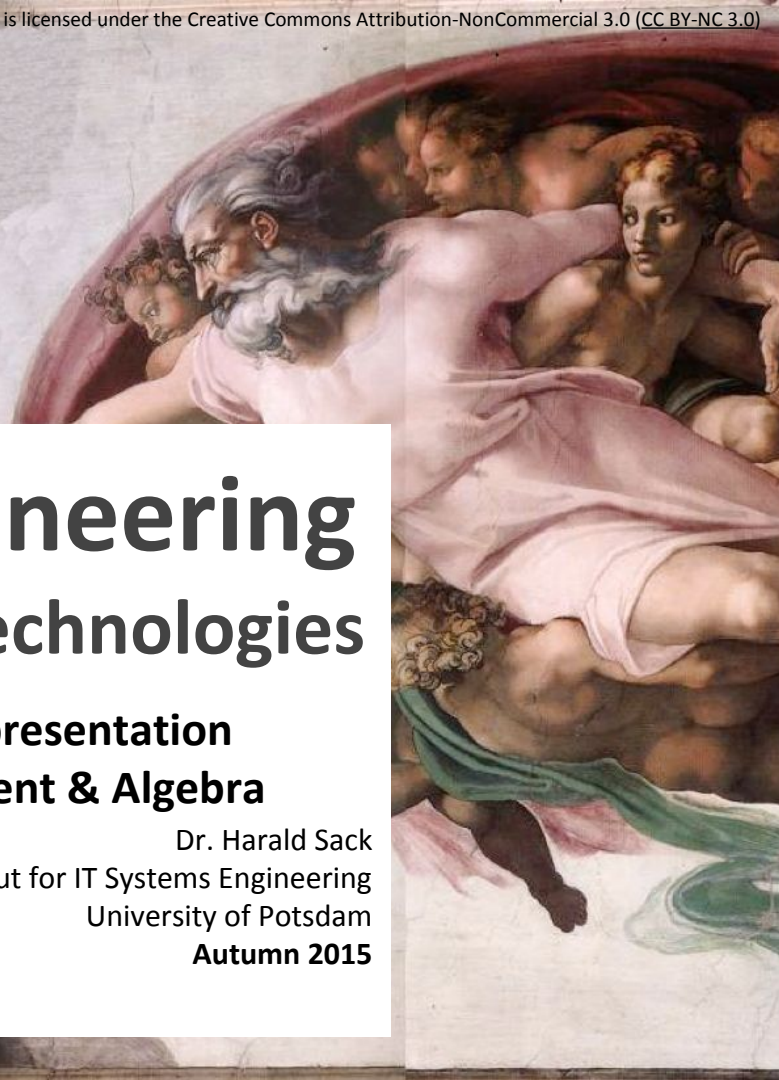
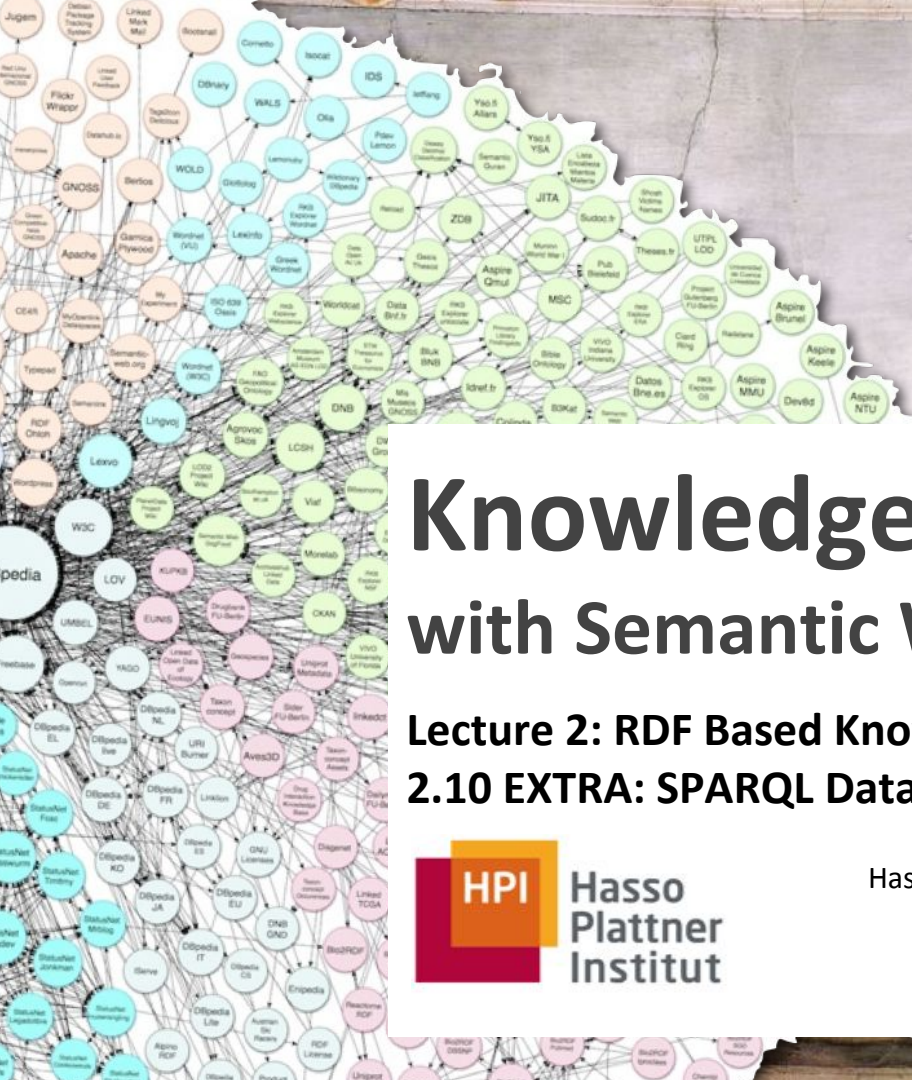




Knowledge Engineering with Semantic Web Technologies

Lecture 2: RDF Based Knowledge Representation

2.10 EXTRA: SPARQL Data Management & Algebra



Dr. Harald Sack
Hasso-Plattner-Institut for IT Systems Engineering
University of Potsdam
Autumn 2015

SPARQL - Update

- Inserting and Deleting RDF Data

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
DELETE DATA
```

```
{ GRAPH <http://example/books>  
  { <http://example/book101> dc:title "From the Earth to the Moon" }  
} ;
```

```
PREFIX dc:      <http://purl.org/dc/elements/1.1/>
```

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
INSERT DATA
```

```
{ GRAPH <http://example/bookStore>  
  { <http://example/book101> dc:title "Around the World in 80 Days" ;  
    dc:creator dbpedia:Jules_Verne . }  
}
```

SPARQL - Update

- Inserting and **Deleting** RDF Data based on bindings for a query pattern

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
WITH <http://example/addresses>
```

```
DELETE {
```

```
    ?person ?property ?value
```

```
}
```

```
WHERE {
```

```
    ?person ?property ?value ;  
        foaf:givenName 'Harald' .
```

```
}
```

- **Inserting** and Deleting RDF Data based on bindings for a query pattern

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

INSERT {
  GRAPH <http://example/bookStore> {
    ?book ?p ?o .
  }
}

WHERE {
  GRAPH <http://example/books> {
    ?book dc:author ?author_name .
    FILTER REGEX(?author_name, "dickens", "i") .
    ?book ?p ?o .
  }
}
```

SPARQL - Data Management

- Bulk Load RDF data from a file

```
LOAD [SILENT] <URI-of-RDF-data> [INTO GRAPH <Graph-URI>]
```

- Delete all RDF data from an RDF graph

```
CLEAR [SILENT] [GRAPH <Graph-URI>]
```

- Create a new RDF graph

```
CREATE [SILENT] [GRAPH <Graph-URI>]
```

- Drop an RDF graph

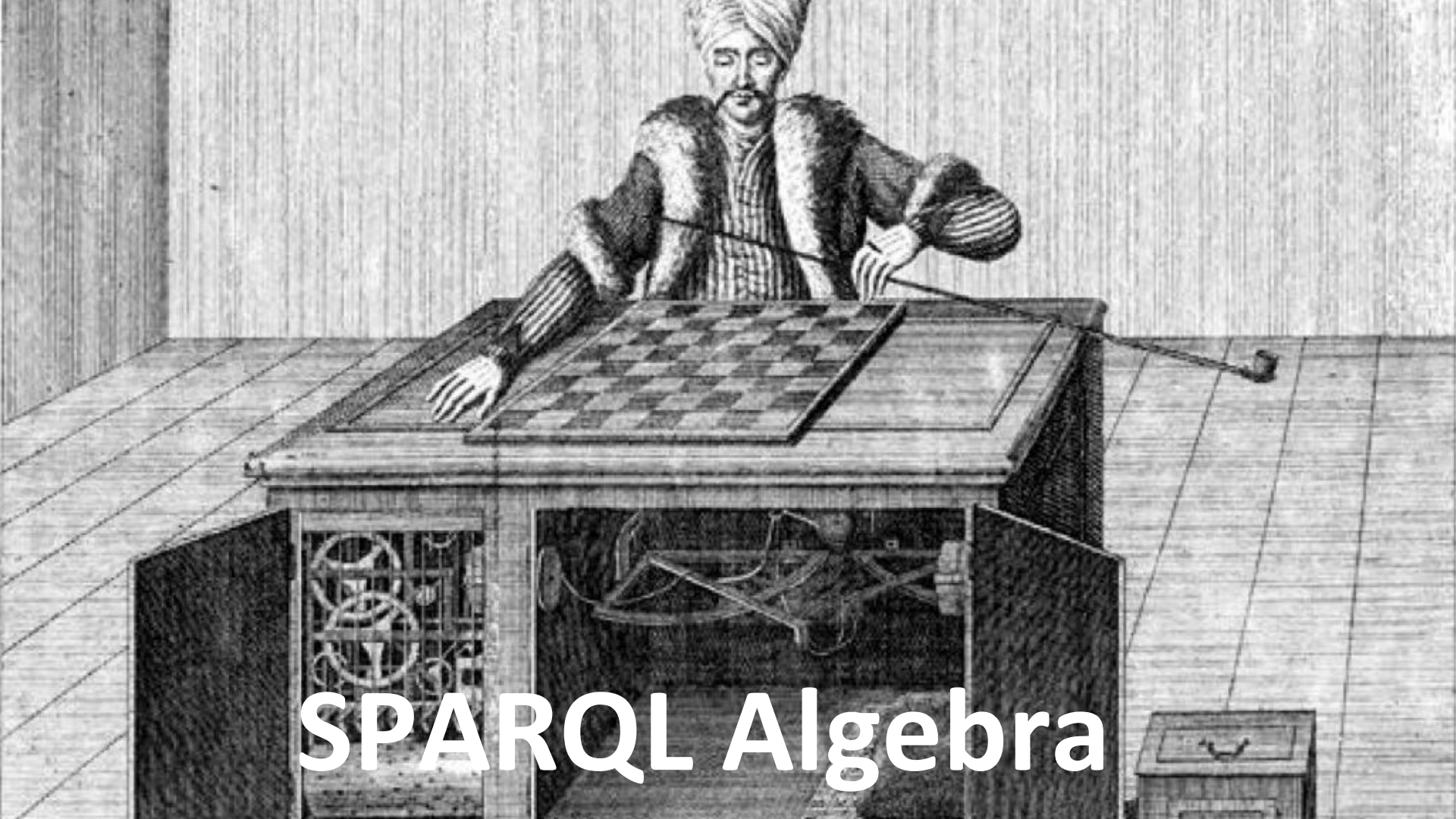
```
DROP [SILENT] [GRAPH <Graph-URI>]
```

- Copy, move, or add an RDF graph to another RDF graph

```
COPY [SILENT] [GRAPH <Graph-URI>] TO [GRAPH <Graph-URI>]
```

```
MOVE [SILENT] [GRAPH <Graph-URI>] TO [GRAPH <Graph-URI>]
```

```
ADD [SILENT] [GRAPH <Graph-URI>] TO [GRAPH <Graph-URI>]
```



SPARQL Algebra

SPARQL Algebra

- Algebra-based approach to define the **SPARQL semantics**
- Definition of “*correct behavior for evaluation of graph patterns and solution modifiers*” [SPARQL Specification]
- Series of steps:
 - (1) **Parsing** a SPARQL query into an **Abstract Syntax Tree**
 - (2) **Converting** the Abstract Syntax Tree to an **Abstract SPARQL Query**
(i.e. an algebra expression)
 - (3) **Evaluating** the Abstract SPARQL Query on an RDF dataset

SPARQL Abstract Syntax Tree

- During parsing of a query, a parse tree is constructed
- The parse tree contains objects that correspond to the grammar
- For the transformation, we traverse the parse tree and recursively build the algebra objects
- The query pattern is a `GroupGraphPattern` consisting of the following elements

```

GroupGraphPattern      ::= '{' TriplesBlock?
                        (( GraphPatternNotTriples | Filter ) ' '?
                          TriplesBlock? ) *
                        '}'

GraphPatternNotTriples ::= OptionalGraphPattern | GroupOrUnionGraphPattern
                        | GraphGraphPattern

OptionalGraphPattern   ::= 'OPTIONAL' GroupGraphPattern
GraphGraphPattern      ::= 'GRAPH' VarOrIRIref GroupGraphPattern
GroupOrUnionGraphPattern ::= GroupGraphPattern ( 'UNION' GroupGraphPattern ) *
Filter                 ::= 'FILTER' Constraint
  
```

SPARQL Algebra Computational Operations

- **Graph Patterns Operations**

each operator returns the result of the subquery it describes

- BGP (basic graph pattern)
- Join (conjunctive patterns)
- LeftJoin (optional conditions)
- Filter
- UNION (alternative patterns)
- GRAPH (named graphs)

- **Solution Sequence Modifiers**

- OrderBy
- Project
- Distinct
- Reduced
- Slice (limit offset)
- ToList (constructs sequence from multiset)

Converting to an Abstract Query

- A **SPARQL Abstract Query** is a tuple (E, DS, R) where:
 - E is a SPARQL algebra expression
 - DS is an RDF dataset – $DS = \{ G, (<u_1>, G_1), \dots, (<u_n>, G_n) \}$
 - R is a query form (`select`, `ask`, `construct`, `describe`)
- Converting an Abstract Syntax Tree to an algebra expression:
 - Step 1: Expand abbreviations for URIs and triple patterns
 - Step 2: Recursively translate the query pattern
 - Step 3: Translate the solution modifiers

Converting to an Abstract Query

- Step 1: Expand abbreviations for URIs and triple patterns

Example:

```
?author rdf:type dbo:Writer ;  
        rdfs:label ?author_name .
```



```
?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<http://dbpedia.org/ontology/Writer> .  
  
?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name .
```

Converting to an Abstract Query

- Step 2: Recursively translate the query pattern

Example:

```
SELECT  ?author ?author_name ?work
FROM <http://dbpedia.org>
WHERE
{
  ?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Writer> .
  ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name
  FILTER ( lang(?author_name) = "en" ) .
  OPTIONAL {?author <http://dbpedia.org/ontology/notableWork> ?work .}
}
UNION
{
  ?work <http://dbpedia.org/ontology/author> ?author .
}
```

Converting to an Abstract Query

- Step 2: Recursively translate the query pattern

Example:

```
{  
  BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Writer> .  
    ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name .)  
  FILTER ( lang(?author_name) = "en" )  
  OPTIONAL {BGP(?author <http://dbpedia.org/ontology/notableWork> ?work) .}  
}  
UNION  
{  
  BGP(?work <http://dbpedia.org/ontology/author> ?author .)  
}
```

replace triple pattern with operator **BGP** ()

Converting to an Abstract Query

- Step 2: Recursively translate the query pattern

Example:

```
{
  LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
               <http://dbpedia.org/ontology/Writer> .
               ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
           BGP(?author <http://dbpedia.org/ontology/notableWork> ?work))
  FILTER ( lang(?author_name) = "en" )
}
UNION
{
  BGP(?work <http://dbpedia.org/ontology/author> ?author)
}
```

introduce `LeftJoin()` for optional parts

Converting to an Abstract Query

- Step 2: Recursively translate the query pattern

Example:

```
Union(  
  LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
    <http://dbpedia.org/ontology/Writer> .  
    ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),  
    BGP(?author <http://dbpedia.org/ontology/notableWork> ?work))  
  FILTER ( lang(?author_name) = "en" ) ,  
  BGP(?work <http://dbpedia.org/ontology/author> ?author)  
)
```

combine alternative graph patterns with `Union()`

Converting to an Abstract Query

- Step 2: Recursively translate the query pattern

Example:

```
Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)
```

Translate a group with filters with the `Filter` operator

Online translation tool: <http://sparql.org/query-validator.html>

Evaluation of SPARQL Algebra

- Semantics of the SPARQL Algebra Operations

$\text{Bgp}(P)$	match/evaluate pattern P
$\text{Join}(M_1, M_2)$	conjunctive join of solutions M_1 and M_2
$\text{Union}(M_1, M_2)$	union of solutions M_1 and M_2
$\text{LeftJoin}(M_1, M_2, F)$	optional join of M_1 and M_2 with filter constraint F (true if no filter is given)
$\text{Filter}(F, M)$	filter solutions M with constraint F
Z	empty pattern (identity for join)

- Evaluating a SPARQL graph pattern results in a **solution sequence** that lists possible bindings (mappings) of query variables to RDF terms in the active graph
- SPARQL algebra operators operate on **multisets** Ω (or sequences) of solution mappings

Basic Graph Pattern Matching

- **Definition:** A **solution mapping** μ is a partial function $\mu: V \rightarrow T$
- **Definition:** A **pattern instance mapping** P is the combination of an RDF instance mapping σ and solution mapping, μ . $P(x) = \mu(\sigma(x))$
- **Definition:** Let **BGP** be a basic graph pattern and let **G** be an RDF graph. μ is a solution for BGP from G when there is a pattern instance mapping P such that
 - $P(\text{BGP})$ is a subgraph of **G** and
 - μ is the restriction of P to the query variables in BGP.

Evaluation of a BGP P over a graph G results in a **multiset Ω of solution mappings μ** , where each μ is a solution for the BGP P

Basic Graph Pattern Matching

- Let $\text{card}[\Omega](\mu)$ denote the **cardinality of each solution mapping** μ in the multiset Ω
- **Definition:** Let **BGP** be a basic graph pattern and let **G** be an RDF graph.
 $\text{card}[\Omega](\mu)$ is the number of distinct RDF instance mappings, σ , such that
 - $P(x) = \mu(\sigma(x))$ is a pattern instance mapping and
 - $P(\text{BGP})$ is a subgraph of **G**.

Evaluation of Union

- $\text{Union}(\Omega_1, \Omega_2) = \{ \mu \mid \mu \text{ is in } \Omega_1 \text{ or } \mu \text{ is in } \Omega_2 \}$
- $\text{card}[\text{Union}(\Omega_1, \Omega_2)](\mu) = \text{card}[\Omega_1](\mu) + \text{card}[\Omega_2](\mu)$

 Ω_1

?author	?work
:Jules_Verne	:From_the_Earth_to_the_Moon
:Jules_Verne	:In_80_Days_Around_the_World

 Ω_2

?work	?author
:Fiesta	:Ernest_Hemingway
:Animal_Farm	:George_Orwell



$\text{Union}(\Omega_1, \Omega_2)$

?author	?work
:Jules_Verne	:From_the_Earth_to_the_Moon
:Jules_Verne	:In_80_Days_Around_the_World
:Ernest_Hemingway	:The_Sun_also_Rises
:George_Orwell	:Animal_Farm

Evaluation of Filter

- **Filter(e,Ω)** = {μ | μ is in Ω and e(μ) evaluates to true}
- card[Filter(e,Ω)](μ) = card[Ω](μ)

Ω

?author	?author_name
:Jules_Verne	"Jules_Verne"@en
:Jules_Verne	"Верн, Жюль"@ru
:Jules_Verne	"儒勒·凡尔纳"@zh



Ω

?author	?author_name
:Jules_Verne	"Jules_Verne"@en

```
FILTER(lang(?author_name)="en" )
```

Evaluation of Join

- $\text{Join}(\Omega_1, \Omega_2) = \{\text{merge}(\mu_1, \mu_2) \mid \mu_1 \text{ is in } \Omega_1 \text{ and } \mu_2 \text{ is in } \Omega_2 \text{ and } \mu_1 \text{ and } \mu_2 \text{ are compatible}\}$
- **Definition:** Two solution mappings μ_1 and μ_2 are **compatible**, iff $\mu_1(v) = \mu_2(v)$ holds for every variable v in $\text{dom}(\mu_1) \cap \text{dom}(\mu_2)$.
- **Definition:** Let μ_1 and μ_2 be compatible solution mappings, then $\text{merge}(\mu_1, \mu_2) = \mu_1 \cup \mu_2$
- Example: $\mu_1 = \{?x \rightarrow :A, ?y \rightarrow \text{"abc"}\}$, $\mu_2 = \{?x \rightarrow :A, ?z \rightarrow :B\}$
 $\text{merge}(\mu_1, \mu_2) = \{?x \rightarrow :A, ?y \rightarrow \text{"abc"}, ?z \rightarrow :B\}$

Evaluation of Join

- $\text{Join}(\Omega_1, \Omega_2) = \{\text{merge}(\mu_1, \mu_2) \mid \mu_1 \text{ is in } \Omega_1 \text{ and } \mu_2 \text{ is in } \Omega_2 \text{ and } \mu_1 \text{ and } \mu_2 \text{ are compatible}\}$
- $\text{card}[\text{Join}(\Omega_1, \Omega_2)](\mu) = \sum \text{card}[\Omega_1](\mu_1) \cdot \text{card}[\Omega_2](\mu_2)$
where μ_1 is in Ω_1 and μ_2 is in Ω_2 and μ_1 and μ_2 are compatible

?author	?work
:Jules_Verne	:From_the_Earth_to_the_Moon
:Jules_Verne	:In_80_Days_Around_the_World

Join

?author	?work	?releaseDate
:Jules_Verne	:From_the_Earth_to_the_Moon	1865
:Ernest_Hemingway	:The_Sun_also_Rises	1926

?author	?work	?releaseDate
:Jules_Verne	:From_the_Earth_to_the_Moon	1865

Evaluation of Diff

- $\text{Diff}(\Omega_1, \Omega_2, e) = \{\mu \mid \mu \text{ is in } \Omega_1 \text{ such that for all } \mu' \text{ in } \Omega_2$
either μ and μ' are NOT compatible
or $e(\text{merge}(\mu, \mu'))$ evaluates to false}
- $\text{card}[\text{Diff}(\Omega_1, \Omega_2, e)](\mu) = \text{card}[\Omega_1](\mu)$

?author	?work
:Jules_Verne	:From_the_Earth_to_the_Moon
:Jules_Verne	:In_80_Days_Around_the_World

?author	?work	?releaseDate
:Jules_Verne	:From_the_Earth_to_the_Moon	1865
:Ernest_Hemingway	:The_Sun_also_Rises	1926

Diff (?work = :From_the_Earth_to_the_Moon)



?author	?work
:Jules_Verne	:In_80_Days_Around_the_World

Evaluation of LeftJoin

- $\text{LeftJoin}(\Omega_1, \Omega_2, e) = \text{Filter}(e, \text{Join}(\Omega_1, \Omega_2)) \cup \text{Diff}(\Omega_1, \Omega_2, e)$
- $\text{card}[\text{LeftJoin}(\Omega_1, \Omega_2, e)](\mu) = \text{card}[\text{Filter}(e, \text{Join}(\Omega_1, \Omega_2))](\mu) + \text{card}[\text{Diff}(\Omega_1, \Omega_2, e)](\mu)$

?author	?work
:Jules_Verne	:From_the_Earth_to_the_Moon
:Jules_Verne	:In_80_Days_Around_the_World

?author	?work	?releaseDate
:Jules_Verne	:From_the_Earth_to_the_Moon	1865
:Ernest_Hemingway	:The_Sun_also_Rises	1926

LeftJoin

?author	?work	?releaseDate
:Jules_Verne	:From_the_Earth_to_the_Moon	1865
:Jules_Verne	In_80_Days_Around_the_World	

Example Evaluation

```
@prefix :      <http://dbpedia.org/resource/> .
@prefix dbo:   <http://dbpedia.org/ontology> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .

:From_the_Earth_To_the_Moon dbo:author :Jules_Verne .
:In_80_Days_around_the_World dbo:author :Jules_Verne .
:The_Sun_also_Rises dbo:author :Ernest_Hemingway.

:Jules_Verne rdf:type dbo:Writer ;
              rdfs:label "Jules_Verne"@en , "Верн, Жюль"@ru ;
              dbo:notableWork :The_Mysterious_Island .
:Ernest_Hemingway rdf:type dbo:Writer ;
                  rdfs:label "Ernest Hemingway"@en , "Хемингуэй, Эрнест"@ru
:George_Orwell dbo:notableWork :Animal_Farm .
```

Example Evaluation

```
Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)
```

?author	?work
:Jules_Verne	:Mysterious_Island
:George_Orwell	:Animal_Farm

Example Evaluation

```
Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)
```

?author	?work
:Jules_Verne	:Mysterious_Island
:George_Orwell	:Animal_Farm

?author	?author_name
:Jules_Verne	"Jules Verne"@en
:Jules_Verne	"Верн, Жюль"@ru
:Ernest_Hemingway	"Ernest Hemingway"@en
:Ernest_Hemingway	"Хемингуэй, Эрнест"@ru

Example Evaluation

```
Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)
```

?author	?author_name	?work
:Jules_Verne	"Jules Verne"@en	:Mysterious_Island
:Jules_Verne	"Верн, Жюль"@ru	:Mysterious_Island
:Ernest_Hemingway	"Ernest Hemingway"@en	
:Ernest_Hemingway	"Хемингуэй, Эрнест"@ru	

Example Evaluation

```
Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)
```

?author	?author_name	?work
:Jules_Verne	"Jules Verne"@en	:Mysterious_Island
:Jules_Verne	"Верн, Жюль"@ru	:Mysterious_Island
:Ernest_Hemingway	"Ernest Hemingway"@en	
:Ernest_Hemingway	"Хемингуэй, Эрнест"@ru	

?work	?author
:From_the_Earth_To_the_Moon	:Jules_Verne
:In_80_Days_aroundf_the_World	:Jules_Verne
:The_Sun_also_Rises	:Ernest_Hemingway

Example Evaluation

```

Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)

```

?author	?author_name	?work
:Jules_Verne	"Jules Verne"@en	:Mysterious_Island
:Jules_Verne	"Верн, Жюль"@ru	:Mysterious_Island
:Ernest_Hemingway	"Ernest Hemingway"@en	
:Ernest_Hemingway	"Хемингуэй, Эрнест"@ru	
:Jules_Verne		:From_the_Earth_To_the_Moon
:Jules_Verne		:In_80_Days_around_the_World
:Ernest_Hemingway		:The_Sun_also_Rises

Example Evaluation

```
Filter((lang(?author_name)="en"),
  Union(
    LeftJoin(BGP(?author <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
      <http://dbpedia.org/ontology/Writer> .
      ?author <http://www.w3.org/2000/01/rdf-schema#label> ?author_name),
      BGP(?author <http://dbpedia.org/ontology/notableWork> ?work)),
    BGP(?work <http://dbpedia.org/ontology/author> ?author)
  )
)
```

?author	?author_name	?work
:Jules_Verne	"Jules Verne"@en	:Mysterious_Island
:Ernest_Hemingway	"Ernest Hemingway"@en	
:Jules_Verne		:From_the_Earth_To_the_Moon
:Jules_Verne		:In_80_Days_around_the_World
:Ernest_Hemingway		:The_Sun_also_Rises

References

- For a detailed reference on SPARQL Algebra and SPARQL semantics:
 - Steve Harris, Andy Seaborne:
SPARQL 1.1 Query Language, W3C Recommendation 21 March 2013
<http://www.w3.org/TR/sparql11-query/#sparqlDefinition>



EXTRA: 11 - RDF on the Web

OpenHPI - Course Knowledge Engineering with Semantic Web Technologies

Lecture 2: RDF Based Knowledge Representation