3.1.4 Sampling and Stopping Criterion

For a given dataset and a given syntactic pattern, we use our tool to get the list of all occurrences to be analyzed. The sampling is then done by automatically shuffling the list and starting with the analysis of the first element. Given that the decision whether an occurrence is a directive or not is a binary decision, the proportion of directives revealed by a pattern is a standard proportion estimation. The stopping criterion is thus the minimum number MIN of occurrences to be analyzed in order to ensure that the estimated proportion is in a certain interval at a certain confidence level. The standard statistical formula is (Singh and Mangat 1996, Sec. 3.7):

$$MIN = \frac{n_0}{1 + \frac{n_0 - 1}{populationSize}} \tag{1}$$

n_0 depends on the selected confidence level and the desired error margin e. n_0 is calculated using the formulae: $n_0 = (Z^2 * 0.25)/e^2$, where Z is a confidence level's z-score. At 95% confidence level Z is 1.96. This formula is sometimes also presented as $n_0 = 4 * (Z^2 * 0.25)/B^2$, where B is the error interval, i.e. $B = 2 * e$. When the population size is large enough, MIN tends towards a constant (e.g. 384 for $e = 5\%$ at 95% confidence level); this is known as proportion estimation for large populations.

Our stopping criterion consists of analyzing MIN API elements for $e = 10\%$ at 95% confidence level. For instance, 1448 elements of the JDK's API documentation elements contain at least one occurrence of "may". If we analyze $MIN = 91$ elements, we are sure that the estimated proportion has a maximum error margin e of 10% at 95% confidence level, i.e. $p = x \pm 10\%$ (where x is the measured proportion on 91 items).

We chose $e = 10\%$ because it keeps the number of elements to be analyzed at a manageable level (5042 API elements for all three datasets and 53 concerns) while still giving us enough confidence in the results. Note that for certain keywords, due to a previous version of the case study, we analyzed many more API elements than MIN, as we will see in Section 5.

3.2 Validity

3.2.1 Completeness of the Taxonomy

In this section, we discuss whether the taxonomy—resulting from our exploratory case study—is complete, i.e. whether we miss an important directive kind. For this, we list the main threats to validity and discuss the taken counter-measures.

One possible threat is that the analyzed corpus is too small or is not representative. However, we think that every API corpus of a certain size and quality will reveal all major directive kinds. Analyzing more APIs would not yield any significant changes in the taxonomy. Though, defining hard criteria to decide whether an API corpus is large enough and representative is not possible, we are still convinced that it is the case for the chosen API corpus for the following reasons. Our API corpus covers a broad range of different domains (collections, IO, UI, math, SQL, security); and it covers different kinds of API usage such as inheritance or instantiation. Furthermore, the corpus covers three documentation processes employed by three different major software organizations (Sun, Eclipse and Apache) and the documentation has been written by a large number of different authors.