Specifying Valid Compound Terms in Interrelated Faceted Taxonomies

Anastasia Analyti[1], Yannis Tzitzikas[1,2], and
Nicolas Spyratos[3]

[1] Institute of Computer Science, FORTH-ICS, Greece
[2] Department of Computer Science, University of Crete, Greece
[3] Laboratoire de Recherche en Informatique, Universite de Paris-Sud, France
{analyti, tzitzik}@ics.forth.gr, spyratos@lri.fr

Abstract. In previous work, we proposed an algebra whose operators allow to specify the valid compound terms of a faceted taxonomy, in a flexible manner (by combining positive and negative statements). In this paper, we treat the same problem but in a more general setting, where the facet taxonomies are not independent but are (possibly) interrelated through narrower/broader relationships between their terms. The proposed algebra, called *Interrelated Facet Composition Algebra* (*IFCA*), is more powerful, as the valid compound terms of a faceted taxonomy can be derived through a smaller set of declared valid and/or invalid compound terms. An optimized (w.r.t. the naive approach) algorithm that checks compound term validity, according to a well-formed IFCA expression, and its worst-time complexity are provided.

Keywords: interrelated faceted taxonomies, valid compound terms, algebra, dynamic taxonomies, web search.

1 Introduction

The provision of effective and efficient general-purpose access services for end-users is a challenging task. In general, we could say that query services are either too simplistic (e.g., free text queries in IR systems or Web search engines), or too sophisticated (e.g., SQL queries or Semantic Web Queries). On the other hand browsing is either too simplistic (e.g., plain Web links) or very application specific (dynamic pages derived by specific application programs). Information exploration services could bridge this gap and provide effective and efficient general purpose access services. Indeed, dynamic taxonomies [8, 10] and faceted search [15, 17, 4] is a successful example [11] that is currently very common in E-commerce applications in the Web (e.g., eBay Express[4]).

Roughly, a *faceted taxonomy* is a set of taxonomies, each one describing the domain of interest from a different (preferably orthogonal) point of view [4]. Having a faceted taxonomy, each domain object (e.g., a book or a Web page) can be indexed using a *compound term*, i.e., a set of terms from the different facets.

[4] http://www.express.ebay.com/

For example, assume that the domain of interest is a set of hotel Web pages in Greece, and suppose that we want to provide access to these pages according to three facets: the *Location* of the hotels, the *Sports* facilities they offer, and the *Season* they are open, as shown in Figure 1. Each object can be described using a compound term. For example, a hotel in Crete which provides sea ski and wind-surfing facilities, and is open during the summer will be described by the compound term $\{Crete, SeaSki, Windsurfing, Summer\}$.

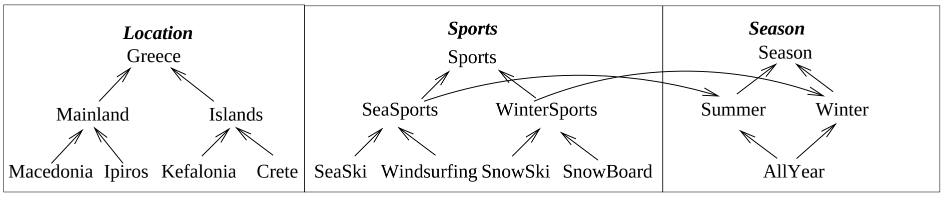

Fig. 1. Three interrelated facets

Faceted taxonomies carry a number of well known advantages over single taxonomies (clarity, compactness, scalability), but they also have a severe drawback: the high cost of avoiding invalid compound terms, i.e. compound terms that do not apply to any object in the domain. For example, the compound term $\{Crete, SnowBoard\}$ is an invalid compound term, as there are no hotels in Crete offering snow-board facilities. The interaction paradigm of faceted search and dynamic taxonomies can enable users to browse only nodes that correspond to valid compound terms [15, 17, 8] (e.g. see demos[5,6]). However, if the computation of such compound terms is based *only* on the objects that have already been indexed (as in [17]) then this interaction paradigm cannot be exploited, in the case where there are no indexed objects.

The availability of algebraic expressions describing the valid compound terms of a faceted taxonomy enables the *dynamic* generation of *navigation trees*, whose nodes correspond to valid compound terms, only [15]. These navigational trees can be used for indexing (for avoiding errors) and browsing. Additionally, if we have a *materialized faceted taxonomy* $\mathcal{M}$ (i.e., a corpus of objects indexed through a faceted taxomony) then specific mining algorithms (such as, these in [13]) can be used for expressing the extensionally valid compound terms of $\mathcal{M}$ in the form of an algebraic expression. Obviously, such mined algebraic expressions enable the user to take advantage of the aforementioned interaction scheme, without having to resort to the (possibly, numerous) instances of $\mathcal{M}$. Furthermore, algebraic expressions describing the valid compound terms of a faceted taxonomy can be exploited in other tasks, such as retrieval optimization [15], configuration management [1], consistency control [14], and compression [12].

[5] http://flamenco.berkeley.edu/demos.html
[6] http://simile.mit.edu/wiki/Longwell_Demos

This algebraic approach was first proposed in [15], where the *Compound Term Composition Algebra* (*CTCA*) was defined. CTCA has *four operators* (two positive and two negative), based on which one can built an *algebraic expression* to specify the valid compound terms of a faceted taxonomy, in a flexible and easy manner. In each algebraic operation, the designer has to declare either a small set of compound terms known to be valid (from which other valid compound terms are inferred), or a small set of compound terms known to be invalid (from which other invalid compound terms are inferred).

For example, if a user declares (in a positive operation) that the compound term $\{Crete, SeaSki\}$ is valid then it is inferred that the compound term $\{Crete,$ $SeaSports\}$ is also valid. On the other hand, if a user declares (in a negative operation) that the compound term $\{Crete, WinterSports\}$ is invalid then it is inferred that the compound term $\{Crete, SnowBoard\}$ is also invalid. In our example, this means that the designer can specify all valid compound terms of the faceted taxonomy by providing a relatively small number of (valid or invalid) compound terms. This is an important feature as it minimizes the effort needed by the designer. Moreover, only the expression defining the set of valid compound terms needs to be stored (and not the set itself), as an inference mechanism can check whether a compound term belongs to the set of defined compound terms, in polynomial time [15]. Based on this inference mechanism, an algorithm for deriving navigation trees, *on the fly*, is provided in [15] and is implemented in the FASTAXON system [16].

In this paper, we also treat the problem of specifying the valid compound terms of a faceted taxonomy but, in contrast to CTCA, we assume that facets can be interrelated through narrower/broader relationships (denoted by $<_{\mathtt{F}}$) between their terms. The proposed algebra, called *Interrelated Facet Composition Algebra* (*IFCA*), includes *three operators* (one positive and two negative). Compared to CTCA, the present approach is more powerful, as the valid compound terms can be derived through a smaller set of declared valid and/or invalid compound terms. Thus, the effort needed to build the desired algebraic expression is reduced. For example, in Figure 1, assume that the facets *Sports* and *Season* are related by adding the following relationships: $SeaSports <_{\mathtt{F}} Summer$ and $WinterSports <_{\mathtt{F}} Winter$, meaning that all hotels offering sea sports are open in summer, and that all hotels offering winter sports are open in winter. Assume now that the designer declares the compound term $\{Crete, SeaSports\}$ as valid. Then, we can infer that $\{Crete, Summer\}$ is also valid. Additionally, if the designer declares that the compound term $\{Hersonissos,$ $Winter\}$ is invalid then we can infer that $\{Hersonissos,$ $WinterSports\}$ is also invalid.

Apart from defining IFCA, in this paper, we present an algorithm for checking compound term validity, according to a well-formed IFCA expression. The algorithm is optimized w.r.t. the naive approach and its worst-time complexity is provided. We could say that CTCA and IFCA, are fully *intensional* algebras, in contrast to dynamic taxonomies [8], or Formal Concept Analysis [3], which are both *intensional* (due to the existence of hierarchies and their semantics) and *extensional* (as they discard compound terms with empty extension). Further, as we have shown in [14, 1], *Description Logics* (DLs) [2] and *definite logic programs* [7] cannot represent the "mode interchange" from positive to negative operations (and vice-versa)[7] that occur in a general CTCA, and thus also IFCA, expression.

The remaining of this paper is organized as follows: Section 2 describes formally compound taxonomies and interrelated faceted taxonomies. Section 3 describes the Interrelated Facet Composition Algebra. Section 4 presents an algorithm that checks compound term validity, according to a well-formed IFCA expression, along with its worst-time complexity. Finally, Section 5 concludes the paper and identifies issues for further research.

2 Interrelated Faceted Taxonomies

In this section, we define compound taxonomies and interrelated faceted taxonomies.

A *terminology* is a finite set of names, called *terms*. A *taxonomy* is a pair $(\mathcal{T}, \leq)$, where $\mathcal{T}$ is a *terminology* and $\leq$ is a partial order over $\mathcal{T}$, called *subsumption*. A *compound term* over $\mathcal{T}$ is any subset of $\mathcal{T}$. For example, the following sets of terms are compound terms over the taxonomy *Sports* of Figure 1: $s_1 = \{SeaSki, Windsurfing\}$, $s_2 = \{SeaSports\}$, and $s_3 = \emptyset$.

A *compound terminology* S over $\mathcal{T}$ is any set of compound terms that contains the compound term $\emptyset$. The set of all compound terms over $\mathcal{T}$ can be ordered using the *compound ordering* over $\mathcal{T}$, defined as: $s \preceq s'$ iff $\forall t' \in s'$, $\exists t \in s$ such that $t \leq t'$. That is, $s \preceq s'$ iff s contains a narrower term for every term of s'. In addition, s may contain terms not present in s'. Roughly, $s \preceq s'$ means that s carries more specific information than s'. For example, $\{SeaSki, Windsurfing\} \preceq \{SeaSports\} \preceq \emptyset$. We say that two compound terms s, s' are *equivalent* $s \sim s'$ iff $s \preceq s'$ and $s' \preceq s$. For example, $\{SeaSki, SeaSports\}$ and $\{SeaSki\}$ are equivalent. Intuitively, equivalent compound terms carry the same information. Note that if $s \sim s'$ then $minimal_{\leq}(s) = minimal_{\leq}(s')$.

A *compound taxonomy* over $\mathcal{T}$ is a pair $(S, \preceq)$, where S is a compound terminology over $\mathcal{T}$, and $\preceq$ is the compound ordering over $\mathcal{T}$ restricted to S. Let $P(\mathcal{T})$ be the set of all compound terms over $\mathcal{T}$ (i.e., the powerset of $\mathcal{T}$). Clearly, $(P(\mathcal{T}), \preceq)$ is a compound taxonomy over $\mathcal{T}$.

Let s be a compound term. The broader and the narrower compound terms of s are defined as follows: $\mathtt{Br}(s) = \{s' \in P(\mathcal{T}) \mid s \preceq s'\}$ and $\mathtt{Nr}(s) = \{s' \in P(\mathcal{T}) \mid s' \preceq s\}$.

Let S be a compound terminology over $\mathcal{T}$. The broader and the narrower compound terms of S are defined as follows: $Br(S) = \cup\{\mathtt{Br}(s) \mid s \in S\}$ and $Nr(S) = \cup\{\mathtt{Nr}(s) \mid s \in S\}$.

We say that a compound term s is *valid* (resp. *invalid*), if, in the current state of affairs, there is at least one (resp. no) object of the underlying domain

[7] Though, as shown in [1], this mode "interchange" can be represented through logic programs with lists and weak negation under Clark's semantics [7], with no computational advantage.

indexed by all terms in s. We assume that every term of $\mathcal{T}$ is valid. However, a compound term over $\mathcal{T}$ may be invalid. Obviously, if s is a valid compound term, all compound terms in $\mathtt{Br}(s)$ are valid. Additionally, if s is an invalid compound term, all compound terms in $\mathtt{Nr}(s)$ are invalid.

One way of designing a taxonomy is by identifying a number k of different aspects of the domain of interest and then designing one taxonomy per aspect. As a result, we obtain a set of taxonomies $F_i = (\mathcal{T}_i, \leq_i)$, for $i = 1, ..., k$, called *facets*. In our framework, facets may be related through a narrower/broader relation $<_{\mathtt{F}}$ between their terms. We require that the transitive closure[8] of the union of $<_{\mathtt{F}}$ with the facet subsumption relations $\leq_i$, for $i = 1, ..., k$, that is $\leq \, = ((\bigcup_{i=1}^{k} \leq_i) \bigcup <_{\mathtt{F}})^*$, is a partial order over $\mathcal{T} = \bigcup_{i=1}^{k} \mathcal{T}_i$. Thus, $\leq$ includes *only trivial* cycles.

Specifically, given a set of facets and a relation $<_{\mathtt{F}}$, we define an *interrelated faceted taxonomy* as follows:

Definition 1 (Interrelated faceted taxonomy). Let $\{F_1, ..., F_k\}$ be a set of taxonomies, where $F_i = (\mathcal{T}_i, \leq_i)$, for $i = 1, ..., k$, and $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset$, if $i \neq j$. Additionally, let $<_{\mathtt{F}} \subseteq \bigcup_{i \neq j} \mathcal{T}_i \times \mathcal{T}_j$ and let $\leq \, = ((\bigcup_{i=1}^{k} \leq_i) \bigcup <_{\mathtt{F}})^*$ such that $\leq$ is a partial order. Then, the pair $\mathcal{F} = (\mathcal{T}, \leq)$, where $\mathcal{T} = \bigcup_{i=1}^{k} \mathcal{T}_i$, is a taxonomy that we call the *interrelated faceted taxonomy generated* by $\{F_1, ..., F_k\}$ and $<_{\mathtt{F}}$. We call the taxonomies $F_1, ..., F_k$ the *facets* of $\mathcal{F}$. Additionally, if $<_{\mathtt{F}} = \emptyset$ then $\mathcal{F}$ is also called *simple faceted taxonomy.*

Clearly, all definitions introduced so far apply also to an interrelated faceted taxonomy $\mathcal{F} = (\mathcal{T}, \leq)$. For example, the set $S = \{\{Islands\}, \{SeaSports\}, \{Greece, SeaSports\}, \emptyset\}$ is a compound terminology over the terminology $\mathcal{T}$ of the interrelated faceted taxonomy, shown in Figure 1. Additionally, the pair $(S, \preceq)$ is a compound taxonomy over $\mathcal{T}$. For reasons of brevity, we omit the term $\emptyset$ from the example compound terminologies.

3 The Interrelated Facet Composition Algebra

Let $\mathcal{F} = (\mathcal{T}, \leq)$ be the interrelated faceted taxonomy, generated by a set of facets $\{F_1, ..., F_k\}$ and a relation $<_{\mathtt{F}}$. To begin with, we associate each facet $F_i = (\mathcal{T}_i, \leq_i)$ with a compound terminology T_i that we call the *basic compound terminology* of F_i. The basic compound terminologies are the "building blocks" of our algebra. Specifically, $T_i = Br\{\{t\} \mid t \in \mathcal{T}_i\} \cap \mathcal{P}(\mathcal{T}_i)$, for $i = 1, ..., k$.

For example, in Figure 1, the basic compound taxonomy of *Season* is: $\{\{Season\}, \{Summer\}, \{Winter\}, \{AllYear\}, \{Summer, Winter\}\}$. As every term t of a facet is considered valid, all compound terms in T_i are valid compound terms over $\mathcal{T}_i$.

Let $\mathcal{S}$ denote the set of all compound terminologies over $\mathcal{T}$. The *Interrelated Facet Composition Algebra* (*IFCA*) is an algebra over $\mathcal{S}$, which includes three operations, namely the *plus-product*, the *minus-product*, and the *minus-self-product*

[8] Given a binary relation R, we shall use R^* to denote its reflexive and transitive closure.

operations. For defining the desired compound taxonomy, the designer has to formulate an algebraic expression e, using these three operations and initial operands the basic compound terminologies. The plus-product, minus-product, and minus-self-product operations of IFCA operate over a set of compound terminologies $S_1, ..., S_n$ and generalize the corresponding operations of CTCA [15].

Let $S_1, ..., S_n$ be compound terminologies over $\mathcal{T}$. The *domain* of $S_1, ..., S_n$, denoted by $\mathcal{D}_{S_1,...,S_n}$, is the powerset of all terms in $\mathcal{T}$ that appear in $S_1, ..., S_n$. For example, let $S_1 = \{\{Greece, Sports\}, \{Season\}\}$ and let $S_2 = \{\{Season\}, \{Summer\}\}$ then[9] $\mathcal{D}_{S_1,S_2} = \mathcal{P}(\{Greece, Sports, Summer\})$. Intuitively, the set of compound terms $\mathcal{D}_{S_1,...,S_n}$ is used to delimit the range of the IFCA plus-product and minus-product operations over $S_1, ..., S_n$.

Additionally, we provide the auxiliary operation $\oplus$ over $\mathcal{S}$, called *product*. This operation results in a compound terminology, whose compound terms are all possible combinations (unions) of compound terms from its arguments. Specifically, let $S_1, ..., S_n \in \mathcal{S}$. The *product* of $S_1, ..., S_n$ is defined as: $S_1 \oplus ... \oplus S_n = \{\ s_1 \cup ... \cup s_n \mid s_i \in S_i\}$. Examples of the product operation are provided in [15]. It is easy to see that: $\emptyset \in\ S_1 \oplus ... \oplus S_n\ \subseteq \mathcal{D}_{S_1,...,S_n}$.

Let $S_1, ..., S_n$ be compound terminologies over $\mathcal{T}$. Intuitively, the *plus-product* operation $\oplus_P(S_1, ...S_n)$ specifies valid compound terms in $\mathcal{D}_{S_1,...,S_n}$, through a declared set of valid compound terms $P \subseteq \mathcal{D}_{S_1,...,S_n}$.

Definition 2 (Plus-product operation). Let $S_1, ..., S_n \in \mathcal{S}$ and $P \subseteq \mathcal{D}_{S_1,...,S_n}$. The *plus-product* of $S_1, ..., S_n$ with respect to P is defined as follows:

$$\oplus_P(S_1, ..., S_n) = Br(S_1 \cup ... \cup S_n \cup P) \cap \mathcal{D}_{S_1,...,S_n}. \ \square$$

This operation results in a compound terminology consisting of the compound terms in $\mathcal{D}_{S_1,...,S_n}$ which are broader than an element of the initial compound terminologies union P. This is because, assuming that all compound terms of S_i, for $i = 1, ..., n$, and P are valid then all compound terms in $Br(S_1 \cup ... \cup S_n \cup P)$ are also valid. We delimit this set to $\mathcal{D}_{S_1,...,S_n}$, as we are interested only in the compound terms, formed by terms appearing in $S_1, ..., S_n$.

It is easy to see that: (i) the operation plus-product is commutative, (ii) the smaller the parameter P, the smaller the resulting compound terminology, and (iii) for any parameter P, we need to consider only its minimal (with respect to $\leq$) elements. The last property can be used for optimization, i.e., for minimizing the space needed for storing the parameter P.

The following proposition shows that the application of a $\oplus_P$ operation[10] on other $\oplus_P$ operations results in a single $\oplus_P$ operation, allowing the simplification of an IFCA expression.

Proposition 1. Let the compound terminologies $S_i \in \mathcal{S}$, for $i = 1, ..., n$. It holds: $(\oplus_{P_1}(S_1, ..., S_l)) \oplus_{P_2} (\oplus_{P_3}(S_{l+1}, ..., S_n)) = \oplus_{minimal_{\leq}(P_1 \cup P_2 \cup P_3)}(S_1, ..., S_n)$.

[9] For $S \subseteq \mathcal{T}$, $\mathcal{P}(S)$ denotes the powerset of S.

[10] For binary operations, we also use the infix notation.

Let $S_1, ..., S_n$, where $n \geq 2$, be compound terminologies over $\mathcal{T}$. Intuitively, the *minus-product* operation $\ominus_N(S_1, ...S_n)$ specifies which compound terms in $S_1 \oplus ... \oplus S_n$ are invalid, through a declared set of invalid compound terms $N \subseteq \mathcal{D}_{S_1,...,S_n}$.

Definition 3 (Minus-product operation). Let $S_1, ..., S_n \in \mathcal{S}$, where $n \geq 2$, and let $N \subseteq \mathcal{D}_{S_1,...,S_n}$. The *minus-product* of $S_1, ..., S_n$ with respect to N is defined as follows:

$$\ominus_N(S_1, ..., S_n) = Br(S_1 \oplus ... \oplus S_n - Nr(N)) \cap \mathcal{D}_{S_1,...,S_n}. \ \Box$$

This operation results in a compound terminology consisting of all compound terms in $\mathcal{D}_{S_1,...,S_n}$, which are broader than a compound term in $S_1 \oplus ... \oplus S_n - Nr(N)$. This is because, all compound terms in $Nr(N)$ are invalid. Assuming a closed-world assumption over $S_1 \oplus ... \oplus S_n$, all compound terms in $S_1 \oplus ... \oplus S_n - Nr(N)$ are considered valid. Therefore, all compound terms in $Br(S_1 \oplus ... \oplus S_n - Nr(N))$ are also valid. We delimit this set to $\mathcal{D}_{S_1,...,S_n}$, as we are interested only in the compound terms, formed by terms appearing in $S_1, ..., S_n$.

It is easy to see that: (i) the operation minus-product is commutative, (ii) the larger the parameter N, the smaller the resulting compound terminology, and (iii) for any parameter N, we need to consider only its maximal (with respect to $\leq$) elements. The last property can be used for optimization, i.e., for minimizing the space needed for storing the parameter N.

Let T_i be a basic compound terminology. Intuitively, the *minus-self-product* operation $\overset{*}{\ominus}_N (T_i)$ specifies which compound terms in $\mathcal{P}(\mathcal{T}_i)$ are invalid, through a declared set of invalid compound terms $N \subseteq \mathcal{P}(\mathcal{T}_i)$.

Definition 4. Let T_i be a basic compound terminology and $N \subseteq \mathcal{P}(\mathcal{T}_i)$. The *minus-self-product* of T_i with respect to N is defined as follows: $\overset{*}{\ominus}_N (T_i) = \mathcal{P}(\mathcal{T}_i) - Nr(N)$. $\Box$

The minus-self-product operation of IFCA coincides with the minus-self-product operation of CTCA.

For defining the desired compound taxonomy, the designer has to formulate an IFCA expression e, defined as follows:

Definition 5 (IFCA expression). An IFCA expression over an interrelated faceted taxonomy $\mathcal{F} = (\mathcal{T}, \leq)$, generated by a set of facets $\{F_1, ..., F_k\}$ and a relation $<_F$, is defined according to the following grammar:

$$e ::= \ \oplus_P(e, ..., e) \mid \ominus_N (e, ..., e) \mid \overset{*}{\ominus}_N (T_i) \mid T_i. \quad \Box$$

The outcome of the evaluation of an expression e is denoted by S_e and is called the *compound terminology* of e. In addition, $(S_e, \preceq)$ is called the *compound taxonomy* of e. If e is the final expression that characterizes an interrelated faceted taxonomy $\mathcal{F} = (\mathcal{T}, \leq)$, the compound terms in S_e are considered *valid*[11] and the compound terms in $\mathcal{P}(\mathcal{T}) - S_e$ are considered *invalid*. We are especially interested in *well-formed* IFCA expressions, defined as follows:

[11] Obviously, in this case $Br(S_e) = S_e$.

Definition 6 (Well-formed expression). An IFCA expression e over an interrelated faceted taxonomy $\mathcal{F}$ is *well-formed* iff:

1. each basic compound terminology T_i appears at most once in e,
2. for every subexpression $\ominus_N(e_1, ..., e_n)$ of e, it holds: (i) $Nr(N) \cap S_{e_i} = \emptyset$, for all $i = 1, ..., n$, and (ii) $Nr(N) \cap S_e = \emptyset$, and
3. for every subexpression $\overset{*}{\ominus}_N (T_i)$ of e, it holds: (i) $Nr(N) \cap T_i = \emptyset$ and (ii) $Nr(N) \cap S_e = \emptyset$. □

Constraint (1) above is applied for simplifying IFCA expressions and improving the performance of our algorithms. This constraint is also imposed to well-formed CTCA expressions. Constraints (2.i) and (3.i) ensure that the valid compound terms of an expression e increase as e expands (see Proposition 2). For example, if we omit constraint (2.i) then a valid compound term according to an expression $T_1 \oplus_P T_2$ could be invalid according to a larger expression $(T_1 \oplus_P T_2) \ominus_N T_1$. Let N be the parameter of a minus-product or minus-self-product subexpression of e. Constraints (2.ii) and (3.ii) ensure that every compound term in $Nr(N)$ will not be found to be valid from another operation in e.

Proposition 2 (Monotonicity). Let $\mathcal{F}$ be an interrelated faceted taxonomy. If e is a well-formed IFCA expression and e' is a subexpression of e then $S_{e'} \subseteq S_e$.

The monotonicity property of well-formed IFCA expressions enables the specification of the valid compound terms of an interrelated faceted taxonomy, in a systematic and gradual manner. Additionally, from the monotonicity property, it follows that if an IFCA expression is well-formed then all subexpressions of e are also well-formed.

The following proposition expresses that IFCA is also more size-efficient than CTCA. We define the *parameter size* of an expression e as: $size(e) = |\mathcal{P}_e \cup \mathcal{N}_e|$, where $\mathcal{P}_e$ denotes the union of all P parameters of e, and $\mathcal{N}_e$ denotes the union of all N parameters of e.

Proposition 3 (Size-efficiency). Let $\mathcal{F}$ be an interrelated faceted taxonomy, generated by a set of facets $\{F_1, ..., F_k\}$ and a relation $<_{\mathtt{F}}$. Additionally, let $\mathcal{F}'$ be the simple faceted taxonomy, generated by the facets $\{F_1, ..., F_k\}$ (with the relation $<_{\mathtt{F}}$ ignored). Then, for every well-formed CTCA expression e' over $\mathcal{F}'$, there is a well-formed IFCA expression e over $\mathcal{F}$ that (i) has smaller or equal parameter size with e' and (ii) $S_e = S_{e'}$.

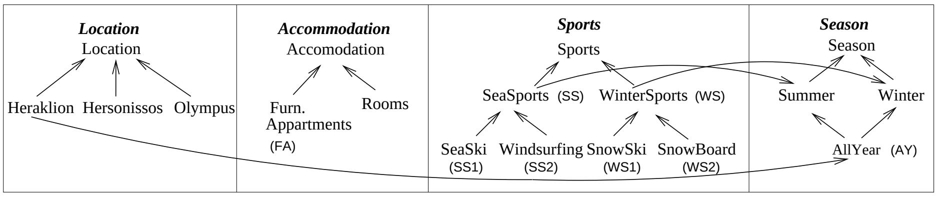

Fig. 2. An interrelated faceted taxonomy

The following proposition shows that a property, similar to that of Proposition 1 for plus-products, also holds for the minus-products of well-formed IFCA expressions.

Proposition 4. Let $\mathcal{F}$ be an interrelated faceted taxonomy. Additionally, let $e' = (\ominus_{N_1}(e_1, ..., e_l)) \ominus_{N_2} (\ominus_{N_3}(e_{l+1}, ..., e_n))$ be a subexpression of a well-formed IFCA expression e over $\mathcal{F}$. It holds: $S_{e'} = \ominus_{maximal_{\leq}(N_1 \cup N_2 \cup N_3)}(S_{e_1}, ..., S_{e_n})$.

As an example of IFCA, suppose that we want to index a set of hotel Web pages, according the *location* of the hotels, the kind of *accommodation*, the *facilities* they offer, and the *season* they are open. Assume now that the designer employs the interrelated faceted taxonomy $\mathcal{F}$, shown in Figure 2. From all possible compound terms, available domain knowledge suggests that only certain compound terms are valid. Omitting the compound terms which are singletons or contain top terms of the facets, and considering from the equivalent compound terms only one, 52 valid compound terms remain.

Rather than being explicitly enumerated, these compound terms can be algebraically specified. For example, the following plus-product operation can be used:

$\oplus_P(Location, Accommodation, Sports, Season)$, where:

$$\begin{aligned} P = \ & \{\{Heraklion, FA\}, \{Heraklion, Rooms\}, \{Hersonissos, FA, SS1\}, \\ & \{Hersonissos, FA, SS2\}, \{Hersonissos, Rooms, SS1\}, \{Hersonissos, Rooms, SS2\}, \\ & \{Olympus, FA, WS1\}, \{Olympus, FA, WS2\}, \{Olympus, Rooms, WS1\}, \\ & \{Olympus, Rooms, WS2\}, \{Olympus, Rooms, AllYear\}\} \end{aligned}$$

Note that the compound terms in P are 11. Alternatively, the same result can be obtained by the shorter minus-product operation:

$\ominus_N(Location, Accommodation, Sports, Season)$, where:

$$\begin{aligned} N = \ & \{\{Heraklion, Sports\}, \{Hersonissos, Winter\}, \{Olympus, SS\}, \\ & \{Olympus, FA, Summer\}, \{SS, Winter\}, \{WS, Summer\}\} \end{aligned}$$

The following, even shorter, IFCA expression e achieves the same result by combining the operations *plus-product* and *minus-product*:

$e = \ominus_N(Location, Accommodation, Sports) \oplus_P (Season)$, where:

$$\begin{aligned} N = \ & \{\{Heraklion, Sports\}, \{Hersonissos, WS\}, \{Olympus, SS\}\} \\ P = \ & \{\{Olympus, Rooms, AllYear\}\} \end{aligned}$$

This algebraic expression e will be our running example (well-formed) IFCA expression. We want to note that if $<_{\mathtt{F}}$ is ignored, the parameter size of the shortest CTCA expression e' such that $S_{e'} = S_e$ is 8 (see Proposition 3).

4 Checking Compound Term Validity

Below, we present an algorithm $IsValid_{\mathtt{I}}(e, s)$ which takes as input a well-formed IFCA expression e over an interrelated faceted taxonomy $\mathcal{F}$=($\mathcal{T}$, $\leq$) and a compound term $s \subseteq \mathcal{T}$, and returns TRUE, if $s \in S_e$, or FALSE, otherwise (i.e., if $s \notin S_e$). As it is shown in the explanations of the algorithm, $\mathit{IsValid}_{\mathtt{I}}(e, s)$ is optimized w.r.t. the naive approach.

Before we present the algorithm, we provide a few notations and definitions. Let $\mathcal{F}$=($\mathcal{T}$, $\leq$) be an interrelated faceted taxonomy, generated by a set of facets $\{F_1, ..., F_k\}$ and $<_{\mathtt{F}}$. Additionally, let e be an IFCA expression over $\mathcal{F}$. The *facets* of e are defined as: $F(e) = \{F_i \mid F_i \text{ appears in } e\}$. Clearly, $F(e) \subseteq \{F_1, ..., F_k\}$. We shall denote by $F(t)$ the facet to which a term $t \in \mathcal{T}$ belongs, e.g., in Figure 2, we have $F(\mathit{SeaSki}) = \mathit{Sports}$. Moreover, if $s \subseteq \mathcal{T}$, we define $F(s) = \bigcup\{F(t) \mid t \in s\}$.

Let $t, t' \in \mathcal{T}$ and let e be an IFCA expression over $\mathcal{F}$. We define: $t' <^{e}_{\mathtt{F}} t$ iff t' is a *maximal* (w.r.t. $\leq$) term such that: $t' \leq t$, $F(t') \in F(e)$, and $F(t') \neq F(t)$. For example, let e' be the first subexpression of our running example IFCA expression e. Then, $\mathit{WinterSports} <^{e'}_{\mathtt{F}} \mathit{Winter}$, while $\mathit{SnowSki} \not<^{e'}_{\mathtt{F}} \mathit{Winter}$ (note that $\mathit{SnowSki} \leq \mathit{WinterSports}$).

Algorithm 41 $\mathit{IsValid}_{\mathtt{I}}(e, s)$
Input: A well-formed IFCA expression e and a compound term $s = \{t_1, ..., t_m\} \subseteq \mathcal{T}$
Output: TRUE, if s belongs to S_e, or FALSE, otherwise

```
(1)   If $s = \emptyset$ then return(TRUE);
(2)   If $F(s) \not\subseteq F(e)$ then return(FALSE);
(3)   If $s$ is singleton then return(TRUE);
(4)   Case($e$)  {   / * Check the parse tree of $e$ */
(5)       $\oplus_P(e_1, ..., e_n)$:
(6)           If $\exists\, p \in P$ such that $p \preceq s$ then return(TRUE);
(7)           For $i = 1, ..., n$ do {
(8)               Let $S' = \{\{t'_1, ...., t'_m\} \mid (t'_j = t_j$ and $F(t_j) \in F(e_i))$ or $(t'_j <^{e_i}_{\mathtt{F}} t_j$ and
                                        $F(t_j) \notin F(e_i))$, for $j = 1, ..., m\}$;
(9)               For all $s' \in S'$ do {     /* Note that $s' \in \mathtt{Nr}(s)$ */
(10)                  If $\mathit{IsValid}_{\mathtt{I}}(e_i, s')$=TRUE then return(TRUE);
                          /* Note that $s' \in S_{e_i}$. Thus, $s \in S_e$ */
                  }    /* End For */
              }    /* End For */
(11)      $\ominus_N(e_1, ..., e_n)$:
(12)          If $\exists\, n \in N$ such that $s \preceq n$ then return(FALSE);
(13)          Let $S' = \{\{t'_1, ...., t'_m\} \mid t'_j = t_j$ or $\exists i \in \{1, ..., n\}$ s.t. $(t'_j <^{e_i}_{\mathtt{F}} t_j$ and
                                    $F(t_j) \notin F(e_i))$, for $j = 1, ..., m\}$;
(14)          For all $s' \in S'$ do {     /* Note that $s' \in \mathtt{Nr}(s)$ */
(15)              Let $\langle s'_1, ..., s'_n\rangle$ be the partition of $s'$ s.t. $F(s'_i) \subseteq F(e_i)$, for $i = 1, ..., n$;
(16)              $i = 1$;
(17)              $flag$ =TRUE;
(18)              While $flag$=TRUE and $i \leq n$ do {
(19)                  If $\mathit{IsValid}_{\mathtt{I}}(e_i, s'_i)$=FALSE then $flag$ =FALSE;
(20)                  $i = i + 1$;
                  }   /* End While */
(21)              If $flag$ =TRUE then return(TRUE); /* $s' \in S_e$. Thus, $s \in S_e$ */
              }    /* End For */
(22)      $\overset{*}{\ominus}_N$ $(T_i)$: If $\exists\, n \in N$ such that $s \preceq n$ then return(FALSE) else return(TRUE);
(23)      $T_i$: If $\exists\, t \in \mathcal{T}_i$ such that $\{t\} \preceq s$ then return(TRUE);    /* $s \in T_i \subseteq S_e$ */
      }   /* End Case */
(24)  return(FALSE);
```

The algorithm $\mathit{IsValid}_{\mathtt{I}}(e, s)$ for a well-formed IFCA expression e and $s = \{t_1, ..., t_m\} \subseteq \mathcal{T}$ is based on the parse tree of the expression e.

- If $e = \oplus_P(e_1, ..., e_n)$ and $F(s) \subseteq F(e)$ then it is checked if it exists $p \in P$ such that $p \preceq s$ (Step 6). If this is the case then $\mathit{IsValid}_{\mathtt{I}}(e, s)$ returns TRUE. Obviously, in this case, $s \in Br(P) \subseteq \oplus_P(e_1, ..., e_n)$. Otherwise, $\mathit{IsValid}_{\mathtt{I}}(e_i, s')$ is called (Step 10), for all $i = 1, ..., n$, and $s' \in S'$, where $S' = \{\{t'_1, ..., t'_m\} \mid (t'_j = t_j$ and $F(t_j) \in F(e_i))$ or $(t'_j <^{e_i}_{\mathtt{F}} t_j$ and $F(t_j) \notin F(e_i))\}$ (Steps 7-8). It holds that $\forall s \in S'$, $s' \preceq s$. Note that Step 8 has been optimized, as in a naive approach all compound terms $s' \preceq s$ would had been considered for computing S'. If any of the $\mathit{IsValid}_{\mathtt{I}}(e_i, s')$ calls, for $i = 1, ..., n$, returns TRUE then $\mathit{IsValid}_{\mathtt{I}}(e, s)$ returns TRUE (Step 10). Obviously, in this case, $s \in Br(S_{e_1} \cup ... \cup S_{e_n}) \cap \mathcal{D}_{S_{e_1}, ..., S_{e_n}} \subseteq \oplus_P(e_1, ..., e_n)$.
- If $e = \ominus_N(e_1, ..., e_n)$ and $F(s) \subseteq F(e)$ then it is checked if it exists $n \in N$ such that $s \preceq n$ (Step 12). If this is the case then $\mathit{IsValid}_{\mathtt{I}}(e, s)$ returns FALSE. Obviously, in this case, $s \in Nr(N)$. Thus, $s \notin S_{e_1} \oplus ... \oplus S_{e_n} - Nr(N)$, and as e is well-formed, $s \notin \ominus_N(e_1, ..., e_n)$. Otherwise, the set $S' = \{\{t'_1, ..., t'_m\} \mid t'_j = t_j$ or $\exists i \in \{1, ..., n\}$ s.t. $(t'_j <^{e_i}_{\mathtt{F}} t_j$ and $F(t_j) \notin F(e_i))$, for $j = 1, ..., m\}$ is computed (Step 13). It holds that $\forall s \in S'$, $s' \preceq s$. Note that Step 13 has been optimized, as in a naive approach all compound terms $s' \preceq s$ would had been considered for computing S'. Then, for all $s' \in S'$, the partition[12] $\langle s'_1, ..., s'_n \rangle$ of s' such that $F(s'_i) \subseteq F(e_i)$, for $i = 1, ..., n$, is computed (Step 15). Then, $\mathit{IsValid}_{\mathtt{I}}(e_i, s'_i)$ is called (Step 19), for all $i = 1, ..., n$. If $\mathit{IsValid}_{\mathtt{I}}(e_i, s'_i)$ returns TRUE, for all $i = 1, ..., n$, then $\mathit{IsValid}_{\mathtt{I}}(e, s)$ returns TRUE. Obviously, in this case, $s \in Br(S_{e_1} \oplus ... \oplus S_{e_n}) \cap \mathcal{D}_{S_{e_1}, ..., S_{e_n}} - Nr(N)$. As e is well-formed, $s \in \ominus_N(e_1, ..., e_n)$.
- If $e = \overset{*}{\ominus}_N (T_i)$ and $F(s) = \{F_i\}$ then it is checked if it exists $n \in N$ such that $s \preceq n$ (Step 22). If this is the case then $\mathit{IsValid}_{\mathtt{I}}(e, s)$ returns FALSE. Obviously, in this case, $s \in Nr(N)$. Otherwise, $\mathit{IsValid}_{\mathtt{I}}(e, s)$ returns TRUE. Obviously, in this case $s \in \mathcal{P}(\mathcal{T}_i) - Nr(N)$.
- If $e = T_i$ and $F(s) = \{F_i\}$ then it is examined if it exists $t \in \mathcal{T}_i$ such that $\{t\} \preceq s$ (Step 23). Obviously, in this case, $s \in T_i = Br\{\{t\} \mid t \in \mathcal{T}_i\} \cap \mathcal{P}(\mathcal{T}_i)$.

Note that since the $\leq$ relation is a partial order, algorithm $\mathit{IsValid}_{\mathtt{I}}(e, s)$ always terminates.

Continuing our running example, note that it holds:
$\mathit{IsValid}_{\mathtt{I}}(e, \{\mathit{Olympus}, \mathit{FA}, \mathit{Winter}\})$ =TRUE.
The trace of this call is as follows:

[12] Since e is a well-formed IFCA expression, there is only one such partition. This is due to condition (1) of Def. 6 (Well-formed expression).

```
Call $IsValid_{\mathtt{I}}(\ominus_N(Location, Accommodation, Sports) \oplus_P (Season),$
                   $\{Olympus, FA, Winter\})$;
   It holds that $\nexists p \in P$ s.t. $p \preceq \{Olympus, FA, Winter\}$;
   Compute $S' = \{\{Olympus, FA, WS\}\}$;
   Call $IsValid_{\mathtt{I}}(\ominus_N(Location, Accommodation, Sports), \{Olympus, FA, WS\})$;
      It holds that $\nexists n \in N$ s.t. $\{Olympus, FA, WS\} \preceq n$;
      Compute $S' = \{\{Olympus, FA, WS\}\}$;
      Compute partition $\langle\{Olympus\}, \{FA\}, \{WS\}\rangle$ of $s' \in S'$;
      Call $IsValid_{\mathtt{I}}(Location, \{Olympus\})$;
         Return(TRUE);
      Call $IsValid_{\mathtt{I}}(Accommodation, \{FA\})$;
         Return(TRUE);
      Call $IsValid_{\mathtt{I}}(Sports, \{WS\})$;
         Return(TRUE);
      Return(TRUE);
   Return(TRUE);
```

Additionally, it holds: $IsValid_{\mathtt{I}}(e, \{Hersonissos, Winter\})$ =FALSE.
The trace of this call is as follows:

```
Call $IsValid_{\mathtt{I}}(\ominus_N(Location, Accommodation, Sports) \oplus_P (Season),$
                   $\{Hersonissos, Winter\})$;
   It holds that $\nexists p \in P$ s.t. $p \preceq \{Hersonissos, Winter\}$;
   Compute $S' = \{\{Hersonissos, WS\}\}$;
      /* Note that $F(\{Hersonissos, WS\}) = \{Location, Sports\}$ */
   Call $IsValid_{\mathtt{I}}(\ominus_N(Location, Accommodation, Sports), \{Hersonissos, WS\})$;
      It holds that $\exists n \in N$ s.t. $\{Hersonissos, WS\} \preceq n$;
      Return(FALSE);
   Compute $S' = \{\}$;      /* Note that $\nexists t' \in \mathcal{T}_{Season}$ s.t. $t' \leq Hersonissos$ */
      Return(FALSE);
   Return(FALSE);
```

To provide the worst-time complexity of $IsValid_{\mathtt{I}}(e, s)$, a few auxiliary definitions are needed.

Let e be a well-formed IFCA expression over an interrelated faceted taxonomy $\mathcal{F} = (\mathcal{T}, \leq)$ and let $s \subseteq\mathcal{T}$. We define: $d^e_s = max_{t\in s}(|\{t'' \in\mathcal{T} \mid t'' <^e_{\mathtt{F}} t', t' \leq t\}|)$. For our running example IFCA expression e and $s = \{Hersonissos, Winter\}$, it holds that $d^e_s = 1$, while $d^{Season}_s = 0$.

Finally, let $|s^{\mathtt{max}}_e|$ be the size of the largest compound term, appearing in a P or N parameter of e. For our running example IFCA expression e, $|s^{\mathtt{max}}_e| = 3$.

Proposition 5. Let e be a well-formed IFCA expression over an interrelated faceted taxonomy $\mathcal{F} = (\mathcal{T}, \leq)$ and let $s \subseteq\mathcal{T}$. The worst-time complexity of $IsValid_{\mathtt{I}}(e, s)$ is in: $O(|s|^{d^e_s+1} * |s^{\mathtt{max}}_e| * |\mathcal{T}|^2 * |\mathcal{P}_e \cup \mathcal{N}_e|)$.

In computing the worst-time complexity of $IsValid_{\mathtt{I}}(e, s)$, the component $|s| * |s^{\mathtt{max}}_e| * |\mathcal{T}|^2$ corresponds to the maximun-time needed to check $p \preceq s'$, for all $p \in\mathcal{P}_e$ and $s' \preceq n$, for all $n \in\mathcal{N}_e$, in lines (6), (12), and (22) of Algorithm 41, respectively. Note that $|s'| \leq |s|$. Additionally, the factor $|s|^{d^e_s}$ corresponds to the maximum number of times that $IsValid_{\mathtt{I}}(.)$ is called in lines (10) and

(19) of Algorithm 41. Specifically, the factor $|s|^{d_s^e}$ is due to lines (8) and (13) of Algorithm 41.

Note that: (i) the call $\mathit{IsValid}_{\mathtt{I}}(e, s)$ can replaced, for optimization reasons, by $\mathit{IsValid}_{\mathtt{I}}(e, \mathit{minimal}_{\leq}(s))$[13], (ii) if s contains only one term of each facet then $|s| \leq |F(e)|$, and (iii) if $<_{\mathtt{F}} = \emptyset$ then $d_s^e = 0$.

Let $\mathcal{F}$ be an interrelated faceted taxonomy with $<_{\mathtt{F}} = \emptyset$ and let e is a well-formed CTCA expression e over $\mathcal{F}$. Then, e can be mapped directly to a well-formed IFCA expression e' such that (i) $S_{e'} = S_e$ and (ii) the computational complexity of $\mathit{IsValid}_{\mathtt{I}}(e', s)$ coincides with the computational complexity of $\mathit{IsValid}(e, s)$ (this algorithm is provided in [15], for checking compound term validity, according to a well-formed CTCA expression e).

5 Concluding Remarks

Faceted taxonomies are used in marketplaces [11], e-government portals [9], publishing museum collections on the Semantic Web [5], browsing large data sets from mobile phones [6], and several other application domains. Interest in faceted taxonomies is also indicated by several projects, like SemWeb[14], SWED[15], and SIMILE[16].

In this paper, we generalized previous work and provided an algebra, called *Interrelated Facet Composition Algebra* (IFCA), for specifying the valid terms over a faceted taxonomy $\mathcal{F}$, whose facets may be interrelated (through narrower/broader relationships between their terms). An optimized (w.r.t. the naive approach) algorithm that checks compound term validity, according to a well-formed IFCA expression, and its complexity were also provided. In contrast to *Compound Term Composition Algebra* (CTCA) [15], IFCA supports narrower/broader relationships between the terms of the different facets, thus reducing the size of the desired algebraic expressions and the effort needed by the designer to build the desired algebraic expression. Additionally, considering $<_{\mathtt{F}}$ during the formulation of a well-formed IFCA expression, we avoid conceptual errors (i.e., that a compound term is incorrectly specified as valid/invalid) that may be introduced during the formulation of a (well-formed) CTCA expression. The complexity of the compound term validity algorithms of CTCA and IFCA coincide, in the case that $\mathcal{F}$ is a simple faceted taxonomy (i.e., an interrelated faceted taxonomy with $<_{\mathtt{F}} = \emptyset$).

Issues for further research include: (i) generalizing the supported framework such that the relation $\leq$ between the terms of $\mathcal{F}$ is allowed to include non-trivial cycles, (ii) devising an algorithm for deciding whether an IFCA expression e is well-formed, (iii) devising *mining* algorithms (similar to these for CTCA [13]) that, given a materialized interrelated faceted taxonomy $\mathcal{M}$, derive well-formed IFCA expressions, defining the extensionally valid compound terms of $\mathcal{M}$. Finally, we plan to implement our proposed IFCA framework.

[13] Obviously, $s \in S_e$ iff $\mathit{minimal}_{\leq}(s) \in S_e$.

[14] `http://www.seco.tkk.fi/projects/semweb/`

[15] `http://www.swed.org.uk/`

[16] `http://simile.mit.edu/`

References

1. A. Analyti and I. Pachoulakis. "Logic Programming Representation of the Compound Term Composition Algebra". *Fundamenta Informaticae*, 73(3):321–360, 2006.
2. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *"The Description Logic Handbook: Theory, Implementation, and Applications"*. Cambridge University Press, 2003.
3. B. Ganter and R. Wille. *"Formal Concept Analysis: Mathematical Foundations"*. Springer-Verlag, Heidelberg, 1999.
4. M. Hearst. "Design Recommendations for Hierarchical Faceted Search Interfaces". In *ACM SIGIR'2006 Workshop on Faceted Search*, pages 26–30, 2006.
5. E. Hyvönen, E. Mäkelä, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. "MUSEUMFINLAND - Finnish Museums on the Semantic Web". *Journal of Web Semantics*, 3(2-3):224–241, 2005.
6. A. K. Karlson, G. G. Robertson, D. C. Robbins, M. P. Czerwinski, and G. R. Smith. "FaThumb: a Facet-Based Interface for Mobile Search". In *Procs. of the SIGCHI conference on Human Factors in Computing Systems (CHI'06)*, pages 711–720, 2006.
7. J. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, second edition, 1987.
8. G. M. Sacco. "Dynamic Taxonomies: A Model for Large Information Bases". *IEEE Transactions on Knowledge and Data Engineering*, 12(3):468–479, May 2000.
9. G. M. Sacco. "Guided Interactive Information Access for E-Citizens". In *Procs. of the 4th Intern. Conf. on Electronic Government (EGOV-2005)*, pages 261–268, 2005.
10. G. M. Sacco. "Research Results in Dynamic Taxonomy and Faceted Search Systems". In *Procs. of the 1st International Workshop on Dynamic Taxonomies and Faceted Search (in conjunction with DEXA'07)*, pages 201–206. IEEE Computer Society, 2007.
11. I. Tofte, K. J. Sæth, and K. Jansson. "A case study of Vinmonopolet.no: faceted search and navigation for e-commerce". In *Procs. of the 4th Nordic Conference on Human-Computer Interaction (NordiCHI-2006)*, pages 489–490, 2006.
12. Y. Tzitzikas. "An Algebraic Method for Compressing Symbolic Data Tables". *Journal of Intelligent Data Analysis (IDA)*, 10(4):243–359, 2006.
13. Y. Tzitzikas and A. Analyti. "Mining the Meaningful Term Conjunctions from Materialised Faceted Taxonomies: Algorithms and Complexity". *Knowledge and Information Systems (KAIS)*, 9(4):430–467, 2006.
14. Y. Tzitzikas, A. Analyti, and N. Spyratos. "Compound Term Composition Algebra: The Semantics". *LNCS Journal on Data Semantics*, 2:58–84, 2005.
15. Y. Tzitzikas, A. Analyti, N. Spyratos, and P. Constantopoulos. "An Algebra for Specifying Valid Compound Terms in Faceted Taxonomies". *Data and Knowledge Engineering (DKE)*, 62(1):1–40, 2007.
16. Y. Tzitzikas, R. Launonen, M. Hakkarainen, P. Kohonen, T. Leppanen, E. Simpanen, H. Tornroos, P. Uusitalo, and P. Vanska. "FASTAXON: A system for FAST (and Faceted) TAXONomy design.". In *Procs. of 23th Int. Conf. on Conceptual Modeling (ER'2004)*, pages 841–843, 2004. (an on-line demo is available at `http://fastaxon.erve.vtt.fi/`).
17. K. Yee, K. Swearingen, K. Li, and M. Hearst. "Faceted Metadata for Image Search and Browsing". In *Proceedings of the Conf. on Human Factors in Computing Systems (CHI'03)*, pages 401–408, April 2003.