Appendix to ShEx Specification (Proof that the semantics is independent on the chosen stratification)

Iovka Boneva

1 Definitions

We use the following notions defined in the ShEx 2.0 specification.

Consider an arbitrary RDF graph G, and an arbitrary ShEx schema Sch than satisfies the *schema requirements* as defined in Section 5.7 of ShEx Specification, fixed for the sequel of the document.

The *shapes* of Sch are as defined in Section 5.5.1 of ShEx Specification.

The *dependency graph* of Sch is as defined in Section 5.7.4 of ShEx Specification. Its vertices are the shapes of Sch.

A *reference*, resp. *negated reference*, from shape s_1 to shape s_2 of Sch, are as defined in Section 5.7.4 of ShEx Specification.

We recall that a *typing* of G and Sch is a set of pairs (n, s) where n is a node in G and s is a shape in Sch.

A *correct typing* is as defined in Section 5.2 of ShEx Specification.

We recall that the number of *strata* of Sch is the number of maximal strongly connected components (written mscc for short) of its dependency graph.

Let k be the number of strata of Sch. We assume that $k \geq 2$.[1]

We recall that a *stratification* of Sch is a function *stratum* that with every shape of Sch associates a natural between 1 and k s.t.

- If the shapes s_1 and s_2 belong to the same mscc, then $\mathit{stratum}(s_1) = \mathit{stratum}(s_2)$
- If there is a reference from shape s_1 to s_2 in Sch and s_1, s_2 do not belong to the same mscc, then $\mathit{stratum}(s_2) < \mathit{stratum}(s_1)$.

We recall the definition of $\mathsf{completeTypingOn}(i, G, Sch)$ for $1 \leq i \leq k$. In ShEx Specification it is defined w.r.t. some fixed stratification, here we extend that definition by giving the stratification as parameter. Thus, let *stratum* be a stratification of Sch. We define

- $\mathsf{completeTypingOn}^{\mathit{stratum}}(1, G, Sch)$ is the union of all correct typings that contain only pairs (n, s) with $\mathit{stratum}(s) = 1$;
- for every $1 \leq i \leq k$, $\mathsf{completeTypingOn}^{\mathit{stratum}}(i, G, Sch)$ is the union of all correct typings that:

[1] If $k = 1$, then there exists a unique stratification of Sch and the property of the semantics that we want to show is trivial.

– contain only pairs (n, s) with $stratum(s) \leq i$, and
– are equal to $\mathsf{completeTypingOn}^{stratum}(i-1, G, Sch)$ when restricted to their pairs (n', s') for which $stratum(s') < i$.

For a *typing*, an RDF node n and a shape s, the predicate $\mathsf{matches}(n, s, G, Sch, typing)$ is as defined is Section 5.5.2 of ShEx Specification.

We will show that

Theorem 1. *For any two stratifications $stratum_1$ and $stratum_2$ of Sch, it holds*

$$\mathsf{completeTypingOn}^{stratum_1}(k, G, Sch) = \mathsf{completeTypingOn}^{stratum_2}(k, G, Sch).$$

2 Proof of the theorem

The proof is an adaptation of the proof that the semantics of a stratified Datalog program is independent on the choice of a stratification, as shown in Theorem 15.2.10 in [2]

We start by recalling some folklore results.

A mscc of a graph is a subgraph induced by some set of vertices, therefore can be identified with that set set of vertices. Let V be the set of mscc of the dependency graph of *Sch*, that is, the elements of V are sets of shapes lying on the same mscc of the dependency graph of *Sch*. We denote with $[k]$ the set $\{1, \ldots, k\}$.

A stratification *stratum* of *Sch* can be lifted to a function from V to $[k]$ by: for every $C \in V$, $stratum(C)$ is the unique $1 \leq j \leq k$ s.t. $stratum(s) = j$ for some shape s in C.

By the definition of a stratification, it follows that

Claim 2. *$stratum : V \to [k]$ is a bijection for every stratum stratification of Sch.* □

As usual, $stratum^{-1}$ denotes the inverse function of *stratum*.

Let D be the graph which set of vertices is V and that has an edge (C, C') iff there exist s shape in C and s' shape in C' s.t. there is a reference from s to s' in *Sch*. For C, C' two mscc of *Sch*, we write $C \prec C'$ if there is a path from C' to C in D.

As it is usual with stratification (e.g. with stratified Datalog programs):

Claim 3. *The graph D is acyclic and $\prec$ is a partial ordering relation on V; we denote $\preceq$ its reflexive closure.* □

Claim 4. *Every stratum stratification of Sch satisfies $stratum(C) < stratum(C')$ iff $C \prec C'$.* □

In other words, every stratification of *Sch* is a linearization of the partial ordering $\prec$. Threfore, if *stratum* and *stratum'* are two stratifications of *Sch*, then *stratum'* can be obtained from *stratum* by a finite sequence of permutation of two adjacent $\prec$-incomparable elements. More formally:

[2]Negation in Datalog. Chapter 15 in *Foundations of Databases* by Serge Abiteboul, Rick Hull and Victor Vianu. Published by Addisson Wesley, 1994.

Claim 5. *If stratum and stratum′ are two stratifications of Sch, then there exists a finite sequence of stratifications* $stratum_1, \ldots, stratum_n$ *s.t.* $stratum = stratum_1$, $stratum' = stratum_n$, *and for every* $1 \leq i < n$, *there exists a natural* $1 \leq j < k$ *s.t. stratum and stratum′ differ only on their pre-images for* j *and* $j+1$, *with:*

- $stratum_{i+1}^{-1}(j) = stratum_i^{-1}(j+1)$,
- $stratum_{i+1}^{-1}(j+1) = stratum_i^{-1}(j)$, *and*
- $stratum_i^{-1}(j)$ *and* $stratum_i^{-1}(j+1)$ *are incomparable for the* $\prec$ *ordering relation.*

□

We now give the elements of the proof that are specific to the semantics of ShEx.

Lemma 6. *Let stratum and stratum′ be two stratifications of Sch and* $1 \leq j < k$ *s.t. stratum and stratum′ differ only on their pre-images for* j *and* $j+1$. *Then* $\mathsf{completeTypingOn}^{stratum}(j+1, G, Sch) = \mathsf{completeTypingOn}^{stratum'}(j+1, G, Sch)$.

An immediate corollary of the above lemma is that with the same hypotheses, $\mathsf{completeTypingOn}^{stratum}(k, G, Sch) = \mathsf{completeTypingOn}^{stratum'}(k, G, Sch)$.

Then Proposition 1 is shown by applying inductively Lemma 6 on the finite sequence of local permutations of $\prec$-incomparable elements described in Claim 5 that allow to change *stratum* to *stratum′*.

In the sequel we prove Lemma 6, starting by some technical results.

The following claim is a consequence of the definitions of matches predicate and is shown using an induction on the $\prec$ ordering relation. It intuitively states that whether a node matches a shape s depends only on the shapes to which s refers directly or indirectly.

Claim 7. *For every* n, s *and typing, it holds that*

$$\mathsf{matches}(n, s, G, Sch, typing) \text{ iff } \mathsf{matches}(n, s, G, Sch, typing_{\preceq s})$$

where $typing_{\preceq s}$ *is typing restricted only on those shapes that precede* s *for the* $\preceq$ *ordering. Formally, if* C_s *is the mscc of Sch that contains* s *and* $Nodes(G)$ *is the set of RDF nodes in* G, *then*

$$typing_{\preceq s} = typing \cap \left(Nodes(G) \times \bigcup_{C \in V, C \preceq C_s} C \right)$$

□

The following claim is a technical corollary of Claim 7.

Claim 8. *Let* $1 < j < k$ *and let* $C_1, \ldots, C_{j-1}, C_j, C_{j+1}$ *be a sequence of distinct elements of* V *compatible with the* $\prec$ *ordering, and s.t.* C_j *and* C_{j+1} *are incomparable for* $\prec$. *That is:*

- *$C_i \in V$ for any $1 \leq i \leq j+1$, and*
- *if $i < l$, then $C_l \not< C_i$, and*
- *$C_j \not< C_{j+1}$ and $C_{j+1} \not< C_j$.*

Let $C = \bigcup_{1 \leq i \leq j-1} C_i$. Let T be a typing using only shapes from C, $typing_j$ be a typing using only shapes from C_j, and $typing_{j+1}$ be a typing using only shapes from C_{j+1}.

Then $T \cup typing_j \cup typing_{j+1}$ is a correct typing iff $T \cup typing_j$ and $T \cup typing_{j+1}$ are both correct typings. □

Proof of Lemma 6. Denote $T_x = \mathsf{completeTypingOn}^{stratum}(x, G, Sch)$ and $T'_x = \mathsf{completeTypingOn}^{stratum'}(x, G, Sch)$ for $x \in [k]$, end let $T_0 = T'_0 = \emptyset$. It immediately follows from the hypotheses that $T_{j-1} = T'_{j-1}$, we set $T = T_{j-1}$ in the sequel.

Let $typing_j$ be the restriction of T_j on the shapes in C_j, $typing_{j+1}$ be the restriction of T_{j+1} on the shapes in C_{j+1}, and similarly $typing'_j$ be the restriction of T'_j on C_{j+1} and $typing'_{j+1}$ be the restriction of T'_{j+1} on C_j.

Then by definition of completeTypingOn it follows that T_j, T_{j+1}, T'_j and T'_{j+1} can be written as the disjoint unions:

- $T_{j+1} = T \cup typing_j \cup typing_{j+1}$,
- $T'_{j+1} = T \cup typing'_j \cup typing'_{j+1}$.

It also follows by Claim 8 and by the hypotheses that these four are correct typings:

- $T \cup typing_j$
- $T \cup typing_{j+1}$
- $T \cup typing'_j$
- $T \cup typing'_{j+1}$

Still using Claim 8 and the definitions we can show that $T_j = T \cup typing_j$ and $T'_j = T \cup typing'_j$, and finally that $typing_j = typing'_{j+1}$ and $typing_{j+1} = typing'_j$, from which the lemma follows immediately. □