SAL Doctoral College
Electronic Based Systems

CRYSTALLINE Program

1ST ASSESSMENT STAGE

Set of scientific tasks

for

P05-C2: Model Learning and Formal Methods

CONTENT

1. INTRODUCTION (OPTIONAL)

As part of your application, you are required to complete two tasks designed to assess your technical and analytical skills in the fields of software engineering and formal methods.
Task 1 involves the comprehension and application of formal definitions, demonstrated in the field of formal methods. You will read a paper on model-learning and manually apply the presented algorithm on a small example.
Task 2 requires the installation, application and small modification of a python library, where you are required to apply the tool AALpy on given black-box systems, to learn a model of their behaviour.

The goal of these tasks is to assess all candidates on their ability in scientific research, as well as their ability to use and develop scientific tools in Python. All solutions need to be described in the accompanying report.md file, and Task 2 also requires the upload of the file task2.py.

2. TASK 1

As a first task, read paper [1], which provides a nice introduction to automata learning. It presents two algorithms for the active automata learning, DHC and L_M^*. To demonstrate your understanding of the paper, perform the following sub-tasks.

Sub-tasks for Task 1:
- In your own words, describe the main differences and possible benefits of the two algorithms (~1/4 page)
- For the Mealy machine below, define (S, s_0, $\sum$, Ω)
- For the same Mealy machine, produce the final observation table by applying the L_M^* algorithm, and explain the steps involved (~1/4 page)
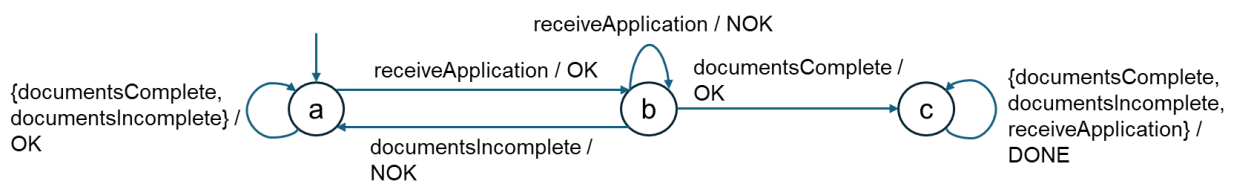
3. TASK 2

In Task 2, you need to use the Python tool AALpy to learn the models of four vending machine implementations. You do not know anything about their logic, you can only interact with them. More precisely, the user can add coins of various values to the vending machines, and press a button that orders one of several possible items. Based on the number of coins, that item might be returned. The task is to learn the models, automatically check their differences in behaviour, figure out which of them is correct, and report on the bugs of the other implementations.

Sub-tasks for Task 2:
- Install AALpy
- Learn models of all provided implementations

- Implement a learning-based testing approach that takes two models and returns differences between them
- Using learning-based testing
    - figure out which implementation is correct and describe its logic in the report
    - for faulty implementations, describe the bugs in the report, providing input-output examples leading to the fault.

Note that all faulty implementations contain exactly one fault. *Hint: The comparison of learned models should be automatic, if you are examining them by looking at them you are doing something wrong.*

To do this assignment you will need Python 3.10. To install all reacquired dependencies, run pip3 install aalpy dill. Make sure you have Graphviz[1] installed and added to the path so that you are able to visualize the learned models.

To run the framework you will need to execute python3 task_2.py

Help in how to use and modify AALpy can be found on https://github.com/DES-Lab/AALpy

4. REFERENCES

[1] Introduction to Active Automata Learning from a Practical Perspective (researchgate.net)

---

[1] https://graphviz.org/