Algorithm 2

```
Input: type of questions, ontology, set of templates
Fetch all asserted and inferred axioms that satisfies the axiom prerequisites for
the selected type of questions using Reasoner
Select one axiom randomly
Fetch the vocabulary elements vocabularyelements of the selected axiom
Check the category of the Object Property (OP) of the selected axiom (Table 2 presents the categories and some examples)
Get the appropriate template and formatted vocabulary element (vocOP) with respect to the type of questions, the selected axiom and the category of the OP from the OP classification {The OP Classifier algorithm (included in the supplementary material) uses SimpleNLG to obtain the third-person singular in the present simple tense and the past participle form of a verb, and WordNet to check the POS (Part-Of-Speech) of words such as verb, noun, adjective and preposition)}
for each voc in vocabularyelements do
  voc = checkFormat(voc) {Camel case, hyphen}
  if isClass(voc) == true then
    if POS(voc) == noun then
      if isCountableNoun(voc) == true then
        formattedVoc = addArticle(voc, "a"/"an"/"") {Choose the right article a/an/(empty) for voc by using SimpleNLG, if the axiom contains an existential quantifier, do not insert any article}
      else
        formattedVoc = voc {formattedVoc is the formatted vocabulary element to fill in the appropriate template}
      end if
    else if POS(voc) == infinitive_verb then
      formattedVoc = gerundForm(voc) {By using SimpleNLG}
    else
      formattedVoc = voc {No transformation if voc is neither a noun nor an infinitive verb}
    end if
  else if isQuantifier(voc) == true then
    formattedVoc = addQuantifier(voc, "some"/"only") {for ∃, use "some"; and for ∀, use "only"}
  else if isObjectProperty(voc) == true then
    formattedVoc = voc {No further transformation, here, voc = vocOP; it is already formatted by the Object Property Classifier)}
  end if
  Fill in the selected appropriate template with formattedVoc {the formatted vocabulary element}
end for
Output: Question
```