

Ejemplos de axiomas

Con base la ontología del congreso de las transparencias Ontologías 1.

En **negrita** en lenguaje OWL-DL, en rojo en formato Protegé

Ejemplo 1

≤ 1 Título \equiv Presentacion Titulo max 1 equivalent to Presentacion

“No hay ninguna presentación con mas de 1 título”

Ejemplo 2

≥ 1 Título \equiv Presentacion Titulo min 1 equivalent to Presentacion

“No hay ninguna presentación con menos de 1 título”

Ejemplo 3

$(\leq 1 \text{ Título}) \cup (\geq 1 \text{ Título}) \equiv$ Presentacion

Titulo exactly 1 equivalent to Presentacion

“Cada presentación tiene un único título”

Ejemplo 4 “Cada presentación tiene un título distinto”

$T \leq 1.\text{Título}^{-1}$

Titulo inverse functional

Ejemplo 5 “Una sesión es del Tipo Posters cuando y solo cuando solamente se presentan Posters”

$\exists \text{Tipo}.\{\text{Posters}\} \equiv \forall P\text{-sesion}^{-1}.\text{Poster}$

(añadimos tiene_presentaciones como inverse of P-sesión)

Tipo value Posters equivalent to tiene_presentaciones only Posters

Nomenclatura para facilitar las expresiones:

a) Propiedades restringidas:

Si P es una propiedad y C una clase, " $P.C$ " será la subpropiedad de P definida por todas las ternas que tengan como segunda componente P y como tercera una instancia de la clase C y así " $\geq n P.C$ " será la clase de todos los elementos del dominio de P que por P esten relacionados con n o mas elementos de C . Por ejemplo

" ≥ 2 tieneMascota.Perro" será la clase de los que tienen 2 o mas perros como mascotas.

b) Número exacto:

Utilizaremos " $=n P$ " para abreviar " $(\geq n P) \cap (\leq n P)$ ". Por ejemplo

" $=2$ tieneMascota" será la clase de los que tienen exactamente 2 mascotas

" $=2$ tieneMascota.Perro" será la clase los que tienen exactamente 2 perros como mascota (aunque podrían tener otras mascotas que no fuesen perros)

Ejemplo 5

“ $\forall P\text{-sesion}^{-1}.\text{Poster}$ ” es la clase de las sesiones (sesion es dominio de $P\text{-sesion}^{-1}$) que solo tienen presentaciones (tener presentación es una forma de expresar $P\text{-sesion}^{-1}$) de la clase Poster o no tienen ninguna presentación.

Por tanto, “cada sesión consistirá en una conferencia invitada, o la presentación de 3 artículos o bien la presentación de un número indeterminado de posters” se puede expresar:

$\text{Sesion} \subseteq (\exists P\text{-sesion}^{-1}.\text{Presentacion}) \cap$

$(\forall P\text{-sesion}^{-1}.\text{Poster}$

$\cup (\forall P\text{-sesion}^{-1}.\text{Articulo_aceptado} \cap =3 P\text{-sesion}^{-1}.\text{Articulo_aceptado})$

$\cup (\forall P\text{-sesion}^{-1}.\text{Conferencia_invitada} \cap =1 P\text{-sesion}^{-1}.\text{Conferencia_Invitada})$

)

(recordemos que hemos añadido tiene_presentaciones como inverse of P-sesión)

Sesion subclass of (tiene_presentaciones some Presentacion) and (

tiene_presentaciones only Poster or

(tiene_presentaciones only Articulo_aceptado and tiene_presentaciones exactly 3 Articulo_aceptado) or

(tiene_presentaciones only Conferencia_invitado and tiene_presentaciones exactly 1 Conferencia_invitada)

)