Introduction

Nowadays is very easy to model and verify properties of real time systems through the use of software modelling tools. Any well-defined system can be translated to a model by first understanding the main functionalities of it. But what if the goal is to create a model based on its output, rather than its actual functionality. Here is when the concept of reverse engineering becomes useful, where output data from a system is extracted and analyzed with the aim of creating a representation of it. But a problem arises when we want to create a representation of a whole system by only basing ourselves on the analysis of its output data. Many systems may have nondeterministic behaviors that will consequently display nondeterministic outputs, either depending on specific parameters or condition in which the system performs. This means that one can never actually predict the outcome of a system. For example, let us picture a very simple system that can only flip coins and register whether the coin landed with either “heads” or “tails” facing up. By analyzing the previous functionality, we can easily represent the system by constructing a model whose sole action is to throw coins and afterwards register the outcome. But now let’s try to construct a model by only analyzing the output of the system. Due to the fact that throwing a coin is a probabilistic even, let’s assume that the system threw a coin 20 times and that the result was always “heads”. After analyzing the previous output, we are only able to represent a model of a system that throws a coin and only registers “heads” as the result. For this particular case, we need to have an output that involves all possible outcomes of the system in order to properly model it.

It would be incorrect to affirm that any system’s functionality can be modelled by only analyzing its output. Which is why this research paper focuses on demonstrating how the output of a system can be transformed into a probabilistic model, that resembles the behavior of such system. If we retake the previous example of a system that throws coins. Let’s now assume that half of the results were “heads” and the other “tails”. After analyzing the results, now it is possible to model a perfect probabilistic system that throws coins, in which there is a 50% of chance to obtain “heads” or” tails”. But how can we actually recreate a system’s model by only analyzing its data? Well, in order to answer that question, we will first go through some relevant concepts regarding modelling systems and software modelling tools like UPPAAL. Followed by a detailed explanation of the methodology of the process of recreating a model by the analysis of its data. Proceeded by section of an actual experiment. And at the end finalizing with a conclusion that wraps up the essence of recreating a model and its importance in the real world.