Integration Integrity for Multigranular Data

Stephen J. Hegner[1] and M. Andrea Rodríguez[2]

[1] Umeå University, Department of Computing Science
SE-901 87 Umeå, Sweden
hegner@cs.umu.se
[2] Departamento Ingeniería Informática y Ciencias de la Computación
Edmundo Larenas 219, Universidad de Concepción
4070409 Concepción, Chile
andrea@udec.cl

Abstract. When data from several source schemata are to be integrated, it is essential that the resulting data in the global schema be consistent. This problem has been studied extensively for the monogranular case, in which all domains are flat. However, data involving spatial and/or temporal attributes are often represented at different levels of granularity in different source schemata. In this work, the beginnings of a framework for addressing data integration in multigranular contexts are developed. The contribution is twofold. First, a model of multigranular attributes which is based upon partial orders which are augmented with partial lattice-like operations is developed. These operations are specifically designed to model the kind of dependencies which occur in multigranular modelling, particularly in the presence of aggregation operations. Second, the notion of a thematic multigranular comparison dependency, generalizing ordinary functional and order dependencies but specifically designed to model the kinds of functional and order dependencies which arise in the multigranular context, is developed.

1 Introduction

Data integration is the process of combining several databases, called the *data sources*, each with its own schema and method of representation, into a single schema for unified access. There are many theoretical issues which must be addressed in order to achieve effective integration. For a survey of these, see for example [19]. One of the most fundamental issues which must be addressed is integrity — to the extent that the information in the source databases overlaps, it must do so in a consistent fashion. Put another way, it must not be possible to derive a contradiction when the databases are combined.

Virtually all existing work on data integration, and in particular on ensuring integrity, has been conducted within the monogranular context, in which the domain of each attribute is a simple set of values. In that setting, the problem of integration integrity becomes one of ensuring that contradictions cannot arise within a unified logical theory upon combining the various data sources [20], [7]. If such contradictions are detected, they may be resolved via so-called *data*

cleaning [23]; in more formal work the idea of restoration of consistency is often called *repair* [3], [1].

In the multigranular context, the notion of contradiction becomes considerably more complex. Consider a multigranular attribute A_{Plc} which represents geographic locations, endowed with a natural poset structure defined by spatial and temporal inclusion. For example, one may write $\mathit{Region\_VIII} \sqsubseteq_{A_{\mathsf{Plc}}} \mathit{Chile}$ to represent that Region VIII lies (entirely) within Chile. Such an attribute has additional structure, however. It is also possible to assert that Chile is composed of exactly fifteen nonoverlapping regions via a join-like rule of the following form.[3]

$$\bigsqcup\!\!\!\!\bigsqcup^{\perp}_{A_{\mathsf{Plc}}} \{\mathit{Region\_R} \mid \mathit{I} \leq R \leq \mathit{XV}\} = \mathit{Chile} \qquad \text{(r-Chile)}$$

The symbol $\bigsqcup\!\!\!\!\bigsqcup^{\perp}_{A_{\mathsf{Plc}}}$ means that its arguments join disjointly; that any pair $\{\mathit{Region\_i}, \mathit{Region\_j}\}$ with $i \neq j$ is disjoint; i.e., nonoverlapping spatially. For the most part, previous work on multigranular attributes has only modelled subsumption (order) structure [8]. A main contribution of this paper is to provide a model of data granules which supports rules such as (r-Chile) economically, as well as a means to use them in the expression of constraints for data integration.

To illustrate the particular issues which arise in the multigranular framework, consider integrating the two databases shown in Fig. 1. In each case, the schema

Source database 1		
A_{Plc}	A_{Tim}	B_{Bth}
Region_I	*Q1Y2014*	n_1
Region_II	*Q1Y2014*	n_2
...	...	...
Region_XV	*Q1Y2014*	n_{15}

Source database 2		
A_{Plc}	A_{Tim}	B_{Bth}
Chile	*Q1Y2014*	b_1
Chile	*Q2Y2014*	b_2
Chile	*Q3Y2014*	b_3
Chile	*Q4Y2014*	b_4

Fig. 1. Two multigranular source databases

consists of the single relation scheme $R_{\mathsf{sumb}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Bth}}\rangle$. A tuple of the form $\langle p, s, n\rangle$ represents that in region p, during time interval s, the total number of births was n. The attribute A_{Plc} is as described above, A_{Tim} is similar but represents time intervals, and B_{Bth} has numerical values representing birth totals.

From a monogranular perspective, it is clear that the functional dependency (FD) $A_{\mathsf{Plc}}A_{\mathsf{Tim}} \rightarrow B_{\mathsf{Bth}}$ is the fundamental constraint with respect to these semantics. If different sources provide data for different places and times, all that need be checked is that the FD holds on a relation which combines the sources. However, information overlap which may occur in the multigranular context requires more complex constraints. In the above example, the semantics require that the sum of the number of births over the regions for *Q1Y2014* agree with the value for all of Chile; that is, $b_1 = \sum_{i=1}^{15} n_i$. A further contribution of this

[3] Actually, there is no Region XIII; it is called Region RM; this detail is ignored here.

paper is to show how the model of granularity which is developed may be used as a foundation for expressing such constraints.

As suggested by the example above, all relations to be integrated are assumed to have the same structure; only the granularities may differ. This simplification is made in order to focus upon the main problem — to deal with multigranularity — without complicating the investigation with questions about how the sources are to be integrated, for example, as local-as-view versus global-as-view [19].

The topic of granularity in the representation of data has received considerable attention during the past twenty years. The modelling of time with a focus upon granularity has been studied exhaustively [4], and was later adapted for use in the context of spatial databases [2]. Integrity constraints concerning multigranular data, however, have received less attention. Related work in the spatial domain includes studies concerning models for checking topological consistency at multiple representations, as well as for data integration [11], [26], [12], [18], with a focus upon modelling the consistency of different representations of the same geometric object. However, these works treat spatial constraints in isolation, without considering the interaction with thematic attributes in a database model. In the context of data warehousing, multigranular approaches have also been employed [17], but largely to save space via aggregation; the issue of integrating data at different granularities does not arise. Recently, functional dependencies and conditional functional dependencies (CFDs) have been extended to the multigranular framework [6]. Another recent work addresses repairs of inconsistent data in the spatial framework [24], but the kinds of constraints considered are not those which characterize differences between data sources which are locally consistent. In [27], *rollup dependencies*, which assert that certain thematic values (such as tax rate) are invariant under rollup, are studied. However, they do not address thematic values which vary with granularity, or which involve aggregation. That which is new to the ideas developed in this paper, which distinguishes it from that cited above, is the formulation and study of constraints which arise specifically when different sources provide the same or similar data, but at different levels of granularity. In particular, the emphasis is upon situations in which the tie between the representations at differing granularities is one of aggregation over attributes representing space or time.

The remainder of the paper consists of two main sections. In Section 2, the ideas of multigranular attributes, with particular emphasis upon how to express the kind of join and disjointness conditions which arise when rules such as (r-Chile) require. In Section 3, the associated integration dependencies are developed in detail, and a sketch of the data structures necessary to implement them efficiently is also given. Section 4 provides conclusions and further directions.

2 Relational Concepts in the Multigranular Setting

In this section, the fundamental notions which underlie a relational database schema are extended to the multigranular framework. As such, this material forms the underpinnings for constraint formulation which is developed in Sec. 3.

It is assumed that the reader is familiar with basic relational database theory, as presented in [21]. However, even an introduction textbook, such as [13], should provide further background for many of the ideas used here.

Notation 2.1 (Some mathematical notation). For any set S, $\mathsf{Card}(S)$ denotes its cardinality. $\mathbf{2}^S$ denotes the set of all subsets of S. $f(x)\downarrow$ denotes that the partial function f is defined on argument x. $S_1 \subseteq_f S_2$ indicates that S_1 is a finite subset of S_2 (while $S_1 \subseteq S_2$ denotes that S_1 is any subset of S_2, finite or otherwise).

$\mathbb{Z}$ denotes the set of integers, $\mathbb{N}$ denotes the set of nonnegative integers, while $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Intervals are always of integers; $[i, j] = \{n \in \mathbb{Z} \mid i \leq n \leq j\}$.

Definition 2.2 (Posets). For elaboration of the ideas surrounding partially ordered sets (posets), see [9] for basic ideas and [15] for more advanced notions. Only essential notation is reviewed here. A *poset* is a pair $\boldsymbol{P} = (P, \leq_{\boldsymbol{P}})$ in which P is a set and $\leq_{\boldsymbol{P}}$ is a partial order on P. $\mathbf{P}$ is *upper bounded* if it has a greatest element $\top_{\boldsymbol{P}}$. If it also has a least element $\bot_{\boldsymbol{P}}$, then it is *bounded*. The bounds may be indicated explicitly in the notation; i.e., $\boldsymbol{P} = (P, \leq_{\boldsymbol{P}}, \top_{\boldsymbol{P}})$, $\boldsymbol{P} = (P, \leq_{\boldsymbol{P}}, \bot_{\boldsymbol{P}}, \top_{\boldsymbol{P}})$. It will always be assumed that in a bounded poset, $\top_{\boldsymbol{P}}$ and $\bot_{\boldsymbol{P}}$ are distinct elements.

For $S \subseteq P$, $\mathsf{GLB}_{\mathbf{P}}\langle S\rangle$ denotes the greatest lower bound of S (when it exists).

In [6], the definitions of granularity and granule are intertwined in a single definition, that of a *domain schema*. In this paper, following the classical approach for monogranular schemata [21, Sec. 1.2], the notion of an attribute (and thus granularity) is defined first, with the associated notion of a domain assignment (and thus granule assignment) for that attribute defined afterwards.

Concept 2.3 (Granulated attributes). In the classical relational model, the columns are labelled with *attributes*, with each attribute A assigned a set of *domain elements* from which the values for A are taken. In the granulated approach, each attribute consists of a partially ordered set of *granularities*. The domain elements, called *granules*, also have a natural order structure which is tied to the granularities. Formally, a *granulated attribute* A is defined by its *granularity poset* $\mathbf{Gran}\langle A\rangle = (\mathsf{Gran}\langle A\rangle, \leq_{\mathsf{Gran}\langle \boldsymbol{A}\rangle}, \top_{\mathsf{Gran}\langle \boldsymbol{A}\rangle})$, a finite upper-bounded poset. The elements in $\mathsf{Gran}\langle A\rangle$ are called the *granularity identifiers* of A; or, less formally, just the *granularities* of A. When the context of the operators is clear, the subscripts may be dropped: $\mathbf{Gran}\langle A\rangle = (\mathsf{Gran}\langle A\rangle, \leq, \top)$.

The scheme $R_{\mathsf{sumb}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Bth}}\rangle$ of Sec. 1 provides a context for examples. First of all, each of the three attributes has a coarsest granularity, which recaptures no information about the domain value: $\top_{\mathsf{Gran}\langle \boldsymbol{A}_{\mathsf{Plc}}\rangle}$ corresponds to all of Chile, $\top_{\mathsf{Gran}\langle \boldsymbol{A}_{\mathsf{Tim}}\rangle}$ lumps all time values into one, and $\top_{\mathsf{Gran}\langle \boldsymbol{B}_{\mathsf{Bth}}\rangle}$ lumps all numbers into one. The spatial attribute A_{Plc} might have, in addition to $\top_{\mathsf{Gran}\langle \boldsymbol{A}_{\mathsf{Plc}}\rangle}$, Region, City, and $\mathsf{NatRegion}$ (identifying natural, as opposed to political, regions) as granularities, with $\mathsf{City} \leq \mathsf{Region} \leq \top_{\mathsf{Gran}\langle \boldsymbol{A}_{\mathsf{Plc}}\rangle}$ and $\mathsf{NatRegion} \leq \top_{\mathsf{Gran}\langle \boldsymbol{A}_{\mathsf{Plc}}\rangle}$. It has no least granularity, since a natural region of Chile may lie in two more more political regions. The temporal attribute A_{Tim} might have, in addition to $\top_{\mathsf{Gran}\langle \boldsymbol{A}_{\mathsf{Tim}}\rangle}$,

$\mathsf{QuarterYr}$, $\mathsf{MonthYr}$, and WeekYr as granularities, with $\mathsf{MonthYr} \leq \mathsf{QuarterYr}$ and $\mathsf{WeekYr} \leq \top_{\mathbf{Gran}\langle A_{\mathsf{Tim}}\rangle}$. Here $\mathsf{QuarterYr}$ represents a quarter of a given year; similarly for $\mathsf{MonthYr}$ and WeekYr. $\top_{\mathbf{Gran}\langle A_{\mathsf{Tim}}\rangle}$ lumps together all of time. Note that neither $\mathsf{WeekYr} \leq \mathsf{MonthYr}$ nor $\mathsf{WeekYr} \leq \mathsf{QuarterYr}$ holds, since a single week may span two months or two quarters. It has no least granularity since the overlap of a week and a month need not correspond to any granularity. Finally, for the attribute B_{Bth}, fix $\mathsf{maxr} \in \mathbb{N}^+$. For $i \in [1, \mathsf{maxr}]$, the granularity round_i identifies rounding to i significant digits. In addition, the granularity round_∞ represents no rounding at all, and is thus the least element of $\mathbf{Gran}\langle B_{\mathsf{Bth}}\rangle$; i.e., $\mathsf{round}_\infty = \bot_{\mathbf{Gran}\langle B_{\mathsf{Bth}}\rangle}$. Thus $\bot_{\mathbf{Gran}\langle B_{\mathsf{Bth}}\rangle} = \mathsf{round}_\infty \leq \mathsf{round}_i \leq \mathsf{round}_j \leq \top_{\mathbf{Gran}\langle B_{\mathsf{Bth}}\rangle}$ for $j < i$. To elaborate these examples, it is necessary to have a representation for granules as well. This issue is substantially more complex, and is examined next.

Discussion 2.4 (Modelling the space of granules). Previous work on multigranular attributes, including [6], have focused entirely upon the poset structure of the granules, without means for the representation of join-like operations, such as that expressed in formula (r-Chile). In considering possible formulations, it is important to keep in mind that the least upper bound (LUB) is not always the desired join. It would be incorrect to express a constraint, similar in form to (r-Chile), which expressed that Chile is composed of its cities, since much of the country does not lie within the borders of any city, even though Chile be the LUB of its cities in the poset of granules. To avoid such problems, one option might be to assume that the space of granules forms a lattice, or at least a semilattice. However, this would result in an enormous number of granules, including many which would be of no use, since any combination of granules would itself be a granule. The approach taken here is to enhance the poset structure of the granules with partial operations which only identify combinations that are also known granules.

Concept 2.5. Subset-based bounded posets One tempting approach to adding constraints to the poset of granules is to allow partial join and meet rules. For binary join and meet operations, the notion of a *weak partial lattice* [15, pp. 52-56] does exactly this. These ideas have been extended to operations of arbitrary finite arity via the notion of a *generalized bounded weak partial lattice* [16]. Unfortunately, as developed in some detail in [16], it is an NP-hard problem to determine whether the added rules will force two elements to coalesce.

The solution forwarded here is to assume additional structure, which is always satisfied in typical applications involving multigranular spatial and temporal attributes. Specifically, a *subset base* for a bounded poset $\boldsymbol{P} = (P, \sqsubseteq_{\boldsymbol{P}}, \bot_{\boldsymbol{P}}, \top_{\boldsymbol{P}})$ is a pair $\langle \mathcal{B}, \iota \rangle$ in which $\mathcal{B}$ is a set, called the *base set*, and $\iota : P \to \mathcal{B}$ is an injective function, called the *concretization function*, for which $\iota(\top_{\boldsymbol{P}}) = \mathcal{B}$, $\iota(\bot_{\boldsymbol{P}}) = \emptyset$, and $(\forall p_1, p_2 \in P)((p_1 \leq_P p_2) \Leftrightarrow (\iota(p_1) \subseteq \iota(p_2)))$. A *subset-based bounded poset* (or *SBBP* for short) is a pair $\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle \rangle$ in which $\boldsymbol{P}$ is a bounded poset and $\langle \mathcal{B}, \iota \rangle$ is a subset base for $\boldsymbol{P}$. An SBBP is *finite* if P is a finite set; $\mathcal{B}$ need not be finite.

To illustrate, consider a spatial attribute such as A_{Plc}. The set $\mathcal{B}_{A_{\mathsf{Plc}}}$ might be the coordinates in a two-dimensional plane, or those of the surface of the a sphere (representing the earth). The concretization function $\iota_{A_{\mathsf{Plc}}}$ would map each geographic unit (city, region, country, park, *etc.*) to the set of points which represent it. Note that the points involved need not even be countable, much less finite. It is only the set of actual granules which need be finite. A similar model, using point in time, applies to the temporal attribute A_{Tim}.

It must be emphasized that the subset base and concretization function are in the background; it is not necessary to represent them explicitly, and in many cases it will not be practical to represent them explicitly. Rather, it is only necessary to know that they exist. This existence comes automatically with spatial and temporal attributes. Mathematically, they guarantee that the poset may be modelled as a *ring of sets*, which ensures distributivity of any associated lattice operations [15, Ch. 2, Thm. 19], such as those defined in Concept 2.6.

Concept 2.6 (Rules for SBBPs). In the context of an SBBP, it is very easy to add rules of the form required to express the kind of constraints needed on granules. Let $\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle$ be an SBBP. A *join rule* over $\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle$ is of the form $\bigsqcup_P S = a$ with $S \subseteq P$ and $a \in P$; a *disjointness rule* over $\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle$ is of the form $\bigsqcap_P \{p_1, p_2\} = \bot_P$ with $p_1, p_2 \in P$; a *disjoint join rule* over $\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle$ is of the form $\bigsqcup\!\!\!\!\bigsqcup_P S = a$ with $S \subseteq P$ and $a \in P$. The semantics of these rules are easily specified. If φ is a rule, use $\models_{\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle} \varphi$ to express that the rule is satisfied in $\boldsymbol{P}$. Then $\models_{\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle} \bigsqcup_P S = a$ iff $\bigcup\{\iota(s) \mid s \in S\} = \iota(a)$; $\models_{\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle} \bigsqcap_P \{p_1, p_2\} = \bot_P$ iff $\iota(p_1) \cap \iota(p_2) = \emptyset$; $\models_{\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle} \bigsqcup\!\!\!\!\bigsqcup_P S = a$ iff $\models_{\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle} \bigsqcup_P S = a$ and for every $p_1, p_2 \in S$ with $p_1 \neq p_2$, $\models_{\langle \boldsymbol{P}, \langle \mathcal{B}, \iota \rangle\rangle} \bigsqcap_P \{p_1, p_2\} = \bot_P$. It is clear that these semantics are the correct ones for spatial and temporal attributes. It must be emphasized once again that $\langle \mathcal{B}, \iota \rangle$ is in the background. For example, to know that Chile is the disjoint union of its fifteen regions, as expressed in (r-Chile), it is not necessary to know the precise geographic coordinates of the regions. It is only necessary to know that their union covers all of Chile, without overlap.

Other rules, such a general meet rules, could be defined easily, but the above selection has been chosen to support that which is needed to express common constraints on granules.

The main notion of a granulated domain assignment, which, in contrast to the formulation of [6], admits join rules as well as simple order statements, may now be given.

Concept 2.7 (Granulated domain assignments). Let A be a granulated attribute. A *(granulated) domain assignment* for A is a four-tuple $\mathsf{GDA}_A = (\mathbf{Dom}_A, \langle \mathcal{B}_A, \iota_A \rangle, \mathsf{Rules}_A, \mathsf{GrtoDom}_A)$ in which $\mathbf{Dom}_A = (\mathsf{Dom}_A, \sqsubseteq_A, \bot_A, \top_A)$ is a finite bounded poset, called the *granulated domain* of A, $\langle \mathcal{B}_A, \iota_A \rangle$ is a subset base for $\mathbf{Dom}_A$ (so that $\langle \mathbf{Dom}_A, \langle \mathcal{B}_A, \iota_A \rangle\rangle$ forms an SBBP), Rules_A is a set of rules over $\langle \mathbf{Dom}_A, \langle \mathcal{B}_A, \iota_A \rangle\rangle$ (see Concept 2.6), and $\mathsf{GrtoDom}_A : \mathsf{Gran}\langle A \rangle \to \mathbf{2}^{\mathsf{Dom}_A}$ is a function which is subject to the following conditions.

(gda-i) $\mathsf{GrtoDom}_A(\top_{\mathsf{Gran}\langle \boldsymbol{A} \rangle}) = \{\top_A\}$.

(gda-ii) $(\forall g \in \mathsf{Dom}_A \setminus \{\bot_A\})(\exists G \in \mathsf{Gran}\langle A \rangle)(g \in \mathsf{GrtoDom}_A(G))$.

(gda-iii) $\mathsf{GrtoDom}_A(\bot_A) = \emptyset$.

(gda-iv) $(\forall G \in \mathsf{Gran}\langle A\rangle)(\forall g_1, g_2 \in \mathsf{GrtoDom}_A(G))$
$(g_1 \neq g_2 \Rightarrow (\bigsqcap_A\{g_1, g_2\} = \bot_A \in \mathsf{Rules}_A))$.

(gda-v) $(\forall G_1, G_2 \in \mathsf{Gran}\langle A\rangle)((G_1 \leq_{\mathsf{Gran}\langle \boldsymbol{A}\rangle} G_2) \Leftrightarrow$
$(\forall g_1 \in \mathsf{GrtoDom}_A(G_1))(\exists g_2 \in \mathsf{GrtoDom}_A(G_2))(g_1 \sqsubseteq_A g_2))$.

(gda-vi) For each $\varphi \in \mathsf{Rules}_A$, $\models_{\langle \boldsymbol{A}, \langle \mathcal{B}, \iota\rangle\rangle} \varphi$.

The elements of Dom_A are called the *granules* of GDA_A. If $g \in \mathsf{GrtoDom}_A(G)$, then g is said to be *of granularity* G or *to have granularity* G. If $g_1 \sqsubseteq_A g_2$, then g_2 is said to be *coarser* than g_1, and g_1 is said to be *finer* than g_2. It is also said that g_2 *subsumes* g_1 and that g_1 is *subsumed by* g_2. As illustrated in (gda-iv) and (gda-vi), to avoid long subscripts, $\models_{\langle \mathbf{Dom}_A, \langle \mathcal{B}, \iota\rangle\rangle}$ is shortened to just $\models_{\langle \boldsymbol{A}, \langle \mathcal{B}, \iota\rangle\rangle}$, and the subscripts in rules are also shortened from $\mathbf{Dom}_A$ to just A; thus $\bigsqcup_{\mathbf{Dom}_A} S = a$ is written as just $\bigsqcup_A S = a$, for example. Condition (gda-iv) asserts a fundamental property of granularities — that distinct granules of the same granularity are disjoint, in the sense that their meet in the SBBP of granules is $\bot_A$. In spatial and temporal modelling, this means that they do not overlap. Condition (gda-v) relates the order of granularities to the order of granules — $G_1 \leq_{\mathsf{Gran}\langle \boldsymbol{A}\rangle} G_2$ just in the case that for every granule g_1 of G_1, there is a coarser granule g_2 of G_2. Finally, (gda-vi) requires that each rule in Rules_A be satisfied in $\mathbf{Dom}_A$.

For the three attributes of $R_{\mathsf{sumb}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Bth}}\rangle$, granulated domain assignments are completely straightforward. For A_{Plc}, the granules are geographic regions, classified according to the granularities identified in Concept 2.3. For example, *Santiago* and *Concepción* are granules of granularity City, while *Region_VIII* is a granule of granularity Region.

Similarly,A_{Tim} is assigned granules identifying time intervals. The granules of B_{Bth} are just natural numbers, rounded as described in Concept 2.3. The constraint of formula (r-Chile) in Sec. 1 may be represented easily in $\mathsf{GDA}_{A_{\mathsf{Plc}}}$ via the single rule $\bigsqcup_{R \in [I,XV]}^{A_{\mathsf{Plc}}} \mathit{Region\_R} = \mathit{Chile}$. Similarly, the constraint that Concepción lies in Region VIII may be expressed using $\mathit{Concepción} \sqsubseteq_{A_{\mathsf{Plc}}} \mathit{Region\_VIII}$, which is not a rule but just an order statement in the poset $\mathbf{Dom}_{A_{\mathsf{Plc}}}$.

The same granule may belong to more than one granularity. For example, it is not inconceivable that a single granule could have granularity both City and Region. This would happen were a city to constitute a region by itself.

An ordinary monogranular attribute A is recaptured by a granularity which contains only $\top_{\mathsf{Gran}\langle \boldsymbol{A}\rangle}$ and the granularity $\bot_{\mathsf{Gran}\langle \boldsymbol{A}\rangle}$ with $\mathsf{GrtoDom}_A(\bot_{\mathsf{Gran}\langle \boldsymbol{A}\rangle}) = \mathsf{FlatDom}\langle A\rangle$, the set of all values which are allowed for attribute A in tuples. $\top_{\mathsf{Gran}\langle \boldsymbol{A}\rangle}$ is something of an artifact. It contains a single granule which is coarser than each element of $\mathsf{FlatDom}\langle A\rangle$. In view of (gda-i), such a granule is required.

Notation 2.8 (Convention). For the rest of this section, unless stated explicitly to the contrary, take A to be a granulated attribute and $\mathsf{GDA}_A = (\mathbf{Dom}_A, \langle \mathcal{B}_A, \iota_A\rangle, \mathsf{Rules}_A, \mathsf{GrtoDom}_A)$ to be a granulated domain assignment for A with $\mathbf{Dom}_A = (\mathsf{Dom}_A, \sqsubseteq_A, \bot_A, \top_A)$.

Observation 2.9 (Uniqueness of subsuming granules). *Given* $g_1, g_2, g_2' \in \mathbf{Dom}_A$ *with* $g_1 \sqsubseteq_A g_2$, $g_1 \sqsubseteq_A g_2'$, *and* $G_2 \in \mathsf{Gran}\langle A\rangle$ *with* $g_2, g_2' \in \mathsf{GrtoDom}_A(G_2)$, *it must be the case that* $g_2 = g_2'$.

Proof. Let g_1, g_2, g_2' and G_2 be as stated. By (gda-iv), $\bigsqcap_A\{g_2, g_2'\} = \bot_A$. However, $g_1 \sqsubseteq_A \bigsqcap_A\{g_2, g_2'\}$, whence it must be the case that $g_2 = g_2'$. □

Concept 2.10 (Coarsening). In order to support the management of source data at differing granularities, it is often necessary to reduce them to a common granularity. The operation of coarsening, which transforms a granule to a one at a coarser granularity, is central to this idea. Formally, the function $\mathsf{Coarsen}_A : \mathsf{Dom}_A \times \mathsf{Gran}\langle A\rangle \to \mathsf{Dom}_A$ is defined on $\langle g_1, G_2\rangle$ iff there is a $g_2 \in \mathsf{GrtoDom}_A(G_2)$ with $g_1 \sqsubseteq_A g_2$. In view of Observation 2.9, this g_2 is unique whenever it exists. In this case $g_2 = \mathsf{Coarsen}_A\langle g_1, G_2\rangle$, and is called the *coarsening* of g_1 to G_2. This operation corresponds to MAP(g_1, G_2) of [6].

In the spatial context of A_{Plc}, the city of Concepción lies in Region VIII of Chile. This would be represented by the coarsening $\mathsf{Coarsen}_{A_{\mathsf{Plc}}}\langle \mathit{Concepción}, \mathsf{Region}\rangle = \mathit{Region\_VIII}$. Similarly, in the temporal context of A_{Tim}, quarter 1 of year 2014 lies with 2014; this would be represented by the coarsening $\mathsf{Coarsen}_{A_{\mathsf{Tim}}}\langle \mathit{Q1Y2014}, \mathsf{Year}\rangle = \mathit{2014}$.

Concept 2.11 (Thematic attributes and orderings). Following common usage in geographic information systems [5], a *thematic attribute* is used to record values associated with aggregating (e.g., spatial or temporal) attributes. For example, in $R_{\mathsf{sumb}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Bth}}\rangle$, B_{Bth} is thematic. When such attributes have numerical domain values, there are often two distinct orders which are used in modelling integrity under integration. First of all, granularities defined by rounding, as explained in Concept 2.3, have a natural poset structure. However, there is also the natural order of numbers, independent of any granularity. This latter order is termed *thematic*. Formally, a *thematic ordering* $\theta_A = \{\leq^G_{\theta_A} \mid G \in \mathsf{Gran}\langle A\rangle\}$ on GDA_A assigns, for each granularity $G \in \mathsf{Gran}\langle A\rangle$, a partial order $\leq^G_{\theta_A}$ to $\mathsf{GrtoDom}_A(G)$, subject to the requirement that for $G_1, G_2 \in \mathsf{Gran}\langle A\rangle$ with $G_1 \leq_{\mathsf{Gran}\langle A\rangle} G_2$, and all $g_1, g_1' \in \mathsf{GrtoDom}_A(G_1)$, if $g_1 \leq^{G_1}_{\theta_A} g_1'$ then $\mathsf{Coarsen}_A\langle g_1, G_2\rangle \leq^{G_2}_{\theta_A} \mathsf{Coarsen}_A\langle g_1', G_2\rangle$. In other words, thematic order must be preserved under coarsening. For B_{Bth}, the thematic order is simple numerical order, while the granular order is based upon subsumption of intervals, as elaborated in Concept 2.3 and Concept 2.7.

Concept 2.12 (Aggregation operators on thematic orderings). Data in a multigranular context are often statistical in nature. As such, thematic values corresponding to coarser spatial or temporal regions may be aggregations of those for finer ones. Therefore, a general formulation of an aggregation operator is central to any effort to model data integration in such a context. Formally, let $\theta_A = \{\leq^G_{\theta_A} \mid G \in \mathsf{Gran}\langle A\rangle\}$ be a thematic ordering on GDA_A. An *aggregation operator* on θ_A is a family

$$\oplus_A = \{\textstyle\bigoplus^G_A : \mathsf{MultisetsOf}\langle \mathsf{GrtoDom_A(G)}\rangle \to \mathsf{GrtoDom}_A(G) \mid G \in \mathsf{Gran}\langle A\rangle\}$$

of functions such that the following two properties hold for any $G \in \mathsf{Gran}\langle A\rangle$.

(ag-i) For any $g \in \mathsf{GrtoDom}_A(G)$, $\bigoplus_A^G\{g\} = g$.

(ag-ii) For any finite multisets $S_1, S_2 \subseteq \mathsf{GrtoDom}_A(G)$, if there is an injective multifunction $h : S_1 \to S_2$ such that $(\forall g \in S_1)(g \leq_{\theta_A}^G h(g))$, then $\bigoplus_A^G(S_1) \leq_{\theta_A}^G \bigoplus_A^G(S_2)$.

In the above, $\mathsf{MultisetsOf}\langle\mathsf{GrtoDom_A(G)}\rangle$ denotes the set of all multisets of $\mathsf{GrtoDom}_A(G)$. A *multiset*, also called a *bag*, is similar to a set, except that an element may have finitely many occurrences. A *multifunction* maps multisets to multisets, with distinct occurrences of each element mapped possibly to distinct elements. The idea should be clear. For aggregation operators such as summation, it is necessary to treat each summand as a distinct element, even for summands of the same value.

Summation, max, and min (using $\geq$ instead of $\leq$) all form aggregation operations on the natural thematic ordering of $\mathbb{N}$, as sketched in Concept 2.11. On $\mathbb{Z}$, max and min form aggregation operations also, but summation does not, since it does not respect the ordering condition. Operations which do not respect order, such as averaging, are not aggregation operators in the sense defined here.

Concept 2.13.]Coarsening tolerance] Coarsening and aggregation need not commute with one another. For example, if the populations of the regions which comprise a country are rounded before they are summed, the result will be different than if they are summed first, and then rounded. Furthermore, data obtained from different sources may vary slightly in thematic values, for any number of reasons. Such data should not automatically be classified as inconsistent. Rather, it is appropriate to build a certain amount of tolerance into the integration constraints. To this end, the notion of a coarsening tolerance is introduced. Formally, let $\theta_A = \{\leq_{\theta_A}^G \mid G \in \mathsf{Gran}\langle A\rangle\}$ be a thematic ordering on GDA_A. A *coarsening tolerance τ_A (for equality)* with respect to θ_A is a $\mathsf{Gran}\langle A\rangle \times \mathbb{N}$-indexed family $\{\tau_A^{\langle G,n\rangle} \subseteq \mathsf{GrtoDom}_A(G) \times \mathsf{GrtoDom}_A(G) \mid (G \in \mathsf{Gran}\langle A\rangle) \wedge (n \in \mathbb{N})\}$ of reflexive and symmetric relations for which the following three properties hold for all $n \in \mathbb{N}$.

(ct-i) $\tau_A^{\langle G,0\rangle} = \{(g,g) \mid g \in \mathsf{GrtoDom}_A(G)\}$.

(ct-ii) For $G \in \mathsf{Gran}\langle A\rangle$ and $(g_1,g_2) \in \tau_A^{\langle G,n\rangle}$, if $g_1', g_2' \in \mathsf{GrtoDom}_A(G)$ with $g_1 \leq_{\theta_A}^G g_1' \leq_{\theta_A}^G g_2' \leq_{\theta_A}^G g_2$, then $(g_1', g_2') \in \tau_A^{\langle G,n\rangle}$ as well.

(ct-iii) for $G, G' \in \mathsf{Gran}\langle A\rangle$ with $G \leq_{\mathsf{Gran}\langle A\rangle} G'$, if $(g_1,g_2) \in \tau_A^{\langle G,n\rangle}$ then $(\mathsf{Coarsen}_A\langle g_1, G'\rangle, \mathsf{Coarsen}_A\langle g_2, G'\rangle) \in \tau_A^{\langle G',n\rangle}$.

The value of n identifies the amount of deviation from equality which is allowed, with larger n permitting larger differences. If $(g_1,g_2) \in \tau_A^{\langle G,n\rangle}$, then g_1 and g_2 are within the specified limit of deviation from equality for tolerance level n. Often, n will indicate the number of elements being aggregated, but this is not absolutely necessary. By default, a coarsening tolerance specifies the amount of deviation from equality which is allowed. However, for certain constraints, a deviation from order may also be specified. More specifically, given a coarsening tolerance τ as above and a thematic ordering θ_A on GDA_A, the associated *ordering tolerance* is $\{\tau_A^{\langle G,n,\leq\rangle} \subseteq \mathsf{GrtoDom}_A(G) \times \mathsf{GrtoDom}_A(G) \mid (G \in$

$\mathsf{Gran}\langle A\rangle) \wedge (n \in \mathbb{N})\}$, given relation by relation according to

$$\tau_A^{\langle G,n,\leq\rangle} = \tau_A^{\langle G,n\rangle} \cup \{(g_1, g_2) \in \mathsf{GrtoDom}_A(G) \times \mathsf{GrtoDom}_A(G) \mid g_1 \leq_{\theta_A}^G g_2\}.$$

In other words, $\tau_A^{\langle G,n,\leq\rangle}$ is obtained from $\tau_A^{\langle G,n\rangle}$ by adding all tuples of granules of the form (g_1, g_2) with $g_1 \leq_{\theta_A}^G g_2$. To facilitate parameterized use of tolerances in formulas, $\tau_A^{\langle G,n\rangle}$ may also be represented as $\tau_A^{\langle G,n,=\rangle}$.

Consider the thematic attribute B_{Bth} of the scheme $R_{\mathsf{sumb}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Bth}}\rangle$, and the associated notions developed in the penultimate paragraph of Concept 2.3. Let the aggregation operator to be supported be summation $\sum$, with results rounded as specified by the granularity round_i. A useful tolerance $\omega_{B_{\mathsf{Bth}}}$ for the granularity round_i has summation accuracy 10^{-i} times the number n of items to be aggregated, so a suitable definition for $\omega_{B_{\mathsf{Bth}}}$ at that level would be $\omega_{B_{\mathsf{Bth}}}^{\langle \mathsf{round}_i, n\rangle} = \{(k_1, k_2) \mid |k_1 - k_2| \leq n \times 10^{-i}\}$. For $i = 0$, this matches the identity tolerance; i.e., $\omega_{B_{\mathsf{Bth}}}^{\langle \mathsf{round}_0, n\rangle} = \{(k, k) \mid k \in \mathbb{N}\}$ for all $n \in \mathbb{N}$.

Leaving the context of this example and returning to the general setting, the *identity tolerance* $\mathsf{IdTol}_A^{\langle G,n\rangle}$ is given by the set of relations which are the identity on each set of granules; specifically, for each $G \in \mathsf{Gran}\langle A\rangle$ and each $n \in \mathbb{N}$, $\mathsf{IdTol}_A^{\langle G,n\rangle} = \{(g, g) \mid g \in \mathsf{GrtoDom}_A(G)\}$. Similarly, $\mathsf{IdTol}_A^{\langle G,n,\leq\rangle} = \{(g_1, g_2) \mid (g_1, g_2 \in \mathsf{GrtoDom}_A(G)) \wedge (g_1 \sqsubseteq_A g_2)\}$.

Concept 2.14 (Thematic triples). For a thematic attribute, it will prove convenient to assemble the thematic ordering, aggregation operator, and tolerance into one notational unit. Specifically, let A be a multigranular attribute. A *thematic triple* for A is of the form $\langle \theta_A, \oplus_A, \tau_A\rangle$, with θ_A a thematic ordering on A, $\oplus$ an aggregation operator for θ_A, and τ a coarsening tolerance for θ_A. In some cases, aggregation is not used, and so the choice of aggregation operator does not matter. In that case, the thematic triple may be written as $\langle \theta_A, \text{-}, \tau_A\rangle$.

Definition 2.15 (Multigranular relation schemes). Let $\mathfrak{U}$ be a set of granulated attributes. Extending the classical definition [21, 1.2], for $k \in \mathbb{N}^+$, a (k-ary) *multigranular relation scheme* over $\mathfrak{U}$ is an expression of the form $R\langle\alpha\rangle$, where $\alpha = \langle A_1, A_2, \ldots, A_k\rangle \in \mathfrak{U}^k$. The symbol R is called the *relation name*, and the list α is called an *attribute vector*.

Given a granulated domain assignment GDA_A (see Concept 2.7) for each $A \in \mathfrak{U}$, a *data tuple* for the attribute vector $\alpha = \langle A_1, A_2, \ldots, A_k\rangle$ is a k-tuple $t \in \mathsf{Dom}_{A_1} \times \mathsf{Dom}_{A_2} \times \ldots \times \mathsf{Dom}_{A_k}$. The set of all data tuples for α is denoted $\mathsf{Tuples}\langle\alpha\rangle$. A *database* for the schema $R\langle\alpha\rangle$ is a set $M \subseteq \mathsf{Tuples}\langle\alpha\rangle$. The set of all databases for $R\langle\alpha\rangle$ is denoted $\mathsf{DB}(R\langle\alpha\rangle)$.

3 Constraints for Data Integration

In this section, the concepts developed in Sec. 2 are used to develop specifications for the most important kinds of dependencies for data integration in the multigranular context. As noted in Sec. 1 integration is over copies of the same schema, albeit with differing granularities. For further simplicity, it will be assumed that all tuples to be integrated have been placed in a single relation.

Notation 3.1 (The context). Throughout this section, take $\mathfrak{U}$ to be a finite universe of granulated attributes (Concept 2.3). In particular, assume that $\{A_1, A_2, \ldots, A_k, B\} \subseteq \mathfrak{U}$. Furthermore, for each $A \in \mathfrak{U}$, there is an associated granulated domain assignment GDA_A, (Concept 2.7).

Concept 3.2 (General notions of TMCDs). The dependencies developed in this section are called *thematic multigranular comparison dependencies*, or *TMCDs*. They resemble ordinary functional and order dependencies [14, 22, 25] in many ways, including that properties of a set of attributes determines those of another. The general notation is $A_1 A_2 \ldots A_k \xrightarrow[(\ell,r)]{\circledast} \langle B : \langle \theta, \oplus, \tau \rangle \rangle$, in which the A_i's and B are attributes and $\langle \theta, \oplus, \tau \rangle$ is a thematic triple for B. The dependencies are classified along three dimensions. First, the comparison operator, shown as $\circledast$ above, is either granular subsumption $\sqsubseteq$ or else equality. Second, the type, shown as (ℓ, r) above, indicates the nature of the expressions which are compared, and will be explained further in the individual cases below. Finally, a dependency may be *unified* or *attributewise*, with the latter indicated by underlining certain attributes on the left-hand side. Although there are many variants in principle, only two will be considered in this paper. Those of type $(1, 1)$, which involve only order conditions and no aggregation, are examined in Concept 3.4, while those of types $(\perp, 1)$ and $(\text{-}, 1)$, which involve fundamental aggregation as illustrated in the examples surrounding R_{sumb} of Sec. 1, are developed in Concept 3.5.

In contrast to the CFDs (conditional functional dependencies) of [6], the TMCDs developed here are specifically oriented towards data integration. CFDs are designed to recapture dependencies which hold only for certain granularities, with no support for aggregation or tolerance. TMCDs, on the other hand, are designed to support these latter two concepts. The overlap of CFDs and TCMDs is therefore minimal; they address complementary issues in the context of constraints for multigranular schemata.

Definition 3.3 (Two useful functions). Before presenting the definitions of specific TMCDs, it is necessary to introduce two special functions, which are defined here for a generic granular attribute A.

$\mathsf{GranSetOf}_A\langle g \rangle$ The function $\mathsf{GranSetOf}_A : \mathsf{Dom}_A \to \mathbf{2}^{\mathsf{Gran}\langle A \rangle}$ returns the set of granularities of the granule g.

$\mathsf{CoarsenSetMUB}_A$: The function $\mathsf{CoarsenSetMUB}_A : \mathbf{2}^{\mathsf{Dom}_A} \to \mathbf{2}^{\mathsf{Gran}\langle A \rangle}$ maps $S \subseteq \mathsf{Dom}_A$ to the minimal elements (under $\leq_{\mathsf{Gran}\langle S \rangle}$) in the set $\{G \in \mathsf{Gran}\langle A \rangle \mid (\forall g \in S)(\mathsf{Coarsen}_A\langle g, G \rangle)\downarrow\}$. In words, it returns the minimal granularities to which all elements of S coarsen.

Concept 3.4 (TMCDs of expression type $(1, 1)$). The template for a TMCD of type $(1, 1)$ is $A_1 A_2 \ldots A_k \xrightarrow[(1,1)]{\circledast} \langle B : \langle \theta, \text{-}, \tau \rangle \rangle$. This is the simplest type of a unified TMCD, and lies closest to ordinary functional dependencies (FDs) and order dependencies (ODs). In particular, no aggregation is involved; this is why the aggregation operator in the thematic triple is shown as a dash; its properties do not matter. Nevertheless, although basic, they are important because a violation can flag fundamental inconsistencies, such as a city having a

greater population than the region which houses it. The parameter $\circledast$ is one of $\sqsubseteq$ or equality ($=$), while the parameter $(1,1)$ indicates that the comparison operation involves only a single tuple on each side. The governing formula for the comparison operation of granular subsumption (when $\circledast$ is $\sqsubseteq_A$) is shown below.

$$(\forall t_1 \in \mathsf{Tuples}\langle\alpha\rangle)(\forall t_2 \in \mathsf{Tuples}\langle\alpha\rangle)(\forall G \in \mathsf{CoarsenSetMUB}_B\langle\{t_1.B, t_2.B\}\rangle)$$
$$(((R\langle t_1\rangle \wedge R\langle t_2\rangle) \wedge (\bigwedge_{j\in[1,k]} (t_1.A_j \sqsubseteq_{A_j} t_2.A_j))$$
$$\Rightarrow \tau_B^{\langle G,1,\leq\rangle}\langle \mathsf{Coarsen}_B\langle t_1.B, G\rangle, \mathsf{Coarsen}_B\langle t_2.B, G\rangle\rangle)$$

To obtain the formula for equality, replace $\sqsubseteq_{A_i}$ with $=$, and $\tau_B^{\langle G,1,\leq\rangle}$ with $\tau_B^{\langle G,1\rangle}$.

Coarsening is essential in the multigranular environment. Consider the concrete case of of the schema $R_{\mathsf{maxp}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Pop}}\rangle$, with $A_1 = A_{\mathsf{Plc}}$, $A_2 = A_{\mathsf{Tim}}$, and $B = B_{\mathsf{Bth}}$. Think of the context described in Sec. 2; specifically, consider τ bound to $\omega_{B_{\mathsf{Bth}}}$, as described in Concept 2.13. There might be two tuples $\langle p_1, s_1, n_1\rangle$ and $\langle p_2, s_2, n_2\rangle$ such that $p_1 \sqsubseteq_{A_{\mathsf{Plc}}} p_2$ and $s_1 \sqsubseteq_{A_{\mathsf{Tim}}} s_2$. When applied to these two tuples, with $t_1 = \langle p_1, s_1, n_1\rangle$ and $t_2 = \langle p_2, s_2, n_2\rangle$, the constraint requires that $n_1 \leq n_2$, up to coarsening to a common granularity and up to the tolerance specified by $\omega_{B_{\mathsf{Bth}}}$. Concretely, if region p_1 is contained in region p_2, and time interval s_1 is contained in time interval s_2, then the number of births in p_1 during s_1 must be no larger than the number of births in p_2 during s_2.

Concept 3.5 (TMCDs of expression type $(\perp,1)$ and $(\text{-},1)$). Together with those of type $(1,1)$ as described in Concept 3.4, these form the most important types of constraints for verifying the integrity of multigranular data from different sources. The template for this dependency, with $\ell \in \{\perp, \text{-}\}$, is

$$\underline{A}_1 \ldots \underline{A}_{i-1} A_i \underline{A}_{i+1} \ldots \underline{A}_k \xrightarrow[(\ell,1)]{\circledast} \langle B : \langle \theta, \oplus, \tau\rangle\rangle.$$

This type of constraint is *attributewise*; only one attribute on the left-hand side (LHS) is allowed to vary in value amongst the tuples to be tested. The values of those which are underlined are identical in all tuples considered. The parameters $(\perp,1)$ and $(\text{-},1)$ indicate that the comparison is between a set of attribute values and a single value, with $\perp$ indicating further that the set of values forms a disjoint join and - indicating that the join need not be disjoint. The general logical formula which covers all cases in which $\circledast$ is equality is shown below, with the symbol $\bigsqcup^{?}$ representing one of $\bigsqcup$ or $\bigsqcup$, depending upon whether the type is $(\perp,1)$ or $(\text{-},1)$. For inequality, replace $((\bigsqcup^{?}_{t_1\in T_1^{A_i}} t_1.A_i) = t_2.A_i)$ with $((\bigsqcup^{?}_{t_1\in T_1^{A_i}} t_1.A_i) \sqsubseteq_A t_2.A_i)$ and $\tau_B^{\langle G,1\rangle}$ with $\tau_B^{\langle G,1,\leq\rangle}$. Due to space limitations, only the case of $\circledast$ being equality will be discussed further, since the most important modelling situations involve that operator.

$$(\forall T_1 \subseteq_f \mathsf{Tuples}\langle\alpha\rangle)(\forall t_2 \in \mathsf{Tuples}\langle\alpha\rangle)(\forall G_1 \in \mathsf{CoarsenSetMUB}_B\langle\{t.B \mid t \in T_1\}\rangle)$$
$$(\forall G_2 \in \mathsf{GranSetOf}_B\langle t_2.B\rangle)(\forall G \in \mathsf{MUB}\langle\{G_1, G_2\}\rangle)$$
$$((\bigwedge_{t_1 \in T_1} R\langle t_1\rangle) \wedge R\langle t_2\rangle \wedge (\bigwedge_{\substack{t_1 \in T_1 \\ j \in [1,k]\setminus\{i\}}} (t_1.A_j = t_2.A_j)) \wedge ((\bigsqcup^{?}_{t_1 \in T_1^{A_i}} t_1.A_i) \;=\; t_2.A_i)$$
$$\Rightarrow \tau_B^{\langle G, \mathsf{Card}(T_1)\rangle}\langle \mathsf{Coarsen}_B\langle \bigoplus^{G_1}_{\substack{B \\ t_1 \in T_1}} \mathsf{Coarsen}_B\langle t_1.B, G_1\rangle, G\rangle, \mathsf{Coarsen}_B\langle t_2.B, G\rangle\rangle)$$

To keep things concrete, consider the cases in which the type is $(\perp, 1)$. This kind of constraint applies when an equality of the form $\bigsqcup\!\!\!\!\!\perp_{A_i} S = a$ holds in GDA_{A_i}. As a specific example, consider the scheme $R_{\mathsf{maxp}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Pop}}\rangle$ with A_i associated with A_{Plc}. Suppose further that $\oplus_B$ is bound to summation. S might be a set of disjoint regions which together are exactly the region a. More concretely, if there are tuples $\{\langle p_i, t, n_i\rangle \mid i \in [1,m]\}$ in the relation, and also a tuple $\langle p, t, n\rangle$, with $(\bigsqcup\!\!\!\!\!\perp_{i \in [1,m]}^{A_{\mathsf{Plc}}} p_i) = p$ holding in $\mathsf{GDA}_{A_{\mathsf{Plc}}}$, then the LHS of the rule is matched and the equality $\sum_{i \in [1,m]} n_i = n$ should hold, modulo coarsening and tolerance. This is exactly what the constraint specifies — that the population of a region, at a given point in time, is the sum of the populations of a set of disjoint regions which cover it completely, without overlap. The reason for coarsening the elements from T_1 first to G_1, and then to G after aggregation, is that it is always desirable to perform aggregation at the finest granularity possible. While it would be possible to aggregate everything to G from the start, this could possibly result in increased error in the aggregation. Inequality arises in this same context when only some of the regions are considered; if $(\bigsqcup\!\!\!\!\!\perp_{i \in [1,m]}^{A_{\mathsf{Plc}}} p_i) \sqsubseteq_{A_{\mathsf{Plc}}} p$, then $\sum_{i \in [1,m]} n_i \leq n$, module coarsening and tolerance.

The corresponding nondisjoint constraint, with $\bigsqcup_{A_i}$ replacing $\bigsqcup\!\!\!\!\!\perp_{A_i}$, applies when the aggregation operator does not require disjointness (e.g., max and min).

Discussion 3.6 (Discarding attributewise specification). In the case that the same thematic order and aggregation operator is used with respect to all attributes on the LHS of a TMCD, it is tempting to consider discarding the attributewise specification, and combine all into one big dependency, which might be represented as $A_1 A_2 \ldots A_k \xrightarrow[(\ell,1)]{=} \circledast\langle B : \langle\theta, \oplus, \tau\rangle\rangle$, with $\ell \in \{\perp, \text{-}\}$, with the following logical formula for type $(\perp, 1)$.

$$(\forall T_1 \subseteq_f \mathsf{Tuples}\langle\alpha\rangle)(\forall t_2 \in \mathsf{Tuples}\langle\alpha\rangle)(\forall G_1 \in \mathsf{CoarsenSetMUB}_B\langle\{t.B \mid t \in T_1\}\rangle)$$
$$(\forall G_2 \in \mathsf{GranSetOf}_B\langle t_2.B\rangle)(\forall G \in \mathsf{MUB}\langle\{G_1, G_2\}\rangle)$$
$$((\bigwedge_{t_1 \in T_1} R\langle t_1\rangle) \wedge R\langle t_2\rangle \wedge (\bigwedge_{i \in [1,k]} (\bigsqcup\!\!\!\!\!\perp_{t_1 \in T_1^{A_i}} t_1.A_i) \;=\; t_2.A_i) \wedge$$
$$\Rightarrow \tau_B^{\langle G, \mathsf{Card}(T_1)\rangle}\langle \mathsf{Coarsen}_B\langle \bigoplus^{G_1}_{\substack{B \\ t_1 \in T_1}} \mathsf{Coarsen}_B\langle t_1.B, G_1\rangle, G\rangle, \mathsf{Coarsen}_B\langle t_2.B, G\rangle\rangle)$$

From a theoretical point of view, this definition is fine. However, without suitable adaptation, it does not recapture what would normally be expected of such a dependency. To illustrate, work within the context of $R_{\mathsf{sumb}}\langle A_{\mathsf{Plc}}, A_{\mathsf{Tim}}, B_{\mathsf{Bth}}\rangle$, with the rules $\bigsqcup_{A_{\mathsf{Plc}}}\{p_1, p_2\} = \bigsqcup_{A_{\mathsf{Plc}}}\{p_3, p_4\} = p$ holding in $\mathsf{GDA}_{A_{\mathsf{Plc}}}$ and the rules $\bigsqcup_{A_2}\{s_1, s_2\} = \bigsqcup_{A_2}\{s_3, s_4\} = t$ holding in $\mathsf{GDA}_{A_{\mathsf{Tim}}}$. Now, suppose that $T_1 = \{\langle p_1, s_1, b_1\rangle, \langle p_2, s_2, b_2\rangle\}$, and $t_2 = \langle p, s, b\rangle$ in the above formula. Assume further that all values for attribute B_{Bth} are at the same granularity G, so no coarsening is necessary. Furthermore, for simplicity, assume that the tolerance τ is bound to the identity. Then the above rule mandates that $b_1 + b_2 = b$. However, this is not realistic modelling. b_1 is the number of births in region p_1 during time s_1, while b_2 is the number of birth in region p_2 during time interval s_2. To get the total number of births in region p during time interval t, it would be necessary to find and add tuples of the form $\langle p_1, s_2, b_3\rangle$ and $\langle p_2, s_1, b_4\rangle$. Then, and only then, would $b_1 + b_2 + b_3 + b_4 = b$ hold. In other words, there must be a tuple which captures every (place,time) point of an appropriate "rectangle" in order to get the correct total number of births.

Unfortunately, things can become even more complex. Suppose instead that $T_1 = \{\langle p_1, s_1, b_1\rangle, \langle p_2, s_1, b_2\rangle, \langle p_3, s_2, b_3\rangle, \langle p_4, s_2, b_4\rangle\}$ and $T_2 = \{\langle p, s, b\rangle\}$. It is easy to see that $b_1 + b_2 + b_3 + b_4 = b$ must hold here as well. In other words, different decompositions of p may be used for different corresponding values of attribute A_{Tim}. From a formal point of view, the most elegant solution is to regard $A_1A_2 \ldots A_k$ as a combined domain, and replace $(\bigwedge_{i=1}^{k}(\bigsqcup_{t_1 \in T_1}^{A_i} t_1.A_i = t_2.A_i))$ with something of the form $\bigsqcup_{t_1 \in T_1}^{A_i}(t_1.A_1A_2 \ldots A_k = t_2.A_1A_2 \ldots A_k)$. However, it seems that to implement something so complex efficiently is almost impossible. Thus, it seems that attributewise specification is a necessity.

Discussion 3.7 (The join logic for granulated domain assignments). The presentation in this paper has focused upon the representation of constraints for data integration in the multigranular environment, but not their implementation. Due to space limitations, a full discussion must be deferred to another paper. Nevertheless, there is an issue which demands at least some brief discussion. Looking particularly at the formula of Concept 3.5 for constraints of types $(\perp, 1)$ and $(\text{-}, 1)$, it cannot help but be noted that quantification for T_1 is over *sets* of tuples, not just individual tuples. It might then be concluded that such constraints cannot possibly be supported efficiently. However, it is not necessary to check all subsets of tuples. Rather, it is sufficient to consider only those whose combined values for attribute A_i match the LHS of some rule in the SBBP, closed under deduction. This may be managed effectively using a propositional Horn logic. Specifically, let A be an attribute, and let X be a set of rules of the form $g_1 \sqsubseteq_A g_2$, with $g_1, g_2 \in \mathsf{Dom}_A$, and of the form $\bigsqcup_S = g$, with $S \subseteq_f \mathsf{Dom}_A$ and $g \in \mathsf{Dom}_A$. The *join logic* of X, denoted $\mathsf{JLogic}\langle X\rangle$, is the propositional Horn logic whose propositions are just the elements of Dom_A, with $\perp_A$ representing the statement which is always true and $\top_A$ representing the statement which is always false. The clauses $\mathsf{Clauses}\langle\mathsf{JLogic}\langle X\rangle\rangle$ of $\mathsf{JLogic}\langle X\rangle$ are given as follows. First, if

$(g_1 \sqsubseteq_A g_2) \in X$, then $(g_2 \Rightarrow g_1) \in \mathsf{Clauses}\langle\mathsf{JLogic}\langle X\rangle\rangle$. Second, if $(\bigsqcup_S = g) \in X$, then $(\bigwedge S \Rightarrow g) \in \mathsf{Clauses}\langle\mathsf{JLogic}\langle X\rangle\rangle$ and $(g \Rightarrow s) \in \mathsf{Clauses}\langle\mathsf{JLogic}\langle X\rangle\rangle$ for all $s \in S$ as well. The utility of this representation is that inference in propositional Horn logic has complexity $\Theta(n)$ or $\Theta(n{\cdot}\log(n))$, depending upon how proposition names are accessed [10]. Thus, inference which operates on joins and order only, and not meets, may be performed very efficiently. Disjointness conditions, necessary to support rules of the form $\bigsqcup\!\!\!\!\!\perp_A S = g$, are not represented in this logic, and so must be handled separately. This may be managed via an auxiliary structure which maintains information on disjointness of all pairs of granules. There are numerous data structures which may be employed to achieve this efficiently, but space limitations preclude further discussion.

4 Conclusions and Further Directions

A method for incorporating join and disjointness rules into the granule structure of multigranular relational attributes has been developed, and these methods have then been applied to the problem of integrating data at different granularities. A family of constraints, the TMCDs, are proposed as a means of checking integrity under such data integration. There are several avenues for further study.

- Data structures for effective implementation: The ideas developed in this paper will only prove useful if they can be implemented effectively. Although a few ideas along these lines are sketched in Discussion 3.7, a much more complete investigation, with implementation, is necessary.
- Query language: The work here proposes only constraints. An accompanying query language which takes into account the special needs of the multigranular framework must also be developed.
- Integration with monogranular approaches: To keep the initial investigation as focused as possible, the context of this paper is limited to sources based upon identical unirelational schemata, differing only in granularity. It is important to extend it to aspects common to monogranular approaches; in particular, multirelational sources based upon different schemata.

Acknowledgments: The work of M. Andrea Rodríguez, as well as a six-week visit of Stephen J. Hegner to Concepción, during which many of the ideas reported here were developed, were partly funded by Fondecyt-Conicyt grant number 1140428. Loreto Bravo was initially a collaborator, but was unable to continue due to other commitments. The authors gratefully acknowledge her contributions and insights during the early phases of this investigation.

References

1. O. Arieli, M. Denecker, and M. Bruynooghe. Distance semantics for database repair. *Ann. Math. Artif. Intell.*, 50(3-4):389–415, 2007.
2. E. Bertino, E. Camossi, and M. Bertolotto. Multi-granular spatio-temporal object models: Concepts and research directions. In *ICOODB*, pages 132–148, 2009.

3. L. E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011.
4. C. Bettini, X. S. Wang, and S. Jajodia. A general framework for time granularity and its application to temporal reasoning. *Ann. Math. Art. Intell.*, 22:29–58, 1998.
5. G. F. Bonham-Carter. *Geographic Information Systems for Geoscientists: Modelling with GIS*. Pergamon, 1995.
6. L. Bravo and M. A. Rodríguez. A multi-granular database model. In *FoIKS*, volume 8367 of *Lecture Notes in Computer Science*, pages 344–360. Springer, 2014.
7. A. Calì, D. Calvanese, G. De Giacomo, and M. Lenzerini. Data integration under integrity constraints. *Inf. Syst.*, 29(2):147–163, 2004.
8. E. Camossi, M. Bertolotto, and E. Bertino. A multigranular object-oriented framework supporting spatio-temporal granularity conversions. *International Journal of Geographical Information Science*, 20(5):511–534, 2006.
9. B. A. Davey and H. A. Priestly. *Introduction to Lattices and Order*. Cambridge University Press, second edition, 2002.
10. W. F. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn clauses. *J. Logic Programming*, 3:267–284, 1984.
11. M. Egenhofer, E. Clementine, and P. D. Felice. Evaluating Inconsistency among Multiple Representations. In *Spatial Data Handling*, pages 901–920, 1995.
12. M. Egenhofer and J. Sharma. Assessing the Consistency of Complete and Incomplete Topological Information. *Geographical Systems*, 1:47–68, 1993.
13. R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Addison Wesley, sixth edition, 2011.
14. S. Ginsburg and R. Hull. Order dependency in the relational model. *Theor. Comput. Sci.*, 26:149–195, 1983.
15. G. Grätzer. *General Lattice Theory*. Birkhäuser Verlag, second edition, 1998.
16. S. J. Hegner. Distributivity in incompletely specified type hierarchies: Theory and computational complexity. In J. Dörre, editor, *Computational Aspects of Constraint-Based Linguistic Description II*, pages 29–120. DYANA, 1994.
17. N. Iftikhar and T. B. Pedersen. Using a time granularity table for gradual granular data aggregation. *Fundam. Inform.*, 132(2):153–176, 2014.
18. B. Kuijpers, J. Paredaens, and J. V. den Bussche. On topological elementary equivalence of spatial databases. In *ICDT*, pages 432–446, 1997.
19. M. Lenzerini. Data integration: A theoretical perspective. In L. Popa, S. Abiteboul, and P. G. Kolaitis, editors, *PODS*, pages 233–246. ACM, 2002.
20. J. Lin and A. O. Mendelzon. Merging databases under constraints. *Int. J. Cooperative Inf. Syst.*, 7(1):55–76, 1998.
21. D. Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.
22. W. Ng. An extension of the relational data model to incorporate ordered domains. *ACM Trans. Database Syst.*, 26(3):344–383, 2001.
23. E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.
24. M. A. Rodríguez, L. E. Bertossi, and M. C. Marileo. Consistent query answering under spatial semantic constraints. *Inf. Syst.*, 38(2):244–263, 2013.
25. J. Szlichta, P. Godfrey, and J. Gryz. Fundamentals of order dependencies. *Proc. VLDB Endow.*, 5(11):1220–1231, 2012.
26. N. Tryfona and M. J. Egenhofer. Consistency among parts and aggregates: A computational model. *T. GIS*, 1(3):189–206, 1996.
27. J. Wijsen and R. T. Ng. Temporal dependencies generalized for spatial and other dimensions. In *STDBM'99*, pages 189–203, 1999.