

Ragionamento automatico

1. Introduzione

Il ragionamento automatico, integrato negli editor di ontologie come Protégé sotto forma di plugin, svolge importanti funzioni per l'editing di ontologie. Non solo, infatti, permette all'utente di verificare le inferenze possibili sull'ontologia, e in particolare sugli individui che ne costituiscono la A-Box, ma supporta inoltre la progettazione e l'editing dell'ontologia stessa come insieme di classi e proprietà (T-Box).

Di seguito, si esamineranno alcune funzioni del ragionamento automatico, relative alle fasi di cui sopra (progettazione, editing e inserimento di individui). Dato che i software per il ragionamento automatico su ontologie (*ontology reasoners*) differiscono tra loro per la tipologia di inferenze compiute, è sempre utile testare la propria ontologia utilizzando più di un reasoner. Gli esempi seguenti sono stati ottenuti utilizzandone uno dei più comuni, cioè Pellet (Sirin et al. 2007).

2. Ragionamento sulle classi

Il ragionamento automatico può essere utilizzato durante la fase di modellazione per verificare che la gerarchia delle classi sia coerente con la definizione delle classi. Il reasoner è in grado di individuare le inconsistenze e di risolverle: nell'interfaccia di Protégé il risultato del ragionamento automatico sulle classi è presentato all'utente come "gerarchia (delle classi) inferita", accessibile tramite uno specifico *tab* nel pannello dedicato alle classi.

Inferenze sulla gerarchia delle classi

Per esempio, creiamo due classi "sorelle", cioè figlie della stessa sottoclasse, di cui però una sussume logicamente l'altra:

- La proprietà *haFiglio* ha come dominio la classe "Persona" e come range la stessa classe "Persona";
- La classe "Genitore" è vincolata ad avere almeno un figlio;
- La classe "Genitore_di_due" è vincolata a averne almeno due;
- Le due classi sono definite come sottoclassi di "Persona".

Si noti che, come visto in precedenza (e come facilmente verificabile dal pannello che descrive la singola classe nell'interfaccia), in Protégé creare una classe come sottoclasse di un'altra attraverso l'interfaccia grafica equivale a dichiarare tale classe come *subClassOf*, cioè sottoclasse, della prima.

A fronte della dichiarazione di classi e loro proprietà sopra descritta, il reasoner colloca correttamente la classe sussunta come sottoclasse di quella più generale, cioè inferisce che la classe "Genitore_di_due" è sottoclasse di Genitore: infatti, "Genitore di due" è un tipo specifico di genitore, ottenuto restringendo il range della proprietà *haFiglio* a una specifica cardinalità minima.

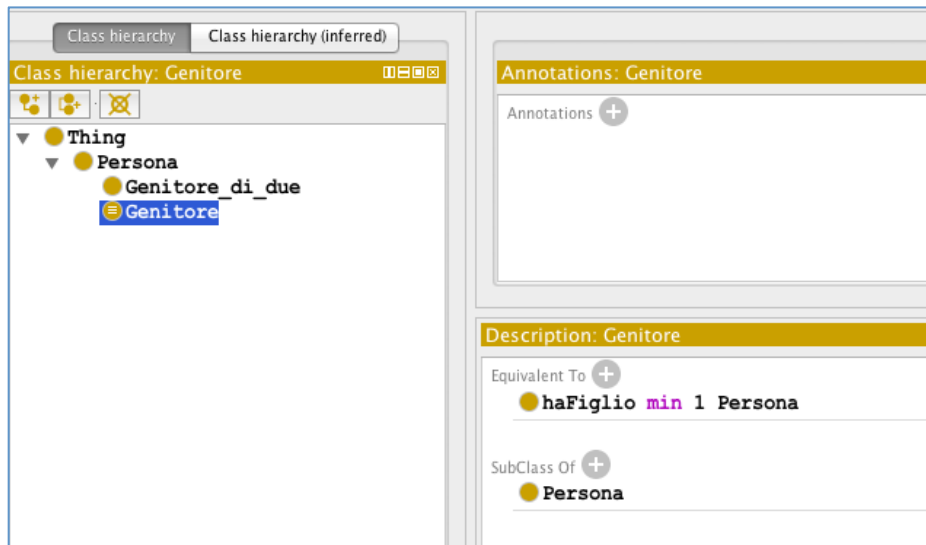


Figura 1: il pannello della gerarchia delle classi.

Nel pannello delle classi, la gerarchia inferita mostra che la classe "Genitore_di_due" è stata spostata sotto la classe "Genitore".

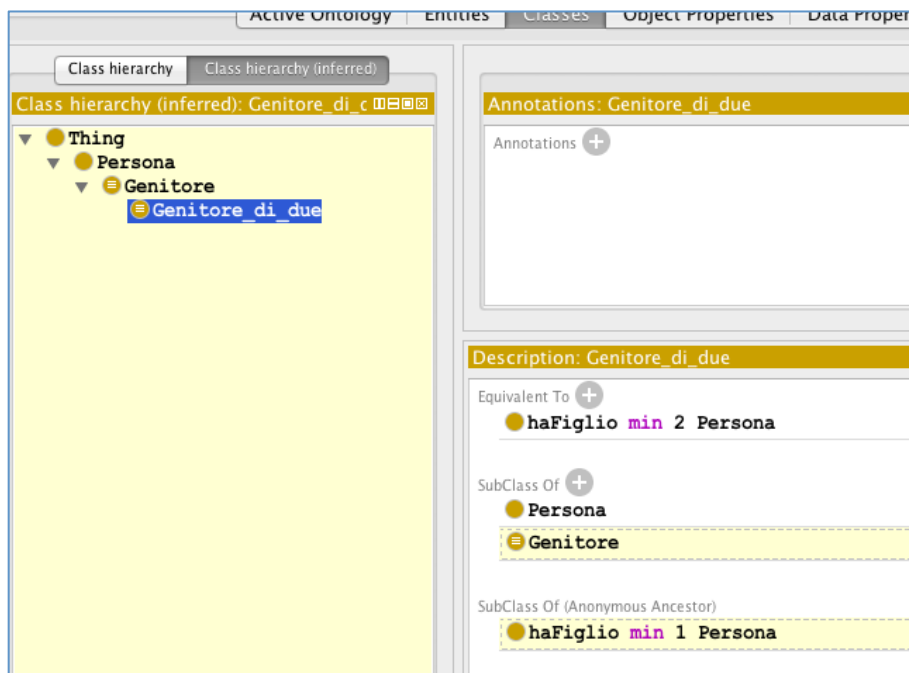


Figura 2: il tab della gerarchia inferita dal reasoner nel pannello della gerarchia delle classi.

Si noti che l'inferenza sulla gerarchia delle classi avviene anche in mancanza di individui nell'ontologia, cioè quando la A-box è vuota (si veda capitolo precedente).

Il reasoner, infatti, è in grado di inferire che non esiste un modello in cui tutti gli individui che fanno parte della classe "Genitore di due" non siano contemporaneamente anche membri della classe "Genitore".

Gerarchia inconsistente?

Dalla capacità del reasoner di fare inferenze sulla gerarchia delle classi discende una buona pratica della modellazione: verificare periodicamente che la gerarchia delle classi non sia né inconsistente né diversa da quella dichiarata. Mentre nel primo caso il reasoner segnala l'errore attraverso l'interfaccia di Protégé, rendendo di fatto impossibile non accorgersi dell'errore, nel secondo caso è necessario che l'utente verifichi la configurazione della gerarchia 'inferita' attraverso il *tab* apposito nel pannello che mostra la gerarchia delle classi (Fig. 2).

2. Ragionamento sugli individui

La classificazione automatica è la funzionalità del reasoner che consiste nel collocare automaticamente un individuo nella classe implicata dalle sue caratteristiche.

Si consideri l'individuo Giovanni, descritto come segue (Fig. 3):

- *haFiglio* Carlo
- è membro della classe "Persona"

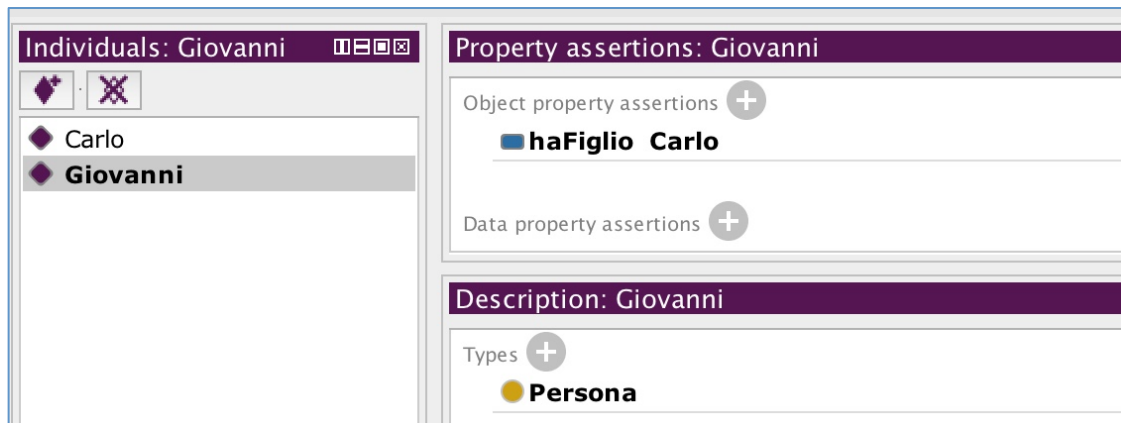


Figura 3: descrizione di un individuo tramite l'assegnazione di proprietà (in alto a destra) e tipo (in basso a destra).

Il *reasoner* colloca l'individuo Giovanni nella classe "Genitore", classe più specifica di quella "Persona" in cui era stato collocato, dato che Giovanni *haFiglio* Carlo (Fig. 4).

L'inferenza scatta perché la classe "Genitore" è una classe definita, cui sono associate condizioni necessarie e sufficienti, che consistono nell'avere almeno un figlio.

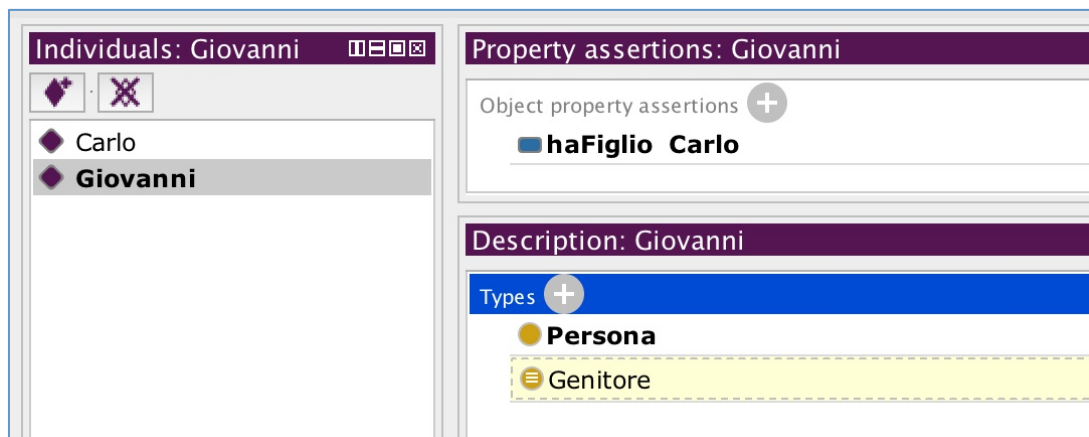


Figura 4: classificazione automatica di un individuo in Protégé: la classe aggiunta nello slot Type appare in giallo.

Infatti la classe "Genitore" è definita come la sottoclasse di "Persona" tale che i suoi membri hanno almeno un figlio (o meglio, sono nella relazione *haFiglio* con almeno un membro della classe Persona), come mostrato in Fig. 5.



Figura 5: descrizione di una classe definita in Protégé.

Appartenenza degli individui alle classi e inconsistenza.

L'utilizzo della disgiunzione tra le classi, utile a introdurre un vincolo sull'appartenenza degli individui alle classi, è una fonte molto comune di incostistenza. Si tratta di un elemento importante del linguaggio OWL (*disjoint with*), che tuttavia deve essere gestito con attenzione per evitare di incorrere in classificazioni contraddittorie.

Per esempio, come si vede nella figura, la classe "Maggiorenni" e la classe "Minorenni" sono state dichiarate come disgiunte, quindi non possono avere elementi in comune. In altre parole, un individuo non può essere membro contemporaneamente di entrambe.

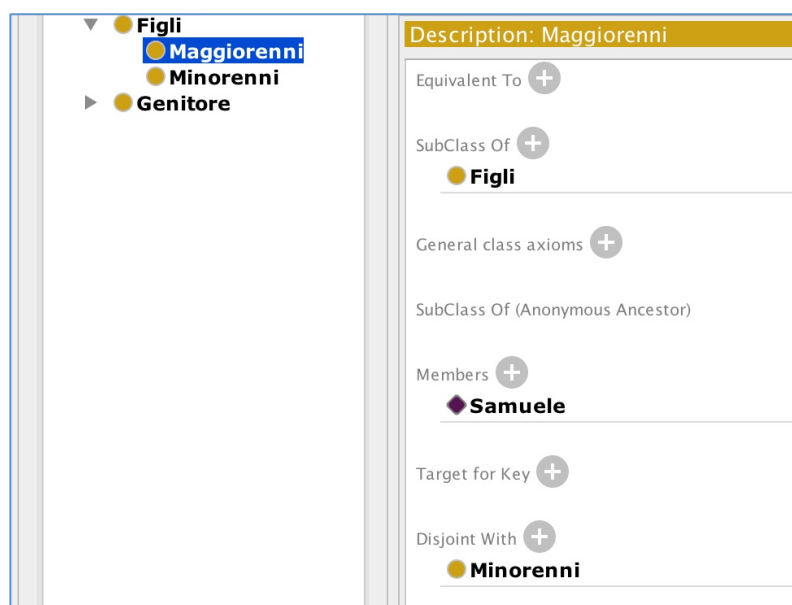


Figura 6: utilizzo della disgiunzione tra classi (in basso a destra): nell'esempio, le classi Maggiorenni e Minorenni sono disgiunte.

Dato che le classi "Maggiorenni" e "Minorenni" sono disgiunte, ne discende che non è possibile collocare lo stesso individuo in entrambe. Se lo stesso individuo, Samuele, viene dichiarato membro di entrambe le classi, il reasoner valuterà l'ontologia come logicamente inconsistente, fornendo normalmente un messaggio come il seguente:

"Reason for inconsistency: individual Samuele is forced to belong to class Minorenni and its complement"

Disgiunzione e inconsistenza.

Le conseguenze dell'uso della disgiunzione talvolta sono di non immediata comprensione e diventano evidenti solo tramite il ricorso al ragionamento automatico.

Si consideri il seguente esempio.

- La classe "GenitoreConFigliMaggiorenni" è una classe definita, ottenuta a partire dalla classe "Genitore" vincolando il range della proprietà "haFigli" ai soli individui della classe Maggiorenni; in altre parole, questa classe raggruppa i genitori che hanno solo figli maggiorenni. Si noti che il vincolo sulla proprietà haFiglio utilizza l'operatore *only* per la quantificazione, corrispondente al quantificatore universale (Fig. 7).
- La classe GenitoreConFigliMinorenni ha, similmente, la proprietà haFigli vincolata alla classe Minorenni.
- Le classi GenitoreConFigliMaggiorenni e GenitoreConFigliMinorenni non sono disgiunte, ma lo sono le classi Maggiorenni e Minorenni.

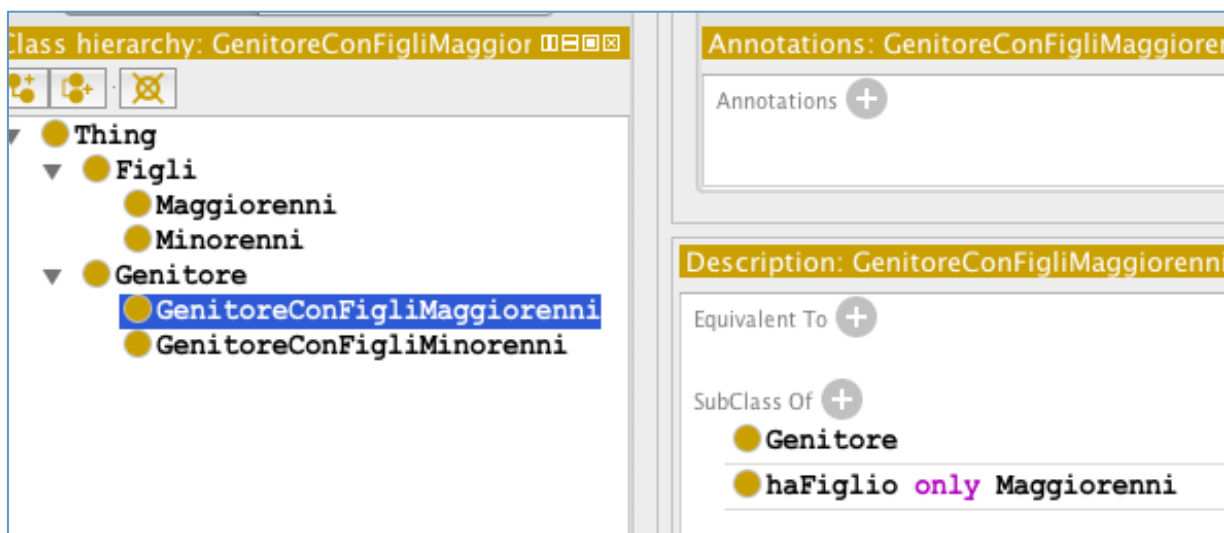


Figura 7: creazione di una classe definita con "only".

Come visto sopra, non è possibile dichiarare un individuo come membro di entrambe le classi, Maggiorenni e Minorenni, pena l'inconsistenza logica dell'ontologia.

La disgiunzione, tuttavia, si riflette anche sulle classi "GenitoreConFigliMaggiorenni" e "GenitoreConFigliMinorenni": se un individuo, infatti, viene collocato in entrambe le classi, il reasoner rileva una inconsistenza. Se Carlo, che *haFiglio* Giulia (minorenne), viene dichiarato come membro di entrambe le classi, il reasoner rileva un'inconsistenza. Giulia infatti si troverebbe a appartenere a due classi distinte, che sono state dichiarate come disgiunte: infatti, in virtù della definizione delle due classi, che comportano l'utilizzo di *only* (quantificatore universale), l'individuo Giulia appartiene alla classe "Maggiorenni" in quanto il proprio genitore è stato dichiarato come appartenente alla classe "GenitoreConFigliMaggiorenni" (e tutti i suoi figli devono essere maggiorenni) e alla classe "Minorenni" in quanto il proprio genitore è stato dichiarato come appartenente alla classe "GenitoreConFigliMinorenni" (e tutti i suoi figli devono essere minorenni). Si noti che la definizione della classi è corretta, ma vincolante per la classificazione degli individui, benché i vincoli che essa determina non siano direttamente evidenti alla maggior parte degli osservatori.

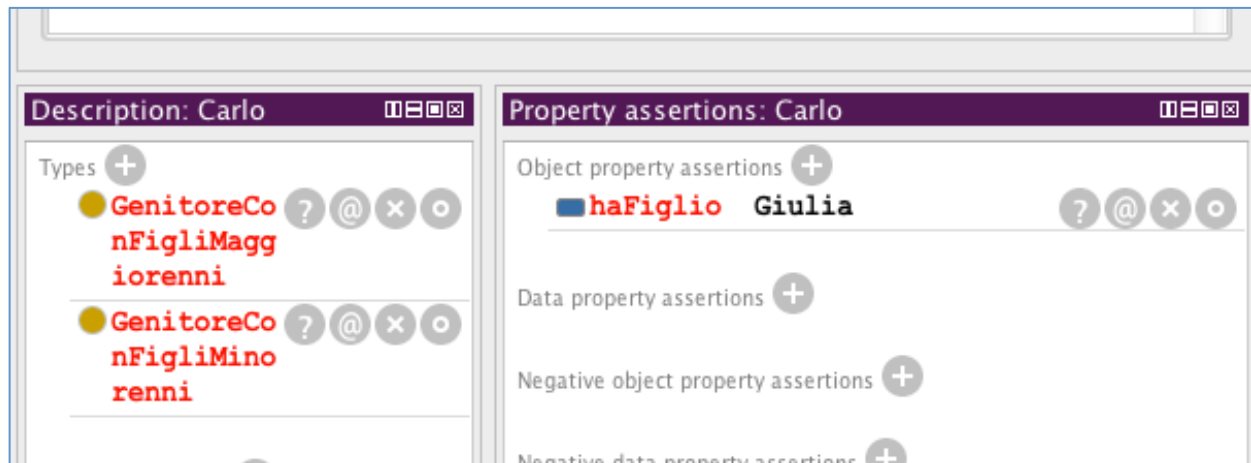


Figura 8: esempio di inconsistenza: le asserzioni appaiono in rosso.

Il messaggio di errore prodotto dal reasoner, in questo caso, è il seguente:

"Reason for inconsistency: individual Giulia is forced to belong to class Minorenni and its complement"

Se si chiede a Protégé di fornire una spiegazione dell'inconsistenza rilevata dal reasoner, si ottiene la seguente spiegazione, che evidenzia il ruolo della disgiunzione nel determinare la stessa:

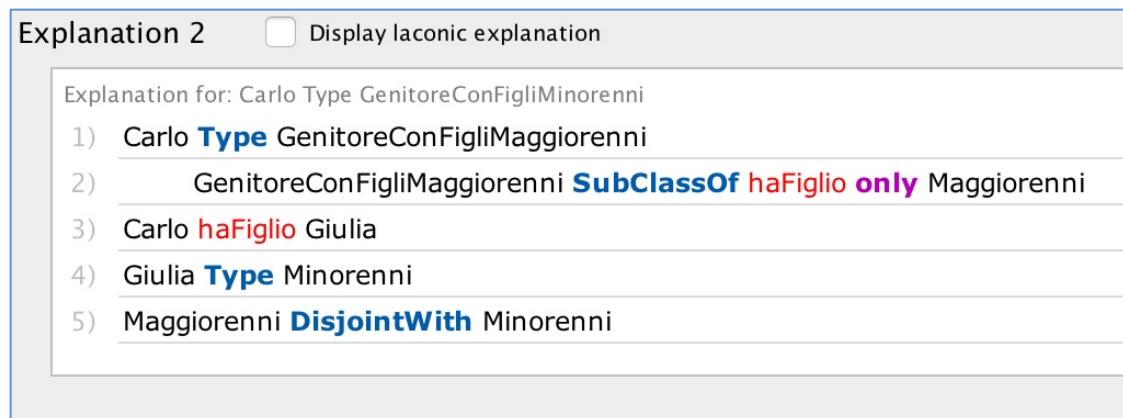


Figura 9: spiegazione dell'inconsistenza rilevata dal reasoner in Protégé.

3. Assunzione di mondo aperto

Le inferenze effettuate dal *reasoner* possono sembrare poco intuitive in alcuni casi. La maggior parte dei sistemi automatici con cui entriamo in contatto quotidianamente, infatti, segue l'assunzione di mondo chiuso:

"ciò che non è rappresentato esplicitamente nel sistema viene assunto falso"

Per esempio, il fatto che sul mio conto bancario non sia presente un determinato accredito implica che tale accredito non sia stato effettuato; secondo la stessa logica, se il mio nome non è sulla lista passeggeri di un volo, significa che non ne faccio parte.

Il ragionamento sulle ontologie OWL segue invece l'*assunzione di mondo aperto*:

Il fatto che un'informazione non sia rappresentata nel sistema non determina che essa sia assunta falsa. Questo fa sì che gli strumenti di ragionamento automatico non rilevino come inconsistenti determinate configurazioni di classi e individui che lo sarebbero per il senso comune. Tornando all'esempio della lista passeggeri, essa ci dice solo ciò che è certamente vero, cioè che i passeggeri presenti sulla lista sono effettivamente tali, ma non implica che nome non presente sulla lista non potrebbe farvi parte.

Assunzione di mondo aperto e classi definite

Si consideri per esempio la classe "PadreDiDueFigli". Tale classe è una classe definita per la quale vale il vincolo che, per ogni membro della classe, la proprietà *haFiglio* lo colleghi almeno con due individui diversi (si veda la Figura 2).

Si noti ora che il fatto che un certo individuo, Giulio, sia padre di un solo figlio, tuttavia, non genera un'inconsistenza nella classificazione, anche se Giulio è stato collocato nella classe "PadreDiDueFigli".

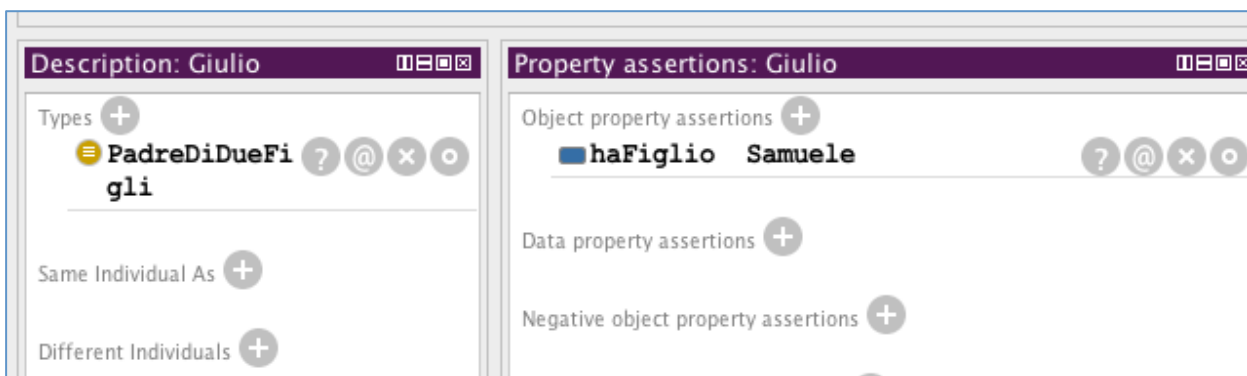


Figura 10: descrizione di un individuo (Giulio) attraverso la proprietà *haFiglio* (a destra) e l'appartenenza alla classe "Padre di due figli" (a sinistra).

Il reasoner infatti, applica l'*assunzione di mondo aperto*: non viene asserito che Giulio non abbia altri figli e quindi l'appartenenza alla classe, se asserita esplicitamente, non è in conflitto con le proprietà dell'individuo Giulio.

In altre parole, il reasoner non classificherebbe l'individuo Giulio in questa classe, ma il fatto che l'utente ve l'abbia collocato non determina un'inconsistenza.

Assunzione di mondo aperto nelle restrizioni quantitative

L'assunzione di mondo aperto si applica anche agli individui. Tale fatto è rilevante nel ragionamento relativo a vincoli di tipo quantitativo.

Si consideri per esempio l'esempio seguente, rappresentato in Figura 11: l'individuo Giovanni è descritto come padre di due individui, Samuele e Carlo, attraverso la proprietà *haFiglio*. Il reasoner inferisce correttamente che Giovanni, in quanto padre di due individui (Samuele e Carlo), appartiene alla classe "PadreDiDueFigli"

Tuttavia la classificazione avviene solo se Samuele e Carlo sono stati specificati esplicitamente come due diversi individui attraverso l'apposito slot presente nell'interfaccia di Protégé (*Different Individual*), come illustrato in Fig. 12.

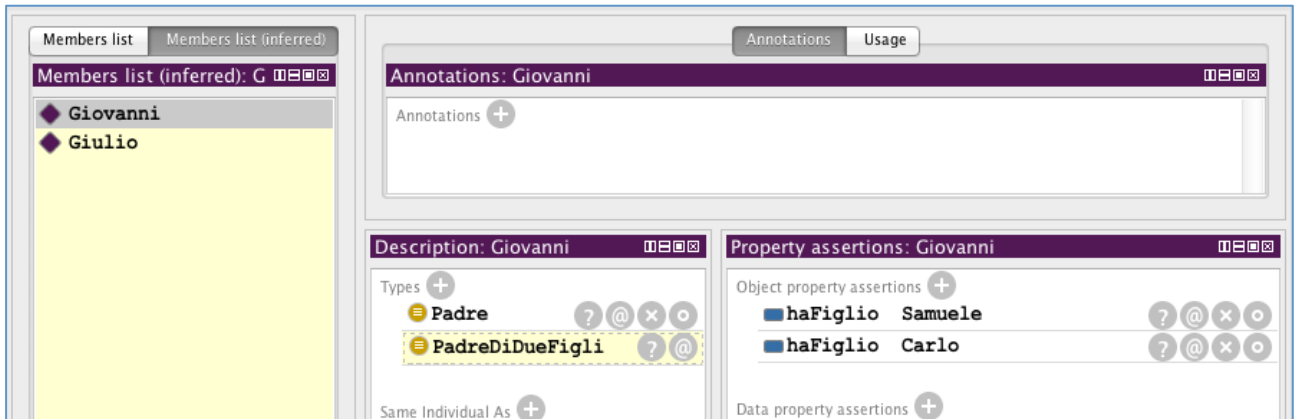


Figura 11: classificazione automatica di un individuo sulla base di vincoli quantitativi (in giallo).

Infatti, per l'assunzione di mondo aperto, Samuele e Carlo potrebbero essere lo stesso individuo e quindi il vincolo posto sulla classe "PadreDiDueFigli" non sarebbe soddisfatto, perché esisterebbe almeno un modello in cui Carlo e Samuele sono lo stesso individuo e il vincolo quantitativo di conseguenza non vale.

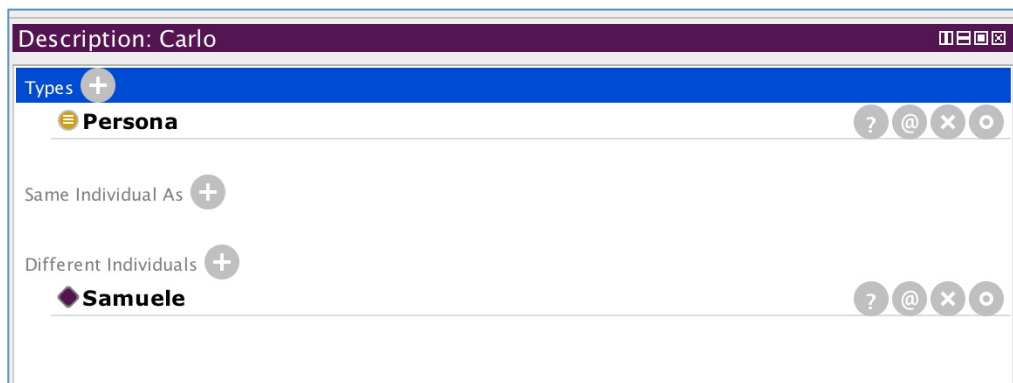


Figura 12: utilizzo di Different Individual From in Protégé (in basso).

Bibliografia

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2), 51-53.