

Rescue Robot Scenario

Esercitazione DL

Sommario

- Tools per l'esercitazione
- Specifica di uno scenario complesso in OWL
- Esercizi di interrogazione

Note all'utilizzo di Protege

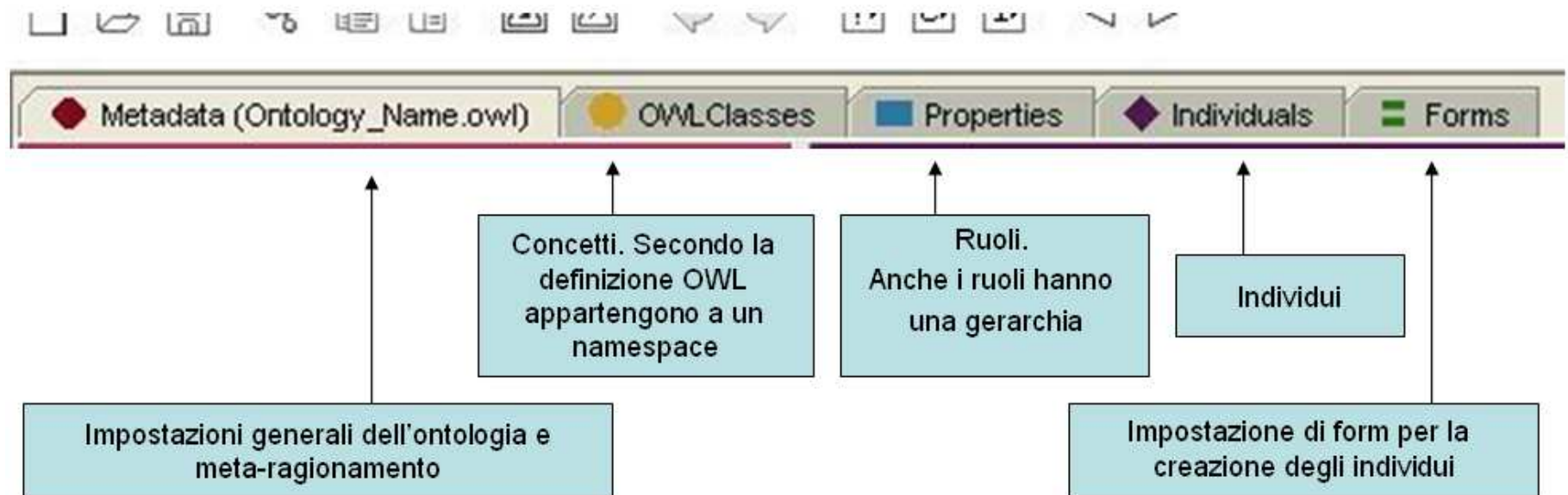
- Controllare che la porta del ragionatore (in Pellet è stampata all'avvio) sia la stessa di quella dell'editor OWL (Protegé).

In Protégé si imposta dal menu OWL selezionando preferences

- Percorso per la creazione di una ontologia che utilizza le description logics

New Project -> OWL Files -> Next -> Next -> OWL DL -> Logic View

Note all'utilizzo di Protege



Nostro esempio - Robot Scenario

Che cosa è un robot?

- Un robot è rappresentato come una collezione di funzionalità.
- Le funzionalità dipendono dal contesto in cui il robot opera.

Nota: E' più generale che dire che un robot è una collezione di sensori. Le stesse funzionalità si possono ottenere con diverse categorie di sensori.

Concetti primitivi

- Definizione: elemento dell'ontologia specificato tramite condizioni necessarie.
- Asserzioni su concetti primitivi: Viene popolato tipicamente come nei database. Un individuo X (a.e. $LISA$) appartiene ad un concetto primitivo A (a.e. $Robot$) se è stato esplicitamente dichiarato tale ($Robot(LISA)$), o se è individuo di un sotto-concetto A (a.e. $TipoB \sqsubseteq Robot$ $TipoB(LISA)$).
- Open World Assumption: ovviamente se nessuna di queste due regole è esplicitata, non posso concludere che "X non è individuo di A", cioè X potrebbe ancora essere tale.

Concetti primitivi - esercizio

Un robot è una collezione di funzionalità. Le funzionalità possibili sono di tipo Attuativo, Sensoriale, di Comunicazione.

Ci sono tre tipi di robot (A,B,C). A è solo funzionalità attuative, B è attuative e sensoriali, C è un robot completo di tutte e tre le funzionalità.

SOLDATINO è un robot di tipo A, LISA di tipo B, REDBACK di tipo C.

Concetti primitivi - soluzione

Attuative, Sensoriali, Comunicazione, Robot, TipoA, TipoB, TipoC sono concetti primitivi, SOLDATINO, LISA e REDBACK sono individui.

$Attuative \sqsubseteq Funzionalità$

$Sensoriali \sqsubseteq Funzionalità$

$Comunicazione \sqsubseteq Funzionalità$

$Robot \sqsubseteq Funzionalità$

$TipoA \sqsubseteq Robot \sqcap Attuative$

$TipoB \sqsubseteq Robot \sqcap Attuative \sqcap Sensoriali$

$TipoC \sqsubseteq Robot \sqcap Attuative \sqcap Sensoriali \sqcap Comunicazione$

$TipoA(SOLDATINO)$

$TipoB(LISA)$

$TipoC(REDBACK)$

Definizione di Ruoli, vincoli, restr. numeriche

Rispetto alle relazioni di un database, i ruoli sono "orientati", anche se la loro rappresentazione interna rimane molto simile.

Nell'arena GIALLA1 ci sono le due vittime Pippo e Pluto.

Un robot ha la funzionalità RicezioneImmagini se è in comunicazione con un altro robot che ha la funzionalità RicezioneImmagini.

In ogni arena ci sono almeno due vittime.

Tutte le scale presenti in un arena arancione devono essere non a pioli.

Ruoli e vincoli - soluzione

presente(GIALLA1, PIPPO)

presente(GIALLA1, PLUTO)

Comunicazione \sqcap \exists *connesso*. (*Comunicazione* \sqcap *RicezioneImmagini*) \sqsubseteq

RicezioneImmagini

Arena \sqsubseteq ≥ 2 *presenteVittima*

ArenaArancione \sqsubseteq \forall *presentiScale*. (\neg *APioli*)

Ruoli e vincoli - esercizio

Ogni robot che ha funzionalità di comunicazione può comunicare con un protocollo TCP o UDP. Modificare la proprietà della ricezione immagini in modo che due robot possano comunicare tra loro solo se usano lo stesso protocollo.

Ogni arena deve avere una porta di entrata e una di uscita.

Ruoli e vincoli - soluzione

$Comunicazione \sqsubseteq \forall \text{protocollo.} ("tcp" \sqcup "udp")$

$Comunicazione \sqcap \exists \text{protocollo.} "tcp" \sqcap \exists \text{connesso.} (Comunicazione \sqcap \exists \text{protocollo.} "tcp" \sqcap RicezioneImmagini) \sqsubseteq RicezioneImmagini$

$Comunicazione \sqcap \exists \text{protocollo.} "udp" \sqcap \exists \text{connesso.} (Comunicazione \sqcap \exists \text{protocollo.} "udp" \sqcap RicezioneImmagini) \sqsubseteq RicezioneImmagini$

$Arena \sqsubseteq \exists \text{presenti.} ENTRATA$

$Arena \sqsubseteq \exists \text{presenti.} USCITA$

Concetti definiti

- Non vengono popolati direttamente, gli individui vengono ritrovati tramite sussunzione.
- Non hanno corrispettivo nei database. Si specificano solo le condizioni necessarie e sufficienti per l'appartenenza al concetto A, e il ragionatore "popolerà" il concetto.
- Le classi primitive sono anche definite se hanno sotto-concetti definiti.

In Protegé i concetti definiti si indicano con il simbolo



Concetti definiti - esercizio

Esistono 3 tipi di arene, rossa arancione e gialla. Un'arena è rossa sse ha terreno accidentato. Un'arena è arancione sse ha scale, e non terreno accidentato. Un arena è gialla sse ha più di due vittime, ma nè scale nè terreno accidentato.

Concetti definiti - soluzione

Arena, *ArenaRossa*, *TerrenoAccidentato*, *Scala*, *Vittima* sono concetti primitivi.

presente è un ruolo $Arena \times \top$

presenteVittima ruolo $Arena \times Vittima$

$ArenaRossa \equiv Arena \sqcap \exists \textit{presente}.TerrenoAccidentato$

$ArenaArancione \equiv Arena \sqcap \neg \exists \textit{presente}.TerrenoAccidentato \sqcap \exists \textit{presente}.Scala$

$\textit{presenteVittima} \sqsubseteq \textit{presente}$

$ArenaGialla \equiv Arena \sqcap \neg \exists \textit{presente}.(TerrenoAccidentato \sqcup Scala) \sqcap \geq 2 \textit{presenteVittima}$

Concetti definiti - esercizio

Una squadra di robot contiene uno o più robot. Nella squadra BARNEYTEAM si trovano i robot SOLDATINO di tipo A e LISA di tipo B. La squadra BARNEYTEAM si trova nell'arena gialla GIALLA1. Il robot REDBACK di tipo C si trova nell'arena rossa ROSSA1.

Creare un concetto contenente tutti e soli i robot della squadra BARNEYTEAM.

Concetti definiti - soluzione

Squadra(BARNEYTEAM)

ArenaGialla(GIALLA1)

ArenaRossa(ROSSA1)

appartiene(BARNEYTEAM, SOLDATINO)

appartiene(BARNEYTEAM, LISA)

appartiene(REDBACKTEAM, REDBACK)

presenti(GIALLA1, BARNEYTEAM)

presenti(ROSSA1, BARNEYTEAM)

ComponentiBarneyTeam $\equiv \exists$ appartiene⁻.BARNEYTEAM

Ragionamento sui concetti

Classificazione ragiona solo sui concetti (Tbox). Applica la sussunzione sui concetti cercando nuove relazioni tra loro non definite esplicitamente.

Menu OWL e seleziona "Classify Taxonomy"

Soddisfacibilità dei concetti verificare se un concetto ammette almeno un individuo; si riconduce alla non-subsumption (in cui il subsumer è il concetto vuoto).

Tasto destro sul concetto e seleziona "Check Concept Consistency"

Menu OWL e seleziona "Check Consistency": verifica soddisfacibilità di tutti i concetti

Equivalenza fra due concetti

Va controllata applicando la definizione: si esegue la classificazione e si verifica che uno sussume l'altro e viceversa

Disgiunzione tra concetti

Va controllata applicando la definizione: si costruisce un concetto intersezione, e si verifica se è inconsistente

Alternativamente Protegé permette di usare le API JAVA, cioè di ampliare le funzionalità scrivendo il codice in java

Ragionamento sugli individui

Instance checking verifica che un individuo sia istanza di un concetto

Tasto destro sull'individuo in "Individuals" e seleziona "-compute types" (in realtà fa instance checking di quell'individuo su tutti i concetti definiti)

Menu OWL e seleziona "Compute inferred types": instance checking di tutti gli individui su tutti i concetti

Retrieval trova tutti gli individui che sono istanza di un dato concetto

Tasto destro sul concetto e seleziona "Compute individuals belonging to class"

Verifica della consistenza (Abox)

Viene eseguita incrementalmente mentre faccio le asserzioni

Quale robot per quale arena?

Data la definizione:

SquadraRescue \equiv *Squadra* \sqcap \exists *appartiene*. *Esplorazione* \sqcap \exists *appartiene*.

Localizzazione \sqcap \exists *appartiene*. *Mapping* \sqcap \exists *appartiene*. *Riconoscimento Vittime*

"Una squadra è una squadra rescue se può fare esplorazione, mapping, localizzazione, riconoscimento vittime".

Esistono 3 funzionalità attuative: Basic, SuScale, SuOgniTerreno, a seconda delle arene che ogni robot può affrontare.

Specificare che un robot ha la funzionalità attuativa Mobilità se può muoversi nell'arena in cui si trova la squadra cui appartiene.

Specificare il vincolo che una squadra, per essere una squadra rescue deve trovarsi in una arena, e almeno un robot deve potersi muovere in quella arena.

Quale robot per quale arena? - Soluzione

$Basic \sqsubseteq Attuative$

$SuScale \sqsubseteq Basic$

$SuOgniTerreno \sqsubseteq SuScale$

$MobilitaRossa \equiv SuOgniTerreno \sqcap (\exists appartiene^-. (\exists presenti^-. ArenaRossa))$

$MobilitaArancione \equiv SuScale \sqcap (\exists appartiene^-. (\exists presenti^-. ArenaArancione))$

$MobilitaGialla \equiv Basic \sqcap (\exists appartiene^-. (\exists presenti^-. ArenaGialla))$

$Mobilita \equiv MobilitaRossa \sqcup MobilitaArancione \sqcup MobilitaGialla$

$SquadraRescue \equiv \dots \sqcap \exists presenti^-. Arena \sqcap \exists appartiene. Mobilita$

Squadre di robot

Interrogare il sistema chiedendo:

- le squadre che possono muoversi in un arena rossa.
 - quale tipo di arena può affrontare la squadra BARNEYTEAM.
 - se ArenaRossa e ArenaGialla sono concetti distinti.
- Si ricordi che la squadra può affrontare una difficoltà se almeno uno dei suoi robot può farlo.

Squadre di robot - Soluzione

1 - Si crea il concetto

SquadrePerArenaRossa $\equiv \exists \text{appartiene.MobilitaRossa}$

e si fa retrieval.

2 - Stiamo chiedendo di fare istanze checking di BARNEYTEAM su SquadraRescue. Basta fare questo per tre volte inserendo la squadra nelle varie arene.

3 - Si costruisce un concetto

Intersezione $\equiv \text{ArenaRossa} \sqcap \text{ArenaGialla}$

e si verifica l'inconsistenza.

Esercizi

- 1 - Può un arena $W1$ essere classificata automaticamente come *ArenaArancione*? Se sì, come? Se no, perchè?
- 2 - Se volessi fare in modo che la porta di entrata e di uscita da un'arena non siano necessariamente due porte differenti, dovrei modificare la regola sulle porte definita nei lucidi precedenti? Se sì come, se no perchè?
- 3 - Se modificassi la definizione di *SquadraRescue* con $SquadraRescue \equiv Squadra \sqcap \exists \text{appartiene} . (Esplorazione \sqcap Localizzazione \sqcap Mapping \sqcap RiconoscimentoVittime)$ quale sarebbe la differenza?
- 4 - Interrogare il sistema chiedendo il contenuto dell'arena GIALLA1.