FORMAL SPECIFICATION DOCUMENT

Formal Specification and Verification of an ATM Machine

December 20, 2022

Project Advisor
Sir Waqas Ali

Submitted By
2020-SE-01 Muhammad Talha

2020-SE-04 Qamar ul Zaman

2020-SE-12 Haider Ali Faizi

Department of Computer Science
University of Engineering and Technology, Lahore, New-Campus, KSK

Contents

1 Abstract

Formal verification is a technique for ensuring the correctness of systems. This work focuses on verifying a model of the ATM System against some specifications. We will construct the model as a state diagram in this document as shown in figure 1. We will express our specifications in LTL and CTL formulas and then verify them to make sure the security, liveness, reach-ability, reliability, and security of the system.

2 Introduction

ATM systems are a typical example of a safety-critical and real-time system. The ATM is the system that is used to access bank accounts to make cash withdrawals. Whenever users want to make withdrawals, they can enter their ATM card and verified PIN, then the user selects the withdrawal option and enters the withdrawal amounts and it will display the amount to be withdrawn in the form of 500s and 1000s. The user is also able to perform one or more transactions.
The ATM unit consists of a display screen, a card reader, a cash dispenser, an envelope slot, and a printer. ATM is in an idle state when there is no operation. The card reader determines the account number from the entered card. The user is prompted to enter a PIN after a card is entered. A menu is displayed with the options like Withdraw, Balance Inquiry, and Exit. The card is ejected when the session is completed. Transactions can be canceled at any prompt by the user pressing the CANCEL, button. Security is the foundation of a good ATM system. For this, a PIN must be entered within 20 seconds. The user must enter the pin correctly within 3 attempts.

3 Scope

The function of the ATM is to support a computerized banking network. A person will be able to transect the amount, withdraw the amount and check the balance in the account.

4 Problem Statement

ATM system has a drastic change from the older working banking system, customer feel inconvenienced with the transaction method as it was in the hands of the bank employees. In our ATM system, the problem is overcome here by the transactions done by the customer thus making customers feel safe and secure. Therefore, the system helps the customer in checking the balance and transaction the amount by validating the PIN therefore ATM system is more user-friendly.

5 Objective

Objective means the goal of the system. In ATMs, the main objectives are:

- Safety
- Security
- Reach-ability
- Liveness
- Fairness
- Reliability

Safety

In the formal specification, the safety properties of a system mean a lot. In ATMs, the main concern of safety is about the transaction.

- If a person is doing the transaction, then in the case of a mishap, like a light off, the person's money remains in his/her account if he/she has not done the transaction.
- If the machine has not enough amount that the user wants, then the amount should be rolled back into the machine instead of withdrawn.
- If the credit limit reaches, then the system will not allow that card to transact the amount.

Security

- System checks the card whether the card is right or not.
- The system will take a PIN to check whether it is the right person or not.
- The system will take care of the user attempts and will block the card in case of 3 wrong attempts.

Reach-ability

- User can withdraw the amount.
- User can view balance in his/her account.
- User can cancel the transaction at any state.

Liveness

The liveness property means that if some input has been given, then eventually some output will be given.

- If a user enters the card, then the system allows selecting language i.e., the output of the given input.
- When a user enters the card, the system allows entering the PIN.
- If a PIN is valid, then the system shows the main page containing different functions to perform.
- If a PIN is not valid, then the system again asks to enter the PIN.
- If a user withdraws the amount, then security and safety measures are checked. If all are valid then the amount is withdrawn.

Fairness

If something is attempted/requested infinitely often, then it will be successful/allocated infinitely often.

- If a user wants to withdraw the amount infinitely, then the amount will be withdrawn till the limit is reached.
- If a user wants to cancel the transaction, then at any time it can be canceled.

Reliability

Reliability means that the machine should be user-friendly and will not deceive users.

- If a user withdraws the amount, then the amount should be withdrawn the user has selected.
- The balance should be updated the amount properly after withdrawing.

6 Requirements

6.1 Functional Requirement

Functional requirements for an ATM might include:

- Accepting deposits and cash withdrawals from users
- Authenticating users through the use of a personal identification number (PIN)
- Displaying account balances and transaction histories
- Providing users with the option to transfer funds between accounts or to make bill payments
- Printing receipts for transactions

6.2 Non-Functional Requirement

Non-Functional requirements for an ATM might include:

- Ensuring that the system is secure and resistant to tampering or fraud
- Ensuring that the system is available for use at all times identification number (PIN)
- Providing a user-friendly interface for interacting with the system
- Ensuring that the system is able to handle high volumes of transactions efficiently

7 Preliminaries

Timed automata are a formalism used to model real-time systems, which are systems that operate in real time and are subject to timing constraints. They are an extension of finite automata, which are used to model systems with discrete states and transitions between those states.

The syntax of a timed automaton consists of:

- A set of states, which represent the possible configurations of the system at a given point in time.
- A set of clocks, which are variables that represent the passage of time. Clocks can be reset to 0 or stopped at any point in a state.
- A set of transitions, which represent changes of state in the system. Transitions are labeled with conditions that must be satisfied in order for the transition to be taken. These conditions may include constraints on the values of the clocks.

The semantics of a timed automaton specify the behavior of the system over time. This includes the rules for updating the values of the clocks and the conditions under which transitions between states are taken.

TCTL (Timed Computation Tree Logic) is a formal logic used to specify properties of timed automata and to reason about their behavior. It allows users to express temporal logic formulas that can be used to verify properties of a system, such as whether it satisfies certain safety or liveness properties.

The syntax of TCTL includes operators for expressing temporal relationships, such as "eventually" and "always", as well as operators for quantifying over time intervals and clocks. The semantics of TCTL specify how these operators are interpreted and how they can be used to reason about the behavior of a system.

8 Transition Diagram of System

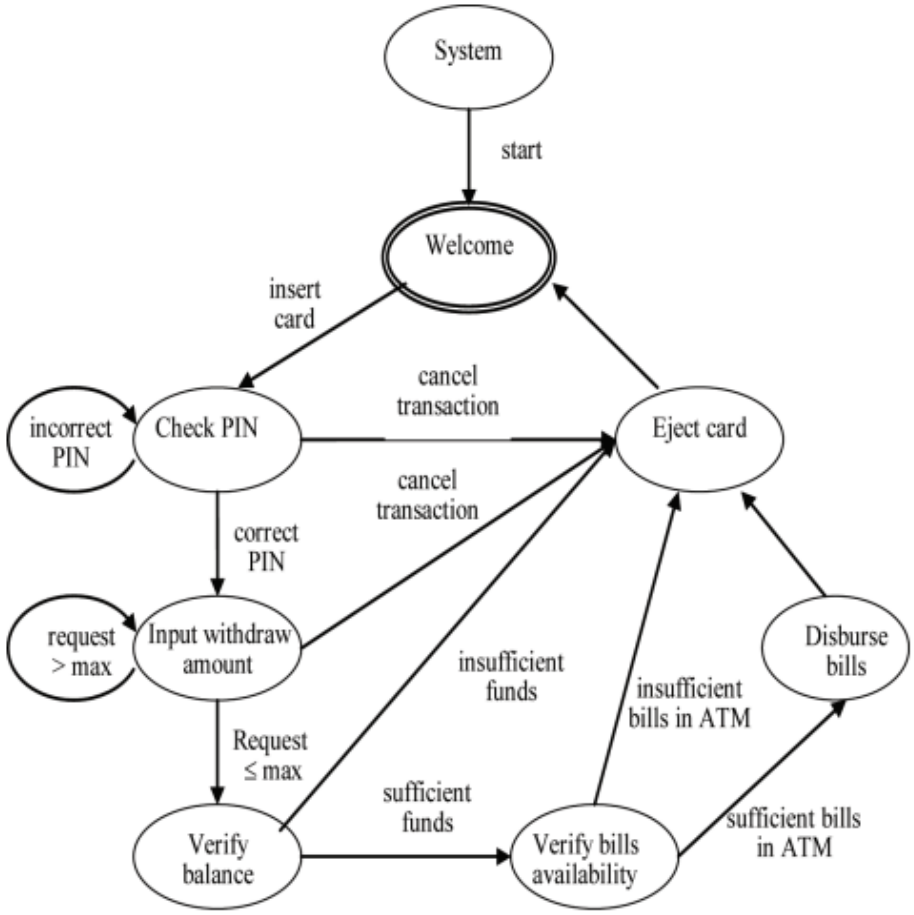

Figure 1: Flow Diagram

Explanation

A flow diagram for an ATM machine (Automated Teller Machine) might include the following steps:

- The user approaches the ATM and inserts their bank card into the card reader.
- The ATM prompts the user to enter their personal identification number (PIN).
- The user enters their PIN and the ATM authenticates the user.

- The ATM displays a menu of options for the user to choose from, such as checking their account balance, making a withdrawal, or transferring funds.
- The user selects an option and follows any additional prompts to complete the transaction.
- The ATM processes the transaction and dispenses any cash or receipts as necessary.
- The ATM returns the user's card and prompts the user to take their card.
- The user takes their card and the transaction is complete.

9 Implementation

9.1 Machine

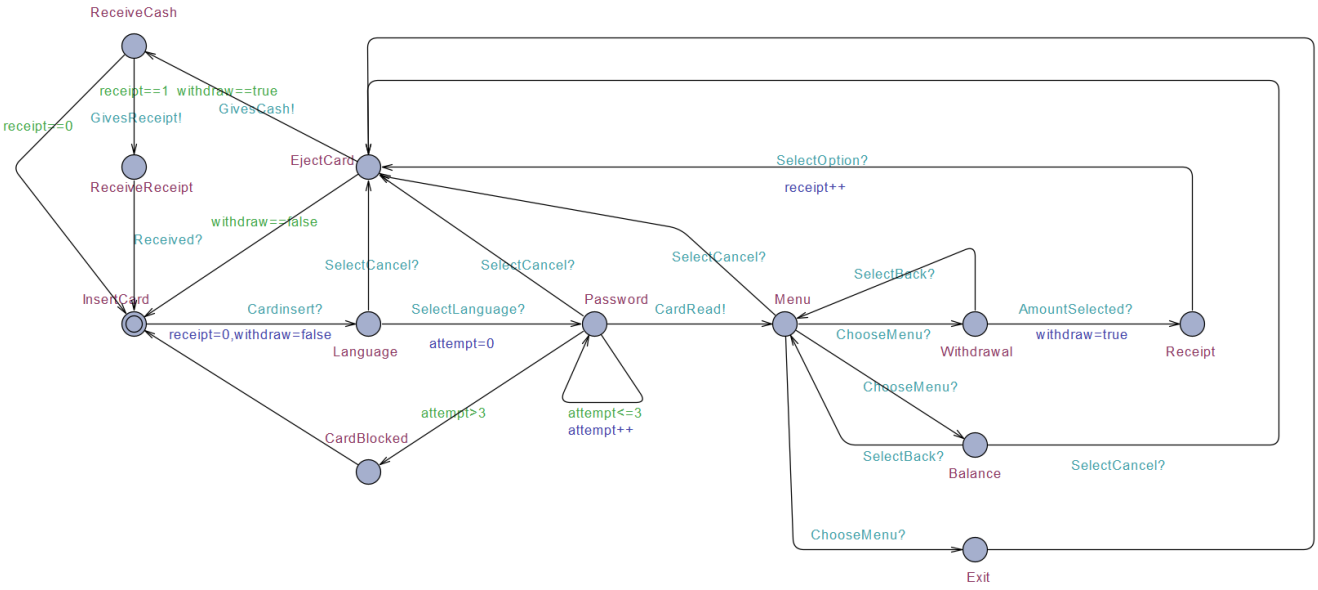

Figure 2: Screen

9.2 User

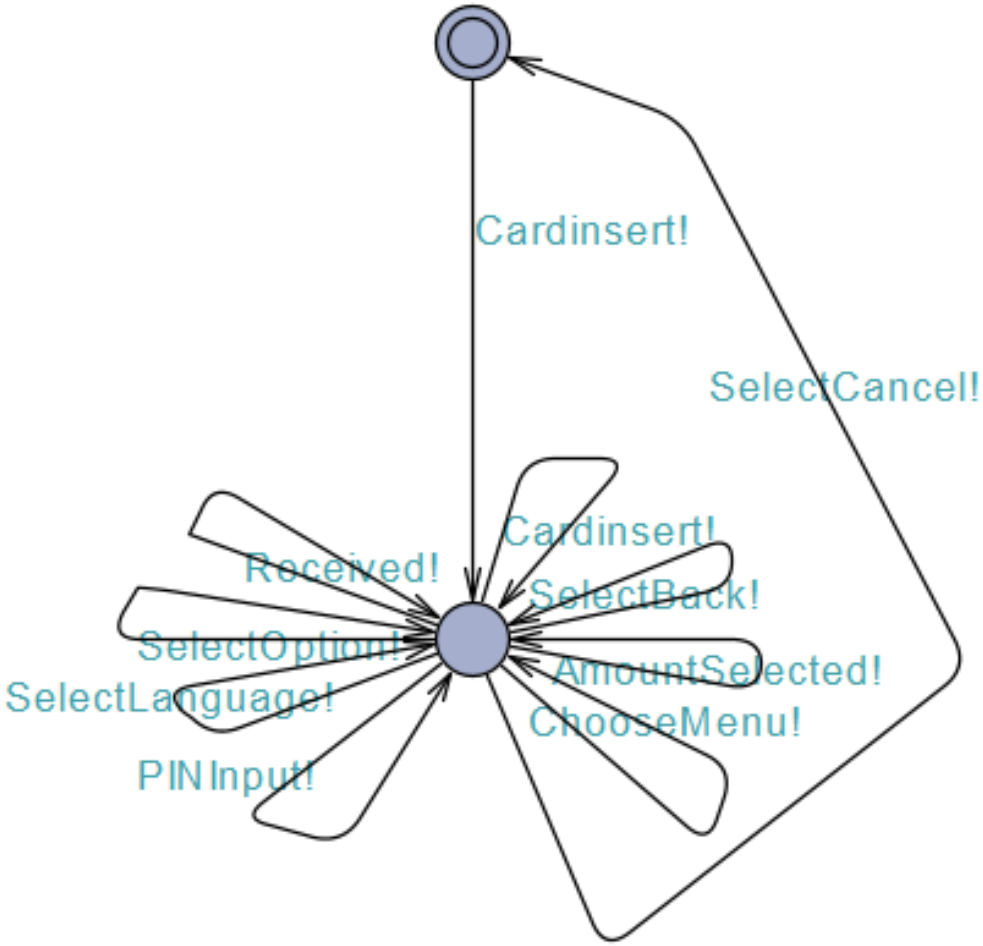

Figure 3: User

9.3 Cash Dispenser

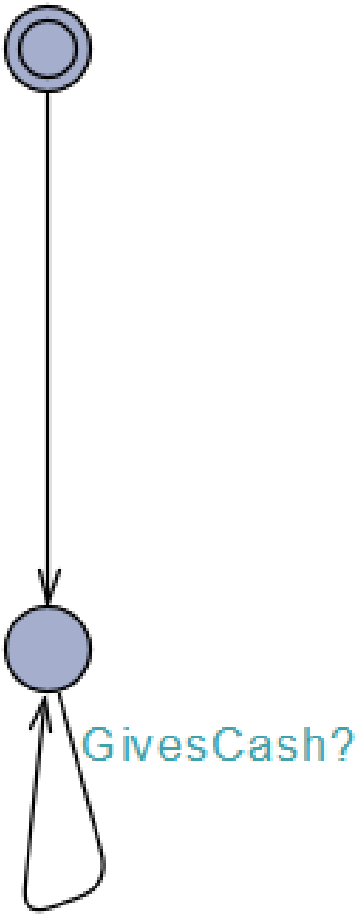

Figure 4: Cash Dispenser

9.4 Card Reader

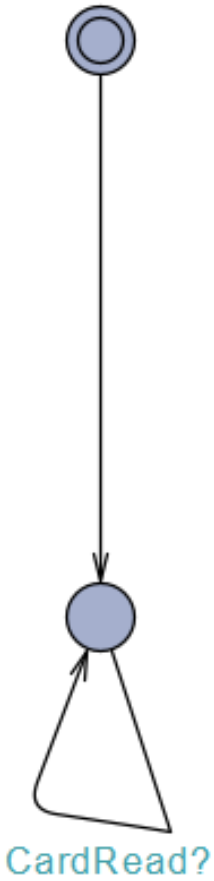

Figure 5: Card Reader

9.5 Receipt

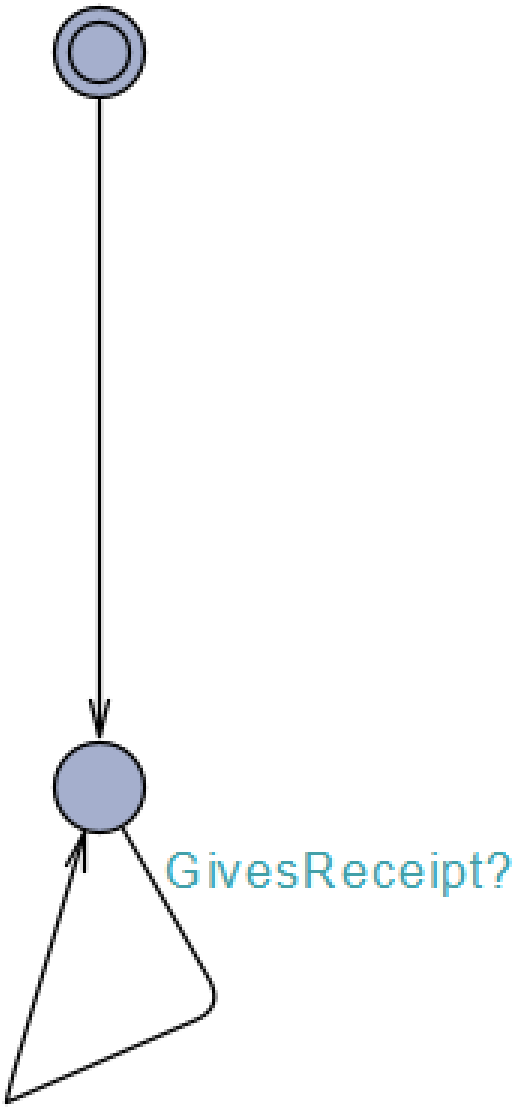

Figure 6: Receipt

Explanation

- First, the user will insert the card. i.e., from Figure 3, cardinsert! the channel will run and give a call to cardinsert? in Figure 2. It will show that the card is inserted. If the card is valid then it will move to the next step otherwise it will eject the card by using selectcancel! channel from Figure 3 which will call selectcancel? from Figure 2.
- Now different languages are shown. In the next step, the user will select a language. i.e., from Figure 3 SelectLanguage! the channel will call SelectLanguage? from Figure 2 and language will be selected for the next step.
- In the next step, the machine will read the card using CardRead! that will call CardRead? component from Figure 5 and then will ask to enter the PIN. If the PIN is correct then move to the next state otherwise ask again. In case of 3 wrong attempts, the card will block.
- Now, users have different functions to perform. i.e., withdraw the amount, check the balance, cancel the transaction, and back to the menu.
- If the user wants to withdraw the amount, ChooseMenu! the channel from Figure 3 will call ChooseMenu? from Figure 2 and select one of the choices.

- User will check the balance and can move back by choosing the menu again. i.e., calling SelectBack? the channel from Figure 2 by the channel SelectBack! from Figure 3.
- When the user will withdraw the amount, the channel AmountSelected! from Figure 3 will call the channel AmountSelected? from Figure 2 which will prompt that amount is selected.
- On the next step, if the amount selected is valid then a card will be ejected and the cash dispenser will work. i.e., Givescash! channel from Figure 2 will call the channel Givescash? from Figure 4 which will give cash to the user.
- If the amount selected is not valid, then simply the card will be ejected and the machine will again be ready to intake a new card and start its function normally.
- After receiving cash, the user will receive a receipt too. i.e., the channel GivesReceipt! from Figure 2 will call the channel GivesReceipt from Figure 6 which will prompt the machine to generate the receipt from the receipt component.
- At the end, the channel received will work which will verify that the receipt and cash have been received by the user.
- At last, the machine starts again and will be able to insert the card.
- That's how the machine will work.

10 Formal Specification

Formal Specifications		
Informal	Formal	Results
There is no deadlock in our system.	A[] not deadlock	Satisfied
For all paths in our system, we will finally be able to insert the card again.(Liveness)	A⟨⟩$ATMScreen.InsertCard == true$	Satisfied
There exists a path in our system where we finally enter our password.(Safety)	E⟨⟩$ATMScreen.Password == true$	Satisfied
There exist a path when you are on the password screen that it will again ask you to enter password.	E⟨⟩($ATMScreen.Password$ $implyATMScreen.Password == true$)	Satisfied
There exist a path in our system where finally our card will be blocked.(Security)	E⟨⟩($ATMScreen.Password$ $implyATMScreen.CardBlocked == true$)	Satisfied
There exist a path when you are on the password screen that it will lead you to menu screen.	E⟨⟩($ATMScreen.Password$ $implyATMScreen.Menu == true$)	Satisfied

Formal Specifications		
Informal	Formal	Results
There exists a path when we will finally be able to enter withdawal amount.	$\mathrm{E}\langle\rangle ATMScreen.Withdrawal == true$	Satisfied
There exist a path where we receive our card after the amount is entered.	$\mathrm{E}\langle\rangle(ATMScreen.Withdrawal\ implyATMScreen.EjectCard == true)$	Satisfied
There exist a path where we receive our cash after the card is received.(Reachability)	$\mathrm{E}\langle\rangle(ATMScreen.EjectCard\ implyATMScreen.ReceiveCash == true)$	Satisfied
There exist a path where we receive the receipt of transaction after the cash is received.	$\mathrm{E}\langle\rangle(ATMScreen.ReceiveCash\ implyATMScreen.ReceiveReceipt == true)$	Satisfied
There exists a path when we will finally be able to receive the cash required.	$\mathrm{E}\langle\rangle ATMScreen.ReceiveCash == true$	Satisfied
There exist a path where the system is again able to insert card after the cash is received.	$\mathrm{E}\langle\rangle(ATMScreen.ReceiveCash\ implyATMScreen.InsertCard == true)$	Satisfied

Formal Specifications		
Informal	Formal	Results
There exists a path where we can view the balance in our card.	$\mathrm{E}\langle\rangle ATMScreen.Balance == true$	Satisfied
There exist a path where we go back to menu from the withdrawal screen.	$\mathrm{E}\langle\rangle(ATMScreen.Withdrawal\ imply ATMScreen.Menu == true)$	Satisfied
There exist a path where we go back to menu after viewing the balance	$\mathrm{E}\langle\rangle(ATMScreen.Balance\ imply ATMScreen.Menu == true)$	Satisfied

11 Discussion

In this whole project, we worked in steps. First, we did brainstorming about the selection of the project. Then after selecting the project we made rough sketch of ATM machine that what are the components in it and how ATM machine works.

First, we wrote introduction and abstract of the system. In this step, we explained what will the project be like and explained each component of the system.

Next we defined the scope of the system and examined the problem statement of the system and made objectives. Our main focus was to follow the formal specifications terms like safety, security, reach-ability, liveness, fairness, and reliability. These terms are major factors while building any project as without them system cannot be perfect.

In the next step, we made model on UPPAAL tool. UPPAAL is a tool for the automatic verification of real-time systems, which can be used to create a formal specification of an automated teller machine (ATM). To create a formal specification of an ATM using UPPAAL, you would first need to define the various components of the ATM and their interactions, using the modeling language provided by UPPAAL.

One way to do this would be to create a state machine model for each component of the ATM, such as the cash dispenser or the card reader. The state machine model would define the various states that the component can be in, as well as the transitions between those states and the actions that are triggered by those transitions.

In formal specification of UPPAAL model of ATM, we have written different specifications which satisfied the system.The specifications include propositions, predicates, linear temporal logic, and computational tree logic(CTL). Some of the queries checked whether system is in deadlock state or working well. We satisfied liveness property like system will eventually be able to intake card wherever the path is followed..

Safety property is also satisfied by the system like user will insert card and system will ask PIN to check validity which shows the safety of the system. We took care of security property as if the user enters wrong PIN 3 times then user's card will be blocked.

Similarly in case of reach-ability property, system is satisfying this property. User will finally be able to receive cash and receipt in case of successful transaction. Also user will finally be able to check the balance and can cancel the transaction at anytime.

If we talk about the reliability of the system, if transaction is not successful due to any reason, the amount in the account will remain same. It builds customer trust on the system and thus it satisfies the reliability. Also the system is available 24/7 times.

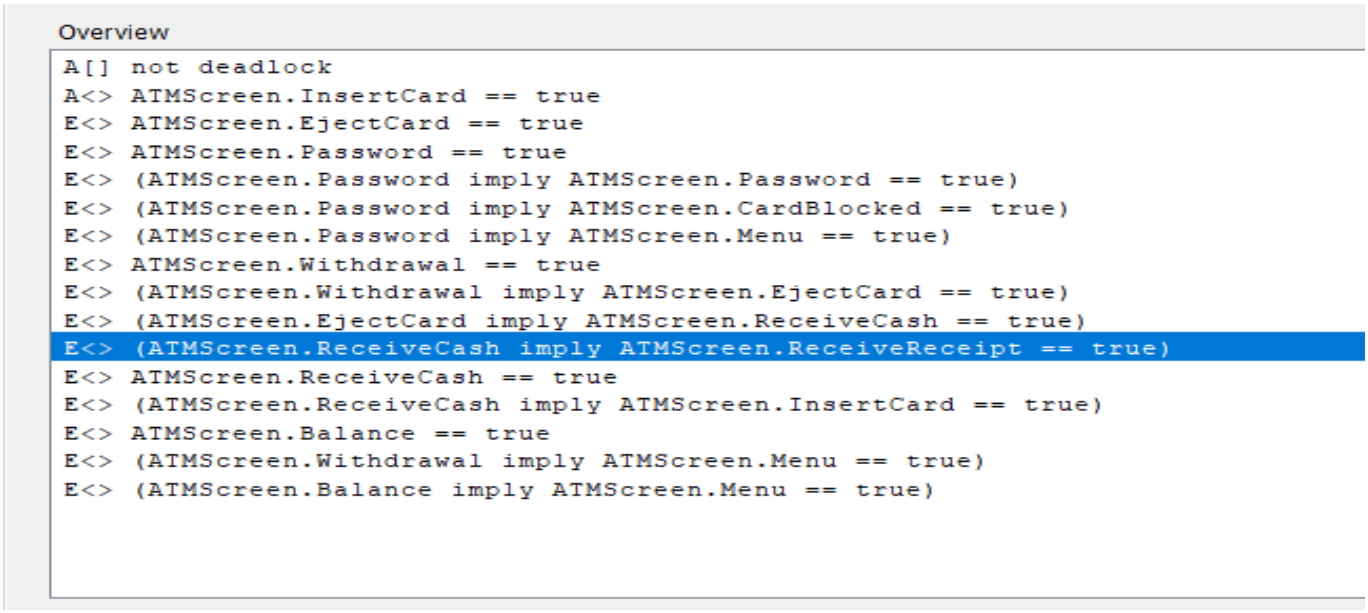

Figure 7: All Queries

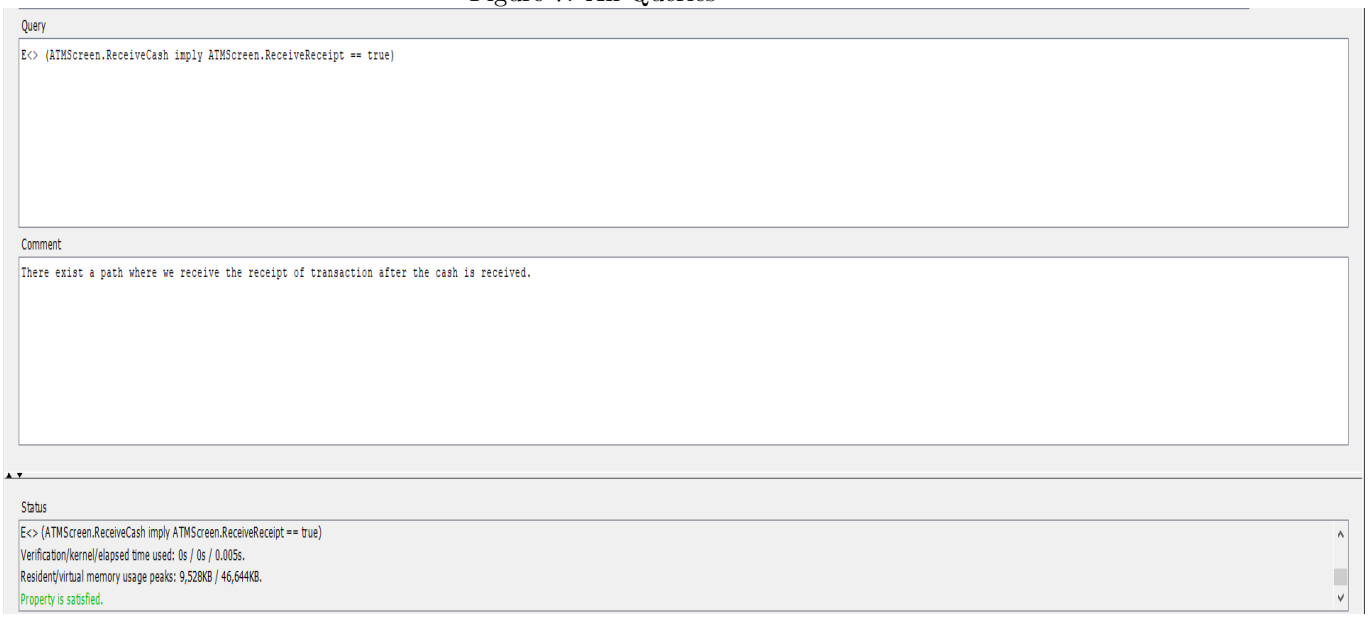

Figure 8: Satisfied Proof

12 Conclusion

Formal methods can be used to create a precise and detailed specification of the ATM's behavior and functions, including input and output, as well as any constraints or limitations on the system's behavior. This specification can then be used to design and implement the system, using techniques such as model checking or theorem proving to verify the correctness of the implementation.

Overall, using formal methods to design and implement an ATM can help to create a reliable and efficient system that meets the needs of users and stakeholders. It is important to carefully consider the trade-offs involved and to choose the most appropriate approach for your specific needs and goals.

13 Acknowledgement For the Project

We would like to express our gratitude and appreciation to all who gave us the opportunity to complete this project.

Also, we take this opportunity to express our deep sense of gratitude to our teacher, Mr./Mrs Waqas Ali under whose valuable guidance, this project work has been carried out.

14 Impression

At start, we found this project difficult as we had no idea about how to use UPPAAL model. But with the passage of time, we got understanding of the project and tool too. Thus, this experience was little bit tough but newer as we had not done any project like this before.

15 References

https://www.researchgate.net/publication/220636950_The_Formal_Design_Model_of_an_Automatic_Teller_Machine_ATM

https://acknowledgementletter.com/acknowledgement-for-englis

https://www.academia.edu/9309171/The_Formal_Design_Model_of_ATM

JAZAK ALLAH!!!