Quantitative Reasoning with Cyclic Intervals: A Proposal

Debasis Mitra

Florida Institute of Technology
Melbourne, Florida, USA
dmitra@cs.fit.edu

Abstract

Reasoning with time is ubiquitous. Many computational problems involving planning, scheduling, reasoning and modeling often refer to the literature and use some simplified versions of the rich body of theoretical works within the spatio-temporal reasoning areas. Although most of such latter works are for qualitative reasoning, one notable exception is Dechter et al's investigation on the *temporal constraint network* where explicit quantitative temporal information is modeled. For obvious reason a quantitative framework finds more application, as computers are still primarily numerical data crunching machines rather than symbol crunching ones. In this article we propose a framework for extending Dechter et al's investigation toward a non-linear underlying space where a finite Real line closes onto itself (cyclic). Such a space is not only esoteric but also practical, as reasoning with many calendar domains uses such a cyclic space, e.g., weekly scheduling of tasks. This is an ongoing work and only the basics are provided here.

1. Introduction

Reasoning with spatio-temporal objects is recognized as an important computational problem. Over the last few decades many such calculi have been identified and studied at length. Most of these calculi are based on qualitative relations between the objects where the exact quantitative information is ignored. Very little attention has been paid within the community over defining calculi with quantitative information. The only one known to this date is the famous *temporal constraint satisfaction problem* or the TCSP of Dechter et al. [1991].

In this article we propose a quantitative framework to reason over a closed space of Real numbers. This is a confluence of three related areas: Balbiani et al's [2000] qualitative interval calculi over cyclic time, modular arithmetic, and Dechter et al's [1991] TCSP. Like the former work on reasoning with cyclic-time intervals we consider the underlying domain as a modulo space. We propose to extend the TCSP works toward such a space whereas the original TCSP was investigated over the infinite Real line. Practical problem domains of such cyclic interval calculus could be planning under weekly, monthly or any such periodic calendar-framework.

2. Background on Referenced Cyclic Interval

Primary object in the interval calculus is a convex interval [a, b], where $-\infty < a < b < \infty$. Whether interval should open or closed is a debate that was never really settled. We extend this concept of interval by defining the underlying domain to be a closed real line $[0, \theta)$ as in modular arithmetic, instead of $(-\infty,\infty)$ as in the interval calculus. The special point 0 or equivalently θ is a point of reference on the cyclic line. In developing cyclic interval algebra Balbiani et al. [2000] and Osmani [2001] did not need such a point of reference as their calculus is qualitative. However, any quantitative expression cannot be expressed without such a point of reference. We call this new calculus as *referenced cyclic interval calculus* or RCI calculus, as opposed to Allen's interval calculus as the *Real interval calculus* or RI calculus.

Definition 1. The primary object in RCI calculus is the *Referenced cyclic interval* [x, y] such that $0 \le \{x, y\} < \theta$.

Definition 2. A *non-crossover RCI* is where $x < y$.

Example 1: Suppose, $\theta = 360$. We will use this in all subsequent examples in this article. Two intervals X = [20, 30] and Y = [40, 60] will have relation (*X before Y*) and (*Y after X*) both in the RCI as well as in the RI calculi.

Note that in Osmani et al's cyclic interval (CI) calculus does not distinguish between the relations *before* and *after* as there is no reference point in the underlying domain. So, in that calculus (*X before Y*) and (*X after Y*) may be true at the same time.

Definition 3. A *crossover RCI* [x, y] is where $x > y$, indicating that the point θ is included within the convex interval.

Example 2: The crossover RCI [350, 20] is really the interval $[350, 360) \cup [0, 20]$.

Definition 4. An *Interval duration* for an RCI X= [x, y] is $|X| = (y - x + \theta) \bmod \theta$.

Example 3. $|[350, 20]| = (20 – 350 + 360) \bmod 360 = 30$, but $|[20, 350]| = (350 – 20 + 360) \bmod 360 = 330$.

The RI calculus has 13 jointly exhaustive and pairwise disjoint (JEPD) primitive relations between the RIs. CI calculus similarly has 16 JEPDs.

Proposition 1. RCI calculus has 52 primitive JEPD primitive relations.

Proposition 1 can be easily verified by writing a small program. These 52 relations subsumes the RI calculus' thirteen primitive relations as well as CI calculus' sixteen primitive relations. However, as our purpose is to develop a framework for quantitative reasoning scheme rather than a qualitative one, we also allow points in the calculus, or x=y in a RCI [x, y]. This scheme may be called as *referenced cyclic point-interval* (RCPI) calculus.

Proposition 2. RCPI calculus has 75 JEPD primitive relations (shown in the Appendix 1).

As in the case of RCI calculus proposition 2 may be also be easily verified by exhaustive computation. Obviously, RCPI primitive relations subsumes those for RCI.

We define two special RCPIs for the sake of completion of the framework.

Definition 5. *Universal Interval.* Π is the unique universal interval over the whole underlying domain [0, θ), with the interval duration θ.

Definition 6. *Empty Interval.* Φ is the unique empty interval without any point in it and with the interval duration 0.

3. TCSP with RCPI

In this section we extend Dechter et al's [1991] *temporal constraint satisfaction problem* with *simple intervals* over the underlying cyclic domain. In TCSP a binary constraint between two temporal events (points) is described as a set of convex intervals.

Example 4. Tom takes twenty to thirty minutes to come to office by car, but fifty to seventy minutes by public bus. Formally this constraint is expressed in TCSP as (D {[20, 30], [50, 70]} A), where at time instant D Tom departs from home and at A he arrives at his office.

A TCSP consists of a set of such binary constraints over multiple temporal events.

Reasoning with only one convex interval over each constraint, called *simple TCSP*, is tractable and has been studied extensively. We propose to develop a reasoning scheme for simple TCSP where the constraint is a RCPI instead of RI. In this article we will redefine the fundamental operators behind such a reasoning scheme. Some of this reasoning operators over RI is too intuitive and never discussed in the literature whereas others are addressed by Dechter et al. [1991] in their seminal work on TCSP.

Proposition 3. The *converse* of a RCPI X = [x, y] is $X^{\cup}$ = [-y+ θ, -x+ θ]. Note that $(A\ X\ B) \Leftrightarrow (B\ X^{\cup} A)$.

Proposition 4. The *inverse* of a RCPI X = [x, y] is ~X = [y, x].
The *empty interval* can be written as the inverse of the *universal interval* [0, θ). Hence, Φ is (θ, 0].

Performing set union and intersection over convex intervals in modular arithmetic, or with the RCPIs are not trivial. The main problem stems from representing crossover RCPI and non-crossover RCPI uniformly for the purpose of reasoning. For this purpose we develop some ordering scheme for the end points of intervals in the cyclic domain.

First, we will address the problem of finding the position of a point *p* with respect to an interval X=[x1, x2]. An algorithm *Order(p, X)* will output the following.
For x1 < x2 (i.e., non-crossover interval),
if p<x1, then the ordering is (p, x1, x2);
if p>x2, then the ordering is (x1, x2, p);
else the ordering is (x1, p, x2).
The first two cases above are when *p* is outside the interval.
For x1 > x2 (i.e., crossover interval),
if p>x1, then the ordering is (x1, p, x2);
if p<x2, then the ordering is (x1, p, x2);
else the ordering is either (x1, x2, p) or (p, x1, x2), i.e., *p* is outside the interval and the two orderings are indistinguishable.

Now we are ready to position the end points of a pair of intervals on the cyclic domain relative to each other. Suppose, the two intervals are X=[x1, x2] and Y=[y1, y2]. None, either, or both of them could be crossover interval(s). We need to create a total ordering of the four end points in order to define the *union* and *intersection* operators between the pair of intervals. An algorithm *Order(X, Y)* will use the previous algorithm four times:
Order(x1, Y), Order(x2, Y), Order(y1, X), Order(y2, X).
The four relative three-point orderings can be subsequently merged to the final four-point orderings as a constraint satisfaction problem. The *Union* and the *Intersection* operations can be done from the resulting four-point orderings as follows.

Definition 7. *Union.*
(1) For the ordering (x1, x2, y1, y2) or (y1, y2, x1, x2), $X \cup Y$ = {X, Y}, because the intervals are disjoint.

(2) For the ordering (x1, y1, y2, x2),
$X \cup Y = X$, and for the ordering (y1, x1, x2, y2),
$X \cup Y = Y$, because one interval subsumes the other.
(3) For the ordering (x1, y1, x2, y2),
$X \cup Y =$ [x1, y2]. Symmetrically, for the ordering (y1, x1, y2, x2),
$X \cup Y =$ [y1, x2].
(4) A special case when the four-point ordering cannot be created is when the three-point orderings are {(x1, y2, x2), (x1, y1, x2), (y1, x2, y2), (y1, x1, y2)}.
Here, $X \cup Y = \Pi$, the universal interval. □

Example 5 (case 4 of definition 7). [320, 20] $\cup$ [10, 330] = Π.

Definition 8. *Intersection*.
(1) For the ordering (x1, x2, y1, y2) or (y1, y2, x1, x2),
$X \cap Y = \Phi$, because the intervals are disjoint.
(2) For the ordering (x1, y1, y2, x2),
$X \cap Y = Y$, and for the ordering (y1, x1, x2, y2),
$X \cap Y = X$, because one interval subsumes the other.
(3) For the ordering (x1, y1, x2, y2),
$X \cap Y =$ [y1, x2]. Symmetrically, for the ordering (y1, x1, y2, x2),
$X \cap Y =$ [x1, y2].
(4) A special case when the four-point ordering cannot be created is when the three-point orderings are {(x1, y2, x2), (x1, y1, x2), (y1, x2, y2), (y1, x1, y2)}.
Here, $X \cap Y =$ {[x1, y2], [y1, x2]}, two disjoint intervals. □

Example 5 (case 4 of definition 8). [320, 20] $\cup$ [10, 330] = {[10, 20], [320, 330]}.

Although the definitions 7 and 8 are really for RCI rather than for RCPI the latter case can be handled by easily extending the definitions.

Another important operation for reasoning is the *composition*. Suppose, *A*, *B*, and *C* are three events or points on the cyclic domain. For the two quantitative constraints (*A X B*) and (*B Y C*), the resulting constraint is (*A Z C*), where *Z* is the composition of two constraints *X* and *Y*, or , $Z = X \otimes Y$.

Definition 9. *Composition*.
For two RCPIs *X* =[x1, x2] and *Y* = [y1, y2], their composition is,

$$X \otimes Y = [(x_1 + y_1)\bmod\theta, (x_2 + y_2)\bmod\theta]$$

Definitions 7, 8, and 9 should cover all the fifty-two JEPD primitive relations for RCI calculus, and appropriately extended definitions should cover the seventy-five such primitive relations over RCPI. These definitions lay the foundation for doing quantitative reasoning over cyclic domains (with RCPI).

4. Related works and discussion

In this work we have developed an outline for reasoning with quantitative constraints over a cyclic domain that we call cyclic-TCSP in the line of Dechter et al's [1991] TCSP works over linear time or the domain of Real numbers. TCSP is tractable with *simple* constraints (convex intervals). The framework presented here is also over similar convex constraints as opposed to the case of multi-interval constraints (e.g., example 4) between temporal events *A* and *B*. A valid question: is a *simple* cyclic-TCSP as presented here also tractable? We do not have any answer for that yet. However, observing the case of qualitative reasoning with cyclic-intervals our guess that this is not so in cyclic-TCSP. Reasoning with CI calculus is intractable even with only primitive relations as constraints (without any disjunction) [Balbiani et al., 2000]. For most other spatio-temporal algebra reasoning with only the primitive relations is trivially tractable.

A related spatial reasoning scheme over 2D Euclidean space is the *Star-algebra* proposed by Mitra [2004] and further investigated by Renz and Mitra [2004]. This is a generalized version of the *Cardinal-directions* algebra studied by Ligozat [1998]. A similar modular nature over the underlying space shows up both in that case and in reasoning with CI or RCI. The non-linearity of the underlying space poses challenges to the corresponding reasoning problems.

Appendix 1: Qualitative JEPD primitive relations over RCPI

We have used four end-points (1, 2, 3, 4/0) over a cyclic domain with $\theta = 4$ for the two intervals. The six end point relations between these points are show in the second column. A name for each primitive relation and its abbreviation is provided in the third column while the relations are indexed in the first column.

#	**Relation**	**Name**
1	[1,1][1,1] =,=,=,=,=,=	non non equals nn-=
2	[1,1][1,2] =,=,<,=,<,<	non non meets nn-m
3	[2,2][2,1] =,=,>,=,>,>	non cross meets nc-m
4	[1,1][2,1] =,<,=,<,=,>	non cross meets inverse nc-mi
5	[1,1][2,2] =,<,<,<,<,=	non non before nn-b
6	[1,1][2,3]	non non before

	=,<,<,<,<,<	nn-b
7	[1,1][3,2] =,<,<,<,<,>	non cross during nc-d
8	[2,2][3,1] =,<,>,<,>,>	non cross before nc-b
9	[2,2][1,2] =,>,=,>,=,<	non non meets inverse nn-mi
10	[2,2][1,3] =,>,<,>,<,<	non non during nn-d
11	[2,2][1,1] =,>,>,>,>,=	non non before inverse nn-bi
12	[3,3][1,2] =,>,>,>,>,<	non non before inverse nn-bi
13	[3,3][2,1] =,>,>,>,>,>	non cross during nc-d
14	[1,2][1,1] <,=,=,>,>,=	non non meets inverse nn-mi
15	[1,2][1,2] <,=,<,>,=,<	non non equals nn-=
16	[1,2][1,3] <,=,<,>,<,<	non non starts nn-s
17	[1,3][1,2] <,=,<,>,>,<	non non starts inverse nn-si
18	[2,3][2,1] <,=,>,>,>,>	non cross starts nc-s
19	[1,2][2,1] <,<,=,=,>,>	non cross meets meets nc-mm
20	[1,2][3,1] <,<,=,<,>,>	non cross meets inverse nc-mi
21	[1,3][2,1] <,<,=,>,>,>	non cross overlaps meets nc-om
22	[1,2][2,2] <,<,<,=,=,=	non non meets nn-m
23	[1,2][2,3] <,<,<,=,<,<	non non meets nn-m
24	[1,3][3,2] <,<,<,=,>,>	non cross meets overlaps nc-mo
25	[1,2][3,2] <,<,<,<,=,>	non cross finishes nc-f
26	[1,2][3,3] <,<,<,<,<,=	non non before nn-b
27	[1,2][3,4] <,<,<,<,<,<	non non before nn-b
28	[1,2][4,3] <,<,<,<,<,>	non cross during nc-d
29	[1,3][4,2] <,<,<,<,>,>	non cross overlaps nc-o
30	[1,3][2,3] <,<,<,>,=,<	non non finishes inverse nn-fi
31	[1,3][2,4] <,<,<,>,<,<	non non overlaps nn-o
32	[1,3][2,2] <,<,<,>,>,=	non non during inverse nn-di
33	[1,4][2,3] <,<,<,>,>,<	non non during inverse nn-di
34	[1,4][3,2] <,<,<,>,>,>	non cross overlaps overlaps nc-oo
35	[2,3][3,1] <,<,>,=,>,>	non cross meets nc-m
36	[2,3][4,1] <,<,>,<,>,>	non cross before nc-b
37	[2,4][3,1] <,<,>,>,>,>	non cross overlaps nc-o
38	[2,3][1,2] <,>,=,>,>,<	non non meets inverse nn-mi
39	[2,3][1,3] <,>,<,>,=,<	non non finishes nn-f
40	[2,3][1,4] <,>,<,>,<,<	non non during nn-d
41	[2,4][1,3] <,>,<,>,>,<	non non overlaps inverse nn-oi
42	[2,3][1,1] <,>,>,>,>,=	non non before inverse nn-bi
43	[3,4][1,2] <,>,>,>,>,<	non non before inverse nn-bi
44	[3,4][2,1] <,>,>,>,>,>	non cross during nc-d
45	[2,1][2,2] >,=,=,<,<,=	cross non meets inverse cn-mi
46	[2,1][2,3] >,=,<,<,<,<	cross non starts inverse cn-si
47	[2,1][2,1] >,=,>,<,=,>	cross cross equals cc-=
48	[3,1][3,2] >,=,>,<,<,>	cross cross starts cc-s
49	[3,2][3,1] >,=,>,<,>,>	cross cross starts inverse cc-si
50	[2,1][3,2] >,<,=,<,<,>	cross cross overlaps meets cc-om
51	[2,1][3,3] >,<,<,<,<,=	cross non during inverse cn-di
52	[2,1][3,4] >,<,<,<,<,<	cross non during inverse cn-di
53	[2,1][4,3] >,<,<,<,<,>	cross cross overlaps overlaps cc-oo
54	[2,1][3,1] >,<,>,<,=,>	cross cross finishes inverse cc-fi
55	[3,1][4,2] >,<,>,<,<,>	cross cross overlaps cc-o
56	[3,2][4,1] >,<,>,<,>,>	cross cross during inverse cc-di
57	[2,1][1,2] >,>,=,=,<,<	cross non meets meets cn-mm
58	[3,1][2,3] >,>,=,<,<,<	cross non meets inverse cn-mi
59	[3,2][1,3] >,>,=,>,<,<	cross non meets overlaps cn-mo
60	[2,1][1,3] >,>,<,=,<,<	cross non overlaps meets cn-om
61	[3,1][2,4]	cross non overlaps inverse

	>,>,<,<,<,<	cn-oi
62	[3,2][1,4] >,>,<,>,<,<	cross non overlaps overlaps cn-oo
63	[2,1][1,1] >,>,>,=,=,=	cross non meets inverse cn-mi
64	[3,1][1,2] >,>,>,=,<,<	cross non meets cn-m
65	[3,2][2,1] >,>,>,=,>,>	cross cross overlaps meets cc-om
66	[3,1][2,1] >,>,>,<,=,>	cross cross finishes cc-f
67	[3,1][2,2] >,>,>,<,<,=	cross non before cn-b
68	[4,1][2,3] >,>,>,<,<,<	cross non before cn-b
69	[4,1][3,2] >,>,>,<,<,>	cross cross during cc-d
70	[4,2][3,1] >,>,>,<,>,>	cross cross overlaps cc-o
71	[3,2][1,2] >,>,>,>,=,<	cross non finishes inverse cn-fi
72	[4,2][1,3] >,>,>,>,<,<	cross non overlaps cn-o
73	[3,2][1,1] >,>,>,>,>,=	cross non during inverse cn-di
74	[4,3][1,2] >,>,>,>,>,<	cross non during inverse cn-di
75	[4,3][2,1] >,>,>,>,>,>	cross cross overlaps overlaps cc-oo

Acknowledgement: Gary brown developed the seventy-five primitive relations presented in the appendix 1. We acknowledge some support from NSF during the research.

References

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, vol. 26, no. 11, pp.832-843.

Dechter, R., Meiri, I., and Pearl, J. (1991) Temporal Constraint Networks, *Artificial Intelligence* 49, pp. 61-95.

Balbiani, P., and Osmani, A. (2000) A Model for Reasoning about Topologic Relations between cyclic intervals. *Prniciples of Knowledge representation and reasoning (KR-2000), Proceedings of*, pp. 378-385.

G. Ligozat. (1998) Reasoning about cardinal directions, *Jnl. of Visual Languages and Computing 9*, pp. 23-44.

Mitra, D. (2004) Modeling and Reasoning with Star Calculus. *AI and Math* Conference, Fort Lauderdale, Florida.

Osmani, A. (2001) Algorithm and Complexity for Reasoning about Cyclic Interval. (*unpublished manuscript*).

Renz, J. and Mitra, D. (2004) Qualitative Direction Calculi with Arbitrary Granularity. *Pacific Rim Conference on AI (PRICAI-04), Proceedings of.*