



Facoltà di Ingegneria dell'informazione  
**Ingegneria della conoscenza 2009–10**  
Appello del 6 settembre 2010 – Soluzioni I parte

|                    |  |
|--------------------|--|
| <b>1</b><br>4 pt.  | Spiegare il concetto di <i>Knowledge Based System</i> (KBS) e descrivere l'architettura generale di un KBS basato su DL.<br><br>Vedi le dispense, parte I, par. 2.4.   |
| <b>2</b><br>10 pt. | Utilizzando la notazione DL, definire in OWL 2 DL la seguente ontologia concernente le tabelle rettangolari di celle contenenti oggetti della classe Valore. Distinguere le classi atomiche dalle proprietà usando notazioni diverse. Per ogni proprietà specificare dominio e codominio. Identificare le eventuali specifiche che non possono essere rappresentate in OWL 2 DL, spiegando perché e, ove possibile, fornendo una rappresentazione approssimata. <ol style="list-style-type: none"><li>ogni tabella ha un numero di righe (<i>positiveInteger</i>) e un numero di colonne (<i>positiveInteger</i>);</li><li>a ogni cella sono associati: una tabella di appartenenza, un indice di riga, un indice di colonna, e un valore (della classe Valore);</li><li>ogni cella appartiene a esattamente una tabella e ha esattamente un indice di riga, un indice di colonna e un valore;</li><li>in ciascuna tabella, soltanto una cella può avere un determinato indice di riga e di colonna;</li><li>l'indice di riga di una cella non può superare il numero di righe della sua tabella, e analogamente per l'indice di colonna;</li><li>ogni tabella ha al massimo 100 celle.</li></ol> Utilizzando ove possibile asserzioni di ABox: <ol style="list-style-type: none"><li>creare la cella di riga 3 e colonna 2 della tabella tab01 e assegnarle il valore val01;</li><li>imporre che tutti i valori delle celle della tabella tab01 appartengano alla classe Valore2.</li></ol> Formulare le seguenti interrogazioni, curando la correttezza della sintassi: <ol style="list-style-type: none"><li>verificare se val01 è il valore di una cella della tabella tab01 (<i>instance check</i>);</li><li>recuperare tutti i valori delle celle della riga 3 della tabella tab01 (<i>retrieval</i>).</li></ol>   |
|                    | <ol style="list-style-type: none"><li>numRighe: Tabella <math>\longrightarrow_D</math> <i>positiveInteger</i>                      numCol: Tabella <math>\longrightarrow_D</math> <i>positiveInteger</i><br/>Tabella <math>\sqsubseteq =1</math> numRighe <math>\sqcap =1</math> numCol (non esplicitamente specificato, ma desumibile dal contesto)</li><li>inTab: Cella <math>\longrightarrow</math> Tabella<br/>indRiga: Cella <math>\longrightarrow_D</math> <i>positiveInteger</i><br/>indCol: Cella <math>\longrightarrow_D</math> <i>positiveInteger</i><br/>valore: Cella <math>\longrightarrow</math> Valore</li><li>Cella <math>\sqsubseteq =1</math> inTab <math>\sqcap =1</math> indRiga <math>\sqcap =1</math> indCol <math>\sqcap =1</math> valore</li><li>HasKey(Cella; inTab; indRiga, indCol)</li><li>non è rappresentabile in OWL 2 DL perché richiederebbe un confronto fra dati</li><li>Tabella <math>\sqsubseteq \leq 100</math> inTab<sup>-</sup></li><li>inTab(cellaX, tab01)                      indRiga(cellaX, 3)                      indCol(cellaX, 2)                      valore(cellaX, val01)</li><li><math>\exists \text{valore}^-(\text{inTab} \ni \text{tab01}) \sqsubseteq \text{Valore2}</math></li><li>?- <math>\exists \text{valore}^-(\text{inTab} \ni \text{tab01}) (\text{val01})</math></li><li>?- <math>\exists \text{valore}^-(\text{inTab} \ni \text{tab01} \sqcap \text{indRiga} \ni "3" \wedge \text{positiveInteger}) (*)</math></li></ol>   |
| <b>3</b><br>4 pt.  | Per ciascuna delle seguenti espressioni SROIQ(D <sub>n</sub> ): <ul style="list-style-type: none"><li>specificare se si tratta di una classe, di un enunciato di TBox, di un enunciato di RBox o di un'asserzione di ABox</li><li>tradurre l'espressione in italiano comune (senza utilizzare variabili o altri termini tecnici della logica o della teoria degli insiemi)</li><li>specificare la semantica dell'espressione in termini di modelli <math>M = \langle \Delta, -^I \rangle</math></li></ul> <ol style="list-style-type: none"><li><math>\exists \text{genitoreDi} . (\forall \text{beve} . \neg \text{BevandaAlcolica})</math>                      classe<br/>i genitori i cui figli bevono tutt'al più bevande non alcoliche (<i>oppure</i>: i genitori i cui figli non bevono bevande alcoliche)<br/><math>(\dots)^I = \{x \in \Delta \mid \text{se per qualche } y \in \Delta \text{ si ha } \langle x, y \rangle \in \text{beve}^I, \text{ allora } y \in \Delta \setminus \text{BevandaAlcolica}^I\}</math></li><li>Città <math>\sqcap \text{siTrovaIn} \ni \text{corsica}</math> (bonifacio)                      asserzione di ABox<br/>Bonifacio è una città che si trova in Corsica<br/><math>M \models \dots</math>    sse    <math>\text{bonifacio}^I \in \text{Città}^I \cap \{x \in \Delta \mid \langle x, \text{corsica} \rangle \in \text{siTrovaIn}^I\}</math></li><li>possiede <math>\circ</math> haParte <math>\sqsubseteq</math> possiede                      enunciato di RBox<br/>chi possiede un oggetto ne possiede ogni parte<br/><math>M \models \dots</math>    sse    <math>\text{possiede}^I \circ \text{haParte}^I \subseteq \text{possiede}^I</math></li><li>Bicicletta <math>\sqsubseteq =2</math> haParte.Ruota                      enunciato di TBox<br/>le biciclette hanno due ruote<br/><math>M \models \dots</math>    sse    <math>\text{Bicicletta}^I \subseteq \{x \in \Delta \mid \#\{y \in \text{Ruota}^I \mid \langle x, y \rangle \in \text{haParte}^I\} = 2\}</math></li></ol> |