
Web Semantico

da un sogno di Tim Berners-Lee

Silvio Peroni

Sommario

Introduzione	1
Logica Descrittiva	3
Che cos'è la logica	3
Motivazioni e definizione formale	3
Concetti avanzati	9
Ontologia e OWL	9
Che cosa significa <i>ontologia</i>	9
Che cos'è <i>OWL</i>	11
Da <i>SHOIN(D_n)</i> a OWL	11
Che cos'è la logica <i>SHOIN(D_n)</i>	11
OWL: i limiti espressivi	12
OWL: i costrutti	12
Ragionamento	14
La conseguenza logica ed il ragionamento	14
Le interrogazioni	16
Editor e ragionatori	17
Cosa e quali sono	17
Protege e RacerPro	18
Conclusioni	20
Bibliografia	20

Nel presente documento sono analizzati alcuni aspetti rilevanti del *semantic web* (o *web semantico*) quali il linguaggio OWL DL, utilizzato per la definizione di ontologie per il web semantico, e la sua astrazione logica *SHOIN(D_n)*.

Introduzione

Come termine generale, la *conoscenza* è l'informazione disponibile per l'azione [IngCon05]. Ricevuta dal soggetto che ne fa uso da varie fonti tra le quali emergono quella dell'*esperienza diretta*, del *ragionamento* e della *comunicazione*, la conoscenza viene utilizzata per tre compiti fondamentali:

- *interpretare la realtà*', in modo da capire le cause di un dato evento che si è verificato;
- *prevedere l'evoluzione della realtà*', in modo da individuare i probabili eventi che si verificheranno in futuro;
- *agire in modo razionale modificando la realtà*', costruendo piani d'azione per raggiungere determinati obiettivi.

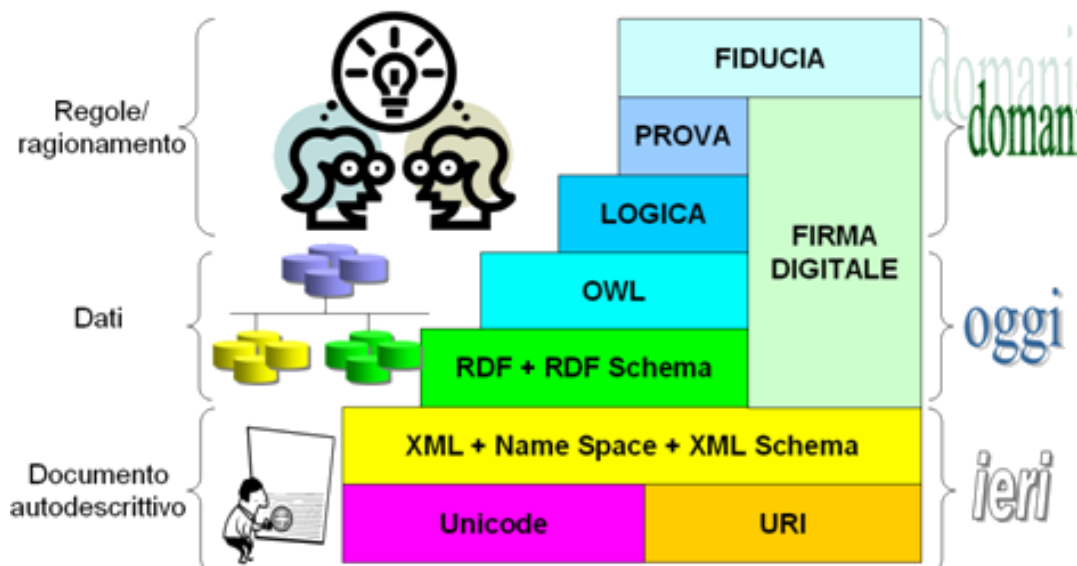
L'uso che ne viene fatto in ambito informatico è sempre stato uno dei temi fondamentali dell'intelligenza artificiale. Fin dalla nascita di questa materia, dopo una prima fase passata a ricercare possibili metodi per la soluzione automatica di problemi mediante strategie per tentativi ed errori, ci si rese subito conto (1965-1975) che la capacità di risolvere i problemi dipendeva essenzialmente dalle conoscenze che si avevano a disposizione. Diventava necessario poter rappresentare tutta la conoscenza accumulata (*Knowledge Representation*) in modo tale che un programma, attraverso un sistema informativo, potesse utilizzarla e manipolarla facilmente utilizzando un linguaggio formale ben definito [WikipediaEN, articolo sulla *Knowledge representation*]. A questo scopo, per la rappresentazione di conoscenze l'approccio adottato è d'ora in poi simbolico: la conoscenza viene

rappresentata attraverso strutture dati contenenti simboli, come sono ad esempio le formule della logica.

Dopo un primo tentativo di realizzazione di applicazioni realistiche e d'interesse industriale (1975-1985), a causa di un crollo della fiducia che il mercato economico aveva riservato ai sistemi basati su conoscenze (1985-1995), si iniziarono a cercare approcci alternativi non simbolici che, in determinati contesti, si rivelarono più adatti per la risoluzione di alcuni tipi di problemi quali l'analisi dei segnali e l'apprendimento automatico. È l'evoluzione del web avvenuta a partire dal 1995 che ha permesso alla rappresentazione simbolica della conoscenza di riemergere in prima linea. L'autore principale di questa evoluzione è stato ed è tuttora Tim Berners-Lee, già creatore del World Wide Web, che con il W3C [<http://www.w3.org>] ha assunto il ruolo di governo nello sviluppo di standard e protocolli legati al web e che, attraverso lo standard XML [<http://www.w3.org/XML/>] e XHTML [<http://www.w3.org/MarkUp/>], ha dato la possibilità di aggiungere alle normali pagine web informazioni semantiche sui contenuti attraverso la definizione di opportuni tag [WikipediaIT, articolo sul *World Wide Web*].

Da queste basi nasce l'idea del *semantic web* [WikipediaIT, articolo sul *Web semantico*], ovvero la continuazione del processo evolutivo del World Wide Web verso un ambiente dove è possibile pubblicare non più solo documenti ma anche informazioni e dati in un formato adatto alla interrogazione, interpretazione e, più in generale, elaborazione automatica.

Figura 1. Pila dei servizi del web semantico



Nella Figura 1, «Pila dei servizi del web semantico» è specificato lo *stack* di tecnologie che sono e saranno le responsabili della creazione fisica del web semantico. Gli standard finora sviluppati per il realizzarsi del web semantico sono:

- *XML* [<http://www.w3.org/XML/>] fornisce una sintassi per strutturare i documenti senza imporre vincoli semantici sul significato di tali documenti;
- *XML Schema* [<http://www.w3.org/XML/Schema>] è un linguaggio che permette di definire la struttura di un documento XML;
- *RDF* [<http://www.w3.org/RDF/>] è un semplice modello per creare dei riferimenti ad oggetti, chiamati *risorse*, e per specificare come questi oggetti sono relazionati fra loro;
- *RDF Schema* [<http://www.w3.org/TR/rdf-schema/>] è un vocabolario per descrivere proprietà e classi per delle risorse RDF con una semantica che definisce una sorta di gerarchia tra queste proprietà e classi;
- *OWL* [<http://www.w3.org/2004/OWL/>] aggiunge nuovi elementi al vocabolario usato per la de-

scrizione di proprieta' e classi.

Riprendendo le parole di Tim Berners-Lee [SemWeb01], la maggior parte del contenuto informativo dell'odierno web e' progettato per essere letto dai soli esseri umani ma non dai programmi in grado di manipolare il significato. I computer possono fare correttamente il *parsing* delle pagine web, possono seguire qualsiasi link presente nella pagina web che stanno analizzando, ma non hanno un modo per processare il significato di una pagina. Il web semantico offre la possibilita' di strutturare il significato del contenuto delle pagine web in modo da creare un ambiente nel quale gli agenti software possano rispondere a domande e svolgere i compiti che gli utenti richiedono loro.

Nella sezione chiamata «Logica Descrittiva», dopo una prima introduzione storica, viene definita la *logica descrittiva* di riferimento per la creazione delle ontologie OWL. Nella sezione chiamata «Ontologia e OWL» viene presentato il concetto sia il *ontologia* sia le motivazioni che hanno portato alla creazione del linguaggio *OWL*, i cui costrutti sono presentati nella sezione chiamata «Da *SHOIN(D_n)* a OWL». Di seguito a cio', nella sezione chiamata «Ragionamento», viene presentato concettualmente cosa sia il *ragionamento* e, nella sezione chiamata «Editor e ragionatori», come questo viene usato da alcune specifiche applicazioni per ottenere delle deduzioni su una base di conoscenza.

Per una corretta lettura e' necessario avere gia' conoscenze nell'ambito della logica del primo ordine, poiche' questa viene usata nel presente documento per definire la semantica della logica descrittiva *SHOIN(D_n)*.

Logica Descrittiva

In questa sezione viene fornita una prima introduzione teorico-storica sul significato della parola *logica* sia a livello filosofico sia a livello matematico, per poi passare ad una trattazione concreta su cosa sia la *logica descrittiva* fornendone un esempio particolare, *SHOIN(D_n)*, completo sia della definizione sintattica sia della definizione semantica.

Che cos'e' la logica

In filosofia, la *logica*, dal greco *logos* (*parola* o *discorso*), e' lo studio dei meccanismi tipici del ragionamento, cioe' della capacita' di trarre conseguenze da un certo insieme di premesse [LogInf97]. Storicamente, sono state tre principali culture a studiare i metodi di ragionamento tipici dell'uomo: quella cinese, quella indiana e quella greca.

Quella che ebbe maggiore influenza e si confermo' come logica di riferimento fu quella aristotelica, che fu la prima formulazione logica come scienza propedeutica ad ogni possibile conoscenza. Nel suo libro, l'*Organon*, egli sviluppo' un sistema informale di sillogismi per il ragionamento corretto che, in via di principio, consentivano a chiunque, date le premesse iniziali, di generare meccanicamente le conclusioni [IA05]. Molto tempo dopo, Ramon Llull ebbe l'idea di utilizzare un'artefatto meccanico per eseguire dei ragionamenti utili creando le famose "ruote della logica", mostrate in Figura 2, «Ruote della logica». Fu Thomas Hobbes, pero', ad ipotizzare che il ragionamento avesse una natura affine al calcolo numerico: "noi eseguiamo addizioni e sottrazioni nei nostri pensieri silenziosi".

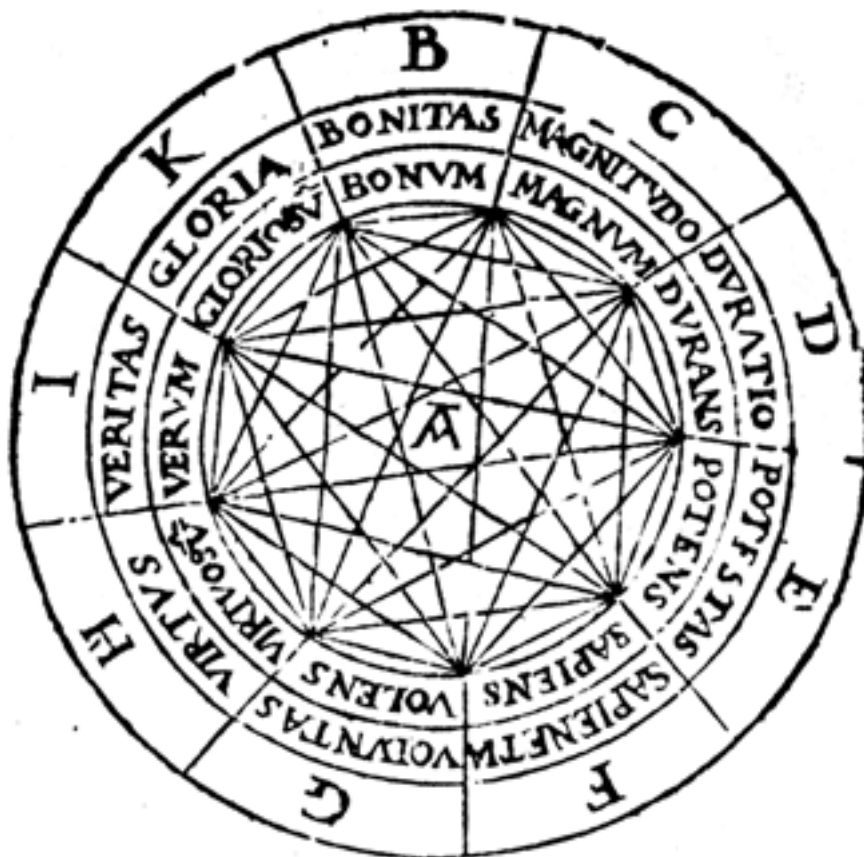
Col tempo e' emersa una certa necessita' di astrarre dal contenuto informativo (il senso) delle frasi, considerando soltanto la loro forma o struttura. Questo ha portato all'utilizzo di un linguaggio di natura algebrica e formale per descrivere sia le frasi (sentenze) che sono oggetto della logica sia le sue regole di inferenza, coprendo gli atteggiamenti sia sintattici sia semantici del linguaggio che si usa per esprimere le sentenze. Questa restrizione alla logica tradizionale e' conosciuta come *logica matematica*, termine coniato dal matematico torinese Giuseppe Peano.

Motivazioni e definizione formale

Una *rappresentazione* di conoscenza intende rappresentare un frammento di realta' interessante per una qualche applicazione. La rappresentazione puo' essere analizzata su tre piani differenti, mostrati dal livello piu' concreto al livello piu' astratto:

1. un *modello concreto* esprime conoscenze *fattuali*, ovvero relative a fatti particolari, del frammento di realta' che si sta rappresentando;
2. un *modello concettuale* esprime conoscenze relative al lessico della lingua usata (conoscenze *terminologiche*) e fornisce una descrizione delle regole generali regolatrici (conoscenze *nomologiche*) del frammento di realta' che si sta rappresentando;
3. un *metamodello* e' la specifica degli strumenti formali usati per definire il modello concreto ed il modello concettuale.

Figura 2. Ruote della logica



In riferimento alle applicazioni web, il problema e' quello di riuscire a rappresentare le conoscenze in modo che siano leggibili e manipolabili da un computer. In questo contesto, la conoscenza deve essere presentata in forma dichiarativa (*knowing that*: conoscere) piuttosto che procedurale (*knowing how*: saper fare). Il paradigma di riferimento per le rappresentazioni dichiarative e' la logica matematica (o simbolica), con particolare riferimento alla *logica dei predicati del primo ordine*, o *FOL* (*first order logic*).

Per riuscire a manipolare le conoscenze, un computer deve svolgere su di esse un certo tipo di ragionamento. Dei tre paradigmi di ragionamento principali, *deduttivo* (dalle premesse alle conclusioni), *abducente* (dagli effetti alle cause), *induttivo* (dallo specifico al generale), un computer che utilizza una FOL per rappresentare della conoscenza e per compiere dei ragionamenti sulla conoscenza a disposizione si basa sul procedimento deduttivo. In generale, una *deduzione* e' un processo che fa passare da alcune espressioni, dette *premesse* o *ipotesi*, ad un'espressione, detta *conclusione* o *tesi*, in modo che se le premesse sono vere, lo sara' anche la conclusione.

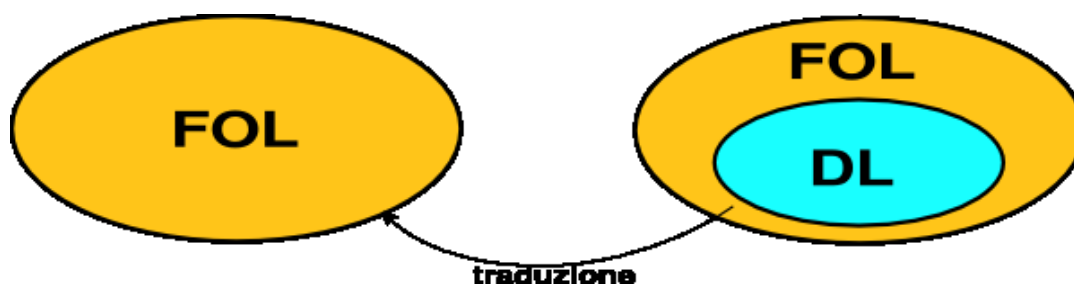
Utilizzando la FOL si possono esprimere conoscenze molto articolate ed eseguire in modo automatico ragionamenti complessi. Il problema di questo tipo di logica e' che il meccanismo di deduzione e'

una procedura di *semidecisione*:

- quando la conclusione e' deducibile date le premesse, la procedura termina in un numero finito di passi producendo una prova;
- quando la conclusione non e' deducibile date le premesse, la procedura puo' non terminare.

Per questa ragione, nel rappresentare delle conoscenze, si cerca di utilizzare un sottoinsieme di FOL che (a) rimanga abbastanza espressivo per le applicazioni, (b) che abbia una procedura di deduzione *decidibile* ovvero che termini in ogni caso dopo un numero finito di passi sia se la conclusione risulta deducibile dalle premesse sia se non lo e', e che (c) la procedura di deduzione abbia un costo computazionalmente accettabile. I sistemi di questo tipo hanno preso il nome di *logiche descrittive* (*description logic*, o *DL*).

Figura 3. Relazione tra le FOL e le DL



Come evidenziato dalla Figura 3, «Relazione tra le FOL e le DL», le DL sono un sottoinsieme proprio delle FOL. Come conseguenza, le DL hanno un livello inferiore di espressivita' rispetto alle FOL e, in generale, ogni formula DL puo' essere tradotta in FOL (ma non viceversa). In un sistema di rappresentazione della conoscenza, una base di conoscenza (*knowledge base* o *KB*) e' composta da due elementi fondamentali:

- una *terminological box*, o *TBox*, che rende disponibile ed accessibile la rappresentazione formale del modello concettuale di un frammento di realta';
- un'*assertion box*, o *ABox*, che rende disponibile e accessibile la rappresentazione formale del modello concreto di un frammento di realta'.

Questi due elementi piu' la definizione di una DL delineano una rappresentazione di un frammento di realta'. La base di conoscenza di questa rappresentazione deve essere riempita con sentenze della DL di riferimento. Per questa ragione bisogna definire una DL specifica, in questo caso *SHOIN(D_n)*, in modo che gli esempi che verranno affrontati di seguito nel documento risultino chiari. Di seguito vengono definite la sintassi e la semantica della DL di riferimento.

Sintassi della DL

Nella Figura 4, «Sintassi della DL considerata» viene mostrata la sintassi della DL utilizzata, in questo caso la logica *SHOIN(D_n)*.

Figura 4. Sintassi della DL considerata

$$\begin{aligned}
 TBox &\rightarrow C \sqsubseteq D \mid C \equiv D \mid R \sqsubseteq S \mid R \equiv S \mid Tr(R) \\
 ABox &\rightarrow C(a) \mid R(a, b) \mid a = b \mid a \neq b \\
 C &\rightarrow \top \mid \perp \mid A \mid \exists R.C \mid \forall R.C \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \leq_n R \mid \geq_n R \mid =_n R \mid \{a_1, \dots, a_n\}
 \end{aligned}$$

I simboli che compaiono nella definizione della sintassi hanno i seguenti significati:

- A rappresenta un termine atomico;
- C e D rappresentano termini arbitrari;
- R e S rappresentano ruoli (atomici);
- a, b, a_n rappresentano nominali;
- n rappresenta un numerale naturale;
- $TBox$ rappresenta un enunciato;
- $ABox$ rappresenta un'asserzione;
- $[...]$ indica una componente opzionale.

Per il significato degli elementi presentati si veda la sezione chiamata «Semantica della DL».

Semantica della DL

Per la definizione della semantica si è preferito usare una traduzione direttamente in FOL poiché tale traduzione per definizione risulta sempre possibile e, a patto di conoscere già la notazione FOL, risulta essere molto più intuitiva di una trattazione insiemistica. Ove necessario, su ogni immagine relativa ai costrutti della DL viene indicata la traduzione in FOL.

Il linguaggio formale di una DL serve a definire *termini* che descrivono concetti. Come messo in evidenza dalla Figura 5, «Termini, equivalenza e sussunzione», i termini base assumono il significato di *vero* e *falso*. In generale, i termini possono essere *atomici* o *composti*. La composizione avviene tramite gli operatori di *and*, di *or* e di *not*.

Figura 5. Termini, equivalenza e sussunzione

Termini base \top, \perp –trad– $[\top]_x = (x = x)$, $[\perp]_x = (x \neq x)$

Termini atomici A, B –trad– $[A]_x = A(x)$

Termini complessi $A \text{ op } B, C, D$ con op uguale a \sqcup, \sqcap o \neg –trad– $[A \text{ op } B]_x = A(x) \text{ op } B(x)$

Equivalenza terminologica $C \equiv D$ con C e D termini complessi –trad– $[C \equiv D] = \forall x([C]_x \leftrightarrow [D]_x)$

Definizione terminologica $A \equiv C$ con C termine complesso –trad– $[A \equiv C] = \forall x([A]_x \leftrightarrow [C]_x)$

Sussunzione terminologica $C \sqsubseteq D$ –trad– $[C \sqsubseteq D] = \forall x([C]_x \rightarrow [D]_x)$

Una relazione di equivalenza come quella in figura si legge *C equivale a D*. Se una relazione di equivalenza si presenta tra un termine semplice ed uno composto si parla di *definizione terminologica* e si dice, dall'esempio in figura, *A è definito come D*. L'ultima espressione della Figura 5, «Termini, equivalenza e sussunzione» fa riferimento all'operazione di *sussunzione terminologica*: *C è assunto da D* o *C è un D* esprime il fatto che ogni individuo descritto dal termine *C* è anche descritto dal termine *D*. Oltre ai termini, corrispondenti in FOL a predicati con un argomento, le DL in generale utilizzano termini corrispondenti a predicati a due argomenti che esprimono relazioni binarie tra individui della realtà. Tali termini vengono detti *ruoli*. Nella Figura 6, «Ruoli quantificati» vengono indicati i ruoli quantificati con le relative traduzioni. La quantificazione può essere di due tipi: semplice, con il solo ruolo, o qualificata. Si noti come la quantificazione universale semplice sia in realtà vera per ogni individuo dell'universo.

Figura 6. Ruoli quantificati

$$\begin{aligned} \exists \text{ semplice } \exists R \text{ -trad- } [\exists R]_x &= \exists y R(x, y) \\ \exists \text{ qualificato } \exists R.C \text{ -trad- } [\exists R.C]_x &= \exists y (R(x, y) \wedge [C]_y) \\ \forall \text{ semplice } \forall R \text{ -trad- } [\forall R]_x &= \forall y R(x, y) \\ \forall \text{ qualificato } \forall R.C \text{ -trad- } [\forall R.C]_x &= \forall y (R(x, y) \rightarrow [C]_y) \end{aligned}$$

Nella maggior parte delle DL e' possibile esprimere vincoli di cardinalita' *semplice* o di cardinalita' *qualificata*, come espresso nella Figura 7, «Cardinalita', equivalenza e sussunzione tra ruoli», dove n indica una costante naturale arbitraria. Nella *SHOIN*(D_n), pero', e' possibile esprimere ruoli con la sola cardinalita' semplice.

Figura 7. Cardinalita', equivalenza e sussunzione tra ruoli

$$\begin{aligned} \text{Cardinalità semplice } \leq_n R, \geq_n R, =_n R \text{ -trad- } [op_n R]_x &= \exists_{op_n} y R(x, y) \\ \text{Cardinalità composta } \leq_n R.C, \geq_n R.C, =_n R.C \text{ -trad- } [op_n R.C]_x &= \exists_{op_n} y (R(x, y) \wedge [C]_y) \\ \text{Sussunzione tra ruoli } R \sqsubseteq S \text{ -trad- } [R \sqsubseteq S] &= \forall x \forall y (R(x, y) \rightarrow S(x, y)) \\ \text{Equivalenza tra ruoli } R \equiv S \text{ -trad- } [R \equiv S] &= \forall y (R(x, y) \leftrightarrow S(x, y)) \end{aligned}$$

Come per i termini, anche tra ruoli e' possibile esprimere relazioni di equivalenza e sussunzione. Entrambe le relazioni sono concesse nella logica *SHOIN*(D_n).

In riferimento alla Figura 8, «Ruolo inverso, dominio e codominio», dato un ruolo R , che esprime una relazione binaria $R(x,y)$, il *ruolo inverso* R^- esprime la stessa relazione a variabili invertite $R(y,x)$.

Figura 8. Ruolo inverso, dominio e codominio

$$\begin{aligned} \exists \text{ inverso } \exists R^-.C \text{ -trad- } [\exists R^-.C]_x &= \exists y (R(y, x) \wedge [C]_y) \\ \forall \text{ inverso } \forall R^-.C \text{ -trad- } [\forall R^-.C]_x &= \forall y (R(y, x) \rightarrow [C]_y) \\ \text{Dominio } \top \sqsubseteq \forall R^-.D \text{ -trad- } [\top \sqsubseteq \forall R^-.D] &= \forall x \forall y (R(y, x) \rightarrow [D]) \\ \text{Codominio } \top \sqsubseteq \forall R.C \text{ -trad- } [\top \sqsubseteq \forall R.C] &= \forall x \forall y (R(x, y) \rightarrow [C]_y) \\ \text{Ruolo funzionale } \top \sqsubseteq \forall R^-.D, \top \sqsubseteq \forall R.C, C \sqsubseteq_{\leq 1} R & \\ \text{Ruolo funzione } \top \sqsubseteq \forall R^-.D, \top \sqsubseteq \forall R.C, D \sqsubseteq_{=1} R & \end{aligned}$$

I ruoli inversi, oltre che essere molto utili, sono necessari per identificare il dominio di un qualsiasi ruolo. L'associazione del dominio e del codominio ad un ruolo R e' necessaria per stabilire a quali insiemi di individui fanno riferimento le variabili x e y dell'espressione $R(x,y)$. Oltre a cio', se un ruolo esprime una relazione funzionale, ovvero una relazione in cui ogni elemento del dominio e' in relazione con al piu' un elemento del codominio, viene chiamato *ruolo funzionale*. Se invece un ruolo esprime una relazione funzionale *funzione*, ovvero una relazione in cui ogni elemento del dominio e' in relazione esattamente con un elemento del codominio, viene chiamato *ruolo funzione*. In una ontologia [1], e' spesso possibile utilizzare anche nomi di individui come a_1, \dots, a_n detti comunemente

¹Vedere a tal proposito la sezione chiamata «Che cosa significa *ontologia*».

nominali. Un termine enumerativo, indicato in Figura 9, «Termine enumerativo e unicità dei nomi», individua un insieme di scelta tra i nominali indicati. Spesso, l'operatore $\{...\}$ viene chiamato *one-of* perché enumera tutti i valori che la variabile x può assumere.

Figura 9. Termine enumerativo e unicità dei nomi

Termine enumerativo $\{a_1, \dots, a_n\}$ –trad– $[\{a_1, \dots, a_n\}]_x = (x = a_1 \vee \dots \vee x = a_n)$

Unicità dei nomi $\neq (a_1, \dots, a_n)$

Contrariamente a quello che succede nelle FOL, nelle DL si può assumere l'*unicità dei nomi* (*unique name assumption*). Se vale tale assunzione, due nominali distinti non possono far riferimento allo stesso individuo nell'universo. Ovviamente questa assunzione risulta irrealistica nell'ambito del web.

Nella maggior parte delle DL, *SHOIN(D_n)* inclusa, non si possono definire ruoli complessi realizzati per composizione come descritto in Figura 10, «Composizione di ruoli e proprietà transitiva».

Figura 10. Composizione di ruoli e proprietà transitiva

Composizione di ruoli $(R \circ S)(x, y) = \exists z (R(x, z) \wedge S(z, y))$

Proprietà transitiva $(R \circ R) \subseteq R$ oppure $Tr(R)$

L'operazione di composizione veste, però, un ruolo fondamentale per la definizione della proprietà transitiva di un ruolo, utilizzata in molti contesti. Di conseguenza in *SHOIN(D_n)* viene fornito il costrutto particolare $Tr(R)$ che permette di associare la proprietà transitiva al ruolo R pur non avendo a disposizione l'operazione di composizione di ruoli. La dichiarazione di transitività svolta in questo modo va considerata a tutti gli effetti come un *assioma terminologico* [2].

TBox e ABox

Come già detto nella sezione chiamata «Motivazioni e definizione formale», una base di conoscenza è formata da una TBox ed un'ABox. La TBox contiene assiomi terminologici, ovvero enunciati terminologici assunti come veri; l'ABox contiene conoscenze fattuali espresse sotto forma di *asserzioni*.

Nelle DL si possono esprimere diversi tipi di conoscenze fattuali, come $C(a)$ o $R(a, b)$, dove a e b sono nominali, non necessariamente distinti. Oltre a ciò, si possono esprimere asserzioni composte da termini composti con specificato il nominale a cui si fa riferimento. La semantica di un'asserzione si ottiene dalla traduzione FOL del termine che si sta considerando sostituendo tutte le occorrenze della variabile x del termine con il nominale relativo all'asserzione. Per ulteriori esempi vedere la Figura 11, «Esempio di semantica per l'ABox».

Figura 11. Esempio di semantica per l'ABox

Semantica delle asserzioni $[C \sqcap \exists R(a_i)] = C(a_i) \wedge \exists y R(a_i, y)$ con a_i nominale

Al contrario delle basi di dati, la semantica dell'ABox è compatibile con una situazione di *conoscenza parziale*, ovvero:

- tutto ciò che è contenuto nell'ABox è vero;

²Un *enunciato terminologico* è un'espressione che esprime o equivalenza o sussunzione tra termini. Un *assioma terminologico* è un enunciato terminologico assunto come vero.

- tutto cio' che non e' contenuto dell'ABox non e' ne vero ne falso, semplicemente non si conosce.

Questo e' in netta contrapposizione con la semantica adottata nelle basi di dati, in cui si tiene conto dell'*assunzione del mondo chiuso*, la quale afferma che tutto quello che non e' contenuto nell'ABox e' necessariamente falso. In realta' questa assunzione nelle DL puo' essere ottenuta su determinati termini utilizzando un costrutto particolare (vedere la sezione chiamata «Le interrogazioni»).

Concetti avanzati

Come gia' detto, i nominali vengono usati per fare riferimento ad individui nell'universo. Tuttavia nelle DL, il concetto di individuo, e di conseguenza di nominale, va interpretato in modo piu' allargato. In generali, i nominali possono essere usati:

- per rappresentare *oggetti individuali* veri e propri, detti *atomi*, come una persona, una bottiglia di vino;
- per rappresentare i possibili *valori di un attributo*, dove l'attributo e' rappresentato da un ruolo ed i possibili valori dell'attributo sono dei nominali che presi collettivamente costituiscono il codominio del ruolo. Ad esempio il ruolo *Sesso* ha come codominio $\{f,m\}$;
- per rappresentare *specie* o *tipi*, ovvero si puo' usare un nominale, ad esempio *barolo*, per rappresentare un tipo di vino;
- per rappresentare *entita' complesse reificate*, ad esempio come il nominale *ia_rapcon* che potrebbe rappresentare un esame inteso come l'aggregato di uno studente, un corso, una data ed un voto.

In relazione ai termini della DL presentata, se *C* *sussume* *D* allora *C* si dice *specie* mentre *D* si dice *genere*. I termini che partecipano ad una sussunzione godono di tre regole fondamentali:

1. l'*ereditarieta'*, in cui le proprieta' del *genere* vengono ereditate dalla *specie*;
2. le *sussunzioni implicite*, dove alcune sussunzioni possono essere derivate implicitamente da definizioni terminologiche che utilizzano o l'intersezione o l'unione di termini;
3. la *specificita'*, grazie alla quale una *specie* o restringe l'insieme dei possibili valori di un attributo del *genere* o possiede direttamente degli attributi che il *genere* non possiede.

Ontologia e OWL

In questa sezione e' presentato, sia in veste filosofica sia in gergo informatico, il concetto di *ontologia*, uno dei pilastri fondamentali del web semantico, e il linguaggio *OWL* realizzato dal W3C per definire ontologie per il web semantico.

Che cosa significa *ontologia*

In termini filosofici, l'ontologia, dal greco *eon* (ente) e *logos* (discorso) - letteralmente *discorso sull'ente* - e' una delle maggiori branche della metafisica e riguarda lo studio dell'essere o dell'esistenza nonche' delle sue categorie fondamentali [WikipediaIT, articolo sull'*Ontologia*]. Il fondatore dell'ontologia e' Parmenide di Elea (V secolo a.c.) che fu il primo a porsi il problema dell'essere considerato nella sua totalita' e, di conseguenza, a porsi il problema sull'ambiguita' tra i piani logico, ontologico, linguistico, elaborati successivamente da altri filosofi quali Platone, Aristotele, Cartesio, Kant.

Nel gergo informatico associato alla rappresentazione di conoscenza, un'ontologia e' il modello concettuale della rappresentazione. Piu' precisamente in informatica, il termine ontologia e' definito dall'unione dei seguenti concetti:

- una rappresentazione di elementi di un dominio;
- le relazioni che intercorrono tra gli elementi considerati.

Seppur tutte le conoscenze definite dall'ontologia siano generali, si possono assumere tre diversi livelli di generalita', dal piu' astratto al piu' concreto:

1. *l'ontologia superiore* fa riferimento a elementi il piu' astratti possibili e di uso praticamente universale ed indipendente dalla singola area applicativa. Elementi come "oggetto fisico", "prodotto", "evento" possono essere membri di quest'ontologia;
2. *l'ontologia mediana* fa riferimento ad un'area applicativa ben definita, come il commercio elettronico, i computer, gli strumenti musicali e cosi' via;
3. *l'ontologia inferiore* e' utilizzata per interfacciare una singola applicazione con una collezione di ontologie mediane.

Siccome la maggior parte delle applicazioni che lavorano con ontologie tendono a definire e manipolare elementi di livello mediano, alcuni enti stanno tuttora lavorando alla definizione di un'ontologia superiore che possa essere adottata come standard di riferimento per lo sviluppo di ontologie mediane ed inferiori. Una di queste enti e' l'IEEE [<http://www.ieee.org>] che con la sua *Standard Upper Ontology* [<http://suo.ieee.org>] sta cercando di definire un'ontologia superiore standard utilizzabile da chiunque e da qualunque applicazione.

Nel web, le ontologie possono essere usate sia per migliorare le gia' esistenti applicazioni web-based sia per rendere partecipi piu' utenti con il web [OwlUseCase]. In letteratura sono stati proposti alcuni casi d'uso per le ontologie su web in modo che potesse essere spiegata la loro effettiva utilita'. Facciamo qualche esempio.

- *Portale web*

Un portale web (*web portal*) e' un sito che riunisce il contenuto informativo con un argomento comune, organizzato ad esempio secondo un determinato dominio di interesse. Un portale permette a chiunque lo desideri di ricercare notizie in relazione all'argomento desiderato, costruire una comunita' e trovare link verso altre risorse web di interesse comune. Normalmente un semplice indice di tutti gli argomenti e di tutti gli articoli contenuti nel portale non fornisce alla comunita' un meccanismo di ricerca efficiente sull'argomento che un qualsiasi membro del portale desidera cercare. Un'ontologia, invece, garantisce una ricerca piu' efficiente poiche' fornisce (a) una terminologia per descrivere i contenuti e (b) una serie di assiomi che permettono di definire nuovi elementi usando elementi gia' presenti nell'ontologia. Un esempio di un portale basato su un'ontologia e' OntoWeb [<http://www.ontoweb.org/>], un portale che serve una comunita' accademico-industriale che e' interessata alla ricerca e allo sviluppo di ontologie.

- *Collezioni multimediali*

Le ontologie possono essere usate per fornire delle annotazioni semantiche per collezioni di immagini, audio ed altri oggetti multimediali in modo da rendere piu' facile ad una macchina l'estrazione di informazioni utili da un file multimediale. Queste ontologie multimediali possono essere essenzialmente di due tipi: orientate al media od orientate al contenuto. Le prime sono caratterizzate da una tassonomia di differenti tipi di media e devono descrivere proprieta' di media differenti. Le seconde devono descrivere il soggetto della risorsa che si sta analizzando.

- *Agenti e servizi*

Il web semantico puo' fornire agenti capaci di comprendere ed integrare diverse informazioni. Un esempio specifico di questo tipo di agenti e' quello dei "pianificatori di attivita'", che ricevono alcune informazioni di preferenza da un utente e le usano per pianificare le attivita' dell'utente per una data ora della giornata: ad esempio, si puo' chiedere all'agente di trovare in quali cinema sono proiettati i film che hanno ricevuto nella settimana passata una valutazione alta dalla critica. Il lavoro di pianificazione di queste attivita' che deve svolgere l'agente dipende anche dalla quantita' di informazione che l'ambiente di ricerca (il web, ad esempio) e' in grado di offrire.

Questo tipo di agente richiede delle ontologie del dominio considerato, che possano rappresentare termini come "film", "cinema", ecc. e delle ontologie dell'ambiente di ricerca in modo che possano essere rappresentati gli elementi che sono utilizzati in quel determinato ambiente. Agentcities [<http://www.agentcities.org/>] e' un esempio di un'iniziativa che studia l'uso di agenti in un servizio distribuito (l'ambiente) su internet.

Utilizzando questi ed altri casi d'uso, il W3C ha proposto uno standard per la definizione di ontologie per il web semantico chiamato *Web Ontology Language* o piu' semplicemente *OWL*.

Che cos'e' OWL

Il web semantico e' la prossima evoluzione del web [OwlSpec] nel quale le informazioni delle pagine sono associate esplicitamente, attraverso appositi tag, ad un significato in modo che i computer le possano processare e manipolare automaticamente. Il web semantico e' costruito sui linguaggi XML e RDF, che forniscono la possibilita' di definire nuovi schemi di tag permettendo un approccio flessibile per la rappresentazione dei dati.

Il primo livello sopra l'RDF richiesto per la formazione del web semantico (vedi la Figura 1, «Pila dei servizi del web semantico») e' un linguaggio per la creazione di ontologie che possa descrivere formalmente il significato della terminologia usata in un documento per il web. Il *Web Ontology Language*, o *OWL*, e' lo standard proposto dal W3C [<http://www.w3.org/2004/OWL/>] per la definizione di ontologie per il web semantico. Esso definisce tre linguaggi differenti, *Lite*, *DL* e *Full*, caratterizzati da un diverso livello di espressivita', dal meno espressivo al piu' espressivo. In questo documento ci concentreremo solamente su *OWL DL* che, implementando la logica descrittiva *SHOIN(D_n)*, risulta essere abbastanza espressivo rimanendo decidibile e dotato di procedure di ragionamento di complessita' nota, approfonditamente studiate ed ormai ben ottimizzate. In generale, pero', ogni linguaggio e' visto come un'estensione del predecessore. Valgono tra i tre linguaggi le seguenti proprieta' (non le inverse):

- ogni ontologia legale espressa con OWL Lite e' anche un'ontologia legale di OWL DL;
- ogni ontologia legale espressa con OWL DL e' anche un'ontologia legale di OWL Full;
- ogni conclusione che risulta valida in OWL Lite e' una conclusione valida in OWL DL;
- ogni conclusione che risulta valida in OWL DL e' una conclusione valida in OWL Full.

Un'ontologia in OWL si articola in una TBox ed un'ABox, ambedue rappresentate da grafi RDF. La rappresentazione RDF delle ontologie OWL presenta grandi vantaggi per l'interoperabilita' delle applicazioni ma risulta essere di difficile lettura e molto confusa a causa delle numerose varianti che RDF consente di utilizzare. Fortunatamente, la maggior parte degli strumenti per la definizione di ontologie OWL utilizzano interfacce grafiche che nascondono all'utente la rappresentazione RDF.

Da SHOIN(D_n) a OWL

In questa sezione viene presentata formalmente la logica *SHOIN(D_n)* sulla quale e' stato basato il linguaggio OWL DL, mettendone in evidenza i limiti espressivi derivati da questa scelta. Di seguito a cio', viene mostrato con quali nomi sono stati tradotti in OWL i costrutti che compongono la logica *SHOIN(D_n)*.

Che cos'e' la logica SHOIN(D_n)

Nella sezione chiamata «Logica Descrittiva» e' stata trattata la sintassi e la semantica della logica *SHOIN(D_n)* che sta alla base del linguaggio OWL presentato nella sezione chiamata «Che cos'e' OWL». Nella presente sezione si vuole chiarire il significato dell'acronimo usato per identificare tale logica.

- La lettera S indica la possibilita' di scrivere enunciati di equivalenza e di sussunzione utilizzando i termini base, *vero* e *falso*, i termini composti dagli operatori di *and*, *or* e *not*, i ruoli quantificati semplici o qualificati e l'assioma di transitivita' dei ruoli.
- La lettera H indica la possibilita' di fare sussunzioni tra ruoli.
- La lettera O indica la possibilita' di definire termini per enumerazione (*one-of*).
- La lettera I indica la possibilita' di definire il ruolo inverso.
- La lettera N indica la possibilita' di definire cardinalita' non qualificate.
- La lettera D_n indica la possibilita' di utilizzare domini concreti, denotati da termini atomici, come NATURAL, FLOAT, CHARACTER, STRING.

Come si puo' notare dal significato dell'acronimo, nella logica $SHOIN(D_n)$ non e' possibile definire cardinalita' qualificate [3].

OWL: i limiti espressivi

Come gia' messo in evidenza nella sezione chiamata «Motivazioni e definizione formale», le DL sono meno espressive delle FOL. Di conseguenza, vi sono enunciati del primo ordine che non possono essere espressi in una DL e, conseguentemente, nemmeno in OWL DL [4]. In particolare sono due le grosse limitazioni che impone una DL $SHOIN(D_n)$:

- non e' possibile esprimere *direttamente* un ruolo R come composizione di altri due ruoli. Il "direttamente" della frase precedente e' dovuto al fatto che, se si compie una *reifificazione*, ovvero una trasformazione da ruolo a termine, del ruolo composto, si puo' ottenere lo stesso effetto. Per esempio, supponiamo di voler esprimere il ruolo *Zio* come composizione dei ruoli *Fratello* e *Figlio*.

Figura 12. Reificazione di un ruolo

$Zio \equiv Fratello \circ Figlio$, non si può esprimere
 $ZIO \equiv \exists Fratello.(\exists Figlio)$, con reificazione

Siccome cio', come gia' detto, non si puo' esprimere in una $SHOIN(D_n)$, si puo' pensare di reificare il ruolo *Zio* nel termine *ZIO* come descritto nella Figura 12, «Reificazione di un ruolo».

- non e' possibile definire il prodotto cartesiano tra due insiemi, cosa che invece in FOL e' possibile.

OWL: i costrutti

Come gia' indicato nella sezione chiamata «Che cos'e' OWL», in questo documento ci limiteremo a trattare esclusivamente il linguaggio OWL DL. Di conseguenza, tutti i futuri riferimenti a OWL sono da intendere come se si stesse considerando il linguaggio particolare OWL DL se non indicato altrimenti.

Nel linguaggio OWL i termini sono chiamati *descrizioni di classi*, gli operatori per la definizione di termini sono chiamati *costruttori di classi*, mentre i ruoli sono chiamati *proprietà*. Sia le classi sia le proprietà hanno un *identificatore*, *Class* per le prime [5] e *Property* per le seconde [6], che svolge la funzione di costruttore. Escludendo l'*identificatore*, OWL prevede altri cinque tipi di descrizioni di

³La lettera che identifica questa possibilita' e' la Q .

⁴Cosa diversa se si considera OWL Full. Questo infatti e' molto piu' espressivo di una FOL.

⁵OWL prevede due identificatori predefiniti per la *classe universale*, rappresentata da *Thing* (termine *vero* nelle DL), e la *classe vuota*, rappresentata da *Nothing* (termine *falso* nelle DL). Tutte le classi che vengono create sono per default anche sottoclassi (*subClassOf*) di *Thing*.

⁶In realta' gli identificatori per queste ultime possono essere *ObjectProperty*, proprietà di individui cioè fra elementi OWL, o *DatatypeProperty*, proprietà di dati appartenenti a tipi di dati RDF.

classi (vedi Figura 13, «Da DL a OWL con esempi»):

- l'*enumerazione*. Una classe finita puo' essere descritta dall'enumerazione di tutti gli individui che le appartengono (*oneOf*);
- la *restrizione di proprieta'* e' una descrizione di classe corrispondente ai ruoli quantificati semplici o qualificati ed ai ruoli con cardinalita' semplice (*someValuesFrom*, *allValuesFrom*, *hasValue*, *maxCardinality*, *minCardinality*, *cardinality*);
- l'*intersezione* e' una descrizione di classe definita come *and* tra un numero finito di classi (*intersectionOf*);
- l'*unione* e' una descrizione di classe definita come *or* tra un numero finito di classi (*unionOf*);
- il *complemento* e' una descrizione di classe definita come *not* di un'altra classe (*complementOf*).

Figura 13. Da DL a OWL con esempi

OWL	DL	Esempi
oneOf	$\{a_1, \dots, a_n\}$	$\{silvio, giovanni\}$
someValuesFrom	$\exists R.C$	$\exists Figlio.MASCHIO$
allValuesFrom	$\forall R.C$	$\forall Figlio.MASCHIO$
hasValue	$\forall R.\{a\}$	$\exists Cittadino.\{italiano\}$
maxCardinality	$\leq_n R.C$	$\leq_3 Figlio.MASCHIO$
minCardinality	$\geq_n R.C$	$\geq_2 Figlio.MASCHIO$
cardinality	$=_n R.C$	$=_1 Figlio.MASCHIO$
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	$PERSONA \sqcap MASCHIO$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	$FILOSOFO \sqcup INGEGNERE$
complementOf	$\neg C$	$\neg MASCHIO$
subClassOf	$C_1 \sqsubseteq C_2$	$PERSONA \sqsubseteq ANIMALE \sqcap BIPEDE$
equivalentClass	$C_1 \equiv \dots \equiv C_n$	$UOMO \equiv PERSONA \sqcap MASCHIO$
disjointWith	$C_1 \sqcap C_2 = \perp$	$FEMMINA \sqcap MASCHIO = \perp$
domain	$\top \sqsubseteq \forall R^-.D$	$\top \sqsubseteq \forall Marito^-.DONNA$
range	$\top \sqsubseteq \forall R.C$	$\top \sqsubseteq \forall Marito.UOMO$
subPropertyOf	$R_1 \sqsubseteq R_2$	$Figlio \sqsubseteq Parente$
equivalentProperty	$R_1 \equiv \dots \equiv R_n$	$Costo \equiv Prezzo$
inverseOf	$R_1 \equiv R_2^-$	$Genitore \equiv Figlio^-$
FunctionalProperty	$\top \sqsubseteq \leq_1 R$	$\top \sqsubseteq \leq_1 Marito$
InverseFunctionalProperty	$\top \sqsubseteq \leq_1 R^-$	$\top \sqsubseteq \leq_1 Nubile^-$
TransitiveProperty	$Tr(R)$	$Tr(Antenato)$
SymmetricProperty	$R \equiv R^-$	$ConnessoA \equiv ConnessoA^-$
TransitiveProperty	$Tr(R)$	$Tr(Antenato)$
sameAs	$a_1 = \dots = a_n$	$vale = valerio$
differentFrom	$a_i \neq a_j$	$silvio \neq giovanni$
AllDifferent	$\neq (a_1, \dots, a_n)$	$\neq (silvio, giovanni, valerio)$

In OWL le definizioni terminologiche della TBox sono dette *assiomi di classe*. Questi possono essere:

- assiomi di *sottoclasse*, che descrive una relazione di *sussunzione* tra classi (*subClassOf*);
- assiomi di *equivalenza*, che descrive una relazione di *equivalenza* tra classi (*equivalentClass*);
- assiomi di *disgiunzione*, che descrive una relazione di disgiunzione tra classi (*disjointWith*).

In OWL anche le *proprieta'* sono viste come particolari tipi di classe. Di conseguenza tali proprieta' possono avere sottoproprieta' e possono essere combinate con vari costruttori. Escludendo l'*identifi-*

catore:

- il *dominio* e il *codominio* possono essere specificati con i costrutti *domain* e *range*;
- le *sottoproprieta'* descrivono una relazione di *sussunzione* tra proprieta' (*subPropertyOf*);
- la *proprieta' equivalente* descrive una relazione di *equivalenza* tra proprieta' (*equivalentProperty*);
- la *proprieta' inversa*, che definisce, data una proprieta' *R*, la proprieta' inversa *R*- (*inverseOf*).

Sulle proprieta' e' possibile definire vincoli globali di cardinalita', ovvero:

- *funzionalita'*, che descrive una *proprieta' funzionale* (*FunctionalProperty*);
- *funzionalita' inversa*, che descrive una *proprieta' funzionale inversa* (*InverseFunctionalProperty*).

Infine in OWL e' possibile dichiarare che una proprieta' e' *simmetrica* (*SymmetricProperty*) o che una proprieta' e' *transitiva* (*TransitiveProperty*).

In OWL le *asserzioni* nell'ABox sono chiamate *fatti*. I due tipi di fatti piu' significativi sono (a) i fatti relativi dell'appartenenza di un individuo ad una classe ($C(a)$) o ai valori di una proprieta' di un individuo ($R(a,b)$), e (b) fatti relativi all'identita' di uno o piu' individui. Per quest'ultimo tipo di fatti, siccome OWL non assume l'unicita' dei nomi, e' possibile asserire che due nomi fanno riferimento allo stesso individuo (*sameAs*). Analogamente e' possibile sia asserire che due nomi fanno riferimento a due individui distinti (*differentFrom*) sia asserire che n individui sono tutti distinti fra loro (*AllDifferent*).

Ragionamento

In questa sezione viene presentato il concetto di *conseguenza logica* e la relazione che ha con il *ragionamento* che deve essere effettuato su una base di conoscenza. Di seguito, come compendio a questo argomento, vengono argomentate le differenze che intercorrono tra una DL ed una base di dati in riferimento alle proprieta' che sono assunte nell'una e nell'altra e come queste proprieta' indirizzano le interrogazioni sulla base di conoscenza.

La conseguenza logica ed il ragionamento

In una qualsiasi logica, quando si parla di *ragionamento* si fa sempre riferimento a ragionamenti di tipo deduttivo, ovvero quel tipo di ragionamento che porta a verificare se un enunciato o un'insieme di enunciati X e' *conseguenza logica* di una base di conoscenza KB .

Figura 14. Conseguenza logica

$A \models B$, ovvero B è *conseguenza logica* di A

In generale [WikipediaEN, articolo sull'*Entailment*], si dice che un insieme di sentenze A implica logicamente un insieme di sentenze B (o B e' conseguenza logica di A) se in ogni *modello* [7] in cui tutte le sentenze di A sono vere allora tutte le sentenze di B sono vere. Il simbolo usato per descrivere la conseguenza logica e' quello in Figura 14, «Conseguenza logica». Chiarito cio', il ragionamento e' composto da tre concetti fondamentali:

- il *compito* (*reasoning task*), caratterizzato dal tipo di enunciati che si desidera dedurre da una ba-

⁷Un *modello* e' una coppia ordinata $M = \langle U, * \rangle$, dove $U = \{u, v, \dots\}$, detto *universo*, e' un insieme non vuoto di *individui* e $*$ e' una *funzione di interpretazione* che associa: (1) un individuo a^* appartenente a U ad un nominale a ; (2) un insieme A^* contenuto in U ad ogni termine atomico A , con i termini di valore *vero* e *falso* rappresentanti rispettivamente tutto l'universo e l'insieme vuoto; (3) una relazione binaria R^* contenuta in $U \times U$ ad ogni ruolo R .

se di conoscenza;

- la *procedura*, ovvero l'algoritmo che consente la deduzione degli enunciati;
- il *servizio*, effettivamente implementato da uno strumento e messo a disposizione delle applicazioni che accedono alla base di conoscenze.

I *compiti di ragionamento* variano a seconda se sono rivolti verso le TBox o verso le ABox. Per le TBox, i compiti di ragionamento piu' significativi sono:

- *sussunzione*. Data una TBox T , stabilire se una sussunzione tra due termini C e D (dove C *sussume* D) e' conseguenza logica di T ;
- *equivalenza*. Data una TBox T , stabilire se un'equivalenza tra due termini C e D (dove C *equivala* a D) e' conseguenza logica di T ;
- *soddisfacibilita'*. Data una TBox T , stabilire se un termine C e' *soddisfacibile*, ovvero che l'insieme degli individue dell'universo che soddisfano il termine non e' vuoto;
- *disgiunzione*. Data una TBox T , stabilire se due termini C e D sono disgiunti, ovvero che l'intersezione dell'insieme degli individui dell'universo che soddisfano i due termini e' vuota.

I quattro compiti di ragionamento sopra elencati possono essere ridotti, attraverso trasformazioni logiche, o alla sola sussunzione o alla sola soddisfacibilita'. Per implementare un servizio di ragionamento per la logica $SHOIN(D_n)$ solitamente si riducono, attraverso alcune trasformazioni logiche, tutti i compiti di ragionamento alla sola soddisfacibilita', come mostrato in Figura 15, «Riduzione dei compiti di ragionamento della TBox alla soddisfacibilita'».

Figura 15. Riduzione dei compiti di ragionamento della TBox alla soddisfacibilita'

sussunzione: dimostrare $T \models C \sqsubseteq D$ equivale a dimostrare che $T \models C \sqcap \neg D$ è insoddisfacibile

equivalenza: dimostrare $T \models C \equiv D$ equivale a dimostrare che $T \models C \sqcap \neg D$ è insoddisfacibile e che $T \models \neg C \sqcap D$ è insoddisfacibile

disgiunzione: dimostrare $T \models C \sqsubseteq \neg D$ equivale a dimostrare che $T \models C \sqcap D$ è insoddisfacibile

Per la logica $SHOIN(D_n)$ si puo' formulare una *procedura di ragionamento* che prende in ingresso una TBox T ed un termine C e, in un numero finito di passi, stabilisce se C e' o non e' soddisfacibile, tenendo conto ovviamente delle definizioni terminologiche di T . Molti ragionatori tuttora in circolazione basano questa procedura sul *metodo dei tableaux*, descritto sia in [TableauxUSA] sia in [TableauxUK] [8].

I piu' significativi compiti di ragionamento relativi all'ABox sono di tre tipi:

- *instance check*. Data una base di conoscenza KB , ovvero una TBox T ed un'ABox A , e dati un termine C ed un nominale a stabilire se $C(a)$ e' conseguenza logica di KB ;
- *retrieval*. Data una base di conoscenza KB , ovvero una TBox T ed un'ABox A , e dato un termine C , tra tutti i nominali presenti in KB trovare tutti i nominali a_1, \dots, a_n tali che $C(a_k)$ e' conseguenza logica di KB ;
- *realizzazione*. Data una base di conoscenza KB , ovvero una TBox T ed un'ABox A , e dati un insieme di termini $\{C_1, \dots, C_n\}$ ed un nominale a , determinare gli m termini $\{C_{i_1}, \dots, C_{i_m}\}$ piu' specifici [9] in $\{C_1, \dots, C_n\}$ per cui $C_k(a)$ e' conseguenza logica di KB .

⁸Per una trattazione del *metodo dei tableaux* in logica proposizionale ed in logica predicativa si veda [Elementi05].

⁹Un termine C si dice piu' specifico di un termine D se C *sussume* D .

I compiti di ragionamento relativi all'ABox possono essere ricondotti al problema di stabilire la *soddisfacibilita'* di un insieme di asserzioni, come descritto in Figura 16, «Riduzione dei compiti di ragionamento dell'ABox alla soddisfacibilita'», per quel che concerne il compito di *instance check*.

Figura 16. Riduzione dei compiti di ragionamento dell'ABox alla soddisfacibilita'

$TBox, ABox \models C(a)$ se e solo se $TBox \models$ "l'unione di $ABox$ e $\{\neg C(a)\}$ " è insoddisfacibile

In linea di principio, sia un compito di *retrieval* sia un compito di *realizzazione* possono essere ricondotti ad un compito di *instance check* e, di conseguenza, ad un problema di *soddisfacibilita'*.

Le interrogazioni

Interrogare una base di conoscenza significa verificare se un enunciato e' conseguenza logica oppure no della base di conoscenza. Questa prova, o verifica, puo' essere effettuata invocando un *servizio di ragionamento*. Al contrario delle basi di dati, in una base di conoscenza non vengono assunte nozioni di chiusura, salvo al piu' l'unicita' dei nomi. Inoltre, la risposta ad un'interrogazione e' data non sulla base di cio' che e' vero in uno specifico modello ma sulla base di cio' che e' *conseguenza logica* della base di conoscenza, ovvero di cio' che e' vero in tutti i possibili modelli della base di conoscenze. Di conseguenza, tutto cio' che non e' esplicitamente dichiarato nella base di conoscenza *non e'* necessariamente falso. Per comprendere meglio questa situazione, vedere l'esempio descritto nella Figura 17, «Interrogazioni ad una base di conoscenza».

Figura 17. Interrogazioni ad una base di conoscenza

$PERSONA \sqsubseteq \top$
 $\top \sqsubseteq \forall Figlio^-. PERSONA$
 $\top \sqsubseteq \forall Figlio. PERSONA$
 $\neq (a, b, c, d, e)$
 $Figlio(a, b)$
 $Figlio(a, c)$
 $Figlio(d, e)$

Interrogazioni:

- chi ha almeno un figlio? — ? — $\exists Figlio(*) \rightarrow \{a, d\}$
- chi ha almeno due figli? — ? — $\geq_2 Figlio(*) \rightarrow \{a\}$
- chi ha almeno tre figli? — ? — $\geq_3 Figlio(*) \rightarrow \{\}$
- chi ha al più un figlio? — ? — $\leq_1 Figlio(*) \rightarrow \{\}$ (!!!!!)

Probabilmente, siccome normalmente si e' abituati a ragionare in termini di una base di dati, dall'ultima interrogazione si ci aspettava una risposta come $\{b, c, d, e\}$.

Figura 18. Interrogazione alla base di conoscenza aggiornata

$\{b, c, d, e\} \sqsubseteq_{\leq_1} Figlio$
Interrogazioni:

- chi ha al più un figlio? — ? — $\leq_1 Figlio(*) \rightarrow \{b, c, d, e\}$

Ma, come gia' ribadito, in una base di conoscenza l'assenza di informazione, in questo caso di

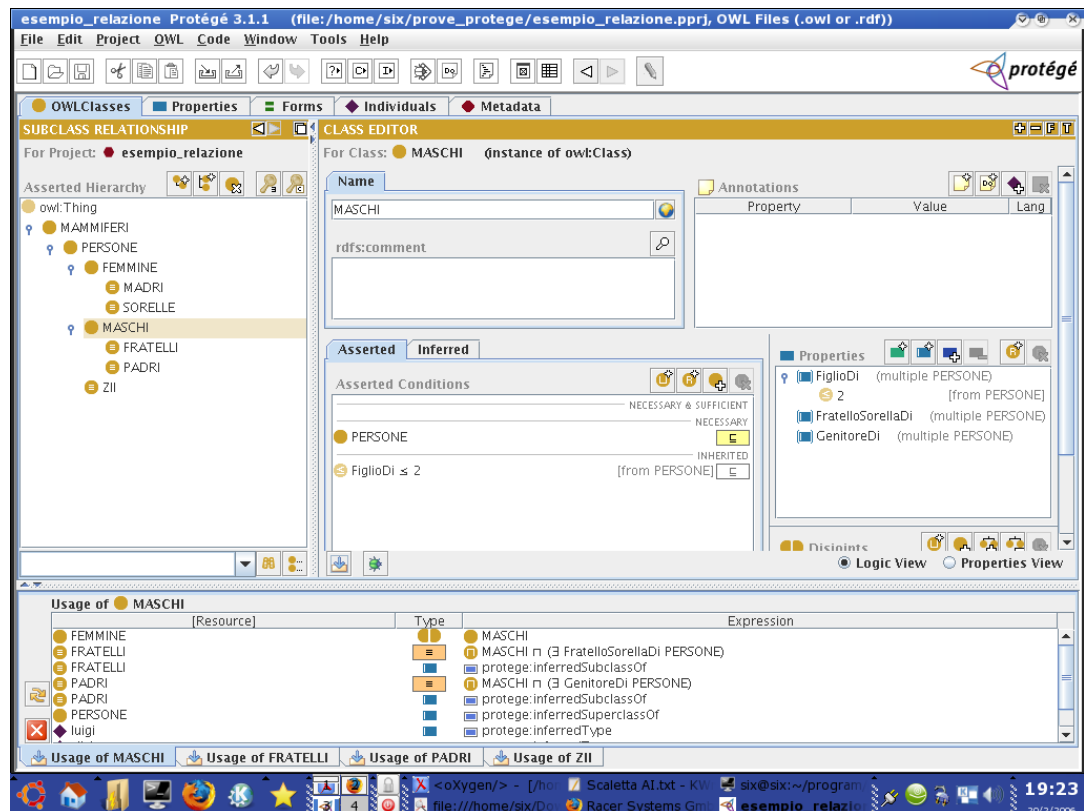
un'asserzione dall'ABox, non equivale ad assumerne la *falsita*'. Ovvero, non e' *esplicitamente* detto in nessun enunciato della base di conoscenza che i nominali *b, c, d, e* abbiano al piu' un figlio. La risposta alla domanda *chi ha al piu' un figlio* risponde con il risultato atteso se aggiungiamo nella base di conoscenza presentata in Figura 17, «Interrogazioni ad una base di conoscenza» quella della Figura 18, «Interrogazione alla base di conoscenza aggiornata». Le prime tre interrogazioni della Figura 17, «Interrogazioni ad una base di conoscenza» hanno dato lo stesso risultato che si ci poteva aspettare da una base di dati perche' in questi casi le assunzioni di chiusura sono irrilevanti.

La logica $SHOIN(D_n)$, come pure tutte le altre DL, sono compatibili con le situazioni in cui la conoscenza del mondo risulti incompleta. Di conseguenza le assunzioni di chiusura (*unicita' dei nomi, chiusura dell'universo, mondo chiuso*) tipiche delle basi di dati, che caratterizzano una conoscenza *completa* del mondo in cui si opera, sono ipotesi troppo forti per essere integrate in una DL. E' anche vero che in alcuni casi particolari, si potrebbe sapere a priori che alcune conoscenze rappresentate nella base di conoscenza sono complete almeno sotto certi aspetti. Di conseguenza potrebbe essere utile utilizzare alcune assunzioni di chiusura, come quella del *mondo chiuso* (vedere la sezione chiamata «TBox e ABox»), in riferimento ad alcuni termini (o ruoli) che si pensano come completi. Per fare cio', e' necessario introdurre un nuovo operatore *K*, da *know* tradotto come *e' noto che*, che fornisce la proprieta' del *mondo chiuso* al termine (o ruolo) che accompagna. Per maggiori dettagli su questo operatore vedere [IngCon05].

Editor e ragionatori

In questa sezione vengono elencati, associati ad altrettanti ragionatori, una serie di editor visuali di ontologie OWL. Grazie all'editor *Protege*, e' stata creata un'ontologia OWL di prova sulla quale, attraverso il ragionatore *RacerPro*, sono state eseguite alcune interrogazioni.

Figura 19. Uno screenshot di Protege (OWL)



Cosa e quali sono

Anche se non e' stata trattata nel dettaglio (vedere a tal proposito [OwlSpec]), la sintassi di OWL e'

molto complicata da scrivere ma, soprattutto, da capire. Il linguaggio OWL e' stato progettato perche' fosse possibile per dei computer, per degli applicativi, usare e manipolare l'informazione contenuta all'interno di un'ontologia OWL facilmente. Questo vantaggio per le macchine, pero', si trasforma in un completo svantaggio per degli esseri umani: un'ontologia espressa nel linguaggio OWL e' praticamente illeggibile.

Proprio per sopperire a questa estrema difficolta' di costruzione, manipolazione e lettura da parte dell'uomo, sono stati costruiti alcuni editor visuali per la progettazione di ontologie OWL. Tra i tanti:

- *SWOOP* [<http://www.mindswap.org/2004/SWOOP/>], e' un editor che ricorda molto da vicino un browser web, anche perche' la navigazione attraverso le entita' dell'ontologia e' realizzata attraverso *hyperlink*. Inoltre tiene traccia delle azioni svolte (bottoni *back* e *next*) e permette di creare dei *bookmarks*;
- *Protege* [<http://protege.stanford.edu/>], e' un editor di ontologie che puo' essere esteso con i numerosi plugin scaricabili direttamente dal sito principale dell'editor. Uno di questi plugin e' il *Protege OWL plugin* grazie al quale si puo' utilizzare *Protege* per la costruzione di ontologie OWL.

Ambedue gli editor offrono la possibilita' di integrazione con un ragionatore/dimostratore. In generale, il primo viene solitamente usato in coppia con *Pellet* [<http://www.mindswap.org/2003/pellet/>], un ragionatore basato sul metodo dei tableaux [TableauxUSA] sviluppato per la logica *SHOIQ(D_n)*, che riesce a supportare completamente tutta l'espressivita' di OWL DL, compresi i costrutti *oneOf* e *hasValue*, e permette le interrogazioni verso l'ontologia sia in SPARQL [<http://www.w3.org/TR/rdf-sparql-query/>] sia in RDQL [<http://www.w3.org/Submission/RDQL/>].

Figura 20. TBox dell'ontologia

$$\begin{aligned}
 &MAMMIFERI \sqsubseteq \top \\
 &PERSONE \sqsubseteq MAMMIFERI \\
 &PERSONE \sqsubseteq \leq_2 \text{FiglioDi} \\
 &FEMMINE \sqsubseteq PERSONE \\
 &MASCHI \sqsubseteq PERSONE \\
 &MASCHI \sqcap FEMMINE \sqsubseteq \perp \\
 &MADRI \equiv FEMMINE \sqcap \exists \text{GenitoreDi}.PERSONE \\
 &SORELLE \equiv FEMMINE \sqcap \exists \text{FratelloSorellaDi}.PERSONE \\
 &PADRI \equiv MASCHI \sqcap \exists \text{GenitoreDi}.PERSONE \\
 &FRATELLI \equiv MASCHI \sqcap \exists \text{FratelloSorellaDi}.PERSONE \\
 &ZII \equiv PERSONE \sqcap \exists \text{FratelloSorellaDi}.(\exists \text{GenitoreDi}.PERSONE) \\
 &\top \sqsubseteq \forall \text{GenitoreDi}^- .PERSONA \\
 &\top \sqsubseteq \forall \text{GenitoreDi}.PERSONA \\
 &\text{FiglioDi} \equiv \text{GenitoreDi}^- \\
 &\top \sqsubseteq \forall \text{FratelloSorellaDi}^- .PERSONA \\
 &\top \sqsubseteq \forall \text{FratelloSorellaDi}.PERSONA \\
 &\text{Tr}(\text{FratelloSorellaDi}) \\
 &\text{FratelloSorellaDi} \equiv \text{FratelloSorellaDi}^-
 \end{aligned}$$

Come *SWOOP*, anche *Protege* puo' essere integrato con un ragionatore, come *RacerPro* [<http://www.racer-systems.com/>], un ragionatore che permette di lavorare su ontologie OWL e creare interrogazioni con una sintassi *lisp-like*.

Protege e RacerPro

In realta', *RacerPro* non e' solo un ragionatore ma e' un intero sistema per la rappresentazione della conoscenza che implementa delle operazioni per il calcolo dei tableaux molto ottimizzate. L'unica

restrizione che comporta e' la logica che implementa: utilizza la *SHIQ* anziche' la *SHOIN(D_n)* utilizzata da OWL. Lo svantaggio peggiore derivato da questa scelta e' che, seppur possa gestire correttamente ontologie OWL, non e' in grado di utilizzare l'operatore *one-of*. Seppur questa sia una restrizione significativa, non e' importante per i semplici esempi di funzionamento che vengono mostrati in questo documento. Usato in accoppiata con Protege, RacerPro e' in grado di verificare la *consistenza*, o *coerenza*, di un'ontologia OWL, *classificarne* la tassonomia e calcolare i tipi di individui inferiti. Molto interessante e' l'operazione di classificazione grazie alla quale vengono evidenziati i termini non soddisfacenti, se ve ne fosse la presenza. Grazie all'editor Protege, e' stata costruita una piccola ontologia di prova, in modo da poter vedere in azione il ragionatore RacerPro attraverso delle interrogazioni, espresse in un linguaggio di sintassi lisp. Nella Figura 20, «TBox dell'ontologia» vengono descritti i termini della base di conoscenza, realizzati con l'editor Protege. La fonte di ispirazione per la progettazione della TBox utilizzata e' [IA05, nel capitolo sulla *Rappresentazione della conoscenza*]. Nella Figura 21, «ABox dell'ontologia» sono descritte le asserzioni della base di conoscenza, realizzate con l'editor Protege.

Figura 21. ABox dell'ontologia

```
≠ (silvio, giovanni, valerio, luigi, stefano, giacomo)
MASCHI(silvio), MASCHI(giovanni), MASCHI(valerio), MASCHI(luigi),
MASCHI(stefano), MASCHI(giacomo)
≠ (tiziana, rossella, chiara, elisa)
FEMMINE(tiziana), FEMMINE(rossella), FEMMINE(chiara), FEMMINE(elisa)
GenitoreDi(tiziana, silvio), GenitoreDi(giovanni, silvio) e inverse
GenitoreDi(stefano, elisa), GenitoreDi(stefano, giacomo), GenitoreDi(rossella, elisa),
GenitoreDi(rossella, giacomo) e inverse
FratelloSorellaDi(tiziana, stefano), FratelloSorellaDi(giacomo, elisa),
FratelloSorellaDi(giacomo, luigi) e inverse
```

Partendo da quest'ontologia, specificata in un file *.owl*, e' stato passato un file di interrogazioni al ragionatore *RacerPro*. La forma contratta di tali interrogazioni con le relative risposte ottenute dal ragionatore e' quella mostrata di seguito.

```
Ask: (all-individuals)
Ans: silvio,tiziana,giovanni,luigi,giacomo,
      stefano,rossella,elisa,chiara,valerio

Ask: (retrieve (?x) (?x |FEMMINE|))
Ans: tiziana,elisa,chiara,rossella

Ask: (retrieve (?x)
      (and
        (?x |FEMMINE|)
        (?x (at-least 2 |GenitoreDi|))))
Ans: rossella

Ask: (retrieve (?x)
      (and
        (?x |FEMMINE|)
        (?x (some |GenitoreDi| |MASCHI|))))
Ans: tiziana,rossella

Ask: (retrieve (?x) (?x |SORELLE|))
Ans: elisa,tiziana

Ask: (retrieve (?x ?y)
      (?x ?y |FratelloSorellaDi|))
Ans: (luigi,giacomo)
      (luigi,elisa)
      (stefano,tiziana)
      (giacomo,luigi)
      (giacomo,elisa)
      (tiziana,stefano)
```

```
(elisa,giacomo)
(elisa,luigi)
```

Nel risultato dell'ultima interrogazione il ragionatore ha dedotto che *luigi* ed *elisa* sono tra loro fratello e sorella, seppur questo non e' direttamente specificato nell'ABox. Questa deduzione, peraltro corretta, e' dovuta al fatto che la proprieta' OWL *FratelloSorellaDi* e' stata definita come *transitiva* e di conseguenza, siccome *giacomo* e' fratello sia di *elisa* sia di *luigi*, necessariamente per transitivita' *luigi* e *elisa* sono fra loro fratello e sorella.

Conclusioni

In questo documento sono state affrontate alcune problematiche legate al *web semantico*, con particolare riferimento al linguaggio per la definizione di ontologie OWL, proposto dal W3C [<http://www.w3.org>]. Di questo linguaggio sono stati trattati prima alcuni aspetti legati alla *logica descrittiva* che OWL implementa, conosciuta col nome di *SHOIN(D_n)*, poi aspetti di natura sintattico/strutturale del linguaggio stesso, relativi alla traduzione degli elementi che tradizionalmente caratterizzano la logica descrittiva presa in considerazione.

Si e' scelto di mantenere per tutto il documento la notazione offerta dalla logica descrittiva anche in ambito OWL, non portando esempi di ontologie OWL seppur queste ontologie siano effettivamente state scritte per mostrare gli esempi contenuti nella sezione chiamata «Protege e RacerPro». Questa scelta e' giustificata dal fatto che la complessita' sintattica del linguaggio ma soprattutto la non facile lettura del codice prodotto avrebbe appesantito inutilmente la lettura. Cio' non toglie che esistono molti esempi di ontologie OWL presenti sul web. Un buon punto di partenza e' il sito ufficiale di OWL [<http://www.w3.org/2004/OWL/>], dal quale si snodano una serie di link da cui si possono reperire la maggior parte delle informazioni per questo linguaggio, ontologie comprese. Tra questi link, ricordiamo SWOOGLE [<http://www.swoogle.org/>], una sorta di motore di ricerca per tutto cio' che in internet e' web semantico, SchemaWeb [<http://www.schemaweb.info/default.aspx>], un contenitore di documenti RDF Schema e OWL, e Owl Tutorial [<http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/>], in cui sono illustrati alcuni esempi di ontologie e come queste sono state costruite. Per avere altre informazioni sulle logiche descrittive, il miglior sito di riferimento, segnalato per altro anche dal W3C, e' quello di Enrico Franconi [<http://www.inf.unibz.it/~franconi/dl/course/>], direttore, presso l'universita' libera di Bolzano [<http://www.unibz.it/>], dell'*European Masters Program in Computational Logic*.

Bibliografia

- [Elementi05] *Elementi di logica matematica*. Alberto Marcone. 2005/2006. Gli appunti del corso sono scaricabili dal sito dell'universita' di Udine [<http://www.dimi.uniud.it/marcone/EILogica.html>].
- [LogInf97] *Logica a informatica*. Andrea Asperti e Agata Ciabattoni. 1997. McGraw-Hill.
- [IngCon05] *Corso di ingegneria della conoscenza*. Marco Colombetti. 2004/2005. Gli appunti del corso sono scaricabili dal sito del Politecnico di Milano [http://www.elet.polimi.it/upload/colombet/IC_2005/index.html].
- [IA05] *Intelligenza Artificiale*. Un approccio moderno. Stuart Russell e Peter Norving. 2005. Prentice Hall.
- [OwlSpec] *Web Ontology Language (OWL)*. Il manuale e' consultabile dal sito del W3C [<http://www.w3.org/2004/OWL/>].
- [OwlUseCase] *OWL Web Ontology Language Use Cases and Requirements*. Il manuale e' consultabile dal sito del W3C [<http://www.w3.org/TR/webont-req/>].
- [SemWeb01] *The Semantic Web*. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Tim Berners-Lee, James Hendler, e Ora Lassila. 2001. L'articolo e' visibile all'indirizzo del Scientific American [<http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF2>].

1&pageNumber=1&catID=2].

[TableauxUK] *A tableaux decision procedure for SHOIQ*. Ian Horrocks e Ulrike Sattler. L'articolo e' scaricabile dal sito della School of Computer Science [<http://www.cs.man.ac.uk/~sattler/publications/shoiq.pdf>] dell'universita' di Manchester (UK).

[TableauxUSA] *A tableaux-based decision procedure for explanations in SHOIN*. Aditya Kalyanpur, Bijan Parsia, Bernardo Cuenca Grau, e Evren Sirin. L'articolo e' scaricabile dal sito della MINDSWAP [<http://www.mindswap.org/papers/tracingKR.pdf>] (Maryland Information and Network Dynamics Lab Semantic Web Agents Project).

[WikipediaEN] *Wikipedia (inglese)*. Wikipedia [http://en.wikipedia.org/wiki/Main_Page] e' un'enciclopedia online, multilingue, a contenuto libero, redatta in modo collaborativo da volontari e sostenuta dalla Wikimedia Foundation, un'organizzazione non-profit.

[WikipediaIT] *Wikipedia (italiana)*. Wikipedia [http://it.wikipedia.org/wiki/Pagina_principale] e' un'enciclopedia online, multilingue, a contenuto libero, redatta in modo collaborativo da volontari e sostenuta dalla Wikimedia Foundation, un'organizzazione non-profit.