

# A Generic Toolkit for $n$ -ary Qualitative Temporal and Spatial Calculi

Jean-François Condotta<sup>1</sup>      Gérard Ligozat<sup>2</sup>      Mahmoud Saade<sup>1</sup>

<sup>1</sup> *CRIL-CNRS, Université d'Artois, 62307 Lens Cedex, France*

<sup>2</sup> *LIMSI-CNRS, Université d'Orsay, 91403 Orsay, France*  
*ligozat@limsi.fr, {condotta, saade}@cril.univ-artois.fr*

**Abstract.** Temporal and spatial reasoning is a central task for numerous applications in many areas of Artificial Intelligence. For this task, numerous formalisms using the qualitative approach have been proposed. Clearly, these formalisms share a common algebraic structure. In this paper we propose and study a general definition of such formalisms by considering calculi based on basic relations of an arbitrary arity. We also describe the **QAT** (the Qualitative Algebra Toolkit), a JAVA constraint programming library allowing to handle constraint networks based on those qualitative calculi.

**Keywords:** Qualitative temporal and spatial reasoning, constraint reasoning.

## 1 Introduction

Numerous qualitative constraint calculi have been developed in the past in order to represent and reason about temporal and spatial configurations. Representing and reasoning about spatial and temporal information is an important task in many applications, such as computer vision, geographic information systems, natural language understanding, robot navigation, temporal and spatial planning, diagnosis and genetics. Qualitative spatial and temporal reasoning aims to describe non-numerical relationships between spatial or temporal entities. Typically a qualitative calculus [1–5] uses some particular kind of spatial or temporal objects (*e.g.* subsets in a topological space, points on the rational line, intervals on the rational line) to represent the spatial or temporal entities of the system, and focuses on a limited range of relations between these objects (such as topological relations between regions or precedence between time points). Each of these relations refers to a particular temporal or spatial configuration. For instance, in the field of qualitative reasoning about temporal data, consider the well known formalism called Allen's calculus [1]. It uses intervals of the rational line for representing temporal entities. Thirteen basic relations between these intervals are used to represent the qualitative situation between temporal entities. An interval can be before the other one, can follow the other one, can end the other one, and so on. The thirteen basic relations are JEPD (jointly exhaustive and pairwise disjoint), which means that each pair of intervals satisfies exactly one basic relation.

Constraint networks called qualitative constraint networks (QCNs) are usually used to represent the temporal or spatial information about the configuration of a specific set of entities. Each constraint of a QCN represents a set of acceptable qualitative configurations between some temporal or spatial entities and is defined by a set of basic relations. The consistency problem for QCNs consists in deciding whether a given network has instantiations satisfying the constraints. In order to solve it, methods based on local constraint propagation algorithms have been defined, in particular methods based on various versions of the path consistency algorithm [6, 7].

In the literature most qualitative calculi are based on basic binary relations. These basic relations

are always JEPD. Moreover, the operators of intersection, of composition and of inverse used for reasoning with these relations are always defined in a similar way. Hence we can assert that these qualitative calculi share the same structure. Recently, non binary qualitative calculi have been proposed. The difference between binary calculi and non binary calculi resides in the fact that new operators are necessary for the non binary case, namely the operator of permutation and the operator of rotation.

In this paper we propose and study a very general definition of a qualitative calculus. This definition subsumes all qualitative calculi used in the literature. Moreover, to our knowledge, implementations and software tools have only been developed for individual calculi. The QAT (Qualitative Algebra Toolkit) has been conceived as a remedy to this situation. Specifically, the QAT is a JAVA constraint programming library developed at *CRIL-CNRS* at the University of Artois. It aims to provide open and generic tools for defining and manipulating qualitative algebras and qualitative networks based on these algebras.

This paper is organized as follows. In Section 2, we propose a formal definition of a qualitative calculus. This definition is very general and it covers formalisms based on basic relations of an arbitrary arity. Section 3 is devoted to qualitative constraint networks. After introducing the QAT library in Section 4, we conclude in Section 5.

## 2 A general definition of Qualitative Calculi

### 2.1 Relations and fundamental operations

A qualitative calculus of arity  $n$  (with  $n > 1$ ) is based on a finite set  $\mathcal{B} = \{B_1, \dots, B_k\}$  of  $k$  relations of arity  $n$  defined on a domain  $D$ . These relations are called basic relations. Generally,  $k$  is a small integer and the set  $D$  is an infinite set, such as the set  $\mathbb{N}$  of the natural numbers, the set  $\mathbb{Q}$  of the rational numbers, the set of real numbers, or, in the case of Allen's calculus, the set of all intervals on one of these sets. We will denote by  $\mathcal{U}$  the set of  $n$ -tuples on  $D$ , that is, elements of  $D^n$ . Moreover, given an element  $x$  belonging to  $\mathcal{U}$  and an integer  $i \in \{1, \dots, n\}$ ,  $x_i$  will denote the element of  $D$  corresponding to the  $i^{\text{th}}$  component of  $x$ . The basic relations of  $\mathcal{B}$  are complete and jointly exclusive, in other words, the set  $\mathcal{B}$  must be a partition of  $\mathcal{U} = D^n$ , hence we have:

**Property 1**  $B_i \cap B_j = \emptyset, \forall i, j \in \{1, \dots, k\}$  such that  $i \neq j$  and  $\mathcal{U} = \bigcup_{i \in \{1, \dots, k\}} B_i$ .

Given a set  $\mathcal{B}$  of basic relations, we define the set  $\mathcal{A}$  as the set of all unions of the basic relations. Formally, the set  $\mathcal{A}$  is defined by  $\mathcal{A} = \{\bigcup B : B \subseteq \mathcal{B}\}$ .

In the binary case, the various qualitative calculi considered in the literature consider a particular basic relation corresponding to the identity relation on  $D$ . We generalise this by assuming that a qualitative calculus of arity  $n$  satisfies the following property:

**Property 2**  $\forall i, j \in \{1, \dots, n\}$  such that  $i \neq j$ ,  $\Delta_{ij} \in \mathcal{A}$  with  $\Delta_{ij} = \{x \in \mathcal{U} : x_i = x_j\}$ .

Note that the relations  $\Delta_{ij}$  are called diagonal elements in the context of cylindric algebras [8]. Given a non empty set  $E \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$  such that for all  $(i, j) \in E$  we have  $i \neq j$ ,  $\Delta_E$  will denote the relation  $\bigcap \{\Delta_{ij} : (i, j) \in E\}$ . We note that from Property 1 and Property 2 we can deduce that  $\Delta_E \in \mathcal{A}$ . Hence, the relation of identity on  $\mathcal{U}$ , denoted by  $\text{Id}_n$ , which corresponds to  $\Delta_{\{(i, i+1) : 1 \leq i \leq n-1\}}$ , belongs to  $\mathcal{A}$ .

In the sequel we will see how to use the elements of  $\mathcal{A}$  to define particular constraint networks

called qualitative constraint networks. Several fundamental operations on  $\mathcal{A}$  are necessary for reasoning with these constraint networks, in particular, the operation of permutation, the operation of rotation and the operation of qualitative composition also simply (and wrongly) called composition or weak composition [9, 11].

In the context of qualitative calculi, the operations of permutation and rotation have been introduced by Isli and Cohn [5] for a formalism using ternary relations on cyclic orderings. These operations are unary operations which associate to each element of  $\mathcal{A}$  a relation belonging to  $\mathcal{U}$ . They can be formally defined in the following way:

**Definition 1.** *Let  $R \in \mathcal{A}$ . The permutation and the rotation of  $R$ , denoted by  $R^{\rightarrow}$  and  $R^{\curvearrowright}$  respectively, are defined as follows:*

- $R^{\rightarrow} = \{(x_1, \dots, x_{n-2}, x_n, x_{n-1}) : (x_1, \dots, x_n) \in R\}$  (Permutation),
- $R^{\curvearrowright} = \{(x_2, \dots, x_n, x_1) : (x_1, \dots, x_n) \in R\}$  (Rotation).

In the binary case, these operations coincide and correspond to the operation of converse. To our knowledge, all binary qualitative calculi satisfy the property that the converse relation of any basic relation is a basic relation. A similar property is required in the general case:

**Property 3** *For each relation  $B_i \in \mathcal{B}$  we have  $B_i^{\curvearrowright} \in \mathcal{B}$  and  $B_i^{\rightarrow} \in \mathcal{B}$ .*

These operations satisfy the following properties:

**Proposition 1.** *Let  $R \in \mathcal{A}$ .*

- $R^{\rightarrow} = \bigcup \{B^{\rightarrow} : B \in \mathcal{B} \text{ and } B \subseteq R\}$  and  $R^{\curvearrowright} = \bigcup \{B^{\curvearrowright} : B \in \mathcal{B} \text{ and } B \subseteq R\}$ .

For binary relations, the operation of composition is a binary operation which associates to two relations  $R_1$  and  $R_2$  the relation  $\circ(R_1, R_2) = \{(x_1, x_2) : \exists u \in \mathcal{D} \text{ with } (x_1, u) \in R_1 \text{ and } (u, x_2) \in R_2\}$ . For several qualitative calculi of arity  $n = 2$  the composition of two relations  $R_1, R_2 \in \mathcal{A}$  is not necessarily a relation of  $\mathcal{A}$  (consider for example the interval algebra on the intervals defined on the integers). A weaker notion of composition is used. This operation, denoted in the sequel by  $\diamond$ , and called qualitative composition, is by definition the smallest relation (*w.r.t. inclusion*) of  $\mathcal{A}$  containing all the elements of the *bona fide* composition :  $\diamond(R_1, R_2) = \bigcap \{R \in \mathcal{A} : \circ(R_1, R_2) \subseteq R\}$ . For an arbitrary arity  $n$ , composition and qualitative composition can be defined in the following way:

**Definition 2.** *Let  $R_1, \dots, R_n \in \mathcal{A}$ .*

- $\circ(R_1, \dots, R_n) = \{(x_1, \dots, x_n) : \exists u \in \mathcal{D}, (x_1, \dots, x_{n-1}, u) \in R_1, (x_1, \dots, x_{n-2}, u, x_n) \in R_2, \dots, (u, x_2, \dots, x_n) \in R_n\}$ ,
- $\diamond(R_1, \dots, R_n) = \bigcap \{R \in \mathcal{A} : \circ(R_1, \dots, R_n) \subseteq R\}$ .

Note that we use the usual definition of the polyadic composition for the operation  $\circ$ . Both operations are characterized by their restrictions to the basic relations of  $\mathcal{B}$ . Indeed, we have the following properties:

**Proposition 2.** *Let  $R_1, \dots, R_n \in \mathcal{A}$ .*

- $\circ(R_1, \dots, R_n) = \bigcup \{\circ(A_1, \dots, A_n) : A_1 \in \mathcal{B}, \dots, A_n \in \mathcal{B} \text{ and } A_1 \subseteq R_1, \dots, A_n \subseteq R_n\}$ ;
- $\diamond(R_1, \dots, R_n) = \bigcup \{\diamond(A_1, \dots, A_n) : A_1 \in \mathcal{B}, \dots, A_n \in \mathcal{B} \text{ and } A_1 \subseteq R_1, \dots, A_n \subseteq R_n\}$ .

Another way to define the qualitative composition is given by the following proposition:

**Proposition 3.** *Let  $R_1, \dots, R_n \in \mathcal{A}$ .*

$$\diamond(R_1, \dots, R_n) = \bigcup \{A \in \mathcal{B} : \exists x_1, \dots, x_n, u \in D, \exists A_1, \dots, A_n \in \mathcal{B} \text{ with } (x_1, \dots, x_n) \in A, (x_1, \dots, x_{n-1}, u) \in A_1, (x_1, \dots, x_{n-2}, u, x_n) \in A_2, \dots, (u, x_2, \dots, x_n) \in A_n, A_1 \subseteq R_1, \dots, A_n \subseteq R_n\}.$$

Hence, tables giving the qualitative composition, the rotation and the permutation of basic relations can be used for computing efficiently these operations for arbitrary relations of  $\mathcal{A}$ . Finally, we have the following properties, which generalize the usual relationship of composition with respect to converse in the binary case:

**Proposition 4.** *Let  $R_1, \dots, R_n \in \mathcal{A}$  and  $OP \in \{\circ, \diamond\}$ .*

- $OP(\emptyset, R_2, \dots, R_n) = \emptyset$ ;
- $OP(R_1, \dots, R_n)^{\curvearrowright} = OP(R_n^{\curvearrowright}, R_1^{\curvearrowright}, R_2^{\curvearrowright}, \dots, R_{n-1}^{\curvearrowright})$ ;
- $OP(R_1, \dots, R_n)^{\curvearrowleft} = OP(R_2^{\curvearrowleft}, R_1^{\curvearrowleft}, R_3^{\curvearrowleft}, \dots, R_n^{\curvearrowleft})$ .

## 2.2 An example of a qualitative calculus of arity 3: the Cyclic Point Algebra

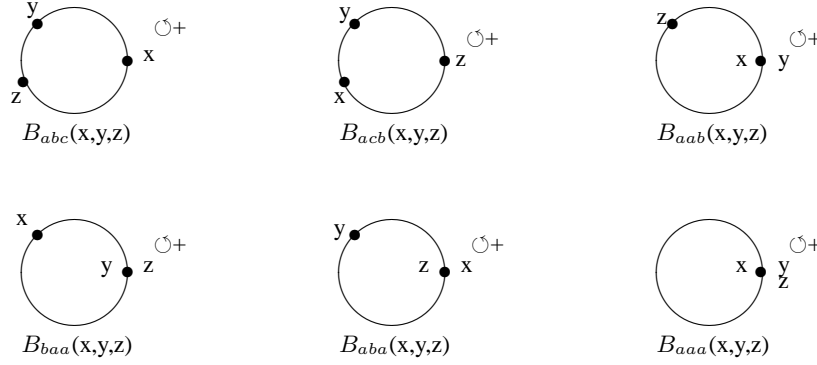
This subsection is devoted to a qualitative calculus of arity 3 known as the Cyclic Point Algebra [5, 12].

The entities considered by this calculus are the points on an oriented circle  $\mathcal{C}$ . We will call these points *cyclic points*. Each cyclic point can be characterised by a rational number belonging to the interval  $[0, 360[$ . This number corresponds to the angle between the horizontal line going through the centre of  $\mathcal{C}$ . Hence, for this calculus,  $D$  is the set of the rational numbers  $\{q \in \mathbb{Q} : 0 \leq q < 360\}$ . In the sequel we assimilate a cyclic point to the rational number representing it. Given two cyclic points  $x, y \in D$ ,  $[[x, y]]$  will denote the set of values of  $D$  corresponding to the cyclic points met between  $x$  and  $y$  when travelling on the circle counter-clockwise. The basic relations of the Cyclic Point Algebra is the set of the 6 relations  $\{B_{abc}, B_{acb}, B_{aab}, B_{baa}, B_{aba}, B_{aaa}\}$  defined in the following way:  $B_{abc} = \{(x, y, z) \in D^3 : x \neq y, x \neq z, y \neq z \text{ and } y \in [[x, z]]\}$ ,  $B_{acb} = \{(x, y, z) \in D^3 : x \neq y, x \neq z, y \neq z \text{ and } z \in [[x, y]]\}$ ,  $B_{aab} = \{(x, x, y) \in D^3 : x \neq y\}$ ,  $B_{baa} = \{(y, x, x) \in D^3 : x \neq y\}$ ,  $B_{aba} = \{(x, y, x) \in D^3 : x \neq y\}$ ,  $B_{aaa} = \{(x, x, x) \in D^3\}$ . These 6 relations are shown in Figure 1. Based on these basic relations, we get a set  $\mathcal{A}$  containing 64 relations. Note that for these basic relations the operation of composition and the operation of qualitative composition are the same operations. Table 1 gives the qualitative composition of a subset of the basic relations. Using Proposition 2, we can compute other qualitative compositions which are not given in this table. For example,  $\diamond(B_{aab}, B_{acb}, B_{abc}) = \diamond(B_{aab}, B_{abc}, B_{acb})^{\curvearrowleft} = \{B_{aab}\}$ . Actually, the table provides a way of computing any composition of basic relations, since all qualitative compositions which cannot be deduced from it in that way yield the empty relation. This is the case for example of the qualitative composition of  $B_{aaa}$  with  $B_{abc}$ , which is the empty relation.

## 3 Qualitative Constraint Networks

### 3.1 Basic notions

Typically, qualitative constraint networks (QCNs in short) are used to express information on a spatial or temporal situation. Each constraint of a constraint network represents a set of acceptable



**Fig. 1.** The 6 basic relations of the Cyclic Point Algebra.

$R_1$	$B_{aaa}$	$B_{aaa}$	$B_{aab}$	$B_{aab}$	$B_{aab}$	$B_{aab}$	$B_{aba}$	$B_{abc}$
$R_2$	$B_{aaa}$	$B_{aab}$	$B_{aba}$	$B_{abc}$	$B_{baa}$	$B_{acb}$	$B_{aab}$	$B_{abc}$
$R_3$	$B_{aaa}$	$B_{aab}$	$B_{baa}$	$B_{acb}$	$B_{aba}$	$B_{abc}$	$B_{abc}$	$B_{acb}$
$\diamond(R_1, R_2, R_3)$	$\{B_{aaa}\}$	$\{B_{aab}\}$	$\{B_{aaa}\}$	$\{B_{aab}\}$	$\{B_{aab}\}$	$\{B_{aab}\}$	$\{B_{abc}\}$	$\{B_{abc}\}$

**Table 1.** The qualitative composition of the Cyclic Point Algebra

$a$	$B_{aaa}$	$B_{aab}$	$B_{aba}$	$B_{baa}$	$B_{abc}$	$B_{acb}$
$a^{\leftrightarrow}$	$B_{aaa}$	$B_{aba}$	$B_{aab}$	$B_{baa}$	$B_{acb}$	$B_{abc}$
$a^{\curvearrowright}$	$B_{aaa}$	$B_{aba}$	$B_{baa}$	$B_{aab}$	$B_{abc}$	$B_{acb}$

**Table 2.** The permutation and the permutation operation of the Cyclic Point Algebra

qualitative configurations between some temporal or spatial entities and is defined by a set of basic relations. Formally, a QCN is defined in the following way:

**Definition 3.** A QCN is a pair  $\mathcal{N} = (V, C)$  where:

- $V$  is a finite set of  $l$  variables  $\{v'_0, \dots, v'_{l-1}\}$  (where  $l$  is a positive integer);
- $C$  is a map which to each tuple  $(v_0, \dots, v_{n-1})$  of  $V^n$  associates a subset  $C(v_0, \dots, v_{n-1})$  of the set of basic relations:  $C(v_0, \dots, v_{n-1}) \subseteq \mathcal{B}$ .  $C(v_0, \dots, v_{n-1})$  are the set of those basic relations allowed between the variables  $v_0, \dots, v_{n-1}$ . Hence,  $C(v_0, \dots, v_{n-1})$  represents the relation of  $A$  corresponding to the union of the basic relations belonging to it.

We use the following definitions in the sequel:

**Definition 4.** Let  $\mathcal{N} = (V, C)$  be a QCN with  $V = \{v'_0, \dots, v'_{l-1}\}$ .

- A **partial instantiation** of  $\mathcal{N}$  on  $V' \subseteq V$  is a map  $\alpha$  of  $V'$  on  $D$ . Such a partial instantiation is consistent if and only if  $(\alpha(v_0), \dots, \alpha(v_{n-1})) \in C(v_0, \dots, v_{n-1})$ , for all  $v_0, \dots, v_{n-1} \in V'$ .
- A solution of  $\mathcal{N}$  is a consistent partial instantiation on  $V$ .  $\mathcal{N}$  is consistent if and only if it has a solution.
- An **atomic QCN** is a QCN whose every constraint is defined by just one basic relation of  $\mathcal{B}$ .

- A **scenario** on a set of variables  $V'$  is an **atomic** QCN whose variables are the set  $V'$ . A **consistent scenario** of  $\mathcal{N}$  is a scenario that admits a solution of  $\mathcal{N}$  as a solution.
- A QCN  $\mathcal{N}' = (V', C')$  is **equivalent** to  $\mathcal{N}$  if and only if  $V = V'$  and both networks  $\mathcal{N}$  and  $\mathcal{N}'$  have the same solutions.
- A **sub-QCN** of a QCN  $\mathcal{N} = (V, C)$  is a QCN  $\mathcal{N}' = (V, C')$  where:  
 $C'(v_0, \dots, v_{n-1}) \subseteq C(v_0, \dots, v_{n-1})$  for all  $v_0, \dots, v_{n-1} \in V$ .

Moreover we introduce the definition of normalized QCNs which intuitively correspond to QCNs containing *compatible* constraints w.r.t. the fundamental operations of rotation and permutation.

**Definition 5.** Let  $\mathcal{N}$  be a QCN. Then  $\mathcal{N}$  is **normalized** iff:

- $C(v_2, \dots, v_n, v_1) = C(v_1, \dots, v_n) \curvearrowright$ ,
- $C(v_1, \dots, v_{n-2}, v_n, v_{n-1}) = C(v_1, \dots, v_n) \curvearrowright$ ,
- $C(v_1, \dots, v_i, \dots, v_j, \dots, v_n) \subseteq \Delta_{ij}, \forall i, j \in \{1, \dots, n\}$  such that  $i \neq j$  and  $v_i = v_j$ .

Given any QCN, it is easy to transform it into an equivalent QCN which is normalized. Hence we will assume that all QCNs considered in the sequel are normalized.

Given a QCN  $\mathcal{N}$ , the problems usually considered are the following: determining whether  $\mathcal{N}$  is consistent, finding a solution, or all solutions, of  $\mathcal{N}$ , and computing the smallest QCN equivalent to  $\mathcal{N}$ . These problems are generally NP-complete problems. In order to solve them, various methods based on local constraint propagation algorithms have been defined, in particular the method which is based on the algorithms of path consistency [6, 7] which we will refer to as the  $\diamond$ -closure method.

### 3.2 The $\diamond$ -closure method

This subsection is devoted to the topic of  $\diamond$ -closed QCNs. These QCNs are defined in the following way:

**Definition 6.** Let  $\mathcal{N} = (V, C)$  be a QCN. Then  $\mathcal{N}$  is  $\diamond$ -closed iff  $C(v_1, \dots, v_n) \subseteq \diamond(C(v_1, \dots, v_{n-1}, v_{n+1}), C(v_1, \dots, v_{n-2}, v_{n+1}, v_n), \dots, C(v_1, v_{n+1}, v_3, \dots, v_n), C(v_{n+1}, v_2, \dots, v_n)), \forall v_1, \dots, v_n, v_{n+1} \in V$ .

For qualitative calculus of arity two this property is sometimes called the path-consistency property or the 3-consistency property, wrongly since qualitative composition is in general weaker than composition (see [13] for a discussion to this subject). In the binary case, the usual local constraint propagation algorithms PC1 and PC2 [6, 7] have been adapted to the qualitative case for computing a sub-QCN which is  $\diamond$ -closed and equivalent to a given QCN. As an extension of PC – 1 to the  $n$ -ary case we define the algorithm PC1<sub>n</sub> (see Algorithm 1). In brief, this algorithm iterates an operation (line 7–8) which suppresses non possible basic relations from the constraints using weak composition and intersection. This operation is repeated until a fixpoint is reached. It can be easily checked that the QCN output by PC1<sub>n</sub> is  $\diamond$ -closed and equivalent to the initial QCN used as input. The time complexity of Algorithm PC1<sub>n</sub> is  $O(|V|^{(n+1)})$  where  $|V|$  is the number of variables of the QCN and  $n$  the arity of the calculus. We can prove the following properties:

**Proposition 5.** Applying the algorithm PC1<sub>n</sub> to a normalized QCN  $\mathcal{N}$  yields a QCN which is normalized,  $\diamond$ -closed, and equivalent to  $\mathcal{N}$ .

---

**Algorithm 1** PC1<sub>n</sub>

---

Compute the closure of a QCN  $\mathcal{N} = (V, C)$

```
1: Do
2:    $\mathcal{N}' := \mathcal{N}$ 
3:   For each  $v_{n+1} \in V$  Do
4:     For each  $v_1 \in V$  Do
5:       ...
6:       For each  $v_n \in V$  Do
7:          $C(v_1, \dots, v_n) := C(v_1, \dots, v_n) \cap$ 
8:            $\diamond(C(v_1, \dots, v_{n-1}, v_{n+1}), C(v_1, \dots, v_{n-2}, v_{n+1}, v_n), \dots, C(v_{n+1}, v_2, \dots, v_n))$ 
9:   Until  $(\mathcal{N} == \mathcal{N}')$ 
10: return  $\mathcal{N}$ 
```

---

In the binary case, a  $\diamond$ -closed QCN is not always 3-consistent but it is  $(0, 3)$ -consistent, which means, respectively, that we cannot always extend a partial solution on two variables to three variables, but that we know that all sub-QCNs on three variables are consistent. This last property can be extended to the  $n$ -ary case:

**Proposition 6.** *Let  $\mathcal{N} = (V, C)$  be a QCN. If  $\mathcal{N}$  is  $\diamond$ -closed then it is  $(0, n)$ -consistent.*

Note that in the same manner, we can extend PC2 to the  $n$ -ary case and prove similar results.

### 3.3 Associating a binary qualitative calculus to a qualitative calculus of arity $n$

Consider a qualitative calculus of arity  $n$ . There is actually a standard procedure for associating a binary calculus to it. Moreover, if a QCN is defined on the  $n$ -ary calculus, it can be represented by QCN in the associated binary calculus. We now proceed to sketch this procedure. Consider a qualitative calculus with a set of basic relations  $\mathcal{B} = \{B_1, \dots, B_k\}$  of arity  $n$  defined on  $D$ . We associate to it a qualitative formalism with a set of basic relations  $\mathcal{B}' = \{B'_1, \dots, B'_{k'}\}$  of arity 2 defined on a domain  $D'$  in the following way:

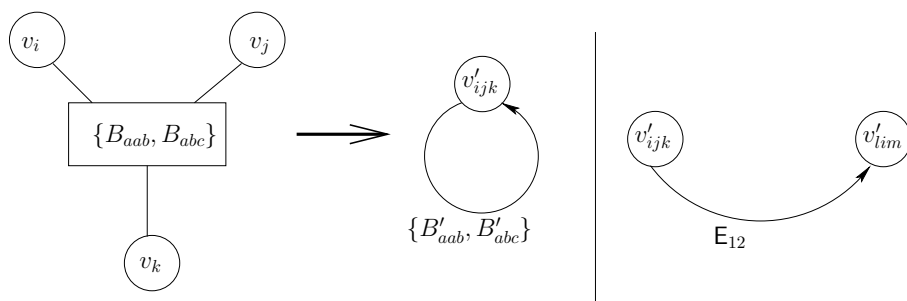
- $D'$  is the set  $D^n = \mathcal{U}$ . Hence, each relation of  $\mathcal{B}'$  is a subset of  $\mathcal{U}' = D' \times D' = D^n \times D^n = \mathcal{U} \times \mathcal{U}$ .
- For each relation  $B_i \in \mathcal{B}$ , with  $1 \leq i \leq k$ , a basic relation  $B'_i$  is introduced in  $\mathcal{B}'$ .  $B'_i$  is defined by the relation  $\{((x_1, \dots, x_n), (x_1, \dots, x_n)) : (x_1, \dots, x_n) \in B_i\}$ . Note that the set of relations  $\mathcal{B}'_P = \{B'_1, \dots, B'_k\}$  forms a partition of the relation of identity of  $D'$  which we will denote by  $\Delta'_{12}$ .
- For all  $i, j \in \{1, \dots, n\}$  we define the relation  $E_{ij}$  by:  
 $E_{ij} = \{((x_1, \dots, x_n), (x'_1, \dots, x'_n)) \in \mathcal{U}' : x_i = x'_j\} \setminus \Delta'_{12}$ .  $E^0 = \{E_{ij} : i, j \in \{1, \dots, n\}\}$ .  
 $E^m$  with  $m > 0$  is inductively defined by  $E^m = \{R_1 \cap R_2, R_1 \setminus (R_1 \cap R_2), R_2 \setminus (R_1 \cap R_2) : R_1, R_2 \in E^{m-1}\}$ . Let  $m'$  the smallest integer such that  $E^{m'} = E^{m'+1}$ .  $\mathcal{B}'_E = \{R \in E^{m'} \text{ such that } R \neq \emptyset \text{ and } \nexists R' \neq \emptyset \in E^{m'} \text{ with } R' \subset R\}$ . The set of relations of  $\mathcal{B}'_E$  are added to the set  $\mathcal{B}'$ .
- Let  $F$  be the binary relation on  $D'$  defined by  $F = \mathcal{U}' \setminus (E_{ij} \cup \mathcal{B}'_P)$ . We add  $F$  to  $\mathcal{B}'$ .

Hence the final set of basic relations is the set  $\mathcal{B}' = \mathcal{B}'_P \cup \mathcal{B}'_E \cup \{F\}$ . The reader can check that  $\mathcal{B}'$  satisfies properties 1, 2 and 3 and hence defines a qualitative calculus of arity 2.

Now, consider a QCN  $\mathcal{N} = (V, C)$  defined on  $\mathcal{B}$ . Let us define an equivalent QCN  $\mathcal{N}' = (V', C')$  on  $\mathcal{B}'$ :

- To define  $V'$ , for each  $n$ -tuples of  $n$  variables  $(v_1, \dots, v_n)$  of  $V$  we introduce a variable  $v'_{\{v_1, \dots, v_n\}}$  in  $\mathcal{B}'$ .
- Given a variable  $v' = v'_{\{v_1, \dots, v_n\}}$  belonging to  $V'$  we define  $C'(v', v')$  by the relation  $\{B'_i : B_i \in C(v_1, \dots, v_n)\}$ .
- Given two distinct variables  $v'_i = v'_{\{v_1^i, \dots, v_n^i\}}$  and  $v'_j = v'_{\{v_1^j, \dots, v_n^j\}}$  belonging to  $V'$ ,  $C'(v'_i, v'_j)$  is the relation  $E$  defined in the following way: let  $\gamma$  the set of pairs of integer defined by  $\{(k, l) \in \mathbb{N} \times \mathbb{N} : v_k^i = v_l^j\}$ .  $E$  is the set of basic relations of  $\mathcal{B}'$  (more precisely of  $\mathcal{B}'_E$ ) defined as the relation  $\bigcap_{(k,l) \in \gamma} E_{kl}$ .

The reader can check that  $\mathcal{N}$  is a consistent QCN iff  $\mathcal{N}'$  is a consistent QCN. This construction is inspired by the technique called dual encoding [10] used in the domain of discrete CSPs to convert  $n$ -ary constraints into binary constraints.



**Fig. 2.** Converting a ternary constraint  $C_{ijk}$  of the cyclic point algebra into a binary constraint (left). Expressing a structural constraint between  $v'_{ijk}$  and  $v'_{ilm}$  for distinct integers  $i, j, k, l, m$  (right).

#### 4 The Qualitative Algebra Toolkit (QAT)

Clearly, all existing qualitative calculi share the same structure, but, to our knowledge, implementations and software tools have only been developed for individual calculi. The QAT (Qualitative Algebra Toolkit) has been conceived as a remedy to this situation. Specifically, the QAT is a JAVA constraint programming library developed at *CRIL-CNRS* at the University of Artois. It aims to provide open and generic tools for defining and manipulating qualitative algebras and qualitative networks based on these algebras. The core of QAT contains three main packages. In the sequel of this section we are going to present each of those packages.

The Algebra package is devoted to the algebraic aspects of the qualitative calculi. While programs proposed in the literature for using qualitative formalisms are *ad hoc* implementations for a specific algebra and for specific solving methods, the QAT allows the user to define arbitrary qualitative algebras (including non-binary algebras) using a simple XML file. This XML file, which respects a specific DTD, contains the definitions of the different elements forming the algebraic structure of the qualitative calculus: the set of basic relations, the diagonal elements, the table of rotation, the table of permutation and the table of qualitative composition. We defined this XML file for many qualitative calculi of the literature: the interval algebra [1], the point



algebra [14], the cyclic point algebra [15], the cyclic interval algebra [16], the rectangle algebra [17], the INDU algebra [4], the multidimensional algebra [18], the RCC-5 algebra [2], the RCC-8 algebra [2], the cardinal direction algebra [3]). Tools allowing to define a qualitative algebra as the Cartesian Product of other qualitative algebras are also available.

The QCN package contains tools for defining and manipulating qualitative constraint networks on any qualitative algebra. As for the algebraic structure, a specific DTD allows the use of XML files for specifying QCNs. The XML file lists the variables and relations defining the qualitative constraints. Functionalities are provided for accessing and modifying the variables of a QCN, its constraints and the basic relations they contain. Part of the QCN package is devoted to the generation of random instances of QCNs. A large amount of the research about qualitative calculi consists in the elaboration of new algorithms to solve QCNs. The efficiency of these algorithms must be validated by experimentations on instances of QCNs. Unfortunately, in the general case there does not exist instances provided by real world problems. Hence, the generation of random instances is a necessary task [19]. The QCN package of the QAT provides generic models allowing to generate random instances of QCNs for any qualitative calculus.

The Solver package contains numerous methods to solve the main problems of interest when dealing with qualitative constraint networks, namely the consistency problem, the problem of finding one or all solutions, and the minimal network problem. All these methods are generic and can be applied to QCNs based on arbitrary qualitative calculi. They make use of the algebraic aspect of the calculus without considering the semantics of the basic relations. In other words, they make abstraction of the definitions of the basic relations and only uniquely manipulate *the symbols* corresponding to these relations. Nevertheless, by using the object-oriented concept, it is very easy to particularize a solving method to a specific qualitative algebra or a particular kind of relations. We implemented most of the usual solving methods, such as the standard generate and test methods, search methods based on backtrack and forward checking, and constraint local propagation methods. The user can configure these different methods by choosing among a range of heuristics. These heuristics are related to the choice of the variables or the constraints to be scanned, of the basic relations in a constraint during a search. The order in which the constraints are selected and the order in which the basic relations of the selected constraint are examined can greatly affect the performance of a backtracking algorithm [19]. The idea behind constraint ordering heuristics is to first instantiate the more restrictive constraints first. The idea behind value ordering basic relations is to order the basic relations of the constraints so that the value that most likely leads to a solution is the first one to be selected. The QAT allows the user to implement new heuristics based on existing heuristics. As for local constraint propagation methods, whereas in discrete CSPs arc consistency is widely used [20], *path consistency* is the most efficient and most frequently used kind of local consistency in the domain of the qualitative constraints. More exactly, the methods used are based on local constraint propagation based on qualitative composition, in the manner of the  $PC1_n$  algorithm described in the previous section. In addition to  $PC1_n$ , we have extended and implemented algorithms based on PC2 [21].

## 5 Conclusions

We propose and study a general formal definition of qualitative calculi based on basic relations of an arbitrary arity. This unifying definition allows us to capture the algebraic structure of all

qualitative calculi in the literature. The main elements of the algebraic structure are diagonal elements, and the operations of permutation, rotation and qualitative composition. We give a transformation allowing to build a qualitative calculus based on binary basic relations from a qualitative calculus based on arbitrary basic relations. The expressive powers of both calculi are similar. Moreover we generalize the constraint propagation method *PC1* to the general case, *i.e.* for relations of any arity. In a second part we describe the QAT<sup>1</sup> (Qualitative Algebra Toolkit), a JAVA constraint programming library allowing to handle constraint networks defined on arbitrary *n*-ary qualitative calculi. This toolkit provides algorithms for solving the consistency problem and related problems, as well as most of the heuristics used in the domain. QAT is implemented using the object oriented technology. Hence, it is an open platform, and its functionalities are easily extendable. New heuristics (resp. methods) can be defined and tested. Among the tools it provides are classes allowing to generate and to use benchmarks of qualitative networks. Hence new heuristics or new solving algorithms can be conveniently evaluated.

## References

1. Allen, J.: An interval-based representation of temporal knowledge. In: Proc. of the Seventh International Joint Conference on Artificial Intelligence (IJCAI'81) (1981) 221–226.
2. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'92). (1992) 165–176.
3. Ligozat, G.: Reasoning about cardinal directions. *Journal of Visual Languages and Computing* **1** (1998) 23–44.
4. Pujari, A.K., Kumari, G.V., Sattar, A.: INDU: An interval and duration network. In: Australian Joint Conference on Artificial Intelligence (1999) 291–303.
5. Isli, A., Cohn, A.G.: A new approach to cyclic ordering of 2D orientations using ternary relation algebras. *Artificial Intelligence* **122** (2000) 137–187.
6. Montanari, U.: Networks of constraints: Fundamental properties and application to picture processing. *Information Sciences* **7** (1974) 95–132.
7. Mackworth, A.K., Freuder, E.C.: The Complexity of Some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problem. *Artificial Intelligence* **25** (1985) 65–74.
8. Hirsch, R. and Hodkinson, I.: *Relation Algebras by Games*. Volume 147. North-Holland, Studies in Logic and the Foundations of Mathematics (2002).
9. Balbiani, P., Condotta, J.F., Ligozat, G.: On the consistency problem for the INDU calculus. In: Proceedings of TIME'2003 (2003) 203–211.
10. Bacchus, F., van Beek P.: On the conversion between non-binary and binary constraint satisfaction problems. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98) (1998) 311–318.
11. Ligozat, G., Renz, J.: What is a qualitative calculus? a general framework. In: Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI) (2004).
12. Balbiani, P., Condotta, J.F., Ligozat, G.: Reasoning about cyclic space: Axiomatic and computational aspects. In: Proceedings of Spatial Cognition 2003, LNCS 2685 (2003) 348–371.
13. Renz, J., Ligozat, G.: Weak composition for qualitative spatial and temporal reasoning. In: Proceedings of the Eleventh International Conference on Principles and Practice of Constraint Programming (CP'05) (2005).
14. Vilain, M., Kautz, H.: Constraint Propagation Algorithms for Temporal Reasoning. In: Proc. of the Fifth Nat. Conf. on Art. Int. (AAAI'86) (1986) 377–382.
15. Balbiani, P., Ligozat, G., Condotta, J.F.: Reasoning about cyclic space: axiomatic and computational aspects. In: Proceedings of Spatial Cognition 2003 (2003).
16. Balbiani, P., , Osmani, A.: A model for reasoning about topologic relations between cyclic intervals. In: Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'00) (2000).
17. Balbiani, P., Condotta, J.F., Fariñas del Cerro, L.: A new tractable subclass of the rectangle algebra. In Dean, T., ed.: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI'99). (1999) 442–447.

<sup>1</sup> The documentation and the source of the QAT library can be found at <http://www.cril.univ-artois.fr/~saade/QAT>.

18. Balbiani, P., Condotta, J.F., Fariñas del Cerro, L.: Spatial reasoning about points in a multidimensional setting. In: Proceedings of the workshop on temporal and spatial reasoning (IJCAI'99). (1999) 105–113.
19. van Beek, P., Manchak, D.W.: The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research* **4** (1996) 1–18.
20. Apt, K.R.: *Principles of Constraint Programming*. Cambridge University Press (2003).
21. Bessière, C.: A Simple Way to Improve Path Consistency Processing in Interval Algebra Networks. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI'96), Volume 1, (1996) 375–380.