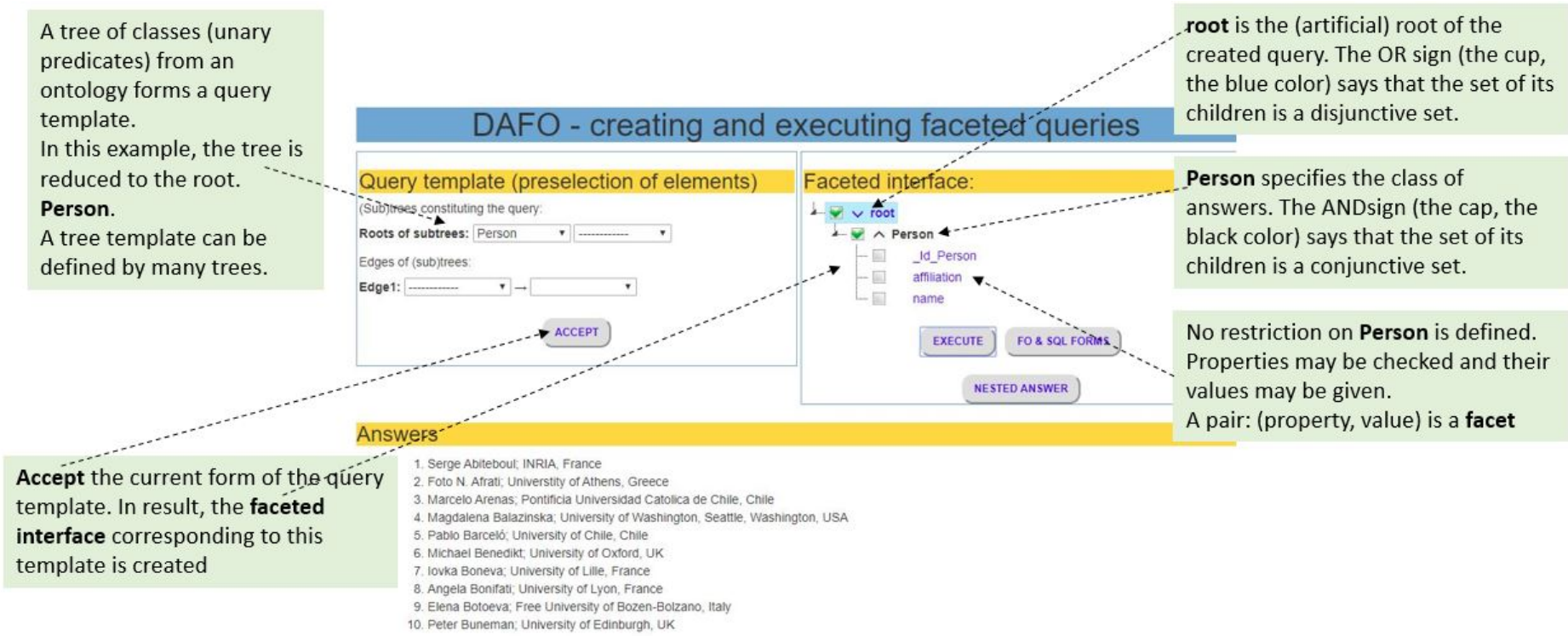
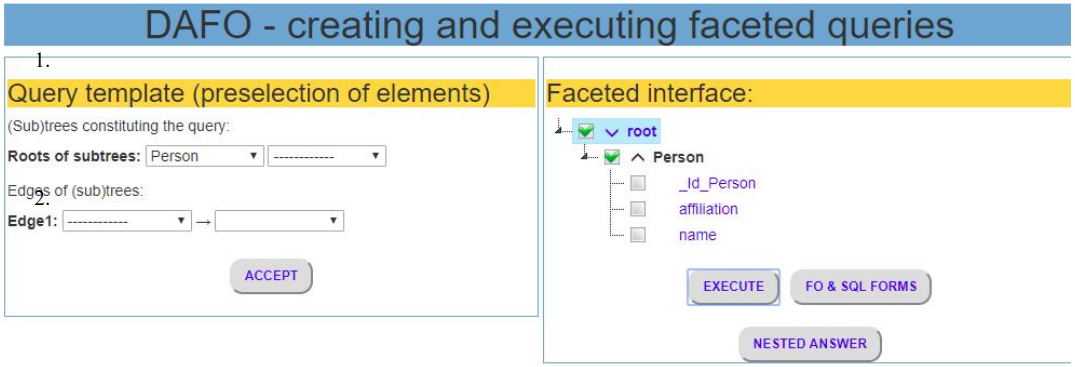
DAFO: Data Access based on Faceted queries over Ontology

Poznan University of Technology, tadeusz.pankowski@put.poznan.pl

A screen-shot and its explanation



Query 1: Get all persons (without any restriction)



"root", is an OR node.

Formulating faceted query

Providing a query templata, a (sub)tree of classes

A user starts giving a class of answers (the root of the query template tree). In the considered case, each answer will be of the class Person. Since no restriction is necessery, the query template is a tree reduced to the root.

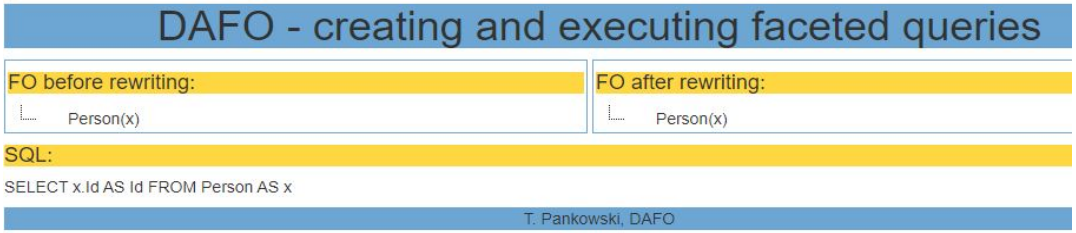
Creating a faceted query

After ACCEPT, a faceted interface determined by the provided query template tree is created. The set of answers is produced in responce to "EXECUTE". On the bottom of the figure, ten first answers is shown, ordered by the surname. The button "FO & SQL FORMS" invokes a new window showing first order forms of the created faceted query, as well as its SQL form (see below).

Coloring convention, conjunctive and disjunctive sets

A faceted interface (and also faceted queries as special forms of faceted interfaces) are AND/OR trees. Each node is the root of a (sub)tree. If a node is accompanied with the AND sigr (a cap), the node is interpreted as an AND node. Additionaly, AND nodes have the black color. Otherwise, the node is an OR node and has the blue color. By default, attributes are understood as OR nodes. To be an AND node meens that the set ot its childs is a conjunctive set. Similarly, children of an OR node constitute a disjunctive set. The context menu can be used to set AND/OR to a node.

At the beggining, all nodes labeled by classes (by unary predicates) are AND nodes, and all nodes labeled by properties (by binary predicates) are OR nodes. The root node, labeled by



Inspecting semantic of faceted queries

1. **Semantic of faceted queries**

   Semantics of a faceted query may not be obvious. To specify the semantics precisely, a first order form of the faceted query is provided (as two syntactic trees).

   In the "before rewriting" form, only names used in the faceted query are present. In "after rewriting" form, all intensional predicates are rewritten into extensional ones.

   The semantics of queries conforms to the semantics of description logic.

2. **First order queries: before and after rewriting**

   In this case, the FOL query is: Person(x). Person is an extensional predicate, so no rewriting is needed.

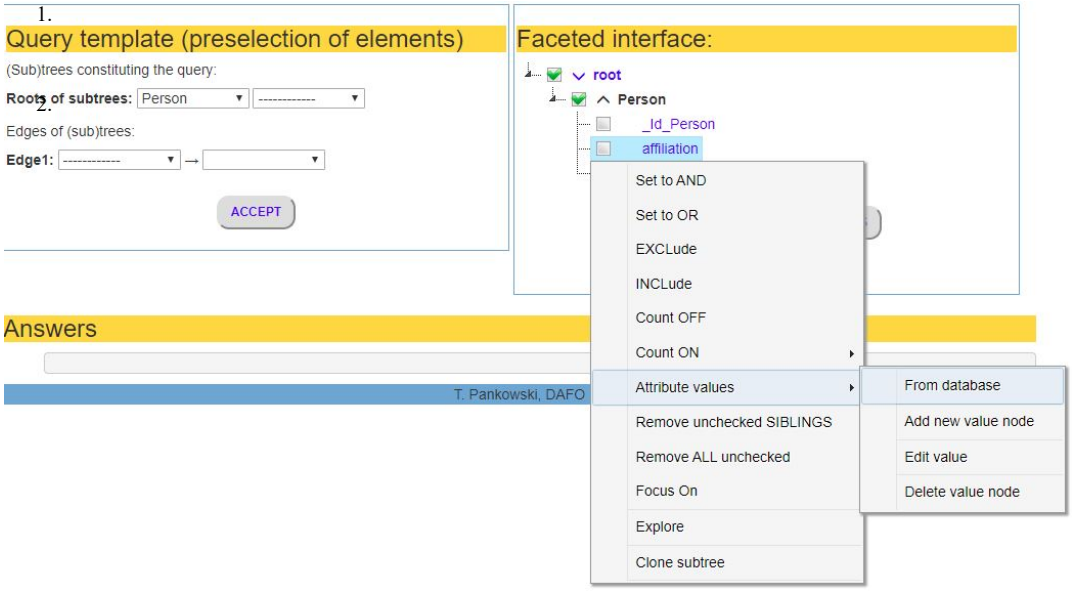
3. **SQL query**

   The rewritten form of the query is transformed into SQL. The SQL query is executed by the relational database engine.

top

Query 2: Persons with a given affiliation. Using context menu

1. 2.



Invoking the context menu

Right click on a node displays the context menu.

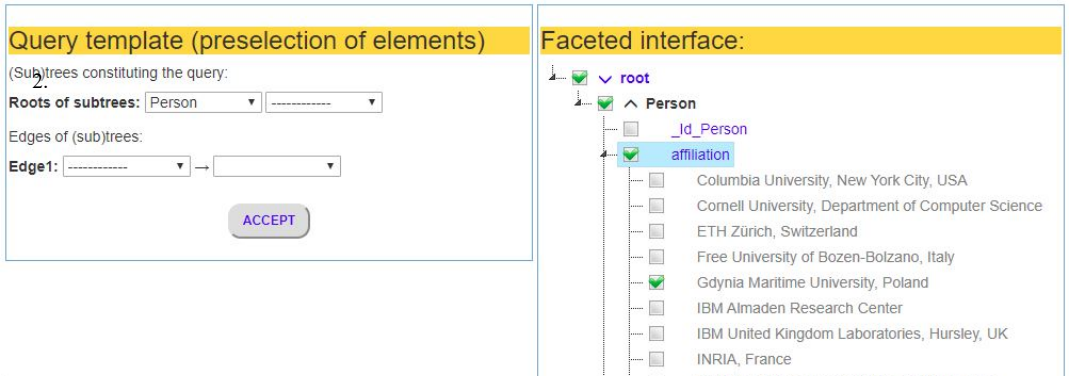
Operations controlled by the context menu: "Attribute value"

"Attribute value" allows to insert a new value of the indicated attribute, as well as edit or delete the attribute value. Stored values can be read from the underlying database: "From database". A new value can be added manually by a user: "Add new value node". The user can edit the indicated value: "Edit value", or delete the value: "Delete value node".

Selecting from a list of values read from a database

1. **List of available values**

2.



The user is provided with a check list of values of the indicated attribute stored in the database and satisfying the current state of the created faceted query

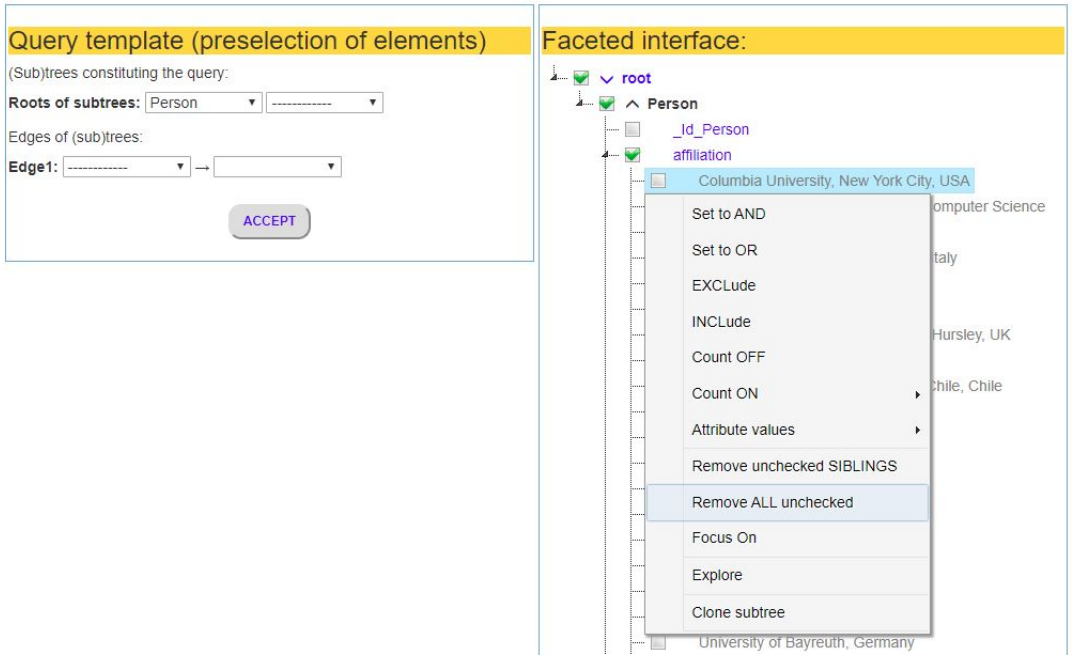
Checking values

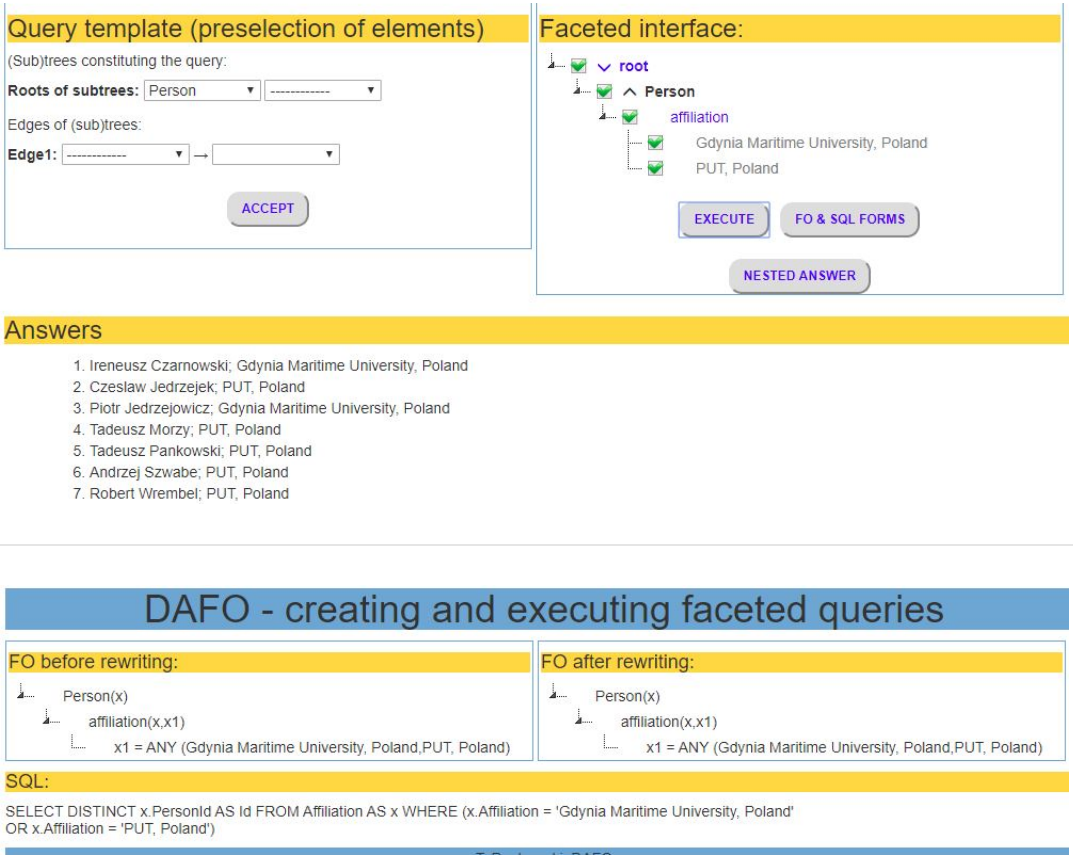
The user selects a set of values checking them. The set is interpreted as a disjunctive set.

Removing unnecessary nodes

1. **Remove ALL unchecked**

   The user can remove all unchecked nodes. The other possibility is to remove all unchecked sibling nodes.





The final form of the created faceted query

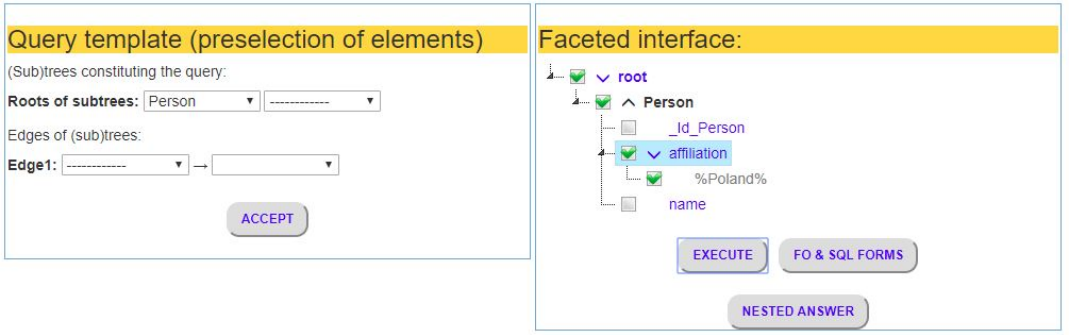
First order semantics of the query and its SQL form

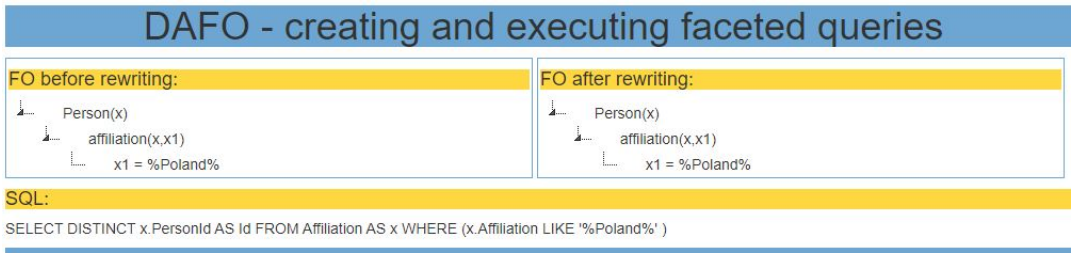
top

Query 3: Persons with an affiliation defined by a pattern

1. **Add new node - inserting an attribut value as a pattern of the form %xxxx%**

   This menu option allows for inserting a value of the indicated attribute manually. The value can be a pattern of the form %string% meaning that we are interested in values containing "string" as a substring. In this case, all affiliations with the substring "Poland" satisfy the restriction.





First order semantics of the query and its SQL form

top

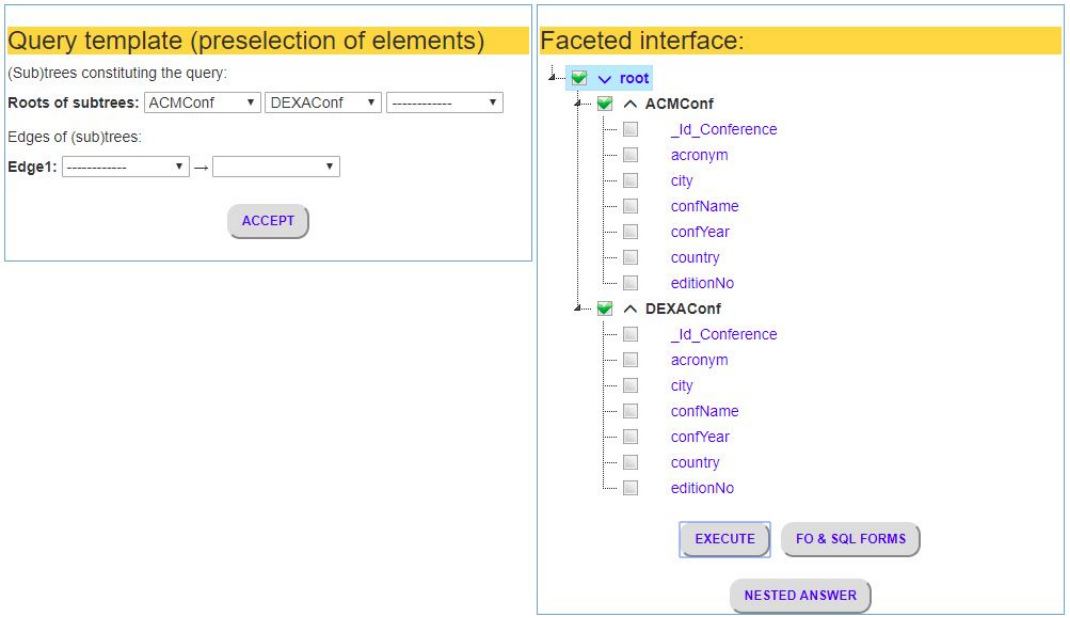
Query 4: UNION of ACM conferences and DEXA conferences

1. **Query template**

   In this case, the query template is a forest of two trees. Each tree consists only of a root, respectively, ACMConf and DEXAConf.

2. **Disjunctive set of children of "root"**

By default, "root" is an OR node and its children constitute a disjunctive set. Thus, the faceted interface represents the disjunction: ACMConf OR DEXAConf.

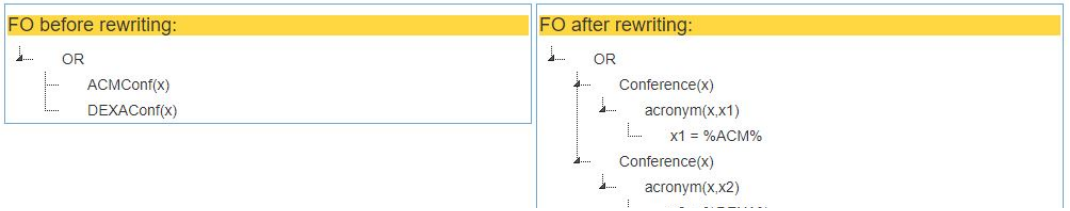


Answers

1. ACM SIGMOD, 2016, ACM SIGMOD Conference 2016: San Francisco, CA, USA, ACM SIGMOD Conference, , USA, San Francisco, CA
2. DEXA, 2016, 27. DEXA 2016: Porto, Portugal - Part II, Database and Expert Systems Applications (DEXA), 27 , Portugal, Porto
3. DEXA, 2016, 27. DEXA 2016: Porto, Portugal - Part I, Database and Expert Systems Applications (DEXA), 27 , Portugal, Porto
4. ACM SIGMOD, 2015, ACM SIGMOD Conference 2015: Melbourne, Victoria, Australia, ACM SIGMOD Conference, , Australia, Melbourne, Victoria

---

DAFO - creating and executing faceted queries



First order semantics of the query and its SQL form

1. **First order forms**

   Now, the two first order forms: before rewriting and after rewriting differ substantially. It follows value (this value has a form of pattern %ACM%). Similarly for DAXAConf.

top

x2 = %DEXA%

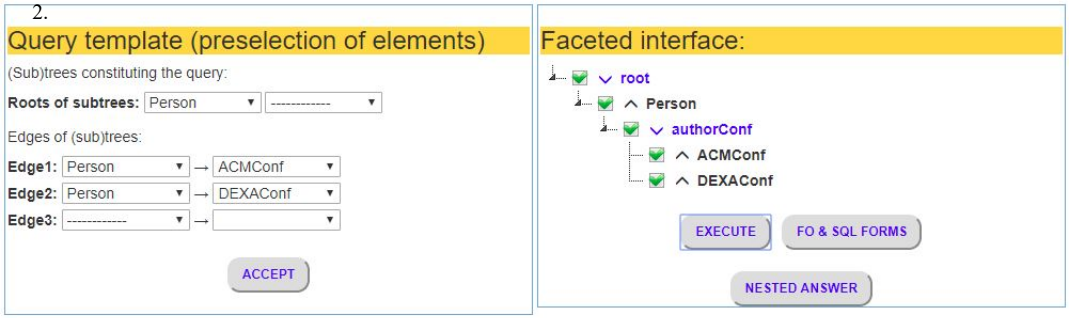
SQL:

```
SELECT DISTINCT x.Id AS Id FROM Conference AS x WHERE (x.Acronym LIKE '%ACM%' )
UNION
(SELECT DISTINCT x.Id AS Id FROM Conference AS x WHERE (x.Acronym LIKE '%DEXA%' ))
```

Query 5: Persons connected with an ACM or a DEXA conference

1. **Query template**

   The query template is a tree rooted in Person. The tree has two edges (Person, ACMConf) and (Person, DEXAConf).

2. 

Disjunctive set of children of "authorConf"

In the faceted interface implied by the query template tree, Person is connected with ACMConf and/or DEXAConf via the object property authorConf. By default, authorConf is an OR node and its children constitute a disjunctive set. Thus, the faceted query is interpreted as the query: Person(authorConf(ACMConf OR DEXAConf)).

Note, that authorConf connects a person with a conference if a paper written by this person was presented at this conference.

Answers

1. Serge Abiteboul; INRIA, France
2. Foto N. Afrati; Universtity of Athens, Greece
3. Marcelo Arenas; Pontificia Universidad Catolica de Chile, Chile
4. Magdalena Balazinska; University of Washington, Seattle, Washington, USA
5. Michael Benedikt; University of Oxford, UK
6. Angela Bonifati; University of Lyon, France

DAFO - creating and executing faceted queries

FO before rewriting:

```
Person(x)
  authorConf(x,x1)
    OR
      ACMConf(x1)
      DEXAConf(x1)
```

FO after rewriting:

```
Person(x)
  authorOf(x,x2)
    Paper(x2)
      inProceedings(x2,x3)
        Proceedings(x3)
          ofConf(x3,x1)
            OR
              Conference(x1)
                acronym(x1,x4)
                  x4 = %ACM%
              Conference(x1)
                acronym(x1,x5)
                  x5 = %DEXA%
```

First order semantics of the query and its SQL form

1. **First order forms**

   Now, the predicates: authorConf, ACMConf and DEXAConf, are intensional predicates. The f

2. **Variables**

   Note, that each valuation of x1 must satisfy at least one predicate ACMConf or DEXAConf (F

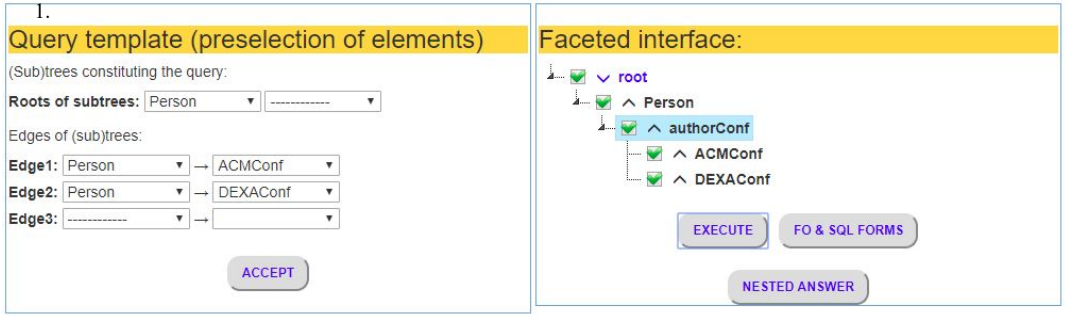
top

SQL:

```
SELECT DISTINCT x.PersonId AS Id FROM AuthorPaper AS x
JOIN(SELECT DISTINCT x2.Id AS Id FROM Paper AS x2
JOIN(SELECT DISTINCT x3.Id AS Id FROM Proceedings AS x3
JOIN(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Acronym LIKE '%ACM%' )
UNION
(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Acronym LIKE '%DEXA%' ))
)AS x1 ON x1.Id = x3.ConfId
)AS x3 ON x3.Id = x2.ProcId
)AS x2 ON x2.Id = x.PaperId
```

Query 6: Persons connected with an ACM and a DEXA conference (incorrect!)

1.

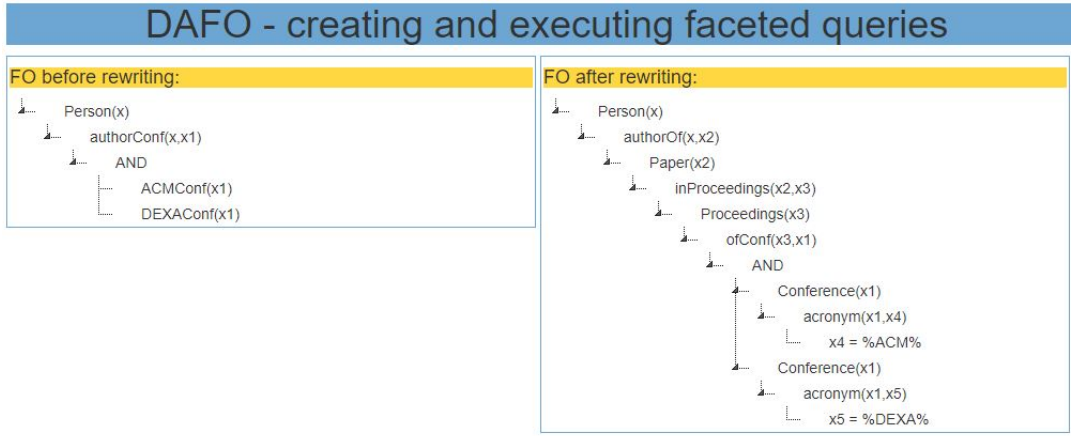


Answers

Changing a disjunctive set authorConf children into a conjunctive one

The OR node authorConf (in the previous example) could be changed into the AND node. Then the set of its children is changed to a conjunctive set. Now, authorConf connects persons with ACM and DEXA conferences. However, it is incorrect since there is no a conference that is both an ACM and DEXA conference. This is clear from the first order semantics shown in the next figure.

This example illustrates the inconsistency between natural language semantics and the formal (firs order) semantics.



SQL:

```
SELECT DISTINCT x.PersonId AS Id FROM AuthorPaper AS x
JOIN(SELECT DISTINCT x2.Id AS Id FROM Paper AS x2
JOIN(SELECT DISTINCT x3.Id AS Id FROM Proceedings AS x3
JOIN(SELECT DISTINCT x1_L.Id AS Id FROM(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Acronym LIKE '%ACM%' )) AS x1_L
```

First order semantics of the query and its SQL form

1. **Variables**

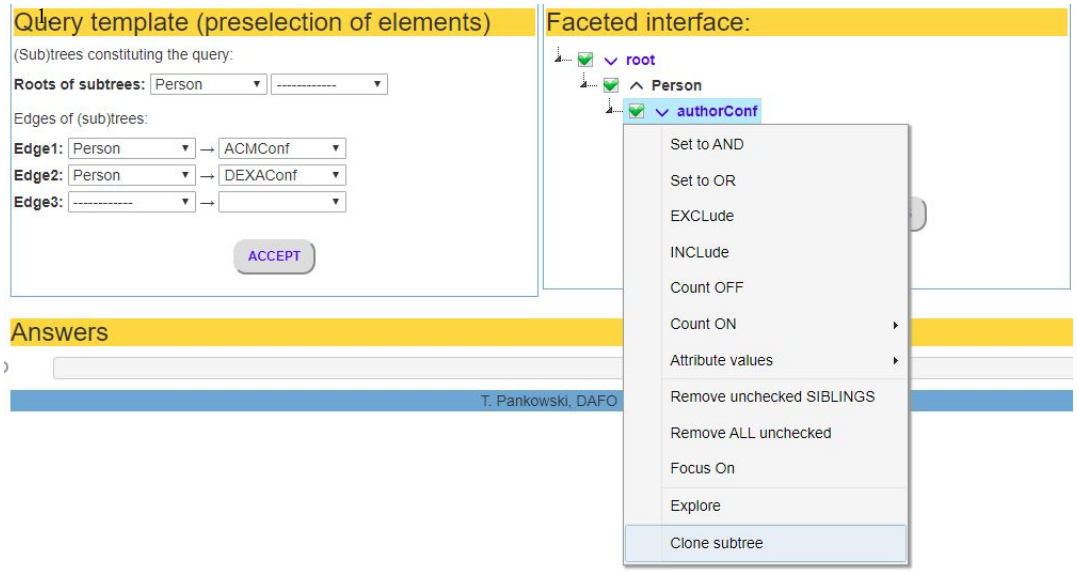
   Note that the variable x1 must satisfy both ACMConf and DEXAConf (FO before rewriting). T

top

```
JOIN(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Acronym LIKE '%DEXA%' )
)AS x1 ON x1.Id = x1_L.Id
)AS x1 ON x1.Id = x3.ConfId
)AS x3 ON x3.Id = x2.ProcId
)AS x2 ON x2.Id = x.PaperId
```

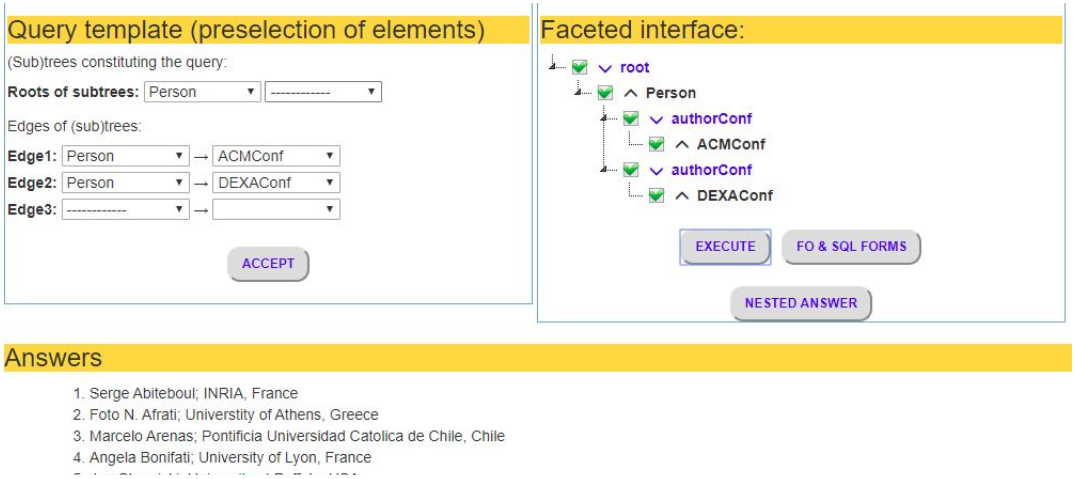
T. Pankowski, DAFO

Query 7: Persons connected with an ACM and a DEXA conference (correct! by cloning)



Cloning the subtree rooted in authorConf

First, we clone a subtree rooted in authorConf. This can be done using the context menu. The option "Clone subtree" duplicates the tree rooted in the indicated node.



Modifying the cloned subtrees

1. **A subtree indicating connections with ACM conferences**

   We leave ACMConf in the first subtree

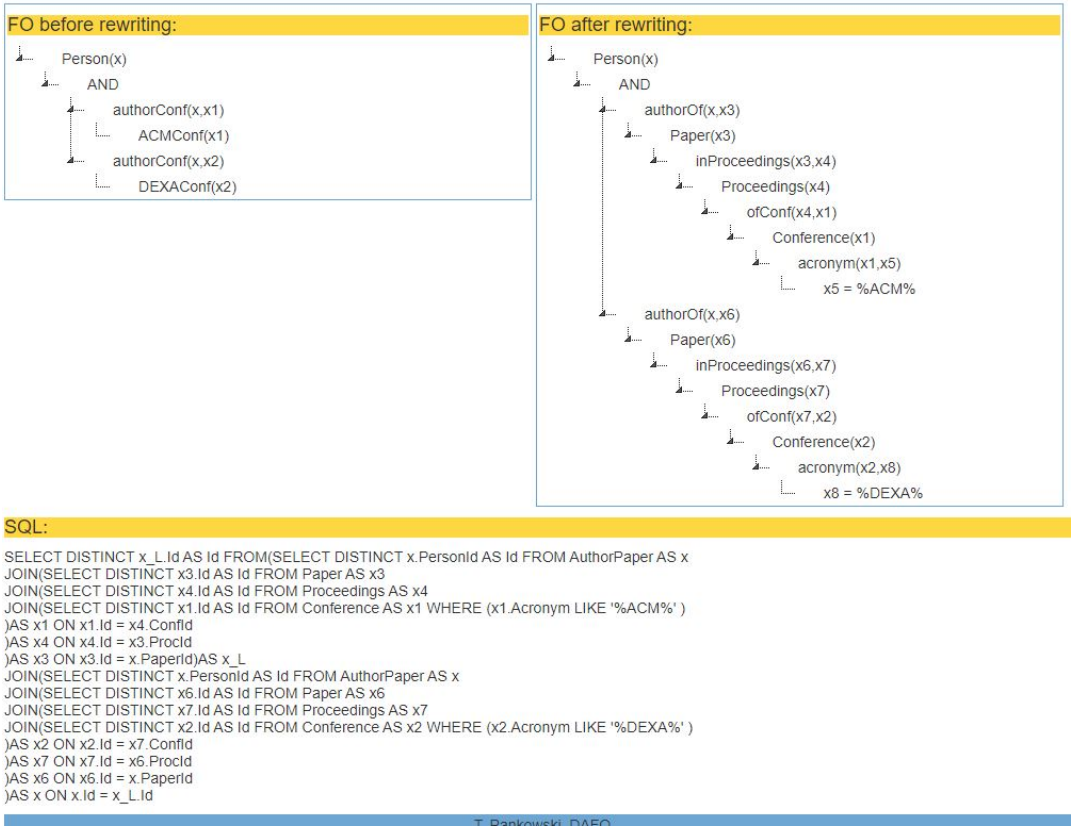
2. **A subtree indicating connections with DEXA conferences**

   We leave DEXAConf in the other subtree

3. **Semantics as expected**

   Since Person is an AND tree, its two children constitute a conjunctive set. This expresses the

5. Jan Chomicki; University at Buffalo, USA
6. Alin Deutsch; University of California, San Diego, USA
7. Georg Gottlob; University of Oxford, UK; TU Vienna, Austria
8. Maurizio Lenzerini; Sapienza University of Rome, Italy
9. Leonid Libkin; University of Edinburgh, UK
10. Tova Milo; Tel Aviv University, Israel
11. Martin Theobald; University of Ulm, Germany; University of Luxembourg



First order semantics of the query and its SQL form

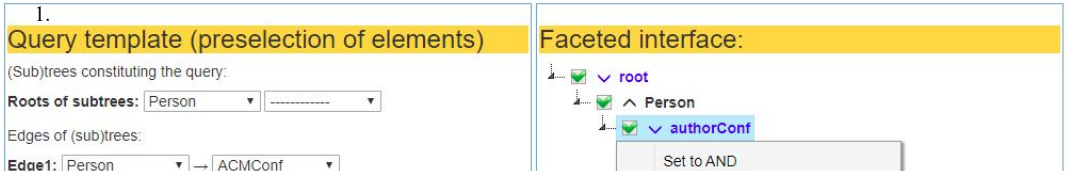
1. **Variables**

   Now, the variable x, denoting a person, is connected with two variables: x1 denoting ACMCor
   And this expresses correctly the expected semantics of the query.

top

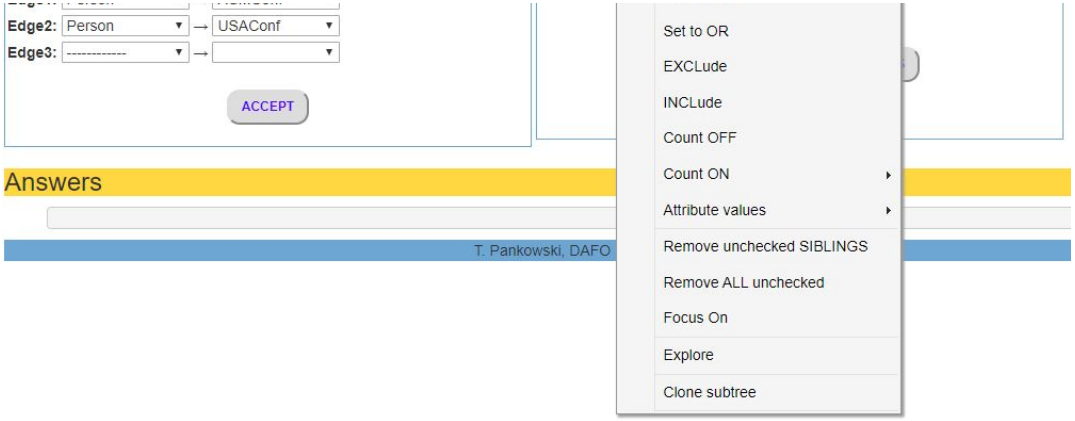
Query 8: Persons connected with ACM conferences in the USA

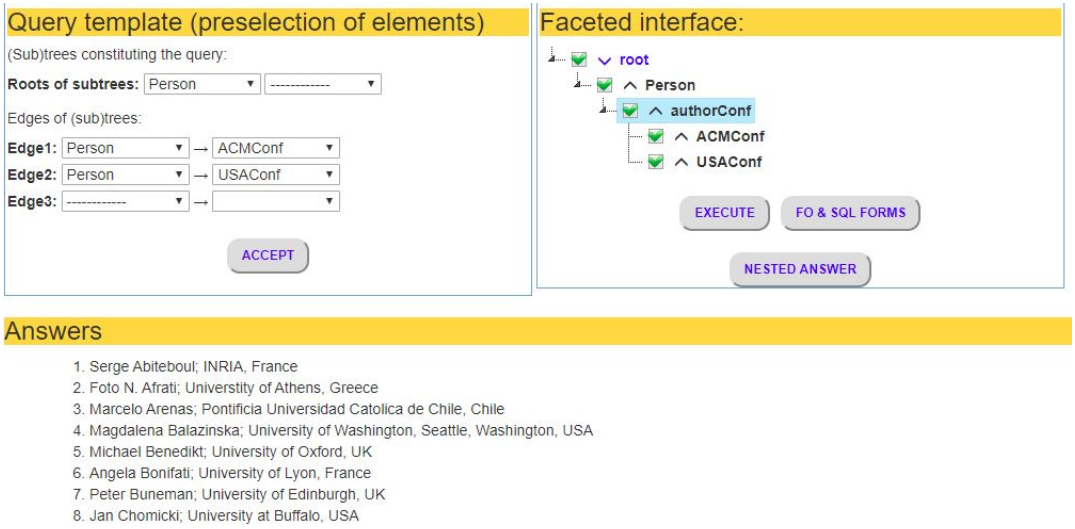
1.



Changing the authorConf from OR to AND

We start with the query template being a tree with the root Person and two edges leading to ACMConf and USAConf, respectively. The proposed faceted interface is an OR/AND tree, in which there is an OR node authorConf. We use the first position with the context menu, "Set to AND", to change the node authorConf to AND node.



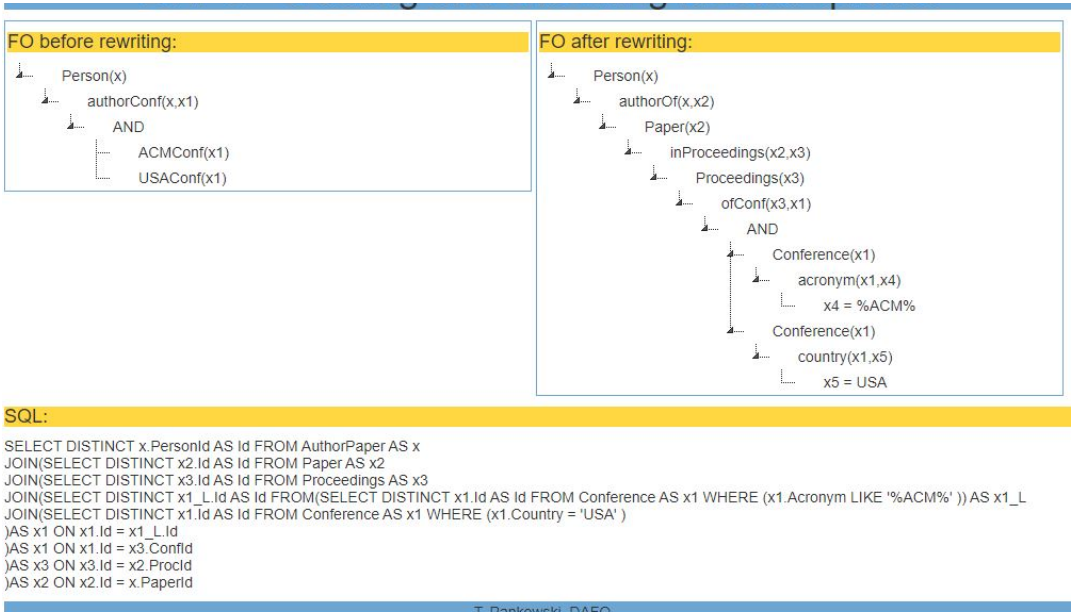


The final form of the faceted query.

First order semantics of the query and its SQL form.

top

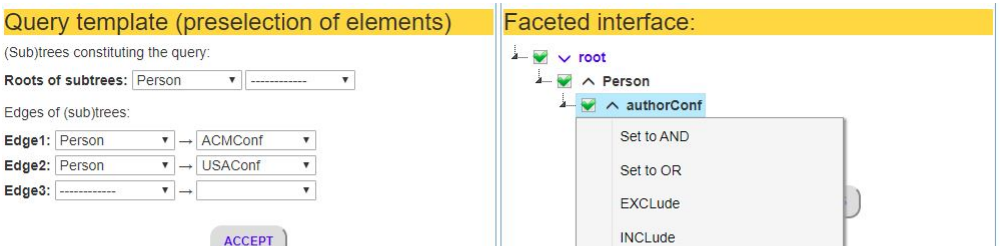
DAFO - creating and executing faceted queries

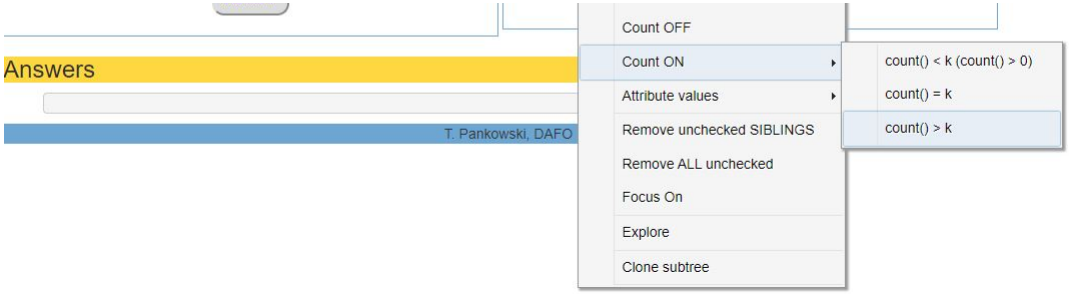


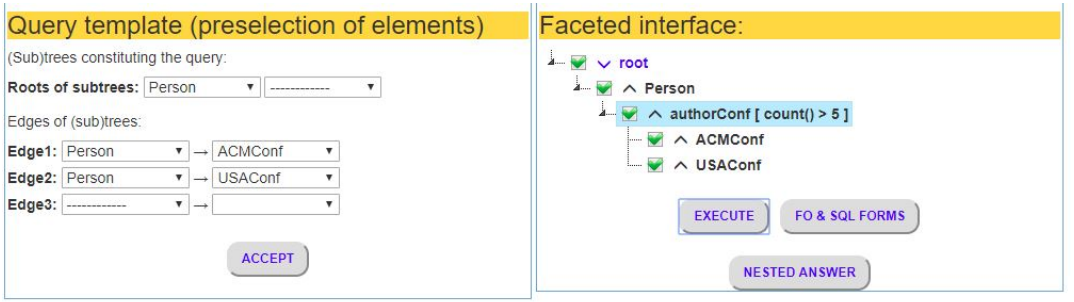
Query 9: People connected with at least five ACM conferences in the USA. Count() restriction

1. **Defining count() restriction**

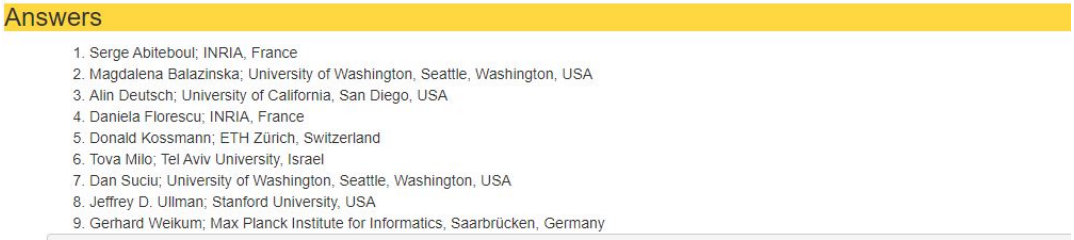
   From the context menu the option "Count ON" is chosen. Next, from a submenu the appropriate position is selected, and the value of k (5 in this case) is given. Note, that the count() restriction is defined with respect to the property node labeled authorConf. The meaning is: a domain value, say a person x, of authorConf must be connected by authorConf with more than 5 distinct values of x1, where x1 denotes elements from the range class of authorConf (i.e., from the Conference class, in the considered example).

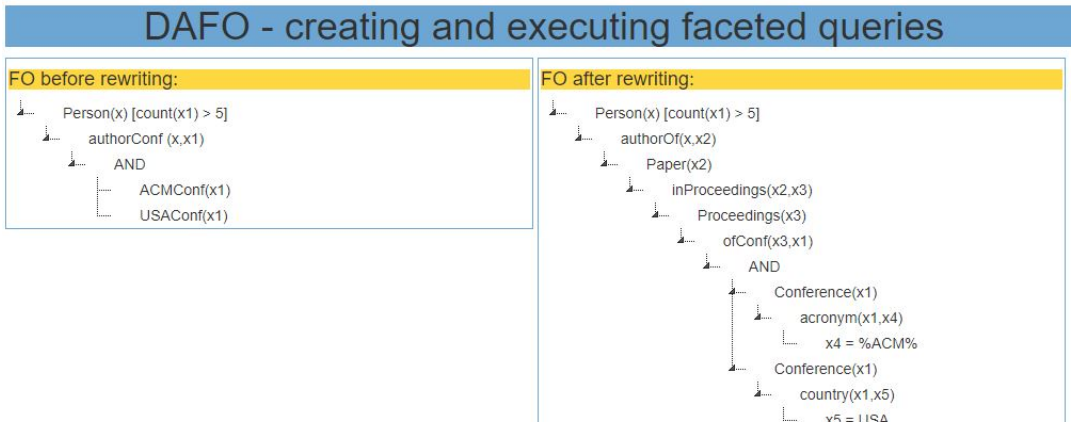






The final form of the faceted query.





1. **First order semantics of the query and its SQL form.**

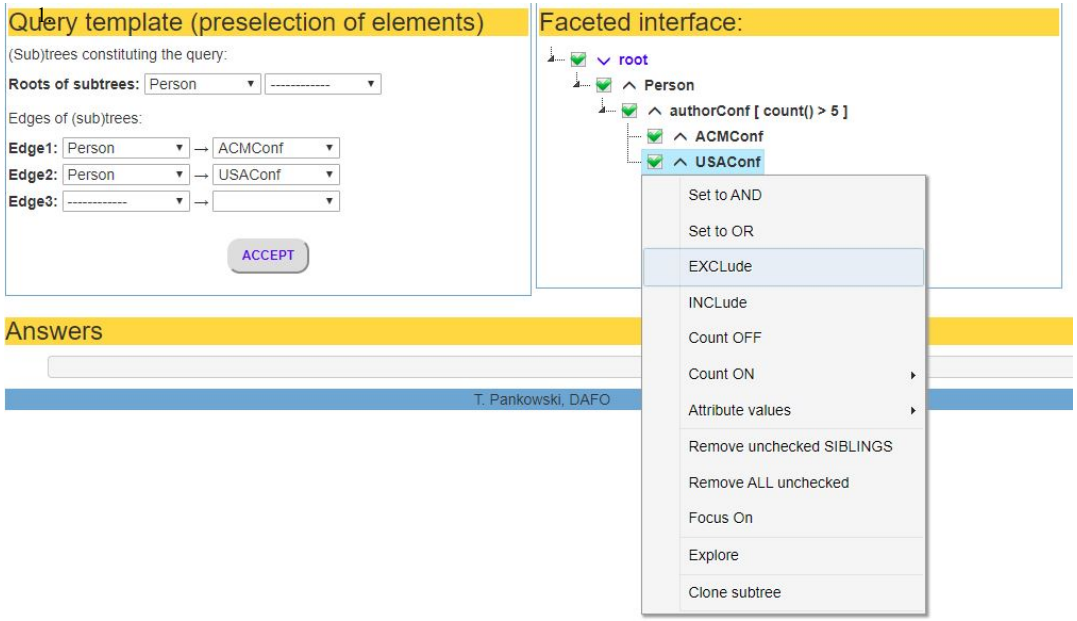
   The information about a count restriction is given by [count(x1) > 5] labeling the formula Pers valuation of x1 satisfying the query.

top

SQL:

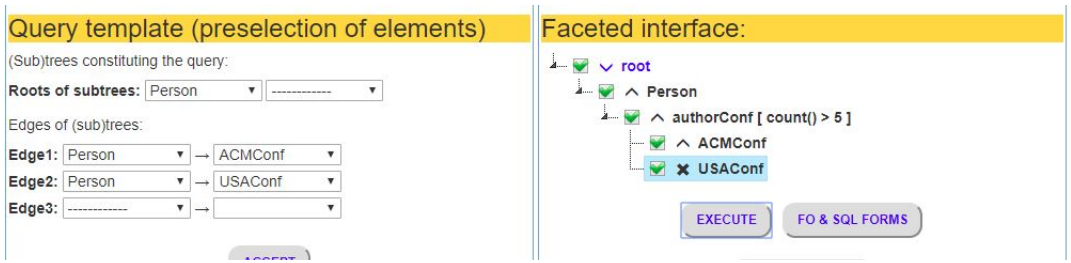
```
SELECT DISTINCT x.Id FROM(
SELECT DISTINCT x.PersonId AS Id, COUNT(DISTINCT x2.x_x1_count) AS x_x1_count FROM AuthorPaper AS x
JOIN(SELECT DISTINCT x2.Id AS Id, x3.x_x1_count FROM Paper AS x2
JOIN(SELECT DISTINCT x3.Id AS Id, x3.ConfId AS x_x1_count FROM Proceedings AS x3
JOIN(SELECT DISTINCT x1_L.Id AS Id FROM(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Acronym LIKE '%ACM%' )) AS x1_L
JOIN(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Country = 'USA' )
)AS x1 ON x1.Id = x1_L.Id
)AS x1 ON x1.Id = x3.ConfId
)AS x3 ON x3.Id = x2.ProcId
)AS x2 ON x2.Id = x.PaperId
GROUP BY x.PersonId) AS x
WHERE x.x_x1_count > 5
```

Query 10: People connected with at least five ACM conferences but NOT in the USA



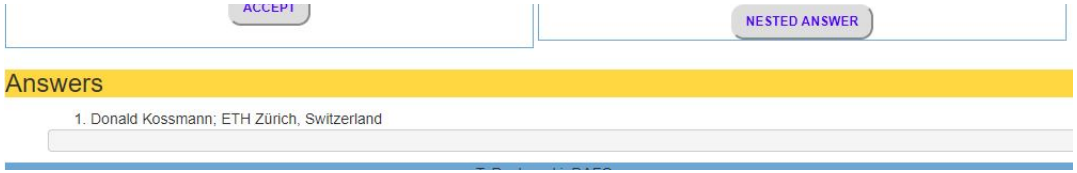
Negation by excluding USAConf

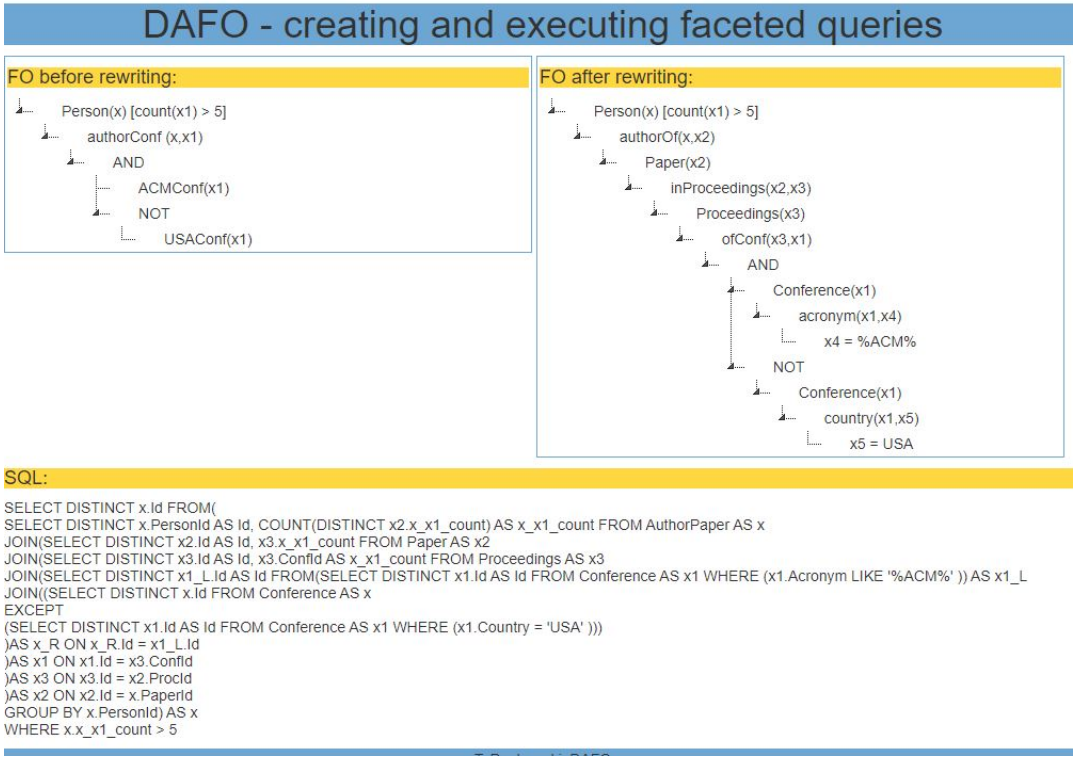
Negation of the formula represented by a subtree, is realized by excluding (negating) this subtree. So, to negate a subtree, the EXCLude option is selected from the context menu.



1. **The final form of the faceted query**

   The exclusion (negation) is denoted by the cross sign proceding the root of the subtree.



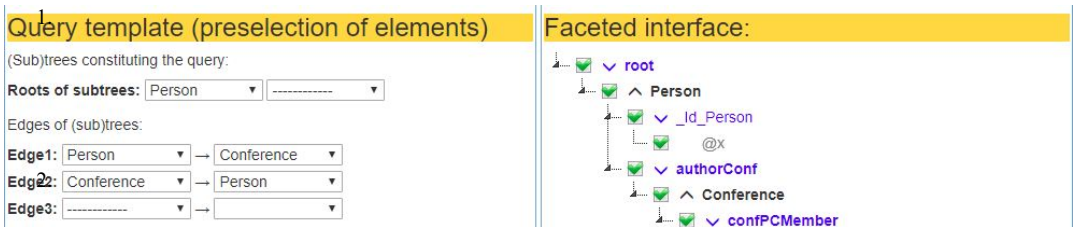


1. **First order semantics of the query and its SQL form.**

   The information about a count restriction is given by [count(x1) > 5] labeling the formula Person(valuation of x1 satisfying the query.

top

Query 11: People with articles at conferences at which they were members of program committees. Cycle



Cycle template

A template of a cycle query is a tree, in which there is a sequence of edges constituting a cycle. It means, that the first edge begins with the same class as the last edge ends. In the considered example, the first edge is (Person, Conference), and the last is (Conference, Person). So, these two edges create a cycle.

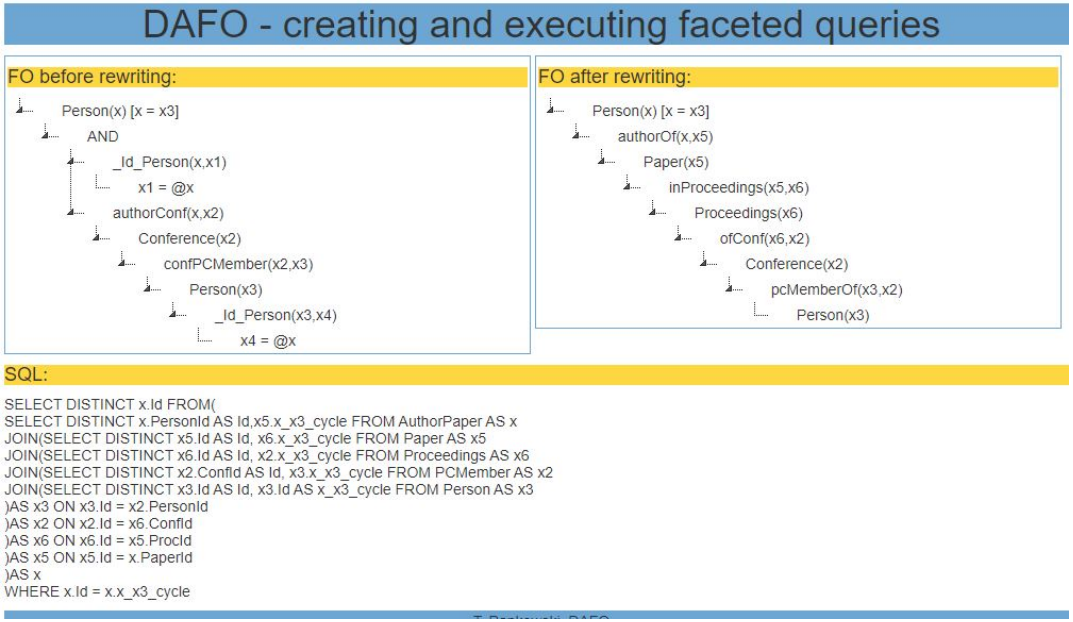
Defining a cycle query

A faceted interface created on the basis of a cycle template, contains a sequence of nodes

Answers

1. Tadeusz Pankowski; PUT, Poland
2. Robert Wrembel; PUT, Poland

A faceted interface created on the basis of a cycle template, contains a sequence of nodes constituting a cycle. A special meaning have identifiers of classes beginning and ending the cycle. Identifiers are attributes of the form "_Id_class". To define a cycle, the user introduces the same variable of the form "@x" to the first and the last identifier in the cycle.



First order semantics of the query and its SQL form.

1. **First order form before rewriting**

   The class beginning the cycle is labeled by an equality indicating identity of variable valuations. be equal to the identifier of the object x3. This implies the equality: x = x3.

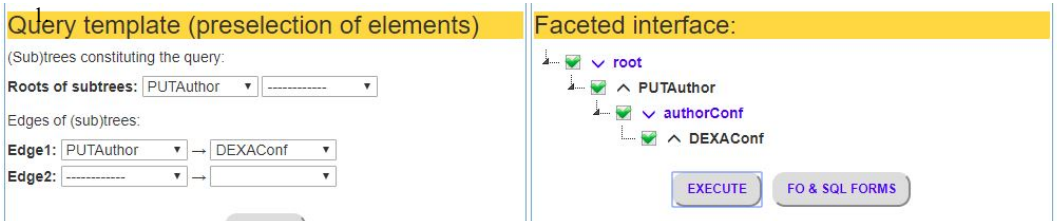
2. **First order form after rewriting**

   In this form, informations about equality betwee identifiers is ommitted.

   The information about a cycle is given by [x = x3] labeling the formula Person(x). According to th as the valuation of x, and satisfies the query.

top

Query 12: PUT authors connected with DEXA conferences. Nested answer



Defining the query

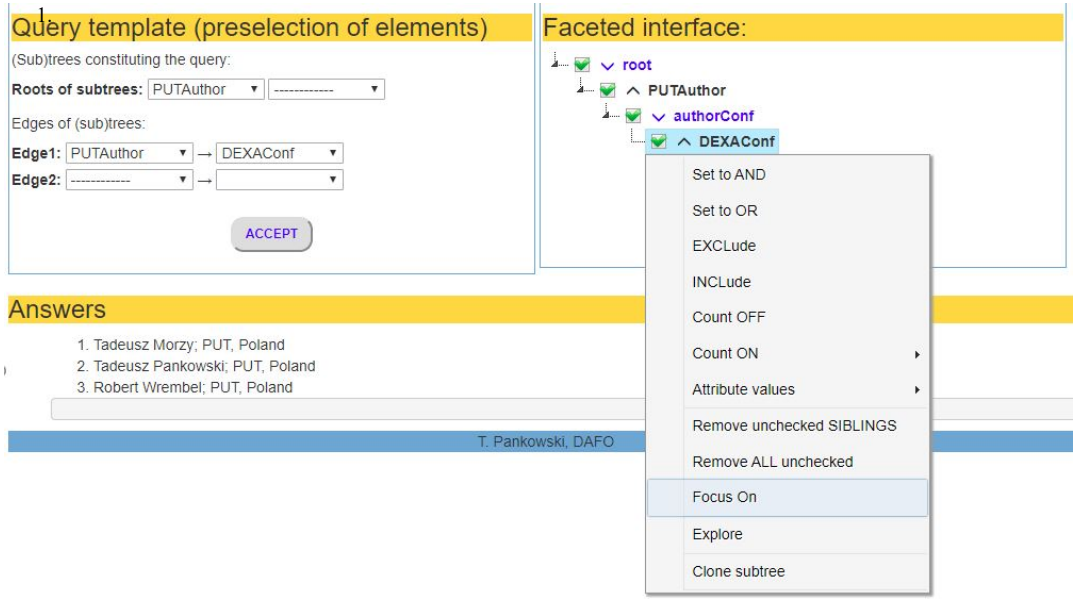
The tree-shaped form of the query can be used as a template of a nested form of the answer. Then, after completion of the query, the button "NESTED ANSWER" is pressed. In response, a nested answer is displayed in a separate window.

Nested form of the answer.

Now, a user can expand/collapse nodes of the nested answer.

top

Query 13: PUT authors connected with DEXA conferences, and inversely. Refocussing

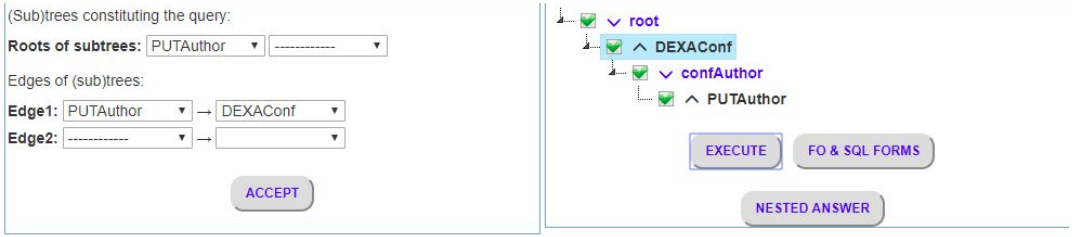


Refocussing

The query in the figure asks about PUT authors (authors from Poznan University of Technology) who are connected with DEXA conferences. If a user wants to know what these conferences are (i.e., all DEXA conferences connected with PUT authors), she/he can use the refocussing (or pivoting) functionalty.

The refocussing is realized by means of the option "Focus On" in the context menu.

Result of refocussing



The result of refocussing is a faceted query rewritten in such a way that: (a) answers are o the set of objects of the same class befor rewriting.

In the considered example, the set of DEXA conferences involved in the query before and

Answers

1. DEXA, 2016, 27. DEXA 2016: Porto, Portugal - Part II, Database and Expert Systems Applications (DEXA), 27 , Portugal, Porto
2. DEXA, 2015, 26. DEXA 2015: Valencia, Spain, Database and Expert Systems Applications (DEXA), 26 , Spain, Valencia
3. DEXA, 2012, 23. DEXA 2012: Vienna, Austria, Database and Expert Systems Applications (DEXA), 23 , Austria, Vienna
4. DEXA, 2010, 21. DEXA 2010: Bilbao, Spain, Database and Expert Systems Applications (DEXA), 21 , Spain, Bilbao
5. DEXA Workshops, 2007, 18. DEXA Workshops 2007: Regensburg, Germany, DEXA Workshops, 18 , Germany, Regensburg
6. DEXA, 2006, 17. DEXA 2006: Krakow, Poland, Database and Expert Systems Applications (DEXA), 17 , Poland, Krakow
7. DEXA Workshops, 2006, 17. DEXA Workshops 2006: Krakow, Poland, DEXA Workshops, 17 , Poland, Krakow
8. DEXA Workshop, 2001, 12. DEXA Workshop 2001: Munich, Germany, DEXA Workshops, 12 , Germany, Munich
9. DEXA Workshop, 1999, 10. DEXA Workshop 1999: Florence, Italy, DEXA Workshops, 10 , Italy, Florence
10. DEXA, 1991, 2. DEXA 1991: Berlin, Germany, Database and Expert Systems Applications (DEXA), 2 , Germany, Berlin
11. DEXA, 1990, 1. DEXA 1990: Vienna, Austria, Database and Expert Systems Applications (DEXA), 1 , Austria, Vienna

Nested answers (expand/collapse items):

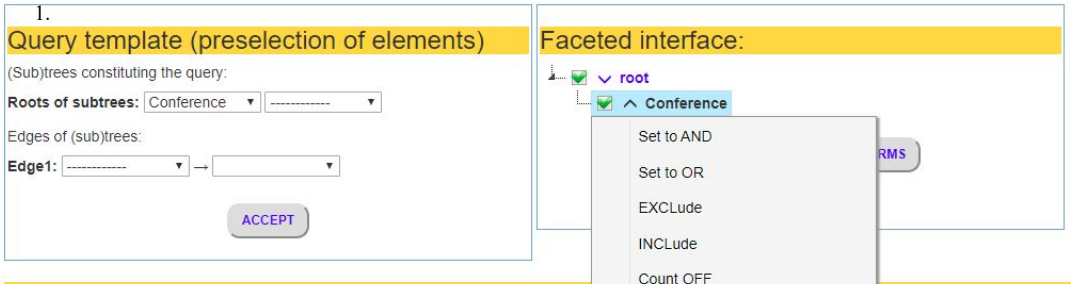
1. DEXA, 2016, 27. DEXA 2016: Porto, Portugal - Part II, Database and Expert Systems Applications (DEXA), 27 , Portugal, Porto
2. DEXA, 2015, 26. DEXA 2015: Valencia, Spain, Database and Expert Systems Applications (DEXA), 26 , Spain, Valencia
   1. Tadeusz Morzy; PUT, Poland
   2. Robert Wrembel; PUT, Poland
3. DEXA, 2012, 23. DEXA 2012: Vienna, Austria, Database and Expert Systems Applications (DEXA), 23 , Austria, Vienna
4. DEXA, 2010, 21. DEXA 2010: Bilbao, Spain, Database and Expert Systems Applications (DEXA), 21 , Spain, Bilbao
5. DEXA Workshops, 2007, 18. DEXA Workshops 2007: Regensburg, Germany, DEXA Workshops, 18 , Germany, Regensburg
6. DEXA, 2006, 17. DEXA 2006: Krakow, Poland, Database and Expert Systems Applications (DEXA), 17 , Poland, Krakow
7. DEXA Workshops, 2006, 17. DEXA Workshops 2006: Krakow, Poland, DEXA Workshops, 17 , Poland, Krakow
   1. Tadeusz Pankowski; PUT, Poland
8. DEXA Workshop, 2001, 12. DEXA Workshop 2001: Munich, Germany, DEXA Workshops, 12 , Germany, Munich

Nested answer to the query after refocussing

The set of DEXA conferences constituting answers to the considered query after refocussi conferences.

top

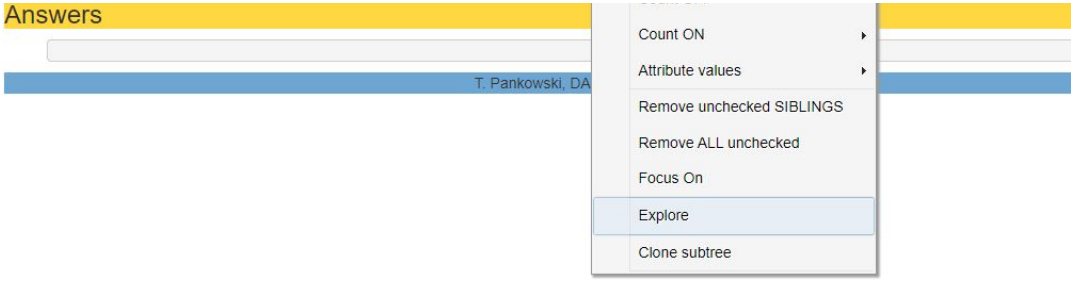
Query 14: Explore the context of a class in the ontology

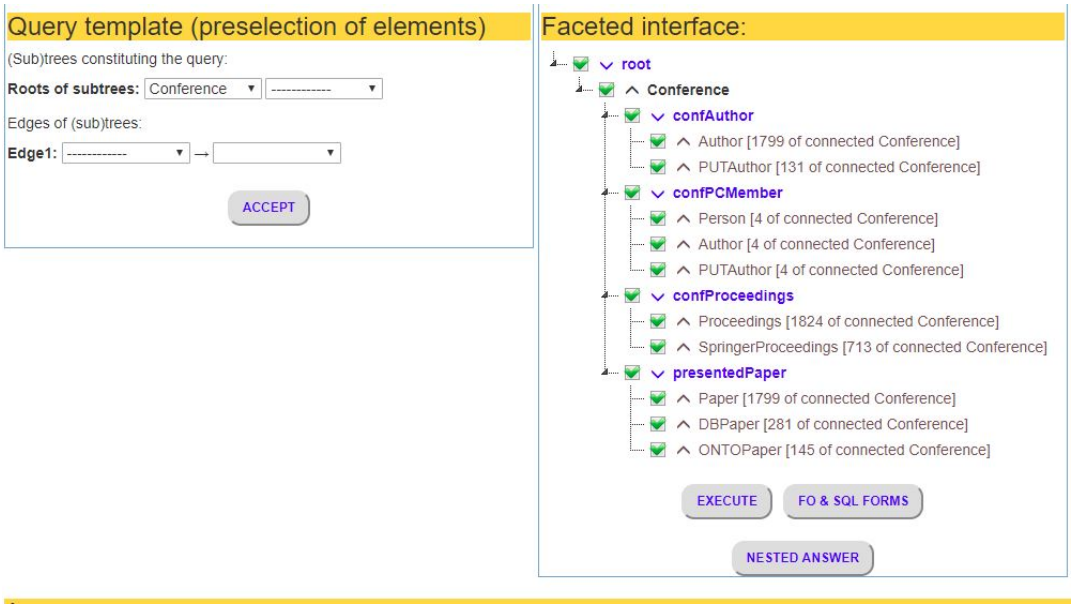


Exploring

The query in the figure asks about all conferences. If a user wants to know the context of this class in the ontology (i.e., all properties having the class Conference as their domain or range) she/he can use the exploring functionalty.

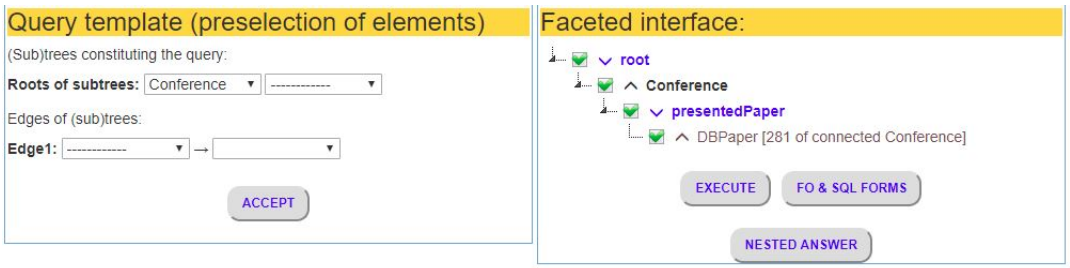
The exploring is realized by means of the option "Explore" in the context menu.





Result of exploring

The result of exploring is a faceted interface expanded with all posible subtrees which root connected to the considered class in a particular subtree.



Selection based on exploring information

In this figure, the user decided to ask about conferences at which DB papers (i.e., papers

top