Politecnico di Milano
Facoltà di ingegneria Milano Leonardo
Corso di Ingegneria della conoscenza e sistemi esperti

Appunti dalle lezioni
Parte IV: capitolo 12

Marco Colombetti

Indice (parte IV)

Rappresentazioni dichiarative e reti semantiche

12.1 Rappresentazioni procedurali e rappresentazioni dichiarative

Nei capitoli precedenti abbiamo visto come degli assiomi logici possano essere tradotti in un linguaggio a regole di produzione come il CLIPS. Occorre notare che la traduzione degli assiomi in regole è solo parziale. Consideriamo ad esempio l'assioma

(12.1) $\forall x\ (Delfino(x) \rightarrow Mammifero(x) \wedge Marino(x))$,

che dice che un delfino è un mammifero marino. La più ovvia traduzione in CLIPS,

```
(defrule R1
  (FACT Delfino ?x)
 =>
  (assert (FACT Mammifero ?x)
          (FACT Marino ?x))
)
```

coglie soltanto un aspetto dell'assioma. Più precisamente, la regola `R1` rappresenta solo uno dei possibili *usi* dell'assioma nel processo deduttivo. Infatti, la regola consente di dedurre il conseguente di 12.1 una volta che sia noto l'antecedente. Tuttavia, l'assioma 12.1 ha altri usi possibili. Ad esempio, permette di:

- creare il sotto-obiettivo di dimostrare che *x* è un delfino se si ha l'obiettivo di dimostrare che *x* è un mammifero;
- creare il sotto-obiettivo di dimostrare che *x* è un delfino se si ha l'obiettivo di dimostrare che *x* è un animale marino;
- dedurre che *x* non è un delfino se è noto che *x* non è un mammifero;
- dedurre che *x* non è un delfino se è noto che *x* non è un animale marino.

Ognuno di questi usi dell'assioma richiede una diversa traduzione in CLIPS. Ad esempio:

```
(defrule R2
  (GOAL Mammifero ?x)
 =>
  (assert (GOAL Delfino ?x))
)
```

e così via.

Da queste considerazioni concludiamo che una regola CLIPS non traduce un assioma logico, ma piuttosto *un particolare uso* di un assioma logico. Per questo motivo, le regole di produzione vengono considerate un metodo di *rappresentazione procedurale* della conoscenza. Gli assiomi logici, al contrario, rappresentano elementi di conoscenza indipendentemente dall'uso che se ne vuole fare, e sono quindi considerati un metodo di *rappresentazione dichiarativa* della conoscenza.

In generale, una rappresentazione procedurale è più debole della corrispondente rappresentazione dichiarativa. Tuttavia, le rappresentazioni procedurali possono essere utilizzate in modo più efficiente perché non codificano solo un elemento di conoscenza, ma anche il modo di utilizzarlo.

È vero tuttavia che a volte si desidera utilizzare direttamente una rappresentazione dichiarativa, proprio perché non si vuole decidere a priori come la conoscenza sarà utilizzata. In linea di principio è possibile rappresentare le conoscenze direttamente in logica e utilizzare un dimostratore generale di teoremi. Tuttavia, se l'applicazione è appena un po' complessa questa strada porta a sistemi disperatamente inefficienti.

Una via alternativa consiste nell'utilizzare in modo dichiarativo un sottolinguaggio della logica del primo ordine che sia ancora abbastanza espressivo, ma sufficientemente limitato da ammettere elaborazioni efficienti. Un sottoliguaggio di questo tipo è costituito dalle reti semantiche, proposte per la prima volta da Quillian (1968), più rigorosamente definite da Woods (1975) e da allora largamente utilizzate nelle applicazioni d'intelligenza artificiale.

12.2 Le reti semantiche e la gerarchia 'isa'

Dal punto di vista formale, una rete semantica è un grafo costituita da nodi ed archi etichettati da simboli. A seconda dei casi, un nodo rappresenta un concetto, una parola, un insieme o un individuo; un arco rappresenta una relazione binaria fra le entità rappresentate dal nodo sorgente e dal nodo bersaglio.

Consideriamo un primo esempio. La rete della figura 12.1 è costituita da due nodi connessi da un arco **isa**, il cui significato sarà spiegato fra poco. Intuitivamente, la rete dice che "un gatto è un mammifero" (in inglese, "a cat *is a* mammal", da cui il simbolo '**isa**').

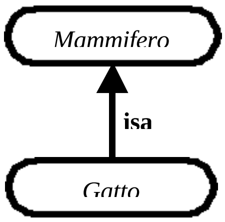

Figura 12.1 *Una semplice rete semantica.*

Di questa rete sono possibili diverse interpretazioni intuitive, e in particolare:

- il *concetto* di gatto è una *specializzazione* del concetto di mammifero;
- il *concetto* di mammifero è una *generalizzazione* del concetto di gatto;
- il *sostantivo* 'gatto' è un *iponimo* del sostantivo 'mammifero';
- il *sostantivo* 'mammifero' è un *iperonimo* del sostantivo 'gatto';
- l'*insieme* dei gatti è un *sottoinsieme* dell'insieme dei mammiferi;
- l'*insieme* dei mammiferi è un *sovrainsieme* dell'insieme dei gatti.

È opportuno assegnare alla rete anche una semantica formale. Ciò può essere fatto in due modi: *direttamente*, con un metodo analogo a quello che abbiamo utilizzato a suo tempo per dare una semantica agli enunciati logici; o *indirettamente*, assegnando una regola di traduzione della rete in un linguaggio formale che abbia già una semantica. Seguiremo la seconda via, traducendo la rete in un linguaggio predicativo del primo ordine.

Per prima cosa, associamo a ogni nodo un predicato a un argomento; nel nostro caso, *Mammifero*(–) e *Gatto*(–). Un arco **isa** fra due nodi C_1 e C_2 si traduce ora nel seguente modo:

$$C_1 \ \mathbf{isa} \ C_2 \quad \longrightarrow \quad \forall x\, (C_1(x) \rightarrow C_2(x)).$$

Quindi, la rete della figura 12.1 è equivalente all'enunciato:

$$\forall x\ (Gatto(x) \rightarrow Mammifero(x)).$$

In base a questa semantica, la relazione **isa** risulta transitiva. Infatti, da $\forall x\ (C_1(x) \rightarrow C_2(x))$ e da $\forall x\ (C_2(x) \rightarrow C_3(x))$ è possibile dedurre $\forall x\ (C_1(x) \rightarrow C_3(x))$, e pertanto:

$$C_1 \textbf{ isa } C_2,\ \ C_2 \textbf{ isa } C_3 \models C_1 \textbf{ isa } C_3.$$

Ritornando alla rappresentazione grafica, possiamo ricavare un nuovo arco **isa** concatenando due archi **isa** preesistenti. Nella figura 12.2 questo fatto è descritto utilizzando una linea tratteggiata per l'arco "dedotto" e due linee continue per gli archi preesistenti; nel seguito, per semplicità, indicherò gli archi **isa** con una freccia grossa e ometterò l'etichetta '**isa**'.

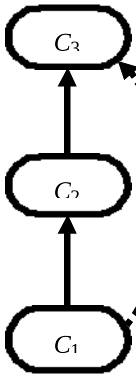

Figura 12.2 *Concatenazione di archi* **isa**.

La possibilità di dedurre nuovi archi isa per concatenazione è molto importante perché consente una notevole economia rappresentativa. Infatti, ci si può limitare a rappresentare esplicitamente gli archi **isa** che collegano due nodi in modo diretto (ovvero, senza nodi intermedi); tutti gli altri archi **isa** possono essere dedotti quando occorre mediante concatenazione. Ad esempio, se diciamo che un gatto è un carnivoro e che un carnivoro è un mammifero, il fatto che un gatto sia un mammifero può essere dedotto (fig. 12.3). Le rappresentazioni strutturate in questo modo sono dette *gerarchie* **isa**. Gerarchie di questo tipo sono ben note in informatica, perché vengono utilizzate sia nella programmazione orientata agli oggetti sia nei modelli concettuali di basi di dati.

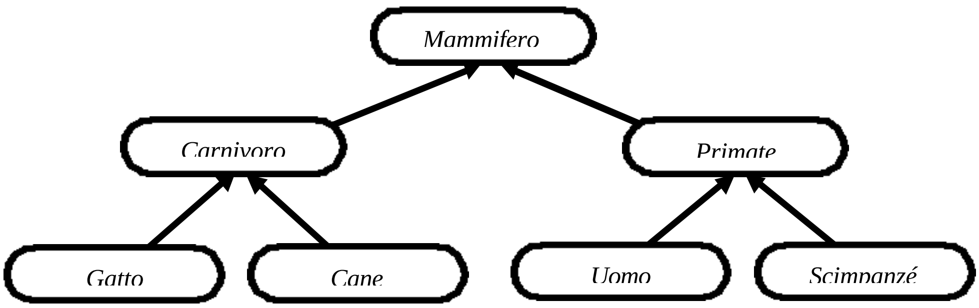

Figura 12.3 *Una gerarchia* **isa**.

È possibile interpretare una rete **isa** anche in modo diverso da quanto visto sopra. Ad esempio, si possono interpretare i simboli associati ai nodi come nomi di insiemi. In questo caso, un arco C_1 **isa** C_2 significa semplicemente: $C_1 \subseteq C_2$. Alternativamente, si possono considerare i simboli associati ai nodi come parole di un linguaggio (ad esempio dell'italiano). In tal caso, un arco C_1 **isa** C_2 dice che la parola C_1 è un iponimo di C_2 (o, equivalentemente, che C_2 è un iperonimo di C_1).

Analogamente a quanto avviene nella programmazione a oggetti, si può avere necessità di inserire in una gerarchia **isa** anche nodi che rappresentano individui specifici. Ciò può essere fatto introducendo un nuovo tipo di arco, **instOf** (abbreviazione di *instance of*); per maggiore chiarezza, inoltre, può essere utile rappresentare i nodi-individuo con una grafia diversa. Ad esempio, la figura 12.4 dice che Pisolo è un particolare gatto (per convenzione, la freccia doppia rappresenta una relazione **instOf**).

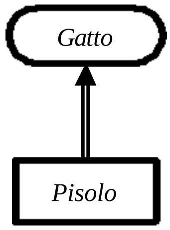

Figura 12.4 *Una rete semantica con individui.*

L'interpretazione logica di una rete del genere è semplice. I nodi-individuo vengono rappresentati come una costante; se a è un nodo del genere abbiamo allora:

$$a \ \mathbf{instOf}\ C \quad \longrightarrow \quad C(a).$$

Questa semantica ci dice che:

$$a \ \mathbf{instOf}\ C_1,\ \ C_1 \ \mathbf{isa}\ C_2 \models a \ \mathbf{instOf}\ C_2.$$

Anche l'interpretazione insiemistica è ovvia: a **instOf** C significa semplicemente che $a \in C$.

12.3 Altre relazioni

Le relazioni **isa** e **instOf** rappresentano due distinti significati del verbo *essere* (vedi: Fufi *è* un gatto, e i gatti *sono* mammiferi). Altre tipi di relazioni sono spesso introdotte linguisticamente con il verbo *avere*. Ad esempio, possiamo dire che ogni cane *ha* un *padrone*, che è una *persona*. Un enunciato di questo genere si può rappresentare con la rete della figura 12.5.

Figura 12.5 *Ogni cane ha un padrone.*

Anche in questo caso possiamo dare una semantica rigorosa alla rete fornendo una regola di traduzione nel linguaggio predicativo. Dati due nodi C_1 e C_2 e una relazione R da C_1 a C_2:

$$C_1 \ \exists R \ C_2 \quad \longrightarrow \quad \forall x\ (C_1(x) \rightarrow \exists y\ (C_2(y) \wedge R(x,y))).$$

Quindi la rete della figura 12.5 corrisponde a:

$$\forall x\ (Cane(x) \rightarrow \exists y\ (Persona(y) \wedge HaPadrone(x,y))).$$

Nell'uso, il quantificatore '∃' sull'arco della rete viene generalmente omesso. Eventuali restrizioni sulle cardinalità possono essere rappresentate introducendo gli opportuni quantificatori. Ad esempio, nella logica predicativa del primo ordine è possibile definire il quantificatore $\exists!$, che significa “esiste esattamente un...”. La definizione di questo quantificatore è la seguente:

$$\exists! x\, j \quad =_{\text{def}} \quad \exists x\, j \wedge \forall y\, (j(y/x) \rightarrow x = y).$$

Ovvero: esiste esattamente un x tale che j, per definizione, significa che

- esiste almeno un x tale che j, e che inoltre
- per ogni y, se j vale dopo aver sostituito la variabile x con la variabile y, allora x è uguale a y.

Utilizzando questo quantificatore è possibile rappresentare l’enunciato “ogni cane ha esattamente una persona come padrone” (fig. 12.6).

Figura 126 *Ogni cane ha esattamente un padrone.*

Spesso il verbo avere viene anche utilizzato per associare un *attributo* a un’entità. Ad esempio, diciamo che ogni oggetto fisico *ha un peso*. La differenza fra questo l’enunciato “ogni cane ha un padrone” e l’enunciato “ogni oggetto ha un peso” è che nel primo caso vengono messe in relazione due entità (un cane e una persona), mentre nel secondo caso viene associato a un’entità un *attributo scalare* (variabile nel campo dei numeri reali positivi). Rappresentiamo i campi di scalari con blocchi arrotondati delimitati da una linea sottile. L’enunciato in questione è allora rappresentato dalla rete della figura 12.7, la cui semantica formale è:

$$\forall x\, (OggFisico(x) \rightarrow \exists! y\, (Real{+}(y) \wedge HaPeso(x,y))).$$

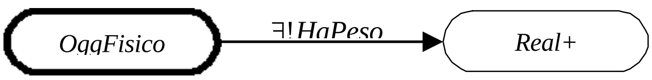

Figura 12.7 *Ogni cane ha esattamente un padrone.*

Così come un’entità può avere specifici individui connessi con un’arco instOf, i campi di scalari possono avere specifici *valori* connessi con un arco **valOf**. Ad esempio, nella figura 12.8 è rappresentato l’enunciato “l’oggetto fisico a pesa 12.5”.

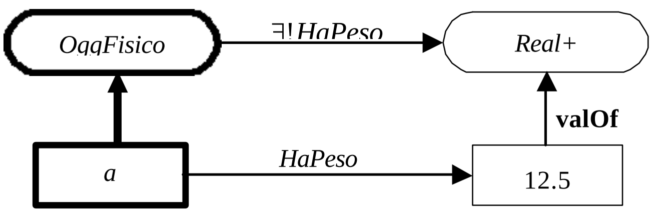

Figura 12.8 *L’oggetto fisico a pesa 12.5.*

La semantica completa della rete in figura 12.8 è la seguente:

$$\forall x\, (OggFisico(x) \rightarrow \exists! y\, (Real{+}(y) \wedge HaPeso(x,y))),$$

$$OggFisico(a),$$

$$Real{+}(12.5),$$

$HaPeso(a,12.5)$.

Le proprietà associate a un nodo tramite una relazione *R* vengono *ereditate* dai nodi lungo le gerarchie **isa**. Ciò significa che:

$C_2\ R\ C_3,\ \ C_1$ **isa** $C_2 \models C_1\ R\ C_3$.

Questo fatto è una conseguenza logica delle definizioni semantiche introdotte finora. Analogamente, gli individui ereditano le proprietà lungo l'arco **instOf** (vedi fig. 12.9).

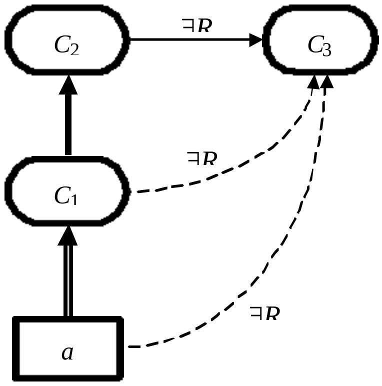

Figura 12.9 *Eredità di una proprietà.*

12.4 La gerarchia delle parti

Fra le relazioni R del tipo visto nel paragrafo precedente ne esistono alcune particolarmente importanti, che danno luogo a gerarchie diverse dalla gerarchia **isa**. Una di queste è la relazione che connette un tutto alle sue parti, siano esse parti fisiche, parti astratte o sostanze componenti. Ad esempio, possiamo essere interessati al fatto che la base, il corpo e il capitello sono parti di una colonna, e che l'abaco e l'echino sono a loro volta parti di un capitello (fig. 12.10).

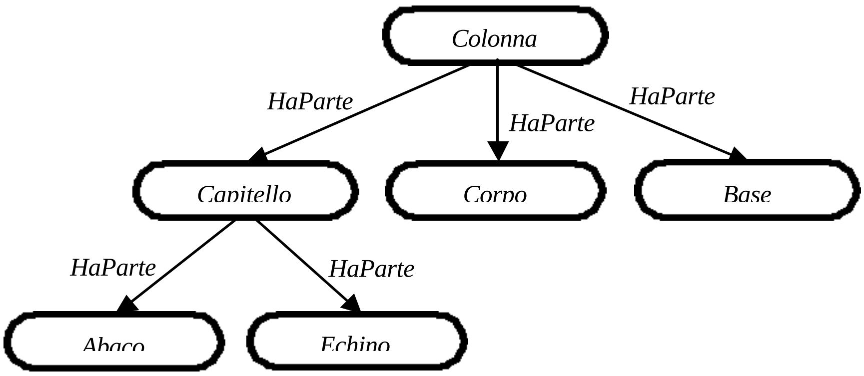

Figura 12.10 *Una gerarchia delle parti*.

La relazione *HaParte* è una relazione come un'altra, per cui:

$$C_1\ \exists HaParte\ C_2 \quad \longrightarrow \quad \forall x\ (C_1(x) \rightarrow \exists y\ (C_2(y) \wedge HaParte(x,y))).$$

La relazione *HaParte* ha una certa somiglianza con la relazione **isa**; come questa, infatti, è transitiva:

$$C_1 \,\exists HaParte\, C_2,\ \ C_2 \,\exists HaParte\, C_3 \ \models\ C_1 \,\exists HaParte\, C_3.$$

Per questo motivo, nuovi archi *HaParte* possono essere dedotti per concatenazione da archi preesistenti dello stesso tipo. Inoltre, si possono formare vere e proprie *gerarchie delle parti* (come nella fig. 12.5). Per accentuare la somiglianza con la relazione **isa**, le frecce vengono spesso tracciate dalla parte al tutto; in questo caso la relazione viene denominata *ParteDi* (o *PartOf*). Occorre però fare molta attenzione alla semantica di questa relazione, che non segue la regola generale vista in precedenza:

$$C_1\ ParteDi\ C_2 \quad \longrightarrow \quad \forall x\ (C_2(x) \rightarrow \exists y\ (C_1(y) \wedge ParteDi(y,x))).$$

Lungo una gerarchia delle parti non vale in generale l'eredità delle proprietà: ad esempio, una colonna ha forma allungata, ma alcune sue parti (come la base e il capitello) non hanno questa proprietà. Per alcuni tipi di proprietà, tuttavia, l'eredità vale: ad esempio, se un oggetto si trova in un luogo x le sue parti fisiche si trovano nello stesso luogo.

12.5 La reificazione

Le reti semantiche rappresentano la realtà mediate relazioni binarie definite fra entità e attributi scalari. Le entità non corrispondono soltanto a cose che possono essere intuitivamente considerate come oggetti fisici. Molto spesso, anzi, le entità vengono utilizzate per rappresentare concetti astratti, eventi, azioni e così via. Il processo che porta a concettualizzare questi aspetti della realtà come entità è detto *reificazione*.

Supponiamo ad esempio di voler rappresentare il contenuto della frase "Filippo regala un libro a sua sorella". La frase rappresenta *una specifica azione* (il cui tipo è *Regalare*), *eseguita da* un individuo di *nome* "*Filippo*" su un individuo di tipo *Libro* a beneficio di un'altro individuo, che è in relazione *SorellaDi* con Filippo. Tutto ciò può essere rappresentato dalla rete della figura 12.11.

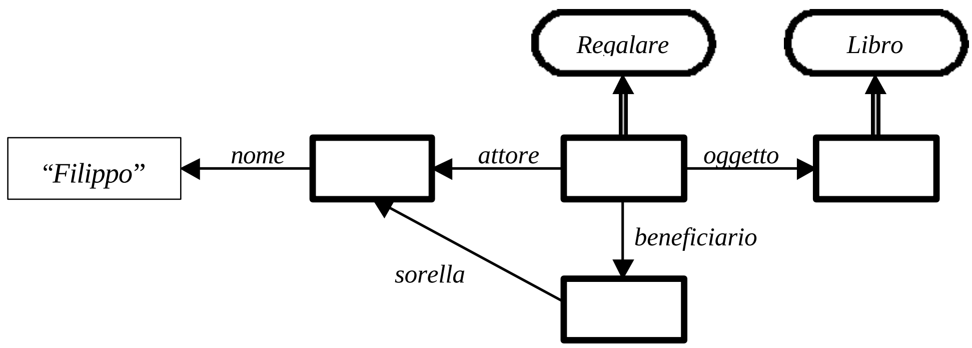

Fig. 12.11 *Filippo regala un libro a sua sorella.*

La prima cosa da osservare è la comparsa di un certo numero di relazioni binarie (*attore, oggetto, beneficiario*) dal nodo che rappresenta l'azione ai nodi che rappresentano gli individui concreti coinvolti nell'azione. Questo tipo di relazioni sono comunemente dette *ruoli*. La seconda cosa da osservare è che in base a conoscenze generali della lingua e del mondo possiamo integrare la rete con ulteriori elementi "sottointesi" (ovvero, non espressi esplicitamente nella frase). Ad esempio, sappiamo che Filippo è un uomo (perché "Filippo" è un nome maschile) e che sua sorella è una donna. A loro volta i concetti di *Regalare, Libro, Uomo* e *Donna* rimandano ad altri nodi della rete che costituisce le conoscenze di un agente. In questo modo è possibile immergere la rappresentazione del significato della frase in una più ampia rete preesistente (vedi ad es. la fig. 12.12) e quindi "comprenderne" il significato in senso più ampio.

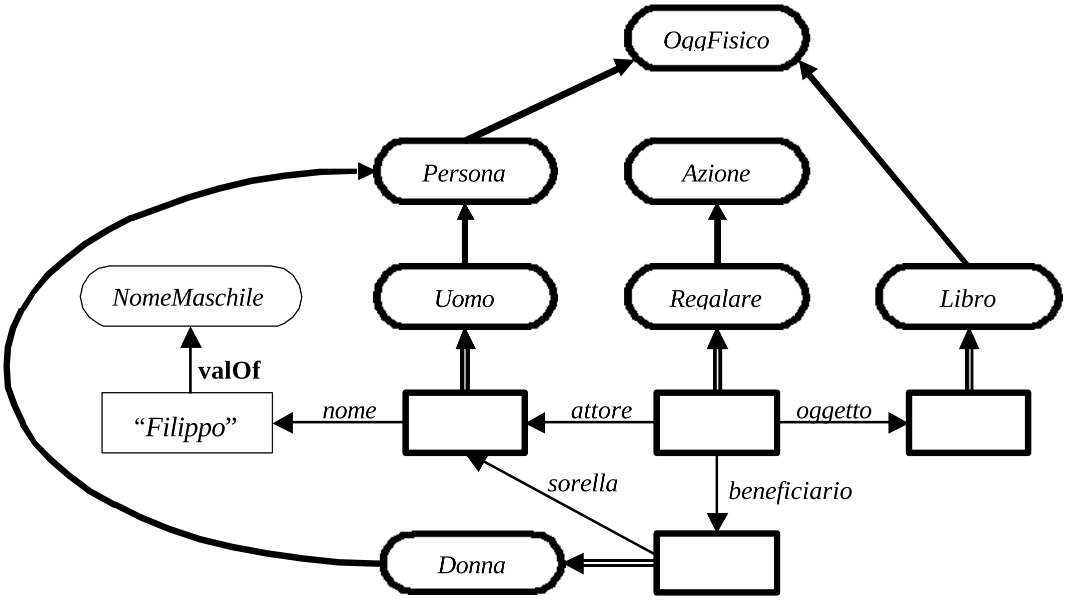

Fig. 12.12 *Immersione del significato di una frase nella conoscenza generale*.

Riferimenti bibliografici (parte IV)

Quillian, M. R. (1968). Semantic memory. In M. Minsky, ed., *Semantic information processing*, MIT Press, Cambridge, MA, 227–270.

Woods, W. A. (1975). What's in a link: Foundations for semantic networks. In D. G. Bobrow and A. Collins, eds., *Representation and understanding: Studies in cognitive science*, Academic Press, New York, 35–82