

IA et résolution de contraintes : TP2 (À rendre)

Validation des concepts généraux d'inclusion en regardant une ontologie

04/11/2014

L'objectif de ce TP est d'utiliser un raisonneur pour vérifier la validation des axiomes de TBox par apport une ontologie. Formellement, supposant que α est une axiome de TBox et O est une ontologie, on dit que α est valide par apport à O si α est une conséquence logique d' O , c'est à dire, $O \models \alpha$ selon la sémantique de la logique de description.

- **L'entré:** un fichier d'ontologie "GRO.xml" et un fichier d'axiome de TBox "gcis". Le fichier d'ontologie peut-être lu par OWLAPI. Dans le fichier "gcis", un vocabulaire qui termine par "_cncpt" est traduit directement comme un concept qui garde le même nom ; et le symbole " \perp " correspond au concept vide owl:Nothing (ou \perp selon la syntaxe de la logique de description). La traduction de chaque ligne du fichier "gcis" au axiome de TBox est fondé sur la fonction $\pi(\cdot)$ définie comme ci-dessous qui fournie la traduction des constructeurs **GCI**, **and**, **exists** (voir le fichier "gcis") aux constructeurs de la logique de description :

$$\begin{aligned}\pi(X) \sqsubseteq \pi(Y) &\longleftarrow \pi(GCI \ X \ Y) \\ \pi(C_1) \sqcap \pi(C_2) \sqcap \dots \sqcap \pi(C_n) &\longleftarrow \pi((and \ C_1 \ C_2 \dots \ C_n)) \\ \exists R. \pi(C) &\longleftarrow (exists \ R \ C)\end{aligned}$$

Par exemple : "(gci (and DNA_cncpt Eukaryote_cncpt))" correspond au axiom :

$$DNA_cncpt \sqcap Eukaryote_cncpt \sqsubseteq \perp$$

"(gci (and (exists LocatedIn (and Promoter_cncpt))) \perp))" correspond au axiom :

$$\exists LocatedIn. Promoter_cncpt \sqsubseteq \perp$$

"(gci (and (exists LocatedIn (and ProteinDomain_cncpt Gene_cncpt))) (and Localization_cncpt)))" correspond au axiom :

$$\exists LocatedIn. (ProteinDomain_cncpt \sqcap Gene_cncpt) \sqsubseteq Localization_cncpt$$

Toutes les lignes qui ne suivent pas la syntaxe ci-dessus peuvent être ignorées.

- **Sortie:** Deux fichiers (validGCIs.txt et validGCIs.xml) qui ne contient que les axiomes valides par apport à l'ontologie d'entrée : l'un dans la format de "gcis" et l'autre dans le format d'ontologie OWL.

Vous pouvez réaliser cette tâche de façon différente (pour étape 2). Choisissez une que vous préférez :

- Étape 1 : écrire une fonction de JAVA qui peut créer une ontologie GRO du fichier "GRO.xml", et créer une nouvelle ontologie validGCIs (sauvegarder le dans le fichier validGCIs.xml).

- Étape 2 : écrire une fonction de JAVA qui peut traduire chaque ligne à deux concepts X et Y , ensuite décider si l'axiome correspond à cette ligne $X \sqsubseteq Y$ est valide en regardant GRO en utilisant le raisonneur HermiT. Si oui, enregistrez la ligne dans un fichier validGCIs.txt, et ajoutez l'axiome dans l'ontologie validGCIs.
- (ou bien) Étape 2' : écrire une fonction de JAVA qui peut traduire chaque ligne (de bonne format) à un axiome ontologique, en utilisant OWLAPI :

```
public OWLSubClassOfAxiom formatOWLAxiom(String line);
```

Écrire ensuite une fonction pour décider si un axiome retourné de cette fonction est valide par rapport l'ontologie GRO en utilisant le raisonneur HermiT. Si oui, enregistrez la ligne dans un fichier validGCIs.txt, et ajoutez l'axiome dans l'ontologie validGCIs.