outputted text. It only actually affects a small number of words, so even though it will limit the bitrate slightly it will not reduce the effect on the quality of the sentence. If the punctuation was removed and reinserted after word replacement, the punctuation may not fit the replacement word.

### 3.1.3 Obfuscation

Obfuscation is the term used to represent hiding data in the text. To obfuscate bits is a relatively simple process. Two components are required, the cover-text and the bitstream (the list of bits that need to be hidden). For each word in turn in the cover text, the full synonym list is retrieved using the process described above. It is then a simple case of choosing the appropriate synonym to be returned. If the next bit in the bitstream is a zero, the first element is returned, if it is a one, the second, and if the next two elements are 10, the third (representing a 2, the value of binary 10) is used (this helps to increase the bitrate).

**Data**: Synset, bit to hide
**Result**: Chosen Synonym
**if** *synset size less than bit* **then**
| return synset element at position bit;
**end**

**Algorithm 2:** Obfuscation Algorithm Pseudo-code

Before the word is actually replaced, two tests are performed. The first is to attempt to deobfuscate the word and see if the original bit(s) is returned, if they are not then the word is unsuitable and so the bit(s) is not removed from the bitstream and the original entered word is used rather than the replacement synonym. The second test is to check the quality of the replacement using the quality test described below. If the quality is below a threshold, the bit(s) are not removed from the bitstream and again the original word is used. If the word is the first in the cover-text, then the quality check is instead the frequency for the most likely word type, taken from the BNC word frequency data. If the replacement word passes both of these tests, then the replacement word is returned and the hidden bits removed from the bitstream so they are not added again. It is worth noting that in many cases the replacement word can be the original word itself. if the bitstream is empty when obfuscation is attempted, the algorithms always tries to hide a 0 in the word. This is due to the fact that the deobfuscation algorithm always expects there to be data hidden in the text.

### 3.1.4 Deobfuscation

Deobfuscation refers to extracting hidden data from the text. To extract the hidden data from a word, the first step is for the word to pass some tests to limit the number of false positives. The primary test is to perform the quality test with the word before it. If the