

Query Typing for the Tripcom Query Language

Philipp Obermeier

Consortium Networked Information Systems
Department of Informatics, Free University of Berlin

September 30, 2007

Table of contents

- 1 The purpose of Query Typing
 - What is Query Typing?
 - Typing aspects (meta-types) and their benefits
- 2 A conceptual query typing approach
 - ECFG - an algebraic Model for the SPARQL-grammar
 - Syntax-tree-based query typing
 - A typing example

What is Query Typing?

Query typing is a categorization of queries under certain aspects.

Typing aspects (meta-types) and their benefits

- **complexity**: Optimization of semantic query processing (smart distribution of query processing jobs according to their costs)
- **processing-relevant query structures**: Optimization of semantic query processing (separate, best-fit evaluation subroutines for different query types)
- **correctness**: Support for syntactic/semantic correctness checking
- **safety**: Prevention of “dangerous” queries
- **security**: Support for access authorization

ECFG - an algebraic Model for the SPARQL-grammar

- **SPARQL** as preliminary proxy for the yet to be designed Tripcom Query Language
- **ECFG** as model for the SPARQL-grammar as given in the W3C-Recommendation

Definition (semi-formal)

An **Extended Context-Free Grammar (ECFG)** is a context-free grammar where every non-terminal A may have infinite rules as long as the right-hand-sides form a regular language (referred with L_A) over the terminals and nonterminals.

Syntax-tree-based query typing

- Typing based on query syntax tree structures
 \rightsquigarrow A **type of a (sub-) query-term** is a nonterminal of the “static type environment” - ECFG
- **Type system calculus** which intuitionial describes a non-deterministic bottom-up (resp. recursive top-down) typing of the nodes/subtrees in a query syntax tree

A typing example (I)

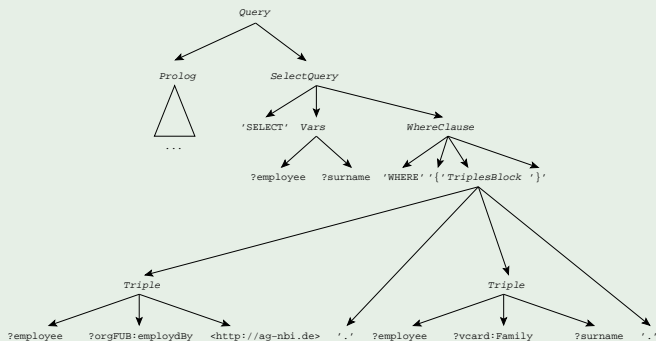
exemplary SPARQL query

```
PREFIX orgFUB: <http://org.fu.berlin.de/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>

SELECT ?employee ?surname
WHERE {
    ?employee orgFUB:employedBy <http://ag-nbi.de> .
    ?employee vcard:Family ?surname .
}
```

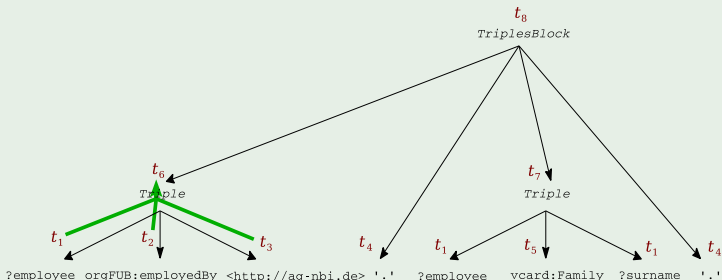
A typing example (II)

query syntax tree (simplified)



A typing example (III)

The typed syntax tree of TriplesBlock



Remark: Multiple types per node possible!

A typing example (IV)

The underlying typing system environment

$$L_{t_1} = \{ '? \} \circ \{ 'a-b', 'A-Z' \}^*$$

$$L_{t_2} = \{ 'orgFUB:employedBy' \} \circ \{ 'a-b', 'A-Z' \}^*$$

$$L_{t_3} = \{ '<http://ag-nbi.de>' \}$$



⋮

$$L_{t_6} = L_{t_1} \circ L_{t_2} \circ L_{t_3}$$





⋮

Remark: Determination of the types' right-hand-side languages crucial for their semantic expressiveness!

References I

-  Jean Berstel and Luc Boasson.
Balanced Grammars and Their Languages.
In Wilfried Brauer, Hartmut Ehrig, Juhani Karhumäki, and Arto Salomaa, editors, *Formal and Natural Computing*, volume 2300 of *Lecture Notes in Computer Science*, pages 3–25. Springer, 2002.
-  Jean Berstel and Luc Boasson.
Formal Properties of XML Grammars and Languages.
ACTAINF: Acta Informatica, 38, March 2002.

References II

-  Anne Brüggemann-Klein and Derick Wood.
On Predictive Parsing and Extended Context-Free Grammars.
In Jean-Marc Champarnaud and Denis Maurel, editors, *CIAA*,
volume 2608 of *Lecture Notes in Computer Science*, pages
239–247. Springer, 2002.
-  Luca Cardelli.
Type Systems, chapter 103.
CRC Press, Boca Raton, FL, 1997.
-  Benjamin C. Pierce.
Types and programming languages.
MIT Press, Cambridge, MA, USA, 2002.
-  SPARQL query language for RDF.
<http://www.w3.org/TR/rdf-sparql-query/>, June 2007.