Customizing Qualitative Spatial and Temporal Calculi

Jochen Renz[1] and Falko Schmid[2]

[1] RSISE, The Australian National University, Canberra, ACT 0200, Australia
[2] SFB/TR8 Spatial Cognition, Universität Bremen, 28334 Bremen, Germany

Abstract. Qualitative spatial and temporal calculi are usually formulated on a particular level of granularity and with a particular domain of spatial or temporal entities. If the granularity or the domain of an existing calculus doesn't match the requirements of an application, it is either possible to express all information using the given calculus or to customize the calculus. In this paper we distinguish the possible ways of customizing a spatial and temporal calculus and analyze when and how computational properties can be inherited from the original calculus. We present different algorithms for customizing calculi and proof techniques for analyzing their computational properties. We demonstrate our algorithms and techniques on the Interval Algebra for which we obtain some interesting results and observations. We close our paper with results from an empirical analysis which shows that customizing a calculus can lead to a considerably better reasoning performance than using the non-customized calculus.

1 Introduction

Qualitative Spatial and Temporal Representation and Reasoning (QSTR) [1] is often useful when exact properties of spatial or temporal entities (e.g. exact location) are not important, but when we are interested in the relationships between different entities. A qualitative spatial or temporal calculus typically takes a particular set of spatial or temporal entities as its domain and defines relations over it to represent an aspect of space or time on a particular level of granularity.

Reasoning over these relations is usually done by exploiting *composition* of relations and the main reasoning problem is the *consistency problem* which asks if a given set of constraints over a set of relations has an instantiation which satisfies all the constraints. The consistency problem is NP-hard for most calculi if all relations are allowed, but in some cases tractable if only the base relations are used. Many spatial and temporal calculi have been defined in the literature [1] and they are all based on the same principles of having a particular domain and base relations over a particular aspect of space and time on a particular level of granularity. Some of these calculi have been intensively analyzed and computational properties are known for many of them.

Let us assume an application for which we want to represent qualitative spatial or temporal information and want to efficiently reason about this information. One problem we might face when selecting an existing calculus is that our application has a different domain than the existing calculus or requires relations on a different level of granularity. One way of dealing with this problem is to use the existing calculus and to ignore the differences. This is not very useful as the computational properties of the existing calculus only hold for the given domain and granularity. Another possibility is

M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 293–304, 2007.

to develop a new calculus which is tailored towards our application. In this paper we look at a third possibility, namely, to customize an existing spatial or temporal calculus to a given application or to some given requirements. The result will also be a tailored calculus, but in many cases it will be possible to simply inherit known computational properties from the existing calculus.

Customizing calculi is not new and there are several examples where customizations of spatial or temporal calculi have been suggested. The best known example is probably RCC5 which is a subcalculus of RCC8 on a coarser level of granularity [3]. In this paper we go a step further and analyze customization in general. We distinguish different ways of customizing calculi, discuss properties which customized calculi have to satisfy, present algorithms for generating coarser calculi which satisfy these properties, and show when and how computational properties of coarser calculi can be inherited from existing calculi. We demonstrate our algorithms and methods on the Interval Algebra (IA) and derive some interesting results and observations. The advantage of using the IA is that we know all maximal tractable subsets of the IA [6] and can therefore easily test our general methods.

2 Qualitative Spatial and Temporal Calculi

A binary qualitative spatial or temporal calculus takes a domain of spatial or temporal entities $\mathcal{D}$ and defines a set of base relations $\mathcal{B}$ that partition $\mathcal{D} \times \mathcal{D}$ into jointly exhaustive and pairwise disjoint sets. Between any two values of the domain, exactly one of the base relations holds. Indefinite information can be expressed by using the union ($\cup$) of base relations, one base relation of the union must hold, but it is not yet known which of them. The set of all relations is therefore the powerset of the base relations $2^{\mathcal{B}}$. Usually the operators converse ($\smile$, or conv()), intersection ($\cap$), complement ($\neg$), and most importantly composition ($\circ$) are defined. Composition of two relations R, S is the relation defined as follows: $R \circ S = \{(a,c) | \exists b.(a,b) \in R$ and $(b,c) \in S\}$. Special relations are the universal relation U which is the union of all base relations, the empty relation $\emptyset$ and the identity relation id. A spatial or temporal calculus is a set of relations $2^{\mathcal{B}}$ which is closed under the operators, i.e., applying the operators to all relations always results in relations of the same set. Due to the definition of the relations, it is clear that $2^{\mathcal{B}}$ is always closed under converse, union, intersection, and complement, but it might not be closed under composition. For some calculi it is therefore necessary to use weak composition ($\circ_w$) instead of composition, which is defined as follows: $R \circ_w S = \{T \in \mathcal{B} | T \cap (R \circ S) \neq \emptyset\}$. $2^{\mathcal{B}}$ is always closed under weak composition.

Spatial and temporal information is usually represented using constraints over the relations, e.g., the constraint xRy, where x, y are variables over the domain $\mathcal{D}$ and $R \in 2^{\mathcal{B}}$, is satisfied if there is an instantiation of x and y with values $a, b \in \mathcal{D}$ such that $(a, b) \in R$. Given a set Θ of such constraints, an important reasoning problem is whether Θ is consistent, i.e., whether there are instantiations of all variables in Θ such that all constraints are satisfied. We write the consistency problem as CSPSAT($\mathcal{S}$) to indicate that only relations of the set $\mathcal{S}$ are used in Θ. This is a constraint satisfaction problem which is NP-hard in general. In order to enable efficient solutions to the consistency problem, the minimum requirement is that CSPSAT($\mathcal{B}$) is tractable. Ideally,

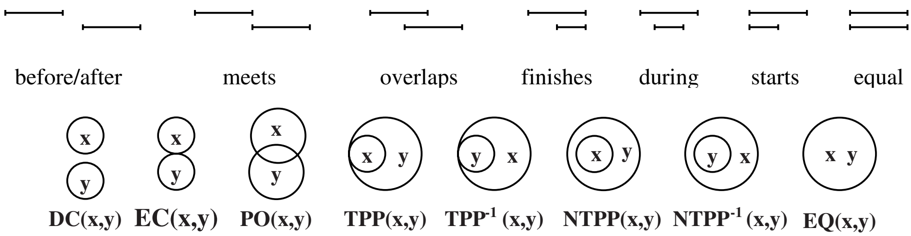


Fig. 1. Illustrations of the base relations of the Interval Algebra and of RCC8

we can identify large tractable subsets $\mathcal{T} \subseteq 2^{\mathcal{B}}$ for which $\mathsf{CSPSAT}(\mathcal{T})$ is tractable[18]. A maximal tractable subset is a tractable subset $\mathcal{T}$ such that CSPSAT is NP-hard for each superset of $\mathcal{T}$. If a tractable subset contains all base relations, it can be used to considerably speed up solving $\mathsf{CSPSAT}(2^{\mathcal{B}})$ [4].

A simple way of approximating consistency of a set Θ is to use the path-consistency algorithm which makes Θ path-consistent by applying the following operation to each triple of variables x, y, z of Θ until either a fixed point is reached or the empty relation is obtained (R_{xy} is the relation between x and y): $\forall x, y, z : R_{xy} := R_{xy} \cap (R_{xz} \circ R_{zy})$. If a calculus uses weak composition instead of composition, we have to use weak composition ($\circ_w$) in the given operation, and the corresponding algorithm is called *algebraic-closure algorithm* [5]. If weak composition is equal to composition, then algebraic-closure is equal to path-consistency. This operation can be performed in cubic time in the number of variables. If the empty relation occurs, then Θ is inconsistent, otherwise the resulting set Θ' is path-consistent/algebraically closed.

The best known spatial and temporal calculi are RCC8 [3] and the IA [7] (see Figure 1). The domain of RCC8 are extended spatial regions, defined as regular subsets of a topological space. RCC8 consists of eight base relations that distinguish topological relationships between spatial regions: DC (disconnected), EC (externally connected), PO (partial overlap), TPP (tangential proper part), NTPP (non-tangential proper part), their converses TPP^{-1} and NTPP^{-1}, and the identity relation EQ (equal).

The domain of the IA are intervals, defined as convex sets of a one-dimensional directed space. The IA consists of thirteen base relations: before ($<$), meets (m), overlaps (o), during (d), starts (s), finishes (f), their converse relations mi, oi, di, si, fi, and the identity relation equal ($=$). The IA is closed under composition and there is one maximal tractable subset (ORD-Horn) that contains all base relations [8] and 17 maximal tractable subsets which do not contain all base relations [6].

3 Customizing Spatial and Temporal Calculi

Qualitative spatial and temporal calculi use a particular domain $\mathcal{D}$ and partition $\mathcal{D}\times\mathcal{D}$ in a particular way into a set of base relations. The computational properties of a calculus hold only for the given choice of domain and base relations. If we have an application for which we want to use a spatial or temporal calculus it is possible that our application uses a different domain or requires different distinctions than those made by an existing calculus. This means that the computational properties of the existing calculus might not apply to our application; we have to develop a new calculus and prove its computational properties. In many cases, however, it is possible to customize an existing calculus to meet some more restricted requirements on domain or distinguished relationships, and to inherit computational properties from the existing calculus.

Assume that we have an application where we want to represent relations between intervals but where all the interval endpoints have to be distinct. As Golumbic and Shamir point out [9], this assumption is frequently made in combinatorics and has also been proved useful in applications of temporal reasoning. Under this assumption, we do not need the granularity offered by the Interval Algebra and would like to use coarser relations. There are two ways of how we can customize the Interval Algebra:

1. We use what Golumbic and Shamir call *macro relations* [9], i.e., unions of base relations. We combine IA base relations and use these macro relations as base relations of our customized calculus. For example, we could combine $\{m, o\}$, $\{mi, oi\}$, $\{s, d, f\}$, $\{si, di, fi\}$ and use them together with the relations $\{<\}$, $\{>\}$, and $\{=\}$ as our new base relations. This corresponds to the algebra $\mathcal{A}_7$ defined in [9].
2. We use only the relations we need, namely, the interval relations $<, >, d, di, o, oi,$ $=$ and do not use m, mi, s, si, f, fi which correspond to intervals with common endpoints. This is similar to the algebra $\mathcal{A}_6$ defined in [9] with the exception of the identity relation which we include in order to be able to use the standard constraint satisfaction algorithms.

In both cases we use only seven "base relations" and their unions (i.e., 2^7 relations). But in both cases we get unwanted relations if we close these sets under composition, intersection and converse which happens if we compute path-consistency or apply other reasoning methods for the given set of constraints. In the first case, the composition of $\{<\}$ and $\{s, d, f\}$, for example, gives the relation $\{<, d, o, m, s\}$ which is outside the relations we wanted to use as f is not included. In the second case we get the same behavior, the composition of $<$ and d gives the relation $\{<, d, o, m, s\}$ which includes the base relations m and s which we did not want to use. In both cases, the relations are not closed under the operators and we end up having to use the full Interval Algebra again. This shows that even though it is straightforward to use a finer calculus for representing coarser information, we need to do a proper customization in order to benefit from having to represent only coarser information.

A good example is the RCC5 calculus consisting of five base relations: DR (discrete), PO (partial overlap), PP (proper part), its converse PP^{-1} and the identity relation EQ. RCC5 is closed under weak-composition and also under the other operators. Two of the RCC5 relations, PO and EQ are the same as the RCC8 relations, DR, PP, and PP^{-1} are macro relations: $\mathsf{DR} = \mathsf{DC} \cup \mathsf{EC}$, $\mathsf{PP} = \mathsf{TPP} \cup \mathsf{NTPP}$, $\mathsf{PP}^{-1} = \mathsf{TPP}^{-1} \cup \mathsf{NTPP}^{-1}$. RCC5 uses the same domain as RCC8 but uses base relations on a different level of granularity. RCC5 can be regarded as a customized version of RCC8 and many of the computational properties of RCC5 can be derived from computational properties of RCC8 as all 2^5 RCC5 relations are contained in RCC8.

But it is not always that simple. Consider the RCC7 calculus [2] which is a customized version of the RCC8 calculus where regions cannot overlap. The base relations of RCC7 are the same as those of RCC8 with the only difference that the RCC8 base relation PO is not allowed and not used. It has been shown that the consistency problem for RCC7 is NP-hard even if only the RCC7 base relations are used.

Another example are the 9-intersection relations [10]. RCC8 and the 9-intersection relations have a similar meaning and the composition tables are identical. However, the domain of RCC8 are regular regions of an arbitrary topological space while the domain of the 9-intersection relations a 2-D regions which are homeomorphic to disks, i.e., they consist of one piece and have no holes. The 9-intersection relations can be regarded as a customized version of RCC8 for a particular kind of spatial entities.

The previous examples can all be considered as customized versions of other calculi. We have seen three different kinds of customizations: (1) Combining base relations to form macro relations, (2) Excluding some of the base relations, (3) Restricting the domain. We can now summarize and formalize the different kinds of customization, and what basic requirements must be met by a customized calculus.

Definition 1 (Coarser Calculus). *Given a qualitative spatial or temporal calculus $\mathcal{F}$ with base relations $\mathcal{B}_\mathcal{F}$ over a domain $\mathcal{D}_\mathcal{F}$. A calculus C with base relations $\mathcal{B}_\mathcal{C}$ and universal relation $U_\mathcal{C}$ over a domain $\mathcal{D}_\mathcal{C}$ is called* coarser than $\mathcal{F}$, *written as $\mathcal{C} < \mathcal{F}$, iff (1) for each base relation $B_F \in \mathcal{B}_\mathcal{F}$ there is a base relation $B_C \in \mathcal{B}_\mathcal{C}$ such that $(B_F \cap U_\mathcal{C}) \subseteq B_C$, (2) $\mathcal{D}_\mathcal{C} \subseteq \mathcal{D}_\mathcal{F}$, and one of the following conditions holds:*

(a) macro relations: *there exists a base relation B_F in $\mathcal{B}_\mathcal{F}$ such that $(B_F \cap U_\mathcal{C}) \subset B_C$,*
(b) unused relation: *there exists a base relation B_F in $\mathcal{B}_\mathcal{F}$ such that $(B_F \cap U_\mathcal{C}) = \emptyset$,*
(c) reduced domain: *$\mathcal{D}_\mathcal{C} \subset \mathcal{D}_\mathcal{F}$.*

In this paper we define the customization of an existing calculus $\mathcal{F}$ as the process of finding a coarser calculus $\mathcal{C}$ which better matches particular requirements about domain and granularity than $\mathcal{F}$. Note that here we lift the requirement that base relations are jointly exhaustive in order to allow the possibility of having unused relations independently of restricting the domain. This is done by assuming that the universal relation U of a calculus is the union of all the base relations which can be smaller than the cross-product of the domain $\mathcal{D} \times \mathcal{D}$ which typically defines the universal relation. Furthermore the coarser calculus has all the properties of a normal calculus, i.e., it is closed under the operators, and is therefore different from tractable subsets of a calculus.

4 Customization Using Macro Relations

We have seen in the previous section that we cannot use any partition of base relations into macro relations in order to obtain a coarser calculus because the new calculus might not be closed under the operators. We implemented an algorithm which enumerates all possible partitions of a given set of base relations into macro relations and tests whether they are closed under the operators or not.

Proposition 1. *Given a set $\mathcal{B}$ of base relations which consists of two types of relations: (1) the relations $S_1, \ldots, S_k \in \mathcal{B}$ are equivalent to their own converse relation, i.e., $conv(S_i) = S_i$, (2) the relations $C_1, \ldots, C_h \in \mathcal{B}$ have a converse relation among themselves, i.e., $conv(C_i) = C_j$ for some $i \neq j$. All macro relations R of a valid partition of $\mathcal{B}$ must satisfy the following properties:*

– *If R contains a relation S_i, then for every relation B_j which is contained in R, the relation $conv(B_j)$ must also be contained in R.*
– *If R contains only relations C_i and $R \neq conv(R)$ then $conv(R)$ must also be contained in the partition.*

It is a mathematically interesting question to compute the number of partitions that satisfy the proposition, i.e., the number of partitions for any number h, k with $h + k = n$, but this is outside the scope of this paper. In Figure 2 we give an algorithm for computing all partitions of a set of base relations $\mathcal{B}$ which are closed under the operators. All the resulting partitions can be used as customized calculi. For the IA it turns out that there are only 16033 partitions which satisfy the proposition and only 117 of them are closed under the operators.

Algorithm: FINDMACROCALCULI($\mathcal{P}, \mathcal{R}$)
Input: a partial partition $\mathcal{P}$ of the base relations $\mathcal{B}$ and the remaining base relations $\mathcal{R}$

1. *if* $\mathcal{R} = \emptyset$ *then*
2. *if* $closed(\mathcal{P})$ *then* print $\mathcal{P}$;
3. select one base relation $B \in \mathcal{R}$;
4. *for all* macro relations S of $\mathcal{R}$ for which $B \in S$ *do*
5. *if* $S \cap conv(S) \neq \emptyset$ *and* $S \neq conv(S)$ *then continue*;
6. FINDMACROCALCULI($\mathcal{P} \cup \{S, conv(S)\}, \mathcal{R} \setminus \{S, conv(S)\}$)

Fig. 2. The recursive algorithm FINDMACROCALCULI computes all partitions of $\mathcal{B}$ which are closed under the operators when started with $\mathcal{P} = \emptyset$ and $\mathcal{R} = \mathcal{B}$

What we are interested in now, is when and how we can inherit computational properties from an existing calculus $\mathcal{F}$ to a customized calculus $\mathcal{C}$. We can only refer to computational properties if they are known for the original calculus, so we assume that we know one or more tractable subsets of the original calculus. For the case of having macro relations, inheriting computational properties is simple as all macro relations are also contained in the powerset of the original base relations $\mathcal{B}_{\mathcal{F}}$. Whenever the base relations $\mathcal{B}_{\mathcal{C}}$ of a customized calculus $\mathcal{C}$ are contained in a tractable subset of the original calculus, the intersection of $\mathcal{C}$ with any tractable subset of $\mathcal{F}$ is also tractable.

Lemma 1. *Given a calculus $\mathcal{F}$ with base relations $\mathcal{B}_{\mathcal{F}}$ over a domain $\mathcal{D}_{\mathcal{F}}$ and a known tractable subset $\mathcal{T}_{\mathcal{F}}$ of $\mathcal{F}$. If a calculus $\mathcal{C}$ has the same domain as $\mathcal{F}$, $\mathcal{C} < \mathcal{F}$, and all relations of $\mathcal{B}_{\mathcal{C}}$ are macro relations of $\mathcal{B}_{\mathcal{F}}$, then $\mathcal{C} \cap \mathcal{T}_{\mathcal{F}}$ is a tractable subset of $\mathcal{C}$. If algebraic closure decides* CSPSAT*($\mathcal{T}_{\mathcal{F}}$), then it also decides* CSPSAT*($\mathcal{C} \cap \mathcal{T}_{\mathcal{F}}$).*

The algorithm in Figure 2 can be modified to check if a customized calculus is contained in the known tractable subsets and to compute the intersection with the known tractable subsets. If the base relations of a customized calculus are macro relations of an existing calculus $\mathcal{F}$ and not contained in a known tractable subset of $\mathcal{F}$, then it is only possible to derive its complexity if all tractable subsets of $\mathcal{F}$ are known. Otherwise, its complexity has to be proved independently.

Table 1. All partitions of the Interval Algebra where deciding consistency is tractable

$\mathcal{M}_1 = \{=, < d\, o\, m\, s, > di\, oi\, mi\, si, f, fi\}$	$\mathcal{M}_2 = \{=, < d\, o\, m\, s, > di\, oi\, mi\, si, f\, fi\}$
$\mathcal{M}_3 = \{=, <> d\, di\, o\, oi\, m\, mi\, s\, si, f, fi\}$	$\mathcal{M}_4 = \{=, <> d\, di\, o\, oi\, m\, mi\, s\, si, f\, fi\}$
$\mathcal{M}_5 = \{=, < d\, o\, m\, s\, f, > di\, oi\, mi\, si\, fi\}$	$\mathcal{M}_6 = \{=, s, si, > d\, oi\, mi\, f, < di\, o\, m\, fi\}$
$\mathcal{M}_7 = \{=, s\, si, > d\, oi\, mi\, f, < di\, o\, m\, fi\}$	$\mathcal{M}_8 = \{=, > di\, oi\, mi\, si\, f, < d\, o\, m\, s\, fi\}$
$\mathcal{M}_9 = \{=, > d\, oi\, mi\, si\, f, < di\, o\, m\, s\, fi\}$	$\mathcal{M}_{10} = \{=, > d\, oi\, mi\, s\, f, < di\, o\, m\, si\, fi\}$
$\mathcal{M}_{11} = \{=, s, si, <> d\, di\, o\, oi\, m\, mi\, f\, fi\}$	$\mathcal{M}_{12} = \{=, s\, si, <> d\, di\, o\, oi\, m\, mi\, f\, fi\}$
$\mathcal{M}_{13} = \{=, <> d\, di\, o\, oi\, m\, mi\, s\, si\, f\, fi\}$	$\mathcal{M}_{14} = \{=\ s\, si, > d\, oi\, mi\, f, < di\, o\, m\, fi\}$
$\mathcal{M}_{15} = \{=\ s\, si, <> d\, di\, o\, oi\, m\, mi\, f\, fi\}$	$\mathcal{M}_{16} = \{< d\, o\, m\, s, > di\, oi\, mi\, si, =\ f\, fi\}$
$\mathcal{M}_{17} = \{<> d\, di\, o\, oi\, m\, mi\, s\, si, =\ f\, fi\}$	$\mathcal{M}_{18} = \{= <> d\, di\, o\, oi\, m\, mi\, s\, si\, f\, fi\}$

For the IA all maximal tractable subsets are known [6] and we can determine the computational properties for all 117 customized calculi we identified above. We applied our modified algorithm to the IA and tested for all 117 coarser calculi whether they are contained in any of the maximal tractable subsets. The result was surprising: (1) For 18 of the 117 valid partitions, all base relations are contained in a maximal trabtable subset. (2) For all 18, the powerset of the relations is also contained in a maximal tractable subset, i.e., the 117 customized calculi of the IA are either completely tractable or completely NP-hard. (3) All 18 are contained in ORD-Horn, the only maximal tractable subset which contains all base relations. Table 1 gives all 18 tractable partitions.

5 Customization by Unused Relations

For every spatial or temporal domain $\mathcal{D}$, we can find base relations which are jointly exhaustive and pairwise disjoint, i.e., for any two entities there will be exactly one base relation which holds for every pair. The universal relation, which is the union of all base relations is then equal to $\mathcal{D} \times \mathcal{D}$. If one or more of the base relations R_i cannot occur due to requirements of an application, then there are two possibilities of modifying a set of base relations $\mathcal{B}$. The first possibility is to say that the relations R_i are forbidden, i.e., they can not occur in any solution of the consistency problem over $2^{\mathcal{B}}$. We have to make sure that whenever we are forced to use a relation R_i, then our set of constraints is inconsistent. Alternatively, the second possibility is to say that the base relations R_i are empty and can be removed. This has the consequence that the base relations are not jointly exhaustive anymore and that the universal relation is not equal to $\mathcal{D} \times \mathcal{D}$ anymore, but it has the advantage of making reasoning simpler as we can just remove the unused relations. In the following we will always remove unused relations.

Given a calculus $\mathcal{F}$ with the base relations $\mathcal{B}_{\mathcal{F}}$, there are $2^{\mathcal{B}} - 1$ possibilities of removing unused base relations in order to obtain a coarser calculus $\mathcal{C}$. If we remove one or more base relations, we have to make sure that these relations are removed from all the compositions of relations of $\mathcal{C}$. All the relations which we do not remove are exactly the same relations as those of $\mathcal{F}$. When we remove base relations, the universal relation $U_{\mathcal{C}}$ of $\mathcal{C}$ is the union of all remaining base relations. Composition of relations C_i, C_j of $\mathcal{C}$ is therefore defined as follows: $C_i \circ_{\mathcal{C}} C_i = (C_i \circ_{\mathcal{F}} C_j) \cap U_{\mathcal{C}}$, where $\circ_{\mathcal{C}}$ is

Algorithm: FINDUNUSEDCALCULI($\mathcal{B}$)
Input: a set of base relations $\mathcal{B}$

for all relations $U \in 2^{\mathcal{B}}$ with $U = conv(U)$
  $\mathcal{N} = \emptyset$
  for all relations $B \in \mathcal{B}$
    if $\{B\} \cap U \neq \emptyset$ then $\mathcal{N} := \mathcal{N} \cup \{B, conv(B)\}$
  for all pairs of relations $N_i, N_j \in \mathcal{N}$ with $i \neq j$ do
    if $(N_i \circ N_j) \cap U = \emptyset$ *then* $\mathcal{N} = \emptyset$; break;
    $\mathcal{N} := \mathcal{N} \cup \{conv(N_i)\} \cup (\{N_i\} \cap \{N_j\}) \cup (N_i \circ N_j)$
  if $\mathcal{N} \neq \emptyset$ *then* print $\mathcal{N}$

Fig. 3. Algorithm FINDUNUSEDCALCULI computes all customized calculi of $\mathcal{B}$ which have unused relations and which satisfy Proposition 2

the composition of relations of $\mathcal{C}$ while $\circ_{\mathcal{F}}$ is the composition of relations of $\mathcal{F}$. Since the relations (and their converses) which we remove with this operation are empty, it is obvious that this way of computing composition is correct.

Another requirement a coarser calculus has to satisfy is that the composition of two relations must not give the empty relation. This can occur if the composition of two relations contains only unused relations. The following proposition lists the properties that a coarser calculus resulting from unused relations has to satisfy.

Proposition 2. *Given a set $\mathcal{B}_{\mathcal{F}}$ of base relations from which we remove the unused relations $S_1, \ldots, S_k \in \mathcal{B}$, resulting in the new set of base relations $\mathcal{B}_{\mathcal{C}}$. The relations $R_i, R_j \in \mathcal{C}$ must satisfy the following properties:*

(1) If $R_i \in \mathcal{C}$ then $conv(R_i) \in \mathcal{C}$; (2) $R_i \circ_{\mathcal{C}} R_j \neq \emptyset$.

The algorithm in Figure 3 computes all coarser calculi $\mathcal{C}$ of a given calculus $\mathcal{F}$ which can be obtained by removing unused relations and which satisfy Proposition 2. We applied the algorithm to the IA and found that 63 calculi satisfied the requirements.

We cannot just compare the coarser calculi $\mathcal{C}$ with the tractable subsets of the original calculus $\mathcal{F}$. The reason for this is that the closure of subsets of $\mathcal{C}$ will be different from the closure of the same subsets of $\mathcal{F}$, and therefore it is possible that the same set of relations has different computational properties depending on whether it is a subset of $\mathcal{C}$ or of $\mathcal{F}$. But we are able to derive some properties if we look at what the actual closure of a set of relations of $\mathcal{C}$ is.

Lemma 2. *Given a calculus $\mathcal{F}$ and known tractable subsets $\mathcal{T}^i_{\mathcal{F}}$ of $\mathcal{F}$ for which algebraic closure decides consistency. Let $\mathcal{C}$ be a coarser calculus of $\mathcal{F}$ with base relations $\mathcal{B}_{\mathcal{C}}$ which results from $\mathcal{F}$ by removing unused base relations. If the closure $\widehat{\mathcal{B}_{\mathcal{C}}}$ of $\mathcal{B}_{\mathcal{C}}$ is contained in one of the sets $\mathcal{T}^i_{\mathcal{F}}$, then* CSPSAT*($\widehat{\mathcal{B}_{\mathcal{C}}}$) is tractable and can be decided by the algebraic closure algorithm.*

Proof. Given a set Θ of constraints over $\widehat{\mathcal{B}_{\mathcal{C}}}$ and let Θ' be the result of applying algebraic closure to Θ. If the empty relation occurs while computing algebraic closure, then Θ is inconsistent. Since $\widehat{\mathcal{B}_{\mathcal{C}}}$ is closed, all constraints of Θ' will also be from $\widehat{\mathcal{B}_{\mathcal{C}}}$. All

$\mathcal{R}_0^* = \{=\}$, $\mathcal{R}_1 = \{<,>\}$, $\mathcal{R}_2 = \{=,<,>\}$, $\mathcal{R}_3 = \{d, di\}$, $\mathcal{R}_4 = \{=, d, di\}$, $\mathcal{R}_5 = \{o, oi\}$, $\mathcal{R}_6 = \{=, o, oi\}$, $\mathcal{R}_7^* = \{d, di, o, oi\}$, $\mathcal{R}_8^* = \{=, d, di, o, oi\}$, $\mathcal{R}_9^* = \{<,>, d, di, o, oi\}$, $\mathcal{R}_{10}^* = \{=,<,>, d, di, o, oi\}$, $\mathcal{R}_{11}^* = \{=,<,>, d, di, o, oi, m, mi\}$, $\mathcal{R}_{12}^* = \{s, si\}$, $\mathcal{R}_{13}^* = \{=, s, si\}$, $\mathcal{R}_{14} = \{<,>, s, si\}$, $\mathcal{R}_{15} = \{=,<,>, s, si\}$, $\mathcal{R}_{16} = \{d, di, s, si\}$, $\mathcal{R}_{17} = \{=, d, di, s, si\}$, $\mathcal{R}_{18} = \{o, oi, s, si\}$, $\mathcal{R}_{19} = \{=, o, oi, s, si\}$, $\mathcal{R}_{20}^* = \{d, di, o, oi, s, si\}$, $\mathcal{R}_{21}^* = \{=, d, di, o, oi, s, si\}$, $\mathcal{R}_{22}^* = \{<,>, d, di, o, oi, s, si\}$, $\mathcal{R}_{23}^* = \{=,<,>, d, di, o, oi, s, si\}$, $\mathcal{R}_{24}^* = \{=,<,>, d, di, o, oi, m, mi, s, si\}$, $\mathcal{R}_{25}^* = \{f, fi\}$, $\mathcal{R}_{26}^* = \{=, f, fi\}$, $\mathcal{R}_{27} = \{<,>, f, fi\}$, $\mathcal{R}_{28} = \{=,<,>, f, fi\}$, $\mathcal{R}_{29} = \{d, di, f, fi\}$, $\mathcal{R}_{30} = \{=, d, di, f, fi\}$, $\mathcal{R}_{31} = \{o, oi, f, fi\}$, $\mathcal{R}_{32} = \{=, o, oi, f, fi\}$, $\mathcal{R}_{33}^* = \{d, di, o, oi, f, fi\}$, $\mathcal{R}_{34}^* = \{=, d, di, o, oi, f, fi\}$, $\mathcal{R}_{35}^* = \{<,>, d, di, o, oi, f, fi\}$, $\mathcal{R}_{36}^* = \{=,<,>, d, di, o, oi, f, fi\}$, $\mathcal{R}_{37}^* = \{=,<,>, d, di, o, oi, m, mi, f, fi\}$, $\mathcal{R}_{38}^* = \{d, di, o, oi, s, si, f, fi\}$, $\mathcal{R}_{39}^* = \{=, d, di, o, oi, s, si, f, fi\}$, $\mathcal{R}_{40}^* = \{<,>, d, di, o, oi, s, si, f, fi\}$, $\mathcal{R}_{41}^* = \{=,<,>, d, di, o, oi, s, si, f, fi\}$, $\mathcal{R}_{42}^* = \{<,>, d, di, o, oi, m, mi, s, si, f, fi\}$

Fig. 4. The 43 customized calculi of the Interval Algebra with unused relations and tractable base relations. The closure of the sets marked with $*$ are contained in ORD-Horn.

relations of $\widehat{\mathcal{B}_\mathcal{C}}$ are exactly the same relations as the corresponding relations of $\mathcal{F}$, so let $\Theta_\mathcal{F} = \Theta'$ be the corresponding set of constraints over $\mathcal{F}$. Since $R_i \circ_\mathcal{C} R_j \subseteq R_i \circ_\mathcal{F} Rj$, $\Theta_\mathcal{F}$ is also algebraically closed. Since $\widehat{\mathcal{B}_\mathcal{C}}$ is contained in a tractable subset of $\mathcal{F}$ for which algebraic closure decides consistency, $\Theta_\mathcal{F} = \Theta'$ is consistent.

Note that if $\widehat{\mathcal{B}_\mathcal{C}}$ is contained in a tractable subset of $\mathcal{F}$ for which algebraic closure does not decide consistency, then $\mathsf{CSPSAT}(\widehat{\mathcal{B}_\mathcal{C}})$ is still tractable. We added this test to our implementation of the algorithm of Figure 3 and applied it to the IA. It turned out that for 25 of the 63 coarser calculi $\widehat{\mathcal{B}_\mathcal{C}}$ is contained in ORD-Horn, and for 18 further coarser calculi, $\widehat{\mathcal{B}_\mathcal{C}}$ is contained in one of the other maximal tractable subsets of the IA. The 43 tractable calculi are given in Figure 4. We can obtain larger tractable subsets of our coarser calculi simply by intersecting them with the known tractable subsets of $\mathcal{F}$. This can be proved in the same way as Lemma 2.

6 Customization by Restricting the Domain

The computational properties of a qualitative calculus strongly depend on the used domain, as can be seen with RCC8 and 9-intersection. 9-intersection has a much more restricted domain than RCC8, but until recently it was unknown if the consistency problem for the 9-intersection problem is decidable at all [11]. This shows that reducing the domain does not necessarily make a calculus simpler. Other restrictions of the RCC8 domain that have been analyzed are the restriction of the domain to convex regions [12] and to closed disks [13]. A well-known example for a domain restriction in the interval domain is to use only intervals whose endpoints are integers.

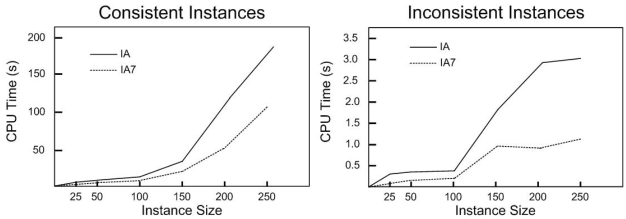
Due to the strong dependence of computational properties to the domain, there are no general methods as in the previous sections for inheriting complexity results to coarser calculi. Instead each customization requires a separate analysis. In the following we will give guidelines about how to inherit complexity results from a calculus $\mathcal{F}$ to a coarser calculus $\mathcal{C}$ with a reduced domain. First of all, we have to analyze how the composition is affected by removing values from the domain. (1) If it turns out that some base relations will not be used anymore, we can first apply the methods discussed in the previous section and then continue our analysis using the calculus and the results we obtain by applying these methods. (2) If the weak composition of the relation stays the same, we can test if the reduced domains allow us to refine some of the base relations to non-overlapping sub-relations. This is a method introduced and discussed in more detail in [14]. If this is not possible (as in the case of intervals over integer endpoints), then the computational properties of the coarser calculus $\mathcal{C}$ are the same as those of $\mathcal{F}$. If it is possible, then algebraic closure does not decide consistency even for the base relations of $\mathcal{C}$ and the computational properties of $\mathcal{F}$ can not be inherited . (3) If only some of the compositions change without completely removing some of the base relations of $\mathcal{F}$, then we have to make a new analysis and cannot inherit results from $\mathcal{F}$.

7 Benefits of Customization

Having a calculus with a smaller number of base relations has several benefits over using a calculus with a larger number of base relations. For larger calculi such as the Directed Intervals Algebra [15] it might not be possible to store the complete composition table in memory. Instead we might always have to compute composition from the basic composition table, which leads to slower reasoning. Independent of composition tables, using the smaller calculus is much more efficient because of the way the reasoning algorithms work, namely, by splitting relations into sub-relations and backtracking over the different sub-relations [16]. For a smaller calculus, relations can be split at most into the coarser base relations, but for larger calculi they are split into the finer base relations leading to much bigger backtracking trees and consequently much slower reasoning.

We empirically compared the reasoning performance of a coarser calculus with that of a finer calculus on exactly the same instances, i.e., we randomly generated only instances that can be expressed by the coarser calculus and then expressed the same instances with the finer calculus. We used a coarser calculus of the IA where intervals cannot have endpoints in common and compare it with the IA. The coarser calculus has seven base relations $<, >, o, oi, d, di, =$ and we call it IA7. It is obtained from the IA by removing all unused relations. Deciding consistency is tractable for the IA7 base relations, but NP-hard for the full calculus. We randomly generated 100 IA7 instances for different sizes. The constraints between the nodes of an instance were chosen randomly from IA7 and had an average density of $d = 5(\pm 2)$ constraints per node, which marks the *phase transition region* [17] for IA7. For our experiments we used the SFB/TR8 Generic Qualitative Reasoner[1] for checking consistency. The experiments where implemented on a Linux-PC with a 1,5Ghz VIA C7 CPU and 1GB RAM.

[1] https://sfbtr8.informatik.uni-freiburg.de/R4LogoSpace/Resources/GQR



As we can see from the graphs, we can solve the same instances much faster if we use IA7 for solving them instead of IA. In most cases we are more than twice as fast. This shows that customizing a calculus to a particular application can lead to considerable performance gains and is therefore highly recommended.

8 Conclusions

Sometimes applications require a coarser representation of spatial or temporal information than what existing calculi offer. In this paper we show different possibilities of how existing calculi can be customized to form coarser calculi which might be more suited for a given application. We present general algorithms for generating coarser calculi. We show when and how we can derive computational properties for coarser calculi from existing results and demonstrate empirically that using customized calculi leads to a much better reasoning performance than using non-customized calculi. We demonstrate our general methods on the IA for which we identify several coarser calculi and easily derive their computational properties.

References

1. Cohn, A.G., Renz, J.: Qualitative spatial Representation and Reasoning. In: Van Hermelen, F., Lifschitz, V., Porter, B.B. (eds.) Handbook of Knowledge Representation, Elsevier, Amsterdam (2007)
2. Gerevini, A., Renz, J.: Combining topological and size information for spatial reasoning. Artificial Intelligence 137(1-2), 1–42 (2002)
3. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Proc. of KR 1992, pp. 165–176 (1992)
4. Renz, J., Nebel, B.: Efficient Methods for Qualitative Spatial Reasoning. Journal of Artificial Intelligence Research 15, 289–318 (2001)
5. Ligozat, G., Renz, J.: What is a Qualitative Calculus? A General Framework. In: Zhang, C., Guesgen, H.W., Yeap, W.K. (eds.) PRICAI 2004. LNCS (LNAI), vol. 3157, pp. 53–64. Springer, Heidelberg (2004)
6. Krokhin, A., Jeavons, P., Jonsson, P.: Reasoning about temporal relations: The tractable subalgebras of Allen's Interval Algebra. J. ACM 50(5), 591–640 (2003)
7. Allen, J.F.: Maintaining knowledge about temporal intervals. CACM 26(11), 832–843 (1983)
8. Nebel, B., Bürckert, H.J.: Reasoning about temporal relations: A maximal tractable subclass of Allen's interval algebra. Journal of the ACM 42(1), 43–66 (1995)
9. Golumbic, M.C., Shamir, R.: Complexity and algorithms for reasoning about time: A graph-theoretic approach. Journal of the ACM 40(5), 1128–1133 (1993)

10. Egenhofer, M.J.: Reasoning about binary topological relations. In: Günther, O., Schek, H.-J. (eds.) SSD 1991. LNCS, vol. 525, pp. 143–160. Springer, Heidelberg (1991)
11. Schaefer, M., Sedgwick, E., Stefankovic, D.: Recognizing string graphs in NP. Journal of Computer and System Sciences (2003)
12. Davis, E., Gotts, N.M., Cohn, A.G.: Constraint networks of topological relations and convexity. CONSTRAINTS 4(3), 241–280 (1999)
13. Düntsch, I., Schmidt, G., Winter, M.: A necessary relation algebra for mereotopology. Studia Logica 69(3), 381–409 (2001)
14. Renz, J., Ligozat, G.: Weak composition for qualitative spatial and temporal reasoning. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 534–548. Springer, Heidelberg (2005)
15. Renz, J.: A Spatial Odyssey of the Interval Algebra: 1.Directed Intervals. In: Proc. IJCAI 2001, pp. 51–56 (2001)
16. Nebel, B.: Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ORD-Horn class. CONSTRAINTS 3(1), 175–190 (1997)
17. Cheeseman, P., Kanefsky, B., Taylor, W.M.: Where the *really* hard problems are. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence, pp.331–337 (1991)
18. Renz, J.: Qulitative Spatial and Temporal Reasoning: Efficient Algorithms for Everyone. In: IJCAI -2007. Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (January 2007)