



# Faceted Taxonomy-based Information Management

Yannis Tzitzikas<sup>1,2</sup> and Anastasia Analyti<sup>2</sup>

1: Assistant Professor, Department of Computer Science, University of Crete

2: Associate Researcher, Institute of Computer Science (FORTH-ICS)

*FIND'2007: International Workshop on Dynamic Taxonomies and Faceted Search*

*Regensburg, Germany, September 3-7, 2007*



# Outline



- Introduction and Motivation
- Compound Term Composition Algebra (CTCA)
  - Introduction
  - Applications
- Taxonomy Evolution and CTCA
- Integration and Personalization of Taxonomy-based Sources
- Concluding Remarks



# Introduction and Motivation



Taxonomies are the probably the **oldest** and **most widely used** conceptual modeling tool.

*examples*

## Manually designed:

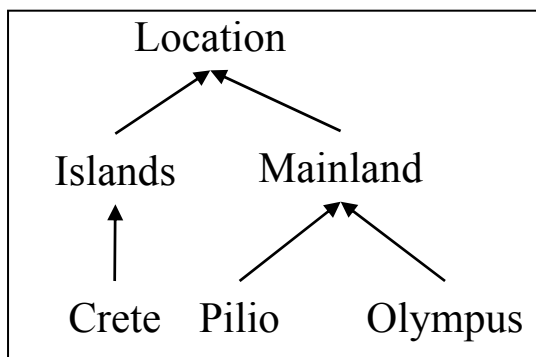
- Web Catalogues (like Yahoo!)
- Thesauri in Library Systems
- Personal Web Bookmarks
- Web Services
- ...

## Extracted/Mined:

- FCA concept lattices
- Ontology extraction
- ...

## Inferred:

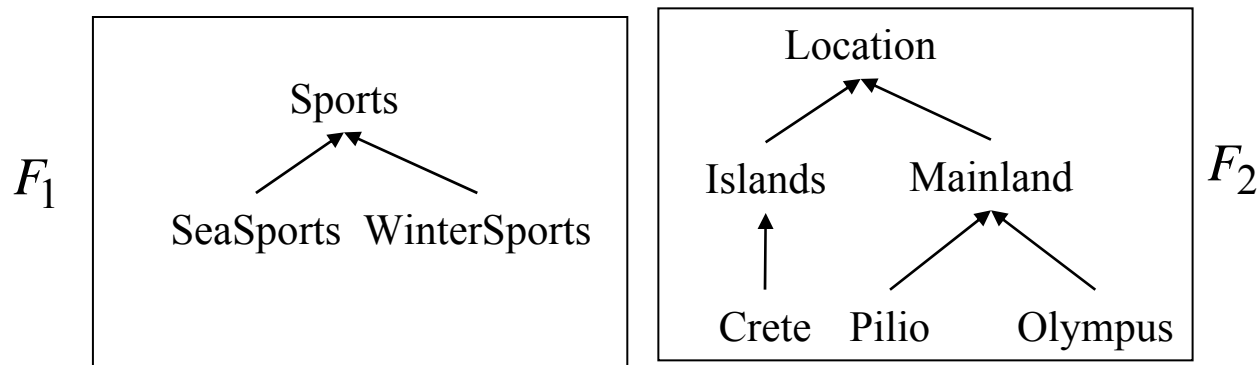
- Description Logics concept lattices
- CTCA compound taxonomies
- ...



A taxonomy is a pair  $(T, \leq)$  where  $T$  is a terminology and  $\leq$  is a preorder relation (reflexive and transitive) over  $T$



A faceted taxonomy is a set of taxonomies each describing the application domain from a different (preferably orthogonal) point of view.



A faceted taxonomy is a set of taxonomies.  
 $F = \{F_1, \dots, F_k\}$  where  $F_i = (T_i, \leq_i)$

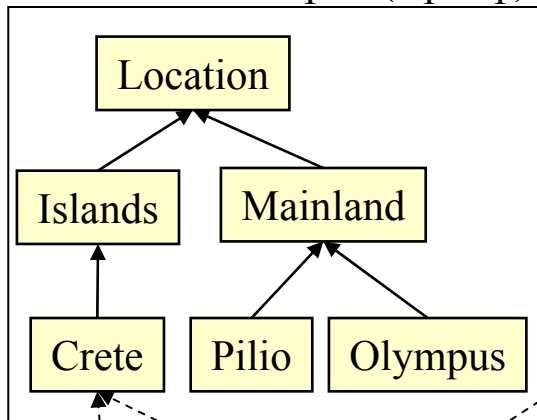
Objects are indexed by assigning to them one or more terms from each facet



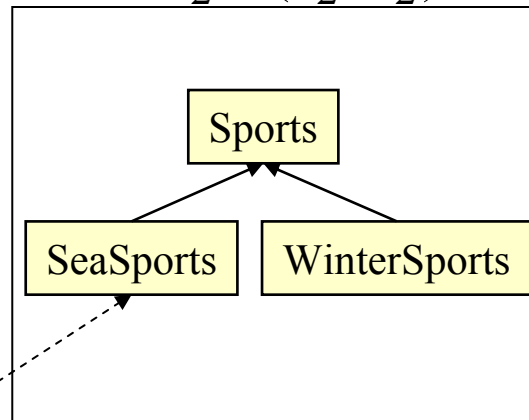
# Taxonomies, Faceted Taxonomies, Materialized Taxonomies

$$\text{Faceted taxonomy } F = \{F_1, F_2\}$$

**Taxonomy  $F_1 = (T_1, \leq_1)$**



**$F_2 = (T_2, \leq_2)$**



**Materialized  
(faceted)  
Taxonomy**

**Interpretation**

$$I: T \rightarrow P(Obj)$$



$$I(\text{Crete}) = \{\text{hotel1}, \text{hotel2}\}$$

$$I(\text{SeaSports}) = \{\text{hotel2}\}$$

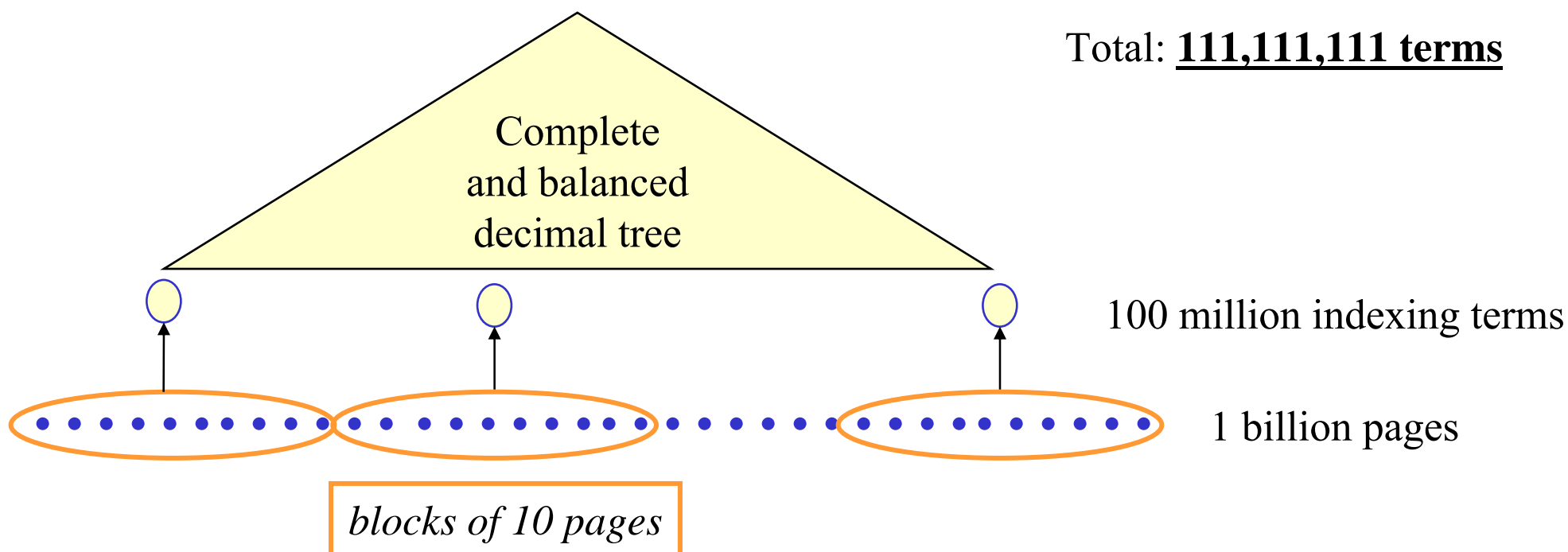
$$\text{index}(\text{hotel1}) = \{\text{Crete}\}$$

$$\text{index}(\text{hotel2}) = \{\text{Crete}, \text{SeaSports}\}$$

**compound term**

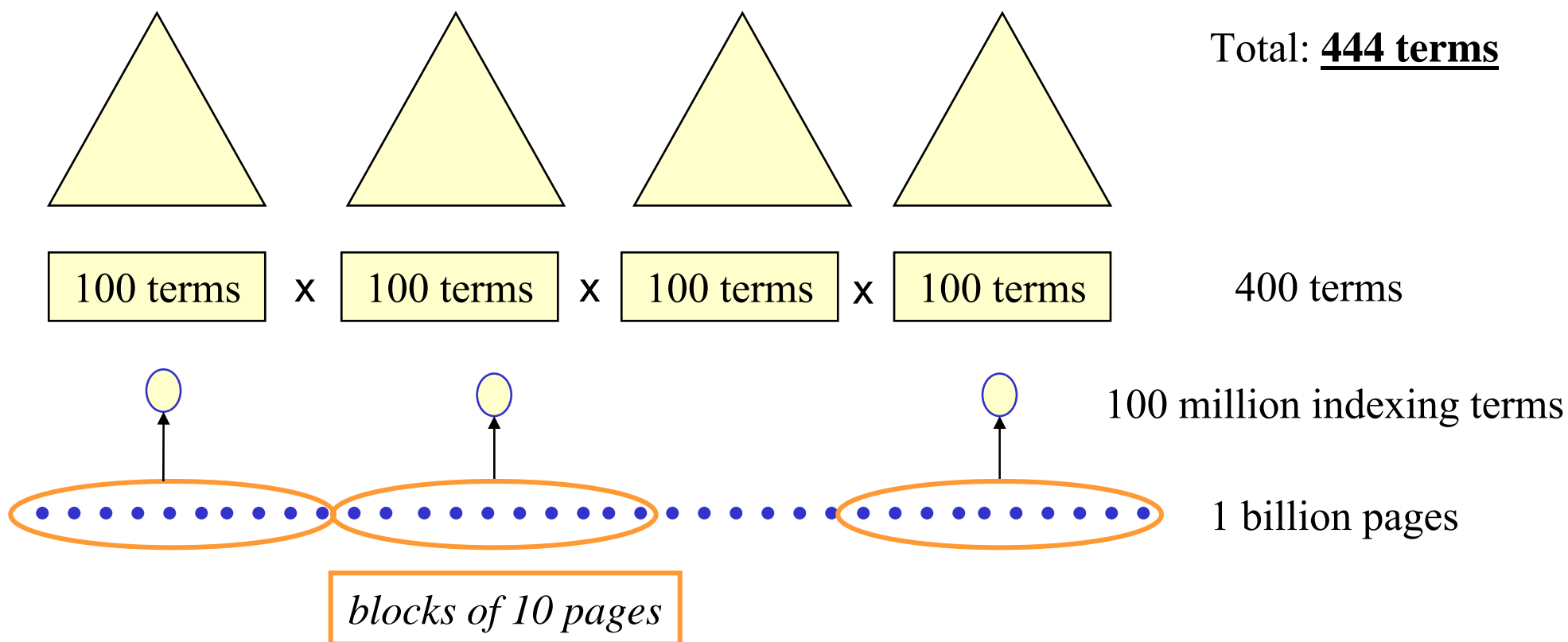


# Example of using one taxonomy



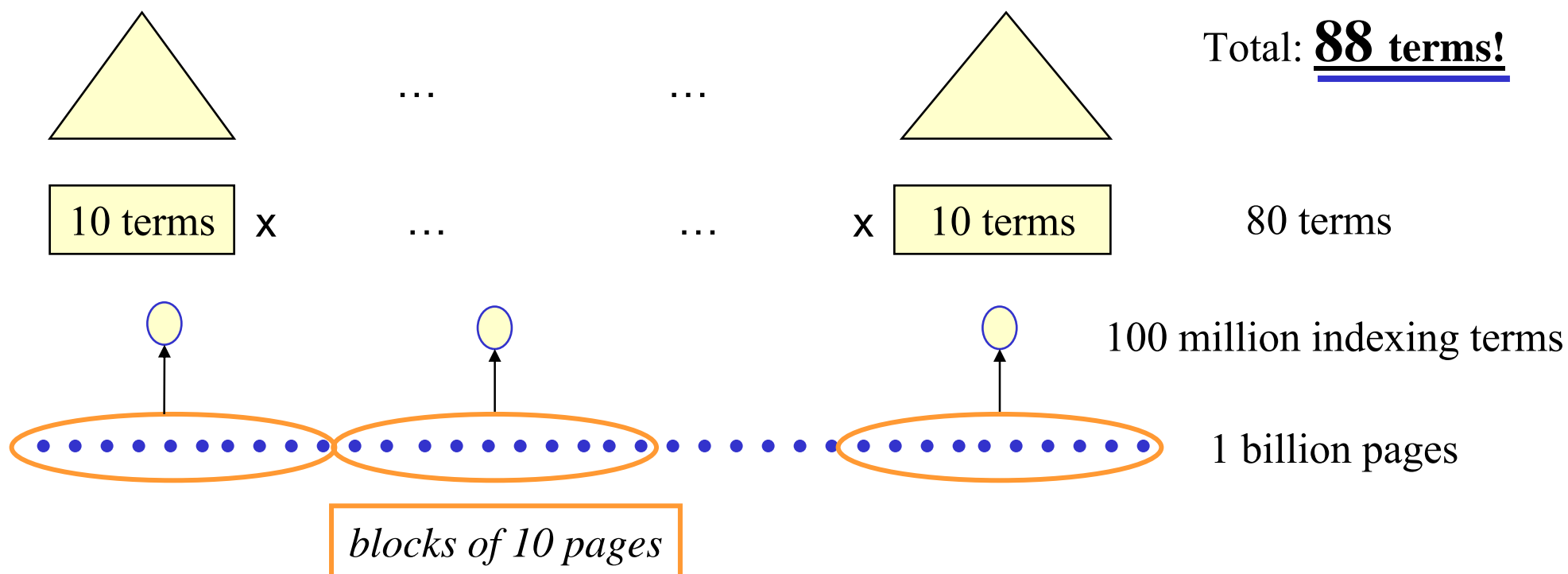


# Example of using a faceted taxonomy consisting of 4 facets



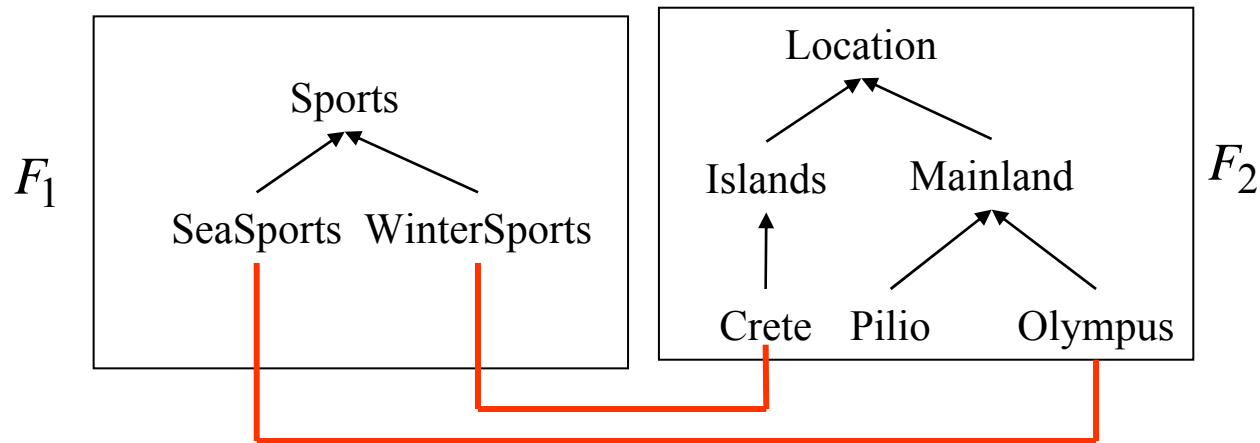


# Example of using a faceted taxonomy consisting of 8 facets





Invalid compound terms: conjunctions of terms that do not apply to any object of the domain.



- Invalid compound terms may affect the quality of indexing and browsing
- The specification of invalid compound terms involves considerable human effort



# Compound Term Composition Algebra

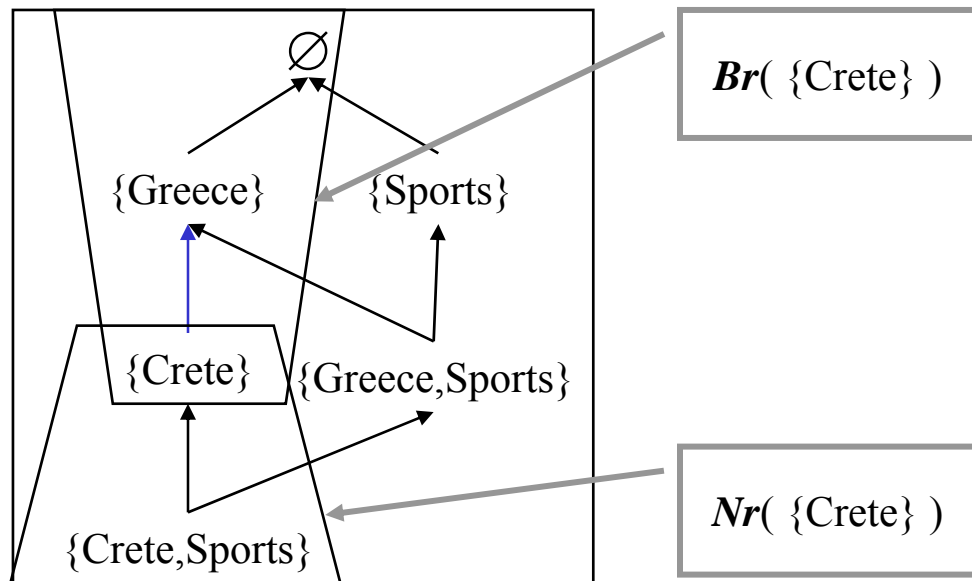


**Compound term:** Any subset of  $T$

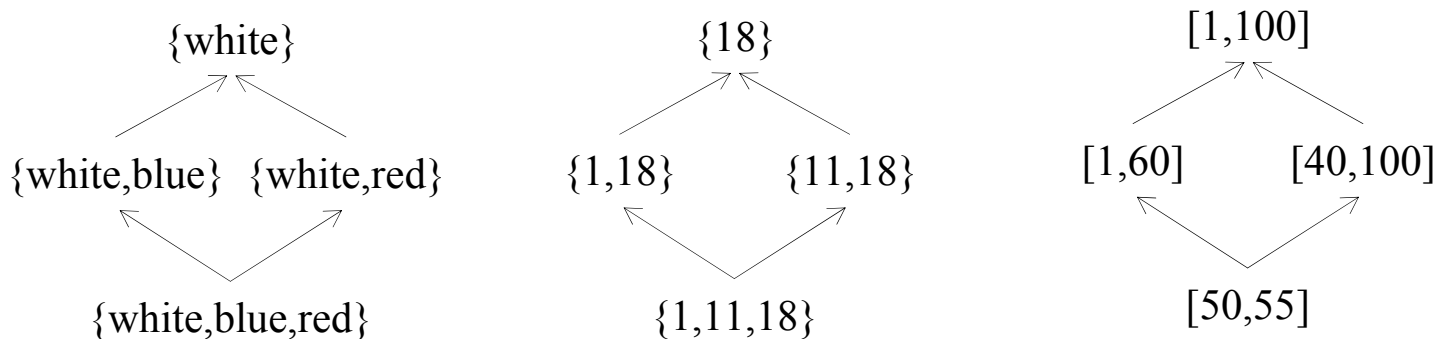
**Compound ordering** over a set of compound terms:

$$s \leq s' \text{ iff } \forall t' \in s' \exists t \in s \text{ such that } t \leq t'$$

Smyth's ordering



E.g. of compound orderings over unordered domains ( $\supseteq$ ):





CTCA is an algebra that can be used to specify the set of valid compound terms in an efficient and flexible manner.

It works on the basis of the original simple terms of the facets and a small set of **positive** and/or **negative** statements.

$\oplus$	<i>product</i>	returns <b>all</b> possible compound terms over $\geq 2$ facets	<i>n-ary</i>
$\oplus_P$	<b>plus</b> - <i>product</i>	takes as <u>parameter</u> a set of <b>valid</b> compound terms (over $\geq 2$ facets) from which <i>more</i> valid are then <i>inferred</i> and returned	<i>n-ary</i>
$\ominus_N$	<b>minus</b> - <i>product</i>	takes as <u>parameter</u> a set of <b>invalid</b> compound terms (over $\geq 2$ facets) from which <i>more</i> invalid are then <i>inferred</i> .	<i>n-ary</i>
$\oplus^*$	<b>self</b> - <i>product</i>	returns <b>all</b> possible combinations of terms from <b>one</b> facet	<i>unary</i>
$\oplus_P^*$	<b>self-plus</b> - <i>product</i>	takes as <u>parameter</u> a set of <b>valid</b> compound terms (over <b>one</b> facets) from which <i>more</i> valid are then <i>inferred</i> and returned	<i>unary</i>
$\ominus_N^*$	<b>self-minus</b> - <i>product</i>	takes as <u>parameter</u> a set of <b>invalid</b> compound terms (over <b>one</b> facets) from which <i>more</i> invalid are then <i>inferred</i> .	<i>unary</i>

In each algebraic operation, we adopt a closed-world assumption with respect to the declared positive or negative statements

*The semantics of CTCA are formally defined in [J. on Data Semantics, 2005] where it has been shown that CTCA cannot be efficiently represented in Description Logics*



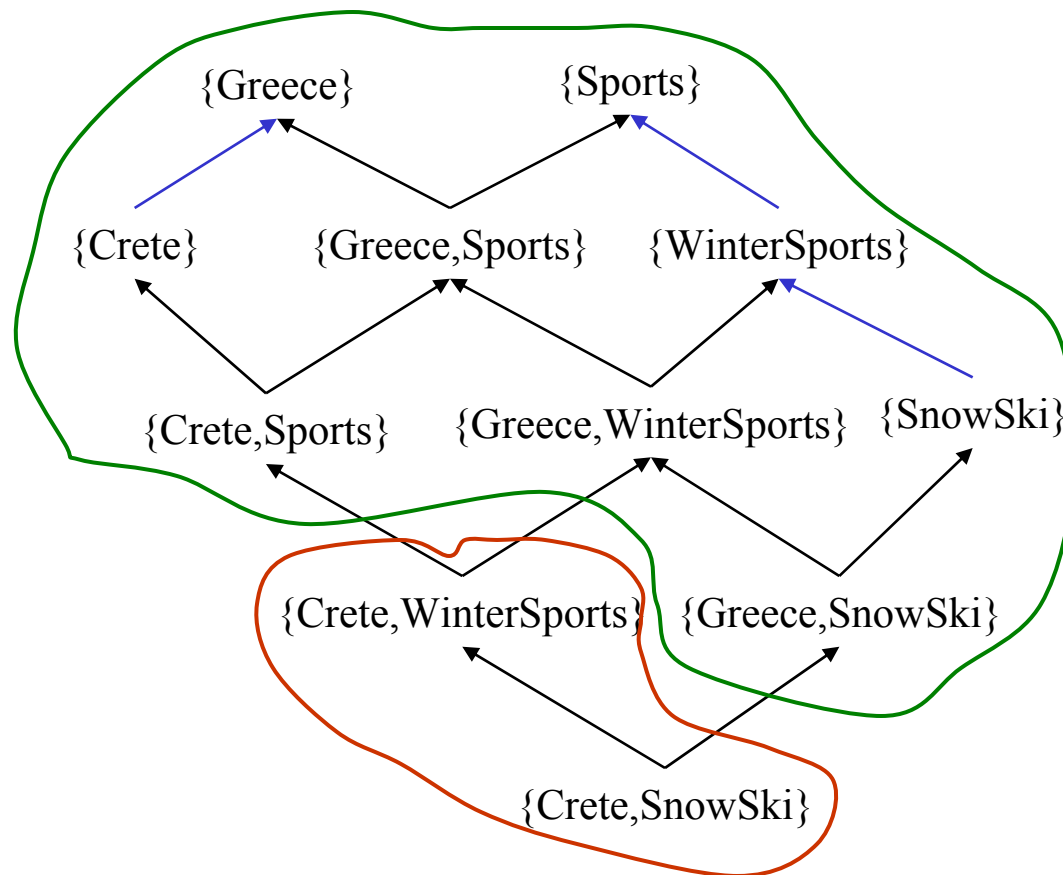
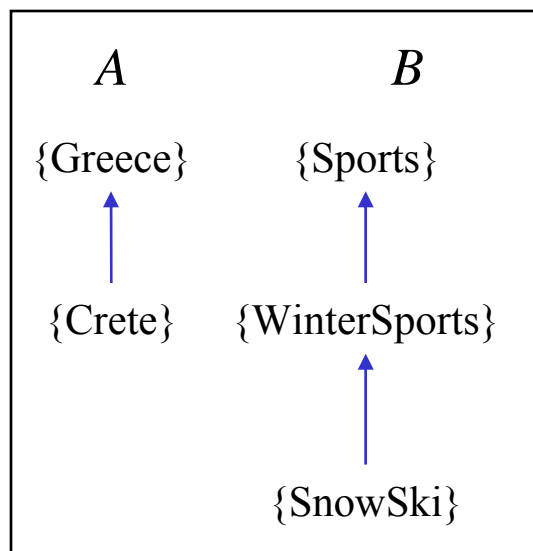
# CTCA: Example expressions

$A \oplus_{\mathbf{P}} B$  where  $\mathbf{P} = \emptyset$

$A \ominus_{\mathbf{N}} B$  where  $\mathbf{N} = \emptyset$

$A \oplus_{\mathbf{P}} B$  where  $\mathbf{P} = \{\{\text{Crete, Sports}\}, \{\text{Greece, SnowSki}\}\}$

$A \ominus_{\mathbf{N}} B$  where  $\mathbf{N} = \{\{\text{Crete, WinterSports}\}\}$

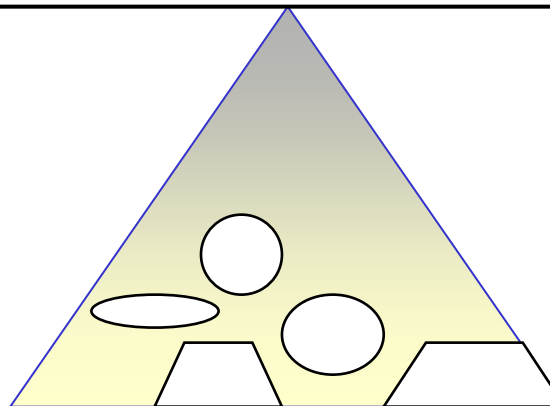


$(A \ominus_{\mathbf{N}} B) \oplus_{\mathbf{P}} C \dots$



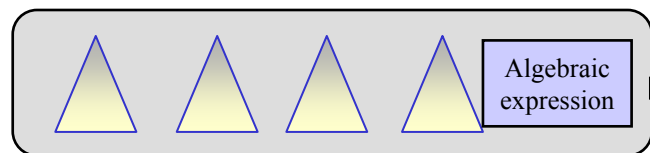
## Taxonomies of existing Web catalogs

big storage space,  
incomplete structure,  
scalability problems



## Faceted Taxonomies + CTC Algebra

small storage space,  
clear and complete structure,  
scalable



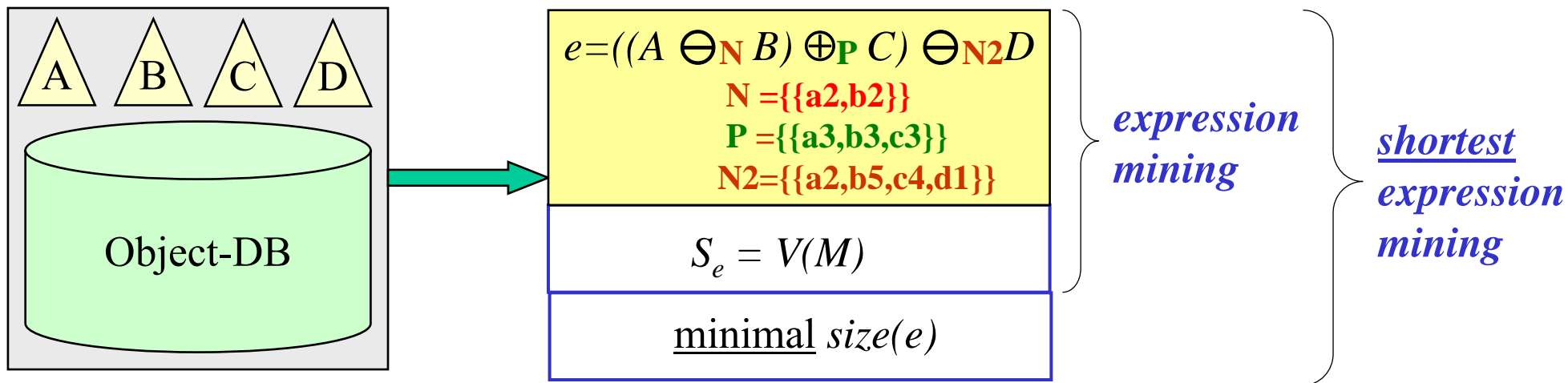
*dynamically*



Navigation Trees

# The reverse problem: (Shortest) Expression Mining

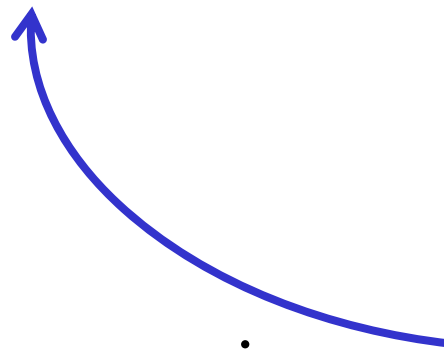
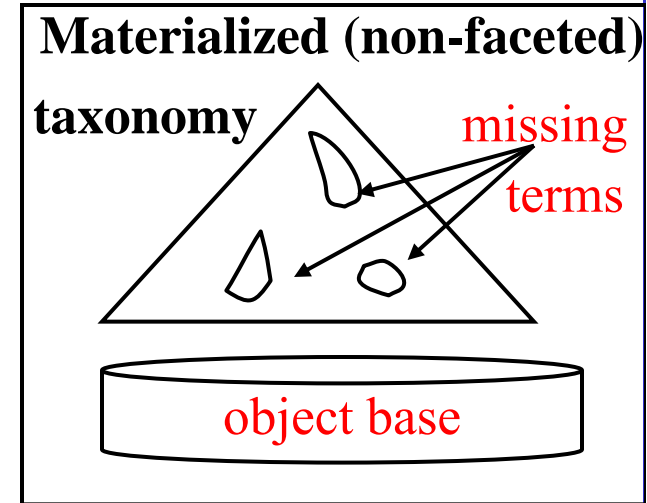
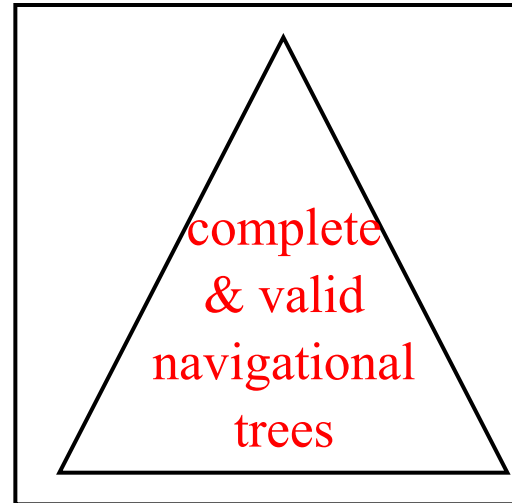
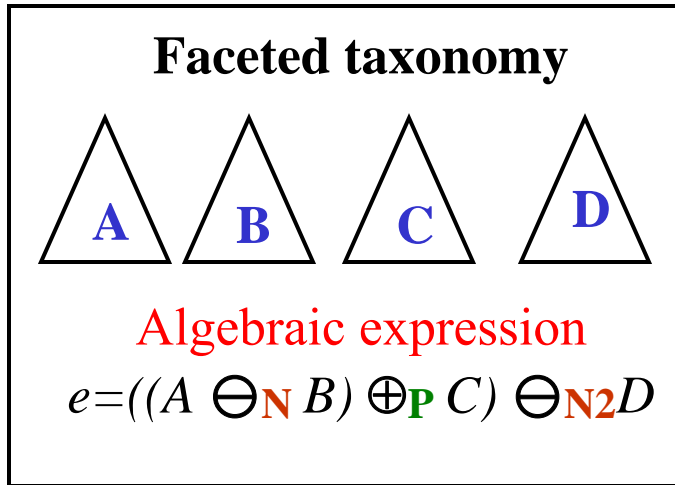
$M$ : materialized fac. taxonomy



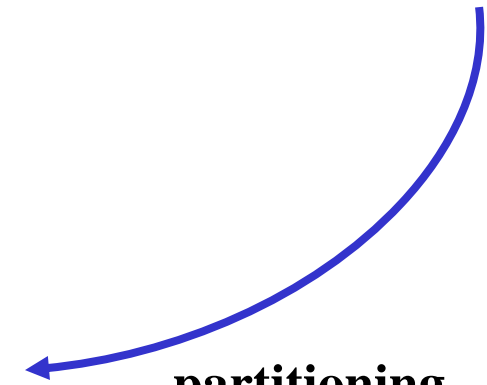
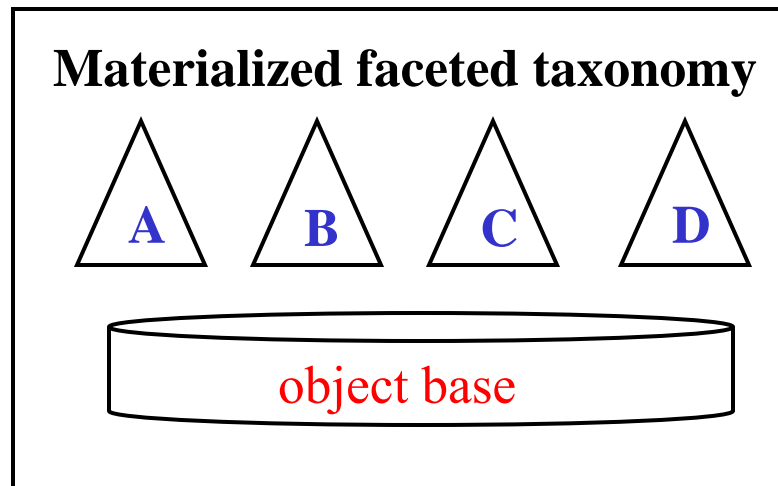
$V(M)$ : the **valid** compound terms of  $M$   
(those with non empty extension):

A	B	C	D
a1	b2	c1	d2
a2	b1	c3	d4
a3	b2	c3	d3
a5	b8	c1	d1
.	.	.	.
.	.	.	.
.	.	.	.
a4	b2	c3	d3
a2	b5	c2	d1
a5	b1	c4	d3

$S_e$  : the **compound terms** defined by an expression  $e$   
 $size(e)$  : the **size** of the parameters of  $e$



expression mining



partitioning terms to facets



# Evolution of CTCA-based Sources



## Scenario:

- Taxonomies evolve over time
  - additions/deletions/renamings of terms
  - additions/deletions of subsumption relationships
- An update operation on a faceted taxonomy  $F$  (resulting to an  $F'$ ) may turn the expression  $e$  not well-formed, or it may make the derived compound terminology  $S_e^{F'}$  to no longer reflect the desire of the designer, i.e. it may no longer reflect the domain knowledge that was expressed originally.

## Objective:

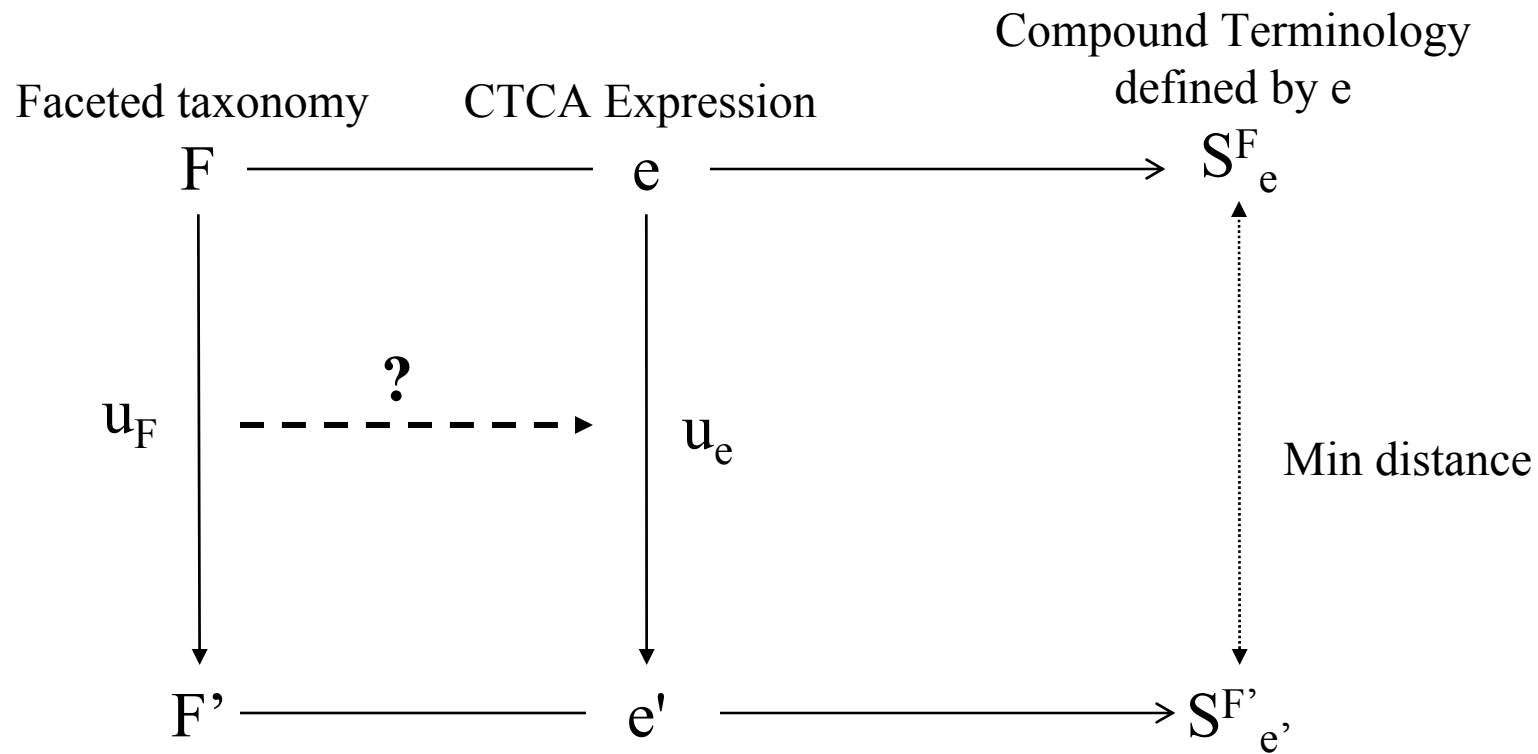
- Update automatically  $e$  to an expression  $e'$  that is (a) well-formed (w.r.t.  $F'$ ), and (b)  $S_{e'}^{F'}$  is as close to  $S_e^F$  as possible.

## Why this is useful?

- This would enhance the robustness and usability of systems that are based on CTCA, like FASTAXON.
  - Moreover it could be useful in other tasks where CTCA can be used like mining [J. KAIS'06] and compression [J. IDA'06]



# CTCA Expression Revision: Problem Statement

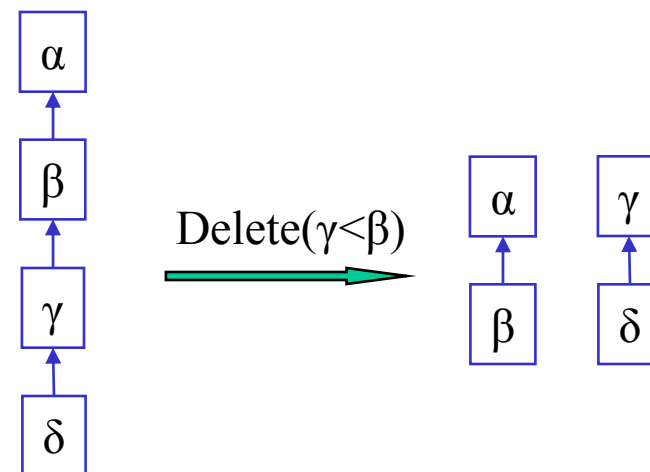
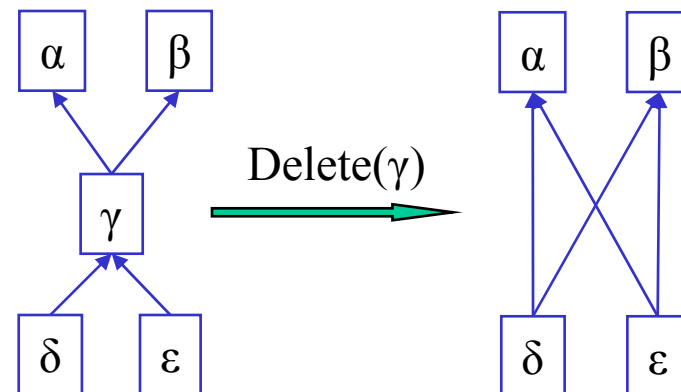




Let  $t$ ,  $t1$ ,  $t2$  denote terms.

We consider the following update operations:

- Add( $t$ )
- Rename( $t1, t2$ )
- Delete( $t$ )
  - postcondition: the parents of  $t$  are now connected with the children of  $t$
- Delete( $t1 < t2$ )
  - precondition:  $t1 < t2$  is member of the transitive reduction of  $\leq$
- Add( $t1 < t2$ )
  - precondition:  $t1 \leq t2$  does not hold





# Solving the Expression Revision Problem

- **Rename( $t_1$ ,  $t_2$ )**
  - Trivial
- **Add( $t$ )**
  - Solvable
    - We can always find an  $e'$  that preserves the original compound terms
  - We have to update the N parameters of minus product operations
- **Delete( $t$ )**
  - Solvable
    - We can preserve all compound terms except those that contain the deleted term  $t$
  - We have to update all P/N parameters that contain the term  $t$
- **Delete( $t_1 < t_2$ )**
  - Solvable
    - We can always find an  $e'$  that preserves the original compound terms
  - We have to extend the P/N parameters (so as to recover the missing compound terms from the semantics of the original expression)



- Add( $t_1 < t_2$ )
  - Difficult. **Not always solvable**
  - Reason: After the addition of a subsumption relationship we may no longer be able to separate (from the semantics) compound terms that were previously separable (i.e. compound terms which were not  $\leq$ -related before the addition of the subsumption link.
  - The combination of  $\oplus P$  and  $\ominus N$  operations can lead to cases where the resulting compound terminology may neither be subset nor superset of the original compound terminology.
    - The effects of adding a subsumption relationship is different in  $\oplus P$  and  $\ominus N$ : the compound terminologies defined by  $\oplus P$  operations become larger, while those defined by  $\ominus N$  operations become smaller.

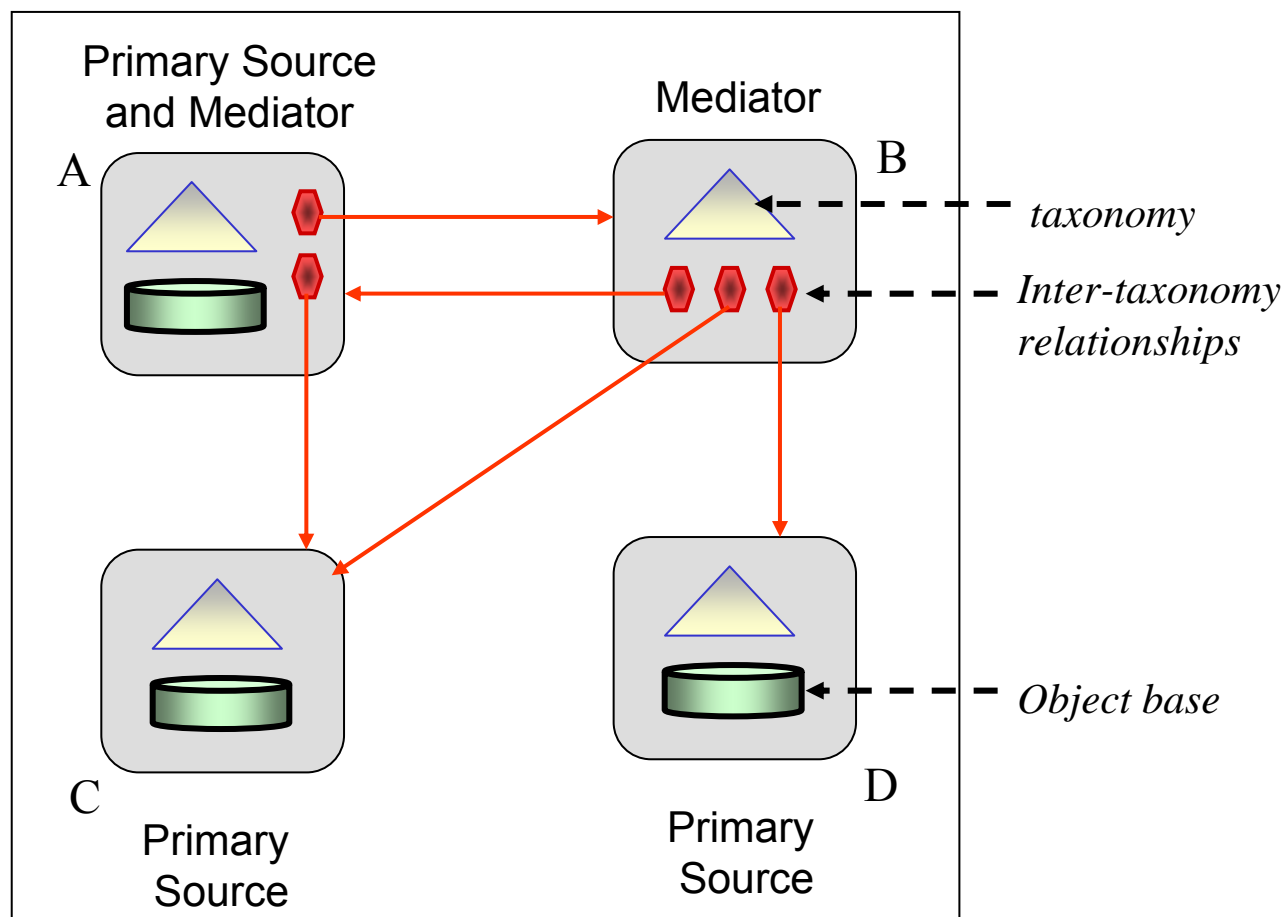


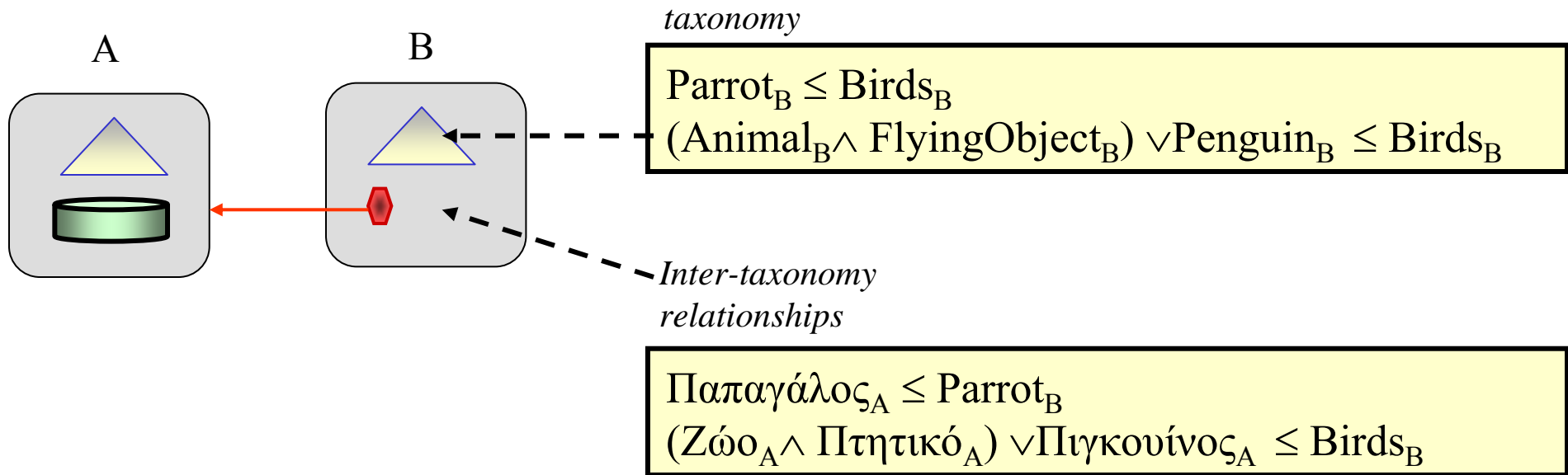
# Integration of Taxonomy-based Sources



## Objectives:

- Design **mediators** and **P2P** systems over this kind of sources
- Investigate **conceptual modeling** (integration) issues
- Derive **algorithms** for **query evaluation**







# Mediators and P2P Systems over Taxonomy-based Sources

FORTH-ICS



## Results:

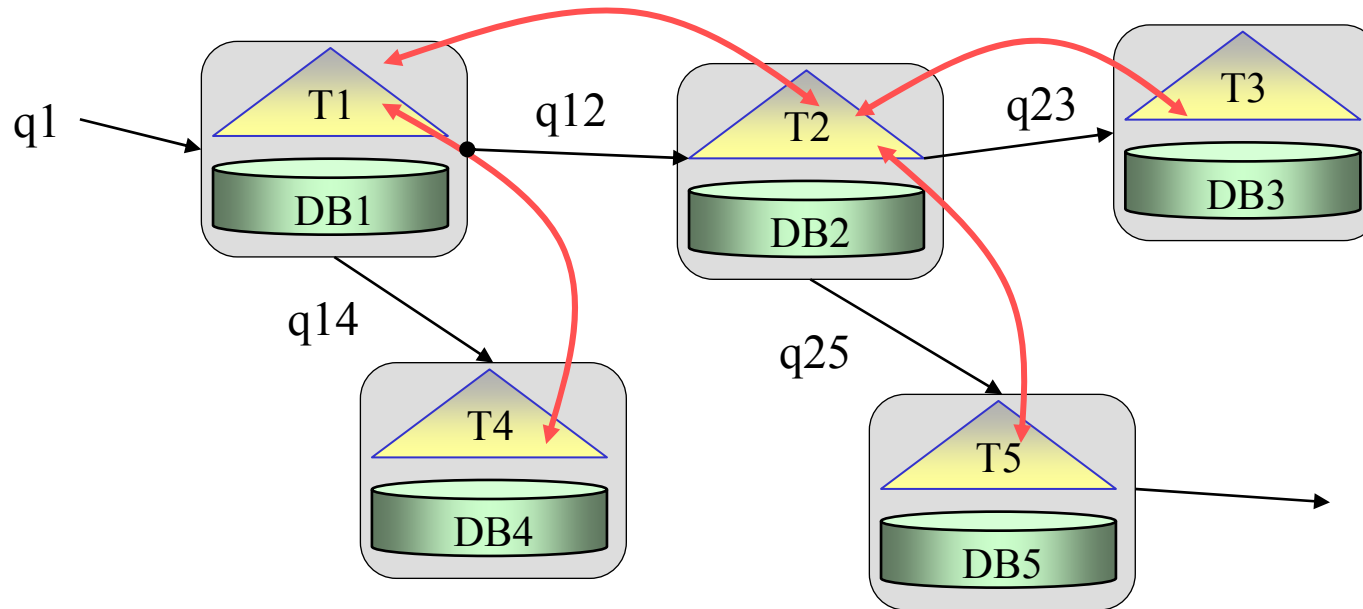
- An arsenal of **query evaluation algorithms** for **pure P2P systems** for sound and complete answers (*rewrite vs evaluate, single vs multiple controller*) and for various combinations of
  - *taxonomies* (of single terms or complex concepts),
  - *queries* (with negation or not), and
  - *mappings* (*term-2-term, term-2-query, query-2-query*)
- If negation appears in term-2-query mappings then query answering is coNP-Hard.
- The same holds even if we haven't negation but we have query-2-query mappings

## Future Research:

- The role of caches
- **VLDB J. 2005 (Taxonomy-based Mediators)**
- **ER'2003 (Tax-based Conc. Modeling 4 P2P)**
- **CoopIS'2003 (Q. Eval algorithms)**
- **ODBASE'2004 (Q. Eval complexity)**



# Automatic Mapping of Taxonomy-based Sources





# Automatic Taxonomy Mapping

FORTH-ICS



## Objectives:

- Find an automatic method for **constructing mappings** between taxonomies, s.t.:
- it is **independent** of the nature of the objects
  - can integrate the entire taxonomies **or** the **desired** subparts
  - can construct mappings also between **queries**
  - can be applied in a pure **P2P system** (can run by two players alone)
  - can be implemented **efficiently** by a communication protocol

## Results:

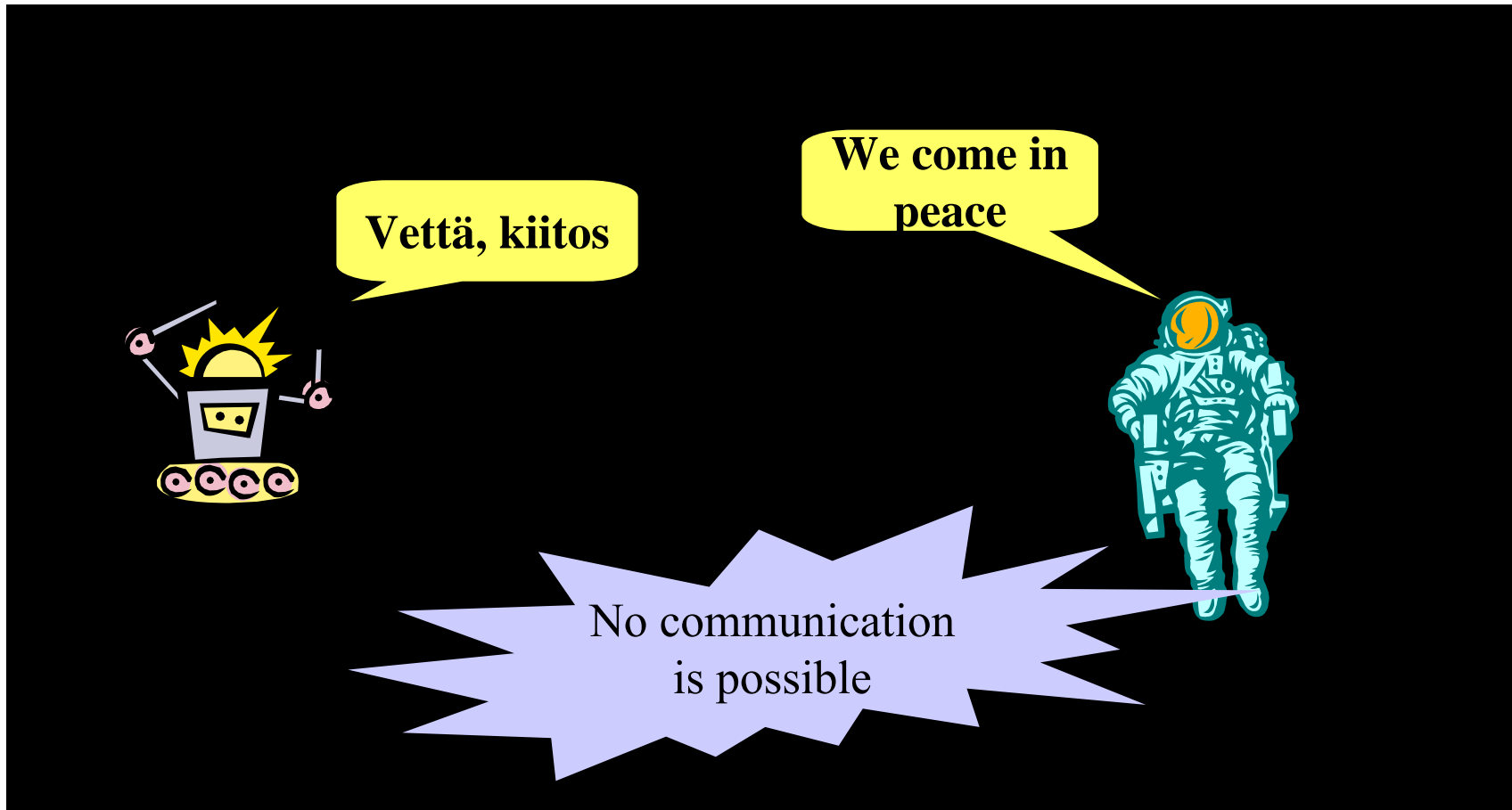
*The Ostensive Mapping Method*      **CIA'2003 (BEST PAPER AWARD)**

## Keypoints:

- The method is based on *approximate naming functions*.

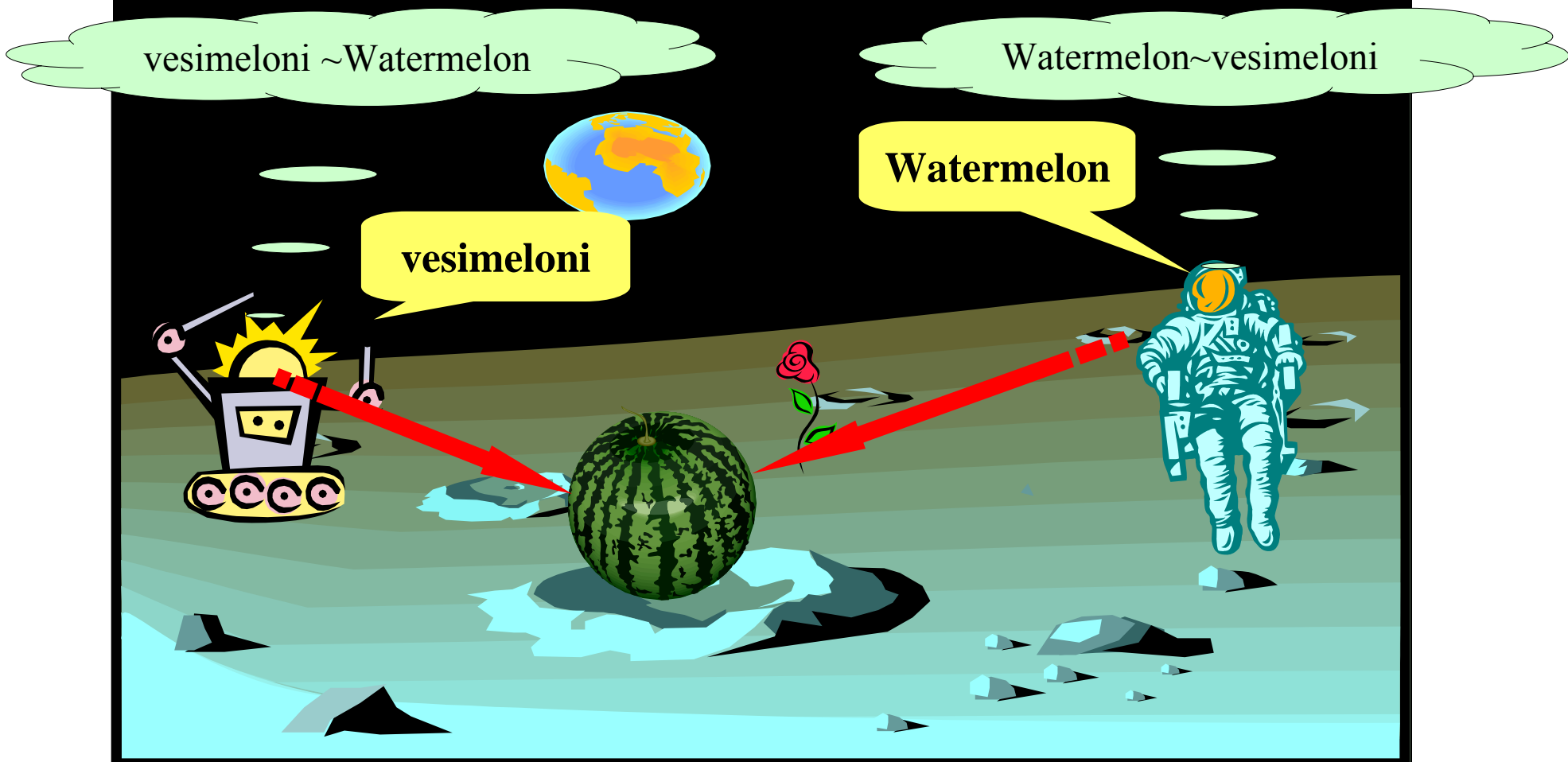
## Future Research:

- Application on other kinds of sources

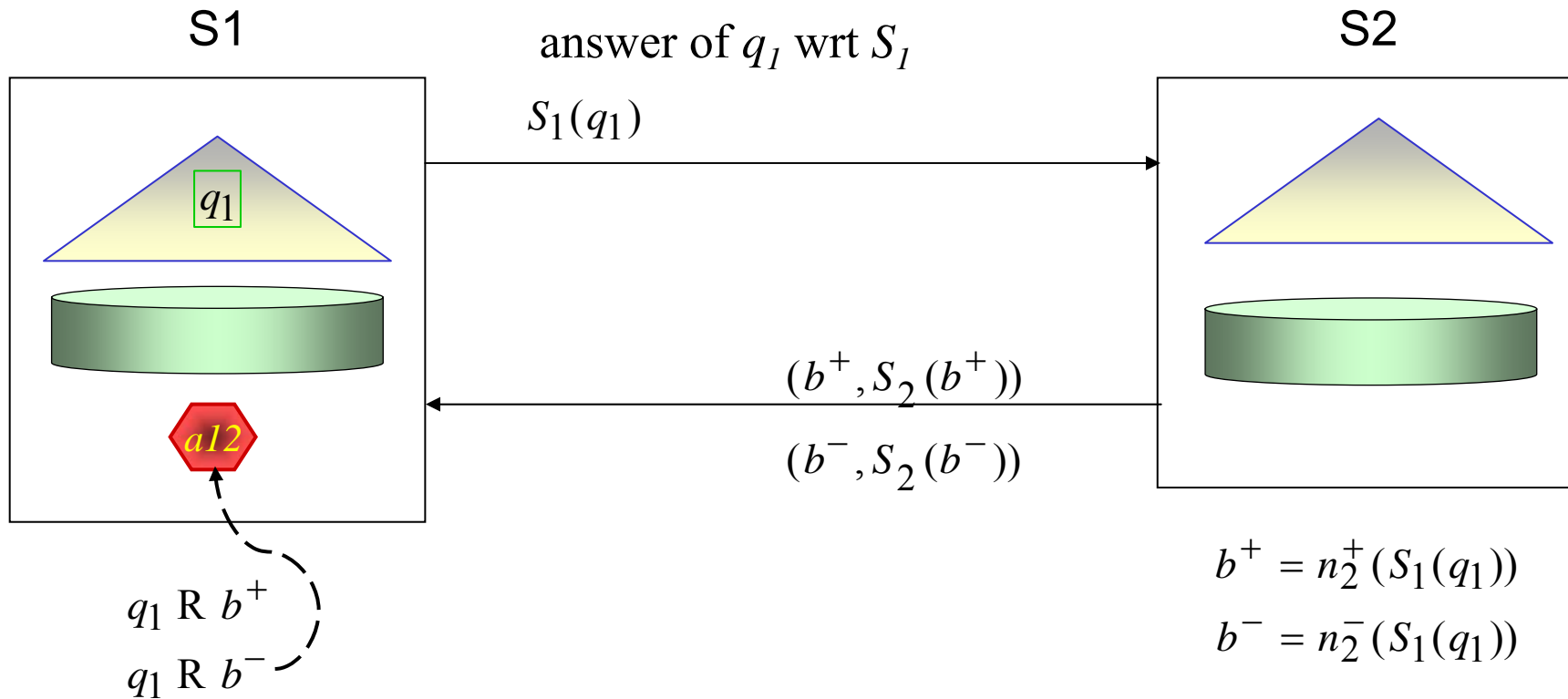




# The method



## The ostensive method



**Upper and Lower Name based on Approximate Naming Functions**

=> Only **two** messages have to be exchanged

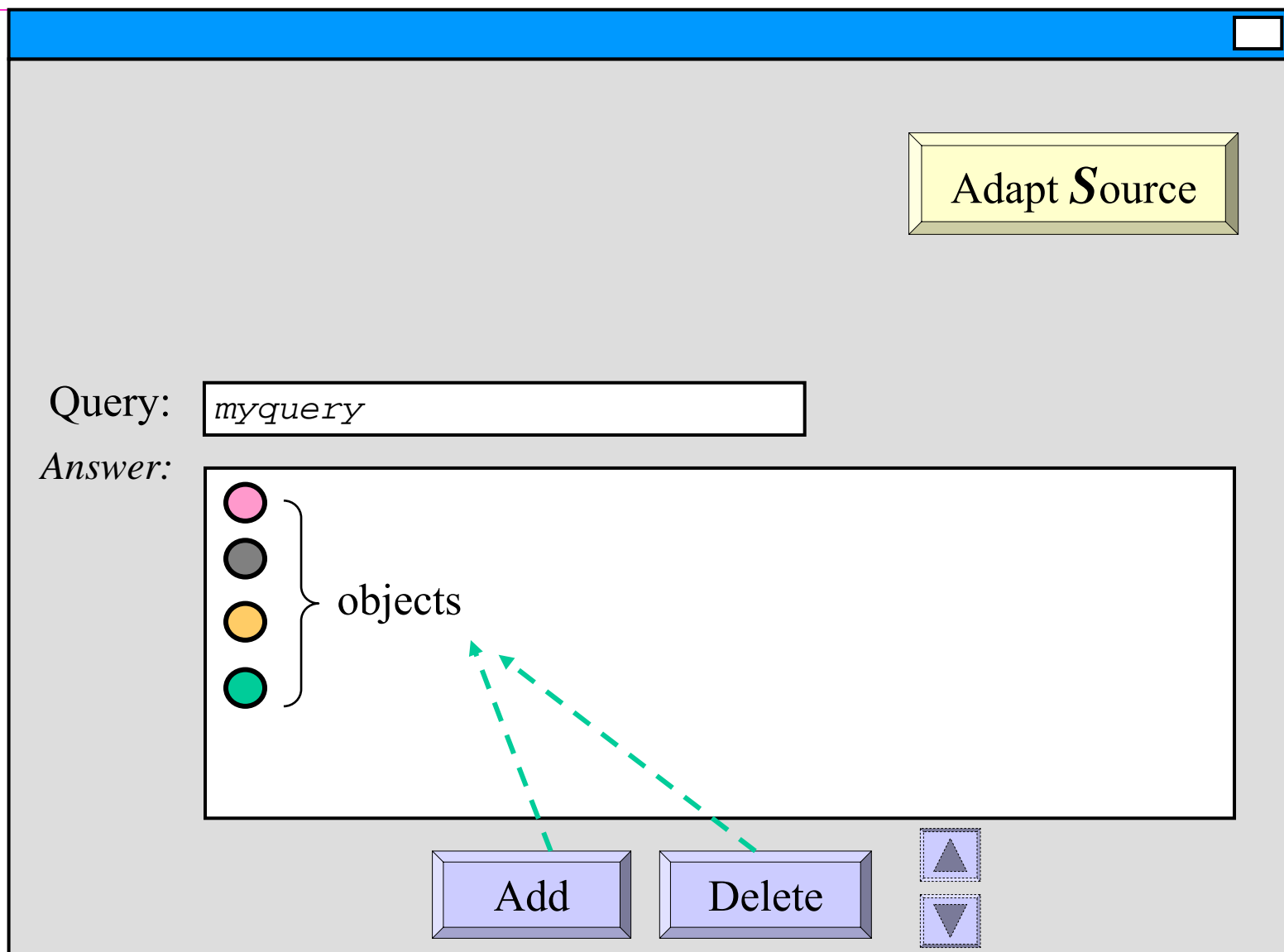
The sources can run this protocol for one, several or all of its terms (or queries)



# Personalization of Taxonomy-based Sources



# A possible UI for this interaction scheme





*Problem statement:*

Given a query  $q$  and an answer  $A$  find a source  $S$  in  $\mathcal{S}$  such that  $S(q)=A$

- There may exist several sources in  $\mathcal{S}$  that satisfy the above equation.
- The current source  $S_{cur}$  should be taken into account.

*Refined problem statement:*

Given a query  $q$  and an answer  $A$  find a source  $S$  in  $\mathcal{S}$  such that  $S(q)=A$   
and  $dist(S, S_{cur})$  is minimal.

Defining the distance between two sources:

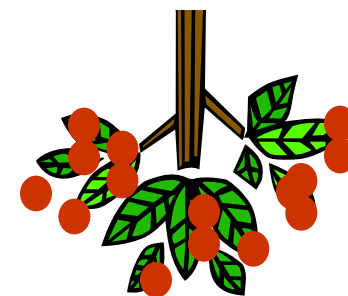
$$dist(S, S') = S \nabla S' = (S - S') \cup (S' - S)$$



Which is the set of all sources  $S$  that we assume ?



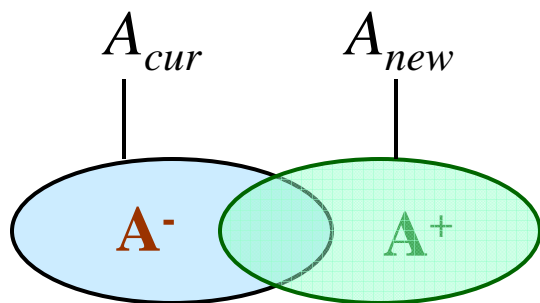
We assume that all sources in  $S$  have the same taxonomy,  
i.e. they differ only in their interpretation.



$(S, q, A_{cur})$ : current focus,  
where  $S = (T, \leq, I_{cur})$  the current stored source



$S_{new} = (T, \leq, I_{new})$   
such that  $S_{I_{new}}(q) = A_{new}$  and  $dist(I_{cur}, I_{new})$  is minimal.



Prosthetic perturbation  $pertAdd(o, t) = |Br(t) \setminus D_{\bar{I}}(o)|$

Apheretic perturbation  $pertDel(o, t) = |Nr(t) \cap D_{\bar{I}}(o)|$

**Conjunctive queries**  $q = t_1 \wedge \dots \wedge t_k$

- **add** each object of  $A^+$  to every term  $t_i$
- **delete** each object of  $A^-$  from only one term, the term with the minimum *pertDel*

**Disjunctive queries**  $q = t_1 \vee \dots \vee t_k$

- **add** each object of  $A^+$  to only one term, the term with the minimum *pertAdd*
- **delete** each object of  $A^-$  from every term in  $q$

The time complexity of source adaptation is

- $O(|A^+| + |A^-| |T|^2)$  for *single term queries*,
- $O(|A^- \cup A^+| k |T|^2)$  for *disjunctive queries* ( $t_1 \vee \dots \vee t_k$ ), and
- $O(|A^-| k |T|^2 + |A^+| k)$  for *conjunctive queries* ( $t_1 \wedge \dots \wedge t_k$ ).



- Experimental evaluation in a wide scale
- Study the case of interrelated facets (where subsumption links cross facet boundaries)
- Automatic facet/taxonomy extraction
- ...

*Thanks for your attention!*