17, Θ will be consistent too. ■

The proof we sketched above works because Lemma 15 guarantees that we know all IA base relations between all implicit and explicit intervals. Consequently, for any three endpoints we have the three intervals that can be formed using these endpoints and get an IS constraint of type $a+b = c$. This gives us any intermediate intervals and information about their sizes that we need to compute the strongest implicit size constraints. In cases where Θ is not atomic, we cannot guarantee that the LI3-Consistency algorithm computes the strongest IS constraints and we might only get an approximation.

Conclusions

There are numerous examples of qualitative spatial or temporal calculi where the standard qualitative reasoning methods fail. There have been attempts to explain this behaviour, but it is largely unclear when and why this happens, how it can be avoided and what can be done about it. Due to the significance of being able to guarantee correct qualitative reasoning results, this is one of the major challenges in the field.

In this paper we take up this challenge, offer an explanation and propose a possible solution to this important problem. Our goal in this paper is not to present a method that works for all qualitative spatial or temporal calculi – not even algebraic-closure does that. Our goal is not to present the fastest (nor even an efficient) algorithm to solve INDU – there are other known algorithms that can do that, and this is also why we do not optimize our algorithm, do not analyse it or prove its complexity, and do not empirically evaluate it.

We do have several goals in this paper. One goal is to demonstrate that implicit entities and implicit constraints exist and are typically ignored. We show this for a number of well-known calculi. One goal is to show that implicit entities and implicit constraints can be responsible in cases where existing qualitative reasoning methods fail and we give two examples where this is the case.

One goal is to show that by making implicit entities and implicit constraints explicit and by adding them to the qualitative representation, we can potentially solve problems qualitatively that could not be solved qualitatively before. We give two example where this is possible. For the simple Containment Algebra it is possible but not necessary. The second example is INDU, which seems to be a relatively simple extension of the Interval Algebra, but which turns out to be so expressive that it can even encode arithmetic over the rational numbers – and therefore seems like a highly unlikely candidate for being able to be solved qualitatively. Despite this, we show that by making implicit entities and implicit constraints explicit, we can solve INDU qualitatively using a novel qualitative reasoning algorithm that works similar to algebraic closure.

Our final goal is to show that qualitative reasoning can be much more powerful than previously thought. We want to show that a failure of algebraic closure for atomic constraint networks is not the end of qualitative reasoning, but that there is much more that can be done.

We believe that we have reached our goals in this paper and hope that the analysis of implicit entities and implicit constraints will become the new standard in qualitative spatial and temporal reasoning research. We believe that such an analysis will be essential for two of the main challenges in this research area: (1) for the integration of different calculi in order to form more expressive and more useful representations, and (2) for the analysis of calculi where algebraic closure fails.

INDU is a good example where integrating two well-behaved calculi leads to difficulties and we expect that the same will happen for many other combinations of calculi. One example that demonstrates the benefit of making implicit constraints explicit is our most recent work on developing a qualitative representation for video analysis (Cohn et al. 2012). In that paper we show that by using the implicit intervals of the Interval Algebra, we can obtain a compact and very comprehensive qualitative representation that integrates a number of different calculi. Regarding the second challenge, there are not many constructive results available yet and the problem remains largely unsolved. While in some cases we can fall back to quantitative methods, we expect that this is not always possible due to the difficulty of adequately representing qualitative information quantitatively.

In addition to working on these challenges, future work includes analysing the use of implicit entities and constraints for non-atomic relations and developing an implicit representation for existing calculi for which qualitative reasoning gives incorrect results. Our long term goal is to obtain standard algorithms for dealing with different kinds of implicit entities and implicit constraints, ideally something similar to algebraic closure. Since the LI3-consistency algorithm deals with some fundamental properties of space and time that are likely to affect other calculi as well, it might serve as a good starting point for future analysis.

Acknowledgements

This research was supported under Australian Research Council's Future Fellowships funding scheme (project number FT0991917).

References

J. F. Allen. 1983. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM* 26(11):832-843.

P. Balbiani, J-F. Condotta, G. Ligozat. 2006. On the Consistency Problem for the INDU Calculus. *Journal of Applied Logic*, 4(2): 119-140.

A. G. Cohn, J. Renz and M. Sridhar. 2012. Thinking Inside the Box: A Comprehensive Spatial Representation for Video Analysis. Proceedings of KR'12.

A. G. Cohn and J. Renz. 2008. Qualitative Spatial Representation and Reasoning. *Handbook of Knowledge Representation*, Elsevier, 551-596.

M. J. Egenhofer, R. D. Franzosa. 1991. Point Set Topological Relations. *International Journal of Geographical Information Science*, 5, 161-174.

A. Gerevini and J. Renz. 2002. Combining Topological and Size Constraints for Spatial Reasoning. *Artificial Intelligence*, 137(1-2):1-42.

L. G. Khachian. 1979. A Polynomial Time Algorithm for Linear Programming. *Soviet Math. Dokl.*, 20:191-194.

P. B. Ladkin and R. D. Maddux. 1994. On Binary Constraint Problems. *Journal of the ACM* 41(3):435-469.

G. Ligozat and J. Renz. 2004. What Is a Qualitative Calculus? A General Framework. *Proceedings of PRICAI'04*, 53-64.

A. K. Mackworth 1977. Consistency in Networks of Relations. *Artificial Intelligence* 8(1):99-118.

B. Nebel and H.-J. Bürckert. 1995. Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *Journal of the ACM* 42(1): 42-66.

A. K. Pujari, G. V. Kumari and A. Sattar. 1999. INDU: An Interval and Duration Network. In *Proceedings of AusAI'99*, 291-303.

D. A. Randell, Z. Cui and A. G. Cohn. 1992. A Spatial Logic based on Regions and Connection. *Proceedings of KR'92*, 165-176.

J. Renz. 2007. Qualitative Spatial and Temporal Reasoning: Efficient Algorithms for Everyone. *Proceedings of IJCAI'07*, 526-531.

J. Renz and G. Ligozat. 2005. Weak Composition for Qualitative Spatial and Temporal Reasoning. In *Proceedings of CP'05*, 534–548.

P. van Beek and D. W. Manchak 1996. The Design and Experimental Analysis of Algorithms for Temporal Reasoning. *Journal of Artificial Intelligence Research* 4:1-18.

M. Vilain, H. A. Kautz, and P. van Beek. 1990. Constraint Propagation Algorithms for Temporal Reasoning: a Revised Report. *Readings in qualitative Reasoning about Physical Systems*, 373-381, Morgan Kaufmann.