CS 481 Project 2: Cybernatic
Group members: Tushar Chawla, Prasanna Prakash

Documentation: Automaton Representation: The Automaton is based on the classes of following model:

The basic model of the project can be found under the Documentation of Project 1 Directory.
Structure:

The structure of the code remains the same for this assignment. The additional classes/Examples that have been added, can be found under the following:

Structure:
Main -> Java -> Automata -> Examples

- ➔ First Order Conditioning
- ➔ First Order Conditioning Extinction (Part A)
- ➔ Second Order Conditioning (Part B)
- ➔ Second Order Conditioning Extinction (Part C)

PART A : First Order Conditioning

In this example, we followed the steps as mentioned in the books and used the constants as asked in the book. The procedure as the following:

The initial setup included having an automata with three states, the Q1 initial state. In this setup, we are to simulate an unconditioned response and input. So, we would be giving the automata UCS+ and UCS- (where + and – are the start of input and end of input resp.).

Then, the setup is processed in a way that it goes back to the initial state and then we process inputs.

NOTE: For the purpose of the project testing and debugging, we have used Integer value of -1 as an alternative for no input for a while. This helps make the automata simpler for testing.

The automata in the above case is given the following set of inputs again and again to change the expectation and output as required:

```
CSPLus
UCSPlus
CSMinus
```

```
UCSMinus
Epsilon
```

Where Epsilon marks the no input received for a certain time period in our test environment.

The Extinction is checked by the following sequence of inputs:

```
CSPLus
CSMinus
Epsilon
```

The expectation is that, the automata will eventually unlearn what it learnt after repeating the above extinction for certain number of times.

PART B : Second Order Conditioning

The Second order conditioning is given the same initial setup as the first order conditioning but with the input sequence changes as follows:

```
CS1PLus
UCSPlus
CS1Minus
UCSMinus
Epsilon
CS2PLus
CS1PLus
CS2Minus
CS1Minus
Epsilon
```

Where again, the epsilon marks the time span for no input received. In this the one conditioned stimuli is followed after a different conditioned stimuli making it a second order conditioning. In this case, the expectation is that the procedure will make the learning so that even the second conditioned stimuli would start learning and producing the outcome of the Unconditioned Stimuli.

PART C : Second Order Conditioning Extinction

Just like Part A’s Extinction there is a certain set of sequence of input that is followed in order to un-learn what was learnt in the above procedure. The input sequence is as follows:

The expectation is that repeating the above sequence a certain number of times would make the learning slowly fade and eventually un-learn what was learnt. The Extinction is basically achieved by giving the conditioned stimuli, which was earlier followed with a unconditioned is now removed. So after conditioned Stimuli, we don't provide an unconditioned stimuli and it un-learns.

NOTE:

As mentioned in the earlier Project 1 documentation, the code is set up as suggested in the provided pseudo code but the conditions do not change as expected. We tried several things to fix those problems which was causing a drift away from the original problem and hence started causing even more problems. The code has been tested for some number of inputs in a loop for each example and updates the functions with some values, where as some of the values are not correct.

How to run the code:

Unzip the project, go inside the project directory and type:

```
gradle clean run
```