# A STUDY ON AUTOMATA & FINITE STATE MACHINES
# PROBABILITY OF HAVING ADS & PDS

STUDENTS / UNIVERSITIES
Güren İÇİM / SABANCI UNIVERSITY
Zeynep BERKSÖZ / SABANCI UNIVERSITY

SUPERVISOR(S)
Hüsnü YENİGÜN

## ABSTRACT

In this project we work on ADS (Adaptive Distinguished Sequence) & PDS (Preset Distinguished Sequence) that are used for Initial State Identification in Finite State Machines (FSMs). Our purpose is to find the probability of the existence of ADS & PDS in a random FSM. The point of the exploration of ADS & PDS is to find short testing sequences, which make it possible to test a software or a device in polynomial time. A similar study has been done before on Synchronizing Sequences [1], a sequence used for Final State Identification. This study shows the probability for a random FSM to have a synchronizing sequence is $1-\theta\left(\frac{1}{n}\right)$, where $n$ is the number of states. When the number of states of a random FSM increases, the probability of finding a Synchronizing Sequence increases.

Based on this result, it is intuitive to think that the probability of existence of ADS & PDS will also increase when the FSM gets larger. However, in this work we experimentally show that the probability of having and ADS/PDS for a random FSM decreases as the size of the FSM increases.

## OBJECTIVES

- Implementing random FSM generation
- Writing/reading FSMs to/from text files
- Implementing Minimality Check, ADS & PDS existence check algorithms
- Generating a set of random FSMs and computing the probability of the existence of ADS/PDS on these FSMs

## DETAILS OF PROJECT

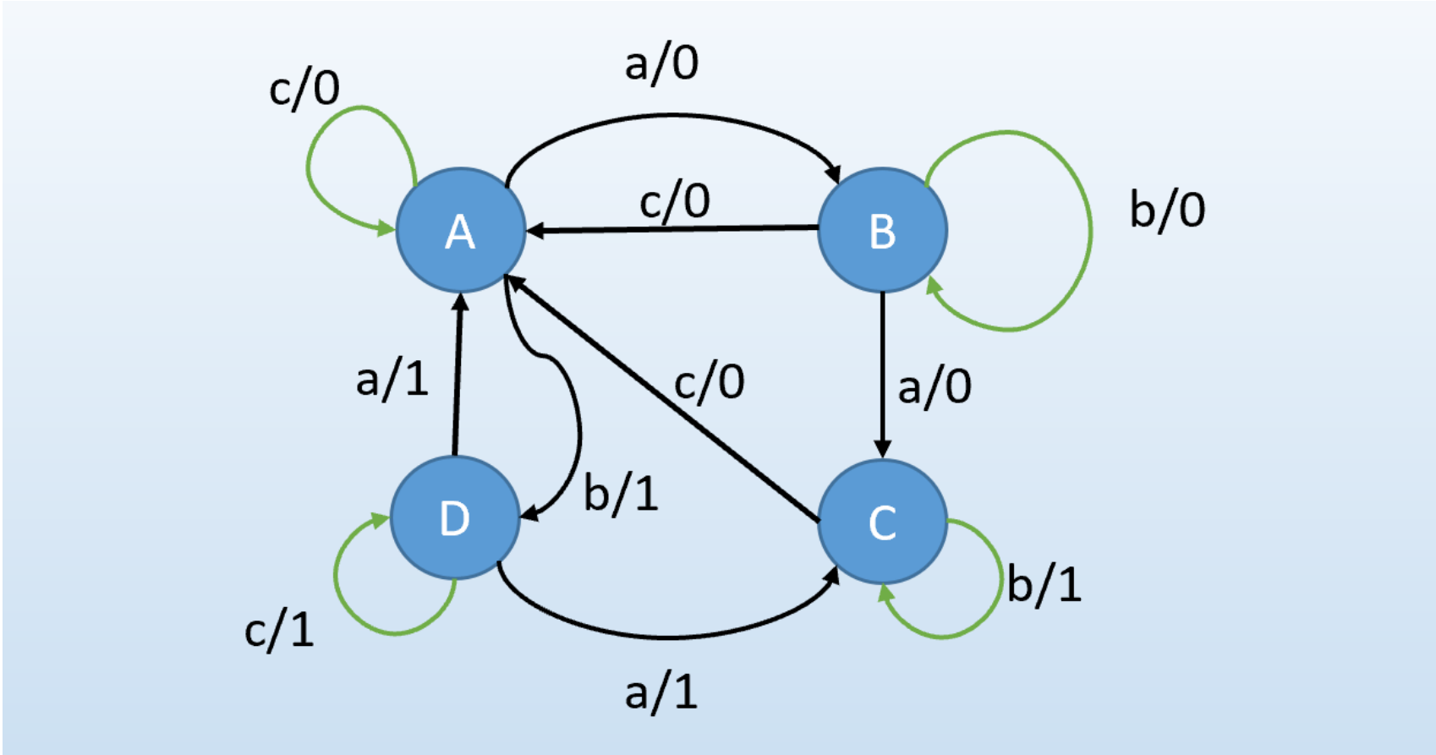

Figure 1. An example FSM *M*

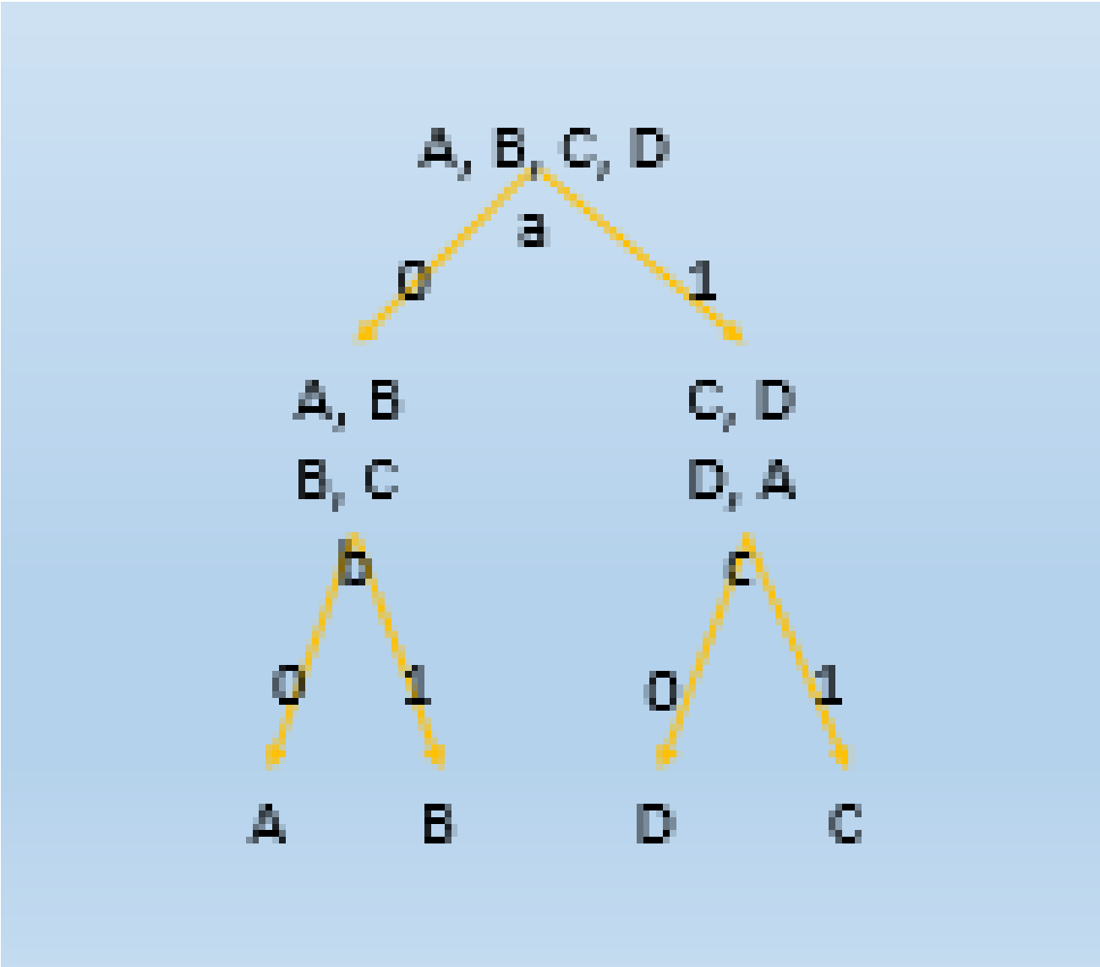

Figure 2. An ADS for FSM *M*

ADS and PDS [2] are two special sequences used for Initial State Identification in FSM based testing [3]. An FSM does not necessarily have an ADS/PDS, but when an FSM has an ADS or PDS, there are FSM based testing methods which can generate polynomial length, high quality test sequences. Since these methods rely on the existence of ADS/PDS, the usability of these methods is directly related to the probability of the existence of ADS/PDS. Therefore, in this work, we aim at finding the probability of the existence of an ADS/PDS in a random FSM.

We calculate this probability by using an experimental approach. We implemented ADS and PDS existence check algorithms for a given FSM. ADS and PDS existence check problems have different complexities: PDS ∈ PSPACE-HARD, ADS ∈ P.

We also implemented algorithms for random FSM generation and for the minimality check for an FSM (ADS and PDS only exists for minimal FSMs). We first construct a randomly generated set of minimal FSMs. Later these FSMs are considered one by one, and for each FSM we record if it has an ADS or a PDS.

We wrote approximately 1700 lines of codes in order to implement the algorithms mentioned above.

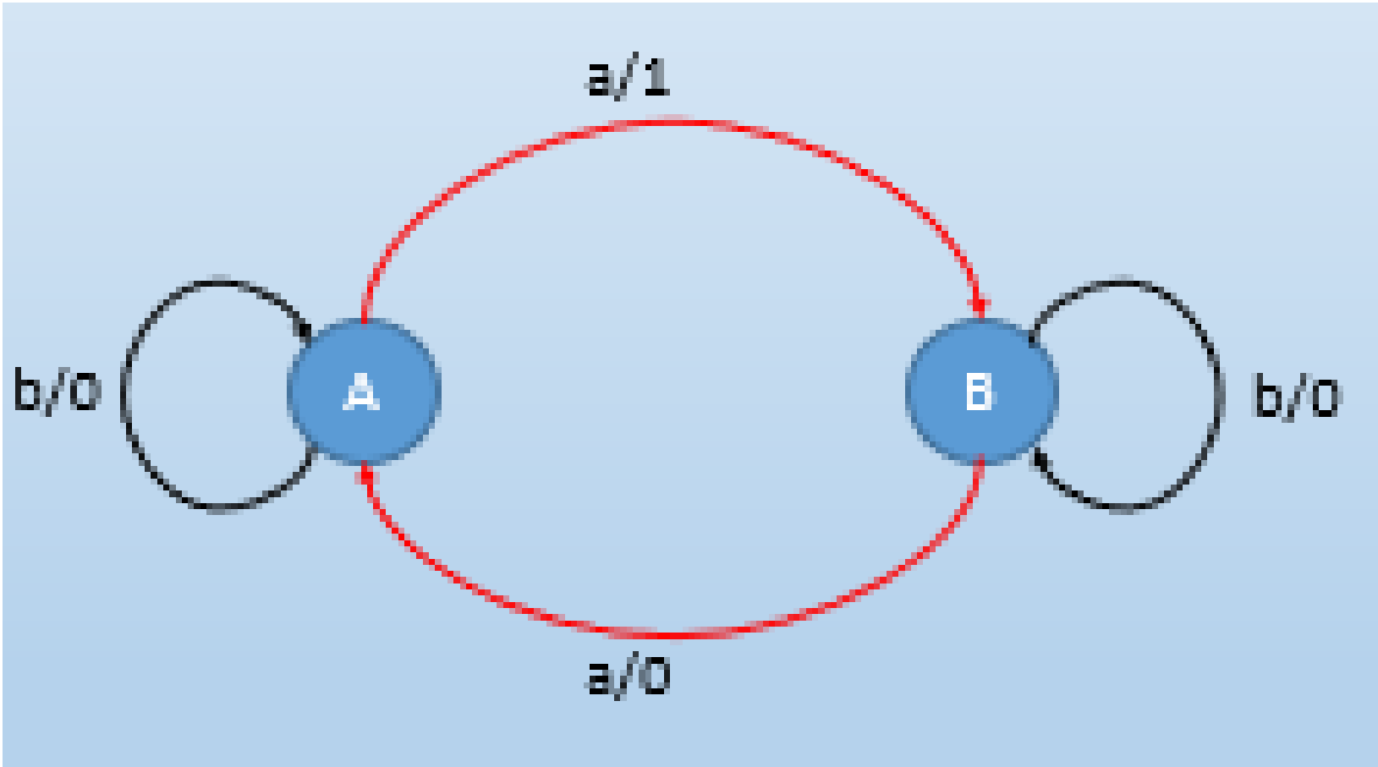

Figure 3. Another example FSM *M*'

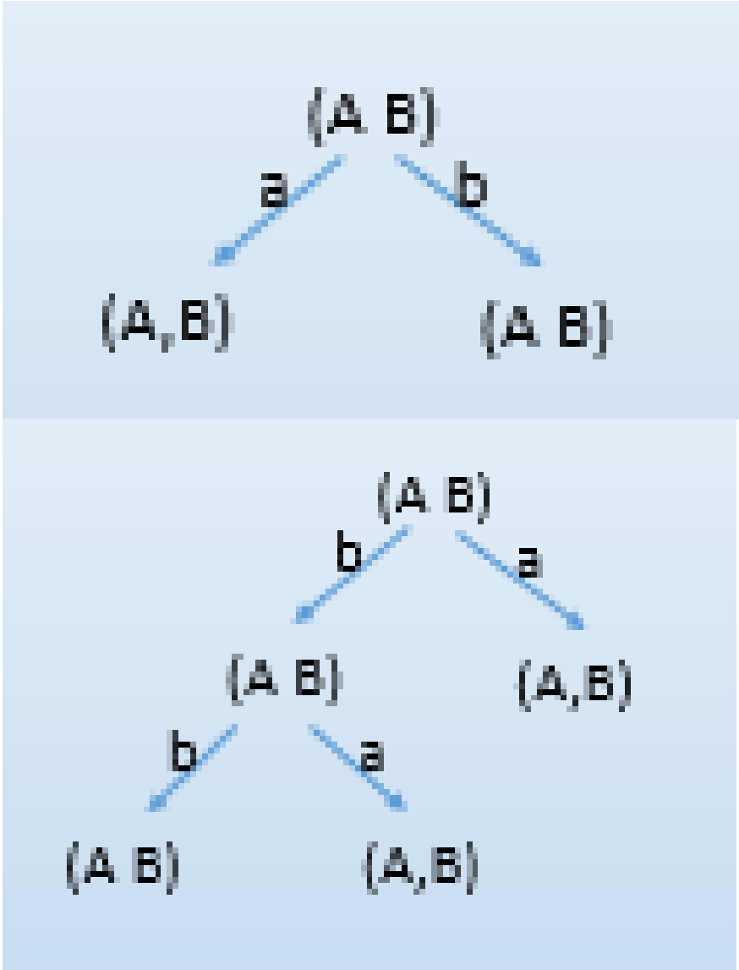

Figure 4. PDS Existence check for *M*'

## MORE ON DETAILS

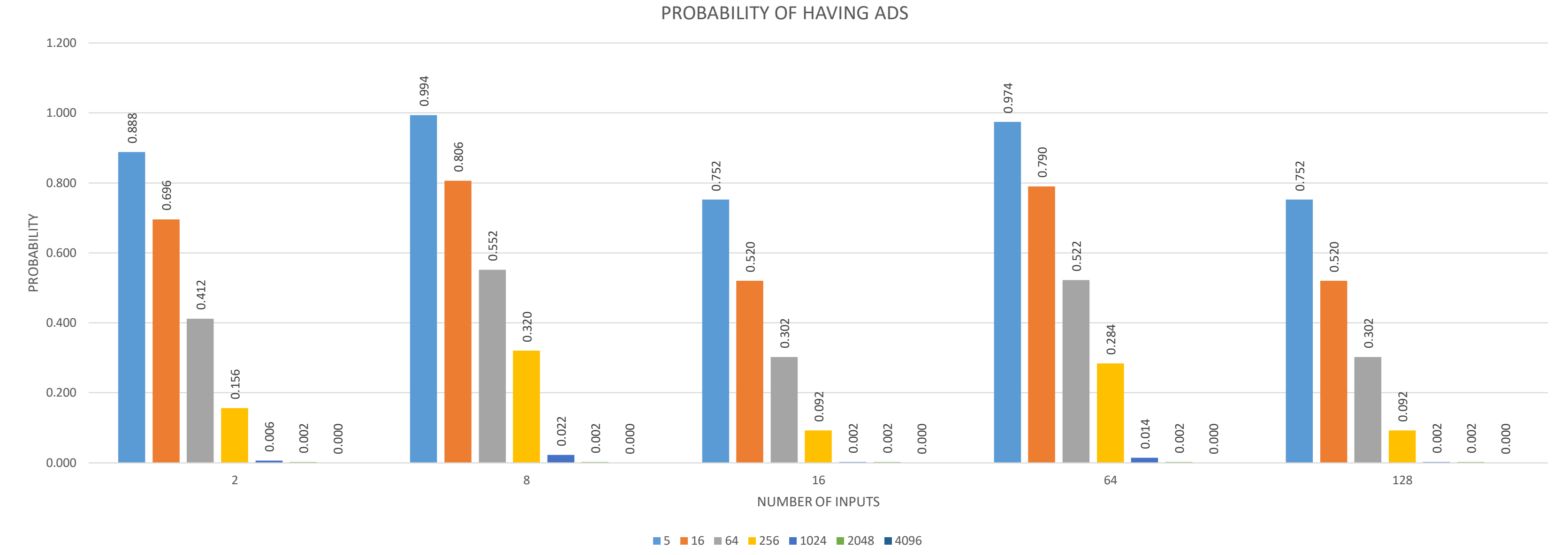

Figure 5. Shows the probabilities of having an ADS

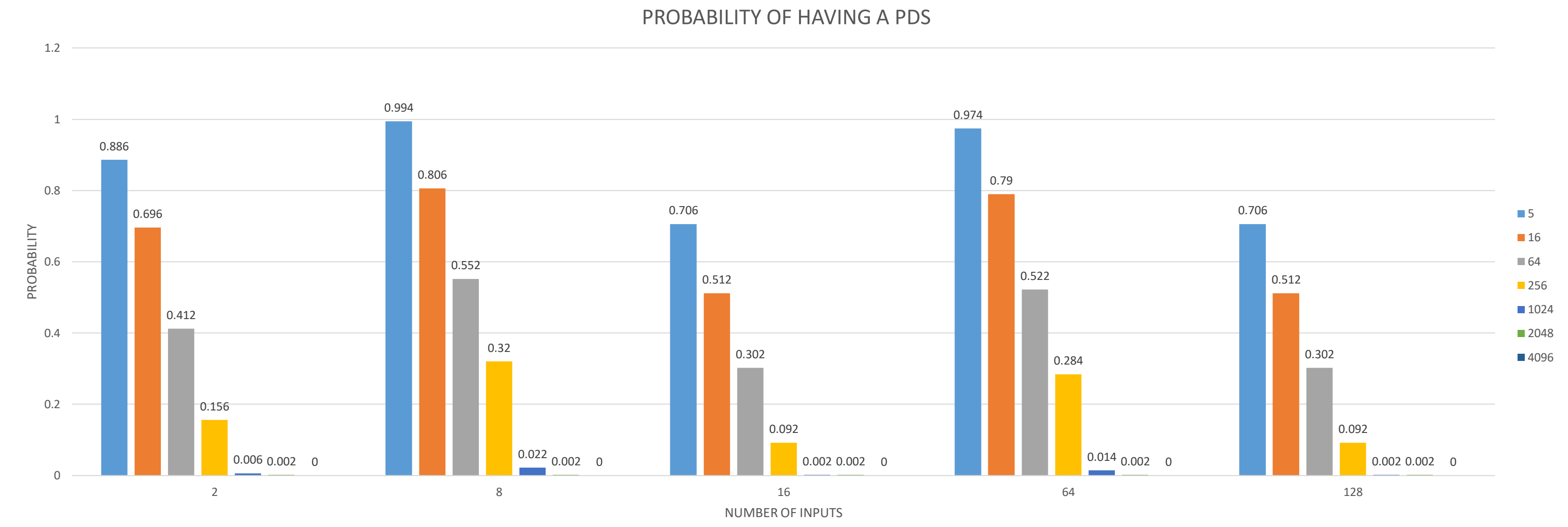

Figure 6. Shows the probabilities of having a PDS

Intuitively we thought that the probability of ADS & PDS existence will decrease when the number of states increases. To test our intuition, we generated 100 random minimal FSMs for each $n$ (state number), $p$ (input number) and $q$ (output number) where

$$n \in \{5, 16, 64, 256, 1024, 2048, 4096\},$$
$$p \in \{2, 8, 16, 64, 128\},$$
$$q \in \{2, 8, 16, 64, 128\}$$

We saw that ADS & PDS existence probabilities decrease as the number of states increases. The results of the experiments are given in Figure 5 for ADS and Figure 6 for PDS.

The drop in the probabilities is quite fast. For FSMs sizes above 4000 states, it is almost impossible to find an ADS or PDS. Also because of the exponential complexity of the PDS existence check algorithm, it takes a very long time to check FSMs with large number of states. We saw different trends with fixed number of states while the other parameters changed.

## CONCLUSION AND FUTURE WORK

Based on our experimental results, we observed that the probability of having an ADS and PDS decreases as the number of states of a random FSM increases. This result is the opposite of the probability of the existence of a synchronizing sequence for a random FSM.

Another unintuitive finding is the following. It is known that an FSM with a PDS always has an ADS. Therefore, the probability of existence of an ADS is believed to be higher than that of a PDS. Our experimental results suggest that this is correct for small state sizes. However, as the number of states increases, the probability of existence of an ADS and PDS become the almost the same. Hence, for large FSMs, an FSM with an ADS is highly likely to have a PDS as well.

In our future work we plan to optimize our implementations. We have already shortened our code and avoided memory leaks. For example we used circular queues for ADS algorithm which uses only a polynomial amount of space, which is allocated initially. PDS existence check requires exponential amount of space. However, it can be improved by a factor. To improve time complexities, we plan to make some changes in our PDS implementation.

We also plan to make some modifications in our algorithm for random FSM generation. Currently, we generate random FSMs and then check if they are minimal after their generation. This means we continue generating until we have 100 minimal FSMs. This takes a lot of time for larger FSMs and unnecessary computations. Instead, we plan to generate FSMs and force them to be minimal in the generation part. This will help us avoid from running unnecessary processes.

We generate only 100 random FSMs out of $qn^{np}$ possible FSMs for each $n, p$ and $q$. The reliability of this experiment size is up for discussion. We have to analyze our results to see how statistically significant they are, and if necessary, we have to increase the number of FSMs we consider in our experiments.

## REFERENCES

[1] *Aistleitner, Christoph & D'Angeli, Daniele & Gutierrez, Abraham & Rodaro, Emanuele & Rosenmann, Amnon. (2019). Circular automata synchronize with high probability.*

[2] *Lee, D., & Yannakakis, M. (1994). Testing finite-state machines: State identification and verification. IEEE Transactions on computers, 43(3), 306-320.*

[3] *Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, Alexander Pretschner (2005): Model-Based Testing of Reactive Systems, Lecture Notes in Computer Science 3472, Springer.*