

RDFS - Semántica Formal

Lenguajes y Tecnologías de la Web Semántica

Concepción de Sistemas de Información

2021



Contenidos

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- Semántica de RDFS
 - Semántica Simple
 - Semántica de Tipos de Datos
 - Semántica de RDF
 - Semántica de RDFS

Interpretar

- Para comprender el significado de algo, se necesita una referencia, un contexto sobre el que se puede interpretar la expresión.
- **Interpretar** es establecer una correspondencia (en general, funcional) entre los símbolos de la expresión y elementos de ese contexto.
- Una **interpretación o modelo** \mathfrak{A} , usualmente se define como una **estructura** $\mathfrak{A} = \langle U, A_1, A_2, \dots \rangle$ en donde U es el universo y A_i son conjuntos, funciones, relaciones o elementos individuales construidos sobre U con dependencias entre ellos. Estas dependencias, pueden ser muy simples o muy complejas.

Interpretación y Estructuras: Ejemplo

- La expresión “Se murió la perra de mi suegra” se puede interpretar en dos modelos diferentes. (Ej. en Primer Orden)
 - $\langle \{p, s\}, \{p\}, \{s\}, \{ \langle p, s \rangle \}, \{p\} \rangle$ donde:
 - El primer elemento es el dominio A (perra \neq suegra).
 - El segundo es el conjunto de los perros ($\subset A$).
 - El tercero es el conjunto de las personas ($\subset A$).
 - El cuarto es la relación “esDe” ($\subset A \times A$).
 - El quinto es el conjunto de “Muertos” .
 - $\langle \{s\}, \{s\}, \{s\}, \{s\} \rangle$
 - El primer elemento es el dominio A (perra = suegra).
 - El segundo es el conjunto de los “perros” ($\subset A$).
 - El tercero es el conjunto de las personas ($\subset A$).
 - El cuarto es el conjunto de “Muertos” .

Interpretación y Estructuras

- Siempre se manejan dos niveles:
 - **Lenguaje:** conjunto de tiras de símbolos que describen afirmaciones y se interpretan usando las estructuras (en el ejemplo “Se murió la perra de mi suegra”).
 - **Semántica:** estructuras que contienen “configuraciones del mundo” en las que las afirmaciones del lenguaje pueden ser *verdaderas* o *falsas*. Estas estructuras, en términos de desarrollo de software, se asocian con *una realidad* de un problema determinado.
- Típicamente se escriben conjuntos de afirmaciones para describir una realidad dada (Especificación) y esto termina refiriendo a conjuntos de estructuras
- Las estructuras que hacen verdaderas a todas las afirmaciones. Se dice que estas estructuras son **Modelos** de las afirmaciones.

Interpretación y Estructuras:

Observaciones

- Hay afirmaciones que se cumplen en cualquier estructura.
 - Se les llama **Verdades Lógicas, Lógicamente Válidas** o **Tautologías**.
 - Como no discriminan las estructuras, no aportan información. Describen (son verdaderas) en cualquier realidad.
 - Ej: La tierra gira alrededor del Sol o la tierra no gira alrededor del sol.
- Hay afirmaciones que no se cumplen en ninguna estructura.
 - Se les llama **Contradicciones**.
 - No describen ninguna realidad.

Interpretación y Estructuras:

Observaciones

- Dado un conjunto de afirmaciones:
 - Si *tiene alguna estructura que las hace verdaderas*, el conjunto de afirmaciones es **consistente**.
 - Si *no hay ninguna estructura tal que todas las afirmaciones son verdaderas*, el conjunto de afirmaciones es **inconsistente**.
 - Si el conjunto es consistente, hay algunas afirmaciones que se pueden deducir como ciertas a partir del conjunto. Esas son las **consecuencias**.
 - En ese caso, si se agrega una afirmación que no es consecuencia, entonces disminuye el conjunto de estructuras que pueden hacer verdaderas a todas la afirmaciones del nuevo conjunto.

Interpretación y Estructuras: Observaciones

- Interesa centrarse en:
 - Decidir si un conjunto de afirmaciones es consistente o no.
 - Decidir si una afirmación es verdadera o no.
 - A partir de un conjunto de afirmaciones dado que es consistente, encontrar las consecuencias.

Consecuencias

- Hay dos enfoques para determinar si una afirmación es una consecuencia de un conjunto de afirmaciones :
- Verificar que todos los modelos del conjunto también son modelos de la afirmación.
- Verificar que no hay ningún modelo del conjunto que no sea modelo de la afirmación.
- Tener un conjunto de reglas sintácticas (no se preocupan de los modelos) que permiten determinar si una afirmación es consecuencia de un conjunto.

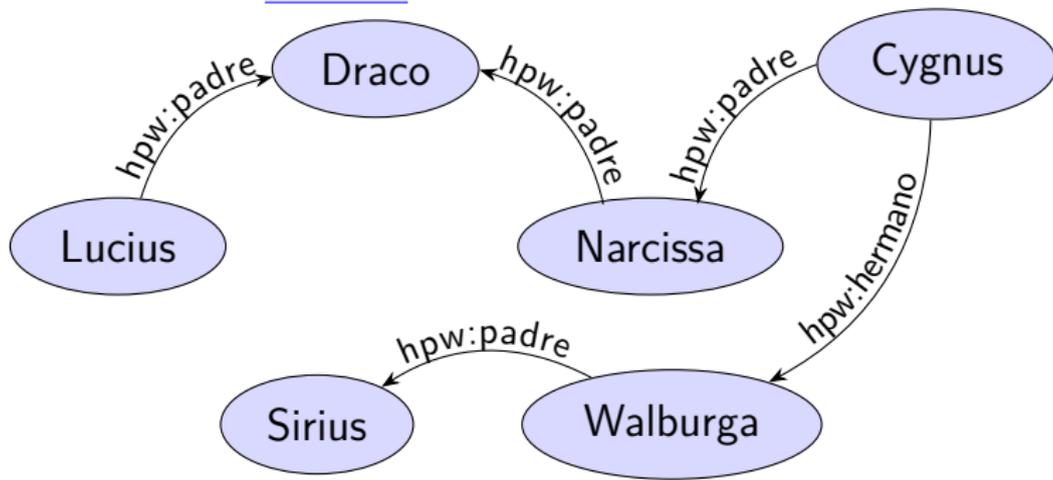
Todos los hombres son mortales. Socrates es un hombre
Socrates es Mortal

Contenidos

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- Semántica de RDFS
 - Semántica Simple
 - Semántica de Tipos de Datos
 - Semántica de RDF
 - Semántica de RDFS

Semántica: Interpretación + Consecuencia

- Una terna, representa un hecho.
 - Ej: Lucius es el padre de Draco.
hpw:Lucius hpw:padre hpw:Draco.
- Una grafo, representa entonces un conjunto de hechos . Ref: [Arbol Familiar](#)



Semántica: Interpretación + Consecuencia

- Una forma de definir la semántica es dar dos elementos:
 - La interpretación: una función que recibe expresiones (un grafo) y que se evalúa sobre un modelo dado devolviendo la verdad, o sea, si los hechos se cumplen o no en ese modelo.
 - La noción de consecuencia (entailment): dice cuáles son los hechos que necesariamente se deben cumplir, a partir de los hechos que conocemos.
- De otra forma, los mecanismos de consecuencia deben devolver (o verificar) grafos que se deducen a partir del grafo de hipótesis.

Contenidos

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- **Semántica de RDFS**
 - Semántica Simple
 - Semántica de Tipos de Datos
 - Semántica de RDF
 - Semántica de RDFS

Semántica de RDFS

- Hay cuatro definiciones de la semántica de RDF:
 - Simple
 - Datatype
 - RDF
 - RDFS
- La diferencia es que en cada caso, se agregan restricciones para incluir la semántica de los términos del vocabulario, tanto en las funciones de interpretación como en la noción de consecuencia. (Extensión Semántica)

Definiciones Básicas

- En una **interpretación** intervienen dos elementos:
 - Una estructura que define un dominio y una serie de funciones asociadas.
 - Una función que indica cómo se mapean las IRI's y los literales sobre el dominio y define las nociones de verdad sobre las ternas y grafos. Esta función se define sobre la estructura anterior.
- Un **nombre** es una IRI o un literal. Un literal tipeado tiene dos nombres: el literal en sí mismo y el la IRI del tipo.
- Un **vocabulario** es un conjunto de nombres.
- Un **grafo base** es un grafo que no tiene nodos blancos.

Definiciones Básicas

- Dada una función M tal que:

$$M : NB \rightarrow (\text{Literales} \cup NB \cup \text{IRIs})$$

y un grafo G , un grafo G' es una **instancia** de G si G' se obtiene de G cambiando algunos nodos blancos por sus correspondientes según M .

Ej:

Sea G el siguiente grafo:

```
hpw:Lucius v:padre _:h1.  
hpw:Narcissa v:padre _:h2.
```

Entonces G' es una instancia de G :

```
hpw:Lucius v:padre hpw:Draco.  
hpw:Narcissa v:padre hpw:Ron.
```

si $M(_ : h1) = hpw : Draco$ y $M(_ : h2) = hpw : Ron$

Definiciones Básicas

- Un **subgrafo** de G es un subconjunto de las ternas de G . Un **subgrafo propio** es un subconjunto de propio de las ternas de G .
- Una **instancia propia** de un grafo es una instancia en donde un nodo blanco ha sido reemplazado por un nombre, o dos nodos blancos han sido mapeados al mismo nodo.
- Un grafo G es **no redundante** (o **magro** - lean) si ningún subgrafo propio es una instancia de él.
- **Repaso:** Un **Recurso** es cualquier elemento referenciado por una IRI.

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- **Semántica de RDFS**
 - **Semántica Simple**
 - Semántica de Tipos de Datos
 - Semántica de RDF
 - Semántica de RDFS

Interpretación Simple

Es una estructura $I = \langle I_R, I_P, I_{Ext}, I_S, I_L \rangle$ donde:

- I_R es un conjunto de recursos llamado dominio o universo.
- I_P es un conjunto de propiedades.
- $I_{Ext} : I_P \rightarrow 2^{(I_R \times I_R)}$ es una función que dada una propiedad, devuelve el conjunto de parejas de recursos asociado. A esta función se le conoce como la **Extensión** de la propiedad.
- $I_S : IRI'S \rightarrow (I_R \cup I_P)$ es una función que dada una IRI devuelve el recurso o propiedad asociado.
- $I_L : Literales \mapsto I_R$ es una función parcial que para cada literal, devuelve el recurso asociado (en general, el mismo literal).

Interpretación Simple

Haciendo sobrecarga de notación, se define la denotación de un grafo base sobre una interpretación I , definiendo una función con igual nombre que la estructura:

Función de Interpretación

- $I(E) = I_L(E)$ si E es un Literal.
- $I(E) = I_S(E)$ si E es una IRI.
- $I(\langle s, p, o \rangle) = true$ si $I(p) \in I_P$ y $\langle I(s), I(o) \rangle \in I_{Ext}(p)$.
En otro caso $I(\langle s, p, o \rangle) = false$ $\langle s, p, o \rangle$ es una terna base.
- $I(E) = false$ si $\exists \langle s, p, o \rangle \in E. \neg I(\langle s, p, o \rangle)$. Donde E es un grafo base (sin nodos blancos).

Interpretación Simple: Nodos Blancos

- La definición anterior sólo funciona para ternas base y grafos base:
 - No hay definición de la interpretación de los nodos blancos.
 - La situación es similar a la de las variables en la Lógica de Primer Orden: Hay que agregar otras condiciones para saber cómo se entiende un grafo con nodos blancos.
- Para esto, dados I una interpretación simple, y $A : NB \mapsto I_R$, se define $(I + A) : I_R \cup I_P \cup NB \rightarrow I_R$ tal que:
 - $(I + A)(x) = I(x)$ si x es un nombre ($x \in I_R \cup I_P$)
 - $(I + A)(x) = A(x)$ si $x \in NB$.

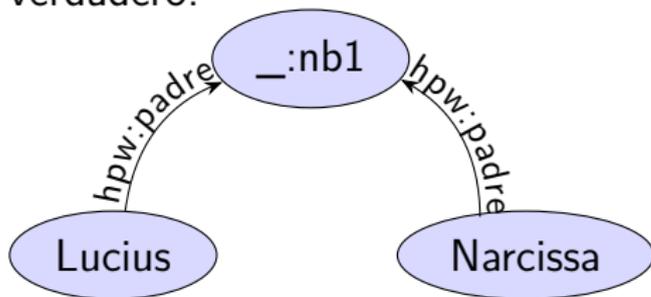
Interpretación de un Grafo

Interpretación de un Grafo RDF

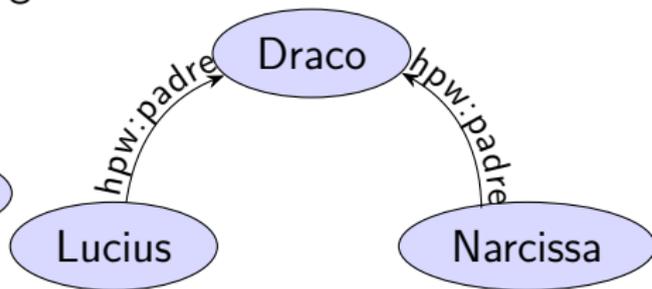
$I(E) = \exists A.(I + A)(E)$ (Si E es un grafo RDF cualquiera, $I(E)$ es verdadero si hay algún A tal que $(I + A)(E)$ es verdadero.)

Ej:

El siguiente grafo es verdadero:



porque existe A tal que $A(_ : nb1) = Draco$, con lo que se obtiene el siguiente grafo.



Satisfactibilidad (Simple)

- Una interpretación I **satisface (simplemente)** un grafo E si $I(E)$ es verdadero.
- Un grafo E es **(simplemente) satisfacible** si existe alguna interpretación I que los satisface. En otro caso es **(simplemente) insatisfacible**.

Consecuencia Simple

- Un grafo E es **consecuencia simple** de un grafo G si todas las interpretaciones que satisfacen simplemente a G también satisfacen a E . Esto se nota como $G \vDash E$.
- Si $G \vDash E$ y $E \vDash G$ entonces se dice que E y G son **lógicamente equivalentes**.
- Cualquier proceso que construya un grafo E a partir de un grafo G es **(simplemente) válido** si $G \vDash E$. En otro caso es **(simplemente) inválido**.

Propiedades de la Consecuencia Simple

- Cualquier grafo es simplemente satisfacible.
- $G \models E$ si y sólo si, un subgrafo de G es una instancia de E .
- El grafo vacío es verdadero y es consecuencia simple de cualquier otro grafo, pero ningún grafo distinto del vacío es consecuencia de él.
- Si E es un subgrafo de G , entonces $G \models E$.
- La Consecuencia Simple es monótona y compacta sobre los subgrafos.
 - Si S es un subgrafo de G y $S \models E$ entonces $G \models E$.
 - Si $G \models E$ y E finito, entonces hay un subgrafo finito $S \subseteq G$ tal que $S \models E$.

Comentarios de la Consecuencia Simple

- La noción consecuencia simple refleja únicamente la semántica del grafo como conjunto de ternas, en donde las ternas representan hechos.
- Es equivalente a una Lógica de Primer Orden, en donde se consideran sólo predicados binarios arbitrarios.
- Los nodos blancos, funcionan variables cuantificadas existencialmente con alcance a todo el grafo.
- Las interpretaciones simples, no consideran Tipos de Datos, ni tampoco el vocabulario de RDF o RDFS. Sólo consideran los conjuntos de ternas.

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- **Semántica de RDFS**
 - Semántica Simple
 - **Semántica de Tipos de Datos**
 - Semántica de RDF
 - Semántica de RDFS

Interpretaciones-D (D-Interpretations)

Las **Interpretaciones-D** consideran un conjunto de tipos de datos y los strings con idiomas.

Interpretación-D

Dado D un conjunto de IRIs (datatypes), una **interpretación-D** es una interpretación simple que cumple las siguientes condiciones:

- Si $\text{rdf} : \text{LangString} \in D$, $IL("s"@l) = \langle s, l' \rangle$ donde l' es l en minúscula.
- Para cualquier IRI $t \in D$, $I(t)$ es el tipo de datos (conjunto de valores) identificado por t , $I_L("s" \sim t) = L2V(I(t))(s)$ donde $L2V$ es tal que devuelve el valor de ese string s según ese tipo de datos t en caso que sea correcto. De lo contrario, está indefinido (string mal tipado).

Consecuencia-D (D-Entailment)

Consecuencias-D

- Un grafo es **D-satisfacible** o **satisfacible reconociendo D** si verdadero en alguna interpretación-D.
- $G \vDash_D E$ (E es **Consecuencia-D** de G) si toda interpretación-D que hace verdadero a G también hace verdadero a E

Propiedades de las Consecuencias-D

- Si una terna tiene como objeto un string mal tipado según un conjunto D de Tipos de Datos, entonces su interpretación es falso.
- Esto introduce la posibilidad de un un grafo sea insatisfacible o sea, inconsistente.
- Cualquier grafo es consecuencia de un grafo inconsistente.
- La consecuencia simple es lo mismo que consecuencia-D con $D = \emptyset$.

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- **Semántica de RDFS**
 - Semántica Simple
 - Semántica de Tipos de Datos
 - **Semántica de RDF**
 - Semántica de RDFS

Qué Falta?

- La “Semántica Simple” describe los grafos como conjuntos de hechos. Básicamente como una Lógica de Primer Orden con predicados binarios.
- La “Semántica de Tipos de Datos” agrega a la semántica simple, la forma de interpretar los tipos de datos y los literales de esos tipos.
- Hasta ahora, no se ha hablado de cómo se interpreta el vocabulario propio de RDF.

Interpretación RDF

Una **Interpretación RDF reconociendo D** es una interpretación-D en donde:

Interpretación RDF

- $\text{rdf} : \text{langString}, \text{xsd} : \text{string} \subseteq D$

- $x \in I_P$ si y sólo si

$$\langle x, I(\text{rdf} : \text{Property}) \rangle \in I_{Ext}(\text{rdf} : \text{type})$$

- Para cualquier tipo $t \in D$, $\langle x, l(t) \rangle \in I_{Ext}(\text{rdf} : \text{type})$ si y sólo si x está en el espacio de valores de $l(t)$.
- Además se cumplen las ternas axiomáticas correspondientes.

Ternas Axiomáticas RDF

```
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:nil rdf:type rdf:List .
rdf:_1 rdf:type rdf:Property .
rdf:_2 rdf:type rdf:Property .
... .
```

Consecuencia RDF

Consecuencia RDF

$G \models_{RDF-D} E$ (E es **consecuencia RDF reconociendo** D de G) cuando toda interpretación RDF reconociendo D que satisface G también satisface E .

Comentarios sobre la Consecuencia RDF

- Si $D = \{\text{rdf} : \text{langString}, \text{xsd} : \text{string}\}$ entonces se dice que $G \models_{RDF} E$ (E es consecuencia RDF de G).
- Las propiedades de la consecuencia simple no se aplican a la consecuencia RDF. Por ejemplo, las ternas axiomáticas son consecuencia del grafo vacío.

- En función de la definición de consecuencia, se pueden definir lo que W3C llama “patrones de Consecuencia”.

Reglas de Inferencia RDF

$$\frac{s \quad p \quad o \wedge t \text{ con } t \in D}{s \quad p \quad _ : n. \quad _ : n \quad \text{rdf} : \text{type} \quad t} \text{rdf}_1$$

$$\frac{s \quad p \quad o}{p \quad \text{rdf} : \text{type} \quad \text{rdf} : \text{Property}} \text{rdf}_2$$

- Interpretación, Modelos y Consecuencias
- Interpretaciones y Web Semántica
- **Semántica de RDFS**
 - Semántica Simple
 - Semántica de Tipos de Datos
 - Semántica de RDF
 - **Semántica de RDFS**

Semántica de RDFS

- La semántica se construye de la misma forma que la semántica de RDF:
 - Se define la interpretación. Se extiende las interpretaciones-D con los elementos necesarios para manejar clases.
 - Se agregan las condiciones semánticas necesarias para:
 - Definir el significado del vocabulario. (Ej: `rdfs:class`).
 - Definir la consecuencia.
 - En base a esos dos elementos, se construyen reglas de inferencia.

Interpretación RDFS (reconociendo D)

Es una estructura

$$I = \langle I_R, I_P, I_C, I_{CExt}, I_{Ext}, I_S, I_L, LV \rangle$$

donde:

- I_C que contiene todas las clases.
- $I_{CExt} : I_C \rightarrow 2^{I_R}$ que mapea las clases en sus extensiones.
- LV contiene los valores de los literales (no los literales).
- El resto conforma una interpretación simple que reconoce el conjunto de Tipos de Datos D .

Interpretación RDFS: Condiciones Semánticas

Interpretación RDFS: Clases

- $I_{CExt}(y) = \{x : \langle x, y \rangle \in I_{Ext}(I(\text{rdf} : \text{type}))\}$
- $I_C = I_{CExt}(I(\text{rdfs} : \text{Class}))$
- $LV = I_{CExt}(I(\text{rdfs} : \text{Literal}))$
- $I_{CExt}(\text{rdfs} : \text{Resource}) = I_R$
- $I_{CExt}(\text{rdfs} : \text{langString}) = \{I(E) : E \text{ es un string con especificación de lenguaje } \}$
- $\forall t \in D. I_{CExt}(I(t))$ es el espacio de valores de $I(t)$
- $\forall t \in D. I(t) \in I_{CExt}(I(\text{rdfs} : \text{Datatype}))$

Interpretación RDFS: Condiciones Semánticas

Interpretación RDFS: Propiedades

- $\langle x, y \rangle \in I_{Ext}(I(\text{rdf} : \text{domain})) \wedge \langle u, v \rangle \in I_{Ext}(x) \rightarrow u \in I_{CExt}(y)$
- $\langle x, y \rangle \in I_{Ext}(I(\text{rdf} : \text{range})) \wedge \langle u, v \rangle \in I_{Ext}(x) \rightarrow v \in I_{CExt}(y)$
- $I_{Ext}(I(\text{rdfs} : \text{subPropertyOf}))$ es transitiva y reflexiva.
- $\langle x, y \rangle \in I_{Ext}(I(\text{rdfs} : \text{subPropertyOf})) \rightarrow$
 $(x, y \subset I_P \wedge I_{Ext}(x) \subseteq I_{Ext}(y))$
- $x \in I_C \rightarrow \langle x, I(\text{rdfs} : \text{Resource}) \rangle \in I_{Ext}(I(\text{rdfs} : \text{subClassOf}))$
- $I_{Ext}(I(\text{rdfs} : \text{subClassOf}))$ es transitiva y reflexiva sobre I_C
- $\langle x, y \rangle \in I_{Ext}(I(\text{rdfs} : \text{subClassOf})) \rightarrow$
 $(x, y \subset I_C \wedge I_{CExt}(x) \subseteq I_{CExt}(y))$

Interpretación RDFS: Condiciones Semánticas

Interpretación RDFS: Propiedades

- $x \in I_{C_{Ext}}(I(\text{rdfs} : \text{ContainerMembershipProperty})) \rightarrow$
 $\langle x, I(\text{rdfs} : \text{member}) \rangle \in I_{Ext} I(\text{rdfs} : \text{subPropertyOf})$
- $x \in I_{C_{Ext}}(I(\text{rdfs} : \text{Datatype})) \rightarrow$
 $\langle x, I(\text{rdfs} : \text{Literal}) \rangle \in I_{Ext}(I(\text{rdfs} : \text{subClassOf}))$

Ternas Axiomáticas en RDFS

```
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 .
```

Ternas Axiomáticas en RDFS

```
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 .
```

Ternas Axiomáticas en RDFS

```
rdfs:comment rdfs:range rdfs:Literal .
rdfs:label rdfs:range rdfs:Literal .
rdf:value rdfs:range rdfs:Resource .
rdf:Alt rdfs:subClassOf rdfs:Container .
rdf:Bag rdfs:subClassOf rdfs:Container .
rdf:Seq rdfs:subClassOf rdfs:Container .
rdfs:ContainerMembershipProperty rdfs:subClassOf
    rdf:Property .
rdfs:isDefinedBy rdfs:subPropertyOf rdfs:seeAlso .
rdfs:Datatype rdfs:subClassOf rdfs:Class .
rdf:_1 rdf:type rdfs:ContainerMembershipProperty .
rdf:_1 rdfs:domain rdfs:Resource .
rdf:_1 rdfs:range rdfs:Resource .
...
```

Consecuencia RDF

$G \models_{RDFS-D} E$ (E es **consecuencia RDFS reconociendo** D de G) cuando toda interpretación RDFS reconociendo D que satisface G también satisface E .

Reglas de Inferencia RDF - Datatypes

- Si un término está en el conjunto de los tipos de datos, entonces está en la clase `rdfs:Datatype`.

$$\frac{s \in D}{s \text{ rdfs:type } rdfs:Datatype.} \text{ rdfs}_1$$

- Si una terna `s p o` es verdadera, y la propiedad `p` tiene dominio `c`, entonces `s` está en la clase `c`.

$$\frac{p \text{ rdfs:domain } c. \quad s \text{ p o.}}{s \text{ rdfs:type } c.} \text{ rdfs}_2$$

Reglas de Inferencia RDF - Resource

- Si una terna $s p o$ es verdadera, y la propiedad p tiene rango c , entonces o está en la clase c .

$$\frac{p \text{ rdfs : range } c. \quad s \ p \ o.}{o \text{ rdf : type } c} \text{ rdfs}_3$$

- Si la terna $s p o$ es verdadera, entonces s y o están en la clase `rdfs:Resource`

$$\frac{s \ p \ o.}{s \ \text{rdf : type} \ \text{rdfs : Resource}.} \text{ rdfs}_{4a}$$

$$\frac{s \ p \ o.}{o \ \text{rdf : type} \ \text{rdfs : Resource}.} \text{ rdfs}_{4b}$$

Reglas de Inferencia RDF - Subpropiedades

- La relación `rdfs:subPropertyOf` es transitiva y reflexiva.

$$\frac{p_1 \text{ rdfs : subPropertyOf } p_2. \quad p_2 \text{ rdfs : subPropertyOf } p.}{p_1 \text{ rdfs : subPropertyOf } p.} \text{ rdfs}_5$$

$$\frac{p \text{ rdf : type } \text{ rdf : Property.}}{p \text{ rdfs : subPropertyOf } p.} \text{ rdfs}_6$$

Reglas de Inferencia RDF - Subpropiedades

- Si p es una subpropiedad de q , y la terna $s \ p \ o$ es verdadera, entonces la terna $s \ q \ o$ es verdadera.

$p \text{ rdf} : \text{subPropertyOf} \ q.$

$s \ p \ o.$

$s \ q \ o.$ $\text{rdf} s_7$

Reglas de Inferencia RDF - Clases y subclases

- Si un término tiene tipo `rdf:Class` entonces es una subclase de `rdfs:Resource`

$$\frac{c \quad \textit{rdf} : \textit{type} \quad \textit{rdf} : \textit{Class}.}{c \quad \textit{rdfs} : \textit{subClass} \quad \textit{rdfs} : \textit{Resource}.} \quad \textit{rdfs}_8$$

- Si un término está en una subclase de una clase c , entonces también está en la clase c .

$$\frac{c_1 \quad \textit{rdf} : \textit{subclass} \quad c. \quad s \quad \textit{rdf} : \textit{type} \quad c_1.}{s \quad \textit{rdf} : \textit{type} \quad c.} \quad \textit{rdfs}_9$$

Reglas de Inferencia RDF - Clases y subclases

- La relación de subclase es reflexiva y transitiva

$$\frac{c_1 \quad rdfs:type \quad rdfs:Class.}{c_1 \quad rdfs:subClassOf \quad c_1.} \quad rdfs_{10}$$

$$\frac{c_1 \quad rdfs:subClassOf \quad c_2. \quad c_2 \quad rdfs:subClassOf \quad c.}{c_1 \quad rdfs:subClassOf \quad c.} \quad rdfs_{11}$$

Reglas de Inferencia RDF - Colecciones y literales

- Cualquier subpropiedad de `rdfs:member` está en la clase `rdfs:ContainerMembershipProperty`.

$$\frac{p \text{ rdfs : type rdfs : ContainerMembershipProperty.}}{p \text{ rdfs : subPropertyOf rdfs : member.}} \text{ rdfs}_{12}$$

- Cualquier instancia de un tipo de datos es un literal.

$$\frac{t \text{ rdfs : type rdfs : Datatype.}}{t \text{ rdf : subclass rdfs : Literal}} \text{ rdfs}_{13}$$

Bibliografía

- Cyganiak, R., , Wood, D. (2012). RDF 1.1 Concepts and Abstract Syntax (W3C Working Draft). W3C. Retrieved from <http://www.w3.org/TR/rdf11-concepts/>
- Hayes, P. y Patel-Schneider, P. 2014. RDF 1.1 Semantics. En: <http://www.w3.org/TR/2014/REC-rdf11-mt-20140225/>. W3C Recommendation. S.l.: W3C.
- Guha, R. y Brickley, D. 2014. RDF Schema 1.1. En: <http://www.w3.org/TR/2014/REC-rdf-schema-20140225/> [en línea]. W3C Recommendation. S.l.: W3C. Disponible en: <http://www.w3.org/TR/rdf-schema/>.
- Hitzler, P., Krötzsch, M., , Rudolph, S. (2009). *Foundations of Semantic Web Technologies*. Chapman , Hall/CRC.