# Appendix

## Appendix 1: NP-hardness proof of the SOS problem

Let us start by some recalls on how to prove that a given problem is in the NP-hard class.

**Theorem 1.** *The Stars ordering and Selection (SOS) problem is NP-Hard.*

We say that an optimization problem A is NP-hard if: [2]

- A belongs to NP: There exists a polynomial algorithm allowing to verify that a candidate is a valid solution of the problem A.
- a decision problem $A^{'}$ related to A is NP-Complete.

**Definition 1.** *(Decision problem) A decision problem [2] is a problem with only two possible instances: "Yes" or "No"*

To prove that a problem $A^{'}$ is NP-Complete, we have to reduce a well known NP-complete problem to our problem, *i.e.*, all inputs of the well known NP-complete problem can be represented as special cases of the problem $A^{'}$ and a solution of $A^{'}$ is a solution of the well known NP-complete problem. The last part of the proof consists in proving the correctness of the reduction.

*Proof.* The decision problem related to SOS problem is defined as follows:

**Definition 2.** *(SOS decision ($SOS_d$) problem)*

- ***Instance****: Let q be a query and C an integer*
- ***Question****: Is there any acceptable plan $\mathcal{P} = < X, f >$ of q such that:*

$$Cost_{net}(X, f) \leq 1 + 2\sum_{i=1}^{C}\left(\frac{1}{2}\right)^{i}$$

*Where,*

$$Cost_{net}(X, f) = \sum_{qs \in \ X} \mathcal{NC}(qs)$$

*With respect to the formula in 1, we take $T_{TR} = 1$ in order to simplify the proof.*

To prove the theorem 1, we make use of the vertex cover (VC) problem [1] as the well known NP-complete problem.

**Definition 3.** *(Vertex cover problem)*

- *INSTANCE: a graph G = (V,E), and a positive integer k.*
- *QUESTION: does G have a subset $V^{'}$, with a size less than or equal to k, of V such that if edge (u, v) is an edge of G, then u is in $V^{'}$, or v is in $V^{'}$, or both.*

For the rest of the proof we assume that:

- $V = \{v_1, ..., v_p\}$
- Edges are of the form $(v_i, v_j)$ where i>j.
- G does not contain edges of the form $(v_i, v_i)$
- $k < p - 1$, otherwise VC is trivial *i.e.*, V is a solution.

The transformation concerns two parts. In the first part, we discuss how to specify the query, and its associated components, from the graph of VC. In the second part, we discuss statistics (*i.e.*, selectivity factors, number of data stars, ...) we used to choose the best plan.

We start by query components creation:

- Variables of q: $Y_a, Y_b, Y_1, ..., Y_p$, each node of the graph of VC is associated with a variable in q. We also add two special variables, *i.e.*, $Y_a$ and $Y_b$.
- Triples of q: $\{(Y_a, P_{ai}, Y_i) | 1 \leq i \leq p\} \cup \{(Y_i, P_{ib}, Y_b) | 1 \leq i \leq p\} \cup \{(Y_i, P_{ij}, Y_j) | (v_i, v_j) \in E\}$ for each arc in the graph of VC, we add a triple to the query. We also add a triples from $Y_a$ to all nodes of the graph of VC and from all the nodes of the graph of VC to $Y_b$.
- We use the notation $\overrightarrow{Y_\alpha}$ to represent forward query star and $\overleftarrow{Y_\alpha}$ to represent backward star query where $\alpha \in \{a, b, 1, ..., p\}$ to represent forward from backward query stars. We also use $\overleftrightarrow{Y_\alpha}$ to say that the related property can be applied to forward and backward query star with the head $Y_\alpha$.
- The set of forward query stars: $\overrightarrow{SQ_q} = \left\{ \overrightarrow{Y_\alpha} | \alpha \in \{a, b, 1, ..., p\} \right\}$.
- The set of backward query stars: $\overleftarrow{SQ_q} = \left\{ \overleftarrow{Y_\alpha} | \alpha \in \{a, b, 1, ..., p\} \right\}$.

For graph fragment statistics, we assume that:

- We associate each star query with one and only one graph fragment. We use the notation $\{Gf_a, Gf_b, Gf_1, Gf_2, ..., Gf_p\}$ to designate the set of graph fragments.
- Graph fragment are hosted by different machines, *i.e.*, we can not find two graph fragment in the same machine.
- SF($\overrightarrow{Gf_a}$,$\overleftrightarrow{Gf_i}$, $P_{ai}$)=1: selectivity factor associated with the forward query star with the head a and the query star have with a head in V (vertices of the graph of VC). With the symbol ($\leftrightarrow$, we indicate that the fragment graph can be forward or backward).
- SF($\overleftrightarrow{Gf_i}$,$\overrightarrow{Gf_a}$, $P_{ia}$)=1/5: selectivity factor between a query star has a head from V and the forward query star with the head a.
- SF($\overleftrightarrow{Gf_i}$,$\overleftrightarrow{Gf_j}$, $P_{ij}$)=1: selectivity factor associated with any two star queries that have a head from V.
- SF($\overleftarrow{Gf_i}$,$\overrightarrow{Gf_i}$, $-1$)=1/2: selectivity factor associated with two star queries that have the same head.
- SF($\overrightarrow{Gf_i}$,$\overleftarrow{Gf_i}$, $-1$)=1/2: selectivity factor associated with two star queries that have the same head.
- SF($\overleftrightarrow{Gf_b}$,$\overleftarrow{Gf_\alpha}$, $P_{b\alpha}$)=1: selectivity factor associated with the query star with a head b and another query star.
- SF($\overleftrightarrow{Gf_i}$,$\overleftrightarrow{Gf_b}$, $P_{ib}$)=$2^{k+3}$: selectivity factor between any query star and the query star with the head b.
- dist($\overleftrightarrow{Gf_i}$)=$2^{k+3}$: number of data stars in a graph fragment related to a head from V.
- dist($\overleftrightarrow{Gf_a}$)=1: number of data stars associated to the graph fragment related to the star query with the head a.

– dist($\overleftrightarrow{Gf_b}$)=1: number of data stars associated to the graph fragment related to the star query with the head b.

We also set C to k.

We must now show that this transformation is correct, *i.e.*, a solution of the instance of our problem is also a solution of the VC problem and a solution of VC is also a solution to the instance of our problem.

Let $X = \{v_{x_1}, ..., v_{x_k}\}$ be the witness that VC (Vertex Cover) has a solution. We have to show that there is an acceptable plan $P_y =< Y, f >$ of $q_G$ (the query built from the graph of VC) such that:

$$Cost_{net}(Y, f) \leq 1 + 2\sum_{i=1}^{k}\left(\frac{1}{2}\right)^i$$

We start by considering a particular plan:

$$P_y = \left[\overrightarrow{V_a}, \overrightarrow{Y_{x_1}}, \overleftarrow{Y_{x_1}}, ..., \overrightarrow{Y_{x_k}}, \overleftarrow{Y_{x_k}}, \overleftarrow{V_b}\right]$$

where for each vertex $x_i$ in X, we associate the two star queries that have x as a head. The forward query star will be followed by the backward star with the head. We also put the forward star with the head a as the first star in the plan and backward star with the head b as the last star of the plan.

We know that : 1) the distinct of $\overrightarrow{Gf_a}$ is 1, 2) each time we add a query star with a different head we have an SF equal to 1, and 3) each time we add a query star with a same head we have an SF equal to $\frac{1}{2}$.

We then calculate the cost of this plan as follows:

$$\begin{aligned} Cost_{NET}(P_y) &= 1 * 1 + 1 * \left(\frac{1}{2}\right) + \left(\frac{1}{2}\right) * (1) + \left(\frac{1}{2}\right) * \left(\frac{1}{2}\right) + \left(\frac{1}{4}\right) * (1) + \\ &... + \left(\frac{1}{2^{k-1}}\right) + \left(\frac{1}{2^k}\right) + \left(\frac{1}{2^k}\right) \\ &= 1 + 2\sum_{i=1}^{k}\left(\frac{1}{2}\right)^i \end{aligned}$$

Regarding the first condition of an acceptable plan, it is obvious that all triples are covered by the graph.

Concerning the second condition of an acceptable plan of definition, if we start with $Y_a$, we can, at any time, trigger the execution of query star. It is obvious that this condition is verified.

$P_y$ is acceptable and it is obvious that $Cost_{Net}(P_y) \leq 1 + 2\sum_{i=1}^{k}\left(\frac{1}{2}\right)^i$. We can deduce that if VC has a solution, decision problem related to plan selection and ordering problem has also a solution.

Let us now consider the other direction, *i.e.*, we have a solution of the instance of our problem and we have to prove that there exists a solution of VC.

Assume $P_y =< Y, f >= [Y_0, Y_1, ..., Y_l]$ is an acceptable plan and $Net_{cost}(P_y) \leq 1 + 2\sum_{i=1}^{c}\left(\frac{1}{2}\right)^i$

Let $X = \left\{ v_\alpha | \overleftrightarrow{Y_\alpha} \in Y \right\}$

X contains query star heads from the plan $P_y$. We use the notation $\overleftrightarrow{Y_\alpha}$ to say that $Y_\alpha$ is backward or forward.

First, we have to show that X cover E (*i.e.*, the set of edges of the graph of VC). Regarding the first condition of an acceptable plan, we can easily, by replacing star cover by vertex cover show that X covers E. In the rest of the proof, we have to show that $|X| \leq k$.

Let n be the number of variables in Y. We show that $cost_{Net}(P_y) \geq 1 + 2 \sum_{i=1}^{n} \left(\frac{1}{2}\right)^i$

First, we start by computing $cost_{Net}$ regarding selectivity factors.

We have:

$$\begin{aligned} Cost_{NET}(P_y) &= dist(Gf_{Y_0}) * SF(Gf_{Y_0}, Gf_{Y_1}, *) + dist(Gf_{Y_0}) * SF(Gf_{Y_0}, Gf_{Y_1}, *) * SF(Gf_{Y_1}, Gf_{Y_2}, *) \\ &+ ... + dist(Gf_{Y_0}) * SF(Gf_{Y_0}, Gf_{Y_1}, *) * SF(Gf_{Y_1}, Gf_{Y_2}, *) * ... * SF(Gf_{|Y|-1}, Gf_{|Y|}, *) \\ &= dist(Gf_{Y_0}) * (1 + \sum_{i=1}^{|Y|} SF(Gf_{Y_{i-1}}, Gf_{Y_i})) \end{aligned}$$

where $Y_0$ is the first star query in the plan.

We know that Y has less than p variables. This implies that at least forward star and backward star formed from a variable from the query is not involved in the plan. Since $Y_a$ has an edge to all nodes in the graph, the only way to get a valid plan (by satisfying the first condition) is to put $Y_a$ in the plan.

Now we will determine the position of $Y_a$. We use the formula in 1.

Let us suppose that: $Y = [Y_0, Y_1, ..., Y_a, .., Y_l]$

$cost_{Net}(Y_p) = 2^{k+3} + 2^{k+3} * ...$

It's obvious that $Net_{cost}(Y_p)$ exceeds $1 + 2 \sum_{i=1}^{k} \left(\frac{1}{2}\right)^i$, indeed $2 \sum_{i=1}^{k} \left(\frac{1}{2}\right)^i$ converges to 2.

The only proposal we left is to put $Y_a$ in the first position.

We know that Y has less than p variables, which means that at least forward star and backward star formed from a variable from the query is not involved in the plan. Since $Y_b$ has an edge to all nodes in the graph. The only way to get a valid plan (by satisfying the first condition) is to put $Y_\beta$ in the plan.

Now we will determine the position of $Y_\beta$. Note that $SF(Y_i, Y_\beta, P_{i\beta})$=$2^{k+3}$, this implies that if we put $Y_\beta$ in a position different from the last position, the cost will exceed the defined limit, namely 2.

We know that Y is in the form $\left[\overrightarrow{Y_a}, ..., \overleftarrow{Y_b}\right]$

We will determine the lower bound of the cost of Y.

Suppose we have two plans $Y$ and $Y^{'}$, with:

$$P_Y = \left[..., \overleftrightarrow{Y_{x_h}}, \overleftrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_j}}, ...,\right]$$

and

$$P_{Y'} = \left[..., \overleftarrow{Y_{x_i}}, \overrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_j}}, ...,\right]$$

With $x_i \neq x_h \neq x_j$

Note that we constructed $P_{Y'}$, by replacing in $P_Y$, $\overleftrightarrow{Y_{x_h}}$ by $\overrightarrow{Y_{x_i}}$ and $\overleftrightarrow{Y_{x_i}}$ by $\overleftarrow{Y_{x_i}}$. We want to show the impact of putting two stars with the same heads one after the other.

Let C be the cost we obtained before evaluating $\overleftrightarrow{Y_{x_h}}$ and $C^{'}$ be the cost we obtained after evaluating $\overleftrightarrow{Y_{x_j}}$.

Using the formula in 1, we obtain:

$Cost_{Net}(P_Y) = C + C * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) + C * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_i}}, *) + C * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) * SF(\overleftrightarrow{Y_{x_h}}, \overleftrightarrow{Y_{x_i}}, *) * SF(\overleftrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_j}}, *) + C^{'}$

We know that:

$$SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) = 1$$

$$SF(\overleftrightarrow{Y_{x_h}}, \overleftrightarrow{Y_{x_i}}, *) = 1$$

$$SF(\overleftrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_j}}, *) = 1$$

which implies that:

$$Cost_{Net}(P_Y) = C + C + C + C + C^{'}$$

In the same vein, for $Y^{'}$, we get :

$Cost_{Net}(Y^{'}) = C + C * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) + C * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_i}}, *) + C * SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_h}}, *) * SF(\overleftrightarrow{Y_{x_h}}, \overleftrightarrow{Y_{x_i}}, *) * SF(\overleftrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_j}}, *) + C^{''}$

We know that:

$$SF(\overleftrightarrow{Y_{x_{h-1}}}, \overleftrightarrow{Y_{x_i}}, *) = 1$$

$$SF(\overleftrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_i}}, *) = 1/2$$

$$SF(\overleftrightarrow{Y_{x_i}}, \overleftrightarrow{Y_{x_j}}, *) = 1$$

$$Cost(P_{Y'}) = C + 1/2 * C + 1/2 * C + 1/2 * C + C^{''}$$

It is obvious that $C^{''} < C^{'}$, which implies that: $Cost(Y_{'}) \leq Cost(Y)$

We can deduce that any plan has a higher or equal cost than

$$P_{Y_2} = \left[\overrightarrow{Y_a}, \overrightarrow{Y_{x_1}}, \overleftarrow{Y_{x_1}}, ..., \overrightarrow{Y_{x_n}}, \overleftarrow{Y_{x_n}}, \overrightarrow{Y_b}\right]$$

$P_{Y_2}$ is obtained by putting $\overrightarrow{Y_{x_i}}$ (or $\overleftarrow{Y_{x_i}}$) directly after the opposite query star, i.e., $\overleftarrow{Y_{x_i}}$ (or $\overrightarrow{Y_{x_i}}$).

We use the cost of $P_{Y_2}$ as the lower bound of the cost.

We will now compute the cost of $P_{Y_2}$.

$$Cost(P_{Y_2}) = 1 + \frac{1}{2} + ... + \frac{1}{2^{n-2}} + \frac{1}{2^{n-1}} + \frac{1}{2^{n-1}} + \frac{1}{2^{n-1}} = 1 + \sum_{i=1}^{n} \left(\frac{1}{2}\right)^i$$

We know that $1 + \sum_{i=1}^{n} \left(\frac{1}{2}\right)^i = Cost(P_{Y_2}) \leq Cost(P_Y) \leq 1 + \sum_{i=1}^{k} \left(\frac{1}{2}\right)^i$
We can induce that:

$$n \leq k$$

We can deduce that X is a vertex cover with size $\leq k$

## References


1. Karp, R.M.: Reducibility among combinatorial problems. In: Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. pp. 85–103 (1972)
2. van Leeuwen, J. (ed.): Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity. Elsevier and MIT Press (1990)

## Appendix 2: Experimental Queries

### *Watdiv*

*1.*

```
SELECT ?v0 ?v1 ?v2 WHERE
{
    ?v0 <http://purl.org/ontology/mo/conductor> ?v1 .
    ?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v2 .
    ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/hasGenre> <http://db.uwaterloo.ca/~galuc/wsdbm/SubGenre83> .
}
```

*2.*

```
SELECT ?v0 ?v2 ?v3 WHERE
{
    ?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://db.uwaterloo.ca/~galuc/wsdbm/ProductCategory7> .
    ?v0 <http://schema.org/description> ?v2 .
    ?v0 <http://schema.org/keywords> ?v3 .
    ?v0 <http://schema.org/language> <http://db.uwaterloo.ca/~galuc/wsdbm/Language0> .
}
```

*3.*

```
SELECT ?v0 ?v1 WHERE
{
    ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/likes> ?v1 .
    ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/subscribes> ?v2 .
}
```

*4.*

```
SELECT ?v0 ?v2 ?v3 WHERE
{
    ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/subscribes> <http://db.uwaterloo.ca/~galuc/wsdbm/Website18627> .
    ?v2 <http://schema.org/caption> ?v3 .
    ?v0 <http://db.uwaterloo.ca/~galuc/wsdbm/likes> ?v2 .
}
```

*5.*

```
SELECT ?v0 ?v2 ?v3 ?v4 ?v5 WHERE
{
    ?v0 <http://ogp.me/ns#tag> <http://db.uwaterloo.ca/~galuc/wsdbm/Topic172> .
    ?v0 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ?v2 .
    ?v3 <http://schema.org/trailer> ?v4 .
    ?v3 <http://schema.org/keywords> ?v5 .
    ?v3 <http://db.uwaterloo.ca/~galuc/wsdbm/hasGenre> ?v0 .
    ?v3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://db.uwaterloo.ca/~galuc/wsdbm/ProductCategory2> .
}
```

### *LUBM*

*1.*

```
SELECT ?x WHERE
{
    ?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://swat.cse.lehigh.edu/onto/univ-bench.owl#GraduateStudent> .
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#takesCourse> <http://www.Department0.University0.edu/GraduateCourse0> .
}
```

*2.*

```
SELECT ?x WHERE
{
    ?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://swat.cse.lehigh.edu/onto/univ-bench.owl#Publication> .
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#publicationAuthor> <http://www.Department0.University0.edu/AssistantProfessor0> .
}
```

*3.*

```
SELECT ?x ?y1 ?y3 WHERE
{
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#worksFor> <http://www.Department0.University0.edu> .
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#name> ?y1 .
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#emailAddress> ?y2 .
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#telephone> ?y3 .
}
```

*4.*

```
SELECT ?x WHERE
{
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#memberOf> <http://www.Department0.University0.edu> .
}
```

*5.*

```
SELECT ?x WHERE
{
    ?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://swat.cse.lehigh.edu/onto/univ-bench.owl#UndergraduateStudent> .
}
```

*6.*

```
SELECT ?x WHERE
{
    ?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://swat.cse.lehigh.edu/onto/univ-bench.owl#GraduateStudent> .
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#takesCourse> <http://www.Department0.University0.edu/GraduateCourse0> .
}
```

*7.*

```
SELECT ?x WHERE
{
    ?x <http://swat.cse.lehigh.edu/onto/univ-bench.owl#undergraduateDegreeFrom> <http://www.University0.edu> .
}
```

*8.*

```
SELECT ?x WHERE
{
    ?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://swat.cse.lehigh.edu/onto/univ-bench.owl#UndergraduateStudent> .
}
```

### *Yago*

*1.*

```
SELECT ?GivenName ?FamilyName WHERE
{
    ?p <http://yago-knowledge.org/resource/hasGivenName> ?GivenName .
    ?p <http://yago-knowledge.org/resource/hasFamilyName> ?FamilyName .
    ?p <http://yago-knowledge.org/resource/wasBornIn> ?city .
    ?p <http://yago-knowledge.org/resource/hasAcademicAdvisor> ?a .
    ?a <http://yago-knowledge.org/resource/wasBornIn> ?city .
}
```

*2.*

```
SELECT ?GivenName ?FamilyName WHERE
{
    ?p <http://yago-knowledge.org/resource/hasGivenName> ?GivenName .
    ?p <http://yago-knowledge.org/resource/hasFamilyName> ?FamilyName .
    ?p <http://yago-knowledge.org/resource/wasBornIn> ?city .
    ?p <http://yago-knowledge.org/resource/hasAcademicAdvisor> ?a .
    ?a <http://yago-knowledge.org/resource/wasBornIn> ?city .
    ?p <http://yago-knowledge.org/resource/isMarriedTo> ?p2 .
    ?p2 <http://yago-knowledge.org/resource/wasBornIn> ?city .
}
```

*3.*

```
SELECT ?name1 ?name2 WHERE
{
    ?a1 <http://yago-knowledge.org/resource/hasPreferredName> ?name1 .
    ?a2 <http://yago-knowledge.org/resource/hasPreferredName> ?name2 .
    ?a1 <http://yago-knowledge.org/resource/actedIn> ?movie .
    ?a2 <http://yago-knowledge.org/resource/actedIn> ?movie .
}
```

*4.*

```
SELECT ?name1 ?name2 WHERE
{
    ?p1 <http://yago-knowledge.org/resource/hasPreferredName> ?name1 .
    ?p2 <http://yago-knowledge.org/resource/hasPreferredName> ?name2 .
    ?p1 <http://yago-knowledge.org/resource/isMarriedTo> ?p2 .
    ?p1 <http://yago-knowledge.org/resource/wasBornIn> ?city .
    ?p2 <http://yago-knowledge.org/resource/wasBornIn> ?city .
}
```

*5.*

```
SELECT ?name1 ?name2 WHERE
{
    ?p1 <http://www.w3.org/2000/01/rdf-schema#label> ?name1 .
    ?p2 <http://www.w3.org/2000/01/rdf-schema#label> ?name2 .
    ?p1 <http://yago-knowledge.org/resource/isMarriedTo> ?p2 .
    ?p1 <http://yago-knowledge.org/resource/wasBornIn> ?city .
    ?p2 <http://yago-knowledge.org/resource/wasBornIn> ?city.
}
```

## *DBLP*

*1.*

```
SELECT ?v0 ?v1 ?v2 WHERE
{
    ?v0 <http://purl.org/dc/elements/1.1/title> ?v1 .
    ?v0 <http://www.w3.org/2002/07/owl#sameAs> ?v3 .
    ?v0 <http://purl.org/dc/terms/bibliographicCitation> ?v2 .
}
```

*2.*

```
SELECT ?v0 ?v1 ?v3 WHERE
{
    ?v0 <http://www.w3.org/2000/01/rdf-schema#seeAlso> ?v1 .
    ?v3 <http://xmlns.com/foaf/0.1/page> ?v1 .
}
```

*3.*

```
SELECT ?v0 ?v3 ?v1 WHERE
{
    ?v0 <http://www.w3.org/2000/01/rdf-schema#seeAlso> ?v1 .
    ?v3 <http://xmlns.com/foaf/0.1/homepage> ?v1 .
}
```

*4.*

```
SELECT ?v0 ?v4 ?v6 WHERE
{
    ?v0 <http://purl.org/dc/terms/tableOfContent> ?v1 .
    ?v0 <http://swrc.ontoware.org/ontology#editor> <http://dblp.l3s.de/d2r/resource/authors/Rodney_W._Topor> .
    ?v0 <http://swrc.ontoware.org/ontology#number> ?v3 .
    ?v0 <http://www.w3.org/2000/01/rdf-schema#label> ?v4 .
    ?v0 <http://purl.org/dc/terms/issued> ?v5 .
    ?v0 <http://purl.org/dc/terms/bibliographicCitation> ?v6 .
}
```

*5.*

```
SELECT ?v0 ?v1 WHERE
{
    ?v0 <http://www.w3.org/2000/01/rdf-schema#seeAlso> <http://dblp.l3s.de/d2r/resource/publications/books/ph/Shasha92> .
    ?v0 <http://purl.org/dc/terms/tableOfContent> ?v1 .
}
```

*6.*

```
SELECT ?v0 ?v1 WHERE
{
    ?v0 <http://purl.org/dc/terms/tableOfContent> ?v1 .
    ?v0 <http://purl.org/dc/terms/bibliographicCitation> <http://dblp.uni-trier.de/rec/bibtex/series/cogtech/Helbig2006> .
}
```