

# Count Aggregation in Semantic Queries

Bogdan Kostov, Petr Křemen

Department of Cybernetics, Czech Technical University,  
Prague, Czech Republic  
{bogdan.kostov, petr.kremen}@fel.cvut.cz

**Abstract.** In this paper we study the *distinct count* aggregation function used in queries into expressive ontologies. The main differences in this settings opposed to aggregation in relational database systems are the Open World Assumption and incomplete knowledge. We propose different interpretations useful in different practical use-cases of the *distinct count* function, i.e. *basic count*, *semantic count*, *epistemic count* and *semantic tuple count* some of which use knowledge derived from the ontology in order to obtain results in accordance with the Open World Assumption. We use *interval semantics* to model the uncertainty of the *distinct count* function's result induced by incomplete knowledge in the ontology. We show that *interval semantics* are particularly useful in aggregate queries with filtering clause and when we need the retrieval of the boundaries of the uncertainty of the *distinct count* function's result. We study and present relationships among the different interpretations and decidability of the *semantic tuple count*. We also propose a theoretical approximation of the *semantic tuple count*. Our results are applicable to a wide range of description logic formalisms allowing to express equality/inequality between individuals, concepts and relations, e.g. Web Ontology Language.

**Keywords:** semantic counting, count function, OWL, semantic query

## 1 Introduction

Aggregation functions are an important feature of modern query languages. Aggregation is well understood in query languages of database systems with Unique Name Assumption (UNA) principle, e.g. SQL for relational databases. Although this simple well-known interpretation of aggregation can in some cases be used correctly in queries over semantic knowledge bases, in general this interpretation will return incorrect results as it will not use knowledge inferred from the knowledge base with the Open World Assumption (OWA). As opposed to that, semantic aggregation relies on the knowledge inferred from the knowledge base in order to provide correct result of aggregation functions.

Currently the research field of semantic aggregation is in its infant state. However the need of semantic aggregation is becoming apparent as more and more applications of semantic technologies emerge. In this paper we propose the need of different interpretations of the *distinct count function* in the settings

of OWA and ontology representing incomplete knowledge, i.e. *basic count* ( $\mathcal{BC}$ ), *semantic tuple count* ( $\mathcal{STC}$ ), *epistemic count* ( $\mathcal{EC}$ ) and *semantic count* ( $\mathcal{SC}$ ). We study the relationships among the different interpretations. We also investigate the decidability of the  $\mathcal{STC}$  interpretation.

## 2 Related Work

In recent years, expressive query languages, like SPARQL-DL [1], OWL-SAIQL [2], SQWRL [3] for OWL 2 [4] or SeRQL [5], SPARQL [6] for RDF, have been introduced and implemented in the field of semantic web. There have been deep studies [7–11] evaluating conjunctive queries in RDF and OWL, but few efforts have been spent on an algebra, as well as aggregation functions.

Recently the RDF query language SPARQL [6] has been extended towards the new SPARQL 1.1 <sup>1</sup> [12] including many new constructs, e.g. aggregation functions or negation as failure. There are already publicly available implementations, e.g. ARQ [13], KGRAM [14], RDF::Query [15] or Sesame [16]. Independently on SPARQL 1.1 authors of [17] discuss the topic of aggregation over data structured as RDF graphs rather than on the relational data returned by the query (the result set table). The authors stress the need of different modes of aggregation. A few years ago the SQWRL query language for OWL [3] was proposed. All these efforts implement or interpret aggregation using the basic relational semantics, i.e. the result of aggregation functions is computed over the results of the non-aggregate variant of the query and neglect the impact of the interplay between aggregation functions and the inferred domain elements.

More closely related to our study is the work [18]. It shows that *exact semantics*, of aggregate queries over a  $DL-Lite_A$  ontology returns results only if the Abox is not empty and if axioms in the Tbox resolve as constraints over cardinalities of the *groups* in the aggregate query. The authors propose *epistemic semantics* of aggregate queries in the settings of data integration use-cases returning the aggregate function’s results known from (inferred by) the ontology. The authors propose an evaluation algorithm for a sub set of aggregate queries, i.e. restricted epistemic aggregate queries, defined using functional dependency of query variables w.r.t. the Tbox of the queried ontology.

Although not exactly in the same framework of query answering like in this paper, the authors in [19] propose an approach using ontology design pattern for incorporating a quantification over types without actually using explicit numerical information.

In this paper we examine the *distinct count* aggregation function, its possible interpretations and their use-cases. We focus our research on the *distinct count* aggregation function alone as it is non-trivial but not too overly complicated. We believe that better understanding of the *distinct count* function and its possible interpretations will cover most of the peculiarities of aggregation in the context of expressive knowledge bases assuming the OWA principle and incomplete knowledge, thus contributing to OWL and RDF.

---

<sup>1</sup> SPARQL 1.1 version recently became a W3C recommendation.

### 3 Motivation

In this section we present a simple ontology which describes a simple taxonomy of teachers categories and contains assertions about teachers and courses. The ontology is presented using the well known description logic syntax, see [20].

*Example 1.* A simple example ontology  $\mathcal{O}_1$  about teachers and the courses they teach.

#### **Tbox**

$\text{BusyTeacher} \sqsubseteq \text{Teacher}$ ,  $\text{Professor} \sqsubseteq \text{Teacher}$ ,  $\exists \text{teaches} \cdot \text{Course} \sqsubseteq \text{Teacher}$ ,  
 $\text{BusyTeacher} \sqsubseteq \geq 3 \text{ teaches}$ ,  $\text{Professor} \sqsubseteq \leq 3 \text{ teaches}$

#### **Abox**

$\text{BusyTeacher}(\text{Sara})$ ,  $\text{Professor}(\text{Steve})$ ,  $\text{Professor}(\text{John})$ ,  $\text{BusyTeacher}(\text{John})$ ,  
 $\text{Course}(\text{math})$ ,  $\text{Course}(\text{physics})$ ,  $\text{Course}(\text{history})$ ,  $\text{teaches}(\text{Dave}, \text{math})$ ,  
 $\text{teaches}(\text{Dave}, \text{physics})$ ,  $\text{teaches}(\text{Dave}, \text{history})$ ,  $\text{teaches}(\text{Sara}, \text{history})$ ,  
 $\text{math} \neq \text{history}$

We will use the following two aggregate queries throughout the paper to demonstrate the differences in the results of the individual interpretations of the *distinct count* function.

$Q_1$  - Find all teachers and the number of distinct courses they teach.

$Q_2$  - Find all teachers that teach more than one distinct courses.

Next we will discuss the need of different interpretations of the *distinct count* function. The most natural interpretation of the *distinct count* function is the *semantics count* interpretation. This interpretation enables users to query for the *distinct count* function value or its constraints as entailed by the queried ontology. For example using this interpretation in query  $Q_1$  we obtain that **John** teaches exactly three courses or using it in query  $Q_2$  we will obtain that **Dave**, **Sara** and **John** teach more than one course. We call this the *semantics count* interpretation and we consider it suitable for knowledge retrieval oriented use-cases. This interpretation is a natural extension of the certain answer semantics for the *distinct count* function and it is monotonic.

Although that is the most natural and in fact correct extension of the semantics of the *distinct count* function, there are some use-cases that need different CWA interpretations. As argued in [18] for ontology based data access (OBDA) use-cases, the certain answer semantics for aggregate queries is not practical as they return trivial results, e.g. empty or very restricted result sets. The authors propose that in this context a practical interpretation will be the one that returns the least known number of courses they teach. This is called the *epistemic count* interpretation. As opposed to the *semantic count*, the result set of query  $Q_1$  with the *epistemic count* interpretation will contain for example that **Dave** and **Sara** teach respectively two and three courses.

Note that the *epistemic count* interpretation may count both named and unnamed entities. This makes the use of this interpretation inappropriate in purely data-centric ontology applications. In [21] the authors use OWL to model

integrity constraints (IC) and propose IC CWA semantics to enable instance data validation. We propose an extension to this semantics for the *distinct count* function. We call this the *semantic tuple count* interpretation. As opposed to the previous interpretations the result of  $Q_2$  with the *semantic tuple count* interpretation will contain only **Dave** as he is the one for which the data in the ontology  $\mathcal{O}_1$  satisfies the condition in the query. The *semantic tuple count* can be used to depict IC for n-ary relations.

The most common interpretation of the *distinct count* function is the *basic count* interpretation. This interpretation is scalable and it is safe to be used in data oriented use-cases in CWA and UNA systems. The usage of this interpretation in applications assuming OWA may return incorrect results, however, it can still be used as an approximation. For example the result of query  $Q_1$  contains the answer **Dave** teaches three courses, which may or may not be true according to the ontology.

We continue with the formal definition of the different interpretations of the *distinct count* function and the discussion of the results of queries  $Q_1$  and  $Q_2$  in these interpretations.

## 4 Preliminaries

In this section we will define basic terms and notions used in the rest of the paper. We will start with the definition of ontology followed by the definition conjunctive queries and aggregate queries with *distinct count* function.

### 4.1 Ontology

**Definition 1.** An ontology  $\mathcal{O}$  is a pair  $\langle S, A \rangle$ , where  $S$  is a signature and  $A$  is a set of axioms. The semantics of ontologies use a first order interpretation  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ , where  $\Delta^{\mathcal{I}}$  is an interpretation domain and  $\cdot^{\mathcal{I}} : S \rightarrow \Delta^{\mathcal{I}}$  is an interpretation function mapping elements from the ontology signature  $S$  to elements from the interpretation domain  $\Delta^{\mathcal{I}}$ . An ontology  $\mathcal{O}$  is satisfied by an interpretation  $\mathcal{I}$ , denoted by  $\mathcal{I} \models \mathcal{O}$ , if all of its axioms are satisfied by the interpretation  $\mathcal{I}$ , such interpretation  $\mathcal{I}$  is called a model of  $\mathcal{O}$ . We say that a set of axioms  $A$  is entailed by the ontology  $\mathcal{O}$ , denoted by  $\mathcal{O} \models A$ , if every model  $\mathcal{I}$  of the ontology  $\mathcal{O}$ , is also a model of  $A$ ,  $\mathcal{I} \models A$ .

Next we define the notion of the monotonic extension  $\mathcal{O}'$  of the ontology  $\mathcal{O}$ .

**Definition 2.** We say that  $\mathcal{O}' = \langle S', A' \rangle$  is an extension of  $\mathcal{O} = \langle S, A \rangle$  if  $A \subset A'$  and  $S \subset S'$ .  $\mathcal{O}'$  is monotonic if the original ontology  $\mathcal{O}$  is entailed by  $\mathcal{O}'$ ,  $\mathcal{O}' \models \mathcal{O}$ . A model  $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$  of the ontology  $\mathcal{O}$  is an extension of the model  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}) \models \mathcal{O}$  if  $\Delta^{\mathcal{I}'} \subseteq \Delta^{\mathcal{I}}$  and  $\cdot^{\mathcal{I}'} \subseteq \cdot^{\mathcal{I}}$ .

For a full description of the syntax and semantics of different description logic formalisms see [20]. Note that our discussion is also applicable for OWL 2 ontologies [22, 23] since OWL 2 is backed by the *SRIOIQ(D)* description logic.

We consider description logics which allow only monotonic extensions. In section 5.2 where we prove decidability of the *semantic tuple count* we further restrict to a subfamily of description logics that enable expressing whether two entity objects are different, the same or it is not known, which we will refer to as *equality/inequality relation*. We require the OWA assumption over the *equality/inequality relation* because it provides means to model incomplete knowledge, and it is the fundamental source of uncertainty of the *semantic tuple count* function's value.

## 4.2 Conjunctive Queries

The discussion in this paper about *distinct count* function and its proposed interpretations is set in the context of conjunctive queries.

**Definition 3.** We will denote conjunctive queries using the following rule like notation

$$Q(\bar{x}) \leftarrow \phi(\bar{x}, \bar{z}). \quad (1)$$

The head of the query  $Q(\bar{x})$  denotes the name of the query and the result variables  $\mathcal{R}_{var}(Q) = \bar{x}$ . The body of the query  $\phi(\bar{x}, \bar{z})$  is a comma separated list of query atoms interpreted as a conjunctive query, as defined in the SPARQL-DL<sup>2</sup> query language. The query atom list  $\phi$  may contain non result variables  $\bar{z}$ . Note that the result variables must be distinguished. By  $\mathcal{V}_{var}(Q)$  we denote the list of all variables in the query. A binding  $\mu : \mathcal{V}_{var}(Q) \rightarrow S$  is a mapping of the variables of the query to elements in the ontology's signature and  $Q|_{\mu}$  is the substitution of the variables in  $Q$  by the binding  $\mu$ . The binding substitution of a tuple of variables  $\bar{v} = (v_1, \dots, v_k)$  is denoted by  $\mu(\bar{v}) = (\mu(v_1), \dots, \mu(v_k))$ . We call  $Q$  a ground query if there are no variables in the query. A solution to the query  $Q$  w.r.t. the ontology  $\mathcal{O}$  is a binding  $\mu$  for which the substitution  $Q|_{\mu}$  is a ground query, the body of which is entailed by the ontology (denoted by  $\mathcal{O} \models Q|_{\mu}$ ). The set of all possible solutions of  $Q$  w.r.t.  $\mathcal{O}$  or the model  $\mathcal{I}$  of  $\mathcal{O}$  is denoted by  $Sat_Q^{\mathcal{O}} = \{\mu | \mathcal{O} \models Q|_{\mu}\}$  or  $Sat_Q^{\mathcal{I}} = \{\mu | \mathcal{I} \models Q|_{\mu}\}$  respectively. The result set of query  $Q$  w.r.t.  $\mathcal{O}$  denoted by  $Q_{\mathcal{O}}$ , is a set of bindings of the result variables  $\mathcal{R}_{var}(Q)$ , formally  $Q_{\mathcal{O}} = \{\bar{a} | \bar{a} = \mu(\mathcal{R}_{var}(Q)) \wedge \mu \in Sat_Q^{\mathcal{O}}\}$ .

## 4.3 Aggregate Queries

Here we define the types of aggregate queries along with their simple syntax used for the purpose of representing aggregate queries in this paper. We also define some additional terminology and symbols used later in the paper.

**Definition 4.** We will denote distinct count retrieval and distinct count filtering queries respectively as follows:

$$Q_a(\bar{x}, \text{count}_{\text{dist}}^S(\bar{y})) \leftarrow \phi(\bar{x}, \bar{y}, \bar{z}) \quad (2)$$

<sup>2</sup> See [1] for list of atoms and their interpretations.

$$Q_{ac}(\bar{x}) \leftarrow (\cdot_{op} \text{ncount}_{\text{dist}}^{\mathcal{S}}(\bar{y})), \phi(\bar{x}, \bar{y}, \bar{z}) \quad (3)$$

In the queries of the type  $Q_a$ , the head  $Q_a(\bar{x}, \text{count}_{\text{dist}}^{\mathcal{S}}(\bar{y}))$  specifies the disjoint sets of grouping  $\mathcal{G}_{\text{var}}(Q_a) = \bar{x}$  and aggregation  $\mathcal{A}_{\text{var}}(Q_a) = \bar{y}$  variables. The distinct count aggregation function which returns the number of distinct tuples according to the semantics mode specified by the superscript  $\mathcal{S}$  w.r.t. the ontology  $\mathcal{O}$  is denoted by  $\text{count}_{\text{dist}}^{\mathcal{S}}$ . The body of  $Q_a$  contains a query atom list.

We also consider distinct count filtering by comparison queries of the form  $Q_{ac}$  show in (3). The head of queries of these type contain only the grouping variables. The body of the query contains cardinality restriction atom<sup>3</sup> where  $\cdot_{op} \in \{>, <, \leq, \geq, =, \neq\}$ . By  $Q^*$  we denote a non aggregate variant of  $Q$ , obtained from  $Q$  by removing its aggregate function from the head or the comparison predicate in the body. The result variables of  $Q^*$  are the union of group and aggregate variables of  $Q$ ,  $\mathcal{R}_{\text{var}}(Q^*) = \mathcal{G}_{\text{var}}(Q) \cup \mathcal{A}_{\text{var}}(Q)$ .

Before we define the general semantics of the result of aggregate queries we clarify and define the auxiliary terms and notations in the following three definitions.

**Definition 5.** Let  $\mathbb{N}^{0, \infty}$  be the extension of the set of natural numbers with zero and infinity  $\mathbb{N}^{0, \infty} = \mathbb{N} \cup \{0, \infty\}$ . Let  $L$  be a subset of  $\mathbb{N}^{0, \infty}$  and let  $\text{Int}(L)$  denote the smallest interval containing  $L$ ,  $\text{Int}(L) = (\inf(L), \sup(L))$ . We extend the intuitive comparison between elements in  $\mathbb{N}^{0, \infty}$  with comparison between sets  $L \subseteq \mathbb{N}^{0, \infty}$  and elements  $n \in \mathbb{N}^{0, \infty}$ .  $L \leq n$  ( $L < n$ ) is true if and only if  $\sup(L) \leq n \vee \sup(L) = n = \infty$  ( $\sup(L) < n$ ).  $L \geq n$  ( $L > n$ ) is true if and only if  $\inf(L) \geq n \vee \inf(L) = n = \infty$  ( $\inf(L) > n$ ). Note that the symbol  $\infty$  is an element of  $\mathbb{N}^{0, \infty}$ .

Next we define the interpretation a tuple and a set of tuples.

**Definition 6.** Let  $\mathcal{O} = \langle S, A \rangle$  be an ontology,  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  a model of  $\mathcal{O}$  and  $T$  be a set of tuples composed of elements in  $S$ . The interpretation of the of the tuple  $\bar{t} = (t_1, t_2, \dots, t_k)$ ,  $\bar{t} \in T$  is  $\bar{t}^{\mathcal{I}} = (t_1^{\mathcal{I}}, t_2^{\mathcal{I}}, \dots, t_k^{\mathcal{I}})$ . The interpretation of  $T$  is  $T^{\mathcal{I}} = \{\bar{t}^{\mathcal{I}} \mid \bar{t} \in T\}$ .

Next we define aggregate groups or simply groups as the set of tuples with common grouping variable binding.

**Definition 7.** Let  $Q$  be an aggregate query of type  $Q_a$  or  $Q_{ac}$ ,  $\mathcal{O}$  be an ontology,  $\mathcal{I}$  a model of  $\mathcal{O}$  and let  $\bar{k} = \mu(\mathcal{G}_{\text{var}}(Q^*))$ , where  $\mu$  is an arbitrary binding. Then the aggregate group, denoted by  $\Gamma(\mathcal{O}, Q, \bar{k})$ , with key  $\bar{k}$  is equal to the set of tuples  $\{\mu(\mathcal{A}_{\text{var}}(Q)) \mid \mu(\mathcal{G}_{\text{var}}(Q)) = \bar{k} \wedge \mu \in \text{Sat}_{\mathcal{O}^*}^{\mathcal{I}}\}$ . The aggregate group with key  $\bar{k}$  in a tuple set  $T$  is  $\Gamma(T, \bar{k}) = \{\bar{a} \mid \bar{t} \in T, (\bar{k}, \bar{a}) = \bar{t}\}$ .

<sup>3</sup> Note that we reuse the well known notation for property cardinality restrictions, see [20].

Note that the definition of the aggregate group  $\Gamma(\mathcal{O}, Q, \bar{k})$  w.r.t. to the ontology  $\mathcal{O}$  can be used also to obtain an aggregate group w.r.t. to the model  $\mathcal{I}$  of  $\mathcal{O}$ , i.e.  $\Gamma(\mathcal{I}, Q, \bar{k})$ .

Next we define results of aggregate queries in terms of the counting function  $f_{[\mathcal{O}, Q]}^S$  and the set  $K^S(\mathcal{O}, Q)$ . Informally the counting function  $f_{[\mathcal{O}, Q]}^S$  is an uncertainty aware generalization of the *distinct count* function, it takes as an input the key of the group to be counted and returns a set of possible values w.r.t.  $\mathcal{O}$  and  $Q$ . The set  $K^S(\mathcal{O}, Q)$  is the set of keys of the groups to be counted. The counting function and the key of sets will be defined in each of the concrete distinct count interpretations. The superscript is used to distinguish among the different interpretations.

**Definition 8.** Let  $\mathbb{D}$  be the set of all tuples of arbitrary length and  $\mathcal{P}(\mathbb{N}^{0,\infty})$  be the power set of  $\mathbb{N}^{0,\infty}$ . Let  $\mathcal{S}$  be an aggregate semantic,  $f_{[\mathcal{O}, Q]}^S : \mathbb{D} \rightarrow \mathcal{P}(\mathbb{N}^{0,\infty})$  be the counting function and  $K^S(\mathcal{O}, Q) \subset \mathbb{D}$  be the key set in  $\mathcal{S}$ . The result of an aggregate query  $Q$  of type (2) w.r.t. the ontology  $\mathcal{O}$  is  $Q_{\mathcal{O}} = \{\{\bar{k} \in K^S(\mathcal{O}, Q), \inf(L)\} | L := f_{[\mathcal{O}, Q]}^S(\bar{k}) \wedge \inf(L) = \sup(L)\}$  and the result set of an aggregate query  $Q$  with comparison filtering, i.e. queries of the form  $Q_{ac}$ , is  $Q_{\mathcal{O}} = \{\{\bar{k} \in K^S(\mathcal{O}, Q) | L := f_{[\mathcal{O}, Q]}^S(\bar{k}) \wedge L \cdot_{op} n\}$ . The interpretation of the last condition  $L \cdot_{op} n$  is defined in Definition 5.

## 5 Distinct count function

In this section we formally define the semantics of each of the various interpretations of the counting function  $f_{[\mathcal{O}, Q]}^S$  and the set of keys  $K^S(\mathcal{O}, Q)$ . We present this definitions in order to be able to show formally the relations between the interpretations and in the case of  $\mathcal{SC}$  and  $\mathcal{STC}$  to be able evaluate comparison filtering queries in OWA. We show and discuss the results of the example queries  $Q_1$  and  $Q_2$  from section 3 in each of the proposed semantics. We also study the relationship between individual interpretations of the *distinct count* function. We prove decidability of the *semantic tuple count STC* interpretation. Finally we show how to enable distinct count queries in the context of the description logic  $\mathcal{SROIQ}$ .

In (4) we show the queries  $Q_1$  and  $Q_2$  from example 1 in the notation defined in Definition 4.

$$\begin{aligned} Q_1(?t, \text{count}_{\text{dist}}?c) &\leftarrow \text{PropertyValue}(\text{teaches}, ?t, ?c) \\ Q_2(?t) &\leftarrow (>1\text{count}_{\text{dist}}(?c)), \text{PropertyValue}(\text{teaches}, ?t, ?c) \end{aligned} \quad (4)$$

The tables in this section showing the results of the queries have the following notation. Only cells highlighted with grey background color <sup>4</sup> are part of the result set of the query. The other cells have white background. Note that through out this section  $\mathcal{O} = \langle S, A \rangle$  is an arbitrary ontology with a signature  $S$  and an axiom set  $A$ ,  $Q$  is an arbitrary aggregate query,  $\mathcal{O}_1$  refers to the ontology in example 1 and  $Q_1$  and  $Q_2$  refer to the queries shown in (4).

<sup>4</sup> The use of colors in this paper is intended to be readable in gray scale copies.

## 5.1 Known interpretations of the *Distinct Count*

In this section we discuss three known interpretations of the *distinct count* function in aggregate queries.

**Basic Count Interpretation** The *basic count* interpretation ( $\mathcal{BC}$ ) is used originally in the SQL query language, but it is also used in semantic query languages, e.g. SPARQL and SQWRL. This interpretation is not adequate for ontological knowledge as it does not infer the *distinct count* function’s value from the ontology. The *basic count* interpretation can be implemented in  $n \log n$  time, where  $n$  is the size of the tuple set to be counted.

**Definition 9.** *The set of keys  $K^{\mathcal{BC}}(\mathcal{O}, Q)$  is the set of all syntactically distinct result bindings of the grouping variables  $\mathcal{G}_{var}(Q)$  of the query  $Q$  over the ontology  $\mathcal{O}$ , i.e.  $K^{\mathcal{BC}}(\mathcal{O}, Q) = \{\bar{k} \mid \bar{k} = \mu(\mathcal{G}_{var}(Q)) \wedge \mu \in \text{Sat}_{Q^*}^{\mathcal{O}}\}$ . For  $\bar{k} \in K^{\mathcal{BC}}(\mathcal{O}, Q)$  the counting function is defined as  $f_{[\mathcal{O}, Q]}^{\mathcal{BC}} = \{|\Gamma(\mathcal{O}, Q, \bar{k})|\}$ .*

The result of the distinct count query  $Q_1$  is shown in table 1(a).

**Table 1.** The result set of the  $\mathcal{BC}$  interpretation of the aggregate queries  $Q_1$  and  $Q_2$ .

(a) result of $Q_1$	(b) result of $Q_2$												
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="border: 1px solid black; padding: 2px 5px;">?t</th> <th style="border: 1px solid black; padding: 2px 5px;">count<math>_{\text{dist}}^{\mathcal{BC}}(?c)</math></th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 2px 5px;">Dave</td> <td style="border: 1px solid black; padding: 2px 5px;">3</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px 5px;">Sara</td> <td style="border: 1px solid black; padding: 2px 5px;">1</td> </tr> </tbody> </table>	?t	count $_{\text{dist}}^{\mathcal{BC}}(?c)$	Dave	3	Sara	1	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="border: 1px solid black; padding: 2px 5px;">?t</th> <th style="border: 1px solid black; padding: 2px 5px;">(&gt; 1count<math>_{\text{dist}}^{\mathcal{BC}}(?c)</math>)</th> </tr> </thead> <tbody> <tr> <td style="border: 1px solid black; padding: 2px 5px;">Dave</td> <td style="border: 1px solid black; padding: 2px 5px;">true</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px 5px;">Sara</td> <td style="border: 1px solid black; padding: 2px 5px;">false</td> </tr> </tbody> </table>	?t	(> 1count $_{\text{dist}}^{\mathcal{BC}}(?c)$ )	Dave	true	Sara	false
?t	count $_{\text{dist}}^{\mathcal{BC}}(?c)$												
Dave	3												
Sara	1												
?t	(> 1count $_{\text{dist}}^{\mathcal{BC}}(?c)$ )												
Dave	true												
Sara	false												

There are two group keys **Dave** and **Sara**. The number of courses **Dave** teaches is three because there are three courses that **Dave** teaches asserted in the ontology, whose names are syntactically different. For **Sara**, who teaches only one course according to the ontology, the result of the count function is one. We can see that this interpretation ignores the implicit knowledge derived from the fact that **Sara** is a **BusyTeacher** that teaches at least three courses.

**Semantic Count Interpretation** Informally the *semantic count* ( $\mathcal{SC}$ ) interpretation counts the number of possible tuples entailed by the queried ontology. This interpretation is similar to the *exact semantics* presented in [18]. However our definition of  $\mathcal{SC}$  presented here does not return a single value of the distinct count function but a set of possible values. While this definition has no effect on the results of queries of type  $Q_a$  it ensures monotonic results in the queries of type  $Q_{ac}$ . The decidability of the  $\mathcal{SC}$  interpretation is an open problem. Nevertheless we believe that the minimum of the  $\mathcal{SC}$  interpretation is more likely to be decidable.



**Definition 10.** The semantic count interpretation is denoted by  $\mathcal{SC}$ . Let  $l = |\mathcal{G}_{var}(Q)|$  be the number of grouping variables in  $Q$  and  $\mathbb{S} = S^l$  is the set of all tuples of length  $l$  composed of elements from the signature  $S$ . The  $\mathcal{SC}$  counting function is defined as follows  $f_{[\mathcal{O}, Q]}^{\mathcal{SC}}(\bar{k}) = \{ |T^{\mathcal{I}}| \mid T := \Gamma(\mathcal{I}, Q, \bar{k}), \forall \mathcal{I} \models \mathcal{O} \}$ . The  $\mathcal{SC}$  key set is defined as follows  $K^{\mathcal{BC}}(\mathcal{O}, Q) = \{ \bar{k} \in \mathbb{S} \mid \sup(f_{[\mathcal{O}, Q]}^{\mathcal{SC}}(\bar{k})) > 0 \}$ .

Next we discuss the results of the example queries  $Q_1$  and  $Q_2$ , with formal representation shown in (4), over the ontology  $\mathcal{O}_1$  from example 1. The last columns in the tables 2(a) and 2(b) show the boundaries of the intervals found for each of the group keys from the first column.

**Table 2.** The result of the aggregate queries  $Q_1$   $Q_2$  in 4 with  $\mathcal{SC}$  interpretation.

(a) result of $Q_1$			(b) result of $Q_2$		
$?t$	$\text{count}_{\text{dist}}^{\mathcal{SC}}(?c)$	$\text{Int}(f_{[\mathcal{O}_1, Q_1]}^{\mathcal{SC}})$	$?t$	$(> 1\text{count}_{\text{dist}}^{\mathcal{SC}}(?c))$	$\text{Int}(f_{[\mathcal{O}_1, Q_2]}^{\mathcal{SC}})$
Dave	-	$\langle 2, \infty \rangle$	Dave	true	$\langle 2, \infty \rangle$
Sara	-	$\langle 3, \infty \rangle$	Sara	true	$\langle 3, \infty \rangle$
Steve	-	$\langle 0, 3 \rangle$	Steve	false	$\langle 0, 3 \rangle$
John	3	$\langle 3, 3 \rangle$	John	true	$\langle 3, 3 \rangle$
math	-	$\langle 0, \infty \rangle$	math	false	$\langle 0, \infty \rangle$

In table 2(a) we show the results of query  $Q_1$ . Next we discuss the intuition of the calculation of the values of the  $\mathcal{SC}$  counting function for each of groups in table 2(a). In order to obtain the final result of the query we need to apply the comparison semantics from definition 8. The last row in both tables show an unexpected group which the algorithm should process. In fact we have two more groups that we omitted from the table which are with keys **history** and **physics**. This is an effect of the OWA assumption. This anomaly is caused by the fact that the ontology  $\mathcal{O}_1$  does not explicitly state that **math**, **history** and **physics** are not teachers and that only teachers can teach and thus allowing the existence of models in which subjects teach something. Moreover the calculated interval is zero to infinity because there are no axioms which constraint it. The  $\mathcal{SC}$  can be used to locate such unwanted behavior. The minimum in the first row with group key **Dave** is obtained from the two semantically distinct courses **math** and **history** that **Dave** teaches. The maximum of the first row is infinity because the ontology does not constrain the number of courses **Dave** can teach. The minimum of the second row with group key **Sara** is determined from the constraint that a **BusyTeacher** teaches at least three courses and the fact that **Sara** is a **BusyTeacher** and who teaches **history**. In this case there are two restrictions, 'at least three' and 'at least one' which constraint the minimum of courses **Sara** teaches. In such cases we should select the most specific one. Therefore the minimum of the second row's interval is three since the first restriction is more specific. The evaluation of the maximum in the second row's interval is analogous to the one in the first row. The third row's minimum is

zero because **Steve** is not constrained to teach a minimum number of courses as **Dave** and **Sara** were in the first and second rows. The maximum in the third row is derived from the fact that **Steve** is a professor and the constraint of the **Professor** class which limits its instances to teach at most three courses. Now we apply interval semantics to the first three rows that we discussed. The three rows are not included in the result of query  $Q_1$  with  $\mathcal{SC}$  semantics because according to the definition of the results of queries of type  $Q_a$  in definition 8 which states that set returned by the counting function must be singleton. The answer contains only the last row because the return value of the counting function is a singleton containing only the number three. This is true because the **John** is both a **Professor** and a **BusyTeacher** the constraints of which were already discussed.

The results of query  $Q_2$  as well as all the relevant group keys of  $Q_2$  are shown in table 2(b). We have discussed the interval boundaries of the intervals for groups of  $Q_1$  and since the  $Q_2$  has identical groups, note that  $K^{\mathcal{SC}}(\mathcal{O}_1, Q_1) = K^{\mathcal{SC}}(\mathcal{O}_1, Q_2)$ , we skip this explanation for query  $Q_2$ . The result of  $Q_2$  contains all rows except for the third one with group key **Steve** that does not satisfies the comparison filter which limits the retrieval of groups with at least two distinct tuples in all models of the ontology  $\mathcal{O}_1$ .

We point out the importance of interval semantics in the settings of incomplete knowledge. Note that in query  $Q_2$  we obtained results that were not present in the result of  $Q_1$ . This proves that the results of query  $Q_2$  obtained by filtering the results of query  $Q_1$  won't contain the full result entailed by the ontology.

**Epistemic Count Interpretation** The *epistemic count* ( $\mathcal{EC}$ ) interpretation is introduced in the work [18]. Informally this interpretation of the *distinct count* function returns a single value which represents the known number of distinct tuples in an aggregate group. For the number  $k$  of known tuples holds that in any model  $\mathcal{I}$  of the ontology  $\mathcal{O}$  there is at least  $k$  distinct tuples for the counted group. Because  $\text{count}_{\text{dist}}^{\mathcal{EC}}$  interpretation is equivalent to the infimum of the  $\mathcal{SC}$  counting function here we define the  $\text{count}_{\text{dist}}^{\mathcal{EC}}$  interpretation in terms of  $\mathcal{SC}$  counting function and key set. The decidability of the  $\mathcal{EC}$  interpretation is an open problem as it is equivalent to the decidability of the minimum of the  $\mathcal{SC}$  interpretation problem.

**Definition 11.** *The  $\mathcal{EC}$  key set is defined as  $K^{\mathcal{EC}}(\mathcal{O}, Q) = K^{\mathcal{SC}}(\mathcal{O}, Q)$ . The  $\mathcal{EC}$  counting function is defined as  $f_{[\mathcal{O}, Q]}^{\mathcal{EC}}(\bar{k}) = \{\inf(f_{[\mathcal{O}, Q]}^{\mathcal{SC}}(\bar{k}))\}$ .*

In tables 3(a) and 3(b) we can see the results of the aggregate queries  $Q_1$  and  $Q_2$  respectively with *distinct count* function interpreted with the  $\mathcal{EC}$  semantics. The results are identical with those of the minimum of the interval in tables 2(a) and 2(b) and are not discussed further.

## 5.2 Semantic Tuple Count Interpretation

The *semantic tuple count* ( $\mathcal{STC}$ ) interpretation is defined using interval semantics. Informally it counts the same tuples as the  $\mathcal{BC}$  interpretation but it uses

**Table 3.** The result of the aggregate queries  $Q_1$  and  $Q_2$  in (4) with the  $\mathcal{EC}$  interpretation.

(a) result of $Q_1$		(b) result of $Q_2$	
?t	$\text{count}_{\text{dist}}^{\mathcal{EC}}(?c)$	?t	$(> 1\text{count}_{\text{dist}}^{\mathcal{EC}}(?c))$
Dave	2	Dave	true
Sara	3	Sara	true
John	3	John	true
math	0	math	false

knowledge in the ontology to derive the *equivalence/inequivalence* relation between the counted tuples which is needed to remove semantically duplicate tuples and also to deal with uncertainty of the count's value.

**Definition 12.** The  $STC$  key set  $K^{STC}$  is defined as the key set of the  $BC$  interpretation, i.e.  $K^{STC}(\mathcal{O}, Q) = K^{BC}(\mathcal{O}, Q)$ . We define the  $STC$  counting function as follows  $f_{[\mathcal{O}, Q]}^{STC}(\bar{k}) = \{|\Gamma(\mathcal{O}, Q, \bar{k})^I| \mid \forall \mathcal{I} \models \mathcal{O}\}$ .

**Table 4.** The result of the aggregate queries  $Q_1$  and  $Q_2$  in (4) with the  $STC$  interpretation.

(a) result of query $Q_1$			(b) result of query $Q_2$	
?t	$\text{count}_{\text{dist}}^{STC} (?c)$	$\text{Int}(f_{[\mathcal{O}, Q_1]}^{STC})$	?t	$(> 1\text{count}_{\text{dist}}^{STC} (?c))$
Dave	-	$\langle 2, 3 \rangle$	Dave	true
Sara	1	$\langle 1, 1 \rangle$	Sara	false

Next we discuss the results of queries  $Q_1$  and  $Q_2$  from example 1 with the  $STC$  interpretation. Here we also show all the group keys that should be considered during evaluation of the  $STC$  interpretation and since the  $STC$  interpretation is interval based we also show the evaluated intervals in the last columns of the tables 4(a) and 4(b). In table 4(a) we have the results of  $Q_1$ . Now we discuss the values of the interval boundaries in rows one and two. The minimum in the first row is based on the two distinct courses **math** and **history**. The maximum is three because there are three entailed tuples in the group of the first row and because there is no other axioms that constraints the maximum to be smaller. The minimum in the second row is one because we have only one tuple for the **Sara** group. This is also the only possible value for the maximum of row two because there are no other entailed tuples in that group. Apparently from the interval semantics only the second row is returned.

The results for the query  $Q_2$  are shown in table 4(b). We already discussed the interval boundaries in the discussion of query  $Q_1$ . Applying the interval semantics we filter the second row because **Sara** teaches only one distinct course and therefore it is not contained in the result. The first is contained in the result

because the minimum boundary, which is 2, is bigger than the less than operand in the filtering atom, which is one.

**Decidability of the *STC* Interpretation** In this section we prove decidability of the *STC* interpretation. We present only the statements of the most relevant lemmas and the proof of the theorem at the end of the section. The omitted proofs and lemmas can be found in the technical report [24]. First we define the family of formalisms for which we prove that the *STC* interpretation is decidable.

**Definition 13.** *We assume that the supported formalisms  $\mathcal{F}$  (i) are capable of expressing the equality/inequality relation among elements of the ontology (ii) consistency check of ontologies and query answering is decidable in  $\mathcal{F}$  and (iii) that  $\mathcal{F}$  is monotonic.*

Implementing the evaluation of the *STC* counting function based on definition 5.2 is not feasible because the ontology  $\mathcal{O}$  might have an infinite number of models. We show that there is a finite number of models sufficient for the calculation of the boundaries of the value of the *STC* counting function.

The next proposition states that the interval of the *distinct count* function with *SC* and *STC* interpretation w.r.t. the ontology  $\mathcal{O}$  will be more specific if we extend the queried ontology. The interval calculated w.r.t. the original ontology will include the one calculated from the extended ontology.

Note that in this section we will use identical equality and inequality axioms, denoted by  $=_a$  and  $\neq_a$  respectively, for all type of entities, i.e. individuals, classes and properties. In the following section we show an equivalent representation of this axioms in the concrete logic *SROIQ*.

**Definition 14.** *Let  $\mathcal{O} = \langle S, A \rangle$  be an ontology,  $T$  be a tuple set composed of elements in the set  $D \subseteq S$ . Let  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  be an interpretation such that  $\cdot^{\mathcal{I}}$  is defined on  $D$  then the complete set of equality/inequality axioms satisfied by  $\mathcal{I}$  is denoted as  $A_{\#}(\mathcal{I}, D) = \{a =_a b \mid \forall a, b \in D, a^{\mathcal{I}} = b^{\mathcal{I}}\} \cup \{a \neq_a b \mid \forall a, b \in D, a^{\mathcal{I}} \neq b^{\mathcal{I}}\}$ . We denote the set all complete sets of equality/inequality axioms between elements in  $D$  w.r.t. the ontology  $\mathcal{O}$  as  $\mathbb{A}_{\#}(\mathcal{O}, D) = \{A_{\#}(\mathcal{I}, D) \mid \forall \mathcal{I} \models \mathcal{O}\}$ . The cannonic model of  $A_{\#} = A_{\#}(\mathcal{I}, D)$ , denoted by  $\mathcal{I}_{A_{\#}} = (\Delta^{\mathcal{I}_{A_{\#}}}, \cdot^{\mathcal{I}_{A_{\#}}})$  is a model with an interpretation function  $\cdot^{\mathcal{I}_{A_{\#}}}$  the domain of which is  $D$ .*

Note that adding equality/inequality axioms to the complete set  $A_{\#}(\mathcal{I}, D)$  wont change the original set or if it does the resulting set is unsatisfiable. Note also that the cannonic interpretation function  $\cdot^{\mathcal{I}_{A_{\#}}}$  can be constructed in polynomial time.

**Lemma 1.** *Let  $\mathcal{O} = \langle S, A \rangle$  be an ontology,  $D \subseteq S$ , then  $\mathbb{A}_{\#}(\mathcal{O}, D)$  is finite.  $\square$*

**Lemma 2.** *Let  $T$  be a tuple set composed of elements from set  $D$  and two interpretations  $\cdot^{\mathcal{I}_1}$  and  $\cdot^{\mathcal{I}_2}$  which agree on the equivalence and inequivalence between elements in  $D$ . Then the number of elements in the sets  $T^{\mathcal{I}_1}$  and  $T^{\mathcal{I}_2}$  is the same,  $|T^{\mathcal{I}_1}| = |T^{\mathcal{I}_2}|$ .  $\square$*

The next lemma states that for extensions of the ontology  $\mathcal{O}$  with an axiom set from  $\mathbb{A}_{\#}(\mathcal{O}, D)$  the *STC* counting function returns a singleton set and that the only value in the set can be calculated in polynomial time.

**Lemma 3.** *Let  $\mathcal{O} = \langle S, A \rangle$  be an ontology,  $Q$  be an aggregate query,  $\bar{k} \in K^{STC}$ ,  $T = \Gamma(Q_{\mathcal{O}}^*, \bar{k})$  and  $D$  be the set of elements composing tuples in  $T$ ,  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  be a model of  $\mathcal{O}$ ,  $A_{\#} = A_{\#}(\mathcal{I}, D)$ ,  $\mathcal{O}' = \langle S, A \cup A_{\#} \rangle$  and  $\cdot^{\mathcal{I}_{A_{\#}}}$  be the cannonic interpretation function of  $A_{\#}$ . Then  $f_{[\mathcal{O}', Q]}^{STC}(\bar{k}) = \{c\}$ , where  $c = |T^{\mathcal{I}}| = |T^{\mathcal{I}_{A_{\#}}}|$ .  $\square$*

Instead of looking for the boundaries of the *STC* interval in the set of all models as the definition 12 suggests, we can search for the boundaries in the finite set of representations of the equality interpretation. We first describe an algorithm which terminates in a final number of steps and then we prove its correctness.

*Algorithm 1* : STC - Semantic Tuple Count procedure

```

PROCEDURE STC
  INPUT : O // the queried ontology,
         T // set of tuples to be counted
  OUTPUT : <a,b> // the calculated interval
  a := inf; b := 0;
  D := {elements used in tuples in T};
  FORALL A# IN A#(O,D) DO
    A' := union(A, A#);
    IF A' is consistent
      construct cannonic interpretation I#(A#);
      c := |T interpreted by I#(A#)|;
      IF a > c THEN a := c;
      IF b < c THEN b := c;
    END-IF
  END-FORALL
  RETURN <a,b>;
END

```

We will prove decidability by proving correctness and termination of algorithm 1.

**Theorem 1.** *The algorithm 1 terminates and evaluates correctly the interval  $\text{Int}(f_{[\mathcal{O}, Q]}^{STC}(\bar{k}))$  for an ontology  $\mathcal{O}$  and aggregate query  $Q$  and group key  $\bar{k}$ .*

*Proof.* Algorithm 1 terminates since there is a finite number of extensions to check. From lemma 3 we have that the *STC* counting function is a subset of the calculated interval. Also from lemma 3 we have that the interval is the smallest because the boundaries correspond to some model of  $\mathcal{O}$ .  $\square$

**Corollary 1.** *Calculation of the *STC* interval is decidable in the family of formalisms defined in Definition 13.  $\square$*

**STC Interpretation approximation** The proposed algorithm 1 is searching through all the possible extensions for the set of elements  $D$ . As we show in corollary 1 there are  $n = 2^{|D|^2}$  possible extensions for each of which we need to make a consistency check. In this section we propose an approximation of the *STC* interpretation which can be used for example as optimization of an evaluation algorithm. The approach presented here utilizes the knowledge inferred from the ontology. The prove of the correctness of the approximation is presented in the technical report [24].

**Definition 15.** Let  $G = (V, E)$  be an undirected graph with nodes  $V$  and edges  $E$ . A connected component  $K$  in  $G$  is a subgraph in  $G$  maximal with the property, for each pair of nodes in  $K$  there is a path in  $K$ . A (maximal) clique  $C$  is a graph maximal with the property  $C$  is a subgraph of  $G$  and is complete. The biggest clique  $C$  in  $G$  is called maximum.

In order to define the approximation formally we need first to define the auxiliary term *difference graph*, and the notion of *maximum clique*.

**Definition 16.** Let  $\mathcal{O} = \langle S, A \rangle$  be an ontology, let  $T$  be a set of tuples of length  $l$  and composed of elements from the set  $D \subseteq S$ . Let  $\bar{s}, \bar{t} \in T$ , equality  $\bar{s} =_a \bar{t}$  and inequality of tuples w.r.t. the ontology  $\mathcal{O}$  is defined respectively as  $\mathcal{O} \models \{s_1 =_a t_1, \dots, s_l =_a t_l\}$  and  $\mathcal{O} \models s_i \neq_a t_i$  for some  $1 \leq i \leq l$ . Let  $G_=(\mathcal{O}, T) = (D, E_=(\mathcal{O}, T))$  be the graph with edges  $E_=(\mathcal{O}, T) = \{(a, b) | \forall (a, b) \in T^2, \mathcal{O} \models a =_a b\}$ . The difference graph is denoted by  $\Delta G(\mathcal{O}, T) = (V, E)$  w.r.t.  $\mathcal{O}$  and  $T$ . The set nodes is the set of connected components of  $G_=(\mathcal{O}, T)$ , there is an edge between the nodes  $u$  and  $v$  in  $V$ ,  $\{u, v\} \in E$  if and only if the ontology entails  $\neq_a$  axiom between some pair of the set  $u \times v$ .

**Theorem 2.** Let  $\mathcal{O}$  be an ontology,  $Q$  an aggregate query,  $\bar{k} \in K^{STC}(\mathcal{O}, Q)$  and let  $T = \Gamma(\mathcal{O}, Q, \bar{k})$  be the group to be counted. Let the  $\Delta G(\mathcal{O}, T) = (V, E)$  be the difference graph w.r.t.  $\mathcal{O}$  and  $T$  and  $C_{max} = (V_C, E_C)$  be the maximum clique graph of  $\Delta G(\mathcal{O}, T)$ .  $\text{Int}(f_{[\mathcal{O}, Q]}^{STC}(\mathcal{V}_{var}(k))) \subseteq \langle |V_C|, |V| \rangle$ .  $\square$

**STC interpretation in SROIQ** In this section we discuss the  $\text{count}_{\text{dist}}^{STC}$  function in a concrete description logic *SROIQ*. The *STC* interpretation is decidable in this description logic since *SROIQ* satisfies all the requirements shown in Definition 13. We consider any query language which supports conjunctive queries with mixed Abox, Tbox, Rbox terms, e.g. SPARQL-DL<sup>NOT</sup> [25]. In order to apply the *STC* interpretation in this scenario we need to show representations of equality and inequality between elements of the signature of the ontology which will be used to generate the complete set of axioms in  $\mathbb{A}_{\#}$ .

5.2 shows the representations for each of the three term types. Individuals  $i_k, k \in 1, 2, 3, \dots$ , are unique and not contained in the ontology. Note that classes and properties have two possible ways of representing the inequality relation. If one of the representations fails we must also test the other when checking whether two terms of type class or property need to be compared for inequality.

**Table 5.** Semantics of comparison of different ontology elements

type	$=_a$	$\neq_a$
individuals	$i_1 \doteq i_2$	$i_1 \neq i_2$
classes	$C_1 \equiv C_2$	$\{C_1(i_1), \neg C_2(i_1)\}$ or $\{C_2(i_2), \neg C_1(i_2)\}$
properties	$p_1 \equiv p_2$	$\{p_1(i_1, i_2), \neg \exists p_2.\{i_2\}(i_1)\}$ or $\{p_2(i_3, i_4), \neg \exists p_1.\{i_4\}(i_3)\}$

## 6 Conclusion

We investigated the *distinct count* function in the context of different semantics, i.e.  $\mathcal{BC}$ ,  $\mathcal{SC}$  and  $\mathcal{EC}$ . We introduced a new interpretation  $\mathcal{STC}$  and compared its behavior with the other interpretations. We found the following relationships between  $\mathcal{SC}$  and  $\mathcal{EC}$ ,  $\inf(f_{[\mathcal{O}, Q]}^{\mathcal{SC}}) = \mathcal{EC}$  and also between  $\mathcal{SC}$  and  $\mathcal{STC}$ ,  $\inf(f_{[\mathcal{O}, Q]}^{\mathcal{STC}}(\bar{k})) \leq \inf(f_{[\mathcal{O}, Q]}^{\mathcal{SC}}(\bar{k}))$  and  $\sup(f_{[\mathcal{O}, Q]}^{\mathcal{STC}}(\bar{k})) \leq \sup(f_{[\mathcal{O}, Q]}^{\mathcal{SC}}(\bar{k}))$ . We proved decidability of the  $\mathcal{STC}$  interpretation in the context of the selected family of formalisms, provided an approximation using basic graph problems and showed how to apply the  $\mathcal{STC}$  in the  $\mathcal{SROIQ}$  description logic.

In future work we would like to focus on the implementation of the evaluation of the  $\mathcal{STC}$  interpretation into the SPARQL-DL<sup>NOT</sup> [25] query language. In order to provide practical implementation research in optimizing the evaluation is needed. In the worst case algorithm 1 will issue  $2^{n^2}$  consistency checks where  $n$  is the number of counted tuples. We are also interested in the investigation of the decidability of  $\mathcal{SC}$  and  $\mathcal{EC}$ .

*Acknowledgments* This work has been supported by the grant by the grant of the Czech Technical University in Prague No. SGS13/204/OHK3/3T/13 Effective solving of engineering problems using semantic technologies. Authors also want to express thanks to the anonymous reviewers for providing useful comments during manuscript preparation.

## References

1. Sirin, E., Parsia, B.: SPARQL-DL: SPARQL Query for OWL-DL. In: 3rd OWL Experiences and Directions Workshop (OWLED-2007). (2007)
2. Kubias, E., Schenk, S., Staab, S., Pan, J.Z.: OWL SAIQL - an OWL DL Query Language for Ontology Extraction. In: In Proc. of OWLED-07. (2007)
3. O'Connor, M.J., Das, A.K.: SQWRL: A query language for OWL. In: OWLED. (2009)
4. Group, W.O.W.: OWL 2 Web Ontology Language Document Overview. W3C Recommendation, W3C (October 2009) <http://www.w3.org/TR/2009/REC-owl2-overview-20091027>, cit. 04.12.2012.
5. Broekstra, J., Kampman, A.: An rdf query and transformation language. In Staab, S., Stuckenschmidt, H., eds.: Semantic Web and Peer-to-Peer. Springer Berlin Heidelberg (2006) 23–39
6. Prud'hommeaux, E., Seaborne, A.: SPARQL Query Language for RDF. W3C Recommendation, W3C (January 2008) <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115>, cit. 3.2013.

7. Horrocks, I., Kutz, O., Sattler, U.: The Even More Irresistible SROIQ. In Doherty, P., Mylopoulos, J., Welty, C.A., eds.: KR, AAAI Press (2006) 57–67
8. Sirin, E., Parsia, B.: Optimizations for Answering Conjunctive ABox Queries. In: Description Logics. Volume 189 of CEUR. (2006)
9. Křemen, P., Kouba, Z.: Conjunctive Query Optimization in OWL2-DL. In: Proceedings of the 22th International Conference on Database and Expert System Applications (DEXA 2011). Volume 6861 of LNCS., Springer Verlag (2011)
10. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive Query Answering in the Description Logic *SHIQ*. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007). (2007)
11. Kollia, I., Glimm, B., Horrocks, I.: Query Answering over SROIQ Knowledge Bases with SPARQL. In: Proceedings of the 2011 International Workshop on Description Logic (DL 2011). (2011)
12. Seaborne, A., Harris, S.: SPARQL 1.1 Query. W3C Working Draft, W3C (October 2009) <http://www.w3.org/TR/2009/TR/2009/WD-sparql11-query-20091022>, cit. 3.2013.
13. Apache: ARQ - A SPARQL Processor for Jena, web site (April 2011) <http://jena.apache.org/documentation/query/index.html>, cit. 3.2013.
14. Corby, O.: Kgram: a knowledge graph abstract machine, web site <http://wimmics.inria.fr/corese>, cit. 3.2013.
15. Williams, G.T.: RDF Query 2.909 - RDF::Query - A complete SPARQL 1.1 Query and Update implementation for use with RDF::Trine, web site (November 2012) <http://search.cpan.org/dist/RDF-Query/>, cit. 3.2013.
16. Arjohn Kampman, Christiaan Fluit, J.B.: Sesame, web site (January 2013) <http://sourceforge.net/projects/sesame>, cit. 3.2013.
17. Seid, D.Y., Mehrotra, S.: Grouping and aggregate queries over semantic web databases. In: ICSC. (2007) 775–782
18. Calvanese, D., Kharlamov, E., Nutt, W., Thorne, C.: Aggregate queries over ontologies. In: ONISW. (2008) 97–104
19. Martínez, D.C., Janowicz, K., Hitzler, P.: A logical geo-ontology design pattern for quantifying over types. In: SIGSPATIAL/GIS. (2012) 239–248
20. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. In Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: Description Logic Handbook, Cambridge University Press (2003)
21. Tao, J., Sirin, E., Bao, J., McGuinness, D.L.: Integrity constraints in owl. In: AAAI. (2010)
22. Motik, B., Parsia, B., Patel-Schneider, P.F.: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C recommendation, W3C (October 2009) <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027>, cit. 12.12.2012.
23. Patel-Schneider, P.F., Motik, B., Grau, B.C.: OWL 2 Web Ontology Language Direct Semantics. W3C Recommendation, W3C (October 2009) <http://www.w3.org/TR/2009/REC-owl2-direct-semantics-20091027>, cit. 12.12.2012.
24. Kostov, B., Křemen, P.: Count aggregation in semantic queries - technical report. Technical report, Czech Technical University in Prague, Dept. of Cybernetics (2013)
25. Křemen, P., Kostov, B.: Expressive OWL Queries: Design, Evaluation, Visualization. International Journal On Semantic Web and Information Systems (2012) IGI Publishing. To appear in 2013.