Domande esame mod sem

# Reti semantiche

1. **Cos’è una rete semantica?**Si tratta di un grafo composto da nodi e archi, dove ai nodi corrispondono i concetti e agli archi le proprietà che mettono in relazione questi concetti. Permette di creare una KB di semplice visualizzazione. Per fare inferenza in una rete semantica basterà seguire un percorso del grafo. Ha un’espressività minore rispetto alla logica del prim’ordine
2. **Cos’è una rete semantica proposizionale?**Si tratta di una rete semantica in cui i nodi, oltre a poter essere dei concetti, possono essere anche delle **proposizioni**. Ciò permette di rappresentare conoscenze soggettive, andando ad introdurre un livello epistemico alla rete. Possiamo dire, ad esempio, che un certo individuo è membro di una certa classe.  
   Ci è permesso usare connettivi, quantificatori, variabili, … così da poter avere la stessa espressività della logica del prim’ordine.  
   PROBLEMA: troppa espressività può rendere impossibile fare inferenza, quindi conviene limitarla un po’
3. **Cos’è SNEPS?**  
   Si tratta di una rete semantica proposizionale dotata di un motore di ragionamento, basato su

* Formule
* Slot
* Percorsi

Gli archi di questa rete possono essere

* **ISA**, indica che un individuo appartiene ad una certa classe OPPURE che una certa classe è sottoclasse di un’altra
* **INSTANCE\_OF**, indica che un individuo è istanza di una certa classe

La conoscenza può contraddirsi (es. pinguini sono uccelli che non volano MA gli uccelli volano), per risolvere il problema si usano le **default rules**, utilizzabile attraverso una logica non monotona (la **default logic**). “Le conoscenze specificate localmente in un certo nodo prevalgono su quelle ereditate”, così facendo possiamo dire che un pinguino non vola, anche se eredita la caratteristica “vola”.

In Sneps è possibile fare 3 tipi di inferenze:

* **Riduzione**, da un grafo dedurne uno più piccolo
* **Inferenze** **basate** **sui** **percorsi**, un percorso fatto di certe relazioni corrisponde ad una relazione (simile alle property chain)
* **Regole**, inserite in modo arbitrario dall’utente

1. **Leggere e spiegare una rappresentazione del sistema SNEPS e dire cosa significa**

Immagine che contiene testo

Descrizione generata automaticamente

1. **Cos’è la teoria dei frame?**

È un’evoluzione delle reti semantiche per rappresentare la conoscenza di tipo stereotipato. Un frame rappresenta una situazione stereotipata (es. essere in salotto, andare al compleanno, …) ed è formato da

* Livello superiore, che descrive la situazione in generale
* Livello inferiore, formato da diversi **slot** con all’interno specifiche istanze per rappresentare una situazione specifica (in uno slot si può inserire un frame, andando a formare un **sistema di frame**)

# RDF, RDFS e OWL

1. **Come è strutturata tripla RDF**

Una tripla RDF è una tripla del tipo <risorsa, proprietà, valore> (o <entità, attributo, valore>) e serve per definire delle relazioni. Si tratta di un modello basato sui grafi che ci permette di vedere risorse e valori come nodi e proprietà come archi.

1. **Differenze tra RDF e RDFS**

* **RDF**:   
  è il data model che ci dice come organizzare e strutturare i dati. Può essere visto come un grafo relazionale composto da nodi (risorse o entità) e archi che connettono i nodi. È un linguaggio utilizzato per descrivere risorse come documenti, persone, oggetti…

È composto da diversi concetti fondamentali, quali:

* + **Risorse**, cosa di cui si vuole parlare (casa, autore, …) a cui viene assegnato un IRI derefernziabile
  + **Proprietà**, risorsa che descrive la relazione tra risorse (es. “amico di”, “scritto da”, …)
  + **Statement**, è una tripla risorsa-proprietà-valore, dove il valore può essere una risorsa o un letterale (es. “Michael” “AmicoDi” “Marco”)
  + **Grafi**, struttura che ci permette di rappresentare graficamente gli statement

Per la sintassi di RDF si possono utilizzare diversi linguaggi, tra cui Turtle

* **RDFS**:   
  è lo strumento utilizzato per attribuire semantica alle nostre risorse (definendo cosa significano). Con RDFS è possibile definire classi e attribuire una gerarchia delle classi**.** Classi e individui sono messi in relazione mediante rdf:type. Grazie a ciò è possibile apporre delle restrizioni sul nostro dominio e permettere di effettuare del ragionamento automatico. Il concetto di classi e proprietà è però diverso dall’OO, in quest’ultimo si creano le classi e si definiscono le sue proprietà, in RDFS si creano le proprietà e le si associano alle classi (in questo caso modificando una proprietà non si modificherà la classe). Come per le classi anche per le proprietà è possibile definire una gerarchia. Le proprietà vengono definite attraverso domain e range.

PS. RDFS è un primitive ontology language

PS2. RDFS è property-centric

1. **Cosa s'intende per vocabolario RDF ed ontologia OWL**

* **Vocabolari RDF:**La definizione di un insieme di classi e proprietà costituisce un vocabolario. Esso viene pubblicato con un certo IRI per far sì che possa essere accessibile a tutti. Noi possiamo usare uno o più vocabolari esterni per descrivere i dati. Alcuni dei vocabolari più usati sono:
  + **FOAF**, vocabolario per descrivere reti sociali (contiene core, molto completo, e social web, un po' più obsoleto)
  + **Dublin** **core**, vocabolario che si occupa di descrivere qualsiasi materiale digitale (video, immagini, pagine Web, ecc.), nonché risorse fisiche, come libri o CD, e oggetti come opere d'arte (Europeana si basa su DC, ed è l’archivio europeo dei beni culturali)
  + **Schema.org**, utilizzato per mantenere dati strutturati riguardo al web
  + **SKOS**, usato per descrivere e allineare terminologie diverse

servono per unificare ontologie diverse in modo che siano interoperabili. Così non è necessario inventare ogni volta nuovi termini per informazioni che così sono facili da reperire e comprendere.

* **OWL**:  
  è un linguaggio che ci permette di creare ontologie. RDF e RDFS insieme permettono di creare triple e di definire gerarchie di classi e proprietà, ma per esprimere conoscenza più avanzata necessitiamo di OWL (es. vogliamo dire che una persona può essere solo maschio o femmina). È un linguaggio che si basa sulle **logiche descrittive** per esprimere conoscenza. Una ontologia è una rappresentazione astratta di concetti e delle loro relazioni. Le ontologie formali, descritte attraverso un formalismo di rappresentazione, permettono di fare inferenza.

Esistono 3 tipi di ontologie:

* + **Top-level**, definiscono concetti fondazionali come spazio, tempo, …
  + **Middle-level**, definiscono concetti generali ma non fondazionali (es. organizzazioni, comunicazione, …)
  + **Domain ontology**, definiscono un dominio specifico

Ci sono tantissimi progetti di ontologie, tra cui:

* + **CYC**, uno dei primi tentativi di creare conoscenza di senso comune, è suddivisa in microteorie, cioè in molti dominii diversi; le microteorie permettono di fare asserzioni su una sola di loro, così da non andare in contrasto con altra conoscenza
  + **SUMO**, ontologia che si basa sulla logica del prim’ordine. È un’ontologia che incorpora altre ontologie esterne; contiene gerarchia delle classi e relazioni tra di esse. Il lessico di sumo e orientato al lessico di WordNet
  + **YAGO**, ottenuto mediante l’estrazione automatica di concetti dai testi
  + **YAGOSUMO**, incorpora conoscenza di YAGO (che era basato su Wikipedia e WordNet) e SUMO
  + **YAGO2**, è YAGO a cui viene aggiunto GeoNames (conoscenza geografica)
  + **Dbpedia**, estrae info strutturata da Wikipedia
  + **Wikidata**

1. **Wikidata**

Mette i dati in uno schema unico per tutti. Prende le informazioni da wikipedia e wikimedia. Molte informazioni sono prese da sorgenti di conoscenza, mentre altre sono scritte a mano. Ogni pagina segue un template, in cui avrà, titolo, descrizione, proprietà, id univico, …

1. **DBPedia**Progetto che si occupa di estrarre informazione struttura da wikipedia, andando a pubblicarle online in una LDP in formato RDF. Usa il meccanismo di **crowdsurcing** (chiunque può modificare le pagine).

I dati vengono estratti dall’infobox di wikipedia, il problema è che la infobox cambia in baso allo stato.  
Viene effettuato dalla infobox il mapping tra classi e proprietà. Sono stati utilizzati dei pattern per fare ciò.

Metodologia:

* L’URI della pagina wikipedia diventa il soggetto
* Il soggetto viene concatenato con le proprietà dell’infbox
* Il valore dell’attributo viene collegato alla proprietà, cosi da avere una tripla ben formata.

1. **OmekaS**

LDP per la pubblicazione di collezioni di contenuti digitali.

1. **Regole R-to-R**

Si tratta del mapping che viene fatto per passare da un database relazionale a un formato RDF. Per farlo ci si avvale di OnTop. Il linguaggio usato per il mapping e R2RML.

1. **Cosa sono le logiche descrittive**Le **logiche** **descrittive** sono una famiglia di formalismi utilizzati per rappresentare la conoscenza in un dominio di applicazione detto **mondo**.   
   In primo luogo, sono definiti i concetti rilevanti per quel dominio e, di seguito, utilizzando questi concetti si specificano le proprietà degli oggetti e degli individui appartenenti al dominio.  
   L’idea è usare dei formalismi per permettere di fare inferenza sulla KB. L’obiettivo è ricavare conoscenza implicita da una conoscenza esplicita. La logica dovrà essere tale da permettere di avere KB **decidibili** e **trattabili**.  
   la logica descrittiva ha queste due caratteristiche

* **Non unicità del nome**
* **Assunzione di mondo aperto**

1. **Quali sono i requisiti per un ontology language?**  
   sono:

* Sintassi ben definita
* Semantica formale
* Sufficiente potere espressivo
* Convenienza delle espressioni
* Supporto di ragionamento efficiente, ci permette di
  + Verificare la consistenza dell’ontologia
  + Verificare le relazioni indesiderate tra classi
  + Verificare la presenza di classificazioni non intenzionali di istanze

1. **Quali sono i linguaggi dello stack del web semantico**I linguaggi dello stack del web semantico sono:

* **RDF**: serve per descrivere risorse (documenti, persone, oggetti...)
* **RDFS**: permette di descrivere relazioni tra risorse
* **OWL**: permette di descrivere ontologie computazionali. Con esso si possono descrivere relazioni complesse.

1. **Elementi di un'ontologia OWL**Le ontologie sono formate da

* Classi
* Proprietà
* Assiomi
* Individui
* Letterali.

Le entità possono essere individui, classi, proprietà (object property, data property), mentre gli assiomi possono essere sulle classi, dichiarazioni, ecc.

1. **OWL vs OWL2**

OWL 1 aveva dei profili diversi e meno funzionalità. Vediamo le funzionalità aggiunte in OWL 2:

* **Property chain**
* **Data types più ricchi, data ranges**
* **Qualified cardinality restriction**
* **Proprietà asimmetriche, riflessive, disgiunte**
* **Enhance annotation capabilities**
* **Keys** (sistema di chiavi)

Profili di OWL 2, ognuno dei quali consiste nel ritaglio di una parte specifica di OWL:

* **EL**, progettato per grandi ontologie (ontologie di ambito medico, tesauri, …)
  + Limitazioni: no negazione e disgiunzione, no quantificazione universale sulle proprietà e no proprietà inverse
* **QL** (query language), concepito per l’interazione con le basi di dati relazionali
  + Limitazioni: no quantificazione universale e no property chain
* **RL** (rule language), concepito per applicazioni che richiedono l’uso di ragionamento automatico
  + Limitazioni: vieta specifici tipi di espressione (ad esempio, non si può dire che ogni persona ha un genitore)

Non sempre è ben specificato a quale profilo appartiene l’ontologia, spesso per rendere l’ontologia conforme a più profili si aggiungono dominii multipli, ad esempio, per gestire i vari profili

1. **Perché può capitare che la gerarchia di classi inferita sia diversa da quella asserita?**Può capitare che il reasoner trovi classi completamente sussunte da altre classi, ciò vale a dire che l'insieme degli Individui che apparterrebbero ad una classe è totalmente incluso anche in quello di un'altra classe.
2. **Come vengono rappresentati i blank nodes in turtle, e cosa succede ad essi quando il file viene caricato su una LDP e viene creato il knowledge graph**  
   I blank nodes in Turtle sono identificati da una coppia di parentesi quadre. I predicati associati ad un blank node sono scritti al suo interno. Quando vengono caricati in una LDP gli viene associato un nome e l'estensione della LDP, ma vengono mantenuti come blank nodes.
3. **Cos'è un named graph**Si tratta di un grafo di una rete semantica a cui è stato dato un nome. In Turtle si usano i prefissi. In questo modo è possibile utilizzare più reti provenienti da fonti differenti.
4. **Cos'è il default graph**Il default graph è una rete semantica a cui non è stato assegnato alcun nome.
5. **Cosa s'intende per quadrupla**La quadrupla è un'estensione della tripla <Soggetto, Predicato, Oggetto> a cui è aggiunto un quarto valore che identifica a quale grafo appartiene la tripla.

# Terminologia OWL

1. **Assioma di sottoclasse comporta relazione tra due class expression, che nel caso più semplice possono essere due classi**
2. **Cosa s'intende per T-Box e A-Box**La T-BOX (terminology box) è tutto l'insieme dei termini che rappresentano la parte concettuale di un'ontologia. L'A-BOX invece (assertion box) è l'insieme delle asserzioni effettuate sugli individui.
3. **Cos'è una classe definita**Una classe definita è un particolare tipo di classe che viene definito come equivalente ad un insieme di restrizioni. Le restrizioni sono le condizioni necessarie e sufficienti per l'appartenenza alla classe. Riducendo l'insieme degli individui a solo quelli che soddisfano le restrizioni, possiamo effettuare alcune forme di ragionamento automatico.
4. **Cos'è una class expression**Le class expressions identificano insiemi di individui grazie alla definizione di condizioni che devono essere soddisfatte dalle proprietà degli individui. Gli individui che le soddisfano si dicono ISTANZE delle rispettive class expression. Sono costruibili utilizzando CONNETTIVI booleani, QUANTIFICATORI, RESTRIZIONI numeriche, ENUMERAZIONI di individui. Esistono anche le Property Expression che permettono di definire PROPRIETA'

# SPARQL

1. **Struttura base di query sparql e cosa restituisce**Sono simili alle query di SQL; prima devono essere specificati i prefissi delle ontologie che si utilizzeranno indicando gli IRI.   
   Si possono specificare i dataset, poi si fa una SELECT specificando i dati che ci interessa ottenere.   
   Successivamente, si specificano nel WHERE le condizioni (sotto forma di triple che formano un pattern corrispondente ad un insieme di grafi) che devono essere soddisfatte dagli elementi che saranno restituiti come risultato, e infine si possono aggiungere dei modificatori della query per ottenere ordinamenti diversi (ad esempio).   
   Il RISULTATO della query è dato da tutte le occorrenze delle variabili contenute nella clausola WHERE nei grafi individuati.

(ritorna un insieme di bindings)

1. **qual è la struttura di base di una query SPARQL**

# Ontology engineering

1. **Cos'è Provenance**

È un pattern utilizzabile per rappresentare

* **Agente**, ha un ruolo e un’attività associata (può essere una persona, un software, …)
* **Attività**, cosa si è fatto per quell’entità
* **Entità** (documenti, immagini, …)

Il ruolo avuto non viene specificato in Provenance.

Non si preoccupa di descrivere gli agenti, lascia il compito a FOAF; i metadati editoriali vengono lasciati a DC.

1. **Cos’è Neon**

Èuna metodologia orientata agli aspetti COLLABORATIVI nello sviluppo e nel mantenimento di RETI di ontologie. È un insieme di 9 scenari, associati a specifiche attività e documenti.   
Ci sono quattro fasi:

* + Initiation
  + Design
  + Implementation
  + Maintenance

ed è un ciclo chiuso che ritorna a Design

1. **Quali sono le metaproprietà di OntoClean**

Le proprietà, che forniscono informazioni su come caratterizzare le classi, sono:

* IDENTITA': Proprietà che permette di identificare un tipo di oggetti. Triangolo -> lunghezza dei lati. [SORTAL => identificabili allo stesso modo]. Viene ereditata dalle sottoclassi.
* UNITA': proprietà di un tipo di oggetto di essere unitario. E' ereditata dalle sottoclassi (ma non vale per la proprietà contraria)
* RIGIDITA': Proprietà di un tipo di oggetto che non è soggetta a cambiamenti: (Persona non cambia, studente lo è temporaneamente). Solo l'ANTIRIGIDITA' viene ereditata.
* DIPENDENZA: Proprietà di un tipo di oggetti di dipendere da un altro per la propria definizione. Studente dipende da scuola.

1. **Cos'è DOLCE e la differenza tra enduranti e perduranti**

DOLCE è un'ontologia orientata alla congnizione e al linguaggio. La definizione chiave è quella di PERDURANTE ed ENDURANTE:

* Evento => Perdurante (ha natura temporale)
* Partecipante => Endurante (non ha natura temporale)

1. **Cos'è una linked data platform?**

E' una piattaforma per l'accesso ai linked data che usa il paradigma REST. E' un servizio WEB basato sulle funzionalità di HTTP per leggere e scrivere i Linked Data.

1. **Come si possono modellare i ruoli nelle ontologie OWL**
2. **Approcci collaborativi**

* **Frame net**: include frame mappati su parti del discorso a cui ad ogni componente del discorso viene associato un ruolo
* **Word net**: consiste in un lessico organizzato secondo relazioni di significato. Gli elementi del lessico (sostantivi, verbi, …) sono raggruppati in insiemi denominati **synset** (insieme dei sinonimi)

1. **SKOS**

Vocabolario che permettere di allenare concetti presi da vocabolari o tassonomie differenti. È possibile specificare il tipo di relazione con i concetti.

Permette di:

* + Strutturare la tassonomia dei concetti
  + Effettuare degli allineamenti

Permette di definire etichetta della risorsa e altre info

1. **Come definire delle regole**

Attraverso swrl o rif

