SPARQL: Eine RDF Anfragesprache

Daniel Beck
<Daniel.Beck AT dfki.de>

Seminar "Semantisches Web und Agenten"
WS 2006/2007

Übersicht

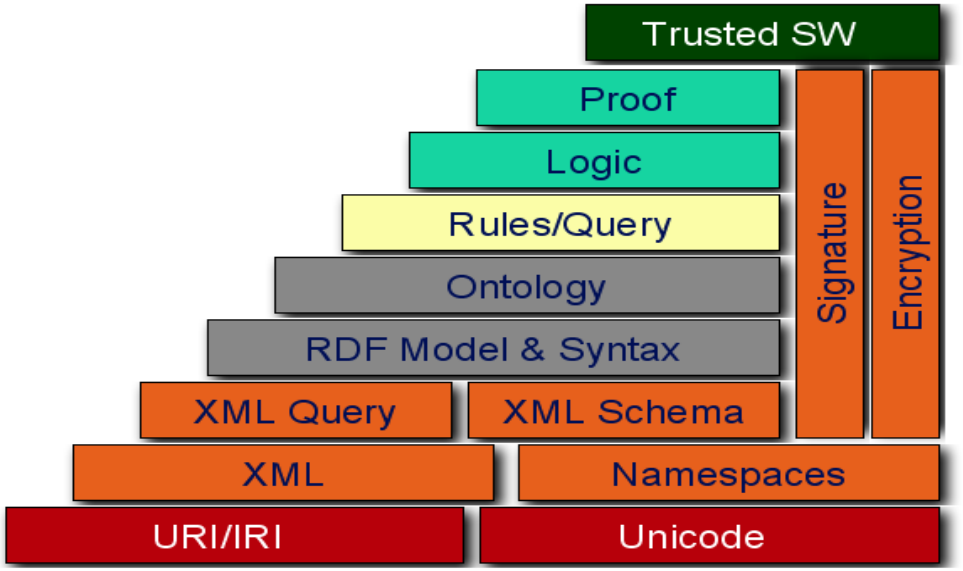
- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

Übersicht

- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

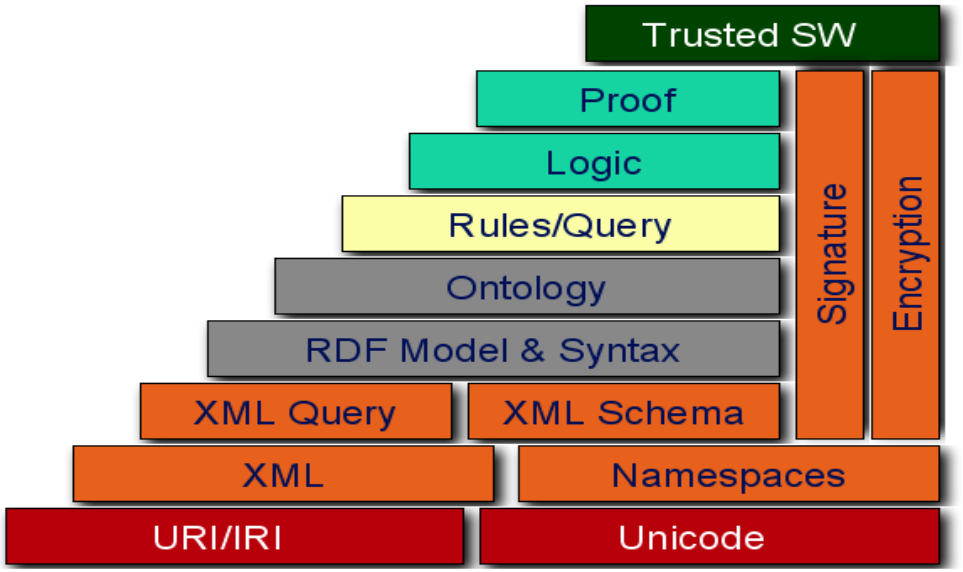
Semantic Web cake

- RDF
  - (Synktatische) Basis des Semantik Web
  - Allzweck-Sprache um Sachverhalte auszudrücken
- RDFS: semantische Erweiterung von RDF
  - Konzepte, Konzepthiearchie, Konzepteigenschaften, Wertebereiche, Instanzen



Semantic Web cake

- OWL:
  - RDF und RDFS sind Grundlage von OWL
  - Disjunkte Klassen, Kardinalität von Eigenschaften, Funktionale und invers-funktionale Eigenschaften

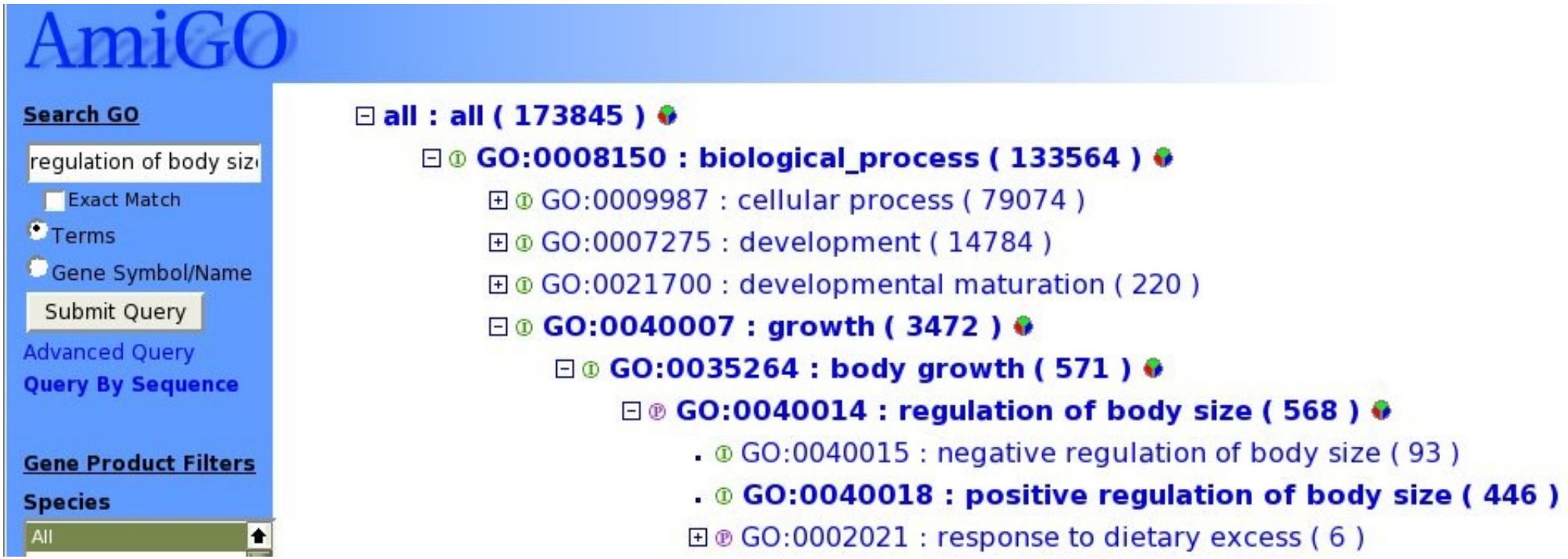


RDF-Anfrage Sprachen

- Es gibt schon viele große Ontologie die RDF/RDFS/OWL-* benutzen
  - LT-World
  - GO
  - Cyc
  - Foundation Model of Anatomy
  - General Ontology for Linguistic Description (GOLD)
  - Plant Ontology...
- Was fehlt: Suchen/Extrahieren von Daten aus eine Ontologie

RDF-Anfrage Sprachen: Bsp

- Extrahieren von Daten aus einer RDF Datenbank
  - Welche Gene sind für den Wachstum eines Menschen Verantwortlich?



RDF-Anfrage Sprachen: Bsp

- Erzeugen einer neuen Ontologie
  - Extrahieren alle Personen aus der „Mitarbeiter Ontologie“ einer Firma (Xtramind)
  - Extrahiere die auf Konferenzen publizierten Papers (LT-World ontologie)
  - Erzeuge daraus eine neue Ontologie
    - (s. Vortrag „OWL-Lite ontology matching „ von Martin)

RDF-Anfrage Sprachen

- Es existieren bereits RDF-Anfragesprachen
- Versa (4 Suite Library)
  - Keine „einfache“ Syntax
- SeRQL
  - Keine Möglichkeit die Quelle der Daten (Data Sources) zu definieren
- RDQL, RQL
  - Nicht sehr ausdrucksstark
- ...

SPARQL

- Spezifikation eines neuen Standard: SPARQL
  - Protokoll + RDF-Abfragesprache
  - Kombination aller gute Features aus alle RDF-Anfragesprachen
  - Produkt der RDF Data Access Working Group (DAWG) des W3C
    - Entwicklungsstufe „Candidate Recommendation“
  - SQL ähnliche Syntax (Aber trotzdem sehr verschieden!)

SPARQL

- Spezifikation eines neuen Standard: SPARQL
  - Protokoll + RDF-Abfragesprache
  - Kombination aller gute Features aus alle RDF-Anfragesprachen
  - Produkt der RDF Data Access Working Group (DAWG) des W3C
    - Entwicklungsstufe ~~„Candidate Recommendation“~~ Working Draft
  - SQL ähnliche Syntax (Aber trotzdem sehr verschieden!)

Übersicht

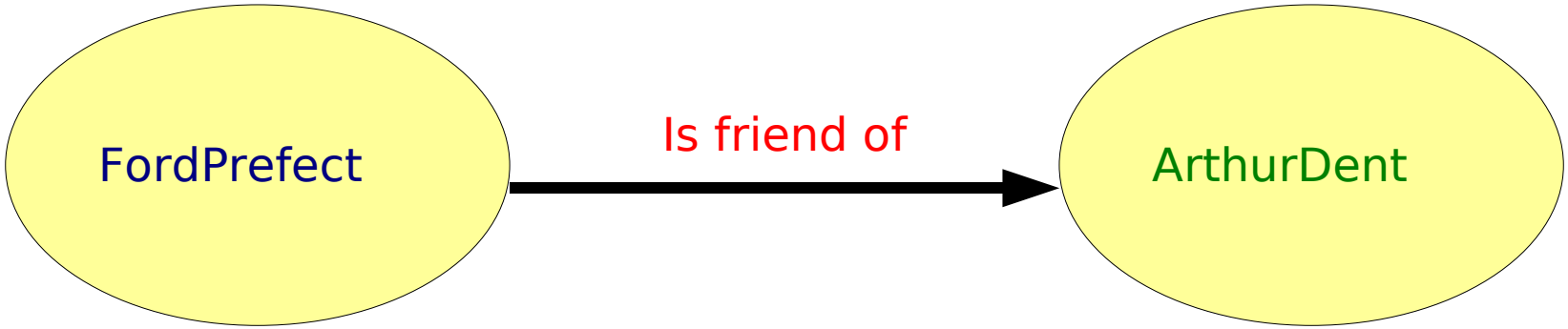
- Motivation
- Einleitung in RDF
  - Ressourcen und Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

URI und Ressourcen

- URIs identifizieren Ressourcen
- Alles was man einem Namen geben kann ist beschreibbar

Fakten in RDF ausdrücken

- Darstellung von Fakten als RDF-Tripel:
  - Prädikat: Relation
  - Subjekt, Objekt: „Sachen“
- Tripel: gelabelte Kante aus einem Graph
- Beispiel: (Ford Prefect, isFriendOf, Arthur Dent)



Fakten in RDF ausdrücken

- Darstellung von Fakten als RDF-Tripel:
  - Prädikat: Relation
  - Subjekt, Objekt: „Sachen“
- Tripel: gelabelte Kante aus einem Graph
- Beispiel: (Ford Prefect, isFriendOf, Arthur Dent)

Ressourcen werden durch URIs denotiert!

Fakten in RDF ausdrücken

- Subjekt: URI oder leer
  - Bsp: http://paddg.org/FordPrefect
- Prädikat: URI
  - Bsp: http://paddg.org/isFriendOf
- Objekt: URI, Literal oder leer

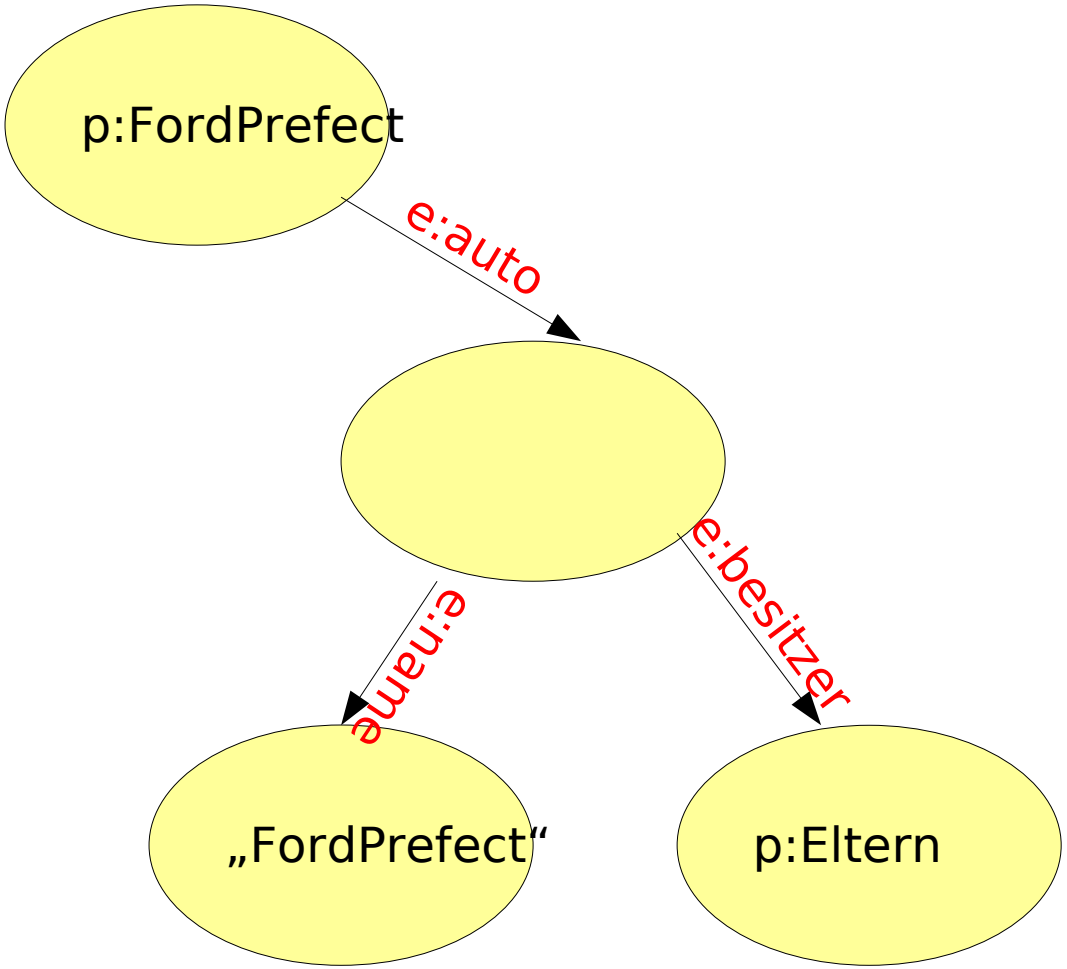
Fakten in RDF ausdrücken

- Subjekt: URI oder leer
  - Bsp: http://paddg.org/FordPrefect
- Prädikat: URI
  - Bsp: http://paddg.org/isFriendOf
- Objekt: URI, Literal oder leer

Nicht alle Knoten haben Namen!

Leere Knoten

- Für die Darstellung von n-stellige Relationen



Übersicht

- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

RDF-Graph

- RDF-Graph besteht aus Menge von Tripeln.
- Gerichteter gelabelter Graph
- Denkt immer an RDF als Graph, nicht als XML oder Dokument!

RDF-Graph

- RDF-Graph besteht aus Ressourcen

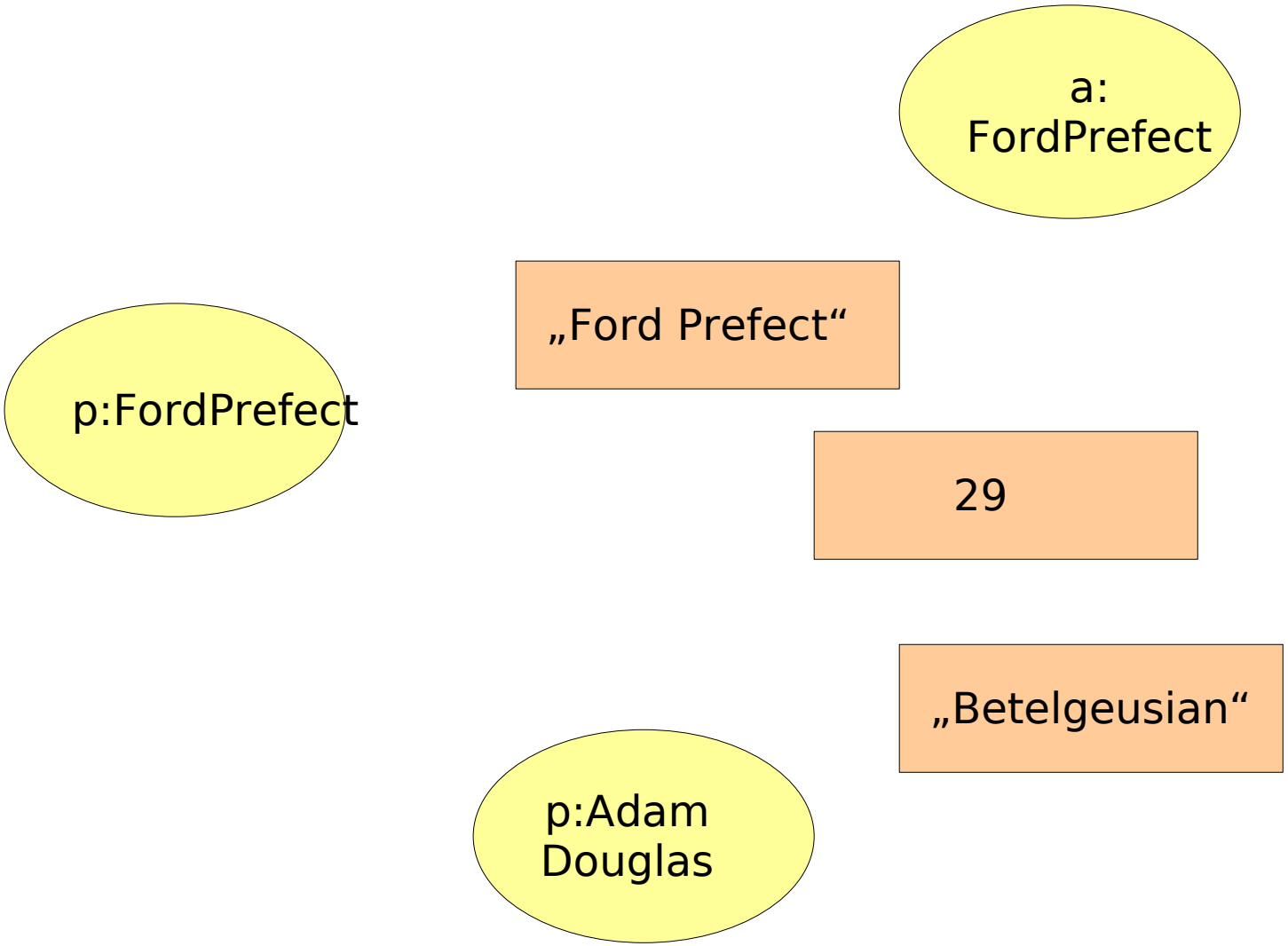
a:
FordPrefect

p:FordPrefect

p:Adam
Douglas

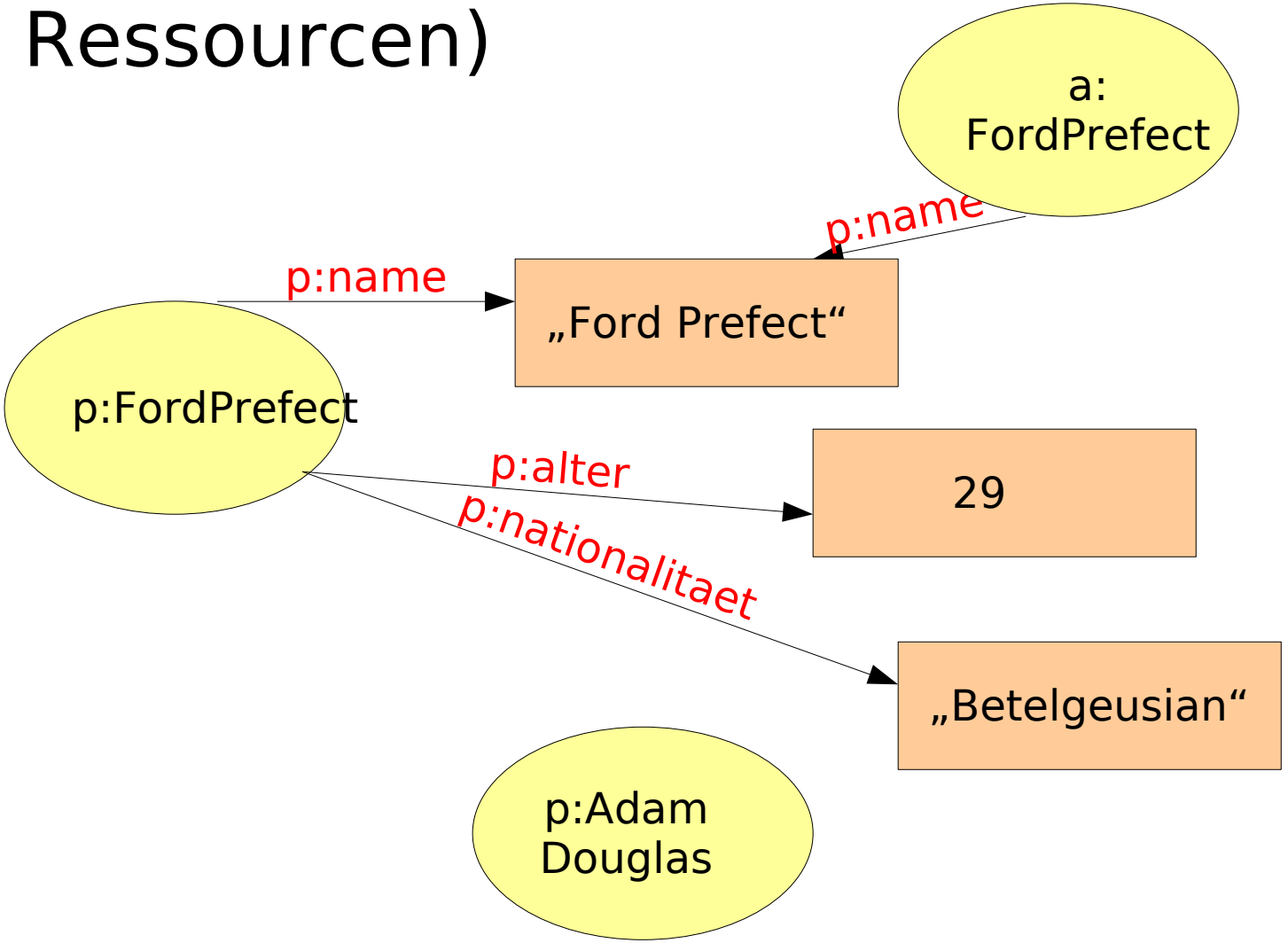
RDF-Graph

- ... und aus Literalen



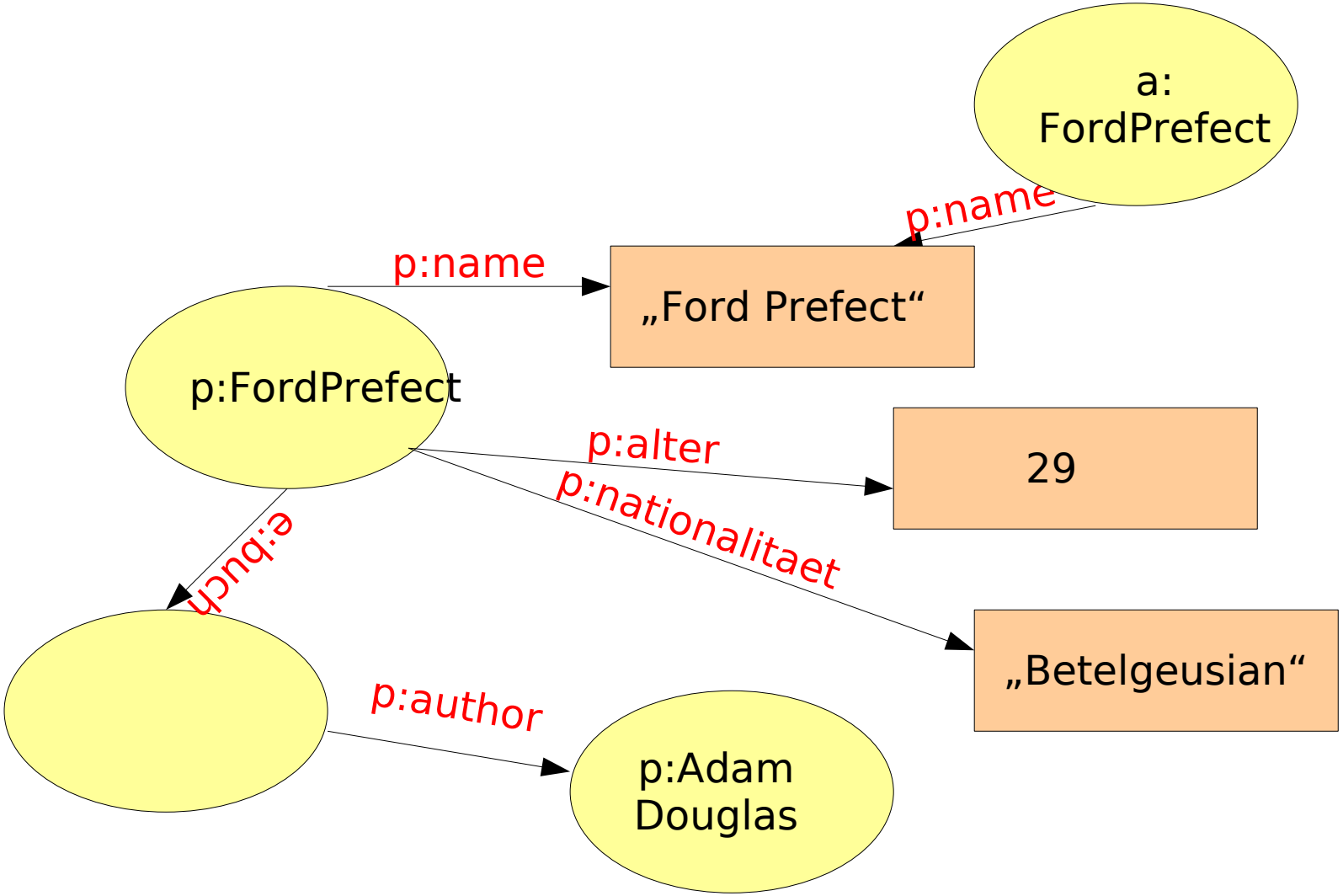
RDF-Graph

- ... aus benannten Relationen (Relationen sind Ressourcen)



RDF-Graph

- ... und aus leeren Knoten



RDF-Serialisation zur Beschreibung von Graphen

- RDF/XML: auf XML basierend

```
<rdf:Description
   rdf:about="http://www.paddg.org/FordPrefect">
<p:name>”Ford Prefect”</p:name>
<p:alter> 24</p:alter>
</rdf:Description>
<rdf:Description
   rdf:about="http://www.autos.org/FordPrefect">
<p:name>”Ford Prefect”</p:name>
</rdf:Description>
```

- N3: leicht von Menschen lesbar

```
@PREFIX p: <http://paddg.org>
@PREFIX a: <http://autos.org>

p:FordPrefect  p:name    „Ford Prefect“ .
a:FordPrefect  p:name    „Ford Prefect“ .
p:FordPrefect  p:freund  p:ArthurDent   .
p:ArthurDent   p:freund  p:FordPrefect  .
p:FordPrefect  p:buch    _:b            .
_:b            P:author  „A. Douglas“   .
```

Übersicht

- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

„OpenWorld assumption“

- „Alles was nicht im RDF-Graph ist, ist unbekannt“
- Tripel hinzufügen oder entfernen ändert nicht den Sinn des restlichen Graphen
- Es ist leicht zwei RDF-Graphen zusammenzufügen
  - nur leere Knoten müssen umbenannt werden

Logische Semantik von RDF; „was bedeutet ein RDF Graph?“

- RDF: „existential binary relational logic“
- Semantik RDF Tripel (Subj, Präd,Obj):
  - $L_{präd}$(Sub,Obj)
    - Wobei $L_{präd}$ = {<x,z> | ∀x.∀y. Triple(x,Präd,z) ist im RDF-Graph}

Logische Semantik von RDF; „was bedeutet ein RDF Graph?“

- Semantik RDF Tripel: (a:FordPrefect,a:name, „FordPerfect“)
  - L_{name}(a:FordPrefect,“FordPrefect“)
- Semantik RDF Tripel: (_:x,a:author, a:AdamDouglas)
  - $\exists x.L_{author}$(x,a:AdamDouglas)

Logische Semantik von RDF; „was bedeutet ein RDF Graph?“

- Semantik des Graphen G: Konjunktion der Semantik aller Tripel aus G

Übersicht

- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

SPARQL-Anfragesprache

- Anfrage ist ein Tupel (GP, DS, SM,R)
  - GraphPattern (GP)
  - Zu durchsuchende Daten (DS)
  - Solution Modifiers (SM)
  - Resultat Form (R)

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?person
FROM    <http://paddg.org>
WHERE
  { ?person p:name "FordPrefect" .
    ?person p:freund p:ArthurDent }
ORDER BY ?person
```

SPARQL Abfragesprache

- Anfrage ist ein Tupel (GP, DS, SM,R)
  - GraphPattern (GP)
  - Zu durchsuchende Daten (DS)
  - Solution Modifiers (SM)
  - Resultat Form (R)

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?person
FROM    <http://paddg.org>
WHERE
  { ?person p:name "FordPrefect" .
    ?person p:freund p:ArthurDent }
ORDER BY ?person
```

Suchsprachen (1)

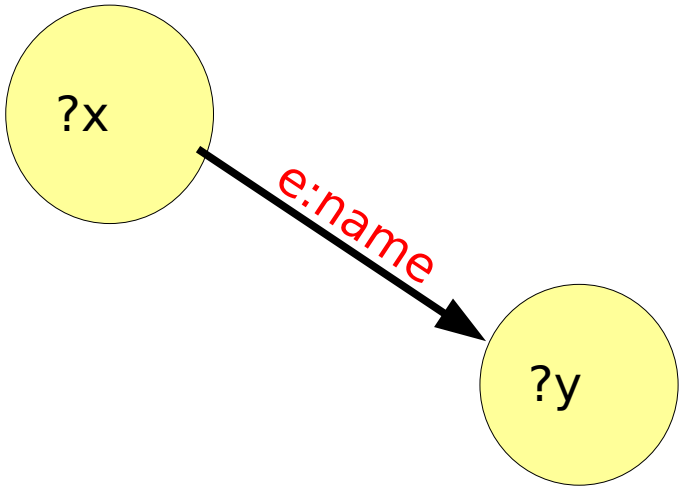
- Dokumentsuche (IR): Suche mittels Ähnlichkeitsmass (z.B. TF-IDF).
  - Daten: „Bag-of-words“-Vektoren aus gewichteten Wörtern
  - Suchbegriff: Wort
- SQL: extrahiert Werte aus Tabelle (mit Joins, Zeile...)
  - Daten: in Tabellen gespeichert
  - Suchbegriff: Name der Zeile, Name der Tabellen...

Suchsprachen (2)

- XQuery: Suche nach Teilbaum
  - Daten: XML – also Baum-Datenstrukturen
  - Suchbegriff: Pfad-Ausdrücke
    - doc("books.xml")/bookstore/book/title
- RDF-Anfragesprachen:
  - Daten: gespeichert als gerichtete Graphen
  - Suchbegriff: gerichtete Graphen mit Variablen-Knoten (Graph Patterns)

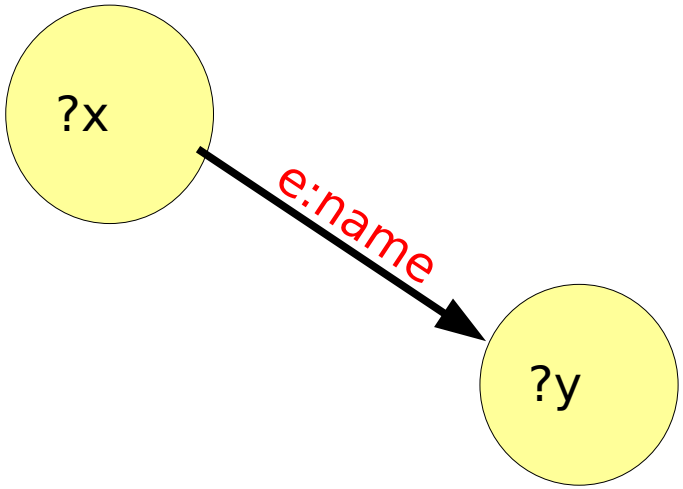
Matchen von Graph Pattern

Graph Pattern



Matchen von Graph Pattern

Graph Pattern

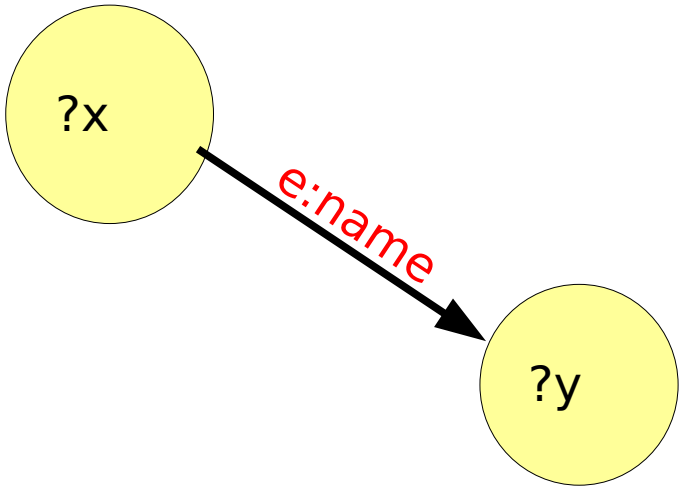


RDF-Pattern Syntax

```
?x e:name ?y.
```

Matchen von Graph Pattern

Graph Pattern



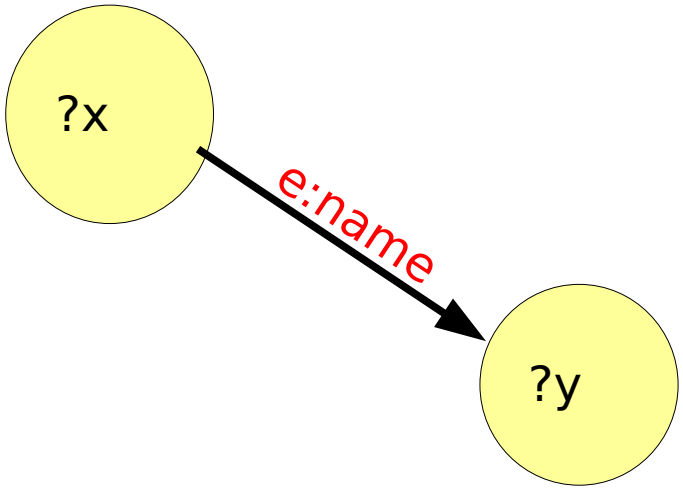
RDF-Pattern Syntax

```
?x e:name ?y.
```

Triple Patern Syntax:
Turtle+variablen

Matchen von Graph Pattern

Graph Pattern



RDF Graph

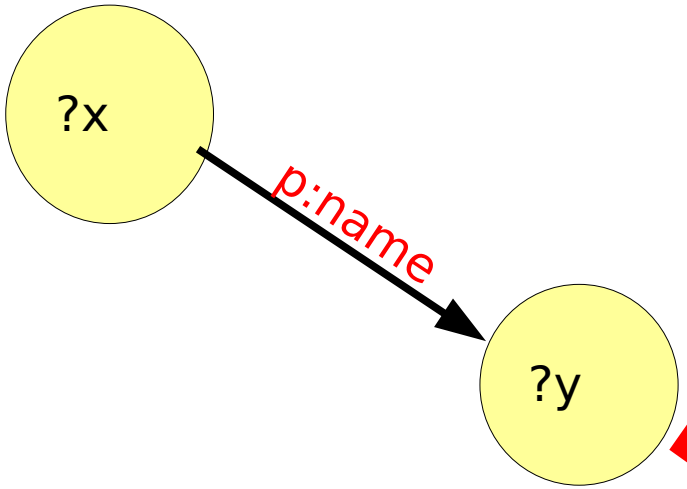
```
@PREFIX p: <http://paddg.org>
@PREFIX a: <http://autos.org>

p:FordPrefect  p:name    „Ford Prefect“ .
a:FordPrefect  p:name    „Ford Prefect“ .
p:FordPrefect  p:freund  p:ArthurDent   .
p:ArthurDent   p:freund  p:FordPrefect  .
p:FordPrefect  p:buch    _:b            .
_:b            p:author  „A. Douglas“   .
```

RDF-Pattern Syntax

```
?x e:name ?y.
```

Matchen von Graph Pattern

Graph Pattern



RDF Graph

```
@PREFIX p: <http://paddg.org>
@PREFIX a: <http://autos.org>

p:FordPrefect  p:name    „Ford Prefect“ .
a:FordPrefect  p:name    „Ford Prefect“ .
p:FordPrefect  p:freund  p:ArthurDent   .
p:ArthurDent   p:freund  p:FordPrefect  .
p:FordPrefect  p:buch    _:b            .
_:b            p:author  „A. Douglas“   .
```

RDF-Pattern Syntax

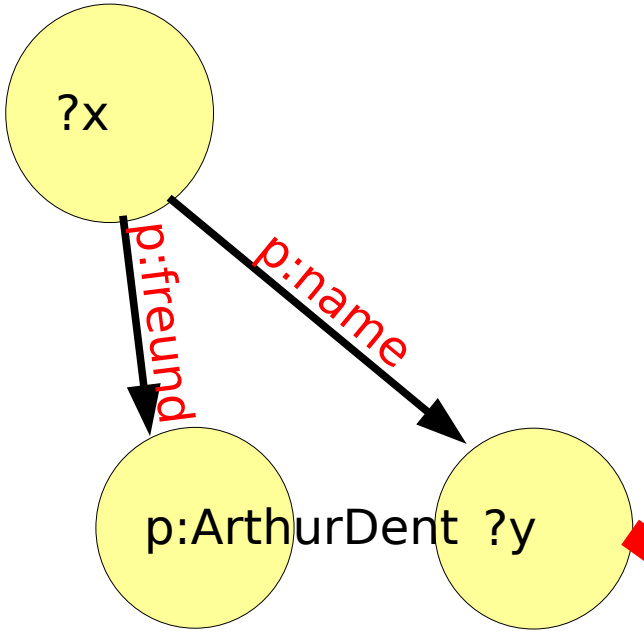
```
?x p:name ?y.
```

Resultate

```
p:FordPrefect p:name „FordPrefect“
a:FordPrefect p:name „FordPrefect“
```

SPARQL (2. Beispiel)

Graph Pattern



RDF Graph

```
@PREFIX p: <http://paddg.org>
@PREFIX a: <http://autos.org>

p:FordPrefect  p:name    „Ford Prefect“ .
a:FordPrefect  p:name    „Ford Prefect“ .
p:FordPrefect  p:freund  p:ArthurDent   .
p:ArthurDent   p:freund  p:FordPrefect  .
p:FordPrefect  p:buch    _:b            .
_:b            p:author  „A. Douglas“   .
```

RDF-Pattern Syntax

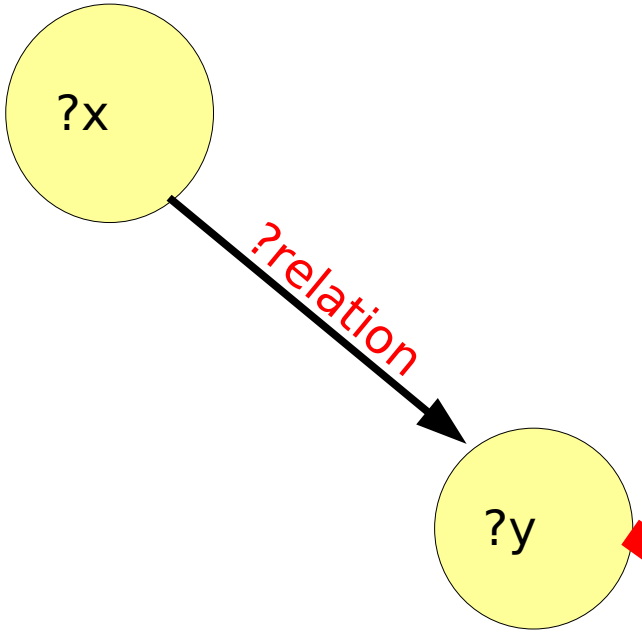
```
?x p:name ?y.
?x p:freund p:ArthurDent
```

Resultate

```
p:FordPrefect p:name „FordPrefect“
p:FordPrefect p:freund p:ArturDent
```

SPARQL (3. Beispiel)

Graph Pattern



RDF Graph

```
@PREFIX p: <http://paddg.org>
@PREFIX a: <http://autos.org>

p:FordPrefect  p:name    „Ford Prefect“ .
a:FordPrefect  p:name    „Ford Prefect“ .
p:FordPrefect  p:freund  p:ArthurDent   .
p:ArthurDent   p:freund  p:FordPrefect  .
p:FordPrefect  p:buch    _:b            .
_:b            p:author  „A. Douglas“   .
```

RDF-Pattern Syntax

```
?x ?relation ?y.
```

Resultate

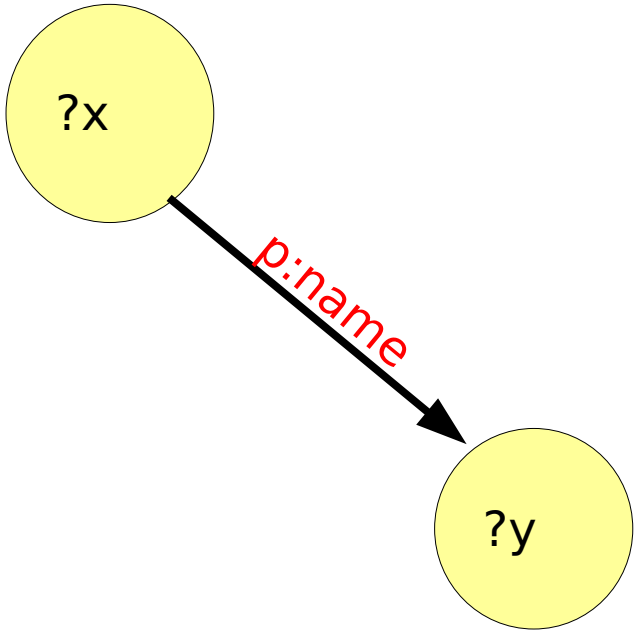
```
p:FordPrefect  p:name    „Ford Prefect“ .
a:FordPrefect  p:name    „Ford Prefect“ .
p:FordPrefect  p:freund  p:ArthurDent   .
p:ArthurDent   p:freund  p:FordPrefect  .
p:FordPrefect  p:buch    _:b            .
_:b            p:author  „A. Douglas“   .
```

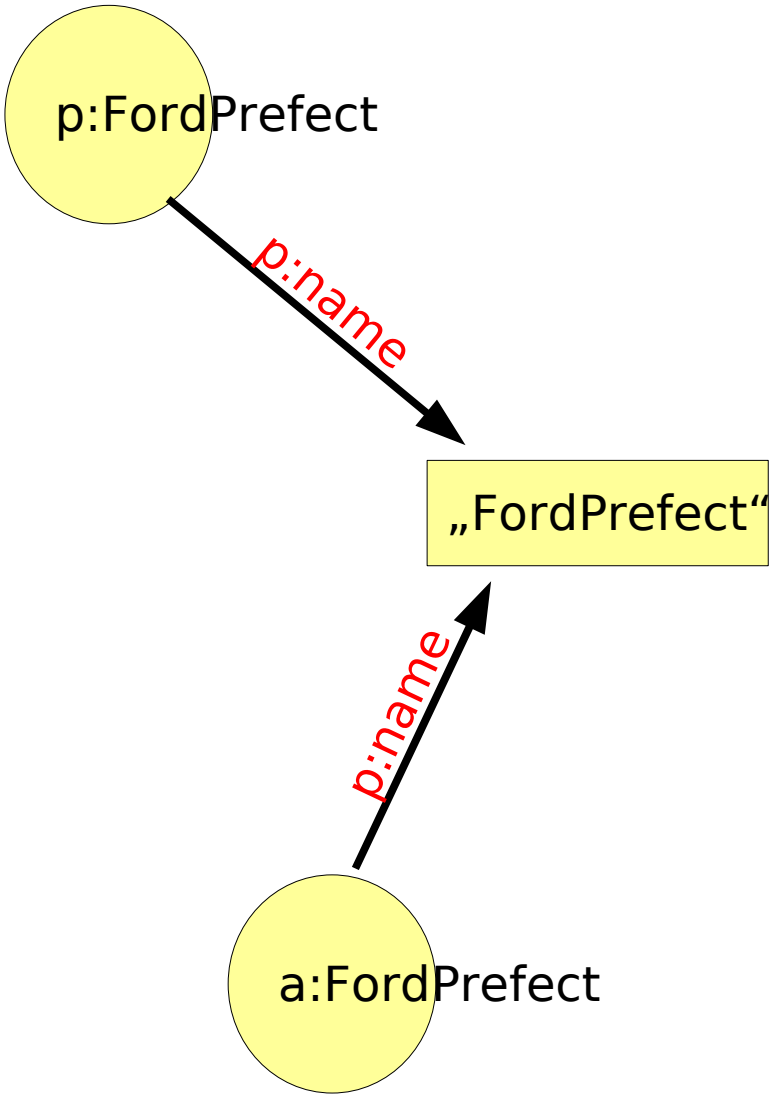
Wann matcht ein RDF Graph zu einer Menge von Graph Pattern?

- Es muss gelten:
  - Literale (Zahlen, Zeichenketten) werden auf gleiche Literale abgebildet
  - URI's werden auf gleiche URI's abgebildet
  - Richtung der Kanten stimmt überein
  - Graph Pattern Variable wird immer auf den selben Knoten abgebildet. (Geltungsbereich entspricht dem ganzen Pattern)

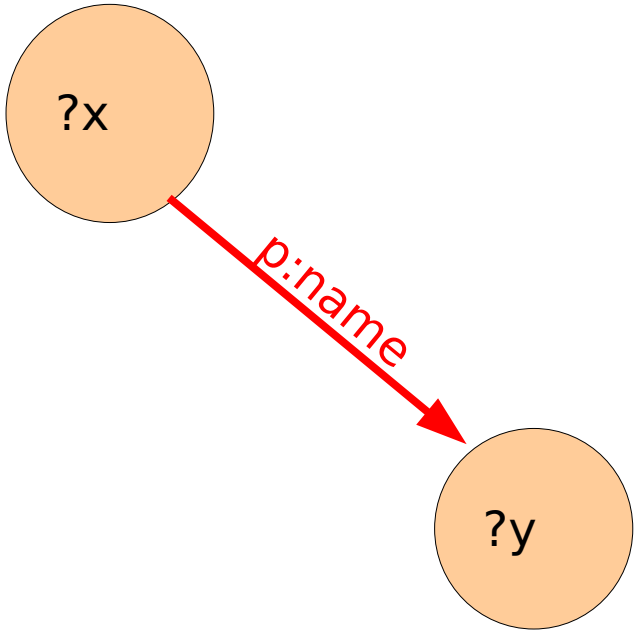
```
?x p:name ?y.
?x p:freund p:ArthurDent
```

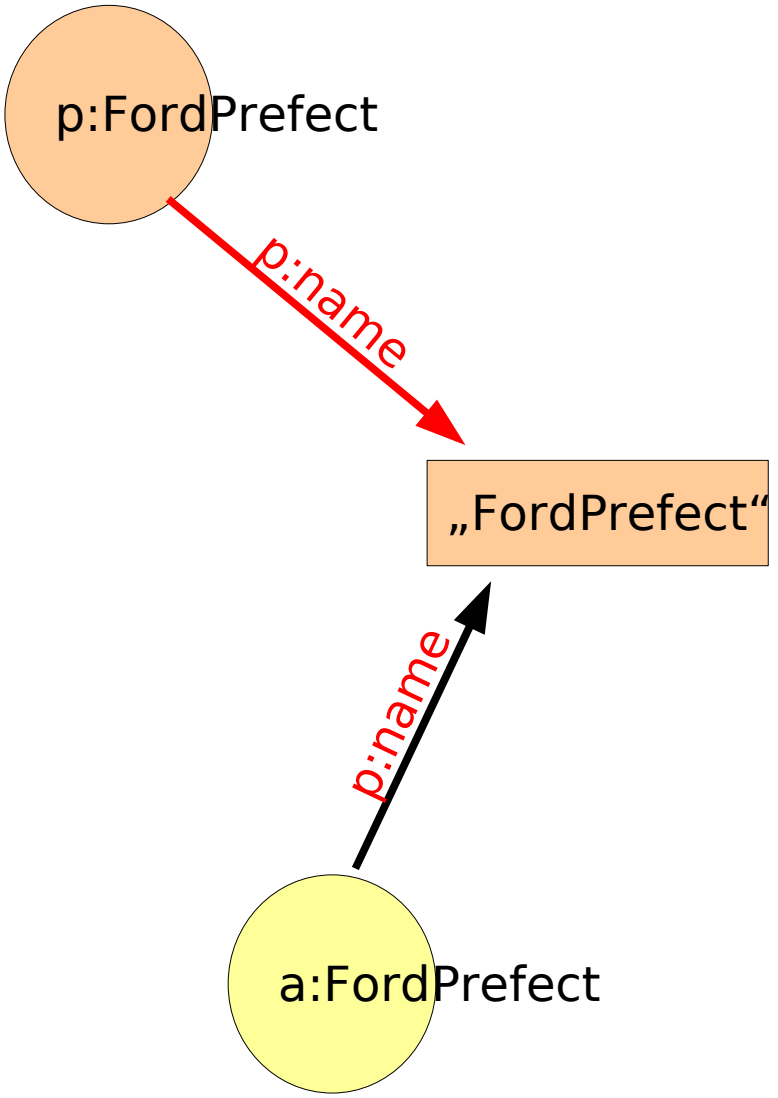
Wann matcht ein RDF Graph zu einer Menge von Graph Pattern?



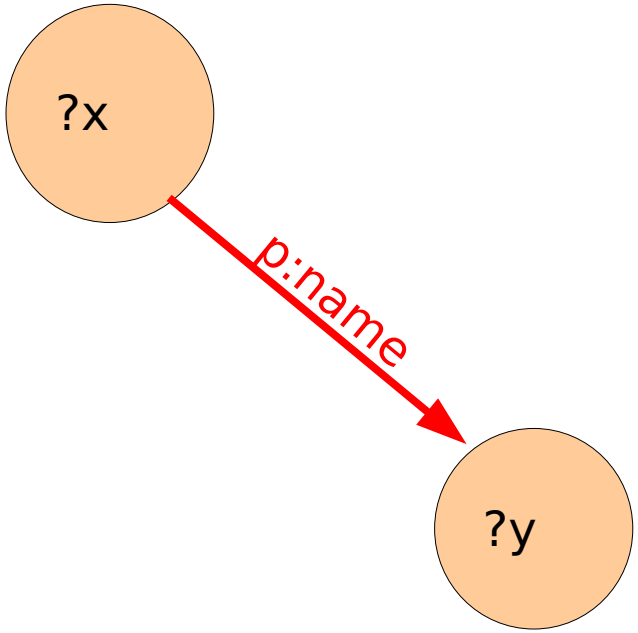


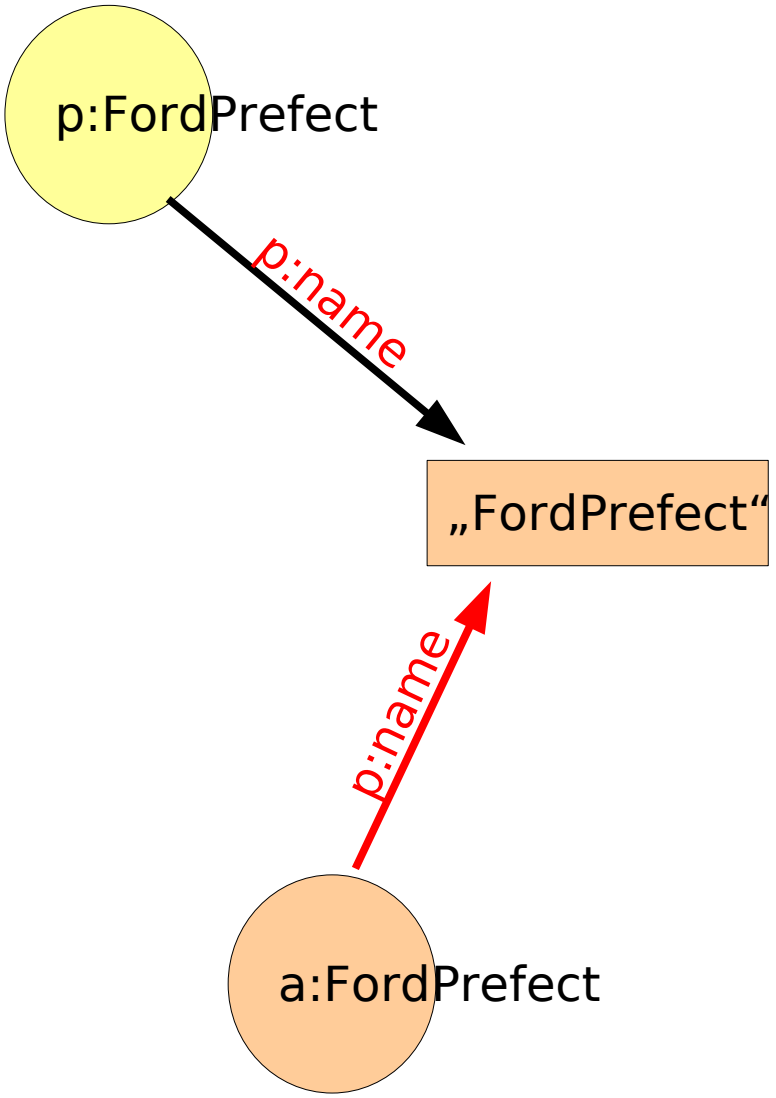
Wann matcht ein RDF Graph zu einer Menge von Graph Pattern?





Wann matcht ein RDF Graph zu einer Menge von Graph Pattern?





Noch mehr über Graph Patterns

- Möglichkeiten:
  - Einfache Graph Patterns (Wie bisher)
  - Konjunktionen
  - Disjunktionen
  - Schachtelungen
  - Optionale Graph Patterns
  - RDF Dataset Graph Pattern
  - ...

Optionale/geschaltete GP

- Optionale Graph Pattern

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?name ?mbox
WHERE
  { ?x foaf:name ?name .
    OPTIONAL { ?x foaf:mbox ?mbox}
  }
```

- Geschachtelte Graph Pattern

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?name ?mbox ?gName
WHERE
  { ?x foaf:mbox ?mbox .
    OPTIONAL { ?x foaf:mbox ?mbox.
               OPTIONAL { ?x foaf:givenName ?gName. }
    }}
```

Disjunktion GP

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?name ?x
WHERE { ?x foaf:name ?name .}
UNION { ?x foaf:nickname ?nname .}
```

Disjunktion GP

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?name ?x
WHERE { ?x foaf:name ?name .}
UNION { ?x foaf:nickname ?nname .}
```

Filter Graph Pattern

- Reguläre Ausdrücke

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?name ?x
WHERE { ?x foaf:name ?name
FILTER regex (?name „Ford“).
}
```

SPARQL Abfragesprache

- Anfrage ist ein Tupel (GP, DS, SM,R)
  - GraphPattern (GP)
  - Zu durchsuchende Daten (DS)
  - Solution Modifiers (SM)
  - Resultat Form (R)

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?person
FROM    <http://paddg.org>
WHERE
  { ?person p:name "FordPrefect" .
    ?person p:freund p:ArthurDent }
ORDER BY ?person
```

Beschränkung der Suche auf bestimmte Graphen

```
PREFIX p:   <http://paddg.org>
SELECT ?x ?name
FROM    <http://autos.org>
FROM    <http://paddg.org>
WHERE { ?x p:name ?name .}
```

x	name
p:FordPrefect	"Ford Prefect"
a:FordPrefect	"Ford Prefect"

Beschränkung der Suche auf bestimmte Graphen

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

SELECT ?email
WHERE
  {     GRAPH p {
        ?x foaf:name <mailto:FordPrefect@paddg.org> .
        ?x foaf:email ?email }
  }
```

src	email
<http://paddg.org>	"FordPrefect@paddg.org"

SPARQL Anfragesprache

- Anfrage ist ein Tupel (GP, DS, SM,R)
  - GraphPattern (GP)
  - Zu durchsuchende Daten (DS)
  - Solution Modifiers (SM)
  - Resultat Form (R)

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?person
FROM    <http://paddg.org>
WHERE
  { ?person p:name "FordPrefect" .
    ?person p:freund p:ArthurDent }
ORDER BY ?person
```

Solution Modifiers

- LIMIT:
  - Gib Resultate zurück wenn #Resultate < 2000
- OFFSET
  - Gib Resultate [20, Unendlich] zurück
- ORDERED BY
  - Ordne die Resultate nach dem „Alter“ (Literal)

SPARQL Anfragesprache

- Anfrage ist ein Tupel (GP, DS, SM,R)
  - GraphPattern (GP)
  - Zu durchsuchende Daten (DS)
  - Solution Modifiers (SM)
  - Resultat Form (R)

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1>
SELECT ?person
FROM    <http://paddg.org>
WHERE
  { ?person p:name "FordPrefect" .
    ?person p:freund p:ArthurDent }
ORDER BY ?person
```

Form der Ergebnisse

- Select
  - Liefert Liste von Variablenbelegungen (Belegung: RDF-T)

x	name
p:FordPrefect	"Ford Prefect"
a:FordPrefect	"Ford Prefect"

- Construct
  - Liefert einen RDF Graph

Form der Ergebnisse

- ASK
  - Anfrage: Tripel
  - Resultat: Anzahl der Ergebnisse>0
    - Gibt es eine Person im RDF Graphen deren Name „Ford Prefect“ ist?
- DESCRIBE
  - „Sag mir etwas nützliches über einen Teil des RDF-Graphen“
    - Z.b.: Informationen über den Server, über einen leeren Knoten, ...
    - Rückgabe ist von RDF DB abhängig

Übersicht

- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

Query Evaluation Problem

- Gegeben
  - RDF-Graph G
  - Graph Pattern P
  - Mapping M
- ist M Lösung der Evaluation von P auf G?

Komplexität der Query Evaluation?

- s. „Semantics and Complexity of SPARQL“
  Jorge Perez, Mercelo Arenas, Claudio Gutierrez (2006).
- Graph Pattern mit
  - AND und FILTER: $O(|P| \cdot |G|)$
  - AND, FILTER und UNION: NP vollständig
  - AND, FILTER, UNION und OPTIONAL: PSPACE vollständig

Übersicht

- Motivation
- Einleitung in RDF
  - Tripel
  - Graph
  - Semantik
- SPARQL
- Komplexität
  - Evaluierung einer SPARQL Anfrage
  - Implementierung „Trick“

AUA!

- Macht es dann immer noch Sinn einen SPARQL Server zu implementieren?
  - Laufzeit, Platz

AUA!

- Macht es dann immer noch Sinn einen SPARQL Server zu implementieren?
  - Laufzeit, Platz
- Ich zeige euch jetzt noch einen Trick
  - So wie es in JENA Programmiert wurde.

Der Trick

- ARQ: wandelt Graph Pattern um mit Regeln wie
  - (T_1 *AND* (T_2 *UNION* T_3)) = (T_1 *AND* T_2) *UNION* (T_1 *AND* T_3)
  - (T_1 *OPT* (T_2 *UNION* T_3)) = (T_1 *OPT* T_2) *UNION* (T_1 *OPT* T_3)
  - ...
- Ergebnis:
  - T_1 *UNION* T_2 *UNION* T_3 *UNION* ... *UNION* T_n

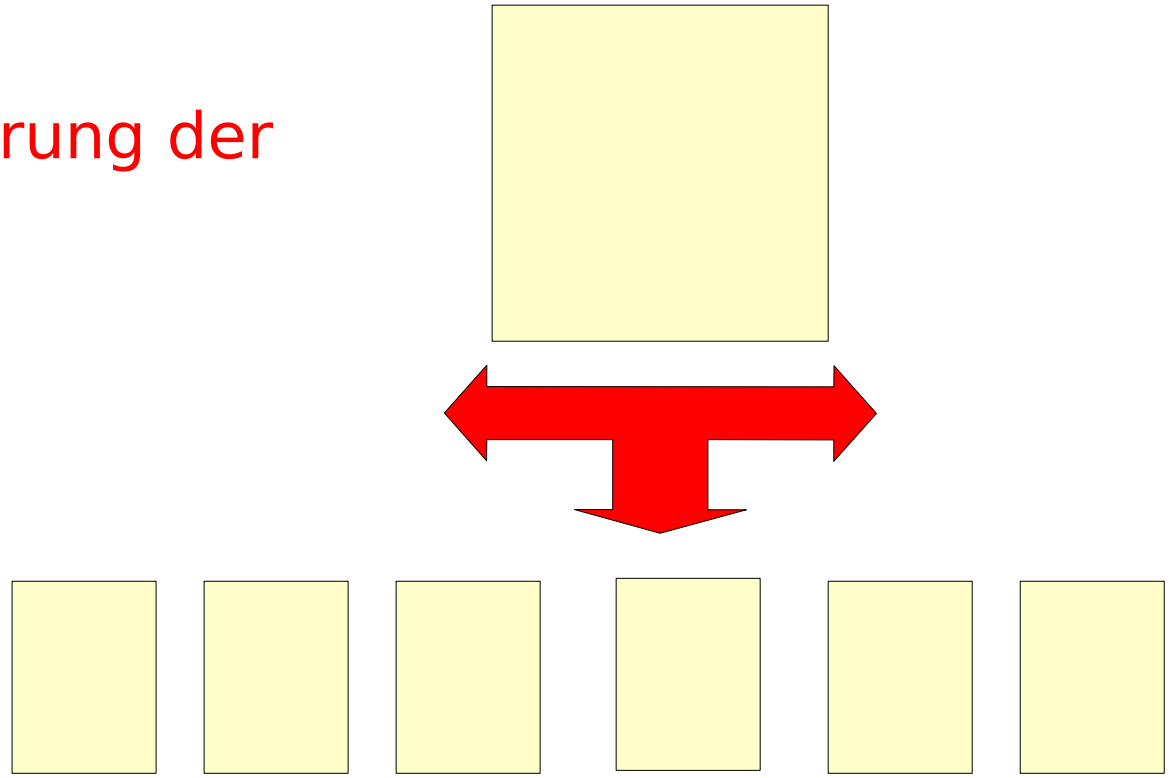
Matchen der Graph Pattern

- Graph Pattern
  - T_1 *UNION* T_2 *UNION* T_3 *UNION* ... *UNION* T_n
- Zuerst T_1 matchen, dann T_2...
- Am Ende, alle alle Ergebnisse zusammenfügen

Der Trick

- ARQ klassifiziert Knoten nach ihre Eigenschaften
  - Eingangs-, Ausgangsgrad der Knoten..

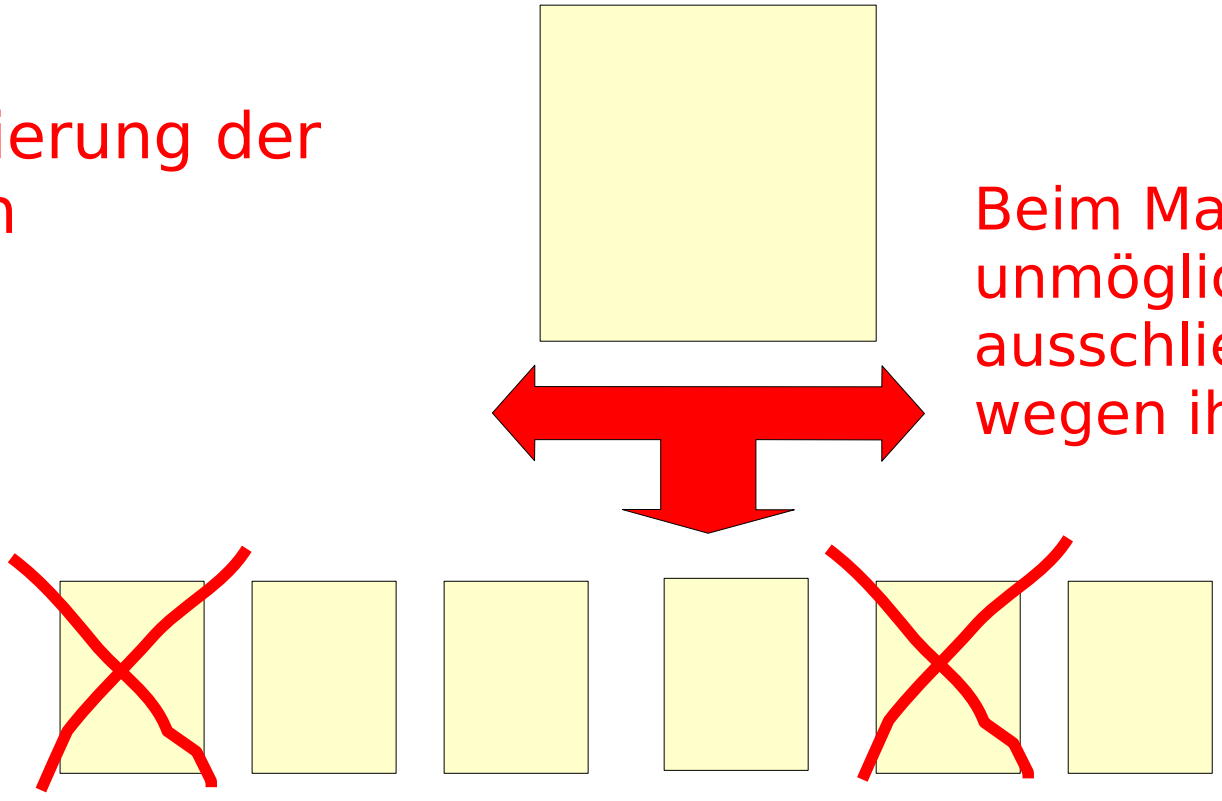
Gruppierung der Knoten



Der Trick

- ARQ klassifiziert Knoten nach ihre Eigenschaften
  - Eingangs-, Ausgangsgrad der Knoten..

Gruppierung der Knoten



Beim Matchen: unmögliche Lösungen ausschließen, z.B. wegen ihrer URI

Conclusion

- RDF Datenmodell ist ein gerichteter Graph
- SPARQL ist eine RDF-Anfragesprache
  - Viele Features
  - Hohe Komplexität
- SPARQL ist aber noch nicht „fertig“, Änderungen sind sehr wahrscheinlich

Referenz

- Querying the Semantic Web using a Relational Based SPARQL, Andrew Newman, oct 2006
- JENA ARQ http://jena.sourceforge.net/
- JJ Carroll (2002) Matching RDF graphs. Proceedings 1st International Semantic Web Conference
- E Prud'hommeaux, A Seaborne (2006). SPARQL Query Language for RDF. W3C Candidate Recommendation http://www.w3.org/TR/rdf-sparql-query/
- Jorge Perez, Mercelo Arenas, Claudio Gutierrez (2006). Semantic and complexity of SPARQL
- Jeremy J. Caroll, Christian Bizer, Pat Hayes, Patrick Stickler (2005). Named Graphs, Provenance and Trust. Proceedings of Worl Wide Web Conference WWW 2005, Japan, ACM Press

Wie Kann Man SQL und RDBMS benutzen?

- Aktive Forschung seit ueber 30 Jahren
- Es gibt wenige SPARQL Anfrage Server – Aber viele SQL Server
- Kann man SPARQL mit SQL-Server benutzen?

Wie Kann Man SQL und RDBMS benutzen?

- Aktive Forschung seit ueber 30 Jahren
- Es gibt wenige SPARQL Anfrage Server – Aber viele SQL Server
- Kann man SPARQL mit SQL-Server benutzen?

- Idee:
  - Schreiben von RDF Graphen in einer datenbank
  - Umformulierungen von den SPARQL Queries nach SQL
    - Oft werden Statements wie optional programmatisch berechnet
    - Schneller: durch direkte SQL Anfrage
      - Siehe dazu Bachelor Thesis von Andrew Newman.

Vergleich

- Andere RDF Anfragesprachen
  - Itql, RDQL, SeRQL: Aenlich zu SPARQL.
  - Itql unterstuetzt mehr features
- Vera ist Path-orientiert