2 On Metrics and Measurements

Rainer Böhme\(^1\) and Felix C. Freiling\(^2\)

\(^1\) Technische Universität Dresden, Germany
\(^2\) University of Mannheim, Germany

The following chapter attempts to define the notions of metric and measurement which underlie this book. It further elaborates on general properties of metrics and introduces useful terms and concepts from measurement theory, without being overly formal.

2.1 On Measurement

In many cases, *to measure* means to attach a number to an object, i.e., to represent some aspect of the object in a quantitative way. For example, scientists can measure the temperature and the humidity of a location at a certain time by coding observations (temperature, humidity) related to the object (location) with numbers. More generally, a measurement function assigns an element of a set to an object, where the specific element is chosen depending on an observation. The set must not necessarily comprise numbers but can also consist of unordered symbols. For example, classifying the weather today as “rainy”, “dry”, “foggy” etc. is also regarded as measurement. However, not every assignment of numbers to objects is considered as measurement. For example, the matriculation number of a student is not a measurement because the number is chosen regardless of the student’s attributes (here we ignore that higher matriculation numbers may be an indicator of later admission).

In the setting of this book we usually want to measure *attributes* of systems or parts thereof, such as methods or processes. As system can be complex, there are many different measurable attributes. Any form of measurement is an *abstraction*: it reduces the complexity of one or several attributes of the original system to a single symbol. The main purposes of this form of abstraction are to classify and compare systems.

It is important to stress the difference between an attribute and its measurement. For example, the complexity of a software system is an attribute which can be measured in many different ways. However, the difference between an attribute and its measurement sometimes blurs because measurements are also taken to define the attributes.

Measurement is closely connected to the notion of a metric. In the course of this book we will use the term *metric* for a precisely defined method which is used to associate an element of an (ordered) set \( V \) to a system \( S \). This definition is used in the area of software quality. In other areas, the term metric only refers to the set \( V \), which contains *indicator values* that answer certain questions asked about a system. As we will see later, our understanding neither corresponds to the strict mathematical definition of a metric (where it is a generalisation of the notion of a distance).

In general, a metric can be formalised as a function \( M \) that takes a particular system from the set of systems \( S \) and maps it to an element of \( V \):

\[
M : S \mapsto V
\]
For example, $M$ may be the assignment of a distance between two measurement points of a system. Then $V$ is the set of real numbers, a totally ordered set. The set $V$ can also be a discrete set like the set of natural numbers in the “lines of code” metric for software. The set $V$ must also not necessarily be totally ordered; it can also be a partially ordered set or an unordered set like in the classification example above where $V$ consists of the elements \{foggy, rainy, dry\} etc.

Attributes can have certain properties which should be reflected in their metrics. For example, the complexity of a software package can be categorised as “low” or “high”. Some attributes are meaningful in the context of composed systems. For example, the attribute “size of a program” can be measured in lines of code. Given two programs $x$ and $y$ we can define their composition $z$ as the concatenation of $x$ and $y$. The metric “lines of code” reflects additivity in the following sense: the sizes of program $x$ and program $y$ together sum up to the size of their composition $z$. Similarly, some attributes allow to state relations between systems. Taking the “size” metric lines of code again, it is possible to say that some program is twice as large as another program.

Determining a suitable metric for an attribute of a system is not always easy. A good metric should reflect the relevant properties of the attribute in a homomorphic way. This means that certain statements which can be made for a certain attribute of systems should be reflected in the measurements of that attribute. In particular, two properties should hold:

- Any sensible relation between systems regarding a particular attribute should be reflected by a corresponding relation between the measurements of this attribute. For example, a system $x$ which is more complex than a system $y$ should be ordered appropriately if some complexity metric $c$ is used, i.e., $c(x) > c(y)$ should hold.
- Any meaningful operation on attributes of a system should have a corresponding operation on the measurements of that attribute. Assume there exists an addition operation (“plus”) for the “size” of programs. If the size of program $x$ “plus” the size of program $y$ equals the size of program $z$, then this should be reflected in the appropriate metric for size. For example, lines of code is an appropriate metric if the “plus” operator refers to concatenation of source code.

Any relations or operations on measurements which do not have a corresponding relation or operation on attributes must not be used to process the measurements.

### 2.2 On Scales

The result of measurements is data, which is further processed or analysed to answer questions of interest to the researcher or practitioner. A useful approach to classify types of data is given in the notion of scales. The term scale refers to the range $V$ of a metric, and the relation between elements within $V$. The most commonly used topology of scales goes back to Stevens [459], who defined a hierarchy of four different types of scales based on the invariance of their meaning under different classes of transformation. He further proposed to derive permissible procedures for data analysis and statistical inference depending on the scale level.