Supplementary Material for: Knowledge Graph Construction with a *Façade*: A Unified Method to Access Heterogeneous Data Sources on the Web

LUIGI ASPRINO, University of Bologna, Italy
ENRICO DAGA, The Open University, UK
ALDO GANGEMI, Consiglio Nazionale delle Ricerche (CNR) and University of Bologna, Italy
PAUL MULHOLLAND, The Open University, UK

APPENDICES

A FORMAL DEFINITIONS

Definition A.1 (Grammar). A grammar G is defined as the tuple $G = (N, \Sigma, P, S)$ where:

- N is a finite set of non-terminal symbols;
- Σ is a finite set of terminal symbols;
- P is a finite set of production rules, each rule of the form $(\Sigma \cup N)^* N (\Sigma \cup N)^* \rightarrow (\Sigma \cup N)^*$;
- S is a distinguished start symbol $S \in N$.

Definition A.2 (Derivation Relation). Given a grammar G, we define the relation $\underset{G}{\Rightarrow}$ (pronounced G derives in one step) as follows:

$$x \underset{G}{\Rightarrow} y \iff \exists u, v, p, q \in (\Sigma \cup N)^* | (x = upv) \wedge (p \rightarrow q \in P) \wedge (y = uqv).$$

Given a grammar G, we define the relation $\overset{*}{\underset{G}{\Rightarrow}}$ (or $\overset{*}{\Rightarrow}$ shortly) (pronounced G derives in zero or more steps) as the reflexive transitive closure of $\underset{G}{\Rightarrow}$.

Definition A.3 (Sentence). A sentence s is any sequence of non-terminal symbols of a grammar G, i.e., $s \in \Sigma^*$.

Definition A.4 (Language). Given a grammar G, we define the language of G, denoted $L(G)$ as all the sentences s that can be derived in a finite number of steps from the start symbol S, that is $L(G) = \{s \in \Sigma^* | S \overset{*}{\underset{G}{\Rightarrow}} s\}$.

Definition A.5 (Delimiters). Given a data format F whose syntax is defined by the grammar G, we define Σ^F as the set of delimiters.

Definition A.6 (Plain Language). Given a data format F whose syntax is defined by the grammar G, we define:

- $\widetilde{\Sigma}$ is the set of value symbols, i.e., $\widetilde{\Sigma} := \Sigma \setminus \Sigma^F$;
- V is the set of all possible values that can be derived from G, that is $V := \{v \in \widetilde{\Sigma}^*\}$;
- Given a sentence s derived from a grammar G, a plain sentence is obtained by removing the delimiters from s, i.e., $\tilde{s} = (v_1 \ldots v_n) \in V^*$;

1533-5399/2023/02-ART6 $15.00
https://doi.org/10.1145/3555312

- $\widetilde{L(G)}$ is the set of all the plain sentences that can be derived from G, that is: $\widetilde{L(G)} := \{\tilde{s} \in V^* | S \overset{*}{\underset{G}{\Longrightarrow}} w\}$.

Definition A.7 (Projection). Given a sentence $s = (v_1 ... v_l)$ of length l, two integers $1 \leq i \leq l$ and $i \leq j \leq l$, we define the projection onto i, j (written $\Pi_i^j(s)$) as the sub-sentence $\Pi_i^j(s) = (v_i ... v_j)$.

Definition A.8 (Reification). A Reification is defined as a function R that associates every tuple to a **uniform resource identifier (URI)**, that is $R : V^* \rightarrow U$ where U denotes the set of all URIs.

Definition A.9 (Container). A container is a tuple whose elements are either values or URIs, that is $C \subseteq (U \cup V)^*$, where C encloses all the containers.

Definition A.10 (Dereference). We define dereference as the function $(D : C \times \mathbb{N} \rightarrow C)$ that given a container $(c = (v_1 ... v_n) \in C)$ and an integer $1 \leq i \leq |c|$, returns a tuple in which v_i is replaced with $R^-(v_i)$, i.e., $D(c, i) = (v_1 .. R^-(v_i) .. v_n)$.

Definition A.11 (Dereference).* We define D^* as the function $(D^* : C \rightarrow V^*)$ that given a container $(c = (v_1 ... v_n) \in C)$ returns, by recursively applying the dereference function, a tuple whose elements are only values.

Definition A.12 (Database Scheme). A database scheme is defined as the tuple $D = (R, U, \Delta)$ where:

- $U \subseteq \mathfrak{U}$ is a finite subset of attributes;
- Δ is the union of the domains of all the attributes, i.e., $\Delta = \bigcup_{A \in U} \Delta(A)$;
- R is a collection of relations R_i over U.

Definition A.13 (Tuple, Relation, and Database). Let $D = (R, U, \Delta)$ be a database scheme, $R_i \in R$ a relation scheme and X a subset of U. We define:

- X-tuple is a mapping from X to Δ such that each $A \in X$ is mapped to an element $\Delta(A)$;
- A relation r over R_i is a finite set of tuple R-tuple;
- A relational database d over D is a set of relations.

B JSON GRAMMAR

```
JSON -> element
value ->  object|array|string|number|"true"|"false"|null
object -> '{' white-spaces '}' | '{' members '}'
members -> member | member ',' members
member -> white-spaces string white-spaces ':' element
array -> '[' white-spaces ']' | '[' elements ']'
elements -> element | element ',' elements
element -> white-spaces value white-spaces
```