Conceptual integration of multiple partial geometric models

Simone Santini[1] and Amarnath Gupta[2]

[1] Bio-Informatics Research Network, Department of Neuroscience, University of California San Diego 9500 Gilman Drive, La Jolla, CA 92093-0715, USA
ssantini@ncmir.ucsd.edu,
WWW home page: http://columbo.ucsd.edu
[2] San Diego Supercomputer Center, University of California San Diego 9500 Gilman Drive, La Jolla, CA 92093-0715, USA
gupta@sdsc.edu

Abstract. Many scientific databases need to manage complex 3D geometric data such as models of the cerebral cortex. Often the complexity of the data forces users to construct muliple, simpler representations, which cover the real, complete model only partially and approximately. In order to recover the original information one needs to integrate these partial and incomplete models. In this paper, we first develop a conceptual model for objects and relationships in 3D geometric data, as well as their partial and approximate representations. We then establish mapping relationships between different approximations of the same data. Finally, we present a geometric information integration technique that will perform the integration where possible, and determine, for some cases, when the integration cannot be performed.

1 Introduction

In this paper, we study a general version of the problem of answering queries using views ([8, 1, 2], see [6] for a recent survey) that arises in certain scientific application that deal with complex geometric data models. Location-sensitive data models were studied, for instance, in [7]. Our application context is that of brain volume rendering, like magnetic resonance imaging (MRI) and functional MRI, which are routinely used for studying abnormalities of the brain, planning surgery, and diagnosing brain tumors. Often these studies involve comparing the structure and behavior of certain brai regions under different treatments, for different patients or during different phases of a progressive disease. Informally expressed, a typical query on a brain scan database is the following: *Find the average thickness of region A for patients over age 60, having Alzheimer's disease, and showing no shrinkage within 0.5 cm around region B*

The cortex is a very complex geometric object, containing numerous folds, troughs and crevices, making the measurement of even seemingly simple properties like distance between two points very complicated and inefficient to compute. Consequently, different research groups have developed software to "simplify"

the cortex geometry, by mapping the original cortex data to geometrically simpler surfaces such as the sphere or the plane. Different representations may be used to label, measure and visualize different properties of the cortex geometry. The convenience of using simplified representations, of course, comes at a price: The distortion introduced during simplification may allow us to retrieve the original value of a property only after some mathematical computation and, often, within a certain error. More importantly, every simplification *preserves* only certain properties of the original data, while others are lost. For example, turning a cortex into a sphere preserves distances but loses the local curvature of the cortical surface. Given an arbitrary query, it is necessary to first find a set of usable representations from which each property (i.e., attribute) referred to in the query can be faithfully retrieved. Thus, the problem of answering queries using multiple representations, is akin to the problem of answering queries using views, but with the important distinction that the version of the attribute seen in a representation, is not identical to the original attribute value but *a functional correlate* of it.

In this paper, we reformulate the answering queries using views problem with two modifications. On one hand, we allow for partial representations which are not subsets of the conceptual model but which are related to it functionally. On the other hand, our model allows to include considerations of geometric invariance in the representation: it may be possible in some cases to compute a function both in the representation as in the conceptual model, but the two functions can't be related because the transformation between model and representation doesn't have the required invariance properties.

2 Preliminaries

Our domain model is a simple entity-relationship model, similar to that in [4], $M = (E, R, A)$, where $E = \{E_1, \ldots, E_n\}$ is the set of entity types, $R = \{R_1, \ldots, R_m\}$ is the set of relations, $A = \{a_{ik} : E_i \to T_{ik}\}$ is the set of entity attribute functions. For the sake of simplicity, we will not consider attributes on relations. An n-ary relation will be represented as a monic from an index set to the product space of the entity types it associates:

$$R_i : d \to E_{i,1} \times \cdots \times E_{i,r_i} \tag{1}$$

In the expression of a query involving the relation R_i we will often eliminate the existential quantifier over the index d, and use the notation

$$R(e_1 : E_1, \ldots, e_n : E_n) \equiv \exists d : R(d) = (e_1, e_2, \ldots, e_n) \tag{2}$$

Finally, we will consider *conjunctive* queries of the type

$$\begin{aligned}(a_1, \ldots, a_a, \beta_1, \ldots \beta_b) \leftarrow R_1(e_{11} : E_{11}, \ldots, e_{1m} : E_{1m}), \ldots \\ R_n(e_{n1} : E_{n1}, \ldots, e_{nm} : E_{1m}), c_1, \ldots, c_p\end{aligned} \tag{3}$$

where the c_i's are conditions on the attributes of relations or entities. Note that we don't allow relation variables, but only entity variables.

The model we will use in the paper is a simple model of the cerebral cortex: in which the cortex is represented as a surface (immersed in the three-dimensional Euclidean space) divided into regions; selected points of each region have associated histological data, such as neuron density measurements, or data about the neurons found in that region.

3 Representations

A representation $\hat{M}^q$ of a model M is defined as $\hat{M}^q = (\hat{E}^q, \hat{R}^q, \hat{A}^q)$. $\hat{E}^q = \{\hat{E}^q_1, \ldots, \hat{E}^q_{n_q}\}$ is the set of entity representations, $\hat{R}^q = \{\hat{R}^q_1, \ldots, \hat{R}^q_{m_q}\}$ is the set of relation representations, and $\hat{A} = \{\hat{a}^q_{ik} : \hat{E}^q_i \to \hat{T}^q_{ik}\}$ is the set of attribute representations.

Definition 1. *A representation $\hat{M}^q$ is* legal *if there exists a morphism ϕ^q such that*

- $\phi^q : E_i \to \hat{E}^q_i$
- $\phi^q : R_i \to \hat{R}^q_i$
- $\phi^q : a_{ik} \to \hat{a}^q_{ik}$
- *For every $\hat{R}^q_i$ the following diagram commutes:*

$$\begin{array}{ccc} d & \xrightarrow{R_i} & E_1 \times \cdots \times E_p \\ {\scriptstyle id}\downarrow & & \downarrow{\scriptstyle \phi^q \times \cdots \times \phi^q} \\ d & \xrightarrow[\hat{R}^q_i]{} & \hat{E}^q_1 \times \cdots \times \hat{E}^q_p \end{array} \tag{4}$$

- *For every a^q_{ik} there exist a function $f^q_{ik} : T_{ik} \to \hat{T}^q_{ik}$ such that the following diagram commutes:*

$$\begin{array}{ccc} E_i & \xrightarrow{a_{ik}} & T_{ik} \\ {\scriptstyle \phi^q}\downarrow & & \downarrow{\scriptstyle f^q_{ik}} \\ \hat{E}^q_i & \xrightarrow[\hat{a}^q_{ik}]{} & \hat{T}_{ik} \end{array} \quad . \tag{5}$$

If ϕ^q is an isomorphism, then the model $\hat{M}^q$ is said to be *faithful*. Similarly to the morphism ϕ^q between the conceptual model and its representation, it is possible to define a morphism between two representations [3].

Definition 2. *ψ is a morphism between the representation $\hat{M}^q$ and the representation $\hat{M}^p$, represented as $\hat{M}^q \xrightarrow{\psi} \hat{M}^p$ if $\psi : \hat{E}^q_i \to \hat{E}^p_i$, $\psi : \hat{R}^q_i \to \hat{R}^p_i$,*

$\psi : \hat{a}^q_{ik} \rightarrow \hat{a}^p_{ik}$, *for every* $\hat{R}^q_i$ *the following diagram commutes:*

$$\begin{array}{ccc} d & \xrightarrow{\hat{R}^q_i} & \hat{E}^q_1 \times \cdots \times \hat{E}^q_n \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle\psi\times\cdots\times\psi} \\ d & \xrightarrow[\psi\hat{R}^q_i]{} & \hat{E}^p_1 \times \cdots \times \hat{E}^p_n \end{array} \quad , \tag{6}$$

for every a^q_{ik} *there exist a function* $f^{qp}_{ik} : \hat{T}^q_{ik} \rightarrow \hat{T}^p_{ik}$ *such that the following diagram commutes:*

$$\begin{array}{ccc} \hat{E}^q_i & \xrightarrow{\hat{a}^q_{ik}} & \hat{T}^q_{ik} \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle f^{qp}_{ik}} \\ \hat{E}^p_i & \xrightarrow[\psi a^q_{ik}]{} & \hat{T}^p_{ik} \end{array} \tag{7}$$

whenever ψa^q_{ik} *is defined.*

An ordering relation can be established between representations as follows:

Definition 3. *Given two representations* $\hat{M}^q$ *and* $\hat{M}^p$*, it is* $\hat{M}^q \leq \hat{M}^p$ *if, for every morphism* $\hat{M}^q \xrightarrow{\psi} \hat{M}^p$*, there is* $\hat{M}^p \xrightarrow{\psi'} \hat{M}^q$ *such that* $\psi' \circ \psi = id$*.*

This partial ordering between representations captures to a certain degree the notion of "structural representativity" of a representation.

In many cases, the representation of a relation $\hat{R}^q_i$ is only partial, in the sense that there exist tuples $e_1, \ldots, e_n$ for which $R_i(e_1, \ldots, e_n)$ hold, but such that the relation $\hat{R}^q_i$ does not hold for the corresponding tuple $(\hat{e}^q_1, \ldots, \hat{e}^q_n)$.

Example

> Consider a model of the cerebral cortex, and a representation $\hat{M}$ of the cortical surface limited to the occipital cortex. The relation adjacent(e_1, e_2) will only be represented if e_1 and e_2 are in the occipital cortex.

This kind of structural restriction of a relation is captured by the concept of *defining condition*:

Definition 4. *Given a relation* R_i *and a representation* $\hat{R}^q_i$ *of that relation, let* P^q_i *be a predicate on the entities that* R_i *relates, and let* $id_{|P^q_i} : E_1 \times \cdots \times E_n \rightarrow E_1 \times \cdots \times E_n$ *be the restriction of the identity of* $E_1 \times \cdots \times E_n$ *to the set of entities for which the predicate* P^q_i *is true. The predicate* P^q_i *is the* defining condition *for the representation* R^q_i *if it makes the following diagram commute:*

$$\begin{array}{ccccc} d & \xrightarrow{R_i} & E_1 \times \cdots \times E_n & \xrightarrow{id_{|P^q_i}} & E_1 \times \cdots \times E_n \\ {\scriptstyle\phi^q}\downarrow & & & & \downarrow{\scriptstyle\phi^q\times\cdots\times\phi^q} \\ \hat{d} & & \xrightarrow[\hat{R}^q_i]{} & & \hat{E}^p_1 \times \cdots \times \hat{E}^p_n \end{array} \tag{8}$$

In terms of sets, this means that the the co-domain of the relation representation $\hat{R}_i^q$ is

$$\mathrm{cod}(\hat{R}_i^q) = \{(\hat{e}_{i,1}, \ldots, \hat{e}_{i,n}) : \hat{e}_{i,j} = \phi^q(e_{i,j}) \wedge e_{i,j} \in \mathrm{cod}(R_i^q) \wedge P_i^q(e_{i,1}, \ldots, \hat{e}_{i,n})\} \tag{9}$$

Attributes that can't be computed directly from a representation can sometimes be computed indirectly through a representation morphism. Given a representation $\hat{M}^q$, the set of attributes directly computable from $\hat{M}^q$ is

$$A_0(\hat{M}^Q) = \{a_{ik} : \exists \hat{a}_{ik}^q = \phi^q(a_{ij})\} \tag{10}$$

In order to define the set of attributes that can be computed indirectly from the representation $\hat{M}^q$, it is first necessary to determine which relations can be reached from $\hat{M}^q$ in a given number of steps. The set of representations directly reachable from $\hat{M}^q$ in k steps (or *k-reachable* from $\hat{M}^q$) is:

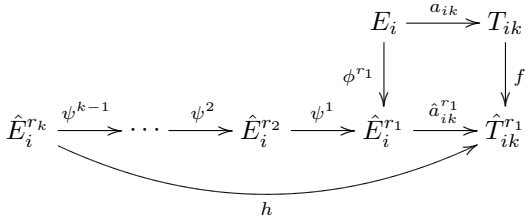
$$R_k(\hat{M}^q) = \left\{\hat{M}^r : \exists \psi_1, \ldots, \psi_k \, \hat{M}^q \xrightarrow{\psi_1 \circ \ldots \circ \psi_k} \hat{M}^r\right\} \tag{11}$$

while the set of k-reachable attributes is

$$A_k(\hat{M}^q) = \left\{a_{ik} : \exists \hat{M}^r \hat{a}_{ik}^r = \phi^r(a_{ij}) \wedge \hat{M}^r \in R_k(\hat{M}^q)\right\} \tag{12}$$

The set of representations and the set of attributes reachable from $\hat{M}^q$ will be indicated with $R_\infty(\hat{M}^q)$ and $A_\infty(\hat{M}^q)$ respectively.

From a functional point of view, an attribute a_{ik} is k-reachable from the representation $\hat{M}^{r_k}$ if there is a function h such that



(13)

An attribute which is not in $A_\infty(\hat{M}^q)$ is said to be *unreachable* from $\hat{M}^q$, while an attribute which does not belong to $A_\infty(\hat{M}^q)$ for any available representation $\hat{M}^q$ is said to be *hopeless*. Any query involving hopeless attributes can't be answered given the available representations.

4 Query rewriting

In a typical database queries are given in terms of the conceptual model M. The database, on the other hand, contains but a series of representations $\{\hat{M}^1, \ldots, \hat{M}^r\}$ and a series of morphisms $\hat{M}^i \xrightarrow{\psi^{ij}} \hat{M}^j$ between the representations.

In a rather general interpretation, a query can be formally defined as a partial recursive function from databases to databases [5]. This definition assume that the conceptual database is always isomorphic to its representation. In our case, this is not true, and we must distinguish between an *conceptual query* and an *grounded query*. A conceptual query is a partial recursive function from the set of models to a *result schema* given by the signature of the set of attributes that the query requires If $S_r = (T_1, \ldots, T_r)$ is the result schema, a query is a function $Q : M \to S_r$. We will write a query as:

$$(a_{i_1 k_1}(\xi_1), \ldots, a_{i_u k_u}(\xi_u)) \leftarrow R_{j_1}(\zeta_{11}, \ldots, \zeta_{1n}), \ldots, R_{j_p}(\zeta_{p1}, \ldots, \zeta_{pn}), \\ c_1(\nu_{11}, \ldots, \nu_{1m}), \ldots, c_v(\nu_{v1}, \ldots, \nu_{vm}) \quad (14)$$

Note that, for the sake of simplicity, we have assumed that all the relations are n-ary, and that all the conditions depend on m variables. The variables ξ_i and ζ_{ij} take values in the set of entities, while the variables ν_{ij} take value in the set of attributes.

The *grounding* of a query over the set of representations $\{\hat{M}^1, \ldots, \hat{M}^m\}$ consists in a representation function $\phi^{i_1} \times \cdots \times \phi^{i_p}$ and a query $Q' : \hat{M}^{i_1} \times \cdots \times \hat{M}^{i_p} \to \hat{S}_r$ such that there is a function f for which the following diagram commutes:

$$\begin{array}{ccc} M & \xrightarrow{\quad Q \quad} & S_r \\ {\scriptstyle \phi^{i_1} \times \cdots \times \phi^{i_p}} \downarrow & & \uparrow {\scriptstyle f} \\ \hat{M}^{i_1} \times \cdots \times \hat{M}^{i_p} & \xrightarrow[\quad Q' \quad]{} & \hat{S}_r \end{array} \quad (15)$$

Query rewriting consists, given the model M and a query Q, in the determination of a suitable set of representations $\hat{M}^{i_1} \times \cdots \times \hat{M}^{i_p}$ and a query Q' which grounds the original query.

One way to transform the query Q into the query Q' in a way that maintains the commutativity of the diagram can be sketched as follows:

1. add formally a representation $\hat{M}^0$ isomorphic to M to the set of representation;
2. translate Q into a query Q'_0 expressed in terms of $\hat{M}^0$ (which is a grounding of Q because of isomorphism);
3. replace one relation of $\hat{M}^0$ with its representation taken from some $\hat{M}^i$; replace all the attributes in $\hat{S}_r$ that can be computed from $\hat{M}^i$ with the corresponding representations;
4. if no relations and no attributes are computed in terms of $\hat{M}^0$, stop, otherwise repeat the previous step.

The substitutions of step 2 come from a set of possible substitutions called *substitution opportunities*, defined as follows:

Definition 5. *A* substitution opportunity *is a query fragment including one relation an w constraints:*

$$R_k(\zeta_{k1}, \ldots, \zeta_{kn}), c_1(\nu_{11}, \ldots, \nu_{1m}), \ldots, c_w(\nu_{w1}, \ldots, \nu_{wm}) \quad (16)$$

for which there is a representation $\hat{M}^q$ such that:

1. *$\hat{R}_k^q = \phi^q(R_k)$ exists,*
2. *All the values of the variables that make $c_1, \ldots, c_w$ true also make the defining condition P_k^q true.*

The substitution *of R_k with the representation $\hat{R}_k^i$ is indicated as $(R_k, c_1, \ldots, c_m) \rightrightarrows (\hat{R}_k^q, c_1, \ldots, c_m)$ or, if the conditions $c_1, \ldots, c_n$ can be omitted without causing confusion, as $R_k \rightrightarrows \hat{R}_k^q$.*

When we rewrite a query using a representation, it is important to guarantee that we don't lose the possibility of computing attributes. The following definition is related to the conditions under which this is guaranteed:

Definition 6. *A substitution*

$$(R_k(\zeta_1, \ldots, \zeta_n), c_1, \ldots, c_m) \rightrightarrows (\hat{R}_k^q(\zeta_1^q, \ldots, \zeta_n), c_1, \ldots, c_m)$$

is strictly non-disruptive *if the following condition is true:*

> *For each variable $\zeta_i : E_i$ that doesn't appear in any other relation but the relation R_k that is being substituted, all the attributes $a_{ij}(\zeta_i)$ that appear in the query can be computed from $\hat{R}_k^q$, that is, $a_{ij}(\zeta_i) = f(\hat{a}_{ij}^q(\zeta_i))$*

The substitution is weakly non-disruptive *if each attribute $a_{ij}(\zeta_i)$ can be computed indirectly from $\hat{R}_k^q$, that is, $a_{ij} \in A_\infty(\hat{M}^q)$*

The rationale of this definition is the following: if the only relation in which the entity type E_i appears is replaced with a representation which does not allow to compute the required attributes, then the attributes can no longer be computed. Non-disruptiveness guarantees that all the attributes that can only be computed from the replaced relation will be computable directly or indirectly from the representation that replaces it.

The following property is the key for the application of substitution operations:

Proposition 1. *Let $\hat{M}^0 \xrightarrow{Q} S_r$ be a query, and $\hat{M}^0 \times \hat{M}^1 \times \cdots \times \hat{M}^{n-1} \xrightarrow{Q'} \hat{S}_r$ be a grounding for it. Let $\hat{M}^0 \times \hat{M}^1 \times \cdots \times \hat{M}^{n-1} \times \hat{M}^n \xrightarrow{Q''} \hat{S}_r$ be a query obtained from Q' by a non-disruptive substitution $(R_k, c_1, \ldots, c_m) \rightrightarrows (\hat{R}_k, c_1, \ldots, c_m)$, then Q'' is a grounding of Q.*

Proof (sketch). Assume, for the sake of simplicity, that the relation R_k is binary: $R_k(\zeta_1, \zeta_2)$, with $\zeta_1 : E_1$, and $\zeta_2 : E_2$, and that the substitution contains a single condition c. Also, set $W = \hat{M}^1 \times \cdots \times \hat{M}^{n-1}$, so that the query Q' can be written as $\hat{M}^0 \times W \xrightarrow{Q'} \hat{S}_r$, and assume that the entity types E_1 and E_2 do not appear in any other relation.

The query Q can be written as the composition of two functions: $Q' = f \circ \sigma$, where σ is the selection fuction which selects the entities that satisfy the query condition, and f is the attribute computation function.

The function σ, in turn, can be decomposed as $\sigma = \sigma_0 \circ \sigma_k$. The function σ_k produces the set of variable assignments $\zeta_1 \leftarrow e_1 : E_1$, and $\zeta_2 \leftarrow e_2 : E_2$ that satisfy $(R_k, c_1, \ldots, c_m)$, with their variable bindings, while σ_0 will select those entities that satisfy all the other conditions and are compatible with the bindings of R_k. The function σ can be written as $\sigma : E_1 \times E_2 \times \cdots E_n \to E_1 \times E_2 \times \cdots E_n$.

After the substitution, the relation R_k is replaced by $\hat{R}_k$, which selects elements in $\hat{E}_1 \times \hat{E}_2$. Because of the properties of substitution, every pair in $E_1 \times E_2$ which make c true also make the defining condition of the representation true, therefore, for every pair (e_1, e_2) which would be selected by σ_k, there is a pair $(\hat{e}_1, \hat{e}_2)$ in $\hat{R}_k$. It is therefore possible to define a function $\hat{\sigma}_k : \hat{E}_1 \times \hat{E}_2 \to \hat{E}_1 \times \hat{E}_2$ which selects the pairs $(\hat{e}_1, \hat{e}_2)$ corresponding to the pairs (e_1, e_2) selected by σ_k, that is, a function $\hat{\sigma}_k$ such that

$$\begin{array}{ccc} E_1 \times E_2 & \xrightarrow{\sigma_k} & E_1 \times E_2 \\ \Big\downarrow{\scriptstyle \phi} & & \Big\downarrow{\scriptstyle \phi \times \phi} \\ \hat{E}_1 \times \hat{E}_2 & \xrightarrow[\hat{\sigma}_k]{} & \hat{E}_1 \times \hat{E}_2 \end{array} \tag{17}$$

Defining $\hat{\sigma} = \sigma_0 \circ \sigma_k$, one can show that $\sigma : \hat{E}_1 \times \hat{E}_2 \times \cdots E_n \to \hat{E}_1 \times \hat{E}_2 \times \cdots E_n$ and that the following diagram commutes

$$\begin{array}{ccc} E_1 \times \cdots \times E_n & \xrightarrow{\sigma_k} & E_1 \times \cdots \times E_n \\ \Big\downarrow{\scriptstyle \phi} & & \Big\downarrow{\scriptstyle \phi \times \phi \times \mathrm{id} \times \cdots \times \mathrm{id}} \\ \hat{E}_1 \times \hat{E}_2 \times E_3 \times \cdots \times E_n & \xrightarrow[\hat{\sigma}_k]{} & \hat{E}_1 \times \hat{E}_2 \times E_3 \times \cdots \times E_n \end{array} \tag{18}$$

In other words, the selection function can be rewritten so that the output is composed of the same tuples of entity, short of a transformation ϕ.

Since the transformation is non-disruptive, it is possible to rewrite in the same way the attribute computing function f as $\hat{f}$ so that the following diagram commutes:

$$\begin{array}{ccccc} E_1 \times \cdots \times E_n & \xrightarrow{\sigma_k} & E_1 \times \cdots \times E_n & \xrightarrow{f} & T_1 \times \cdots \times T_n \\ \Big\downarrow{\scriptstyle \phi} & & \Big\downarrow{\scriptstyle \phi \times \phi \times \mathrm{id} \times \cdots \times \mathrm{id}} & & \Big\downarrow{\scriptstyle \phi \times \phi \times \mathrm{id} \times \cdots \times \mathrm{id}} \\ \hat{E}_1 \times \hat{E}_2 \times E_3 \times \cdots \times E_n & \xrightarrow[\hat{\sigma}_k]{} & \hat{E}_1 \times \hat{E}_2 \times E_3 \times \cdots \times E_n & \xrightarrow[\hat{f}]{} & \hat{T}_1 \times \hat{T}_2 \times E_3 \times \cdots \times E_n \end{array} \tag{19}$$

In conclusion, the query rewriting problem outlined in this section can be cast in the following terms:

- Let M be a model, with representations $\hat{M}^1, \ldots, \hat{M}^r$; $M \xrightarrow{Q} S_r$ a query; $a = \{a_1, \ldots, a_k\}$ the set of attributes required by the query, $R = \{R_1, \ldots, R_n\}$ the set of query relations, and $\hat{R} = \{\hat{R}_1^1, \hat{R}_1^2 \ldots, \hat{R}_n^m\}$ the set of rewriting opportunity.
- find a subset $P \in \hat{R}$ such that: (1) for each $R_i \in R$ there is a ϕ^q such that $R_i^q = \phi^q R_i \in P$ and (2) for each $a_{ik} \in a$ there is $R_i^q \in \rho$ coming from a representation $\hat{M}^q$ such that $a_{ik} \in A_{k_i}(\hat{M})$ for some k_i.

5 Geometric Functions

Up to this point, we haven't quite considered the geometric nature of our data, but we have focussed on the structural properties of the representations, that is, on whether a given representation preserved the relations and the attributes of the conceptual model.

In particular, so far we have always considered that the conditions c_i were only on the value of attributes, and that the result of the query was also a set of tuples formed by atribute values. In geometric queries, however, one has often to compute functions that require the conservation of certain properties.

Example

> In the cortex model above, each region has an attribute cent_i, of type Point representing the centroid of the region. A typical spatial query, then, might request all the regions whose centroid is within a certain distance from the centroid of a given region. That is, the query will contain a condition like
>
> $$c_i(\nu_{i1}, \nu_{i2}) = d(\nu_{i1}\text{cent}, \nu_{i2}\text{cent}) \leq D \tag{20}$$
>
> where D is a constant.

This example leads to a number of observations. First, the representation of the entities ν_{i1}, ν_{i2} requires not only that the representation contain the attribute cent, but also that it preserve the properties of the distance function. By and large, every representation of regions will allow the computation of centroids, but there is a priori no guarantee that the distance between centroids in the representation will correspond to the distance between the centroids on the suface. Second, although the surface of the conceptual model is immersed in $\mathbb{R}^3$, for many applications (including brain modeling) the distance that is computed is not the $\mathbb{R}^3$ distance between the centroids of the regions, but the distance along the surface (more precisely: the lenght of the shortest surface geodesic that passes through the two points). The way in which this distance is computed, therefore, depends on the surface that is, on the representation. As a consequence, when the condition is rewritten, the distance function d will have to be replaced with a representation $d^q = \phi^q d$. The central concept for this type of replacement is that of *invariance*.

Definition 7. *Let X, Y be two spaces, and $f : X^n \to Y$ a function. Let $\mathfrak{G} : X \to X$ be a group of transformations on X. The function f is invariant with respect to the group $\mathfrak{G}$ if, for every $g \in \mathfrak{G}$, $f \circ g^n = f$.*

In particular, we are interested in invariance with respect to the following groups: the identity group (consisting only of the unit) $\mathfrak{I}$, the group of translations $\mathfrak{T}$, the group of direct isometries (rotation and translation) $\mathfrak{D}$, the group of homeomorphisms $\mathfrak{H}$, and the general permutation group $\mathfrak{P}$.

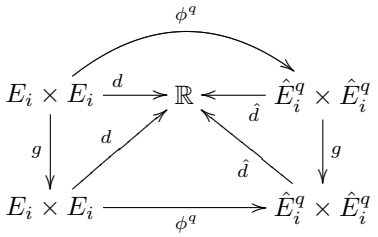
A function invariant to $\mathfrak{I}$ is the function that computes the coördinates of the center of a particulat region in a given reference system. A function invariant to $\mathfrak{T}$ is that which computes the orientation of a region with respect to a reference line. A typical example of a function invariant to $\mathfrak{D}$ is distance, while functions invariant to $\mathfrak{H}$ are, for instance, functions that determine whether two regions touch each other, or count the number of holes in a region.

When we go from the original model to a representation, the transform may fail to be invariant with respect to some of these groups. Consequently, functions that rely on the corresponding invariants can't be computed from the representation.

Example

> A *flat map* is a projection of the cortical surface on a plane in a way that maintains, as well as possible, the area of certain anatomically relevant regions. Since the cortical surface is not topologically equivalent to a plane, when a flat map is created, *cuts* are introduced which may go across regions. The topological invariant *connectedness* and the isometric invariant *distance* are lost in this map: regions that are connected in the cortex may fail to be connected in the map, and it is impossible to recover cortical distances from the flat map. Therefore, every query containing predicates about the connectdeness of regions, or conditions on the distance between points can't be answered using the flat map.

The conditions under which this happen depend on the representation morphism ϕ^q, in particular, for an invariance involving the entity type E_i, on the function $\phi^q : E_i \to \hat{E}_i^q$. Consider, in the way of example, the case of distance computation. The situation is summarized by the following diagram:

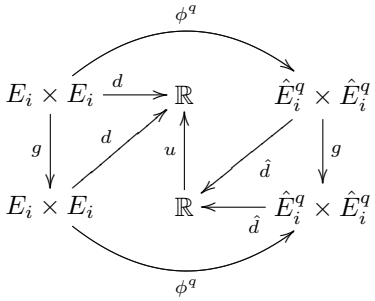


(21)

From the diagram it is clear that the representation of E_i will be distance-invariant if for all $g \in \mathfrak{G}$, $\phi^q = g^{-1} \circ \phi^q \circ g$, and $d = \hat{d} \circ \phi^q$. In general, if this

is true for a group $\mathfrak{G}$, we will say that the representation is $\mathfrak{G}$-*covariant in the strong sense* (or *strongly* $\mathfrak{G}$-*covariant*).

In some cases, this condition is too restrictive: all we really need is that there be a way to compute the distance d starting from the distance $\hat{d}$, that is, that there exist a function u such that:



(22)

note that it follows from this diagram that if $d = \hat{d} \circ \phi^q$, then $u = \mathrm{id}$. In the general case, one requires that the function u be well defined and computable, that is, that it be expressible only as a function of $\hat{d}$ and ϕ^q. In general, if this is true for a group $\mathfrak{G}$, we will say that the representation is $\mathfrak{G}$-*covariant in the weak sense* (or *weakly* $\mathfrak{G}$-*covariant*).

Definition 8. *Let $c(\nu_1 : E_1, \ldots, \nu_n : E_n)$ be a condition in the query, which can depend on the computation of certain functions on the entity variables ν_i. Let $\mathfrak{g}(c)$ the group of transformations to which c is invariant. A rewriting $\{\hat{M}^1, \ldots, \hat{M}^m\}$ is* $\mathfrak{g}$-preserving *if the following conditions are true:*

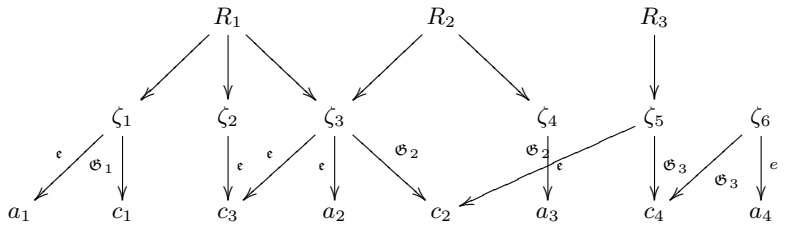
1. *there is a representation $\hat{M}^p$ which contains a representation of all the entity types of c;*
2. *all the variables ν_i have been replaced with variables ν_i^p which take values in the representation $\hat{M}^p$;*
3. *the representation $\hat{M}^p$ is $\mathfrak{g}$-covariant, at least in the weak sense.*

Proposition 2. *Let Q be a query and Q' a rewriting so that for every condition c invariant with respect to a group $\mathfrak{G}$ the rewriting is $\mathfrak{G}$-preserving and such that for every attribute function a_{ik} invariant with respect to a group $\mathfrak{H}$, the rewriting is $\mathfrak{H}$-preserving. Then the query Q' is a grounding of Q.*

The proof is a repeated application of the invariance property, and is omitted.

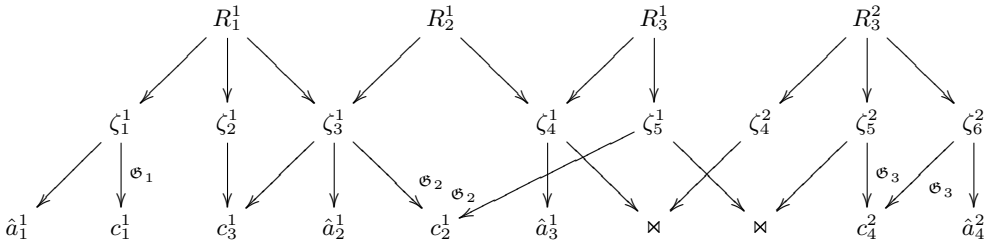
6 The Query rewriting process

A query like that of (14) can be represented as in the following diagram



(23)

where we have already identified all the common variables between the conditions, the relations, and the attribute computation inserting, if necessary, variable matching conditions of the form $c(\zeta_1, \zeta_2) = (\zeta_1 \equiv \zeta_2)$. The symbols $\mathfrak{G}_i$ on some of the arrows mean that the condition or attribute at the end of the arrow is invariant to the group $\mathfrak{G}_i$. The process of query rewriting consists of transforming this diagram, through a process of repeated substitutions of relations, into an equivalent diagram composed exclusively of representations. For the query (23), one such diagram is the following:



(24)

In this diagram, the superscripts attached to conditions and attributes refer to the representation that is used to compute them. The diagram uses two representations: $\hat{M}^1$, which provides representations for the relations R_1, R_2, and R_3, and $\hat{M}^2$, which represents the relation R_3. Note that the relations R_3 is split between two representations: $\hat{R}_3^1$ is used to compute the condition c_2 and the attribute a_3, and the representation $\hat{R}_3^2$, which is used to compute the condition c_4 and the attribute a_4. While the entity type of the variable $\zeta_5 : E_5$ in the original relation is invariant to groups $\mathfrak{G}_2$ and $\mathfrak{G}_3$, in the derived diagram, the representation $\hat{E}_5^1$ is invariant only to $\mathfrak{G}_2$, and the representation $\hat{E}_5^2$ is invariant only to $\mathfrak{G}_3$. Since the relation is split, it is necessary to bind the variables that appear in both reprsentations (ζ_4 and ζ_5, in this case) so that they represent the same instance of the entity. This is done by introducing binding relations[3] $\bowtie$ between the variables ζ_4^1 and ζ_4^2 and between the variables ζ_5^1 and ζ_5^2. We assume

[3] We use this symbol to represent the binding relation because, in most cases, the condition results in a join on the unique identifier of the entities between the different relations.

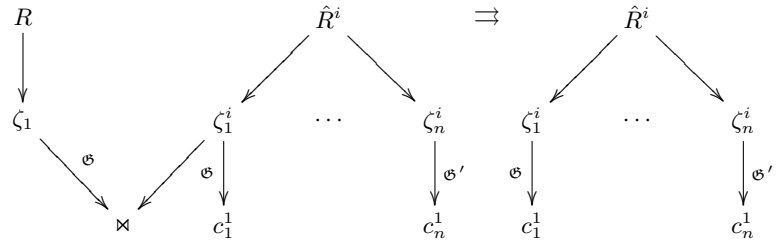
that all the instances of all entities have a unique identifier so that the binding condition can be written as $\zeta_4^1.\mathrm{id} = \zeta_4^2.\mathrm{id}$.

The passage from one diagram to another is done through *legal substitutions*, changes that leave the semantics of the query unchanged, while reducing the presence of the conceptual model and introducing representations of the various relations. Legal substitutions belong to four groups, which we call α-substitutions, β-substitutions, γ-substitutions, and ϵ-substitutions, defined below. From the diagrams above it is clear that attributes and conditions play a similar rôle and, from the point of view of representation, they are interchangeable, therefore we will consider diagrams with conditions only, to avoid the multiplication of special cases.

ϵ-substitution ϵ-substitutions are the simplest: they remove from the diagram a relation that is not used to compute anything. There are two types of ϵ-substitution. The simplest is

$$R \rightrightarrows \emptyset \tag{25}$$

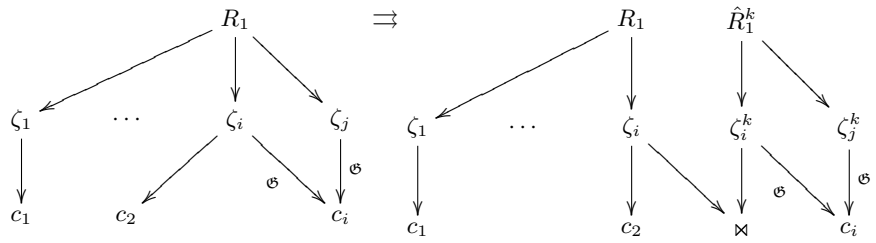
which states that a relation R disconnected from any variable can be eliminated from the diagram. The second ϵ-substitution is



(26)

which states that a relation with variables that do not participate in the computation of any condition or attribute can be eliminated.

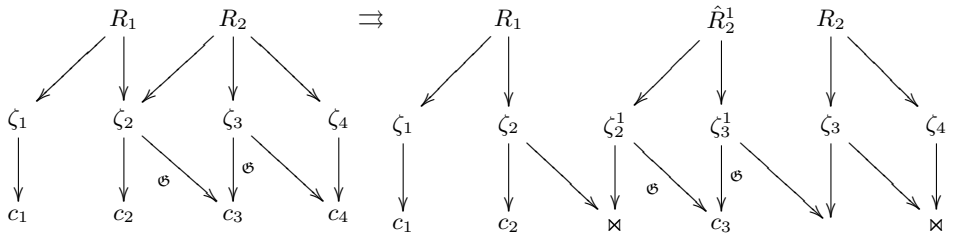
α-substitutions. α-substitutions deal with the replacement of a single relation, or part of a single relation, with a representation. A general form of an α-substitution is:



(27)

The conditions under which the substitution can be made are that the entity types E_i and E_j on which the variables ζ_i and ζ_j take values be represented in $\hat{R}_1^k$ and that they both be $\mathfrak{G}$-invariant.

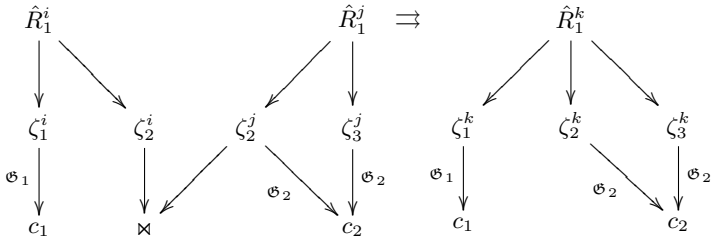
β-substitutions β-substitutions intoduce representations for entity variables involved in two or more relations. A rather general example of β-substitutions is the following:



(28)

Similarly to the previous case, the substitution can be done if all the entity types involved in the condition being represented are contained in the chosen representation, and if they are $\mathfrak{g}(c)$-invariant.

γ-substitutions γ-substitutions merge redundant representations that have been introduced while removing model relations using α- and β-substitutions. Their general form is the following:



(29)

The problem of query rewriting can therefore be cast into the problem of finding the optimal path in an optimization tree: the nodes of the tree represent diagrams and the edges are marked with the substitutions that bring from a diagram to another. The detailed description of the optimization algorithms is beyond the scope of this paper.

The following proposition is an immediate consequence of the fact that all the subsitutions presented here are non-disruptive:

Proposition 3. *Let* $\hat{M}^0 \xrightarrow{Q} S_r$ *be a query, and* $\hat{M}^0 \times \hat{M}^1 \times \cdots \times \hat{M}^{n-1} \xrightarrow{Q'} \hat{S}_r$ *be a query obtained from* Q *through* α-, β-, γ-, *and* ϵ-*substitutions. Then* Q' *is a grounding for* Q.

In other words, the set of α-, β-, γ-, and ϵ-substitutions is sound. Its completeness is a more complicated issue. It is possible to show that, every query Q that can be represented by a given set of representations can be grounded in that set using only α-, β-, γ-, and ϵ-substitutions. We still don't know whether *all* groundings of Q can be found using only α-, β-, γ-, and ϵ-substitutions.

7 Conclusions

In this paper we have begun to discuss the problem of answering queries from multiple partial representations of a conceptual model under two defining conditions: (1) the representations are not subsets of the model (as is the case for query answering using views), but are functionally related to the model, and (2) the conditions and attributes are geometric, which entails that they have invariance properties with respect to certain transformation group that must be preserved.

We have discussed the relations between representations and model, and between representations, and we have shown that there are operations, which we call *substitutions*, which can be used to rewrite the query Q from the conceptual model in which it is expressed to the representations that the database use. The use of these substitutions for query rewriting results in an optimization problem.

We are continuing the work presented in this paper in three directions: first, we are exploring the completeness properties of the set of substitutions. In particular, we are interested in whether *all* possible groundings of a query can be derived using ony the substitutions. Second, we are studying PTIME optimization algorithms to solve the problem posed by the query rewriting. Finally, we are trying to extend the model to other circumstances of practical interest, most notably the case in which the representations does not allow an exact computation of the attributes of the model, but itroduce an error.

References

1. Serge Abiteboul and Oliver M. Duschka. Complexity of answering queries using materialized views. In *Proceedings of PODS*, pages 254–263, 1998.
2. Foto Afrati, Che Li, and Jeffrey Ullman. Generating efficient plans for queries using views. In *Proceedings of SIGMOD*, pages 319–330, 2001.
3. Andrea Asperti and Giuseppe Longo. *Categories, Types, and Structures*. MIT Press, 1991.
4. Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Accessing data integration systems through conceptual schemas. In *Proceedings of ER2001*, pages 270–284, 2001.
5. A. Chandra and D. Harel. Computable queries for relational databases. *Journal of Computer and System Sciences*, 21(2):156–178, 1980.
6. Alon Halevy. Answering queries using views: a survey. *Very Large Data Bases Journal*, 2001.
7. Yoshiharu Ishikawa and Hiroyuki Kitagawa. Source description-based approach for the modeling of spatial information integration. In *Proceedings of ER2001*, pages 41–55, 2001.
8. Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering queries using views. In *Proceedings of PODS*, pages 95–104, 1995.