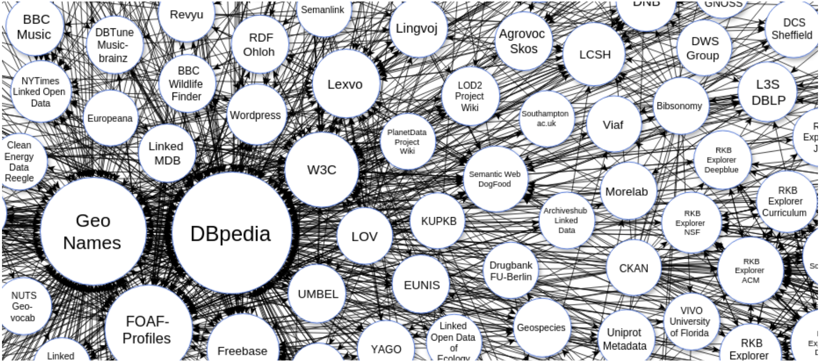
ABSTATInf: Inference of Knowledge Patterns for Linked Data Sets Summarization

Riccardo Porrini

DISCo, University of Milano-Bicocca
riccardo.porrini@disco.unimib.it

Assigment for the Course: The Impact of Logic: from Proof Systems to Databases

Linked Data Set Understanding



What types of resources are there in a data set? How are they described?

What properties are used to link different types of resources? How frequently?

Are data described as prescribed by the ontology?

ABSTAT Linked Data set summarization framework [1]

- compact and concise representation of a data set (i.e. summary)
- formal modeling of minimal type patterns (C, P, D) extracted from RDF data
- assertions $< a, P, b >$ where C min. type of a and D min. type of b
- minimalization based on a type graph that represent the ontology (schema)
- **occurrence** statistics computed for minimal type patterns

[1] M. Palmonari, A. Rula, R. Porrini, A. Maurino, B. Spahiu and V. Ferme et al. ASBTAT: Linked Data Summaries with ABstraction and STATistics. *ESWC Posters and Demos*, 2015

ABSTAT Linked Data set summarization framework [1]

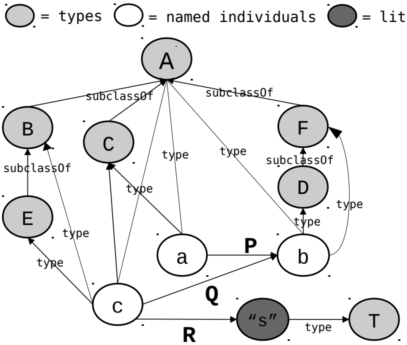
- compact and concise representation of a data set (i.e. summary)
- formal modeling of minimal type patterns (C, P, D) extracted from RDF data
- assertions $< a, P, b >$ where C min. type of a and D min. type of b
- minimalization based on a type graph that represent the ontology (schema)
- **occurrence** statistics computed for minimal type patterns

ABSTATInf

- implementation the ABSTAT summarization model in logic programming
- adds pattern inference from the minimal type patterns
- and the computation of the respective occurrence statistics in the data set

[1] M. Palmonari, A. Rula, R. Porrini, A. Maurino, B. Spahiu and V. Ferme et al. ASBTAT: Linked Data Summaries with ABstraction and STATistics. *ESWC Posters and Demos*, 2015

Linked Data Set



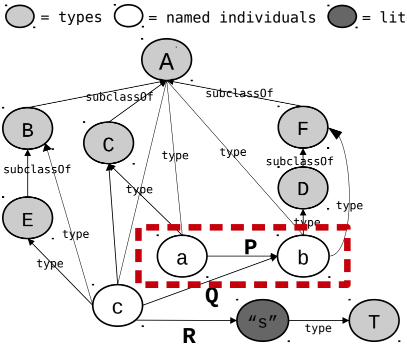
Patterns

(E, Q, F) (A, Q, D)
(C, P, A) (C, Q, A) (B, Q, D)
(C, P, F) (C, Q, F) (B, Q, A)
(A, P, A) (E, Q, D) (B, Q, F)
(A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D) (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
(C, R, T)

Minimal Type Pattern Base

Relational Assertions: $P(x, y)$ from the triple $< x, P, y >$



Patterns

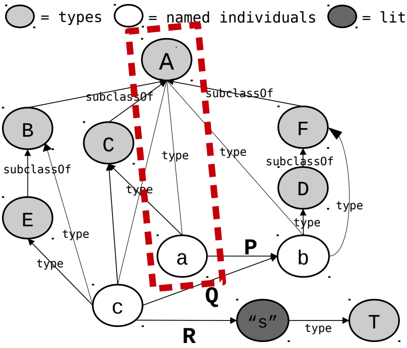
```
                (E, Q, F) (A, Q, D)
      (C, P, A) (C, Q, A) (B, Q, D)
      (C, P, F) (C, Q, F) (B, Q, A)
      (A, P, A) (E, Q, D) (B, Q, F)
      (A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
      (A, P, D) (E, Q, A) (A, Q, F) (A, R, T)
```

```
(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
                              (C, R, T)
```

Minimal Type Pattern Base

ABSTAT Model

Typing Assertions: $C(x)$ from the triple $< x, \mathtt{rdf:type}, C >$



Patterns

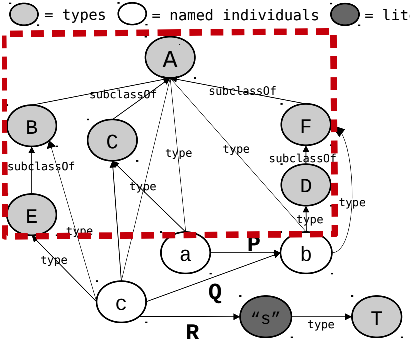
(E, Q, F) (A, Q, D)
(C, P, A) (C, Q, A) (B, Q, D)
(C, P, F) (C, Q, F) (B, Q, A)
(A, P, A) (E, Q, D) (B, Q, F)
(A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D) (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
(C, R, T)

Minimal Type Pattern Base

ABSTAT Model

Type Graph: $G = (\mathrm{N}, \preceq^G)$, $\preceq^G$ defines a partial order over N



Patterns

(E, Q, F) (A, Q, D)
(C, P, A) (C, Q, A) (B, Q, D)
(C, P, F) (C, Q, F) (B, Q, A)
(A, P, A) (E, Q, D) (B, Q, F)
(A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D) (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
(C, R, T)

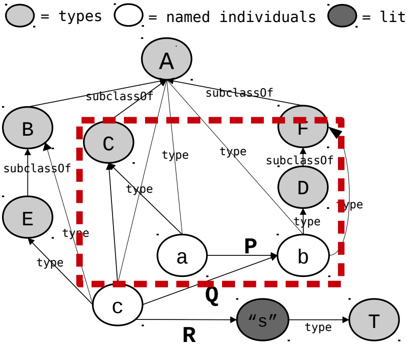
Minimal Type Pattern Base

ABSTAT Model

Abstract Knowledge Patterns and their Occurrence

(C, P, D) s.t. $\exists x \exists y (C(x) \wedge D(y) \wedge P(x, y))$ - **existential** definition

(C, P, D) **occurs** iff ...



Patterns

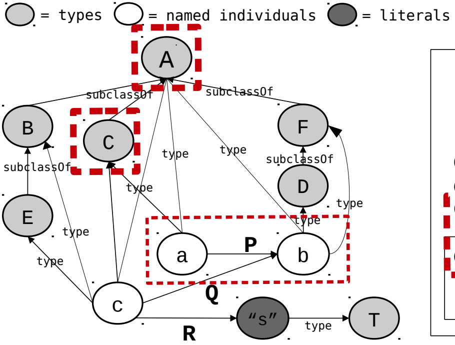
(E, Q, F) (A, Q, D)
(C, P, A) (C, Q, A) (B, Q, D)
(C, P, F) (C, Q, F) (B, Q, A)
(A, P, A) (E, Q, D) (B, Q, F)
(A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D) (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
(C, R, T)

Minimal Type Pattern Base

ABSTAT Model

Subpatterns: $(C, P, D) \preceq^G (C', P, D')$, iff $C \preceq^G C'$ and $D \preceq^G D'$



Patterns

(E, Q, F) (A, Q, D)
(C, P, A) (C, Q, A) (B, Q, D)
(C, P, F) (C, Q, F) (B, Q, A)
(A, P, A) (E, Q, D) (B, Q, F)
(A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D) (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
(C, R, T)

Minimal Type Pattern Base

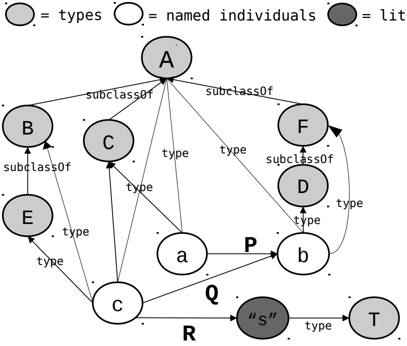
ABSTAT Model

Minimal Type Pattern Base

(C, P, D) occurs as $P(a, b)$

not exist C' s.t. $C'(a)$ and $C' \prec^G C$ or a D' s.t $D'(b)$ and $D' \prec^G D$

ensures **compactness**



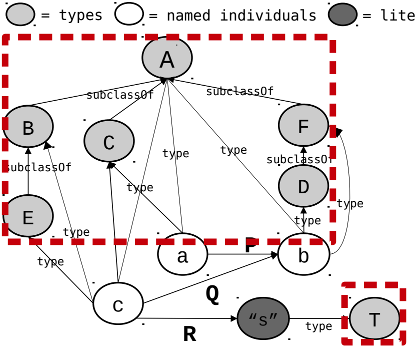
Patterns

```
           (E, Q, F) (A, Q, D)
(C, P, A)  (C, Q, A) (B, Q, D)
(C, P, F)  (C, Q, F) (B, Q, A)
(A, P, A)  (E, Q, D) (B, Q, F)
(A, P, F)  (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D)  (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D)  (E, Q, D) (C, Q, D) (E, R, T)
                               (C, R, T)
```

Minimal Type Pattern Base

Linked Data Set Summary



Patterns

(E, Q, F) (A, Q, D)
(C, P, A) (C, Q, A) (B, Q, D)
(C, P, F) (C, Q, F) (B, Q, A)
(A, P, A) (E, Q, D) (B, Q, F)
(A, P, F) (C, Q, D) (A, Q, A) (B, R, T)
(A, P, D) (E, Q, A) (A, Q, F) (A, R, T)

(C, P, D) (E, Q, D) (C, Q, D) (E, R, T)
(C, R, T)

Minimal Type Pattern Base

ABSTATInf

- implementation the ABSTAT summarization model in Prolog
- implements pattern inference from the minimal type patterns
- and the computation of the respective occurrence statistics in the data set

Formal Properties

of the minimal type pattern base

Relational Assertions Coverage

given (C, P, D), all instances of type C and D linked by the property P can be retrieved by applying the existential definition of patterns as a conjunctive query.

Formal Properties

of the minimal type pattern base

Relational Assertions Coverage

given (C, P, D), all instances of type C and D linked by the property P can be retrieved by applying the existential definition of patterns as a conjunctive query.

Completeness w.r.t. Pattern Inference

the minimal type pattern base includes all and only the patterns from which all other patterns can be inferred.

Pattern Inference and Occurrence Statistics

```
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, Supertype).
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, X),
        descendant(X, Supertype).

minimalType(Entity, Type) :-
        rdf(Entity, rdf:type, Type).

mPattern(C, P, D, Instance) :-
        rdf(Subject, P, Object),
        minimalType(Subject, C),
        minimalType(Object, D),
        Instance = {Subject, Object}.

iPattern(C, P, D, Instance):-
        descendants(C, ICs),
        descendants(D, IDs),
        mPattern(IC, P, ID, Instance),
        member(IC, ICs),
        member(ID, IDs).

occurrence(C, P, D, O) :-
        iPatterns(C, P, D, Patterns),
        length(Patterns, O).
```

Pattern Inference and Occurrence Statistics

```
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, Supertype).
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, X),
        descendant(X, Supertype).

minimalType(Entity, Type) :-
        rdf(Entity, rdf:type, Type).

mPattern(C, P, D, Instance) :-
        rdf(Subject, P, Object),
        minimalType(Subject, C),
        minimalType(Object, D),
        Instance = {Subject, Object}.

iPattern(C, P, D, Instance):-
        descendants(C, ICs),
        descendants(D, IDs),
        mPattern(IC, P, ID, Instance),
        member(IC, ICs),
        member(ID, IDs).

occurrence(C, P, D, O) :-
        iPatterns(C, P, D, Patterns),
        length(Patterns, O).
```

Instances of minimal type patterns are retrieved by applying their existential definition.
(via the coverage property)

Pattern Inference and Occurrence Statistics

```
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, Supertype).
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, X),
        descendant(X, Supertype).

minimalType(Entity, Type) :-
        rdf(Entity, rdf:type, Type).

mPattern(C, P, D, Instance) :-
        rdf(Subject, P, Object),
        minimalType(Subject, C),
        minimalType(Object, D),
        Instance = {Subject, Object}.

iPattern(C, P, D, Instance):-
        descendants(C, ICs),
        descendants(D, IDs),
        mPattern(IC, P, ID, Instance),
        member(IC, ICs),
        member(ID, IDs).

occurrence(C, P, D, O) :-
        iPatterns(C, P, D, Patterns),
        length(Patterns, O).
```

Instances of inferred patterns are aggregated from minimal type patterns.
(via the completeness w.r.t. pattern inference property)

Pattern Inference and Occurrence Statistics

```
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, Supertype).
descendant(Subtype, Supertype) :-
        rdf(Subtype, skos:broader, X),
        descendant(X, Supertype).

minimalType(Entity, Type) :-
        rdf(Entity, rdf:type, Type).

mPattern(C, P, D, Instance) :-
        rdf(Subject, P, Object),
        minimalType(Subject, C),
        minimalType(Object, D),
        Instance = {Subject, Object}.

iPattern(C, P, D, Instance):-
        descendants(C, ICs),
        descendants(D, IDs),
        mPattern(IC, P, ID, Instance),
        member(IC, ICs),
        member(ID, IDs).

occurrence(C, P, D, O) :-
        iPatterns(C, P, D, Patterns),
        length(Patterns, O).
```

Occurrece statistics are trivially computed

Demo

Conclusions

- ABSTATInf, a Prolog implementation of the ABSTAT model

Future work

Conclusions

- ABSTATInf, a Prolog implementation of the ABSTAT model
- pattern inference with occurrence statistics computation

Future work

Conclusions

- ABSTATInf, a Prolog implementation of the ABSTAT model
- pattern inference with occurrence statistics computation
- built on ABSTAT model's formal properties

Future work

Conclusions and Future Work

Conclusions

- ABSTATInf, a Prolog implementation of the ABSTAT model
- pattern inference with occurrence statistics computation
- built on ABSTAT model's formal properties

Future work

- Consider also subproperty relations

Conclusions and Future Work

Conclusions

- ABSTATInf, a Prolog implementation of the ABSTAT model
- pattern inference with occurrence statistics computation
- built on ABSTAT model's formal properties

Future work

- Consider also subproperty relations
- Experiments with large data sets

Questions?

riccardo.porrini@disco.unimib.it
http://rporrini.info