Software Formal Specification

Program:

Course Code: CSE326

Course Name: Software Formal Specification

Examination Committee

Dr. Islam El-Maddah

Ain Shams University
Faculty of Engineering
International Credit Hours Engineering Programs (I-CHEP)
Spring Semester – 2020

Student Personal Information for Group Work

Student Names:	Student Codes:
Engy Samy Salah	16p3004
Mayar Wessam Nour	16p3008

Plagiarism Statement

I certify that this assignment / report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they are books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment / report has not been previously been submitted for assessment for another course. I certify that I have not copied in part or whole or otherwise plagiarized the work of other students and / or persons.

Signature/Student Name: **Engy Samy Salah** **Date:** **1/6/2020**

Mayar Wessam Nour

Submission Contents

01

First Topic

System Description Mentioning the Importance of the System

In this project we have chosen the washing machine system and designed it using the smv language. This system is very important to every house worldwide as it helps save a lot of time in washing our clothes and cleaning them easily with no effort needed.

We divided the system into three states: {Ready, Washing, Drying, and Error}

1.1 The Ready state:

When the washing machine has completed its work (whether washing or drying) without any errors the system goes into a ready state for another action. This is the initial state. If the system had an error and the reset button is pressed the system goes back into a ready state.

1.2 The Washing state:

When the door is closed and the start button has an input of 1 the system goes into a washing state and starts washing the clothes and that is the main function of the system.

1.3 The Drying state:

When the washing machine is in a washing state then the complete flag is up indicating that it has finished , the door is closed and the dry button is pressed, the washing machine goes to the drying state and starts drying the clothes.If the washing machine is in the ready state, the door is closed and the dry button is pressed it goes into the drying state.

1.4 The Error State:

When the start button or dry button (input) is pressed but the door is opened, the system goes into an Error state .If the washing machine was in a washing state and it hasn't completed its work and the door was opened, it goes into the error state. Also if the same happened during the drying state, the system goes into the error state. The state is never changed till the reset button is pressed,only then it can go back to the ready state.

02

Second Topic

The Reasons to Develop This System Formally

The problems we could face:

- Using non formal languages can lead to ambiguous meanings and can lead to many interpretations.
- One statement can contradict the other easily if the system is developed non-formally.
- If the system has many specifications and written in a non-formal way it can be vague and not precious.
- It can miss listing all the errors that can happen and all the limitations.

In our case, some limitations like maybe going into the error state while washing or drying can be unhandled if the system is not well described. Moreover, if the specifications are not clear enough then having the ability to go from the ready state to the drying state direct can be missed or not handled.

So we developed this system formally for the following reasons:

- Consistency
- Unambiguous: to help interpreting a sentence into only one meaning.
- Completeness.
- Uncover defects early that can be missed using traditional specification methods.
- Allows reasoning (by following the state diagram we can understand better the system)
- Helps seeing the system in a more abstract view as in what the system does not how it should do it. This helps seeing the specifications separately from the design so it can be handled well.

03

Third Topic

Set of informal requirements

There are 3 inputs that the user has the control of: start, reset, open and close.

The washing machine has some conditions to start and to avoid any errors:

- User has to close the door of the washing machine before clicking the start or dry button.
- User can't open the door while the machine is in the washing or drying states.
- The user can't force the washing machine to enter the washing or the drying state if the door is open.
- If any error occurs due to starting without closing the door or opening the door before finishing then the user has to push the reset button to get rid of the error.
- The only way to put the machine into the washing state is to push the start button and close the door.

- To put the machine into the drying state, the user has to close the door and push the dry button. Whether it was in a ready state or in a washing state.

04

Formal Specification and Restriction

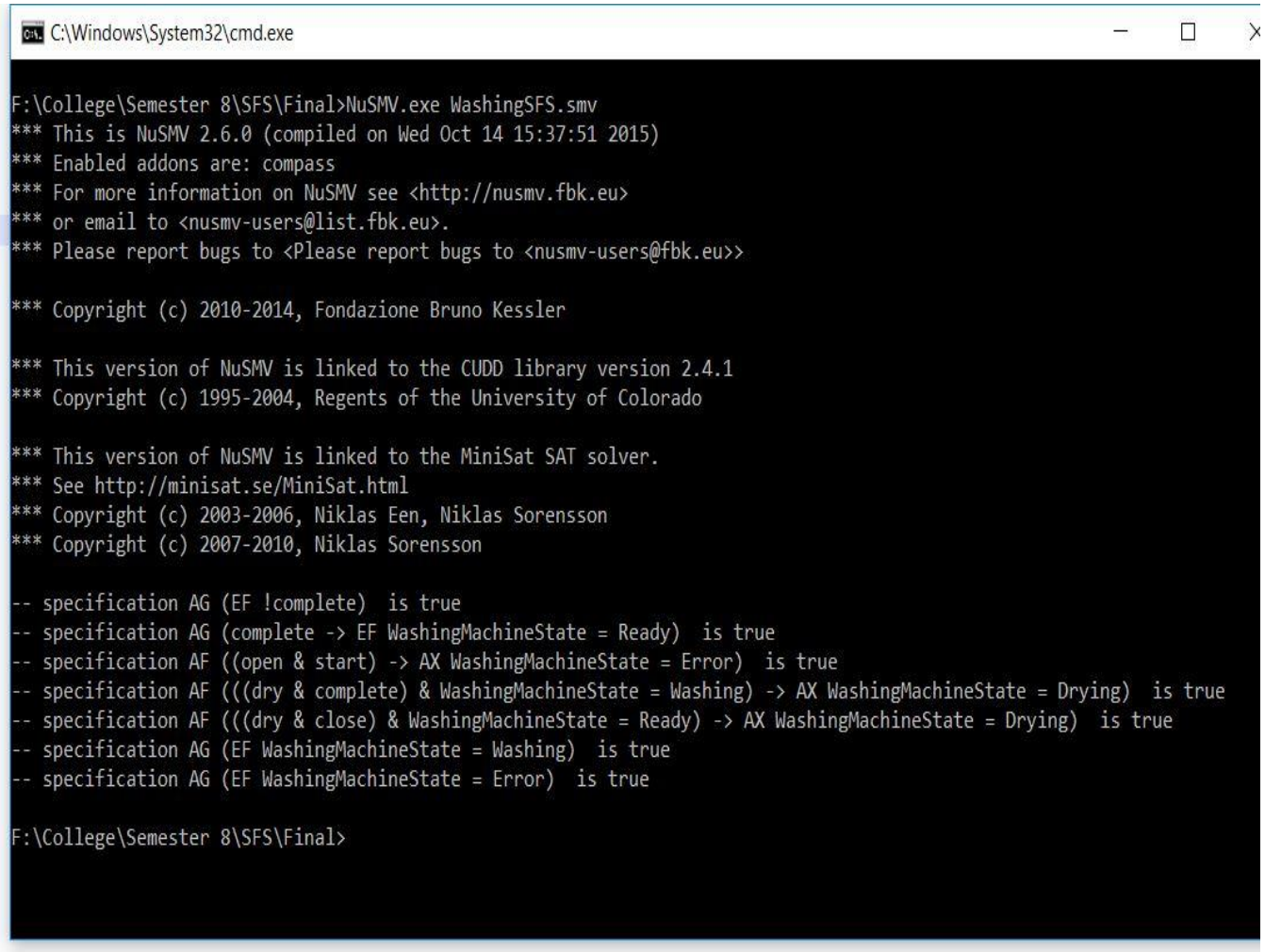
Fourth Topic

```
MODULE main

VAR

start: boolean;
reset: boolean;
open: boolean;
close: boolean;
complete: boolean;
dry: boolean;

WashingMachineState: {Ready,Error,Washing, Drying};

ASSIGN

init(WashingMachineState):= Ready;


next(WashingMachineState):=
    case
        WashingMachineState= Ready & start & open: Error;
        WashingMachineState= Ready & start & close : Washing;
        WashingMachineState= Ready & dry & close : Drying;
        WashingMachineState= Error & reset: Ready;
        WashingMachineState= Washing & open : Error;
        WashingMachineState= Washing & complete: Ready;
        WashingMachineState= Washing & complete & dry & close: Drying;
        WashingMachineState= Drying & open : Error;
        WashingMachineState= Drying & complete : Ready;
        TRUE: WashingMachineState;
    esac;

SPEC AG(complete -> EF(WashingMachineState=Ready))
SPEC AF((open & start) -> AX(WashingMachineState=Error))
SPEC AF((dry & complete & WashingMachineState=Washing ) -> AX(WashingMachineState=Drying))
SPEC AF((dry & close & WashingMachineState=Ready ) -> AX(WashingMachineState=Drying))
SPEC AG(EF WashingMachineState=Washing)
SPEC AG(EF WashingMachineState=Error)
SPEC AG(EF !complete)
```

05

Fifth Topic

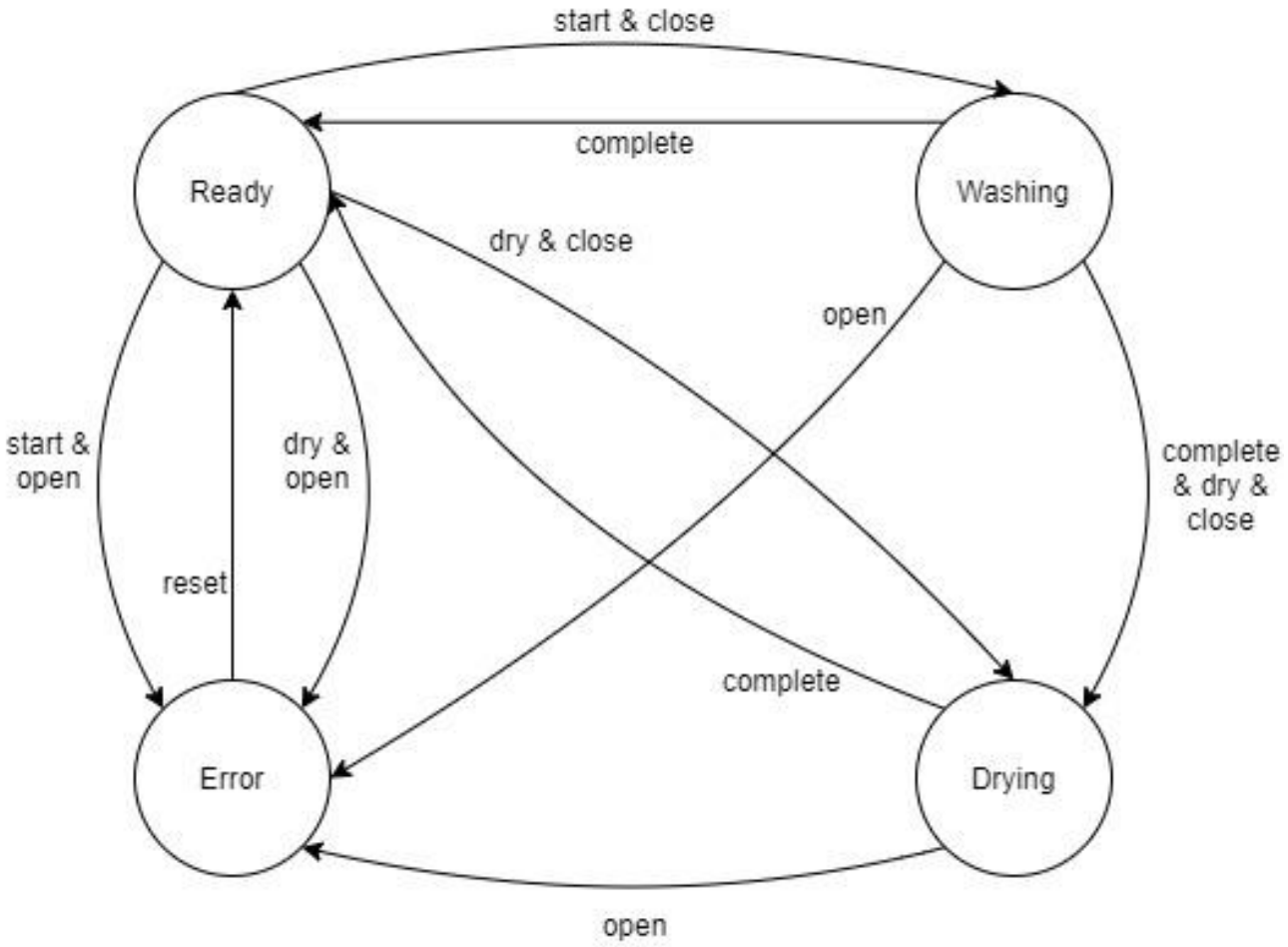
Nusmv Development Model Output Screen-shots



06

Sixth Topic

State Diagram



07 Appendix

Seventh Topic

```
MODULE main
VAR
start: boolean;
reset: boolean;
open: boolean;
close: boolean;
complete: boolean;
dry: boolean;
WashingMachineState: {Ready,Error,Washing, Drying};
ASSIGN
init(WashingMachineState):= Ready;
next(WashingMachineState):=
        case
                WashingMachineState= Ready & start & open: Error;
                WashingMachineState= Ready & start & close : Washing;
                WashingMachineState= Ready & dry & close : Drying;
                WashingMachineState= Error & reset: Ready;
                WashingMachineState= Washing & open : Error;
                WashingMachineState= Washing & complete: Ready;
                WashingMachineState= Washing & complete & dry & close: Drying;
                WashingMachineState= Drying & open : Error;
                WashingMachineState= Drying & complete : Ready;
                TRUE: WashingMachineState;
        esac;
```

SPEC AG(complete -> EF(WashingMachineState=Ready))

SPEC AF((open & start) -> AX(WashingMachineState=Error))

SPEC AF((dry & complete & WashingMachineState=Washing) -> AX(WashingMachineState=Drying))

SPEC AF((dry & close & WashingMachineState=Ready) -> AX(WashingMachineState=Drying))

SPEC AG(EF WashingMachineState=Washing)

SPEC AG(EF WashingMachineState=Error)

SPEC AG(EF !complete)