Diseño y Construcción de Ontologías

Razonamiento Lógico

Dra. Maricela Bravo
mcbc@correo.azc.uam.mx

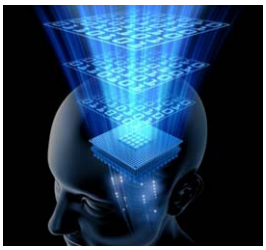
UNIVERSIDAD AUTONOMA METROPOLITANA
Casa abierta al tiempo Azcapotzalco

CONTENIDO

1. Razonadores
2. Tareas de razonamiento
   a. Consistencia lógica
   b. Satisfactibilidad
   c. Clasificación
   d. Realización

Razonamiento

- Los razonadores son aplicaciones computacionales que permiten realizar inferencias y generar conocimiento a partir de un conjunto de axiomas y hechos.
- Los razonadores utilizan un motor de inferencias y un conjunto de reglas expresadas en un lenguaje semántico como Web Ontology Language (OWL).
- Ejemplo:
  - FACT++
  - Racer
  - Pellet
  - Flora-2
  - KAON-2



3

Pellet

- Pellet soporta el razonamiento con la variante de OWL-DL (SHOIN) y ha sido extendido para soportar la especificación de OWL 2 (SROIQ).
- Pellet proporciona todos los servicios de inferencia tradicionales:
  - Chequeo de consistencia lógica
  - Satisfactibilidad de Conceptos
  - Clasificación
  - Realización

Tareas de razonamiento

- **Chequeo de consistencia lógica**
  - Asegura que una ontología no contenga ningún hecho contradictorio.
- **Satisfactibilidad de Conceptos**
  - Determina si es posible que una clase tenga instancias. Si una clase es insatisfactible, entonces definir una instancia de dicha clase causará que toda la ontología sea inconsistente.

Tareas de razonamiento

- **Clasificación**
  - Computa las relaciones de subclases entre cada clase nombrada para crear la jerarquía de clases completa. La jerarquía de clases puede ser usada para responder preguntas tales como obtener todas o solamente las clases directas de las subclases.

Tareas de razonamiento

- **Realización**
  - Encuentra las clases más específicas a las que un individuo pertenece.
  - La realización computa los tipos directos (pertenencia de clases) por cada individuo.
  - La realización solo puede hacerse después de la clasificación, ya que los tipos directos se definen con respecto a la jerarquía de clases.
  - Usar la jerarquía clasificada también permite obtener todos los tipos para cada individuo.

Arquitectura de Pellet

- El núcleo del razonador Pellet es el algoritmo Tableaux que verifica la consistencia de la base de conocimiento, formado por el par de un ABox y un TBox.
- El razonador incluye un verificador de tipos de datos que puede checar la consistencia de las conjunciones (construidas o derivadas) de los tipos de datos simples del esquema XML.

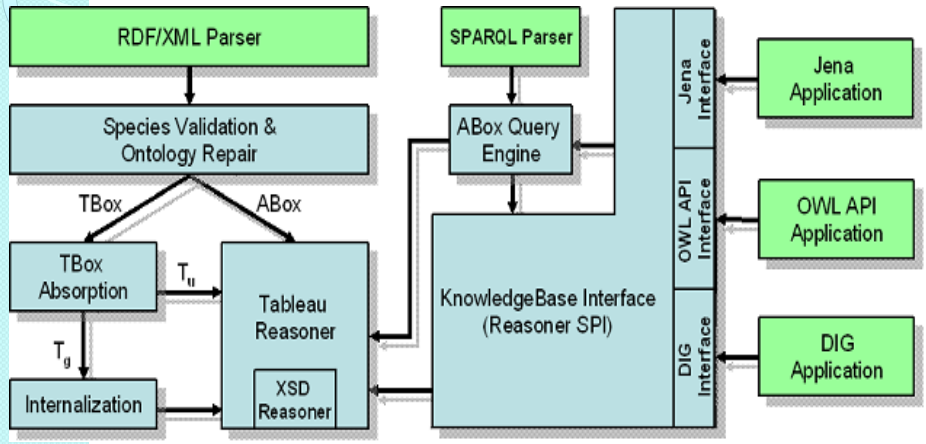
8

Arquitectura de Pellet

- Durante la fase de carga, los axiomas acerca de las clases se colocan en el componente TBox y las aserciones acerca de los individuos son almacenadas en el componente ABox.
- Los axiomas del componente TBox pasan por el pre-procesamiento de los razonadores DL antes de que sean alimentados en el razonador Tableaux.

9

Arquitectura de Pellet



10

Arquitectura de Pellet

- La funcionalidad de un razonador Tableau es checar la satisfactibilidad de un ABox con respecto al TBox.
- Las demás tareas de razonamiento se reducen al chequeo de consistencia lógica con una transformación apropiada.
- La interface de programación del sistema de Pellet proporciona funciones genéricas para manejar tales transformaciones.

11

Ejercicio

- Utilizando la API de Pellet
- Diseñar y desarrollar una clase en Java que ofrezca los siguientes métodos para el razonamiento con Pellet:
  - Consistencia
  - Clasificación
  - Realización
- Probar la clase con:
  - 3 ontologías públicas
  - Todas las ontologías del curso

12

¿Preguntas?

13