

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 template (preselection of elements)

(Sub)trees constituting the query:
Roots of subtrees: Person
Edges of (sub)trees:
Edge1: _____ → _____

ACCEPT

Faceted interface:

root
Person
 |_d_Person
 | affiliation
 | name
 | EXECUTE
 | FO & SQL FORMS
 | NESTED ANSWER

Answers

1. Serge Abiteboul; INRIA, France
2. Foto N. Afrati; University of Athens, Greece
3. Marcelo Arenas; Pontificia Universidad Católica de Chile, Chile
4. Magdalena Balazinska; University of Washington, Seattle, Washington, USA
5. Pablo Barceló; University of Chile, Chile
6. Michael Benedikt; University of Oxford, UK
7. Iovka Boneva; University of Lille, France
8. Angela Bonifati; University of Lyon, France
9. Elena Botoeva; Free University of Bozen-Bolzano, Italy
10. Peter Buneman; University of Edinburgh, UK

A tree of classes (unary predicates) from an ontology forms a query template.
In this example, the tree is reduced to the root.
Person.
A tree template can be defined by many trees.

root is the (artificial) root of the created query. The OR sign (the cup, the blue color) says that the set of its children is a disjunctive set.

Person specifies the class of answers. The ANDsign (the cap, the black color) says that the set of its children is a conjunctive set.

No restriction on **Person** is defined.
Properties may be checked and their values may be given.
A pair: (property, value) is a **facet**

Accept the current form of the query template. In result, the **faceted interface** corresponding to this template is created

file:///C:/dafo-client/src/dafo-examples.html

1/20

Query 1: Get all persons (without any restriction)

DAFO - creating and executing faceted queries

1. **Query template (preselection of elements)**
 (Sub)trees constituting the query:
 Roots of subtrees: Person
 Edges of (sub)trees:
 Edge1: _____ → _____
 ACCEPT

2. **Faceted interface:**

 EXECUTE FO & SQL FORMS NESTED ANSWER

3. **Answers**

1. Serge Abiteboul; INRIA, France
 2. Foto N. Afrati; University of Athens, Greece
 3. Marcelo Arenas; Pontificia Universidad Católica de Chile, Chile
 4. Magdalena Balazinska; University of Washington, Seattle, Washington, USA
 5. Pablo Barceló; University of Chile, Chile
 6. Michael Benedikt; University of Oxford, UK
 7. Iovka Boneva; University of Lille, France
 8. Angela Bonifati; University of Lyon, France
 9. Elena Botoeva; Free University of Bozen-Bolzano, Italy
 10. Peter Buneman; University of Edinburgh, UK

"root", is an OR node.

Formulating faceted query

Providing a query template, 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 necessary, 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 response to "EXECUTE". On the bottom of the figure, ten first answers are 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 sign (a cap), the node is interpreted as an AND node. Additionally, 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 means that the set of its children 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 beginning, 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

DAFO - creating and executing faceted queries

FO before rewriting:
 Person(x)

FO after rewriting:
 Person(x)

SQL:
 SELECT x.Id AS Id FROM Person AS x

T. Pankowski, DAFO

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.

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.

top

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

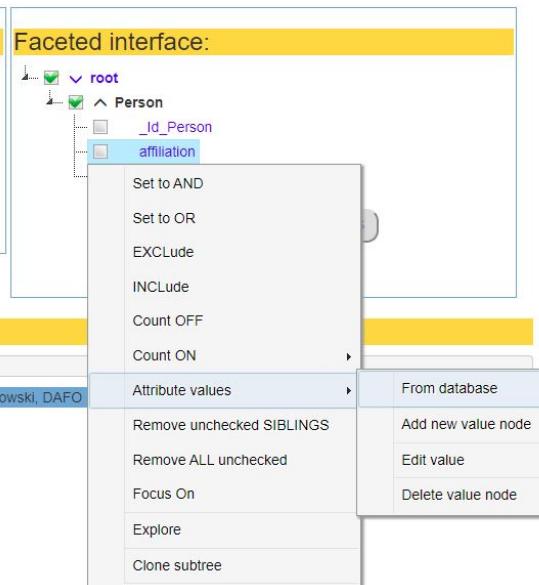
1. Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:
Edge1: ----- → -----

ACCEPT



Answers

T. Pankowski, DAFO

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

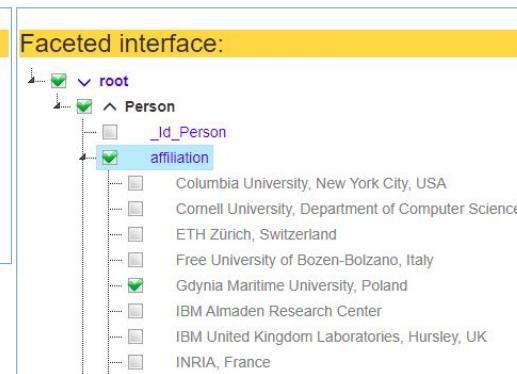
Query template (preselection of elements)

(Sub)trees constituting the query:

2. Roots of subtrees: Person -----

Edges of (sub)trees:
Edge1: ----- → -----

ACCEPT



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 screenshot displays two panels: "Query template (preselection of elements)" and "Faceted interface".

Query template (preselection of elements):

- (Sub)trees constituting the query:
- Roots of subtrees: Person
- Edges of (sub)trees: Edge1
- ACCEPT button

Faceted interface:

- Faceted interface showing a tree structure with nodes like root, Person, _Id_Person, affiliation, Columbia University, New York City, USA, etc.
- A context menu is open over the node "Columbia University, New York City, USA". The menu options include:
 - Set to AND
 - Set to OR
 - EXCLUDE
 - INCLUDE
 - Count OFF
 - Count ON
 - Attribute values
 - Remove unchecked SIBLINGS
 - Remove ALL unchecked** (highlighted in blue)
 - Focus On
 - Explore
 - Clone subtree
- Nodes visible in the tree include: University of Bayreuth, Germany; Chile, Chile; Hursley, UK; Italy; Computer Science; Columbia University, New York City, USA; _Id_Person; Person; root.

Query template (preselection of elements)

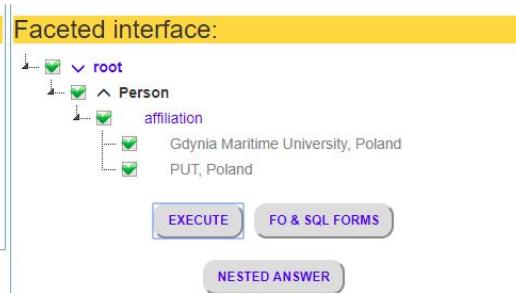
(Sub)trees constituting the query:

Roots of subtrees: Person

Edges of (sub)trees:

Edge1: ----- → -----

ACCEPT

**Answers**

1. Ireneusz Czarnowski; Gdynia Maritime University, Poland
2. Czesław Jędrzejek; PUT, Poland
3. Piotr Jedrzejowicz; Gdynia Maritime University, Poland
4. Tadeusz Morzy; PUT, Poland
5. Tadeusz Pankowski; PUT, Poland
6. Andrzej Szwabe; PUT, Poland
7. Robert Wrembel; PUT, Poland

DAFO - creating and executing faceted queries**SQL:**

```
SELECT DISTINCT x.PersonId AS Id FROM Affiliation AS x WHERE (x.Affiliation = 'Gdynia Maritime University, Poland'
OR x.Affiliation = 'PUT, Poland')
```

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.

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person

Edges of (sub)trees:
Edge1: →

Faceted interface:

```

graph TD
    root["root"] --> Person["Person"]
    Person --> IdPerson["_Id_Person"]
    Person --> affiliation["affiliation"]
    affiliation --> Poland["%Poland%"]
    affiliation --> name["name"]
    
```

Answers

1. Ireneusz Czarnowski; Gdynia Maritime University, Poland
2. Czesław Jedrzejek; PUT, Poland
3. Piotr Jedrzejowicz; Gdynia Maritime University, Poland
4. Filip Mazowiecki; University of Warsaw, Poland
5. Tadeusz Morzy; PUT, Poland
6. Filip Mirlak; University of Warsaw, Poland
7. Tadeusz Pankowski; PUT, Poland
8. Paweł Parys; University of Warsaw, Poland
9. Andrzej Szwabe; PUT, Poland
10. Robert Wrembel; PUT, Poland

DAFO - creating and executing faceted queries

FO before rewriting:

```

graph TD
    Person[Person(x)]
    Person --> affiliation[affiliation(x,x1)]
    affiliation --> Poland[x1 = "%Poland%"]
    
```

FO after rewriting:

```

graph TD
    Person[Person(x)]
    Person --> affiliation[affiliation(x,x1)]
    affiliation --> Poland[x1 = "%Poland%"]
    
```

SQL:

```
SELECT DISTINCT x.PersonId AS Id FROM Affiliation AS x WHERE (x.Affiliation LIKE '%Poland%')
```

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.

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: ACMConf ▾ DEXAConf ▾ ----- ▾

Edges of (sub)trees:
Edge1: ----- ▾ → ----- ▾

ACCEPT

Faceted interface:

root

- ACMConf
 - _Id_Conference
 - acronym
 - city
 - confName
 - confYear
 - country
 - editionNo
- DEXAConf
 - _Id_Conference
 - acronym
 - city
 - confName
 - confYear
 - country
 - editionNo

EXECUTE **FO & SQL FORMS** **NESTED ANSWER**

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

FO before rewriting:

```
OR
  ACMConf(x)
  DEXAConf(x)
```

FO after rewriting:

```
OR
  Conference(x)
    acronym(x,x1)
      x1 = %ACM%
  Conference(x)
    acronym(x,x2)
```

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 DEXAConf.

top

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.

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person

Edges of (sub)trees:

- Edge1: Person → ACMConf
- Edge2: Person → DEXAConf
- Edge3: ----- → -----

Faceted interface:

```

graph TD
    Root --> Person
    Person --> authorConf
    authorConf --> ACMConf
    authorConf --> DEXAConf
  
```

EXECUTE FO & SQL FORMS NESTED ANSWER

ACCEPT

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; University of Athens, Greece
3. Marcelo Arenas; Pontificia Universidad Católica 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:

```

graph TD
    Person --> authorConf
    authorConf --> OR
    OR --> ACMConf
    OR --> DEXAConf
  
```

FO after rewriting:

```

graph TD
    Person --> authorOf
    authorOf --> Paper
    Paper --> inProceedings
    inProceedings --> Proceedings
    Proceedings --> ofConf
    ofConf --> OR
    OR --> Conference1
    Conference1 --> acronym1
    acronym1 --> x4
    x4 --> ACMDexa
    OR --> Conference2
    Conference2 --> acronym2
    acronym2 --> 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_L
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.

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person

Edges of (sub)trees:

- Edge1: Person → ACMConf
- Edge2: Person → DEXAConf
- Edge3: ----- → -----

Faceted interface:

EXECUTE FO & SQL FORMS NESTED ANSWER

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 (first order) semantics.

DAFO - creating and executing faceted queries

FO before rewriting:

FO after rewriting:

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%') ) AS x1_L
AS x1 ON x1.Id = x3.ConfId
AS x3 ON x3.Id = x2.ProcId
AS x2 ON x2.Id = x.PaperId

```

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)

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:

Edge1: Person → ACMConf
 Edge2: Person → DEXAConf
 Edge3: ----- → -----

ACCEPT

Faceted interface:

The faceted interface shows a tree structure with nodes: root, Person, and authorConf. A context menu is open over the authorConf node, listing various operations like Set to AND, Set to OR, EXCLUDE, INCLUDE, Count OFF, Count ON, Attribute values, Remove unchecked SIBLINGS, Remove ALL unchecked, Focus On, Explore, and Clone subtree. The 'Remove unchecked SIBLINGS' option is highlighted in yellow.

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.

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:

Edge1: Person → ACMConf
 Edge2: Person → DEXAConf
 Edge3: ----- → -----

ACCEPT

Faceted interface:

The faceted interface shows the cloned subtree structure where the authorConf node now has two children: ACMConf and DEXAConf. At the bottom, there are buttons labeled EXECUTE, FO & SQL FORMS, and NESTED ANSWER.

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

Answers

1. Serge Abiteboul; INRIA, France
2. Foto N. Afrati; University of Athens, Greece
3. Marcelo Arenas; Pontificia Universidad Católica de Chile, Chile
4. Angela Bonifati; University of Lyon, France

- 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 ACMConf and x2 denoting DEXAConf. And this expresses correctly the expected semantics of the query.

top

Query 8: Persons connected with ACM conferences in the USA

The screenshot shows the DAFO interface with two main sections:

- 1. Query template (preselection of elements):** A form for defining a query template. It includes fields for "Roots of subtrees" (set to Person), "Edges of (sub)trees" (set to Person → ACMConf), and "Edae1" (set to Person → ACMConf).
- Faceted interface:** A tree diagram showing a facet structure. The root node is labeled "root". It has a "Person" node as a child. The "Person" node has an "authorConf" node as a child. A context menu is open over the "authorConf" node, with the option "Set to AND" highlighted.

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.

dafo-examples.html

Edge2: Person → USAConf
Edge3: ----- → -----
ACCEPT

Answers
T. Pankowski, DAFO

Set to OR
EXCLUDE
INCLUDE
Count OFF
Count ON
Attribute values
Remove unchecked SIBLINGS
Remove ALL unchecked
Focus On
Explore
Clone subtree

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----
Edges of (sub)trees:
Edge1: Person → ACMConf
Edge2: Person → USAConf
Edge3: ----- → -----
ACCEPT

Faceted interface:

root
Person
authorConf
ACMConf
USAConf
EXECUTE
FO & SQL FORMS
NESTED ANSWER

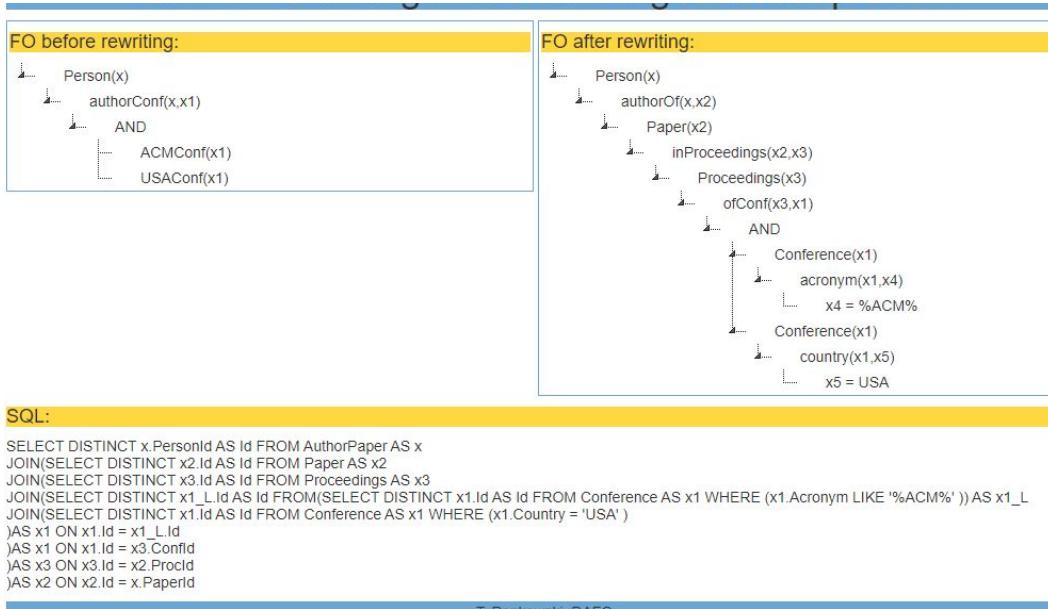
The final form of the faceted query.

Answers

1. Serge Abiteboul; INRIA, France
2. Foto N. Afrati; University of Athens, Greece
3. Marcelo Arenas; Pontificia Universidad Católica 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
7. Peter Buneman; University of Edinburgh, UK
8. Jan Chomicki; University at Buffalo, USA

First order semantics of the query and its SQL form.

top



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 x_1 , where x_1 denotes elements from the range class of `authorConf` (i.e., from the `Conference` class, in the considered example).

The screenshot shows the "Query template (preselection of elements)" and "Faceted interface". The "Query template" section includes fields for "Roots of subtrees" (set to Person), "Edges of (sub)trees" (Edge1: Person → ACMConf, Edge2: Person → USAConf, Edge3: Person →), and an "ACCEPT" button. The "Faceted interface" section shows a tree structure under "root": `root > Person > authorConf`. A context menu is open over the `authorConf` node with options: Set to AND, Set to OR, EXCLUde, and INCLUDE.

Answers

T. Pankowski, DAFO

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:

Edge1: Person → ACMConf
Edge2: Person → USAConf
Edge3: ----- → -----

ACCEPT

Faceted interface:

The final form of the faceted query.

The final form of the faceted query.

Answers

1. Serge Abiteboul; INRIA, France
2. Magdalena Balazinska; University of Washington, Seattle, Washington, USA
3. Alin Deutsch; University of California, San Diego, USA
4. Daniela Florescu; INRIA, France
5. Donald Kossmann; ETH Zürich, Switzerland
6. Tova Milo; Tel Aviv University, Israel
7. Dan Suciu; University of Washington, Seattle, Washington, USA
8. Jeffrey D. Ullman; Stanford University, USA
9. Gerhard Weikum; Max Planck Institute for Informatics, Saarbrücken, Germany

DAFO - creating and executing faceted queries

FO before rewriting:

```

  ↳ Person(x) [count(x1) > 5]
    ↳ authorConf (x,x1)
      ↳ AND
        ↳ ACMConf(x1)
        ↳ USAConf(x1)
  
```

FO after rewriting:

```

  ↳ Person(x) [count(x1) > 5]
    ↳ authorOf(x,x2)
      ↳ Paper(x2)
        ↳ inProceedings(x2,x3)
          ↳ Proceedings(x3)
            ↳ ofConf(x3,x1)
              ↳ AND
                ↳ Conference(x1)
                  ↳ acronym(x1,x4)
                    ↳ x4 = %ACM%
                ↳ Conference(x1)
                  ↳ country(x1,x5)
                    ↳ v5 = USA
  
```

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

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.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_R
AS x1 ON x1.Id = x1_L.Id
AS x1 ON x1.Id = x1_R.Id
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

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:

- Edge1: Person → ACMConf
- Edge2: Person → USAConf
- Edge3: ----- → -----

Faceted interface:

ACCEPT

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.

Answers

T. Pankowski, DAFO

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:

- Edge1: Person → ACMConf
- Edge2: Person → USAConf
- Edge3: ----- → -----

Faceted interface:

ACCEPT

1. The final form of the faceted query

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

ACCEPT

NESTED ANSWER

Answers

1. Donald Kossmann, ETH Zürich, Switzerland

DAFO - creating and executing faceted queries

FO before rewriting:

```

    ↴ Person(x) [count(x1) > 5]
      ↴ authorConf (x,x1)
        ↴ AND
          ↴ ACMConf(x1)
          ↴ NOT
            ↴ USAConf(x1)
  
```

FO after rewriting:

```

    ↴ Person(x) [count(x1) > 5]
      ↴ authorOf(x,x2)
        ↴ Paper(x2)
          ↴ inProceedings(x2,x3)
            ↴ Proceedings(x3)
              ↴ ofConf(x3,x1)
                ↴ AND
                  ↴ Conference(x1)
                    ↴ acronym(x1,x4)
                      ↴ x4 = %ACM%
                  ↴ NOT
                    ↴ Conference(x1)
                      ↴ country(x1,x5)
                        ↴ x5 = USA
  
```

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 x.Id FROM Conference AS x
EXCEPT
(SELECT DISTINCT x1.Id AS Id FROM Conference AS x1 WHERE (x1.Country = 'USA' ))
)AS x_R ON x_R.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
  
```

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

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Person -----

Edges of (sub)trees:

Edge1:	Person	→	Conference
Edge2:	Conference	→	Person
Edge3:	-----	→	-----

Faceted interface:



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

dafo-examples.html

A facetized 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.

Answers

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

DAFO - creating and executing faceted queries

FO before rewriting:

```

    Person(x) [x = x3]
      AND
        _Id_Person(x,x1)
          x1 = @x
        authorConf(x,x2)
        Conference(x2)
          confPCMMember(x2,x3)
            Person(x3)
              _Id_Person(x3,x4)
                x4 = @x
  
```

FO after rewriting:

```

    Person(x) [x = x3]
      authorOff(x,x5)
        Paper(x5)
          inProceedings(x5,x6)
            Proceedings(x6)
              ofConf(x6,x2)
                Conference(x2)
                  pcMemberOf(x3,x2)
                    Person(x3)
  
```

SQL:

```

SELECT DISTINCT x.Id FROM(
SELECT DISTINCT x.PersonId AS Id,x5.x_3_cycle FROM AuthorPaper AS x
JOIN(SELECT DISTINCT x5.Id AS Id, x6.x_3_cycle FROM Paper AS x5
JOIN(SELECT DISTINCT x6.Id AS Id, x2.x_3_cycle FROM Proceedings AS x6
JOIN(SELECT DISTINCT x2.ConfId AS Id, x3.x_3_cycle FROM PCMember AS x2
JOIN(SELECT DISTINCT x3.Id AS Id, x3.Id AS x_3_cycle FROM Person AS x3
)AS x3 ON x3.Id = x2.ConfId
)AS x2 ON x2.Id = x6.ConfId
)AS x6 ON x6.Id = x5.ProcId
)AS x5 ON x5.Id = x.PaperId
)AS x
WHERE x.Id = x.x_3_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 between identifiers is omitted.

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

top

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

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: PUTAuthor -----

Edges of (sub)trees:

Edge1: PUTAuthor -----> DEXAConf
Edge2: -----> -----

Faceted interface:

EXECUTE FO & SQL FORMS

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.

ACCEPT

NESTED ANSWER

Answers

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

Nested answers (expand/collapse items):

- 1. Tadeusz Morzy; PUT, Poland
 - 1. DEXA, 2015, 26. DEXA 2015: Valencia, Spain, Database and Expert Systems Applications (DEXA), 26 , Spain, Valencia
 - 2. DEXA Workshop, 2001, 12. DEXA Workshop 2001: Munich, Germany, DEXA Workshops, 12 , Germany, Munich
 - 3. DEXA Workshop, 1999, 10. DEXA Workshop 1999: Florence, Italy, DEXA Workshops, 10 , Italy, Florence
 - 4. DEXA, 1991, 2. DEXA 1991: Berlin, Germany, Database and Expert Systems Applications (DEXA), 2 , Germany, Berlin
 - 5. DEXA, 1990, 1. DEXA 1990: Vienna, Austria, Database and Expert Systems Applications (DEXA), 1 , Austria, Vienna
- 2. Tadeusz Pankowski; PUT, Poland
- 3. Robert Wrembel; PUT, Poland

T. Pankowski, DAFO

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

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: PUTAuthor

Edges of (sub)trees:

Edge1: PUTAuthor → DEXAConf

Edge2: ----- → -----

ACCEPT

Faceted interface:

The faceted interface shows a tree structure with nodes: root, PUTAuthor, authorConf, and DEXAConf. A context menu is open over the DEXAConf node, listing options such as Set to AND, Focus On, Explore, and Clone subtree.

Answers

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

T. Pankowski, DAFO

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) functionality.

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

Query template (preselection of elements) | **Faceted interface:**

Result of refocussing

(Sub)trees constituting the query:

Roots of subtrees: PUTAuthor -----

Edges of (sub)trees:

Edge1: PUTAuthor ----- → DEXAConf
Edge2: ----- → -----

ACCEPT

The result of refocusing is a faceted query rewritten in such a way that: (a) answers are o the set of objects of the same class before 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 refocusing

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

top

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

1.

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Conference -----

Edges of (sub)trees:

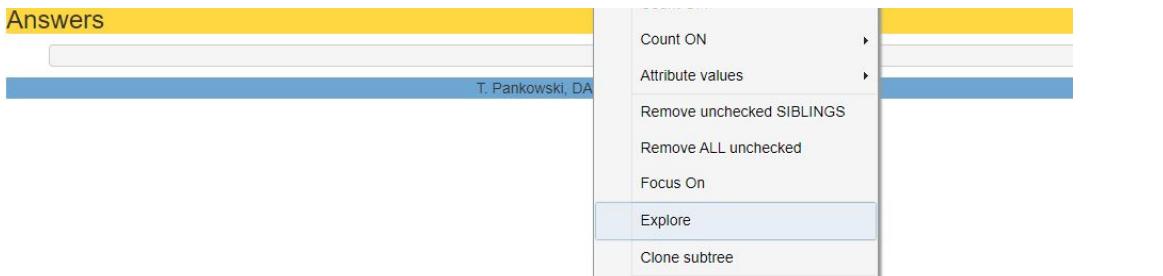
Edge1: ----- → -----

ACCEPT

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 functionality.

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



Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Conference

Edges of (sub)trees:

Edge1: [] → []

ACCEPT

Faceted interface:

root

- Conference
 - confAuthor
 - Author [1799 of connected Conference]
 - PUTAuthor [131 of connected Conference]
 - confPCMember
 - Person [4 of connected Conference]
 - Author [4 of connected Conference]
 - PUTAuthor [4 of connected Conference]
 - confProceedings
 - Proceedings [1824 of connected Conference]
 - SpringerProceedings [713 of connected Conference]
 - presentedPaper
 - Paper [1799 of connected Conference]
 - DBPaper [281 of connected Conference]
 - ONTOPaper [145 of connected Conference]

EXECUTE FO & SQL FORMS NESTED ANSWER

Result of exploring

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

Query template (preselection of elements)

(Sub)trees constituting the query:

Roots of subtrees: Conference

Edges of (sub)trees:

Edge1: [] → []

ACCEPT

Faceted interface:

root

- Conference
- presentedPaper
 - DBPaper [281 of connected Conference]

EXECUTE FO & SQL FORMS NESTED ANSWER

Selection based on exploring information

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

top