Semantic Web Technologies I

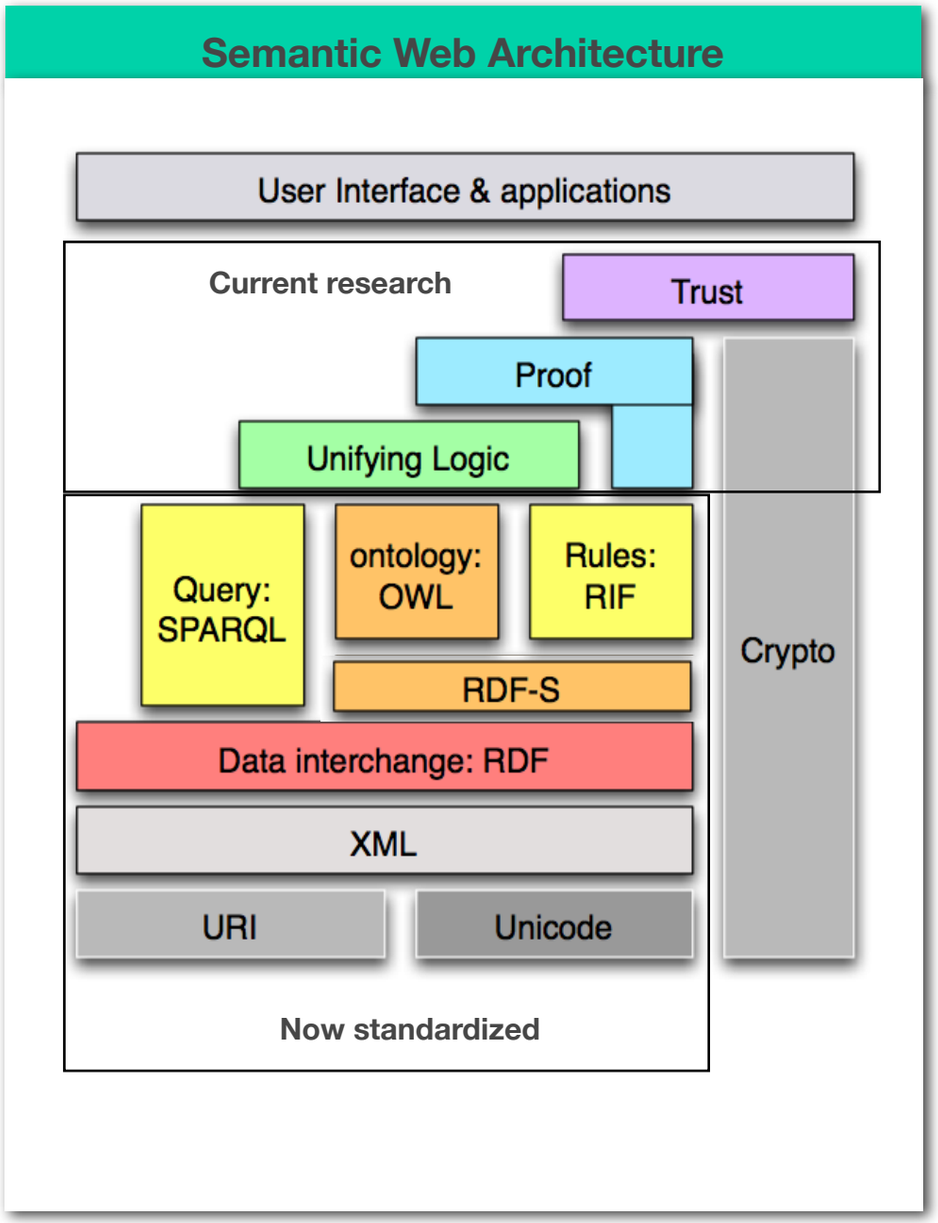
Lehrveranstaltung im WS10/11

Dr. Andreas Harth
Dr. Sebastian Rudolph
M.Sc. Anees ul Mehdi

AIFB

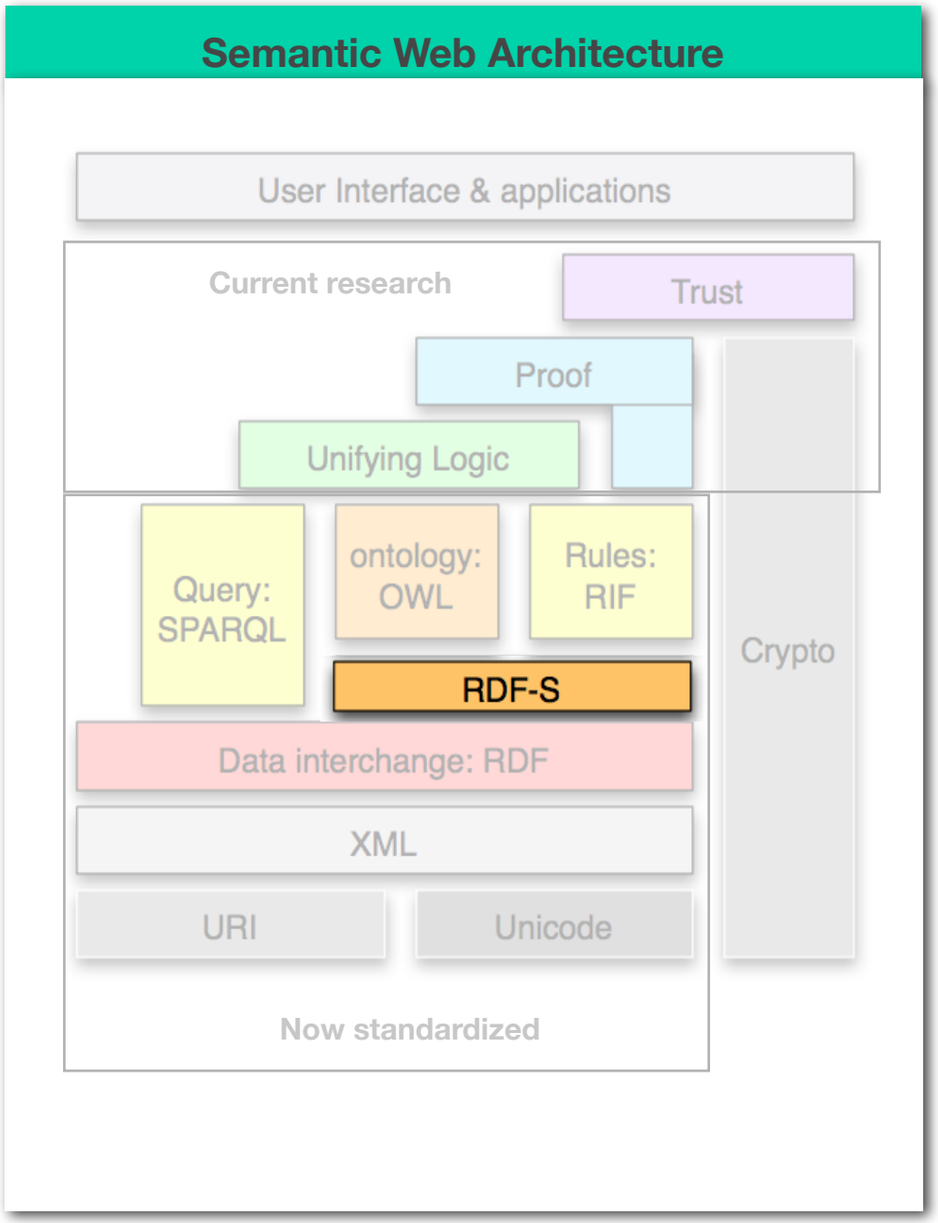
Semantik von RDF(S)

Dr. Sebastian Rudolph



Semantik von RDF(S)

Dr. Sebastian Rudolph



Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

Warum formale Semantik?

- nach Einführung von RDFS Kritik von Tool-Herstellern: verschiedene Tools - Inkompatibilitäten (trotz Spezifikation)
- z.B. bei triple stores:
  - gleiches RDF-Dokument
  - gleiche SPARQL-Anfrage
  - verschiedene Antworten
- daher: modelltheoretische Semantik für RDF(S)

Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

Was ist die Syntax?

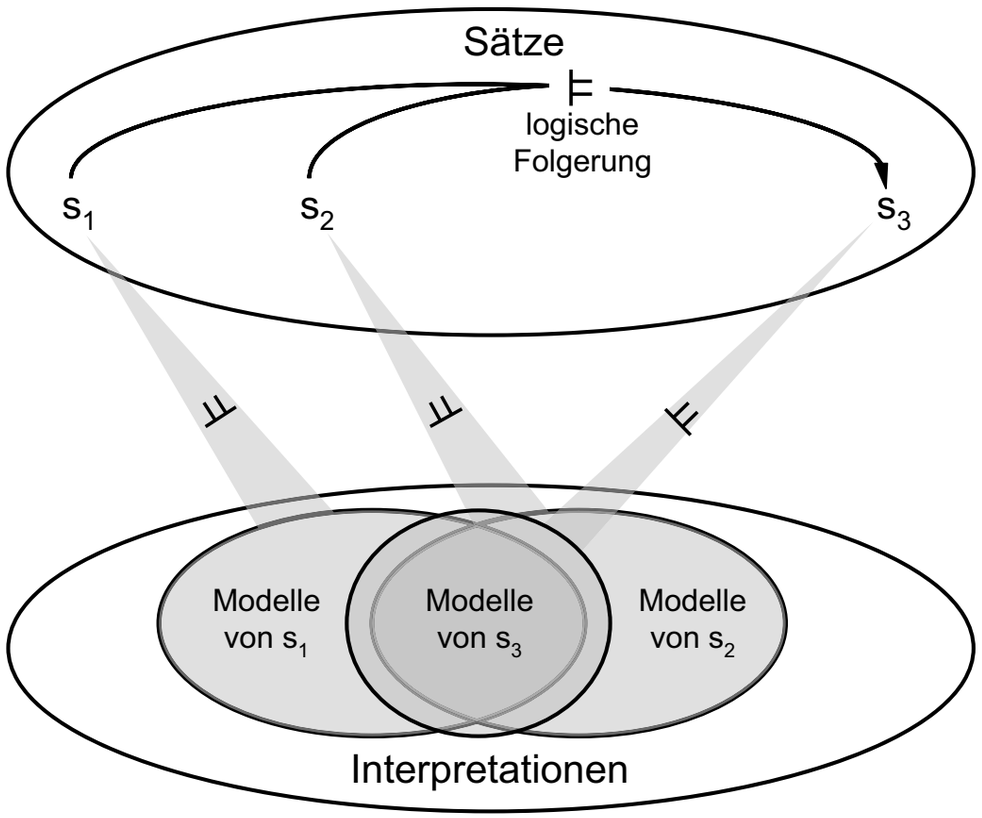
- also: was sind die Sätze in RDF(S)?
  - Grundelemente (Vokabular V): URIs, bnodes und Literale (sind selbst keine Sätze)
  - jedes Tripel

    (s,p,o) ∈
    (URI ∪ bnode) × URI × (URI ∪ bnode ∪ Literal)

    ist ein Satz
  - jede endliche Menge von Tripeln (genannt Graph) ist ein Satz

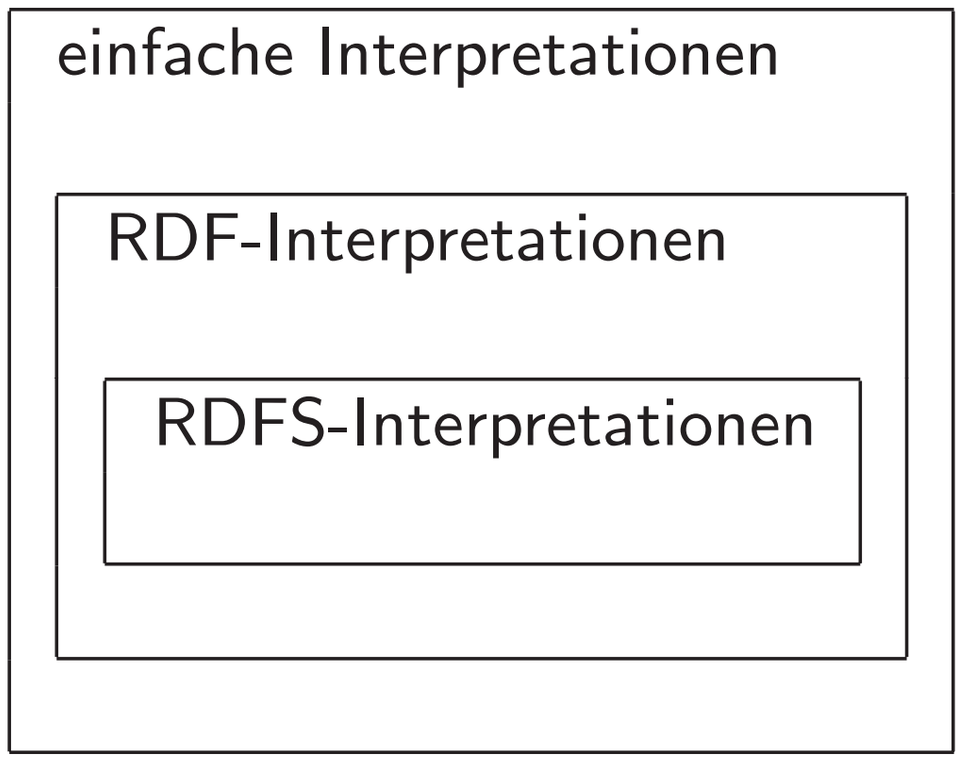
Was ist die Semantik?

- Konsequenzrelation, die sagt, wann ein RDF (S)-Graph G´ aus einem RDF(S)-Graphen G folgt, d.h. G ⊨ G´
- Modelltheoretische Semantik: wir definieren Menge von Interpretationen und legen fest, wann eine Interpretation Modell eines Graphen ist



Was ist die Semantik?

- Vorgehen schrittweise:



- je eingeschränkter die Interpretationen umso stärker die Folgerungsrelation

Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

Semantik der einfachen Folgerung

- einfache Interpretation:

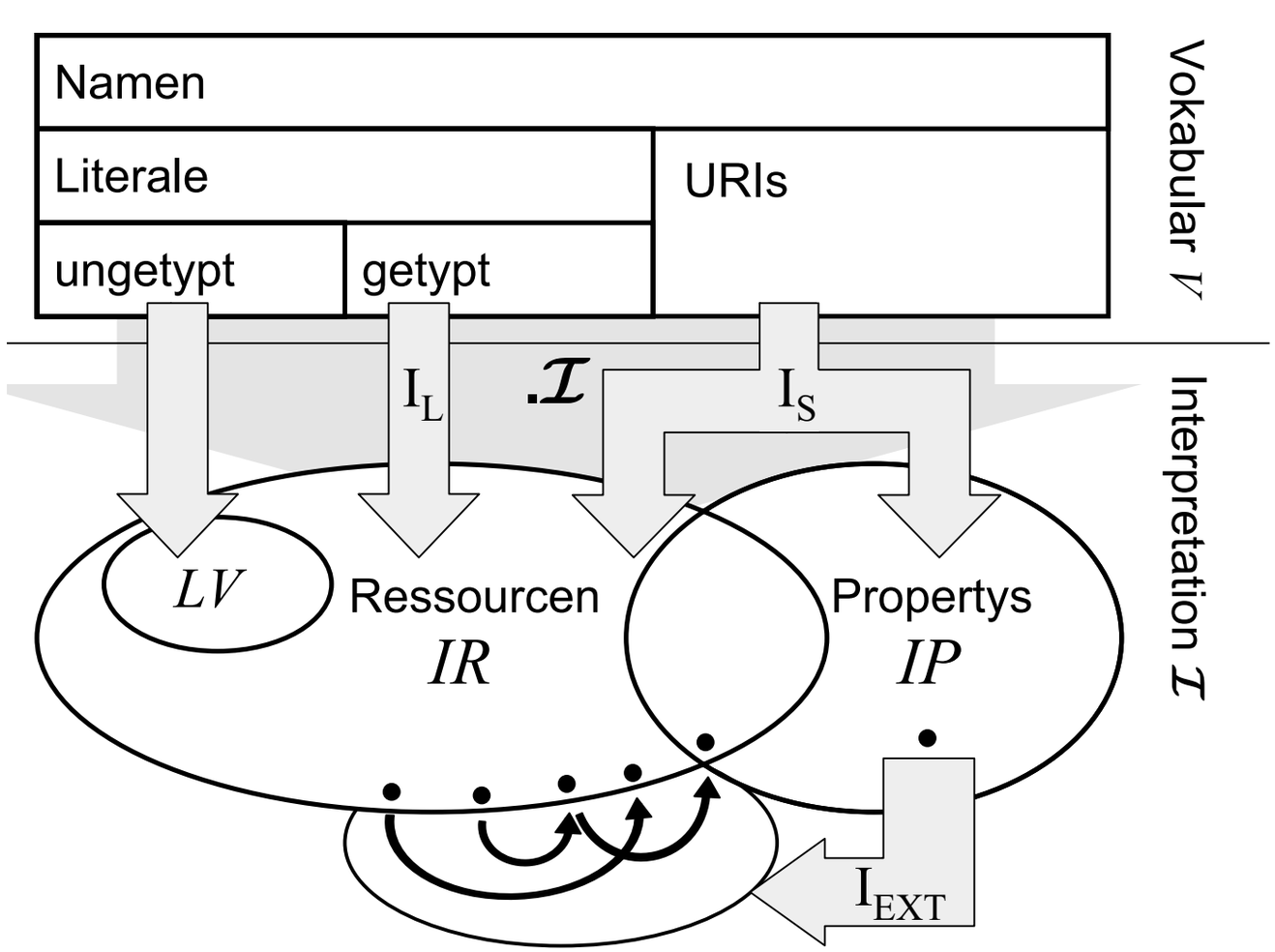
Wir definieren also: Eine *einfache Interpretation* $\mathcal{I}$ für ein Vokabular V besteht aus

- IR, einer nichtleeren Menge von *Ressourcen*, auch genannt Domäne oder (Diskurs-)Universum von $\mathcal{I}$,
- IP, der Menge der *Propertys* von $\mathcal{I}$
- $\mathrm{I_{EXT}}$, einer Funktion, welche jeder Property eine Menge von Paaren aus IR zuordnet, also $\mathrm{I_{EXT}} : IP \rightarrow 2^{IR \times IR}$, dabei nennt man $\mathrm{I_{EXT}}(p)$ auch die *Extension* der Property p,
- $\mathrm{I_S}$, einer Funktion, welche URIs aus V in die Vereinigung der Mengen IR und IP abbildet, also $\mathrm{I_S} : V \rightarrow IR \cup IP$,
- $\mathrm{I_L}$, einer Funktion von den getypten Literalen aus V in die Menge IR der Ressourcen und
- LV einer speziellen Teilmenge von IR, genannt Menge der Literalwerte, die (mindestens) alle ungetypten Literale aus V enthält.

Semantik der einfachen Folgerung

- jedes ungetypte Literal "a" wird auf a abgebildet: $(\text{"}a\text{"})^{\mathcal{I}} = a$,
- jedes ungetypte Literal mit Sprachangabe "a"@t wird auf das Paar $\langle a, t \rangle$ abgebildet: $(\text{"}a\text{"@}t)^{\mathcal{I}} = \langle a, t \rangle$,
- jedes getypte Literal l wird auf $\mathrm{I_L}(l)$ abgebildet: $l^{\mathcal{I}} = \mathrm{I_L}(l)$ und
- jede URI u wird auf $\mathrm{I_S}(u)$ abgebildet: $u^{\mathcal{I}} = \mathrm{I_S}(u)$.

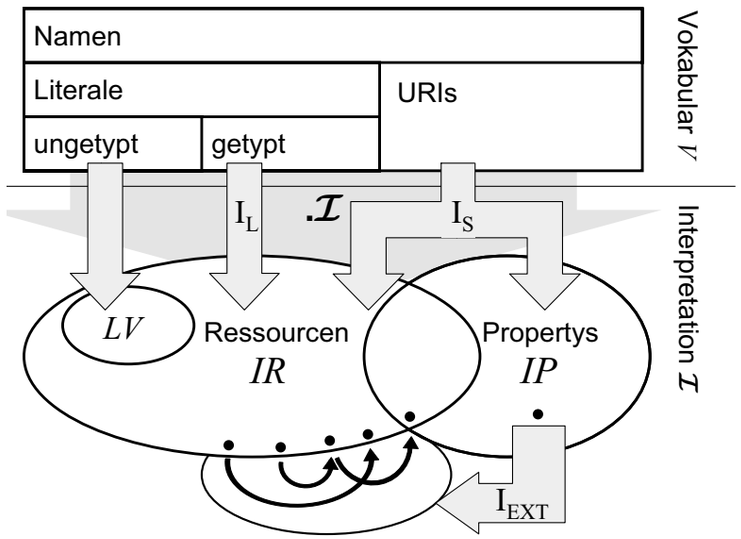
- Interpretation (schematisch):

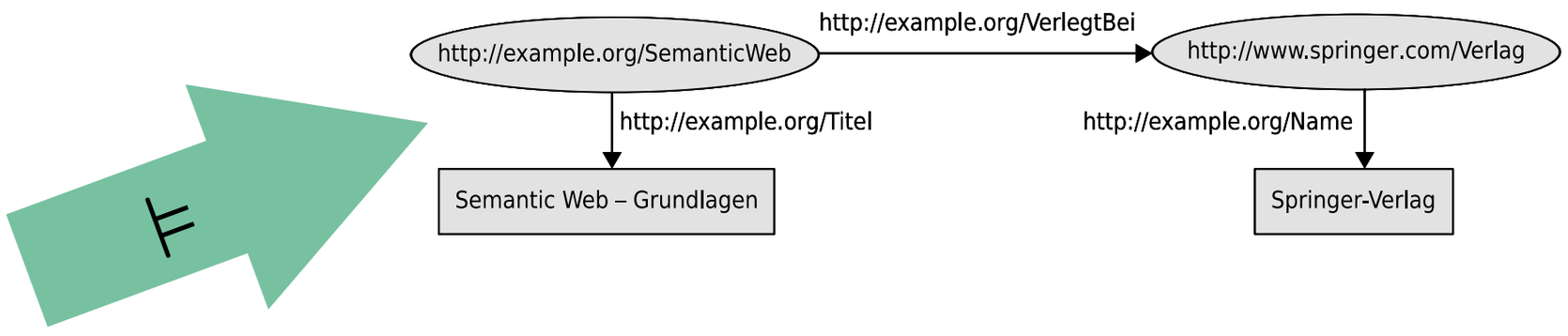


Semantik der einfachen Folgerung

AIFB

- Frage: Wann ist eine gegebene Interpretation Modell eines Graphen?



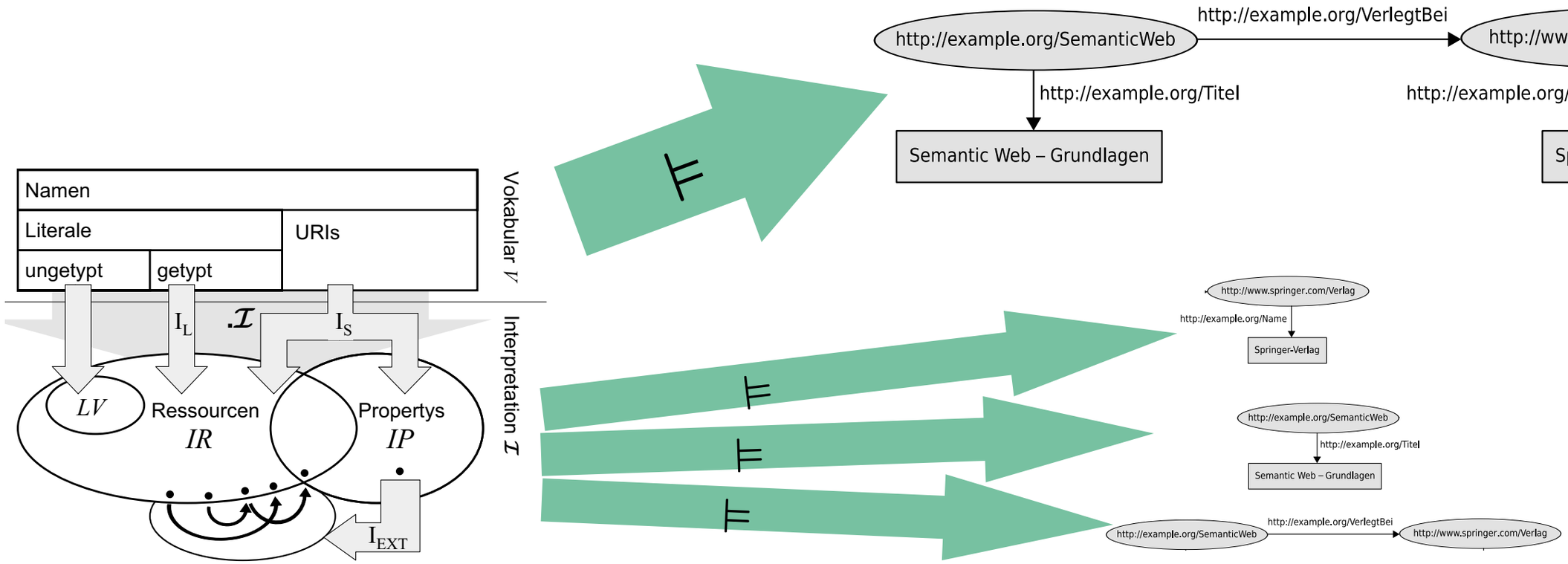


- ...wenn sie Modell jedes Tripels des Graphen ist!

Semantik der einfachen Folgerung

AIFB

- Frage: Wann ist eine gegebene Interpretation Modell eines Graphen?

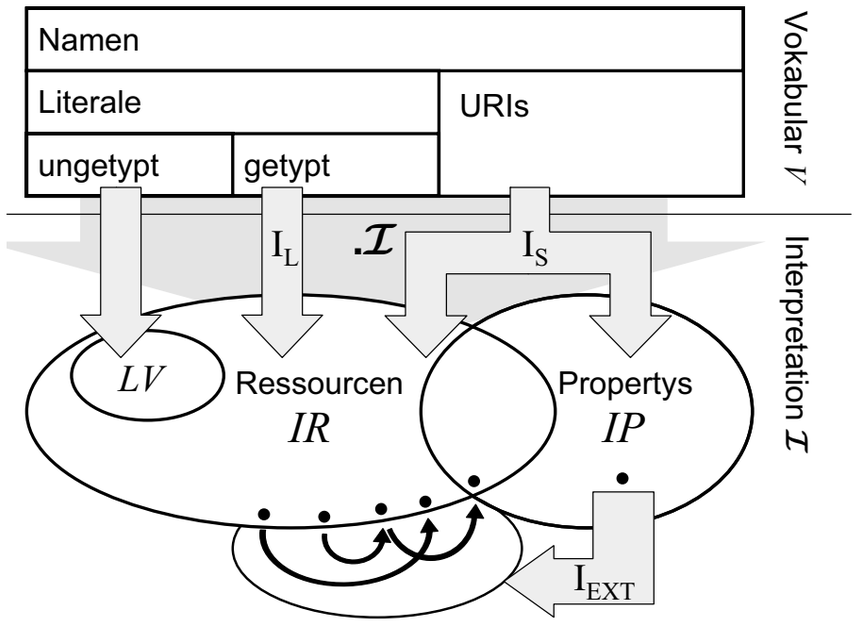


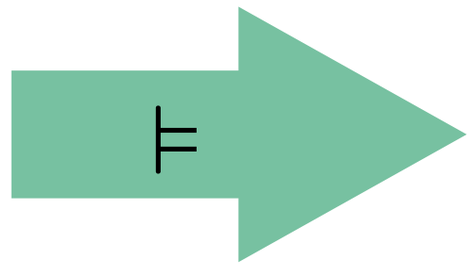
- ...wenn sie Modell jedes Tripels des Graphen ist!

Semantik der einfachen Folgerung

AIFB

- Frage: Wann ist eine gegebene Interpretation Modell eines Tripels?





Tripel

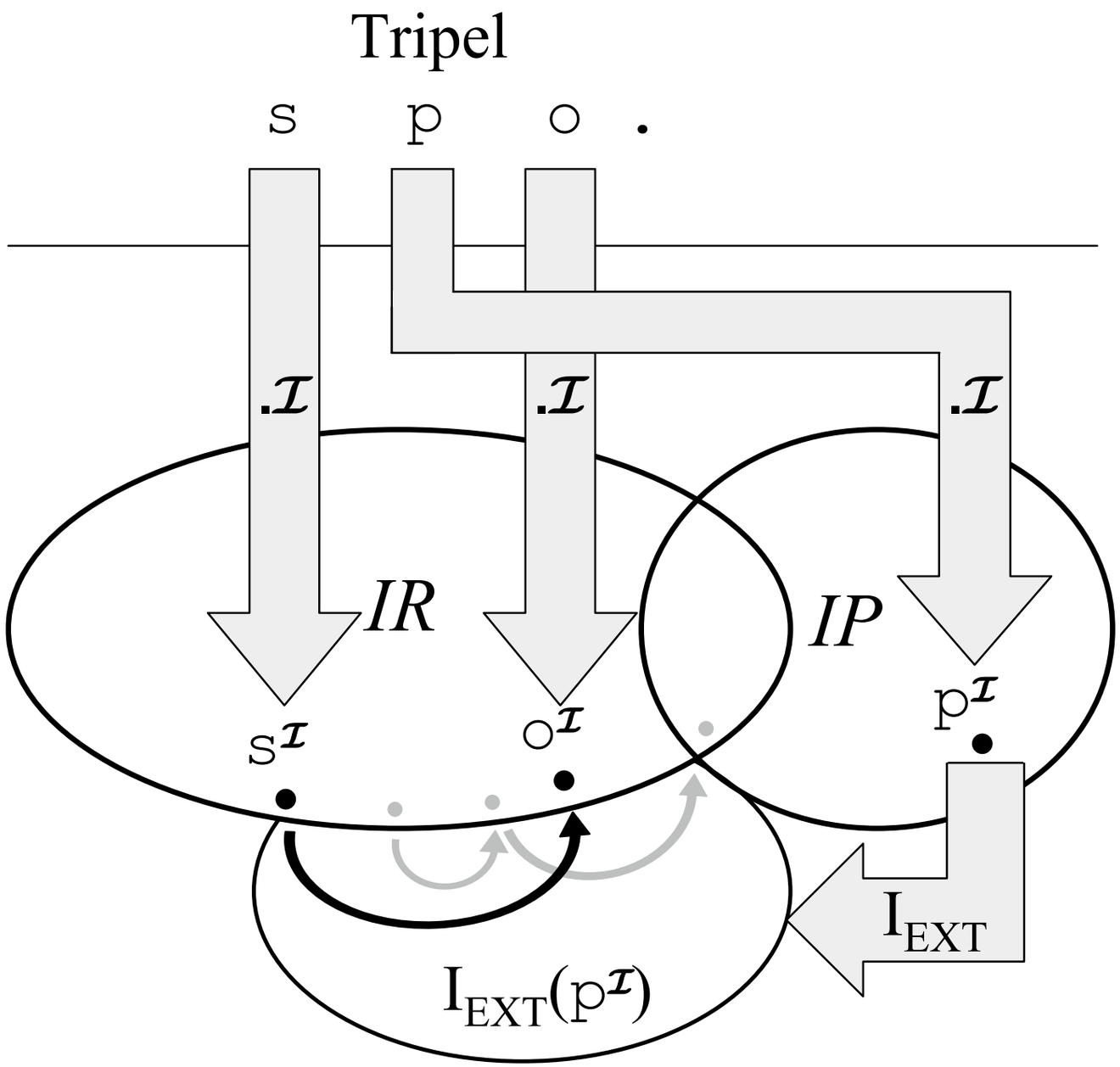
s p o .

- ...wenn Subjekt, Prädikat und Objekt in V enthalten sind und außerdem:

$$\langle \mathtt{s}^{\mathcal{I}}, \mathtt{o}^{\mathcal{I}} \rangle \in \mathrm{I}_{\mathrm{EXT}}(\mathtt{p}^{\mathcal{I}})$$

Semantik der einfachen Folgerung

- schematisch:

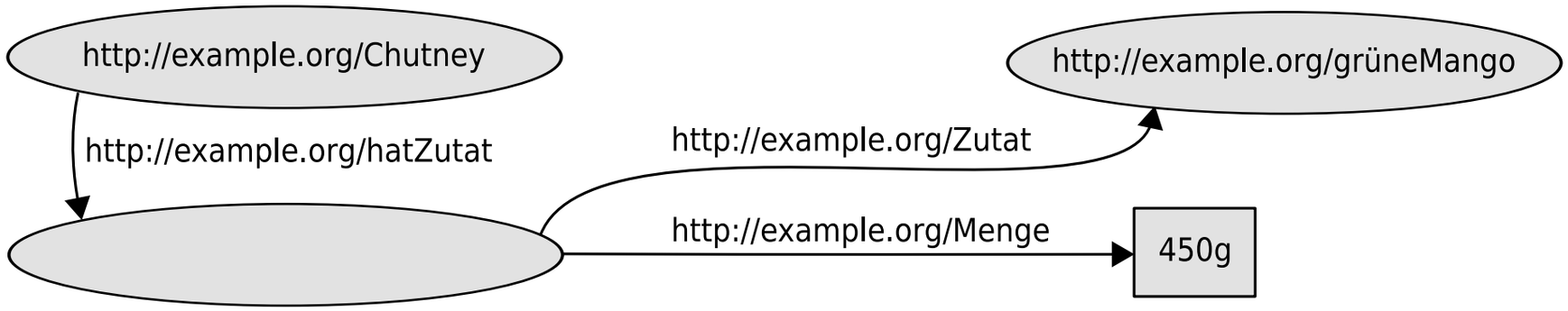


Semantik der einfachen Folgerung

- ...ups, wir haben die bnodes vergessen!
- wird nachgeholt: sei *A* Funktion, die alle bnodes auf Elemente von IR abbildet
- für eine Interpretation $\mathcal{I}$, sei $\mathcal{I}$+*A* wie $\mathcal{I}$, wobei zusätzlich für jeden bnode *b* gilt $b^{\mathcal{I}+A} = A(b)$
- eine Interpretation $\mathcal{I}$ ist nun Modell eines RDF-Graphen *G*, wenn es ein *A* gibt, so dass alle Tripel bezüglich $\mathcal{I}$+*A* wahr werden

Einfache Interpretation: Beispiel

- gegeben Graph *G*:



und Interpretation $\mathcal{I}$:

$$
\begin{aligned}
IR &= \{\chi, \upsilon, \tau, \nu, \epsilon, \iota, 450g\} \\
IP &= \{\tau, \nu, \iota\} \\
LV &= \{450g\} \\
\mathrm{I_{EXT}} &= \tau \mapsto \{\langle \chi, \epsilon \rangle\} \\
&\quad\ \nu \mapsto \{\langle \epsilon, \upsilon \rangle\} \\
&\quad\ \iota \mapsto \{\langle \epsilon, 450g \rangle\}
\end{aligned}
$$

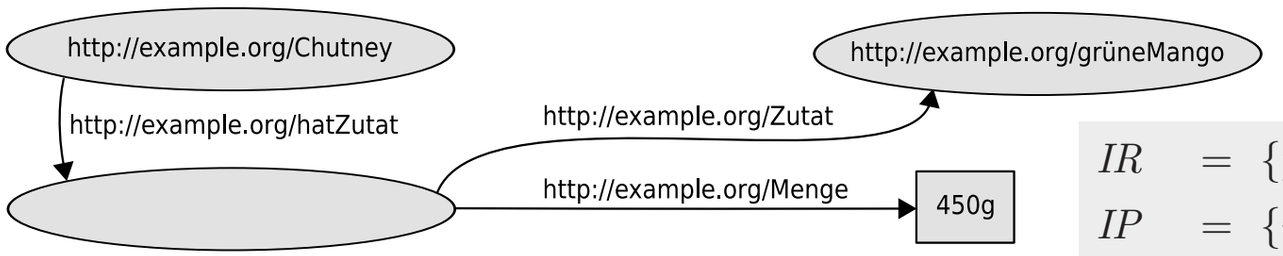
$$
\begin{aligned}
\mathrm{I_S} = \ & \texttt{ex:Chutney} && \mapsto \chi \\
& \texttt{ex:grüneMango} && \mapsto \upsilon \\
& \texttt{ex:hatZutat} && \mapsto \tau \\
& \texttt{ex:Zutat} && \mapsto \nu \\
& \texttt{ex:Menge} && \mapsto \iota
\end{aligned}
$$

$\mathrm{I_L}$ ist die „leere Funktion“, da es keine getypten Literale gibt.

- ...ist $\mathcal{I}$ ein Modell von *G*?

Einfache Interpretation: Beispiel

AIFB



$$
\begin{aligned}
IR &= \{\chi, \upsilon, \tau, \nu, \epsilon, \iota, 450g\} \\
IP &= \{\tau, \nu, \iota\} \\
LV &= \{450g\} \\
\mathrm{I_{EXT}} &= \tau \mapsto \{\langle \chi, \epsilon \rangle\} \\
&\quad\ \nu \mapsto \{\langle \epsilon, \upsilon \rangle\} \\
&\quad\ \iota \mapsto \{\langle \epsilon, 450g \rangle\}
\end{aligned}
\qquad
\begin{aligned}
\mathrm{I_S} = \ & \texttt{ex:Chutney} && \mapsto \chi \\
& \texttt{ex:grüneMango} && \mapsto \upsilon \\
& \texttt{ex:hatZutat} && \mapsto \tau \\
& \texttt{ex:Zutat} && \mapsto \nu \\
& \texttt{ex:Menge} && \mapsto \iota
\end{aligned}
$$

$\mathrm{I_L}$ ist die „leere Funktion“, da es keine getypten Literale gibt.

- wählt man $A : \texttt{\_:id1} \mapsto \epsilon$,
  dann ergibt sich

$$
\begin{aligned}
\langle \texttt{ex:Chutney}^{\mathcal{I}+A}, \texttt{\_:id1}^{\mathcal{I}+A} \rangle &= \langle \chi, \epsilon \rangle &&\in \mathrm{I_{EXT}}(\tau) = \mathrm{I_{EXT}}(\texttt{ex:hatZutat}^{\mathcal{I}+A}) \\
\langle \texttt{\_:id1}^{\mathcal{I}+A}, \texttt{ex:grüneMango}^{\mathcal{I}+A} \rangle &= \langle \epsilon, \upsilon \rangle &&\in \mathrm{I_{EXT}}(\nu) = \mathrm{I_{EXT}}(\texttt{ex:Zutat}^{\mathcal{I}+A}) \\
\langle \texttt{\_:id1}^{\mathcal{I}+A}, \texttt{"450g"}^{\mathcal{I}+A} \rangle &= \langle \epsilon, 450g \rangle &&\in \mathrm{I_{EXT}}(\iota) = \mathrm{I_{EXT}}(\texttt{ex:Menge}^{\mathcal{I}+A})
\end{aligned}
$$

- also ist $\mathcal{I}$ Modell von *G*

Einfache Folgerung

- Definition der einfachen Interpretation legt (modelltheoretisch) einfache Folgerung für RDF-Graphen fest
- Frage: wie lässt sich diese (abstrakt definierte) Semantik im Sinne des automatischen Schlussfolgerns umsetzen
- Antwort: Ableitungsregeln

Einfache Folgerung

- Ableitungsregeln für einfache Folgerung:

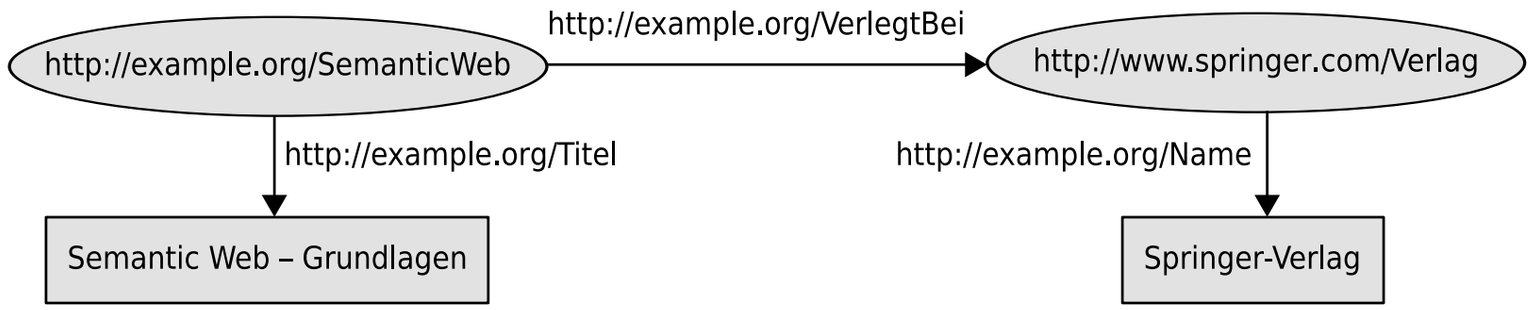
$$\frac{u \;\; a \;\; x \;\; .}{u \;\; a \;\; \_:n \;\; .}\ \text{se1}$$

$$\frac{u \;\; a \;\; x \;\; .}{\_:n \;\; a \;\; x \;\; .}\ \text{se2}$$

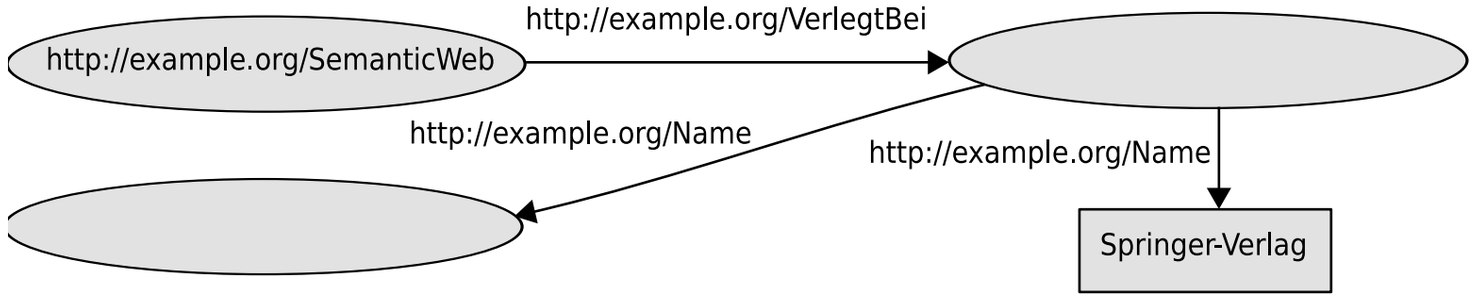
- Bedingung für Anwendung: leerer Knoten nicht bereits anderer URI / anderem Literal zugeordnet

Einfache Folgerung

- Satz: Ein Graph G_2 folgt einfach aus einem Graphen G_1, wenn G_1 mithilfe der Regeln se1 und se2 zu einem Gaphen $G_1´$ ergänzt werden kann, so dass G_2 in $G_1´$ enthalten ist.
- Bsp.: aus



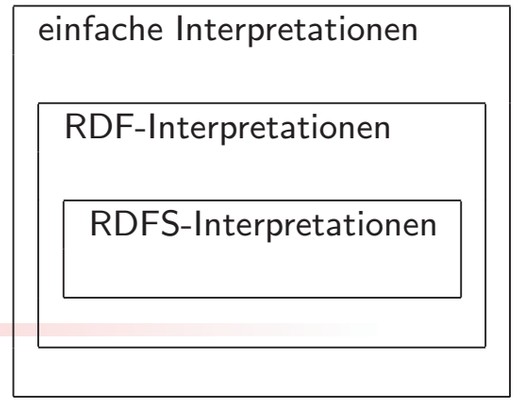
folgt einfach



Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

RDF-Interpretationen



- ...RDF-Interpretationen sind spezielle einfache Interpretationen, wobei für die URIs des RDF-Vokabulars

```
rdf:type rdf:Property rdf:XMLLiteral rdf:nil
rdf:List rdf:Statement rdf:subject rdf:predicate rdf:object
rdf:first rdf:rest rdf:Seq rdf:Bag rdf:Alt
rdf:_1 rdf:_2 ...
```

zusätzliche Forderungen gestellt werden, die die intendierte Semantik der RDF-Bezeichner realisieren:

RDF-Interpretationen

Eine *RDF-Interpretation* für ein Vokabular V ist nun eine einfache Interpretation für das Vokabular $V \cup V_{\mathrm{RDF}}$, welche zusätzlich folgende Bedingungen erfüllt:

- $x \in IP$ genau dann, wenn $\langle x, \texttt{rdf:Property}^{\mathcal{I}} \rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdf:type}^{\mathcal{I}})$.

  x ist eine Property genau dann, wenn es mit der durch `rdf:Property` bezeichneten Ressource über die `rdf:type`-Property verbunden ist (dies führt auch automatisch dazu, dass für jede RDF-Interpretation $IP \subseteq IR$ gilt).

- wenn `"`*s*`"^^rdf:XMLLiteral` in V enthalten und *s* ein wohlgeformtes XML-Literal ist, dann
  - $\mathrm{I}_{\mathrm{L}}(\texttt{"}s\texttt{"\^\^rdf:XMLLiteral})$ ist der XML-Wert[1] von *s*;
  - $\mathrm{I}_{\mathrm{L}}(\texttt{"}s\texttt{"\^\^rdf:XMLLiteral}) \in LV$;
  - $\langle \mathrm{I}_{\mathrm{L}}(\texttt{"}s\texttt{"\^\^rdf:XMLLiteral}), \texttt{rdf:XMLLiteral}^{\mathcal{I}} \rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdf:type}^{\mathcal{I}})$

- wenn `"`*s*`"^^rdf:XMLLiteral` in V enthalten und *s* ein *nicht* wohlgeformtes XML-Literal ist, dann
  - $\mathrm{I}_{\mathrm{L}}(\texttt{"}s\texttt{"\^\^rdf:XMLLiteral}) \notin LV$ und
  - $\langle \mathrm{I}_{\mathrm{L}}(\texttt{"}s\texttt{"\^\^rdf:XMLLiteral}), \texttt{rdf:XMLLiteral}^{\mathcal{I}} \rangle \notin \mathrm{I}_{\mathrm{EXT}}(\texttt{rdf:type}^{\mathcal{I}})$.

RDF-Interpretationen

- zusätzliche Forderung: jede RDF-Interpretation muss Modell der folgenden, „axiomatischen“ Tripel sein:

```
rdf:type        rdf:type    rdf:Property .
rdf:subject     rdf:type    rdf:Property .
rdf:predicate   rdf:type    rdf:Property .
rdf:object      rdf:type    rdf:Property .
rdf:first       rdf:type    rdf:Property .
rdf:rest        rdf:type    rdf:Property .
rdf:value       rdf:type    rdf:Property .
rdf:_1          rdf:type    rdf:Property .
rdf:_2          rdf:type    rdf:Property .
...
rdf:nil         rdf:type    rdf:List .
```

RDF-Folgerungen

- automatische Folgerungen werden wieder über Ableitungsregeln realisiert:

$$\frac{}{u \quad a \quad x}\ \text{rdfax}$$

jedes axiomatische Tripel „u a x .“ kann immer abgeleitet werden

$$\frac{u \quad a \quad l \quad .}{u \quad a \quad \_:n \quad .}\ \text{lg}$$

Literals dürfen durch nicht anderweitig gebundene bnodes ersetzt werden

$$\frac{u \quad a \quad y \quad .}{a \ \texttt{rdf:type rdf:Property} \ .}\ \text{rdf1}$$

für jedes Tripelprädikat kann abgeleitet werden dass es eine Entität aus der Klasse der Properties ist

$$\frac{u \quad a \quad l \quad .}{\_:n \ \texttt{rdf:type rdf:XMLLiteral}}\ \text{rdf2}$$

wenn _:n durch lg dem wohlgeformten XML-Literal l zugewiesen wurde

RDF-Folgerung

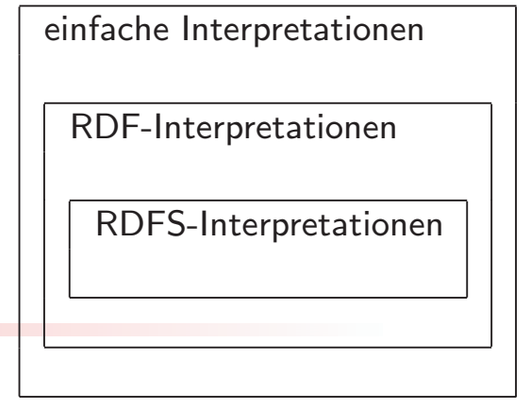
- Satz: Ein Graph G_2 RDF-folgt aus einem Graphen G_1, wenn es einen Graphen G_1' gibt, so dass
  - G_1' aus G_1 via lg, rdf1, rdf2 und rdfax hergeleitet werden kann und
  - G_2 aus G_1' einfach folgt.

- Beachte: zweistufiger Folgerungsprozess.

Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

RDFS-Interpretationen



- ...RDFS-Interpretationen sind spezielle RDF-Interpretationen, wobei für die URIs des RDFS-Vokabulars

```
rdfs:domain rdfs:range rdfs:Resource rdfs:Literal rdfs:Datatype
rdfs:Class rdfs:subClassOf rdfs:subPropertyOf rdfs:member
rdfs:Container rdfs:ContainerMembershipProperty rdfs:comment
rdfs:seeAlso rdfs:isDefinedBy rdfs:label
```

zusätzliche Forderungen gestellt werden, die die intendierte Semantik der RDF-Bezeichner realisieren:

RDFS-Interpretationen

Eine *RDFS-Interpretation* für ein Vokabular V ist eine RDF-Interpretation des Vokabulars $V \cup V_{\text{RDFS}}$, welche zusätzlich die folgenden Kriterien erfüllt:

- $IR = \text{I}_{\text{CEXT}}(\texttt{rdfs:Resource}^{\mathcal{I}})$
  Jede Ressource ist vom Typ `rdfs:Resource`.
- $LV = \text{I}_{\text{CEXT}}(\texttt{rdfs:Literal}^{\mathcal{I}})$
  Jedes ungetypte und jedes wohlgeformte getypte Literal ist vom Typ `rdfs:Literal`.
- Wenn $\langle x, y\rangle \in \text{I}_{\text{EXT}}(\texttt{rdfs:domain}^{\mathcal{I}})$ und $\langle u, v\rangle \in \text{I}_{\text{EXT}}(x)$,
  dann $u \in \text{I}_{\text{CEXT}}(y)$.
  Ist x mit y durch die Property `rdfs:domain` verbunden und verbindet die Property x die Ressourcen u und v, dann ist u vom Typ y.
- Wenn $\langle x, y\rangle \in \text{I}_{\text{EXT}}(\texttt{rdfs:range}^{\mathcal{I}})$ und $\langle u, v\rangle \in \text{I}_{\text{EXT}}(x)$,
  dann $v \in \text{I}_{\text{CEXT}}(y)$.
  Ist x mit y durch die Property `rdfs:range` verbunden und verbindet die Property x die Ressourcen u und v, dann ist v vom Typ y.
- $\text{I}_{\text{EXT}}(\texttt{rdfs:subPropertyOf}^{\mathcal{I}})$ ist reflexiv und transitiv auf IP.
  Die `rdfs:subPropertyOf`-Property verbindet jede Property mit sich selbst.
  Darüber hinaus gilt: Verbindet `rdfs:subPropertyOf` die Property x mit Property y und außerdem y mit der Property z, so verbindet `rdfs:subPropertyOf` auch x direkt mit z.

RDFS-Interpretationen

- Wenn $\langle x, y\rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdfs:subPropertyOf}^{\mathcal{I}})$,
  dann $x, y \in IP$ und $\mathrm{I}_{\mathrm{EXT}}(x) \subseteq \mathrm{I}_{\mathrm{EXT}}(y)$.
  Wird x mit y durch `rdfs:subPropertyOf` verbunden, dann sind sowohl x als auch y Propertys und jedes in der Extension von x enthaltene Ressourcenpaar ist auch in der Extension von y enthalten.
- Wenn $x \in IC$,
  dann $\langle x, \texttt{rdfs:Resource}^{\mathcal{I}}\rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$.
  Bezeichnet x eine Klasse, dann muss es eine Unterklasse der Klasse aller Ressourcen sein, d.h., das Paar aus x und `rdfs:Resource` ist in der Extension von `rdfs:subClassOf`.
- Wenn $\langle x, y\rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$,
  dann $x, y \in IC$ und $\mathrm{I}_{\mathrm{CEXT}}(x) \subseteq \mathrm{I}_{\mathrm{CEXT}}(y)$.
  Stehen x und y in der `rdfs:subClassOf`-Beziehung, sind sowohl x als auch y Klassen und die (Klassen-)Extension von x ist Teilmenge der (Klassen-)Extension von y.
- $\mathrm{I}_{\mathrm{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$ ist reflexiv und transitiv auf IC.
  Die `rdfs:subClassOf`-Property verbindet jede Klasse mit sich selbst. Darüber hinaus folgt, wann immer diese Property Klasse x mit Klasse y und Klasse y mit Klasse z verbindet, dass sie x auch direkt mit z verbindet.

RDFS-Interpretationen

- Wenn $x \in \mathrm{I}_{\mathrm{CEXT}}(\texttt{rdfs:ContainerMembershipProperty}^{\mathcal{I}})$,
  dann $\langle x, \texttt{rdfs:member}^{\mathcal{I}} \rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdfs:subPropertyOf}^{\mathcal{I}})$.
  Ist x eine Property vom Typ `rdfs:ContainerMembershipProperty`, so steht sie in der `rdfs:subPropertyOf`-Beziehung zur `rdfs:member`-Property.
- Wenn $x \in \mathrm{I}_{\mathrm{CEXT}}(\texttt{rdfs:Datatype}^{\mathcal{I}})$,
  dann $\langle x, \texttt{rdfs:Literal}^{\mathcal{I}} \rangle \in \mathrm{I}_{\mathrm{EXT}}(\texttt{rdfs:subClassOf}^{\mathcal{I}})$
  Ist ein x als Element der Klasse `rdfs:Datatype` „getypt“, dann muss dieses auch eine Unterklasse der Klasse aller Literalwerte (bezeichnet mit `rdfs:Literal`) sein.

- ...dazu kommen dann noch jede Menge weitere axiomatische Tripel:

RDFS-Interpretationen

AIFB

```
rdf:type            rdfs:domain      rdfs:Resource .
rdfs:domain         rdfs:domain      rdf:Property .
rdfs:range          rdfs:domain      rdf:Property .
rdfs:subPropertyOf  rdfs:domain      rdf:Property .
rdfs:subClassOf     rdfs:domain      rdfs:Class .
rdf:subject         rdfs:domain      rdf:Statement .
rdf:predicate       rdfs:domain      rdf:Statement .
rdf:object          rdfs:domain      rdf:Statement .
rdfs:member         rdfs:domain      rdfs:Resource .
rdf:first           rdfs:domain      rdf:List .
rdf:rest            rdfs:domain      rdf:List .
rdfs:seeAlso        rdfs:domain      rdfs:Resource .
rdfs:isDefinedBy    rdfs:domain      rdfs:Resource .
rdfs:comment        rdfs:domain      rdfs:Resource .
rdfs:label          rdfs:domain      rdfs:Resource .
rdf:value           rdfs:domain      rdfs:Resource .

rdf:type            rdfs:range       rdfs:Class .
rdfs:domain         rdfs:range       rdfs:Class .
rdfs:range          rdfs:range       rdfs:Class .
rdfs:subPropertyOf  rdfs:range       rdf:Property .
rdfs:subClassOf     rdfs:range       rdfs:Class .
rdf:subject         rdfs:range       rdfs:Resource .
rdf:predicate       rdfs:range       rdfs:Resource .
rdf:object          rdfs:range       rdfs:Resource .
rdfs:member         rdfs:range       rdfs:Resource .
rdf:first           rdfs:range       rdfs:Resource .
rdf:rest            rdfs:range       rdf:List .
rdfs:seeAlso        rdfs:range       rdfs:Resource .
rdfs:isDefinedBy    rdfs:range       rdfs:Resource .
rdfs:comment        rdfs:range       rdfs:Literal .
rdfs:label          rdfs:range       rdfs:Literal .
rdf:value           rdfs:range       rdfs:Resource .
```

```
rdfs:ContainerMembershipProperty
                    rdfs:subClassOf  rdf:Property .
rdf:Alt             rdfs:subClassOf  rdfs:Container .
rdf:Bag             rdfs:subClassOf  rdfs:Container .
rdf:Seq             rdfs:subClassOf  rdfs:Container .

rdfs:isDefinedBy    rdfs:subPropertyOf  rdfs:seeAlso .

rdf:XMLLiteral      rdf:type         rdfs:Datatype .
rdf:XMLLiteral      rdfs:subClassOf  rdfs:Literal .
rdfs:Datatype       rdfs:subClassOf  rdfs:Class .

rdf:_1              rdf:type
                       rdfs:ContainerMembershipProperty .
rdf:_1              rdfs:domain      rdfs:Resource .
rdf:_1              rdfs:range       rdfs:Resource .
rdf:_2              rdf:type
                       rdfs:ContainerMembershipProperty .
rdf:_2              rdfs:domain      rdfs:Resource .
rdf:_2              rdfs:range       rdfs:Resource .
...
```

RDFS-Folgerung

AIFB

- automatische Folgerungen werden wieder über Ableitungsregeln realisiert:

$$\frac{}{u\ a\ x}\ \text{rdfsax}$$

$$\frac{u\ \texttt{rdfs:subPropertyOf}\ v\ .\quad v\ \texttt{rdfs:subPropertyOf}\ x\ .}{u\ \texttt{rdfs:subPropertyOf}\ x\ .}\ \text{rdfs5}$$

$$\frac{u\ \texttt{rdf:type rdfs:ContainerMembershipProperty}\ .}{u\ \texttt{rdfs:subPropertyOf rdfs:member}\ .}\ \text{rdfs12}$$

$$\frac{u\ a\ \_{:}n\ .}{u\ a\ l\ .}\ \text{gl}$$

$$\frac{u\ \texttt{rdf:type rdf:Property}\ .}{u\ \texttt{rdfs:subPropertyOf}\ u\ .}\ \text{rdfs6}$$

$$\frac{u\ \texttt{rdf:type rdfs:Datatype}\ .}{u\ \texttt{rdfs:subClassOf rdfs:Literal}\ .}\ \text{rdfs13}$$

$$\frac{u\ a\ l\ .}{\_{:}n\ \texttt{rdf:type rdfs:Literal}\ .}\ \text{rdfs1}$$

$$\frac{a\ \texttt{rdfs:subPropertyOf}\ b\ .\quad u\ a\ y\ .}{u\ b\ y\ .}\ \text{rdfs7}$$

$$\frac{a\ \texttt{rdfs:domain}\ x\ .\quad u\ a\ y\ .}{u\ \texttt{rdf:type}\ x\ .}\ \text{rdfs2}$$

$$\frac{u\ \texttt{rdf:type rdfs:Class}\ .}{u\ \texttt{rdfs:subClassOf rdfs:Resource}\ .}\ \text{rdfs8}$$

$$\frac{a\ \texttt{rdfs:range}\ x\ .\quad u\ a\ v\ .}{v\ \texttt{rdf:type}\ x\ .}\ \text{rdfs3}$$

$$\frac{u\ \texttt{rdfs:subClassOf}\ x\ .\quad v\ \texttt{rdf:type}\ u\ .}{v\ \texttt{rdf:type}\ x\ .}\ \text{rdfs9}$$

$$\frac{u\ a\ x\ .}{u\ \texttt{rdf:type rdfs:Resource}\ .}\ \text{rdfs4a}$$

$$\frac{u\ \texttt{rdf:type rdfs:Class}\ .}{u\ \texttt{rdfs:subClassOf}\ u\ .}\ \text{rdfs10}$$

$$\frac{u\ a\ v\ .}{v\ \texttt{rdf:type rdfs:Resource}\ .}\ \text{rdfs4b}$$

$$\frac{u\ \texttt{rdfs:subClassOf}\ v\ .\quad v\ \texttt{rdfs:subClassOf}\ x\ .}{u\ \texttt{rdfs:subClassOf}\ x\ .}\ \text{rdfs11}$$

RDFS-Folgerung

- wichtige Definition: XML-Clash

```
ex:hatSmiley          rdfs:range      rdf:Literal .
ex:böseBemerkung      ex:hatSmiley    ">:->"^^XMLLiteral .
```

- tritt auf, wenn einem Knoten vom Typ rdf:Literal ein nicht-wohlgeformter Literalwert zugewiesen werden muss.

RDFS-Folgerung

- Satz: Ein Graph RDFS-folgt aus G_1 genau dann, wenn es einen Graphen G_1' gibt, der durch Anwendung der Regeln lg, gl, rdfax, rdf1, rdf2, rdfs1 – rdfs13 und rdfsax aus G_1 folgt, so dass
  - G_2 aus G_1' einfach folgt oder
  - G_1' einen XML-Clash enthält.

Agenda

- Motivation
- Vorbetrachtungen
- einfache Folgerung
- RDF-Folgerung
- RDFS-Folgerung
- Unzulänglichkeiten von RDF(S)

Was kann RDF(S) nicht?

- bestimmte (vernünftig) scheinende Folgerungen können nicht RDFS-gefolgert werden, z.B.

```
ex:sprichtMit    rdfs:domain        ex:Homo .
ex:Homo          rdfs:subClassOf    ex:Primates .
```

impliziert

```
ex:sprichtMit    rdfs:domain        ex:Primates .
```

- mögliche Lösung: noch stärkere, „extensionale“, Semantik
- keine Möglichkeit, Negation auszudrücken