Qualitative Constraint-based Reasoning

A Gentle Introduction

Michael Sioutis

Junior Professor
LIRMM UMR 5506, Université Montpellier / CNRS, France

1er semestre 2023

Outline

Bio

Personal Details

- Greek, born in Wickede, Germany
- Languages: Greek $\succ$ English $\succ$ German $\succ$ Spanish $\succ$ French
- Chaire de Professeur Junior at Université Montpellier (France)

International Outlook

- Postdocs in France (IUT de Lens), Sweden (Örebro University), Finland (Aalto University), Germany (Bamberg University)

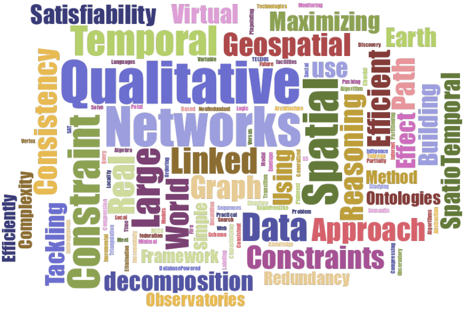
- Ph.D. in Computer Science in 2017 from Université d'Artois (France); brief stay in University of Technology Sydney (Australia)

centre de recherche en informatique de lens

UNIVERSITÉ D'ARTOIS

Research Interests

- Artificial Intelligence
- Constraint Programming
- Semantic Web



Communication

- Mailing list: `m2-constraints@lirmm.fr`
- Personal E-mail: `michael.sioutis@lirmm.fr`
- Office hours: Wednesdays 9h30–11h30

Qualitative Reasoning: An Overview

Limitations of ML Systems

Statistical information can hardly form the basis of robust reasoning

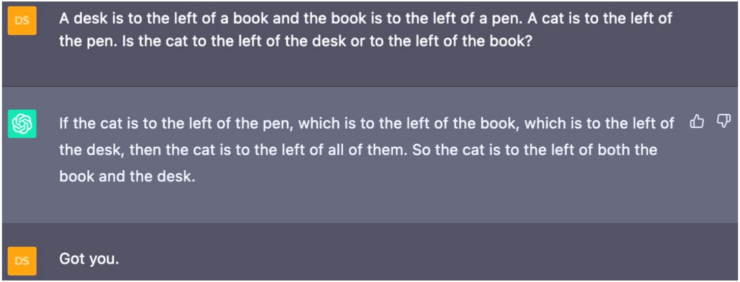


Figure: Assessing the spatio-temporal reasoning capabilities of ChatGPT; this example is thanks to Dr Jae Hee Lee of Hamburg University (Germany)

Many more examples of the above form exist[1]

[1] A. G. Cohn, J. Hernandez-Orallo: *Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of LLMs* . `https://arxiv.org/abs/2304.11164` (2023)

Neuro-Symbolic AI to Tame ML

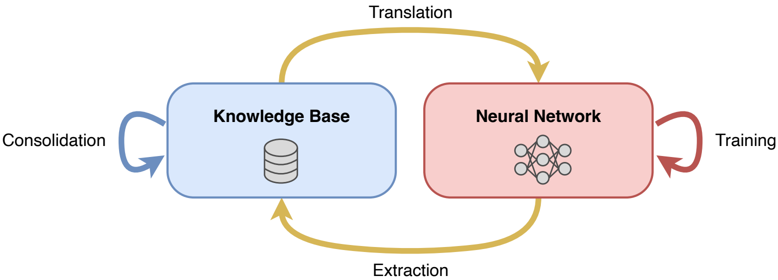


Figure: Cyclical interaction in Neuro-Symbolic AI; a symbolic system feeds symbolic (partial) knowledge to a neural network system, which can be trained on raw data, and knowledge acquired through machine learning can then be extracted back to the symbolic system, and made available for further processing in symbolic form[2]

[2]S. Bader, P. Hitzler: *Dimensions of Neural-symbolic Integration - A Structured Survey*. We Will Show Them! Essays in Honour of Dov Gabbay 1 (2005)

Any Ruleset is Good?

- “if I accelerate faster than the vehicle directly in front of me, then I will overtake it”

- “if I accelerate faster than the vehicle directly in front of me, then I will bump into it”

Flavors of Logic

ML models should be tied to assumptions that align with physics and human cognition to allow for generalization

B. Schölkopf et al.[3]

[3] B. Schölkopf et al.: *Toward Causal Representation Learning*. Proc. IEEE 109 (2021)

Injecting Causality via Qualitative Calculi

- Qualitative Spatial & Temporal Reasoning (QSTR) is a major field of study in KR, and Symbolic AI in general[4]
- QSTR abstracts from numerical quantities of space and time by using natural descriptions instead (e.g., *precedes*, *contains*, *is left of*), grounded on *physics* and *human cognition*

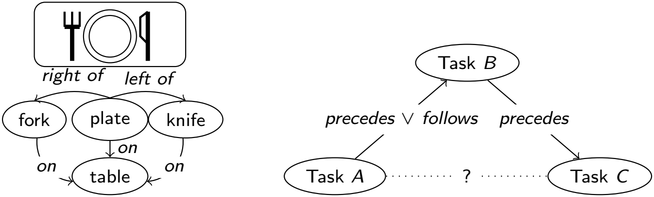


Figure: Abstraction of a spatial configuration (left), temporal constraint network of three variables (right); ? denotes complete uncertainty

[4] G. Ligozat.: *Qualitative Spatial and Temporal Reasoning*. ISTE Series. Wiley, 2011

Example Calculus: RCC8

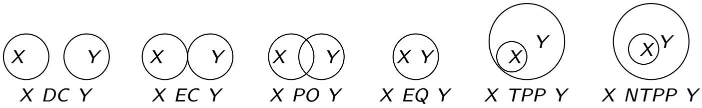


Figure: The base relations of RCC8; inverses are omitted in the figure

Example Calculus: Allen's Interval Algebra

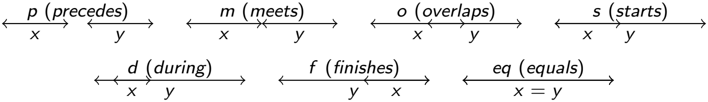


Figure: The base relations of Interval Algebra; inverses are omitted in the figure

Aspects of Space and Time ... and More

- Abundance of calculi dealing with trajectories, occlusion, intervals, and so on[5]
- Translating terminological knowledge into region spaces, e.g., *document PO paper*[6]
- Qualitative models also involved in biology, economics, robotics, and and more

[5] F. Dylla et al.: *A Survey of Qualitative Spatial and Temporal Calculi: Algebraic and Computational Properties*. ACM Comput. Surv. 50 (2017)

[6] Z. Bouraoui et al.: *Region-Based Merging of Open-Domain Terminological Knowledge*. In: KR 2022

Applications: Geospatial Semantic Segmentation

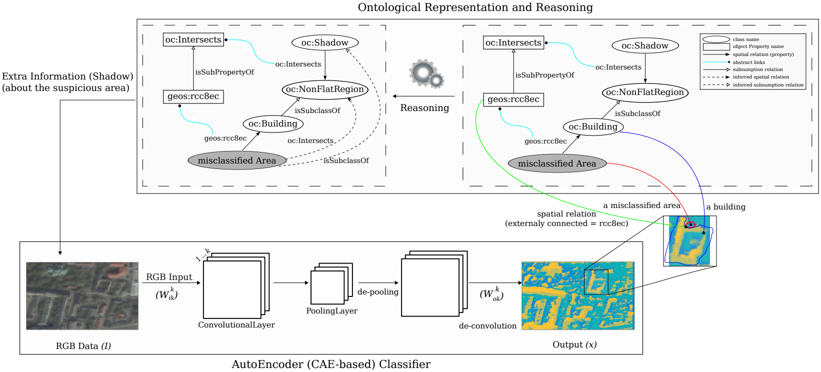


Figure: A semantic referee reasons about the mistakes made by the classifier based on ontological concepts and provides additional information back to the classifier that prevents the classifier from making the same misclassifications[7]

[7] M. Alirezaie et al.: *Semantic referee: A neural-symbolic framework for enhancing geospatial semantic segmentation*. Semantic Web 10 (2019)

Applications: Medicine / Image Processing

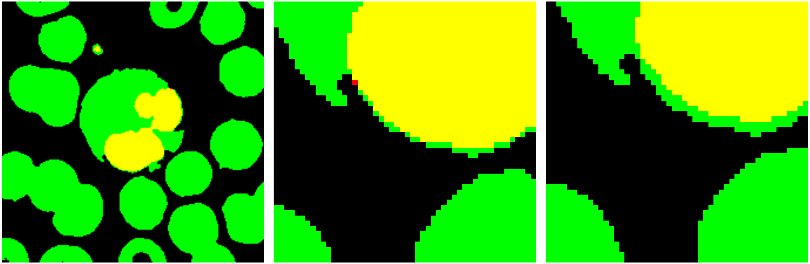


Figure: Left: segmented cell bodies (green), lobulated cell nuclei (yellow and red) and background (black), Middle: segmented cell nucleus extending outside border of host cell (red pixels), Right: the result of applying a morphological erosion operator; in this case the original *partially overlaps* relation changes to *proper part*[8]

[8] M. Sioutis et al.: *Ordering Spatio-Temporal Sequences to Meet Transition Constraints: Complexity and Framework*. In: AIAI 2015

Applications: Region Approximation

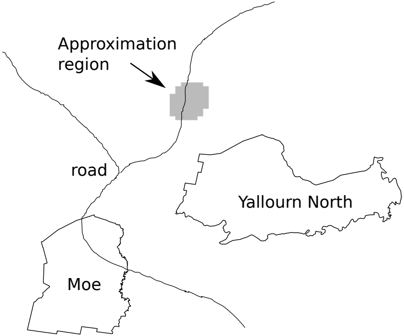


Figure: Illustration of locating a region by natural language descriptions, e.g., "Bushfire burning about 5km northwest of Yallourn North" and "I saw fire about 10km northeast from Moe", with the help of a region approximation method[9]

[9] Z. Long et al.: *Approximating Region Boundaries Based on Qualitative and Quantitative Information*. IEEE Intell. Syst. 37 (2022)

Applications: Drone Monitoring

Figure: "Never fly over an urban area for more than 3 minutes": $\forall r \in$ UrbanRegion, $\Box(PO \vee TPP \vee NTPP(\text{Drone}, r) \rightarrow \Diamond_{[0,180s]} DC(\text{Drone}, r))$[10]

[10] F. Heintz, D. de Leng: *Spatio-Temporal Stream Reasoning with Incomplete Spatial Information*. In: ECAI 2014

Qualitative Constraint Languages

Qualitative Constraint Language: Definition

Definition

A qualitative constraint language is based on a finite set B *of base relations with the following properties:*

- *the base relations are defined over an infinite domain* D
- *the base relations are jointly exhaustive & pairwise disjoint (*JEPD*)*
- B *contains the identity relation* Id
- B *is closed under the converse operation* $(^{-1})$
- 2^{B} *is equipped with the usual set-theoretic operations union and intersection, the converse operation, and the weak composition operation* $(\diamond)$

We will look into these notions in detail in the next slides!

Base Relations: Domain

- They are defined over an infinite domain D and have the same arity ϵ, for some integer $\epsilon > 1$; for this course, $\epsilon = 2$
- D itself represents elements that correspond to spatial or temporal entities (e.g., $\mathbb{R}$, $\mathbb{Q}^2$, and so on)

Example

For $\mathrm{D} = \mathbb{R}$, *a binary* ($\epsilon = 2$) *base relation* $b \in \mathrm{B}$ *called "less than" could have the following form:* $b = \{(1.001, 3.9), (2.12, 2.121), (0.01, 11.2), \ldots\}$

Note that, because of the infinitity of the domain D, a base relation is typically a set of infinite size!

Base Relations: JEPD Property

- They are *jointly exhaustive and pairwise disjoint* (JEPD):
  - $\bigcup\{b \in \mathrm{B}\} = \overbrace{\mathrm{D} \times \ldots \times \mathrm{D}}^{\epsilon \text{ times}}$ (jointly exhaustive)
  - $\forall b, b' \in \mathrm{B}$ such that $b \neq b'$, we have that $b \cap b' = \emptyset$ (pairwise disjoint)
- In other words: A tuple of ϵ elements of D can appear in / satisfy[11] *at most one single* base relation $b \in \mathrm{B}$, i.e., each base relation represents a distinct set of tuples

[11] We will define this term later on

Base Relations: Expressiveness

- Every base relation $b \in \mathrm{B}$ encodes the definite knowledge between any two or more entities

- A union of base relations $b_1 \cup \ldots \cup b_j$ with $j \leq |\mathrm{B}|$ encodes indefinite knowledge and is represented by the set $\{b_1, \ldots, b_j\}$

- Without any ambiguity, B will also denote the *universal relation*, which is the union of all the base relations (i.e., D^{ϵ})

Note that 2^{B} expresses all relations (definite and indefinite knowledge)

Base Relations: Satisfaction

- A tuple of ϵ elements $(x_1, \ldots, x_\epsilon) \in \mathrm{D}^\epsilon$ *satisfies* a base relation $b \in \mathrm{B}$, denoted by $b(x_1, \ldots, x_\epsilon)$, if and only if $(x_1, \ldots, x_\epsilon) \in b$
- Likewise, for a relation $r \in 2^{\mathrm{B}}$, the tuple *satisfies* relation r, denoted by $r(x_1, \ldots, x_\epsilon)$, if and only if $\exists b \in r$ such that $b(x_1, \ldots, x_\epsilon)$
- When $\epsilon = 2$, like in this course, the infix notation may be used: $x\ b\ y$ and $x\ r\ y$ will correspond to $b(x, y)$ and $r(x, y)$ respectively

Example

If we consider the "less than" base relation of the previous example, $(3.01, 5.13)$ *would be a tuple that satisfies it, whereas* $(5.13, 3.01)$ *one that would not.*

Base Relations: Identity Relation

- It is a relation $r \in 2^{\mathsf{B}}$ that serves as the identity relation for D^{ϵ},[12] denoted by Id

- Typically, and for sure in this course, Id corresponds to a single base relation, i.e., $\mathsf{Id} = b$ for some $b \in \mathsf{B}$

- We can consider it as all tuples of D^{ϵ} whose elements are all *equal* to one another

- It is assumed that $\mathsf{B} \supset \{\mathsf{Id}\}$ (i.e., we are dealing with non-trivial languages)

[12] This will become clearer later on when discussing relational operations

Base Relations: Point Algebra

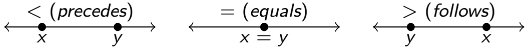


Figure: The 3 base relations of Point Algebra; $>$ is the inverse of $<$

- $\mathsf{D} = \mathbb{Q}$ (i.e., the set of rational numbers)
- $\mathsf{B} = \{<, = \ (= \mathsf{Id}), >\}$
- $2^{\mathsf{B}} = \{\{<,=,>\}, \{<,>\}, \{<,=\}, \{=,>\}, \{<\}, \{>\}, \{=\}, \emptyset\}$
- Arity $\epsilon = 2$; this will always be the case in this course from now on!

Example

$\mathit{precedes} = \{(x, y) \in \mathbb{D}^2 \mid x < y\}$[13]

[13] '$<$' here is a comparison operator and, hence, different from the '$<$' *symbol* in the figure, albeit they convey the same semantics (any symbol can be chosen, but we opt for intuitive ones)

Base Relations: Allen's Interval Algebra

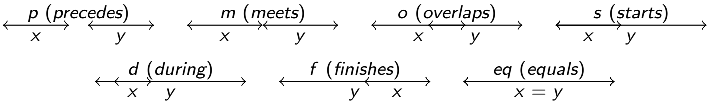


Figure: The 13 base relations of Interval Algebra; inverses are omitted in the figure

- $\mathrm{D} = \{x = (x^-, x^+) \in \mathbb{Q}^2 \mid x^- < x^+\}$
- $\mathrm{B} = \{eq\ (= \mathrm{Id}),\ p,\ pi,\ m,\ mi,\ o,\ oi,\ s,\ si,\ d,\ di,\ f,\ fi\}$

Example

$during = \{(x, y) \in \mathrm{D}^2 \mid x^- > y^- \ \wedge\ x^+ < y^+\}$

Base Relations: RCC8

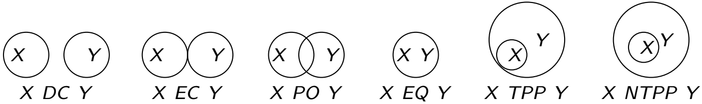


Figure: The 8 base relations of RCC8; inverses are omitted in the figure

- D = the set $\mathcal{T}_{\text{reg}}$ of all non-empty regular closed subsets of some topological space $\mathcal{T}$, in particular the spaces $\mathbb{R}^n$ for some $n \geq 1$
- B = $\{DC, EC, EQ\ (= \mathrm{Id}), PO, TPP, TPPi, NTPP, NTPPi\}$[14]

[14] Respectively, *disconnected*, *externally connected*, *equals*, *partially overlaps*, *tangential proper part*, *tangential proper part inverse*, *non-tangential proper part*, *non-tangential proper part inverse*

Base Relations: RCC8 Origins

Table: Definition of the relations of RCC; relations in bold are included in RCC8

Relation	Description	Definition
$C(x, y)$	*connects with*	primitive relation
$\mathbf{DC}(x, y)$	*disconnected*	$\neg C(x, y)$
$P(x, y)$	*part*	$\forall z(C(z, x) \rightarrow C(z, y))$
$PP(x, y)$	*proper part*	$P(x, y) \wedge \neg P(y, x)$
$\mathbf{EQ}(x, y)$	*equals*	$P(x, y) \wedge P(y, x)$
$O(x, y)$	*overlaps*	$\exists z(P(z, x) \wedge P(z, y))$
$\mathbf{PO}(x, y)$	*partially overlaps*	$O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$
$DR(x, y)$	*discrete*	$\neg O(x, y)$
$\mathbf{TPP}(x, y)$	*tangential proper part*	$PP(x, y) \wedge \exists z(EC(z, x) \wedge EC(z, y))$
$\mathbf{EC}(x, y)$	*externally connected*	$C(x, y) \wedge \neg O(x, y)$
$\mathbf{NTPP}(x, y)$	*non-tangential proper part*	$PP(x, y) \wedge \neg\exists z(EC(z, x) \wedge EC(z, y))$
$Pi(x, y)$	*part inverse*	$P(y, x)$
$PPi(x, y)$	*proper part inverse*	$PP(y, x)$
$\mathbf{TPPi}(x, y)$	*tangential proper part inverse*	$TPP(y, x)$
$\mathbf{NTPPi}(x, y)$	*non-tangential proper part inverse*	$NTPP(y, x)$

Relational Operations: Converse

- As a reminder, B is closed under the *converse* operation $(^{-1})$
- In this course, it is assumed that the converse of any base relation $b \in \mathrm{B}$ is itself a base relation of B[15]
- The converse of $b \in \mathrm{B}$ is defined as $b^{-1} = \{(y, x) \mid (x, y) \in b\}$
- The converse of $r \in 2^{\mathrm{B}}$ is defined as $r^{-1} = \bigcup\{b^{-1} \mid b \in r\}$

We already saw examples of converse (inverse) relations in previous slides

[15] As a side note, there exist languages for which the converse of a base relation corresponds to a relation comprising more than one base relation

Relational Operations: Converse Tables

Table: Converse tables for Point Algebra, Interval Algebra, and RCC8, respectively

$\mathbf{b}$	$\mathbf{b}^{-1}$
$<$	$>$
$>$	$<$
$=$	$=$

$\mathbf{b}$	$\mathbf{b}^{-1}$
p	*pi*
pi	*p*
o	*oi*
oi	*o*
m	*mi*
mi	*m*
d	*di*
di	*d*
si	*s*
s	*si*
f	*fi*
fi	*f*
eq	*eq*

$\mathbf{b}$	$\mathbf{b}^{-1}$
DC	*DC*
EC	*EC*
PO	*PO*
TPP	*TPPi*
TPPi	*TPP*
NTPP	*NTPPi*
NTPPi	*NTPP*
EQ	*EQ*

We can summarize and maintain all results regarding converse in tables

Relational Operations: Union and Intersection

- Given $r, r' \in 2^{\mathsf{B}}$, $r \cup r'$ is the relation of 2^{B} that comprises the base relations of B that exist in either r *or* r' (set union)

- Given $r, r' \in 2^{\mathsf{B}}$, $r \cap r'$ is the relation of 2^{B} that comprises the base relations of B that exist in both r *and* r' (set intersection)

Example

$\{<,=\} \cup \{=,>\} = \{<,=,>\}$, *whereas* $\{<,=\} \cap \{=,>\} = \{=\}$

Note that the intersection may produce the empty relation $\emptyset$; what would that mean?

Relational Operations: Composition

Given two base relations $b, b' \in \mathsf{B}$, we have:

$$b \circ b' = \{(x, y) \in \mathsf{D}^2 \mid \exists z \in \mathsf{D} \text{ such that } (x, z) \in b \wedge (z, y) \in b'\}$$

Given two relations $r, r' \in 2^{\mathsf{B}}$, we have:

$$r \circ r' = \bigcup \{b \circ b' \mid b \in r,\ b' \in r'\}$$

Example

$\{<, =\} \circ \{<\} = \{<\}$
(But how did we compute this since base relations are inifinite sets?)

Relational Operations: Composition Semantics

- Given the two relations between two entities x and z, and z and y, respectively, composition infers the third relation between x and z

- Formally, we must consider an infinite number of tuples to compute composition results, which is unfeasible

- But for well-structured domains (such as points, which form ordered domains) they can be computed using the semantics of the relations

Relational Operations: Composition Problem

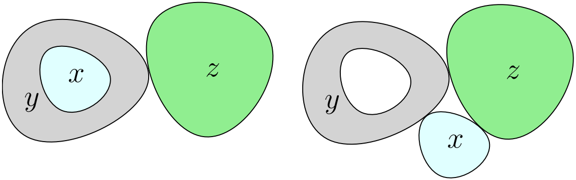


Figure: Two spatial configurations using regions of RCC8

Does $\{EC\} \circ \{EC\} \supseteq \{EC\}$ hold?

Relational Operations: Weak Composition

Given two base relations $b, b' \in \mathsf{B}$, we have:

$$b \diamond b' = \{b'' \in \mathsf{B} \mid b'' \cap (b \circ b') \neq \emptyset\}$$

Given two relations $r, r' \in 2^{\mathsf{B}}$, we have:

$$r \diamond r' = \bigcup \{b \diamond b' \mid b \in r,\ b' \in r'\}$$

Example

$\{EC\} \diamond \{EC\} \supseteq \{EC\}$, *and, of course,* $\{<, =\} \diamond \{<\} = \{<\}$

Relational Operations: Weak Composition Semantics

- The weak composition result of two base relations $b, b' \in \mathsf{B}$ is defined as the smallest relation $r \in 2^{\mathsf{B}}$ that includes $b \circ b'$

- For any given qualitative constraint language it holds that, for any $r, r' \in 2^{\mathsf{B}}$, $r \circ r' \subseteq r \diamond r'$

- For well-structured domains we get the $=$ part of $\subseteq$, and for vague ones (such as arbitrary spatial regions) we get the $\subset$ part of $\subseteq$

Relational Operations: RCC8 Weak Composition Table

⋄	DC	EC	PO	TPP	NTPP	TPPi	NTPPi	EQ
DC	B	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*DC*	*DC*	*DC*
EC	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*DC*, *EC*, *PO*, *TPP*(*i*), *EQ*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*EC*, *PO*, *TPP*, *NTPP*	*PO*, *TPP*, *NTPP*	*DC*, *EC*	*DC*	*EC*
PO	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	B	*PO*, *TPP*, *NTPP*	*PO*, *TPP*, *NTPP*	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*PO*
TPP	*DC*	*DC*, *EC*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*TPP*, *NTPP*	*NTPP*	*DC*, *EC*, *PO*, *TPP*(*i*), *EQ*	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*TPP*
NTPP	*DC*	*DC*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	*NTPP*	*NTPP*	*DC*, *EC*, *PO*, *TPP*, *NTPP*	B	*NTPP*
TPPi	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*EC*, *PO*, *TPPi*, *NTPPi*	*PO*, *TPPi*, *NTPPi*	*PO*, *TPP*(*i*), *EQ*	*PO*, *TPP*, *NTPP*	*TPPi*, *NTPPi*	*NTPPi*	*TPPi*
NTPPi	*DC*, *EC*, *PO*, *TPPi*, *NTPPi*	*PO*, *TPPi*, *NTPPi*	*PO*, *TPPi*, *NTPPi*	*PO*, *TPPi*, *NTPPi*	*PO*, *TPP*(*i*), *NTPP*(*i*), *EQ*	*NTPPi*	*NTPPi*	*NTPPi*
EQ	*DC*	*EC*	*PO*	*TPP*	*NTPP*	*TPPi*	*NTPPi*	*EQ*

Relation Algebras

Table: Axioms for relation algebras, where $r, s, t \in 2^{\mathsf{B}}$

Axiom	Definition
$\cup$-commutativity	$r \cup s = s \cup r$
$\cup$-associativity	$r \cup (s \cup t) = (r \cup s) \cup t$
Huntington axiom	$\overline{\overline{r} \cup \overline{s}} \cup \overline{\overline{r} \cup s} = r$
$\diamond$-associativity	$r \diamond (s \diamond t) = (r \diamond s) \diamond t$
$\diamond$-distributivity	$(r \cup s) \diamond t = (r \diamond t) \cup (s \diamond t)$
identity law	$r \diamond \mathsf{Id} = r$
$^{-1}$-involution	$(r^{-1})^{-1} = r$
$^{-1}$-distributivity	$(r \cup s)^{-1} = r^{-1} \cup s^{-1}$
$^{-1}$-involutive distributivity	$(r \diamond s)^{-1} = s^{-1} \diamond r^{-1}$
Tarski/de Morgan axiom	$r^{-1} \diamond \overline{r \diamond s} \cup \overline{s} = \overline{s}$

If a qualitative constraint language satisfies the above axioms, it is a *relation algebra* in the sense of Tarski[16]

[16] A. Tarski: *On the calculus of relations*. J. Symb. Log. (1941)

Relation Algebras: Result

Proposition

Each of the following qualitative constraint languages is a relation algebra with the respective algebraic structure $(2^{\mathsf{B}}, \mathsf{Id}, \diamond, ^{-1})$*:*

- *Point Algebra*
- *Interval Algebra*
- RCC8

Of course, many more calculi exist, and new ones may be constructed, with the above property

Relation Algebras: Importance

- Several reasoning optimizations become possible
- Some examples follow:
  - $r \diamond s \diamond t$ for $r, s, t \in 2^{\mathsf{B}}$ yields the same result both from left to right and from right to left ($\diamond$-associativity)
  - only one of the constraints $x\ r\ y$ or $x\ r^{-1}\ y$ needs to be stored, as any of the two can be reconstructed from the other ($^{-1}$-involution)

Relation Algebras: Peircean law / De Morgan's Theorem K

From the relation algebra axioms we can obtain other useful rules

$$(r \diamond s) \cap t^{-1} \iff (s \diamond t) \cap r^{-1}$$

Example

Let us consider that $r = \{<\}$, $s = \{<\}$, *and* $t = \{<\}$, *then:*

- $(r \diamond s) \cap t^{-1} = (\{<\} \diamond \{<\}) \cap \{>\} = \{<\} \cap \{>\} = \emptyset$
- $(s \diamond t) \cap r^{-1} = (\{<\} \diamond \{<\}) \cap \{>\} = \{<\} \cap \{>\} = \emptyset$

OK.. So, how does this help us?[17]

[17] This will become more obvious when talking about qualitative constraint networks

Subclass of Relations: Definition

- 2^{B} is by definition closed under weak composition, union, intersection, and converse

- In the context of refinement algorithms that we will see later on, union is not important to us

Definition

A subclass of relations is a subset $\mathcal{A} \subseteq 2^{\mathsf{B}}$ that contains the singleton relations of 2^{B} and is closed under converse ($^{-1}$), intersection ($\cap$), and weak composition ($\diamond$)

Clearly, the entire set of relations 2^{B} is also a subclass of relations

Subclass of Relations: Usage

Depending on the subclass used:

- spatio-temporal information may be tractable to reason with, or not
- stronger properties may be defined, like *weak global consistency*
- more refined theoretical analysis can be perfomed
- faster reasoning algorithms can be implemented, tailored to that subclass

The notion of subclasses of relations will become clearer as the course progresses

Reasoning with Qualitative Constraint Networks

Qualitative Constraint Network: Definition

Spatial or temporal information for a set of entities can be represented by a qualitative constraint network (QCN)

Definition

*A qualitative constraint network (*QCN*) of some qualitative constraint language is a tuple* (V, C) *where:*

- V *is a set of variables over the infinite domain* D *of the language;*
- *and* C *is a mapping* $C : V \times V \rightarrow 2^{\mathrm{B}}$ *associating a relation (set of base relations) of the language with each pair of variables in* V

By definition, a QCN is defined w.r.t some qualitative constraint language, like Point Algebra and so on[18]

[18] This becomes obvious through the use of the set B

Qualitative Constraint Network: Assumptions

- Clearly, by definition our QCNs are binary
- Further, $\forall v \in V,\ C(v, v) = \{\mathsf{Id}\}$
- Last, QCNs are *normalized*: $\forall v, v' \in V,\ C(v, v') = (C(v', v))^{-1}$

By taking into account the aforementioned assumptions, it can be deduced that we get 2-consistency for free[19]

[19] How could we enforce 2-consistency otherwise?

Qualitative Constraint Network: 2-consistency

Let us consider a non-normalized QCN where we have the constraints:

$$C(i, j) = \{<, =\} \text{ and } C(j, i) = \{<, =\}$$

This would mean that we want to have:

$i \{<, =\} j$ and $j \{<, =\} i$ (impossible to strictly order the variables)

We can enforce the normalization condition by performing:

$$C(i, j) \leftarrow (C(j, i))^{-1} \cap C(i, j)$$

$$C(j, i) \leftarrow (C(i, j))^{-1} \cap C(j, i)$$

Qualitative Constraint Network: Example (1/2)

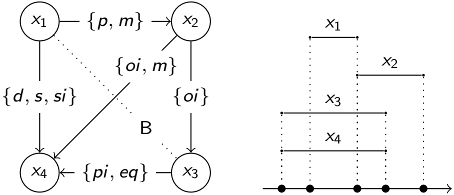


Figure: A QCN of Interval Algebra along with a solution

In what follows, for conciseness, converse relations (reverse arcs) or Id loops are not shown in the figures of QCNs

Qualitative Constraint Network: Example (2/2)

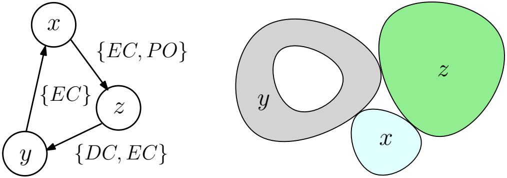


Figure: A QCN of RCC8 along with a solution

Qualitative Constraint Network: More Definitions :)

- A QCN $\mathcal{N} = (V, C)$ is *trivially inconsistent* iff $\exists v, v' \in V$ such that $C(v, v') = \emptyset$
- A *solution* of a QCN $\mathcal{N} = (V, C)$ is a mapping $\sigma : V \rightarrow \mathrm{D}$ such that, $\forall v, v' \in V$, $\exists b \in C(v, v')$ such that $(\sigma(v), \sigma(v')) \in b$
- $\mathcal{N}$ is *satisfiable* (or *consistent*)[20] if and only if it admits a solution
- A *sub*-QCN[21] $\mathcal{N}'$ of $\mathcal{N}$, denoted by $\mathcal{N}' \subseteq \mathcal{N}$, is a QCN (V, C') such that, $\forall u, v \in V$, $C'(u, v) \subseteq C(u, v)$
- A *scenario* of $\mathcal{N}$ is a consistent atomic sub-QCN $\mathcal{S}$ of $\mathcal{N}$, where a QCN $\mathcal{S} = (V, C')$ is *atomic* iff, $\forall v, v' \in V$, $|C'(v, v')| = 1$

[20] What is the difference between the two terms, if any?

[21] This term can also be found under the name *refined* QCN or *refinement* throughout the literature

Qualitative Constraint Network: Qualitative Solutions

- We are doing *qualitative* reasoning!
- In general, we will deal with qualitative solutions, i.e., *scenarios*, as defined earlier
- Solutions will still be important to us to understand the particularities of a domain D

Qualitative Constraint Network: Example Scenario

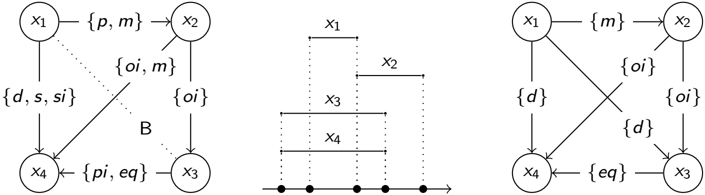


Figure: A QCN of Interval Algebra along with a solution *and a scenario of it*

Reasoning Problems of QCNs: Satisfiability Checking

Definition

The satisfiability checking problem of a QCN $\mathcal{N}$ *is deciding whether* $\mathcal{N}$ *is satisfiable, i.e., whether it admits a solution*

The satisfiability checking problem is NP-complete for most calculi[22]

[22] Notably, it is PTIME for Point Algebra

Satisfiability Checking: Interval Algebra

- NP-hardness follows from a polynomial-time many-one reduction (Karp reduction) from 3-SAT[23]
- 3-SAT formulas: $(l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge \ldots \wedge (l_{i,1} \vee l_{i,2} \vee l_{i,3})$
- Each literal and its negation in a 3-SAT formula is associated with a pair of intervals
- The above two intervals are then related to a "truth determining" third interval `middle`:
  - if an interval is *before* `middle` then the corresponding literal is false
  - if an interval is *after* `middle` then the corresponding literal is true
- Finally, each 3-SAT clause is formed in a way such that at most two corresponding intervals are *before* `middle`

[23] M. Vilain et al.: *Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report*. Readings in Qualitative Reasoning About Physical Systems. Morgan Kaufmann, 1990

Reasoning Problems of QCNs: Minimal Labeling

Definition

Given a QCN $\mathcal{N} = (V, C)$ *and a constraint* $C(u, v)$ *with* $u, v \in V$, *the minimal labeling problem is deciding if* $C(u, v)$ *contains unfeasible base relations (i.e., base relations that do not appear in any scenario of* $\mathcal{N}$*)*

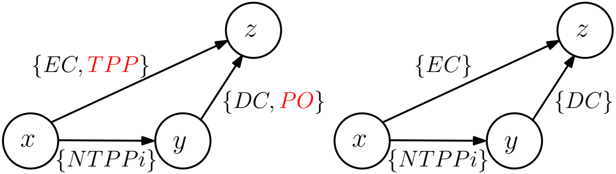


Figure: A RCC8 network (left) and its minimal network (right)

The minimal labeling problem is polynomial-time Turing reducible (Cook reducible) to the satisfiability checking problem

Reasoning Problems of QCNs: Redundancy

Definition

Given a QCN $\mathcal{N} = (V, C)$ *and a constraint* $C(u, v)$ *with* $u, v \in V$*, the redundancy problem is deciding if* $C(u, v)$ *is entailed by the rest of the constraints of* $\mathcal{N}$ *(i.e., it is redundant in* $\mathcal{N}$*)*

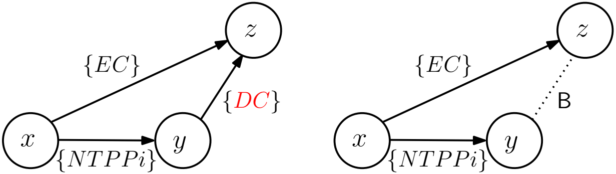


Figure: A RCC8 network (left) and its prime network (right)

Similarly to the minimal labeling problem, the redundancy problem is polynomial-time Turing reducible to the satisfiability checking problem

Reasoning Problems of QCNs: Note on Turing Reductions

- A $\leq_T$ B: An algorithm that solves problem A using an oracle for problem B

- A $\leq_T^P$ B: A $\leq_T$ B that uses a polynomial number of calls to the oracle for problem B, and polynomial time outside of those calls

- NP is NOT closed under polynomial-time Turing reductions (unless NP = co-NP)

Local Consistencies: Usage

- Approximate, or even decide, satisfiability

- Simplify a QCN / prune search space

- Realize forward-checking in a backtracking algorithm

Reminder: Weak Composition Operation ($\diamond$)

$EC \diamond NTPP$ yields the set of base relations $\{NTPP, TPP, PO\}$

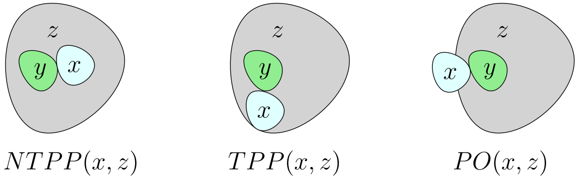


Figure: Possible types of configurations for regions x, y, z w.r.t $EC(x, y) \diamond NTPP(y, z)$

We can precompute and store all weak composition outputs in memory

$\diamond_G$-Consistency: Definition

Definition

Given a QCN $\mathcal{N} = (V, C)$ *and a graph* $G = (V, E)$, $\mathcal{N}$ *is* $\diamond_G$-consistent *iff,* $\forall$ $\{v_i, v_j\}$, $\{v_i, v_k\}$, $\{v_k, v_j\} \in E$, $C(v_i, v_j) \subseteq C(v_i, v_k) \diamond C(v_k, v_j)$

Intuitively, $\diamond_G$-consistency checks if all triples of variables in $\mathcal{N}$ that correspond to triangles in G are closed under weak composition[24]

[24] M. Sioutis et al.: An Efficient Approach for Tackling Large Real World Qualitative Spatial Networks. Int. J. Artif. Intell. Tools 25 (2016)

$\diamond_G$-Consistency: Complete Graph and Example

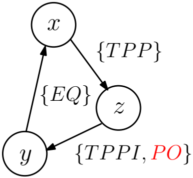


Figure: A QCN of RCC8 that is not $\diamond$-consistent

$\diamond$-consistency denotes $\diamond_G$-consistency where G is a complete graph

$\diamond_G$-Consistency: Importance

For each of the qualitative constraint languages of Point Algebra, Interval Algebra, and RCC8[25] we have:

Property

Every $\diamond$-consistent atomic QCN *of Point Algebra, Interval Algebra, or* RCC8 *is satisfiable*

In fact, later on we will see that the above result holds for $\diamond_G$-consistency too, when G satisfies certain criteria other than being complete

[25] Many more calculi exist with this property, but these are the ones we focus on in this course

$\diamond_G$-Consistency: Complexity of Checking

Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, we need to check if:

$$\forall \{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E, C(v_i, v_j) \subseteq (v_i, v_k) \diamond C(v_k, v_j)$$

- Basically, and as mentioned earlier, we need to visit all triangles in G
- Thus, runtime is: $O(\Delta \cdot |E|)$, where Δ is the maximum degree of G
- If G is a complete graph, we get $O(|V|^3)$ (why?)

$\overset{\diamond}{G}$-Consistency: Complexity of Enforcing

Given a QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$, we need to iteratively perform the operation below until a fixed state is reached:

$$\forall\{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E, C(v_i, v_j) \leftarrow C(v_i, v_j) \cap ((v_i, v_k) \diamond C(v_k, v_j))$$

- The above procedure is called *algebraic closure (under $\overset{\diamond}{G}$-consistency)*
- It is essentially path consistency as in CSPs, where $\circ$ is replaced by $\diamond$
- Runtime depends on how it will be implemented![26]

[26] We will see two ways in what follows

Algebraic Closure: Soundness and Completeness

$\diamond_G(\mathcal{N})$ denotes the algebraic closure of $\mathcal{N}$ under $\diamond_G$-consistency

Property (Soundness)

Given a QCN $\mathcal{N} = (V, C)$ *and a graph* $G = (V, E)$, *if* $\emptyset \in \diamond_G(\mathcal{N})$, *then the* QCN *is unsatisfiable*

In general, the algebraic closure is NOT complete for deciding satisfiability of a QCN![27]

[27] Unless, with what we have seen so far, the QCN is refined to an $\diamond$-consistent atomic sub-QCN; later this result will be generalized to certain non-atomic QCNs too

Algebraic Closure: Properties w.r.t $\diamond_G$-Consistency

- $\diamond_G(\mathcal{N})$ is the largest[28] $\diamond_G$-consistent sub-QCN of $\mathcal{N}$ (Dominance)
- $\diamond_G(\mathcal{N})$ is equivalent[29] to $\mathcal{N}$ (Equivalence)
- $\diamond_G(\diamond_G(\mathcal{N})) = \diamond_G(\mathcal{N})$ (Idempotence)
- if $\mathcal{N}' \subseteq \mathcal{N}$ then $\diamond_G(\mathcal{N}') \subseteq \diamond_G(\mathcal{N})$ (Monotonicity)

[28] W.r.t $\subseteq$

[29] Two QCNs are equivalent if the have the same set of solutions

Algebraic Closure: Naive Approach

1. We perform $C(v_i, v_j) \leftarrow C(v_i, v_j) \cap ((v_i, v_k) \diamond C(v_k, v_j))$ for each triple $\{v_i, v_j, v_k\}$ s.t. $\{v_i, v_j\}, \{v_i, v_k\}, \{v_k, v_j\} \in E$

2. If some constraint $C(v_i, v_j)$ was revised, we repeat Step 1, otherwise we are done

We revise (Step 1) at most $O(|E|)$ constraints, $O(|\mathrm{B}|)$ times, so we get:

$$O(\Delta \cdot |E|^2 \cdot |\mathrm{B}|)$$

Note: The revision here (Step 1) costs $O(\Delta \cdot |E|)$ time

Algebraic Closure: SOTA Algorithm (1/3)

- We start by performing Step 1 of the naive approach (close triples of variables under weak composition)
- If some constraint $C(v_i, v_j)$ was revised, we *only* visit the constraints that may be affected by this revision
- The possibly affected constraints are the ones forming a triangle with $C(v_i, v_j)$
- Basically, for $C(v_i, v_j)$, we consider all v_k s.t. $\{v_i, v_k\}, \{v_k, v_j\} \in E$
- The number of possibly affected constraints is upper bounded by Δ, i.e., it is $O(|\Delta|)$

Algebraic Closure: SOTA Algorithm (2/3)

Algorithm 1: PWC($\mathcal{N}$, G)

in : A QCN $\mathcal{N} = (V, C)$ and a graph $G = (V, E)$.
output : $\diamond_G(\mathcal{N})$.

begin
  $Q \leftarrow E$;
  while $Q \neq \emptyset$ **do**
    $\{v, v'\} \leftarrow Q.pop()$;
    foreach $v'' \in V \mid \{v, v''\}, \{v', v''\} \in E$ **do**
      $r \leftarrow C(v, v'') \cap (C(v, v') \diamond C(v', v''))$;
      if $r \subset C(v, v'')$ **then**
        $C(v, v'') \leftarrow r$;
        $C(v'', v) \leftarrow r^{-1}$;
        $Q \leftarrow Q \cup \{\{v, v''\}\}$;
      $r \leftarrow C(v'', v') \cap (C(v'', v) \diamond C(v, v'))$;
      if $r \subset C(v'', v')$ **then**
        $C(v'', v') \leftarrow r$;
        $C(v', v'') \leftarrow r^{-1}$;
        $Q \leftarrow Q \cup \{\{v'', v'\}\}$;
  return $\mathcal{N}$;

Algebraic Closure: SOTA Algorithm (3/3)

We revise at most $O(|E|)$ constraints, $O(|\mathrm{B}|)$ times, so we get:

$$O(\Delta \cdot |E| \cdot |\mathrm{B}|)$$

Note: The revision here costs $O(\Delta)$ time[30]

[30] Contrast this with $O(\Delta \cdot |E|)$ of the naive approach

Algebraic Closure: Sum-Up

- Naive approach: $O(\Delta \cdot |E|^2 \cdot |\mathrm{B}|)$ time[31]
- SOTA approach: $O(\Delta \cdot |E| \cdot |\mathrm{B}|)$ time[32]
- What about space complexity?
  - Naive approach: $O(1)$ (why?)
  - SOTA approach: $O(|E|)$ (maintained by a queue)
- What about best case complexity? $\Omega(\Delta \cdot |E|)$ for both

[31] $O(|V|^5 \cdot |\mathrm{B}|)$ time when G is complete
[32] $O(|V|^3 \cdot |\mathrm{B}|)$ time when G is complete

Algebraic Closure: Heuristics

- Simple choice between using a FIFO or LIFO queue
- Prioritization of constraints based on their *constrainedness*:
  - Static schemes, e.g., $\{=\}$ is more rectrictive than $\{<\}$ (why?);
  - Dynamic schemes, e.g., counting local models[33]

[33] M. Sioutis et al.: *Dynamic branching in qualitative constraint-based reasoning via counting local models*. Inf. Comput. 281 (2021)

Tractable Subclasses of Relations: Definition

Reminder of subclass of relations: A *subclass of relations* is a subset $\mathcal{A} \subseteq 2^{\mathsf{B}}$ that contains the singleton relations of 2^{B} and B and is closed under converse $(^{-1})$, intersection $(\cap)$, and weak composition $(\diamond)$

Definition

A subclass of relations $\mathcal{A}$ is tractable iff the class of QCN*s defined over $\mathcal{A}$ is tractable, i.e., the satisfiability of every* QCN *in that class can be decided in polynomial time*

In this course, we focus on tractability w.r.t $\overset{\diamond}{_G}$-consistency[34]

[34] Indeed, other polytime methods may exist for deciding the satisfiability of a QCN

Tractable Subclasses of Relations: Identification

- Transform sets of qualitative relations to known traclable classes, e.g., Horn or Krom formulas
- Exploit geometrical characterizations of qualitative relations, e.g., *(pre)-convex* relations[35]
- Exclude qualitative relations that lead to NP-hardness, work with the remainings ones
- Implement dedicated polynomial algorithms from scratch and prove their correctness
- ...

[35] G. Ligozat: *A New Proof of Tractability for ORD-Horn Relations*. In: AAAI/IAAI 1996

Tractable Subclasses of Relations: Interval Algebra (1/3)

A Horn theory of Interval Algebra can be based on that of partial orders:

$$\begin{array}{cc} x \leq z \wedge z \leq y \rightarrow x \leq y & x = y \rightarrow x \leq y \\ x \leq y \wedge y \leq x \rightarrow x = y & x = y \rightarrow y \leq x \\ x = y \wedge x \neq y \rightarrow \bot & x \neq x \rightarrow \bot \end{array}$$

Then, every interval variable $x = (x^-, x^+)$ in a QCN can be translated as follows (remember that $x^- < x^+$):

$$x^- \leq x^+ \wedge x^- \neq x^+$$

In addition, for all distinct interval variables x, y, and z in our QCN, and all their endpoints, we need to enforce the theory of partial orders above

Tractable Subclasses of Relations: Interval Algebra (2/3)

- Given a QCN, we can obtain a CNF formula for each of its constraints

- The formula is Horn if it contains only clauses with:

  - *at most one* positive literal, i.e., of the form $x = y$ or $x \leq y$

  - and an arbitrary number of negative literals, i.e., of the form $x \neq y$

- If all the formulas (constraints) are Horn, the QCN is tractable (because of Horn-satisfiability)

Tractable Subclasses of Relations: Interval Algebra (3/3)

A constraint defined by the relation $\{s, o, fi\}$ yields the CNF formula:[36]

$$(x^- \leq y^-) \wedge (y^- \leq x^+) \wedge (y^- \neq x^+) \wedge (x^+ \leq y^+) \wedge (x^- \neq y^- \vee x^+ \neq y^+)$$

A constraint defined by the relation $\{p, pi\}$ yields the CNF formula:

$$(x^+ \leq y^- \vee y^+ \leq x^-) \wedge (x^+ \neq y^-) \wedge (x^- \neq y^+)$$

Which one is Horn? Which is not? Why?

[36] In reality, every clause involving interval variables $x, y \in V$ is captured by a set of propositional variables p_{xy}^s, where $s \in \{-, +\} \times \{\leq, =\} \times \{-, +\}$

Tractable Subclasses of Relations: Remarks

- Typically, every satisfiable QCN defined over such a subclass can be refined to an $\diamond$-consistent atomic sub-QCN in polynomial time
- Likewise, every unsatisfiable QCN will be refined to an empty QCN (all constraints will be defined by $\emptyset$)
- Hence, algebraic closure (under $\diamond$-consistency) becomes a sound *and* complete method for deciding satisfiability!

Tractable Subclasses of Relations: Maximality

Definition

A tractable subclass of relations $\mathcal{A}$ is maximal iff there is not other tractable subclass that properly/strictly contains $\mathcal{A}$

- Once a tractability result is known/proved, other (large) tractable classes may be identified automatically[37]

- However, *maximality* of a tractable subclass requires formal theoretical analysis

[37] J. Renz: *Qualitative Spatial and Temporal Reasoning: Efficient Algorithms for Everyone*. In: IJCAI 2007

Tractable Subclasses of Relations: Distributivity

Given three relations r, rs, $t \in 2^{\mathsf{B}}$, weak composition distributes over intersection if we have that $r \diamond (s \cap t) = (r \diamond s) \cap (r \diamond t)$ and $(s \cap t) \diamond r = (s \diamond r) \cap (t \diamond r)$

Definition

A tractable subclass of relations $\mathcal{A}$ is *distributive* iff weak composition distributes over non-empty intersection $\forall r, s, t \in \mathcal{A}$

Distributive subclasses of relations exhibit *convexity* in Helly's sense[38]

[38] L. Danzer et al.: *Helly's Theorem and Its Relatives*. Proceedings of symposia in pure mathematics: Convexity 7 (1963)

Distributivity: Helly's Property

Definition

A subclass of relations $\mathcal{A} \subseteq 2^{\mathsf{B}}$ is Helly if and only if for any n relations $r_1, r_2, \ldots r_n \in \mathcal{A}$ we have:

$$\bigcap_{i=1}^{n} r_i \neq \emptyset \text{ iff, } \forall i, j \in \{1, \ldots n\},\ r_i \cap r_j \neq \emptyset$$

Then, we have the following result by Long and Li:[39]

Theorem

A subclass of relations $\mathcal{A} \subseteq 2^{\mathsf{B}}$ of a qualitative constraint language that is a relation algebra is distributive if and only if it is Helly

[39] Z. Long and S. Li: *On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi*. In: COSIT 2015