
Algorithm 1 OWL API Object Class Permuter

The ontology object *ontology* has access to all classes, properties, individuals, etc. in the ontology.

The list *unknowns* is a list of individuals to be permuted.

Assumptions:

1. An individual which is set to unknown is of a class type where each of the sub-classes can only have one value in the ABox.
2. All information about an individual in class-form are classes that are sub-classes of a main class.
3. Need some kind of assumption that makes it possible to do this:
superClass \leftarrow The super-class of *u* on the depth below owl:Thing/Top
Gir disse antagelsene i det hele tatt mening?

procedure PERMUTE(*Ontology* : *ontology*, *List* : *permutables*)

if *ontology* is inconsistent **then**
 return

if *permutables* is empty **then**
 write (consistent) *ontology* to file
 return

List : *restOfPermutables* \leftarrow copy of *permutables* without the first element

Individual : *u* \leftarrow first element of *permutables*

 // smallest as in lowest in the class hierarchy

Class : *superClass* \leftarrow the "smallest" super class of all the class assertions of *u*

List : *directSubclasses* \leftarrow all direct sub classes of *superClass*

List : *axiomLists* \leftarrow instantiate new list (which is a list of lists)

for all *c_s* \in *directSubclasses* **do**

List : *leafSubclasses* \leftarrow all sub classes of *c_s* without sub classes
 add generateAxioms(*leafSubclasses*, *u*) to *axiomLists*

List : *permutations* \leftarrow all permutations of the lists in *axiomLists*

for all *permutation* \in *permutations* **do**

List : *newAxioms* \leftarrow new list

 // make sure to not add and remove an axiom that is already in the ontology

for all *a* \in *permutation* **do**

if *a* \notin *ontology* **then**
 add *a* to *newAxioms*

for all *a* \in *newAxioms* **do** 1

 add *a* to the ontology

 permute(*ontology*, *restOfPermutables*)

for all *a* \in *newAxioms* **do**

 remove *a* from the ontology

Algorithm 2 Helper Method(s)

Generates a list of axiom lists, i.e. $[[\text{Reservoir}(u), \text{Shale}(u)],$
 $[\text{InFeederchannel}(u), \text{InDistributarychannel}(u)]]$

procedure GENERATEAXIOMS($List : \text{leafClasses}, \text{Individual} : u$)

$List : \text{axioms} \leftarrow$ new list

for all $c_l \in \text{leafClasses}$ **do**

$Axiom : a \leftarrow$ assert u is of class c_l

add a to axioms

return axioms
