Multigranular Attributes for Relational Database Systems

Stephen J. Hegner
Umeå University, Sweden (retired)
Hegner Consulting, LLC, USA

M. Andrea Rodríguez
University of Concepción, Chile

The Relational Model of Data

- In the relational model, the data are stored in tables.

Employee

FName	MInit	LName	SSN	BDate	Address	Sex	Salary	Super_SSN	DNo
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zeyala	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4

Attributes: The columns are defined by *attributes*, shown in green.

Domain: The *domain* of each attribute is the set of possible values.

- $\text{Dom}(\text{Sex}) = \{\text{M}, \text{F}\}$.
- $\text{Dom}(\text{SSN}) =$ strings of exactly 9 digits.
- $\text{Dom}(\text{BDate}) =$ dates in YYYY-MM-DD format.

Operations; In general, the only intra-domain operations supported are simple comparison (including equality).

Examples: $333445555 < 888665555$; $1955\text{-}12\text{-}08 < 1965\text{-}01\text{-}09$.

The Idea of Multigranular Attributes

Spatio-temporal attributes

Thematic attributes

cdd = ciudad/city
cmn = comuna/county
prv = provincia/province

Place	Time	Births
Concepción_cdd	*Y2016Q1*	b_1
Concepción_cmn	*Y2016Q1*	b_2
Concepción_prv	*Y2016Q1*	b_3
Concepción_cmn	*Y2016*	b_4

Granules: The domain values are called *granules*.

Granular order: The granules of spatial and temporal attributes have inherent order structure.

Spatial containment:

Concepción_cdd $\sqsubseteq$ *Concepción_cmn* $\sqsubseteq$ *Concepción_prv*

Temporal interval containment: *Y2016Q1* $\sqsubseteq$ *Y2016*

Typical constraints: Functional dependency (FD) $\{\text{Place}, \text{Time}\} \rightarrow$ Births, births monotonic w.r.t. space/time, so $b_1 \leq b_2 \leq b_3$, $b_2 \leq b_4$.

Lattice-Like Operations on Granules

Place	Time	Births
Arauco_prv	*Y2016Q1*	b_1
BíoBío_prv	*Y2016Q1*	b_2
Concepción_prv	*Y2016Q1*	b_3
Ñuble_prv	*Y2016Q1*	b_4
BíoBío_rgn	*Y2016Q1*	b_5

Join: The four provinces join to the region.

$$\mathit{BíoBío\_rgn} = \bigsqcup\{\mathit{Arauco\_prv}, \mathit{BíoBío\_prv}, \mathit{Concepción\_prv}, \mathit{Ñuble\_prv}\} \quad .$$

Meet: Distinct provinces are disjoint (six possibilities in all).

$$\bigsqcap\{\mathit{Arauco\_prv}, \mathit{BíoBío\_prv}\} = \bot$$

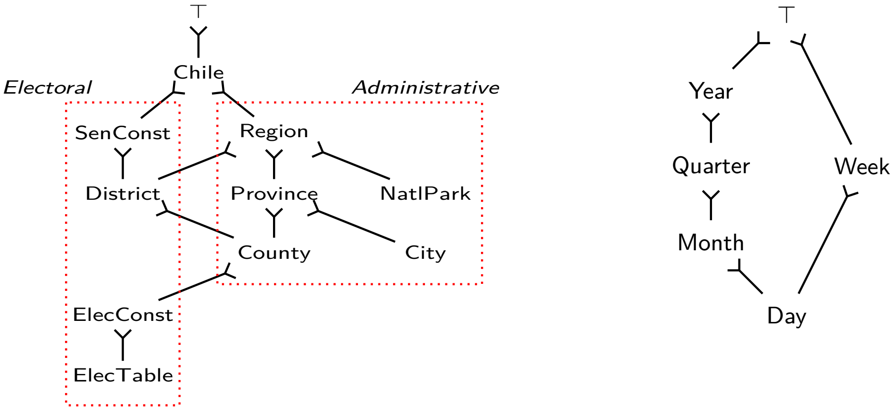
Disjoint Join: The four provinces join *disjointly* to the region.

$$\mathit{BíoBío\_rgn} = \overset{\perp}{\bigsqcup}\{\mathit{Arauco\_prv}, \mathit{BíoBío\_prv}, \mathit{Concepción\_prv}, \mathit{Ñuble\_prv}\}$$

Consequence: $\sum_{i=1}^{4} b_i = b_5$.

Observation: These lattice-like operations are *partial*.

Granularities — Organizing Granules



- The granules of each attribute are partitioned into a hierarchy of *granularities*.

Order: $G_1 \leq G_2 \Leftrightarrow ((\forall g_1 \in \text{Granules}\langle G_1 \rangle)(\exists g_2 \in \text{Granules}\langle G_2 \rangle)(g_1 \sqsubseteq g_2))$.

Disjointness: Distinct granules of the same granularity are disjoint.

Formalizing Granularity Schemata

Granularity schema: $\mathfrak{S} = (\mathbf{Glty}\langle\mathfrak{S}\rangle, \mathbf{Gnle}\langle\mathfrak{S}\rangle, \Pi_{\mathsf{Gnle}}\langle\mathfrak{S}\rangle)$

Granularity preorder: $\mathbf{Glty}\langle\mathfrak{S}\rangle = (\mathsf{Glty}\langle\mathfrak{S}\rangle, \leq_{\mathbf{Glty}\langle\mathfrak{S}\rangle}, \top_{\mathbf{Glty}\langle\mathfrak{S}\rangle})$

Granule preorder: $\mathbf{Gnle}\langle\mathfrak{S}\rangle = (\mathsf{Granules}\langle\mathfrak{S}\rangle, \overline{\sqsubseteq}_{\mathfrak{S}}, \top_{\mathfrak{S}}, \bot_{\mathfrak{S}})$

Granule partition: $\Pi_{\mathsf{Gnle}}\langle\mathfrak{S}\rangle = \{\mathsf{Granules}\langle\mathfrak{S}|G\rangle \mid G \in \mathsf{Glty}\langle\mathfrak{S}\rangle\}$ of $\mathsf{Granules}_{\not\perp}\langle\mathfrak{S}\rangle$

Additional properties:

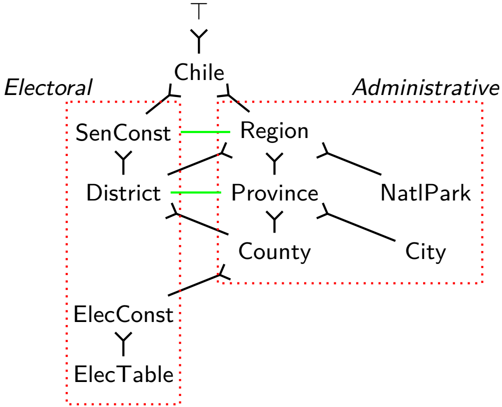
The top granularity consists only of the top granules:
$\mathsf{Granules}\langle\mathfrak{S}|\top_{\mathbf{Glty}\langle\mathfrak{S}\rangle}\rangle = [\top_{\mathfrak{S}}]_{\mathfrak{S}}$ ($[\text{-}]_{\mathfrak{S}}$ = equivalence class under $\sqsubseteq_{\mathfrak{S}}$)

Distinct granules of the same granularity are never equivalent:
$(g_1 \neq g_2 \in \mathsf{Granules}\langle\mathfrak{S}|G\rangle) \Rightarrow ([g_1]_{\mathfrak{S}} \neq [g_1]_{\mathfrak{S}}))$

Distinct granules of the same granularity have nothing in common:
$(g_1 \neq g_2 \in \mathsf{Granules}\langle\mathfrak{S}|G\rangle) \Rightarrow (\mathsf{GLB}_{\mathbf{Gnle}\langle\mathfrak{S}\rangle}\langle\{g_1, g_2\}\rangle = \bot_{\mathfrak{S}})$

Granularity order and granule order: $(G_1 \leq_{\mathbf{Glty}\langle\mathfrak{S}\rangle} G_2) \Leftrightarrow$
$((\forall g_1 \in \mathsf{Granules}\langle\mathfrak{S}|G_1\rangle)(\exists g_2 \in \mathsf{Granules}\langle\mathfrak{S}|G_2\rangle)(g_1 \overline{\sqsubseteq}_{\mathfrak{S}}\ g_2))$

Equivalence of Granularities



Question: Why not make the granularity order partial?

Answer: Some distinct granularities might become identical (with respect to granules) at other points in time.

Near partial order: Require the order instead to be *near partial*:

$$(G_1 \leq_{\mathsf{Glty}\langle\mathfrak{S}\rangle} G_2 \leq_{\mathsf{Glty}\langle\mathfrak{S}\rangle} G_1) \Rightarrow (G_1 \cong G_2).$$

Formalization of Granule Structure

- A *granule structure* is a model for the constraints imposed by the granularity schema.
- $\sigma = (\mathsf{Dom}\langle\sigma\rangle, \mathsf{GnletoDom}_\sigma)$

Domain: $\mathsf{Dom}\langle\sigma\rangle$ is a (not necessarily finite) set.

Granule semantics function: $\mathsf{GnletoDom}_\sigma : \mathsf{Granules}\langle\mathfrak{S}\rangle \rightarrow \mathbf{2}^{\mathsf{Dom}\langle\sigma\rangle}$.

$\bot_\mathfrak{S}$ maps to $\emptyset$: $\mathsf{GnletoDom}_\mathfrak{S}(\bot_\mathfrak{S}) = \emptyset$.

Granule subsumption maps to set inclusion:

$$(g_1 \mathrel{\bar{\sqsubseteq}_\mathfrak{S}} g_2) \Rightarrow (\mathsf{GnletoDom}_\sigma(g_1) \subseteq \mathsf{GnletoDom}_\sigma(g_2)).$$

Distinct granules of the same granularity are disjoint:

$$(\forall G \in \mathsf{Glty}\langle\mathfrak{S}\rangle \setminus \{\top_{\mathsf{Glty}\langle\mathfrak{S}\rangle}\})(\forall g_1, g_2 \in \mathsf{Granules}\langle\mathfrak{S}|G\rangle)$$
$$(g_1 \neq g_2) \Rightarrow (\mathsf{GnletoDom}_\sigma(g_1) \cap \mathsf{GnletoDom}_\sigma(g_2) = \emptyset).$$

Two granules have the same semantics iff they are equivalent under $\bar{\sqsubseteq}_\mathfrak{S}$:

$$(\mathsf{GnletoDom}_\sigma(g_1) = \mathsf{GnletoDom}_\sigma(g_2)) \Leftrightarrow [g_1]_\mathfrak{S} = [g_2]_\mathfrak{S}.$$

Examples of Granule Structure

Example: σ_{Place} for the granularity schema of space.

- $\text{Dom}\langle\sigma\rangle = \mathbb{R}^2$.
- $\text{GnletoDom}_{\text{Place}}(\text{Some\_entity})$

  $=$ the geographic region defining that entity.

Example: σ_{Time} for the granularity schema of time.

- Model all days starting with 1970-01-01.
- $\text{Dom}\langle\sigma\rangle = \mathbb{N}$.

Number days consecutively with 1970-01-01 day zero:

$\text{GnletoDom}_{\text{Time}}(\text{yyyy-mm-dd}) =$

$\{\text{number of days yyyy-mm-dd is after 1970-01-01}\}$.

All other granules consist of a set of days:

$$\text{GnletoDom}_{\text{Time}}(X) = \bigcup\{\text{GnletoDom}_{\text{Time}}(d) \mid d \in X\}.$$

Common properties:

Subsumption: Recaptures the usual notion of spatial/temporal subsumption.

Disjointness: Recaptures the notion *for granules of the same granularity only*.

Canonical Primitive Rules and Their Semantics

Question: How are constraints which are not part of the basic granularity schema modelled?

Rules: All additional constraints are expressed in terms of rules.

Examples:
- Disjointness of granules of different granularities.
- Join constraints: $g \sqsubseteq_{\mathfrak{G}} \bigsqcup_{\mathfrak{G}} S$; $g = \bigsqcup_{\mathfrak{G}} S$; $g = \biguplus_{\mathfrak{G}} S$;

Canonical primitive rules: All rules are defined in terms of those which are of the following two forms.

Basic subsumption rule: $g \sqsubseteq_{\mathfrak{G}} \bigsqcup_{\mathfrak{G}} S$. ($S$ finite and nonempty)

Convention: Regard $g \sqsubseteq_{\mathfrak{G}} g'$ as $g \sqsubseteq_{\mathfrak{G}} \bigsqcup_{\mathfrak{G}} \{g'\}$.

Basic disjointness rule: $\bigsqcap_{\mathfrak{G}} \{g_1, g_2\} = \bot_{\mathfrak{G}}$

Semantics: The semantics of these rules are defined with respect to a granule structure σ using: $\bigsqcup \mapsto \bigcup \quad \bigsqcap \mapsto \bigcap \quad \sqsubseteq \mapsto \subseteq \quad = \mapsto =$.

- $\sigma \in \mathsf{ModelsOf}\langle g \sqsubseteq_{\mathfrak{G}} \bigsqcup_{\mathfrak{G}} S \rangle$ iff $\mathsf{GnletoDom}_{\mathfrak{G}}(g) \subseteq \bigcup_{s \in S} \mathsf{GnletoDom}_{\mathfrak{G}}(s)$.
- $\sigma \in \mathsf{ModelsOf}\langle \bigsqcap_{\mathfrak{G}} \{g_1, g_2\} = \bot_{\mathfrak{G}} \rangle$ iff $\mathsf{GnletoDom}_{\mathfrak{G}}(g_1) \cap \mathsf{GnletoDom}_{\mathfrak{G}}(g_2) = \emptyset$.

Basic Rules and Their Semantics

Basic join rule: $g = \bigsqcup_{\mathfrak{S}} S$ is defined as the conjunction

$$(g \sqsubseteq_{\mathfrak{S}} \bigsqcup_{\mathfrak{S}} S) \wedge (\bigwedge_{s \in S} (s \sqsubseteq_{\mathfrak{S}} g)).$$

Basic disjoint join rule: $g = \biguplus_{\mathfrak{S}} S$ is defined as the conjunction

$$(g = \bigsqcup_{\mathfrak{S}} S) \wedge (\bigwedge_{s_1 \neq s_2 \in S} (\bigsqcap_{\mathfrak{S}} \{s_1, s_2\} = \bot_{\mathfrak{S}})).$$

Basic disjoint subsumption rule: $g \sqsubseteq_{\mathfrak{S}} \biguplus_{\mathfrak{S}} S$ is defined as the conjunction

$$(g \sqsubseteq_{\mathfrak{S}} \bigsqcup_{\mathfrak{S}} S) \wedge (\bigwedge_{s_1 \neq s_2 \in S} (\bigsqcap_{\mathfrak{S}} \{s_1, s_2\} = \bot_{\mathfrak{S}})).$$

- These rules, together with the canonical primitive rules:
  - $g \sqsubseteq_{\mathfrak{S}} \bigsqcup_{\mathfrak{S}} S$
  - $g \sqsubseteq_{\mathfrak{S}} g'$
  - $\bigsqcap_{\mathfrak{S}} \{g_1, g_2\} = \bot_{\mathfrak{S}}$

  are the only ones used in this work.

BaRules$\langle \mathfrak{S} \rangle$: This combined collection is denoted BaRules$\langle \mathfrak{S} \rangle$.

Expression of Constraints

Question: How are constraints expressed in a multigranular attribute?

Two solutions:

Definition by structure: Choose a single granule structure σ, and then take exactly those constraints which hold in σ to be the true ones.

Definition by constraint satisfaction: Given a set Φ of constraints, the set of all constraints which hold are precisely those which hold in every structure in which Φ is satisfied.

- The choice depends upon the multigranular attribute.
  - Definition by structure works best for Time.
  - Definition by constraint satisfaction works best for Place.

Definition by Structure

Idea of definition by structure: The constrained granularity schema $\mathfrak{S}$ is modelled as a single structure $\sigma_{\mathfrak{S}}$.

True rules: The rules which are true are precisely those of $\mathsf{ModelsOf}\langle\sigma_{\mathfrak{S}}\rangle$.

False rules: All other rules are taken to be false.

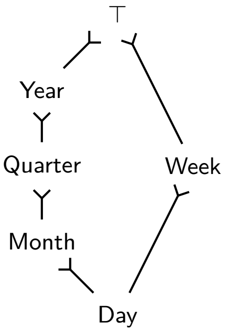
Complete information: There is complete information about which rules are true and which are false.

Example: The granular attribute Time is well suited to definition by structure.

Man made: With a formal, mathematical structure.

Complete information: It is an exact model, not a partial one.

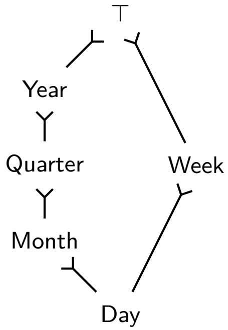
- Recall model from Slide 8.



Recall Structure of Granular Attribute Time

Example: σ_{Time} for the granularity schema of time.

- Model all days starting with 1970-01-01.
- $\text{Dom}\langle\sigma\rangle = \mathbb{N}$.



Number days consecutively with 1970-01-01 day zero:

$$\text{GnletoDom}_{\text{Time}}(\text{yyyy-mm-dd}) = \{\text{number of days yyyy-mm-dd is after 1970-01-01}\}.$$

All other granules consist of a set of days:

$$\text{GnletoDom}_{\text{Time}}(X) = \bigcup\{\text{GnletoDom}_{\text{Time}}(d) \mid d \in X\}.$$

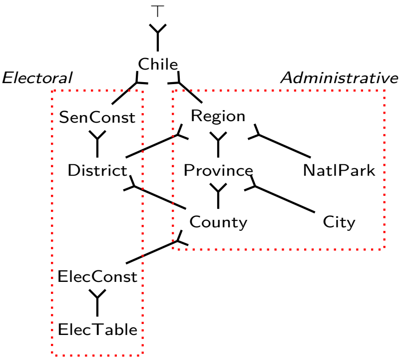
Limitations of Definition by Structure

Possibilities: for single structure σ_{Place}:

- $\text{Dom}\langle\sigma_{\text{Place}}\rangle = \mathbb{R}^2$.
- $\text{Dom}\langle\sigma_{\text{Place}}\rangle =$ a huge set of polygons.

Problems:

- Extremely costly to support.
- Some arbitrary choices necessary.
    - ElecTable (*mesa electoral*).



Observation: The above proposals embody much more information than necessary.

- Only need knowledge of subsumption, disjointness, and join.
- Detailed topography is extraneous.
- ElecTable problem can be solved easily if topography not used.

Solution: Use definition by constraint satisfaction.

Definition by Constraint Satisfaction

Idea of definition by constraint satisfaction: The constrained granularity schema $\mathfrak{S}$ is modelled using a set $\mathsf{Constr}(\mathfrak{S})$ of rules (which include the built-in rules of the schema).

Models: Every $\sigma \in \mathsf{ModelsOf}\langle\mathsf{Constr}(\mathfrak{S})\rangle$ is a possible alternative for the structure.

True rules: The rules φ which are true are precisely those for which $\mathsf{Constr}(\mathfrak{S}) \models_{\mathfrak{S}} \varphi$.

Incomplete information: Little or no information about which rules are false.

Use (partial) CWA to fix incomplete information? Take (some of) those rules which cannot be proven true to be false.

Example of default reasoning from AI 1 course: CWA does not always work.

- Knowledge base is $A \vee B$.
- $A \vee B \not\models A$; $A \vee B \not\models B$;
- But $\{A \vee B, \neg A, \neg B\}$ is not satisfiable.

Good news: It works here, for rules in the multigranular framework.

Armstrong Models

Context: A set $\mathcal{S}$ of sentences (constraints).

Armstrong model: A structure σ is an *Armstrong model (relative to $\mathcal{S}$)* for a consistent set $\Phi \subseteq \mathcal{S}$ if σ is a model of those constraints of $\mathcal{S}$ which are implied by Φ and no others.

Observation: $A \vee B$ has no Armstrong model with

$\mathcal{C}$ = propositional sentences.

Original setting of Armstrong: $\mathcal{S}$ = functional dependencies.

- The ideas work in very general settings [Fagin82].

Theorem: BaRules$\langle \mathfrak{S} \rangle$ admits Armstrong models. $\square$

Utility: Any sentence in $\mathcal{C}$ which is not implied by Φ may be taken to be false without creating a contradiction.

CWA: The (possibly partial) *closed-world assumption* may be applied to BaRules$\langle \mathfrak{S} \rangle$ without risk of inconsistency.

Negating Rules

- A canonical primitive rule consists of only only one conjunct.
  - $g \sqsubseteq_{\mathfrak{S}} \bigsqcup_{\mathfrak{S}} S$. ($S$ finite and nonempty)
  - $\bigsqcap_{\mathfrak{S}} \{g_1, g_2\} = \bot_{\mathfrak{S}}$
- Other basic rules are formed as conjuncts of canonical primitive ones.

Example: $g = \biguplus_{\mathfrak{S}} S$ is defined as the conjunction

$$(g \sqsubseteq_{\mathfrak{S}} \bigsqcup_{\mathfrak{S}} S) \wedge (\bigwedge_{s \in S}(s \sqsubseteq_{\mathfrak{S}} g)) \wedge (\bigwedge_{s_1 \neq s_2 \in S}(\bigsqcap_{\mathfrak{S}} \{s_1, s_2\} = \bot_{\mathfrak{S}})).$$

Theory: Such compound rules may be negated safely.

Practical problem: It will not be known which of the conjuncts are false.

$$\neg(\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n) \equiv (\neg\varphi_1) \vee (\neg\varphi_2) \vee \ldots \vee (\neg\varphi_n)$$

Policy: Only canonical primitive rules may be negated.

- For rules defined by conjunction, it must be stated explicitly which conjuncts are false.

Form of constraints: $\langle \mathrm{Constr}\langle\mathfrak{S}\rangle, \mathrm{cwa}\langle\mathfrak{S}\rangle\rangle$.

- $\mathrm{Constr}\langle\mathfrak{S}\rangle$ consists of basic rules.
- $\mathrm{cwa}\langle\mathfrak{S}\rangle$ consists of canonical primitive rules (negations to hold).

Satisfiability

Recall context: $\langle \mathsf{Constr}\langle \mathfrak{S} \rangle, \mathsf{cwa}\langle \mathfrak{S} \rangle \rangle$.

Question: How to determine whether even $\mathsf{Constr}\langle \mathfrak{S} \rangle$ is satisfiable.

☹ This problem is NP-very hard.

Mathematically: Reduces to whether the rules can be embedded into a Boolean algebra (or a distributive lattice).

- The issue is distributivity of the operations.

Practical cop out: For real spatial and temporal attributes, there is always an underlying "real" model which satisfies the conditions.

- For the spatial example, use an $\mathbb{R}^2$ model of the actual geographical regions.
- This model suffices to provide "proof" of distributivity, even if it is too complex to be used in practice.

Bigranular Rules

Terminology for a rule:

Head $\quad g \sqsubseteq_{\mathfrak{S}} / = \bigsqcup^{?}_{\mathfrak{S}} S \quad$ Body

Observation: Most join rules which occur in practice are *bigranular*.

Bigranular rule: of type $\langle G_1, G_2 \rangle$ (with $G_1 \neq G_2$).

- The body $S \subseteq \text{Granules}\langle \mathfrak{S} | G_1 \rangle$ and the head $g \in \text{Granules}\langle \mathfrak{S} | G_2 \rangle$.

Examples:
- Every province is the disjoint join of counties.
- Every park is contained in a minimal set of provinces.

Always disjoint join: $\bigsqcup^{?}_{\mathfrak{S}} = \bigsqcup^{\perp}_{\mathfrak{S}}$ for a bigranular rule.

Preliminary observation: In the case of bigranularity

$$S = \{g_1 \in \text{Granules}\langle \mathfrak{S} | G_1 \rangle \mid \textstyle\bigsqcap_{\mathfrak{S}} \{g_1, g\} \neq \perp_{\mathfrak{S}}\}.$$

- In other words, the body can be determined from the head if complete information about nondisjointness is known.
  - It is not necessary to store the entire rule explicitly.
  - Store only the head and the rule existence.

☡ This is almost true, but must be formulated more carefully.

Resolvability

Context: $\langle \mathsf{Constr}\langle \mathfrak{S} \rangle, \mathsf{cwa}\langle \mathfrak{S} \rangle \rangle$.

Notation: $\mathsf{AllConstr}\langle \mathfrak{S} \rangle = \mathsf{Constr}\langle \mathfrak{S} \rangle \cup \{\neg\varphi \mid \varphi \in \mathsf{cwa}\langle \mathfrak{S} \rangle\}$.

Resolvability: Say that $\varphi \in \mathsf{BaRules}\langle \mathfrak{S} \rangle$ is *resolvable* from $\mathsf{AllConstr}\langle \mathfrak{S} \rangle$ if the truth value of φ can be determined from $\mathsf{AllConstr}\langle \mathfrak{S} \rangle$.

- Either $\mathsf{AllConstr}\langle \mathfrak{S} \rangle \models_{\mathfrak{S}} \varphi$ or else $\mathsf{AllConstr}\langle \mathfrak{S} \rangle \models_{\mathfrak{S}} \neg\varphi$ must hold.

Disjointness resolvability: A pair $\langle G_1, G_2 \rangle$ of granularities is *disjointness resolvable* if $(\bigsqcap_{\mathfrak{S}} \{g_1, g_2\} = \bot_{\mathfrak{S}})$ is resolvable for every

$$\langle g_1, g_2 \rangle \in \mathsf{Granules}\langle \mathfrak{S} | G_1 \rangle \times \mathsf{Granules}\langle \mathfrak{S} | G_2 \rangle.$$

- In other words, it is always known whether two granules are disjoint.
- There is no incomplete information on disjointness.

Theorem: For a $\langle G_1, G_2 \rangle$-bigranular rule of the form

$$(g \sqsubseteq_{\mathfrak{S}} \textstyle\bigsqcup^{\perp}_{\mathfrak{S}} S) \quad \text{or} \quad (g = \textstyle\bigsqcup^{\perp}_{\mathfrak{S}} S),$$

the set S is uniquely determined by g via

$$S = \{g_1 \in \mathsf{Granules}\langle \mathfrak{S} | G_a \rangle \mid \mathsf{AllConstr}\langle \mathfrak{S} \rangle \models_{\mathfrak{S}} \textstyle\bigsqcap_{\mathfrak{S}} \{g_1, g\} \neq \bot_{\mathfrak{S}}\},$$

provided that $\langle G_1, G_2 \rangle$ is disjointness resolvable. $\square$

Order Properties of Granularity Pairs

Usual granularity order $G_1 \leq^{\Phi}_{\mathfrak{S}} G_2$: Every $g_1 \in \text{Granules}\langle\mathfrak{S}\rangle G_1$ is subsumed by some $g_2 \in \text{Granules}\langle\mathfrak{S}|G_2\rangle$.

Shown as $\longrightarrow\!\!<$.

Combined order $G_1 \mathrel{\underline{\lessdot}}^{\Phi}_{\mathfrak{S}} G_2$:

G_1 is a refined partitioning of G_2.

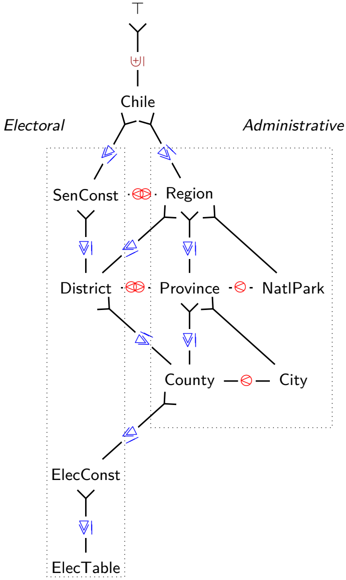
- $g_2 = \bigsqcup^{\perp}_{\mathfrak{S}} S$ plus
- Every $g_1 \in \text{Granules}\langle\mathfrak{S}|G_1\rangle$ used in some join.

Strong inequality join order $G_1 \mathrel{\circledless}^{\Phi}_{\mathfrak{S}} G_2$:

Every $g_2 \in \text{Granules}\langle\mathfrak{S}|G_2\rangle$ is covered by granules of G_1: $g_2 \sqsubseteq_{\mathfrak{S}} \bigsqcup_{\mathfrak{S}} S$.

Subgranularity order $G_1 \mathrel{\underline{\uplus}}^{\Phi}_{\mathfrak{S}} G_2$:

$\text{Granules}\langle\mathfrak{S}|G_1\rangle \subseteq \text{Granules}\langle\mathfrak{S}|G_2\rangle$.



Implementation Strategy

Development philosophy: Base the design on good theory.

- Develop theory first.

Underlying system: PostgreSQL open-source DBMS.

Stage 1: Initial support for the following:

Lookup of properties of granules: Granularity membership, subsumption, (non)disjointness.

Support for join rules: Focus on those defined implicitly by granularity order relationships, particularly combined order.

Dataset: Use publicly available data on the Chilean electoral system.

Added relations: All support implemented by adding relations; no augmentation of the DBMS itself.

Stage 2: Develop the following.

Query language supporting multigranular aggregation:

Preprocessor for the augmented query language:

Possibly a true PostgreSQL addon:

Other Completed Features Not Discussed

Spatio-temporal attributes

Place	Time	Births
Arauco_prv	*Y2016Q1*	b_1
BíoBío_prv	*Y2016Q1*	b_2
Concepción_prv	*Y2016Q1*	b_3
Ñuble_prv	*Y2016Q1*	b_4
BíoBío_rgn	*Y2016Q1*	b_5

Thematic attributes

Thematic attributes: Such attributes are also multigranular, with the granularities corresponding to levels of precision.

- Aggregation: A thematic attribute includes aggregation operators.
- Tolerance: Expresses how much distinct aggregations (at different granularities of the spatio-temporal data) may differ.

$$\textit{BíoBío\_rgn} = \bigsqcup^{\perp} \{\textit{Arauco\_prv}, \textit{BíoBío\_prv}, \textit{Concepción\_prv}, \textit{Ñuble\_prv}\}$$

Consequence: $\sum_{i=1}^{4} b_i = b_5$ (within tolerance).

TMCDs: Used to express the above type of constraints, which arise when integrating multigranular data.

The End

Tack för er uppmärksamhet!

Thank you for your attention.

Frågor?

Questions?