

UNIVERSIDAD DE ALCALÁ

Departamento de Automática

Grado en Ingeniería Informática Grado en Ingeniería de Computadores

Práctica 3. Parte optativa: Creación de ontología *parentesco.owl*

Técnicas de diseño de sistemas de supervisión y entrenamiento remoto

Índice

1. Desarrollo de la ontología parentesco.owl	3
--	---

1. Desarrollo de la ontología parentesco.owl

Como parte optativa de la práctica 3 se pide desarrollar la siguiente ontología `parentesco.owl`. Utilice el razonador *FaCT++* en Protégé con el fin de inferir datos de la ontología cuando sea necesario para responder a las preguntas que se van planteando a medida que construye la ontología:

1. Introduzca la siguiente taxonomía:

Sexo. Introduzca ya las instancias de Sexo: Masculino y Femenino.

Persona

Padre subclase de **Persona**

Madre subclase de **Persona**

Hijo subclase de **Persona**

Hija subclase de **Persona**

Progenitor subclase de **Persona**

Descendiente subclase de **Persona**

Hermano subclase de **Persona** (se refiere a hermano o hermana)

2. Propiedades de Persona:

- Propiedades de datos: `nombreYapellidos` y `fechaNacimiento` (decida el tipo, dominio, y tipo de cada propiedad).
- Propiedades de objeto: `tieneDescendiente`, `tieneHermano` (simétrica), `tieneProgenitor`, `tieneSexo`. Piense el tipo de propiedades que son (funcionales o no), sus dominios y rangos, etc.
- Introduzca ahora la propiedad `tieneAbuelo`: añada la siguiente expresión como *SubProperty Of (Chain)* de `tieneAbuelo`: `tieneProgenitor o tieneProgenitor`. Esto significa que si "X tieneProgenitor Y e Y tieneProgenitor Z, entonces X tieneAbuelo Z" (X tiene abuelo/abuela Z).

Pregunta 1: ¿Porqué `tieneHermano` es una propiedad simétrica? ¿Y `tieneDescendiente` y `tieneProgenitor`, porqué son inversas la una de la otra?

3. Clases *equivalentes*:

Padre: cualquier persona que tiene un hijo-a (descendiente) y sexo masculino (*Equivalent To: Persona and tieneDescendiente some Persona and tieneSexo value Masculino*).

Madre: similar al anterior.

Progenitor: persona que tiene un hijo (*Equivalent To: Persona and tieneDescendiente some Persona*).

Hijo: es una persona que tiene progenitor y tiene sexo masculino (*Equivalent To: Persona and tieneProgenitor some Persona and tieneSexo value Masculino*).

Hija: similar al anterior.

Descendiente: persona que tiene progenitor (*Equivalent To: Persona and tieneProgenitor some Persona*).

Hermano: persona que tiene algún hermano-a (*Equivalent To: Persona and tieneHermano some Persona*).

Pregunta 2: Ejecute el razonador: *Fact++*. ¿Qué sucede?

4. **Individuos:** Cree las instancias de `parentesco.owl` de acuerdo a la jerarquía padre-hijo mostrada en la Figura 1.

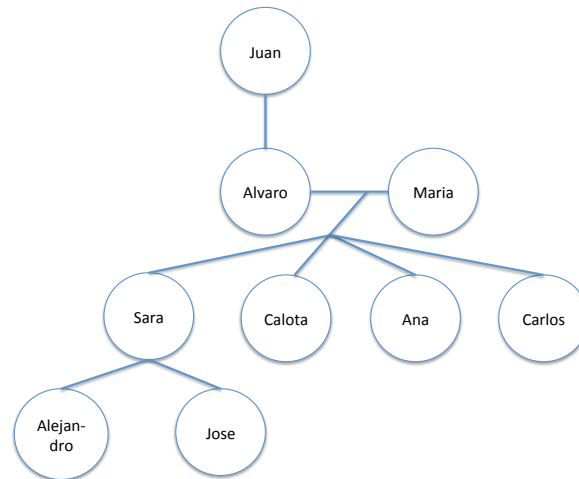


Figura 1: Instancias de la ontología `parentesco.owl`

Por ejemplo, introduzca **Sara** y con propiedades: `tieneProgenitor Alvaro`, `tieneProgenitor Maria`, `tieneSexo Femenino`, `tieneHermano Carlota`. Introduzca a continuación la instancia **Alejandro** de la misma forma, con la propiedad `tieneHermano Jose` de acuerdo a la jerarquía de la Figura 1.

Pregunta 3: ¿Porqué no es necesario definir el valor de `tieneDescendiente` de la instancia **Maria** con valor **Sara** o que no es necesario definir **Alejandro** hijo de **Sara**?

Introduzca ahora las instancias **Ana** y **Carlos**, con los valores adecuados en la propiedad `tieneProgenitor` de ambos y en la propiedad `tieneHermano` de ambos, introducir **Sara** y **Carlota**.

Pregunta 4: Ejecute el razonador FaCT++. ¿Qué hechos se han inferido en la instancia de **Sara**? ¿Porqué?

Pregunta 5: Añada ahora la instancia de **Persona** denominada **Miguel**, hermano de **Alvaro** e hijo de **Juan**. Ejecute de nuevo el razonador. ¿Qué se ha inferido para la instancia **Juan** y para la instancia **Alvaro**? ¿Es lógicamente correcto?