Algorithm: Object Property Classifier

- **Input: Object Property (OP)**
- formatted = **Format**(OP) {e.g., is-eatenBy -> is eaten by}
- words = **Split**(formatted) {e.g., is eaten by -> [is, eaten, by]}
- finished = false, index = 0, state = 0, vocOP = '' {formatted object property}
- propertyType = PropertyType.OP_UNKNOWN
- while (*finished == false*)
    - word = words[index]
    - switch (*state*)
        - **case 0:**
            - if *word=='is'* then
                - state = 1
            - elseif *word=='has'* then
                - state = 2
            - elseif ***IsVerbPastParticiple****(word)* then {e.g., eaten}
                - state = 4
            - elseif ***IsVerb****(word)* then {e.g., eats}
                - state = 3
            - else
                - state = 101
            - endif
            - break

        - **case 1:**
            - index++
            - if *index>****Length****(words)* then
                - state = 100
                - continue
            - else
                - word = words[index]
            - end if
            - if ***IsVerbPastParticiple****(word)* then {e.g., eaten}
                - state = 4
            - elseif ***isDeterminer****(word)* then {e.g., a/an}
                - state = 8
            - else
                - state = 7 {e.g., noun or adjective}
            - endif
            - break

- **case 2:**
  - index++
  - if *index >= **Length**(words)* then
    - state = 100
      {Final state unreachable (no more words)}
    - continue
  - else
    - word = words[index]
  - endif
  - if ***IsDeterminer**(word)* then
    - state = 12
  - else
    - state = 11
  - endif
  - break

- **case 3:**
  - index++
  - if *index >= **Length**(words)* then
    - finished = true
    - **propertyType = PropertyType.OP_VERB**
  - else
    - word = words[index]
    - if ***IsPreposition**(word)* then
      - state = 15
    - else
      - state = 100
        {The next word is not a preposition}
    - endif
  - endif
  - break

- **case 4:**
  - index++
  - if *index >= **Length**(words)* then
    - word = word+"-by" {In this case, the word 'by' is missing}
    - words[index-1] = word
    - state = 5
  - else
    - word = words[index]
    - if *word=="by"* then
      - state = 5

        - elseif ***IsPreposition****(word)* then
          {It is a phrasal verb}
            - state = 6
        - elseif ***IsAdjectiveParticiple****(word)* || ***IsNoun****(word)* then
            - state = 41
        - else
            - state = 100
              {The word is neither by nor empty}
        - endif
    - endif
    - break

- **case 5:**
    - index++
    - if *index >=* ***Length****(words)* then
        - if *words[0]!= 'is'* then
            - words[0]= 'is '+words[0]
        - endif
        - **propertyType = PropertyType.OP_IS_PAST_PARTICIPLE_BY**
        - finished = true
    - else
        - state = 4
    - endif
    - break

- **case 6:**
    - if *index+1 >=* ***Length****(words)* then {Processing 'after preposition'}
        - state = 61
    - else
        - index++
        - word = words[index]
        - if *word== 'by'* then
            - state = 5
        - else
            - state = 100
        - endif
    - endif
    - break

- **case 7:**
    - words[index] = 'a '+word
    - newFormatted = ''

- for (*k = 0; k < **Length**(words); k++*)
    - newFormatted += words[k]+ ‘ ’
- endfor
- newFormatted = **Trim**(newFormatted)
- words = **Split**(newFormatted)
- index = 0
- state = 0
- word = words[index]
- break

- **case 8:**
    - index++
    - if *index >= **Length**(words)* then
        - state = 100
          {Final state unreachable (no more words)}
        - continue
    - else
        - word = words[index]
    - endif
    - if ***IsNoun**(word)* then
        - state = 9
    - else if ***IsAdjectiveParticiple**(word)* then
        - state = 81
    - else
        - state = 100
          {The next word is neither a noun nor an adjective nor a participle}
    - endif
    - break

- **case 9:**
    - index++
    - if *state != 10 and index >= **Length**(words)* then
        - state = 100
          {Final state unreachable (no more words)}
        - continue
    - else
        - word = words[index]
    - endif
    - if ***IsNoun**(word)* then
        - state = 9
    - elseif ***IsPreposition**(word)* then
        - state = 10

- else
    - state = 100
      {The next word is neither a noun nor a preposition}
- endif
- break

- **case 10:**
    - index++
    - if *index >= **Length**(words)* then
        - finished = true
        - **propertyType = PropertyType.OP_IS_NOUNS_PREP**
    - else
        - word = words[index]
        - if ***IsDeterminer**(word)* then
            - state = 8
        - elseif ***IsAdjectiveParticiple**(word)* then
            - state = 81
        - elseif ***IsNoun**(word)* then
            - state = 9
        - else
            - state = 100
              {The next word is neither a noun nor an adjective/part nor a determiner}
        - endif
    - endif
    - break

- **case 11:**
    - word = 'a '+word
    - words[index] = word
    - newFormatted = ''
    - for (*j = 0; j < **Length**(words); j++*) {
        - newFormatted+=words[j]+ ' '
    - endfor
    - words = **Split**(newFormatted)
    - state = 12
    - break

- **case 12:**
    - index++
    - if index >= **Length**(words) then
        - state = 100
          {Final state unreachable (no more words)}

        - continue
    - else
        - word = words[index]
    - endif
    - if ***IsNoun****(word)* then
        - state = 13
    - else if ***IsAdjectiveParticiple****(word)* then
        - state = 121
    - else
        - state =100
          {The next word is neither a noun nor an adjective/part}
    - endif
    - break

- **case 13:**
    - index++
    - if *index >=* ***Length****(words)* then
        - finished = true
        - **propertyType = PropertyType.OP_HAS_NOUNS**
    - else
        - word = words[index]
        - if ***IsPreposition****(word)* then
            - state = 14
        - else if ***IsNoun****(word)* then
            - state = 13
        - else
            - state = 100
              {the next word is neither a noun nor a preposition}
        - endif
    - endif
    - break

- **case 14:**
    - index++
    - if *index >=* ***Length****(words)* then
          {final state unreachable (no more words)}
        - state = 100
    - else
        - word = words[index]
    - endif
    - if ***IsPreposition****(word)* then
        - state = 14
    - elseif ***IsNoun****(word)* then

        - state = 13
    - else
        - state = 100
          {The next word is neither a preposition nor an adjective/part}
    - endif
    - break

- **case 15:**
    - index
    - if *index >= **Length**(words)* then
        - finished = true
        - **propertyType =PropertyType.OP_VERB_PREP**
    - else
        - state = 100
          {Final state unreachable (no more words)}
    - endif
    - break

- case 41: {format again the name of OP and redo transformation}
    - newFormatted = 'is a '+formatted
    - words = **Split**(newFormatted2)
    - state = 0
    - index = 0
    - word = words[0]
    - break

- case 61:
    - index++
    - if index >= **Length**(words) then
        - if *words[0]!= 'is'* then
            - words[0]= 'is '+words[0]
        - endif
        - **propertyType = PropertyType.OP_IS_PAST_PARTICIPLE_PREP**
        - finished = true
    - else
        - state = 4
    - endif
    - break

- **case 81:**
    - index++

- if *index >= **Length**(words)* then
    - state = 100
    - continue
- else
    - word = words[index]
- endif
- if ***IsNoun**(word)* then
    - state = 9
- elseif ***IsAdjectiveParticiple**(word)* then
    - state = 81
- elseif ***IsPreposition**(word)* then
    - state = 10
- else
    - state = 100
- endif
- break

- **case 100:**
    - finished = true
    - break

- **case 101:**
    - if ***IsPreposition**(lastWord)* then
      {it means that the auxiliary 'be' missing}
        - newFormatted = **Format**('is '+formatted)
        - words = **Split**(newFormatted)
        - index = 0
        - state = 0 {restart the operation to the beginning with this transformation}
    - else
        - newFormatted = **Format**('has '+formatted)
          {it this case, 'has' is missing on the name the OP}
        - words = **Split**(newFormatted)
        - state = 2
    - endif
    - break

- **case 121:**
    - index++
    - if *index >= **Length**(words)* then
          {Final state unreachable (no more words)}
        - state = 100
        - continue

- else
    - word = words[index]
- endif
- if ***IsAdjectiveParticiple****(word)* then
    - state = 121
- elseif ***IsNoun****(word)* then
    - state = 13

  else
    - state = 100

      {The next word is neither a noun nor an adjective/part}
- endif
- break

- endswith

- endwhile

- **vocOP** = ‘’
- foreach *word in words* do
    - **vocOP** +=word+‘ ’
- endforeach
- **vocOP** =**Trim**(**vocOP**)
- **vocOP** = **Format**(**vocOP**)

- **Output: propertyType**, **formatted Object property (vocOP)**