SCIENCE CHINA
Information Sciences

• LETTER •

December 2018, Vol. 61 129102:1–129102:3
https://doi.org/10.1007/s11432-017-9344-5

# Expressivity issues in SPARQL: monotonicity and two-versus three-valued semantics

Xiaowang ZHANG[1,2*], Chenhong MENG[1,2] & Lei ZOU[3]

[1]*School of Computer Science and Technology, Tianjin University, Tianjin* 300350*, China;*
[2]*Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin University, Tianjin* 300350*, China;*
[3]*Institute of Computer Science and Technology, Peking University, Beijing* 100871*, China*

Received 24 May 2017/Revised 13 October 2017/Accepted 28 December 2018/Published online 6 June 2018

**Citation** Zhang X W, Meng C H, Zou L. Expressivity issues in SPARQL: monotonicity and two-versus three-valued semantics. Sci China Inf Sci, 2018, 61(12): 129102, https://doi.org/10.1007/s11432-017-9344-5

Dear editor,

In 2008, the SPARQL query language was the official world wide web consortium (W3C) recommendation for a resource description framework (RDF) query language [1], which is a database query language that can retrieve and manipulate data stored in the RDF format. SPARQL allows a query to have triple patterns, conjunctions (AND), disjunctions (UNION), optional patterns (OPT) and built-in conditions (constraints) that can be filtered (FILTER).

The official semantics of SPARQL recommended by the W3C is a three-valued semantics in the built-in conditions [1], wherein each mapping is assigned to true, false or error, and error is assigned to mappings that do not contain all of the variables occurring in the constraints [2]. The two-valued semantics of SPARQL, called the two-valued semantics, were introduced by [3] in 2006 wherein each mapping was assigned to either true or false. As a traditional and important problem, the expressivity of SPARQL has attracted considerable attention ever since SPARQL was released. Although the expressivity of SPARQL under the two-valued semantics is the same as the expressivity of SPARQL under the three-valued semantics [4], it is interesting to differentiate the three-valued semantics and two-valued semantics of SPARQL patterns; i.e., the two semantics of a given pattern. Moreover, the primitivity of operators in SPARQL 1.0 has been previously investigated [4]. Although AND is expressed by OPT and FILTER, it is still interesting to distinguish OPT from the other operators, such as AND, under the two semantics. Moreover, although the expressivity of non-monotonic operators in SPARQL 1.1 such as MINUS, BIND, and VALUES have been investigated [5], the expressivity of the non-monotonic fragments persists to be an-open problem, especially with regards to the further classification of the non-monotonic fragments.

In this study, we investigated the expressivity of different operators in SPARQL 1.0, by conducting a comparison among the expressivity of AND, OPT, OPT from well-designed patterns, under the two-valued and three-valued semantics via constraints, monotonicity, weak monotonicity, and non-optionally monotonicity. We summarize the main contributions of this study as follows.

- To differentiate between the two patterns semantics, we present two constraints, namely, the positive and bound constraints. Subsequently, we introduce two kinds of fragments with the corresponding constraints, namely, the positive SPARQL fragment (denoted by $\text{SPARQL}^{+}$) and the bound SPARQL fragment (denoted by $\text{SPARQL}^{\text{b}}$). We demonstrate that the two semantics are equivalent in $\text{SPARQL}^{+}$, while the three-valued semantics are slightly stronger than the two-valued semantics in $\text{SPARQL}^{\text{b}}$.

* Corresponding author (email: xiaowangzhang@tju.edu.cn)

 

• To distinguish OPT from AND, we present the monotonicity of SPARQL, which is defined as normal [6], and we provide proofs by which OPT dissatisfies monotonicity, while AND preserves it. Moreover, we demonstrate that $\mathcal{AFU}$ is the maximal fragment of SPARQL 1.0 preserving monotonicity under the two semantics.

• To distinguish OPT from patterns and OPT from well-designed patterns, where OPT is restricted [3], we introduce the non-optionally monotonicity of SPARQL, which extends the weak monotonicity [7] on non-optional sub-patterns, and demonstrate that $\mathcal{OU}$ is the maximal fragment of SPARQL 1.0 preserving the non-optionally monotonicity under the two semantics. However, we show that the non-optionally monotonicity fails in $\mathcal{AO}$ and $\mathcal{FO}$.

*Two-valued pattern semantics versus-three-valued pattern semantics.* To accurately differentiate the two semantics of SPARQL patterns, we first introduce two kinds of constraints to be filtered, namely, positive constraints and bound constraints. Formally, positive constraint ($C^+$) and bound constraint ($C^{\mathrm{b}}$) are defined as follows.

• Constraint in $C^+$ is formulated as follows:

$$\mathrm{bound}(?x) \mid ?x = c \mid ?x = ?y \mid C_1 \vee C_2 \mid C_1 \wedge C_2.$$

• Constraint in $C^{\mathrm{b}}$ is constructed as follows:

$$\mathrm{bound}(?x) \mid ?x = c \mid ?x = ?y \mid ?x \neq c \mid ?x \neq ?y \mid C_1 \vee C_2 \mid C_1 \wedge C_2.$$

Intuitively, $C^+$ is ¬-free and $C^{\mathrm{b}}$ is ¬bound-free. Now, let $\mathcal{F}$ be a fragment of SPARQL. We use $\mathcal{F}^+$ and $\mathcal{F}^{\mathrm{b}}$ to denote the sub-fragments of $\mathcal{F}$ only consisting of $C^+$ and $C^{\mathrm{b}}$, respectively.

We denote SPARQL$^+$ as $\mathcal{AFOU}^+$, which is called the positive SPARQL and SPARQL$^{\mathrm{b}}$ as $\mathcal{AFOU}^{\mathrm{b}}$, which is called the bound SPARQL.

The following result states that the two semantics are equivalent in the positive SPARQL.

**Proposition 1.** The two-valued pattern semantics are equivalent to the three-valued pattern semantics in SPARQL$^+$.

However, the equivalence between the two-valued pattern semantics and three-valued pattern semantics does not preserve SPARQL$^{\mathrm{b}}$ patterns.

**Proposition 2.** The two-valued pattern semantics are not equivalent to the three-valued pattern semantics in SPARQL$^{\mathrm{b}}$.

However, the expressivity of SPARQL under the two-valued semantics is the same as the expressivity of SPARQL under the three-valued semantics [4, Proposition 17].

Next, we discuss the expressivity of SPARQL$^{\mathrm{b}}$ patterns under the two semantics, since only ¬bound is not allowed in SPARQL$^{\mathrm{b}}$ with respect to SPARQL.

Formally, we should define the meaning of some fragment $\mathcal{F}$ being "expressible" in some fragment $\mathcal{F}'$. Here, we will simply interpret that for every pattern $P$ in fragment $\mathcal{F}$, there exists a pattern $Q$ in the given fragment $\mathcal{F}'$, such that for every RDF graph $G$, we have $[\![P]\!]_G = [\![Q]\!]_G$.

Finally, we demonstate that SPARQL is expressible in SPARQL$^{\mathrm{b}}$.

**Proposition 3.** SPARQL is expressible in SPARQL$^{\mathrm{b}}$ under the two semantics.

As a result, the expressivity of SPARQL$^{\mathrm{b}}$ is the same as the expressivity of SPARQL under the two semantics.

*Monotonicity of* SPARQL. We will use monotonicity to distinguish OPT from other operators such as AND. Indeed, we know that the OPT operator is primitive [4]; i.e., each fragment comprising of OPT is not expressible in any OPT-free fragment.

Formally, the monotonicity of SPARQL can be defined as follows: let $P$ be a pattern in SPARQL. $P$ is monotonic if $[\![P]\!]_{G_1} \subseteq [\![P]\!]_{G_2}$ for any $G_1$ and $G_2$, $G_1 \subseteq G_2$. A fragment $\mathcal{F}$ is called monotonic if all patterns in $\mathcal{F}$ are monotonic.

Next, we discuss which fragment of SPARQL is monotonic under the two semantics.

Note that all OPT-free SPARQL patterns are monotonic under the three-valued semantics [4].

We also conclude that $\mathcal{AFU}$ is monotonic under the two-valued semantics since two semantics are equivalent [4, Proposition 17].

**Proposition 4.** $\mathcal{AFU}$ is monotonic under the two-valued semantics.

In general, not all SPARQL patterns are monotonic. We would like to know whether there exists a monotonic fragment comprising of OPT. Here, the answer is negative.

**Proposition 5.** Any fragment consisting of OPT is non-monotonic under the two semantics.

In conclusion, by Propositions 4 and 5, we can demonstrate that the monotonicity can be used to distinguish OPT from other operators.

However, the monotonicity cannot be used to distinguish SPARQL$^+$ from SPARQL$^{\mathrm{b}}$ if OPT is absent under the two semantics.

**Corollary 1.** Both OPT-free SPARQL$^+$ and OPT-free SPARQL$^{\mathrm{b}}$ are monotonic under the two semantics.

*Non-optional monotonicity of* SPARQL. The cause of the monotonicity failing in any fragment comprising of OPT is that the set inclusion relationship is too strong for OPT in characterizing some "non-decreasing"-like property (by Proposition 5, OPT causes non-monotonicity).

(1) Weak monotonicity of SPARQL. Let $\mu_1$ and $\mu_2$ be the two mappings. We say that $\mu_1$ is subsumed by $\mu_2$ [3], written by $\mu_1 \sqsubseteq \mu_2$, if $\mathrm{dom}(\mu_1) \subseteq \mathrm{dom}(\mu_2)$ and $\mu_1 \sim \mu_2$.

Let $\Omega_1$ and $\Omega_2$ be two sets of mappings. We define $\Omega_1 \sqsubseteq \Omega_2$ if for all $\mu_1 \in \Omega_1$, there exists some $\mu_2 \in \Omega_2$ such that $\mu_1 \sqsubseteq \mu_2$ (defined in [3]).

A weak monotonicity to such a "non-decreasing"-feature for $\sqsubseteq$ in the general variables is introduced in [7]. Formally, a fragment $\mathcal{F}$ is weakly monotonic if for all $\mathcal{F}$-pattern $P$, for any two RDF graphs $G_1$ and $G_2$, $G_1 \subseteq G_2$ implies $[\![P]\!]_{G_1} \sqsubseteq [\![P]\!]_{G_2}$ [8].

Well-designed patterns are introduced in order to characterize weak monotonicity [3]. All well-designed patterns are weakly monotonic [7].

(2) Non-optional monotonicity. Let $P$ be a pattern. We use $\Delta(P)$ to denote the least reduction of $P$ (defined in [3]).

Formally, given a pattern $P$, $P$ is called the non-optionally monotonic if for any two RDF graphs $G_1$ and $G_2$, $G_1 \subseteq G_2$; then, for any mapping $\mu_1 \in [\![P]\!]_{G_1}$, there exists some mapping $\mu_2 \in [\![P]\!]_{G_2}$ such that the following holds: for any mapping $\mu \in [\![\Delta(P)]\!]_{G_1}$, $\mu \sqsubseteq \mu_1$ implies $\mu \sqsubseteq \mu_2$.

Now, we discuss the relationship between weak monotonicity and non-optionally monotonicity.

**Proposition 6.** For any pattern $P$, if $P$ is weak monotonic then $P$ is non-optionally monotonic under the two semantics, but not vice versa.

**Proposition 7.** For any pattern $P$ in $\mathcal{AFU}$, $P$ is monotonic, weak monotonic, and non-optionally monotonic under the two semantics.

The following result shows that the non-optionally monotonicity and the weak monotonicity are identical even for well-designed patterns.

**Proposition 8.** Each well-designed pattern is weak monotonic and non-optionally monotonic under the two semantics.

(3) Non-optional monotonicity of SPARQL. A fragment $\mathcal{F}$ is non-optionally monotonic if all patterns in $\mathcal{F}$ are non-optionally monotonic. We investigate the non-optionally monotonicity of all SPARQL fragments.

Next, we show that $\mathcal{OU}$ satisfies the non-optionally monotonicity.

**Proposition 9.** $\mathcal{OU}$ is non-optionally monotonic under the two semantics.

Now, we would like to know whether the two fragments $\mathcal{AO}$ and $\mathcal{FO}$ are non-optionally monotonic. Here, the answer is negative.

**Proposition 10.** $\mathcal{AO}$ and $\mathcal{FO}$ are not non-optionally monotonic under the two semantics.

We introduce the difference operator DIFF for patterns [6, 9] in order to answer $\mathcal{FO}$ with two classes of constraints.

**Proposition 11.** Let $\mathcal{F}$ be a fragment of SPARQL. If DIFF is expressible in $\mathcal{F}$, then $\mathcal{F}$ is not non-optionally monotonic under the two semantics.

Now, we demonstrate that $\mathcal{OU}$ is the maximal fragment in SPARQL 1.0 preserving the non-optionally monotonicity under the two semantics.

**Proposition 12.** $\mathcal{FO}^{+}$ and $\mathcal{FO}^{\mathrm{b}}$ are not non-optionally monotonic under the two semantics.

*Conclusion.* This study investigated the expressivity issues of the three-valued and the two-valued pattern semantics and some interesting SPARQL 1.0 fragments under the two semantics.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant No. 61502336) and National Key Research and Development Program of China (Grant No. 2016YFB1000603).

**Supporting information** Appendix: Preliminaries and Proofs. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

## References

1 Prudhommeaux E, Seaborne A. SPARQL query language for RDF. W3C Recommendation. 2008. https://www.w3.org/TR/rdf-sparql-query/
2 Angles R, Gutierrez C. The expressive power of SPARQL. In: Proceedings of the 7th International Conference on the Semantic Web, Karlsruhe, 2008. 114–129
3 Pérez J, Arenas M, Gutierrez C. Semantics and complexity of SPARQL. ACM Trans Database Syst, 2009, 34: 1–45
4 Zhang X, Van den Bussche J. On the primitivity of operators in SPARQL. Inf Process Lett, 2014, 114: 480–485
5 Kontchakov R, Kostylev E. On expressibility of non-monotone operators in SPARQL. In: Proceedings of International Conference Principles of Knowledge Representation and Reasoning (KR-16), Cape Town, 2016. 369–379
6 Zhang X W, Van den Bussche J, Picalausa F. On the satisfiability problem for SPARQL patterns. J Artif Intel Res, 2016, 56: 403–428
7 Arenas M, Gutierrez C, Miranker D P, et al. Querying semantic data on the web? SIGMOD Rec, 2013, 41: 6–17
8 Kaminski M, Kostylev E. Beyond well-designed SPARQL. In: Proceedings of International Conference on Database Theory (ICDT-16), Bordeaux, 2016
9 Zhang X W, Van den Bussche J, Wang K, et al. On the satisfiability problem of patterns in SPARQL 1.1. In: Proceedings of AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, 2018. 2054–2061