Trabajo Práctico 2: Ontologías

Introducción

En este proyecto nos dedicaremos a plantear, especificar y modelar una ontología a elección dentro de las propuestas dadas, y luego razonar sobre la misma (en el sentido computacional).
Les pediremos el modelado en formato OWL y un breve informe indicando puntos clave para la comprensión de la misma.

¿Qué enviar?

- archivo .owl que pueda ser abierto por Protégé e instrucciones adicionales para configurar el entorno para poder abrir y utilizar la ontología,
- archivo .pdf o similar con informe de la ontología que contenga:
    - información específica sobre el dominio seleccionado: limites, alcances, objetivos,
    - la/s fuente/s de donde se obtuvo información para la representación del dominio,
    - aclarar si se tomó inspiración o si se consideraron otros modelos o patrones de diseño ontológicos,
    - el listado de conceptos representados con una breve descripción de cada uno,

Trabajo Práctico 2: Ontologías

- el diagrama de la ontología con todas las clases, las relaciones de “subclass of”, las principales propiedades de objeto y de datos, y las restricciones más importantes (considerar el uso del toolbox “Graffo” o “OWLViz” que vienen por defecto en Protégé),
- un esquema de las instancias propuestas a modo de ejemplo y sus relaciones,
- las consultas realizadas con su formulación en lenguaje natural y en la sintaxis de “DL Query” (o SPARQL si lo prefiere, siempre aclarar el razonador elegido),
- un breve análisis del razonamiento que sigue el razonador para responder las queries planteadas, si fue posible responderlas, o por qué no le fue posible responderlas,
- una breve conclusión destacando si surgieron diferentes posturas en la conceptualización, los inconvenientes encontrados al realizar la especificación, y si hay cosas que no pudieron ser correctamente representadas y por qué.

Nota: para las descripciones y consideraciones de los conceptos representados se pueden añadir “annotation properties” a los elementos, en tal caso mencionar que se realizó esto en el informe.

Pasos a Seguir

1. Elegir uno entre los dominios propuestos para representar.
2. Siga las pautas de la metodología de desarrollo ontologías vista en clase. Plantee un conjunto básico de preguntas de competencia (e.g.: ¿Qué compone a una Pizza?, ¿Qué toppings son usuales en una Pizza Margherita? ¿Cómo se define una Pizza Vegetariana?)
3. Realizar el modelado conceptual y la especificación del dominio en Protege. Es decir, defina conceptos/clases, propiedades que se necesitan representar, restricciones, etc.
   Considerar al menos:
   a. un par de clases disjuntas
   b. al menos una clase definida
   c. algunas propiedades con subpropiedades y propiedades inversas
4. Realizar la instanciación y evaluación de consistencia. La cantidad de instancias debe ser de al menos 3 en la mayoría de las clases principales. Por otro lado, **algunas instancias no deben tener todos los datos cargados de manera de poder inferir conocimiento a partir del modelo planteado.**
5. Experimente con el razonar utilizando queries de diferente nivel de complejidad, y/o utilizando diferentes razonadores.

Nota: considere explorar otras opciones dentro de Protegé si le da el tiempo, y buscar cuál es su sustento teórico dentro de la Description Logic, por ejemplo, las relaciones ternarias. Recuerde que se está formando a sí mismo para luego saber aplicar los conocimientos relevantes de la IA a problemas futuros.

Trabajo Práctico 2: Ontologías

Propuestas de Ontologías

Lenguajes de Programación

Existen numerosos lenguajes de programación, cada uno con propiedades y casos de uso diversos. Según el problema que se quiera resolver, y las características que posee, será más o menos conveniente utilizar un lenguaje de programación.
Se pide modelar una ontología de lenguajes de programación que contenga distintos lenguajes y su relación con características como son el sistema de tipado, el nivel de abstracción en que se programa, y demás paradigmas en los que se basa. La idea final es que pueda ayudar a un programador a elegir un lenguaje adecuado para el proyecto que quiere llevar a cabo con el mismo.
Nos interesa además incluir algunos paquetes de software que hayan sido desarrollados con éstos lenguajes.
Se plantean las siguientes queries a resolver:

- ¿Qué lenguajes funcionales tienen tipado estático y fuerte?
- ¿Qué lenguajes fueron utilizados para realizar paquetes de software que hacen uso de paralelismo?
- ¿Cuáles programas fueron escritos en más de un lenguaje, donde alguno de los lenguajes es de tipado dinámico?

seleccione una de éstas para resolver y plantee al menos dos propias.

Librerías de Python

Python posee una rica y variada gama de librerías indexadas que son de fácil acceso y distribución.
Queremos generar una ontología que nos permita modelar distintas características de las librerías, como son la licencia que posee, el lenguaje en que están escritas originalmente (algunas librerías son wrappers de librerías en otros lenguajes), el status de desarrollo en que se encuentra (puede ser alpha, stable…), y si utiliza hints de tipado o no. Además se busca entender cuando una librería es utilizada por otra. Y finalmente nos interesa clasificarlas por la audiencia a la que sirven, por ejemplo si son librerías de tipo matemática, financiera, médica, etc.
Nos interesaría además incluir algunos paquetes de software que hagan uso de algunas de éstas librerías para modelar sus dependencias.
Se plantean las siguientes queries a resolver:

- ¿Qué paquetes de software hacen uso de una o más librerías con licencias permisivas?
- ¿Qué librerías de licencias privativas hacen uso de una o más librerías con licencias permisivas?
- ¿Cuáles paquetes de software están desarrollados con más de dos librerías que tienen relación con la industria médica?

seleccione una de éstas para resolver y plantee al menos dos propias.

Trabajo Práctico 2: Ontologías

Kernel Linux

Queremos modelar los componentes, relaciones y funcionamiento de un kernel Linux. Un sistema operativo se compone de un Kernel, que provee conceptos de servicios, módulos de seguridad, un planificador de tareas, acceso a red, sistemas de archivos, y acceso a dispositivos de hardware. Por su parte, cada una de estas partes se subdivide según sus usos más particulares. Por ejemplo, algunos servicios que corren sin interfaz suelen llamarse “daemons”, o los planificadores de tareas pueden requerir que los servicios o tareas estén asociados a una prioridad dada.
Además nos interesaría modelar una realización de un sistema particular, en donde existen ciertos usuarios que están corriendo ciertos procesos con archivos determinados abiertos y teniendo permisos específicos.
Pedimos que se modele un subconjunto suficientemente expresivo como para resolver queries tales como:

- ¿Qué servicios de Linux que requieran escritura de archivos tienen además acceso a la red?
- ¿Qué servicios que acceden a la terminal también están interactuando con al menos dos dispositivos de hardware?
- ¿Qué usuario tiene corriendo un servicio de terminal pero no tiene permisos para escribir en root?

seleccione una de éstas para resolver y plantee al menos dos propias.

Sistema Recomendador de Películas (Última Opción)

Queremos modelar el universo del cine, teniendo como objetivo general ayudar a un usuario a elegir una película. En este caso nuestra ontología debería por lo menos separar temas y orígenes, incluir información sobre actores y directores y mucha otra información relevante. Una posibilidad es acotarse a un único género (animé, sci-fi, etc.) pero ser mucho más específico en los conceptos. (Aclaración: consideremos que en las películas animadas de todo tipo los actores son los que hacen las voces).
Se plantean las siguientes queries a resolver:

- ¿Qué película italiana tiene a un director y a una actriz premiados?
- ¿Hay películas realizadas en más de un país calificadas con 5 estrellas?
- ¿Qué película de terror comparte director con una película con un actor/actriz español?

seleccione una de éstas para resolver y plantee al menos dos propias.

Trabajo Práctico 2: Ontologías

Protégé

Ya se realizó una breve introducción a Protégé en clase, sin embargo proveemos links con recursos de utilidad por si se requieren:

- Protégé: https://protege.stanford.edu/
  - recuerde que puede instalar el razonador Pellet que recomendamos en “File > Check for plugins” y seleccionando Pellet, luego podrá encontrarlo en “Reasoner”,
  - recuerde activar todos los ítems de razonamiento o inferencia en “File > Preferences > Reasoner”,
  - puede visualizar su ontología con “OWLViz” u “OntoViz” yendo a “Window > Views > Ontology Views” y eligiendo el visualizador de preferencia,
  - puede visualizar el creador de consultas yendo a “Window > Views > Query Views” y seleccionando DL Query o SPARQL.
- Documentación de Protégé: http://protegeproject.github.io/protege/
- Ejemplo de Pizzas / Tutorial (Michael DeBellis): https://github.com/yasenstar/protege_pizza (recuerden mirar el .pdf que se encuentra entre sus archivos, también tiene videos de YouTube, este tutorial contiene información sobre queries DL y SPARQL)
- Ejemplo de Pizzas / Tutorial (Matthew Horris): https://mariaiulianadascalu.files.wordpress.com/2014/02/owl-cs-manchester-ac-uk_-eowltutorialp4_v1_3.pdf (este no viene con código asociado)
- Documentación de DL Query: https://protegewiki.stanford.edu/wiki/DLQueryTab
- Tutorial de SPARQL simplificado: https://sachipcranasinghe.medium.com/intro-to-sparql-part1-103957d8e854 (ver partes 1, 2 y 3)
- Video de queries SPARQL: https://www.youtube.com/watch?v=0zUos1zWB5k