

Facoltà di Ingegneria dell'informazione

Ingegneria della conoscenza 2009-10

Appello del 28 luglio 2010 – Soluzioni

Analizzare le principali differenze fra FOL e una DL come SROIQ(D_n), spiegando per quali ragioni questa è ritenuta più adatta alle applicazioni del web semantico.

Vedi le dispense, parte I, par. 2.

- Definire in OWL 2 DL la seguente ontologia concernente i grafi orientati. Identificare le eventuali specifiche che non possono essere rappresentate in OWL 2 DL, spiegando perché e, ove possibile, fornendo una rappresentazione approssimata.
- pt. ATTENZIONE: i "se" ai punti 4, 6 e 7 vanno interpretati come tali e non come dei "se e solo se"!

SUGGERIMENTO: usare come aiuto il diagramma grafico delle proprietà.

- 1. un grafo è costituito da nodi e da archi (che nel seguito diremo "appartenere" al grafo);
- 2. grafi, nodi e archi sono disgiunti;
- 3. ogni arco ha esattamente un nodo di origine e un nodo di destinazione:
- 4. se un arco appartiene a un determinato grafo, allora i nodi di origine e di destinazione dell'arco appartengono anch'essi a quel grafo;
- 5. ciascun nodo e ciascun arco appartiene a esattamente un grafo;
- 6. un nodo x "vede direttamente" un nodo y se c'è un arco di cui x è l'origine e y è la destinazione;
- 7. un nodo x "vede" un nodo y se esiste una sequenza finita di nodi che inizia in x, termina in y, e in cui ciascun nodo vede direttamente il nodo seguente;
- 8. un nodo è un "nodo stabile" se, e solo se, vede direttamente se stesso.

Distinguere le classi atomiche dalle proprietà usando notazioni diverse. Per ogni proprietà specificare dominio e codominio. Una volta definita l'ontologia formulare le seguenti interrogazioni, curando la correttezza della sintassi:

- 9. verificare se il nodo n1 vede un nodo che è visto direttamente dal nodo n2 (usare un'interrogazione di instance check);
- 10. elencare i nodi appartenenti al grafo cui appartiene l'arco a12.

1. haNodo: Grafo → Nodo haArco: Grafo → Arco

2. DisCla(Grafo,Nodo,Arco)

haOrig: Arco → Nodo
haDest: Arco → Nodo
Arco ⊑ = 1 haOrig □ = 1 haDest

4. haArco ∘ haOrig ⊑ haNodo haArco ∘ haDest ⊑ haNodo

5. Nodo = -1 haNodo

non è rappresentabile perché "haNodo" è composita per il punto 3, ma è approssimabile come: Nodo ⊑ ∃haNodo⁻

Arco = =1 haArco

6. $vedeDir: Nodo \longrightarrow Nodo$ ha $Orig^- \circ haDest \subseteq vedeDir$

7. vede: Nodo \longrightarrow Nodo vedeDir \sqsubseteq vede Tra(vede)

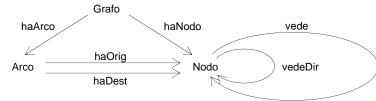
NB: non si tratta di un'approssimazione, ma di tutto ciò che è richiesto dal punto 6 (dato che il "se" non va interpretato come "se e solo se")

8. NodoStabile = EvedeDir.Self

non è rappresentabile perché "vedeDir" è composita per il punto 5, e non è approssimabile

9. $?- \exists vede.(vedeDir^- \ni n2) (n1)$

10. ?- ∃haNodo¯.(haArco ∋ a12) (*)



- 3 Per ciascuna delle seguenti espressioni SROIQ(D_n):
 - specificare se si tratta di una classe, di un enunciato di TBox, di un enunciato di RBox o di un'asserzione di ABox
 - tradurre l'espressione in italiano comune (senza utilizzare variabili o altri termini tecnici della logica o della teoria degli insiemi)
 - specificare la semantica dell'espressione in termini di modelli $M = \langle \Delta, -^{l} \rangle$
- 1. ∀amicoDi.(∃genitoreDi)

4 pt.

classe

tutto ciò che è amico solo di persone con figli

 $(...)^{\prime} = \{x \in \Delta \mid \text{ se per qualche } y \in \Delta \text{ si ha } \langle x, y \rangle \in \text{amicoDi}^{\prime}, \text{ allora per qualche } z \in \Delta \text{ si ha } \langle y, z \rangle \in \text{genitoreDi}^{\prime}\}$

2. Poliglotta ≡ ≥2 conosce.Lingua enunciato di TBox

un poliglotta è qualcuno che conosce almeno due lingue

M = ... sse Poliglotta¹ = $\{x \in \Delta \mid \#\{y \in Lingua^1 \mid \langle x, y \rangle \in conosce^1\} \ge 2\}$

3. padreDi ∋ galileo (vincenzo) asserzione di ABox

Vincenzo è padre di Galileo

 $M = \dots$ sse vincenzo' $\in \{x \in \Delta \mid \langle x, galileo' \rangle \in padreDi'\}$ ovvero sse $\langle vincenzo', galileo' \rangle \in padreDi'$

4. DisPro(alleatoDi,avversarioDi) asserzione di RBox

non si può essere allo stesso tempo alleato e avversario della stessa persona

 $M \models ...$ sse alleatoDi¹ \cap avversarioDi¹ $\subseteq \emptyset$