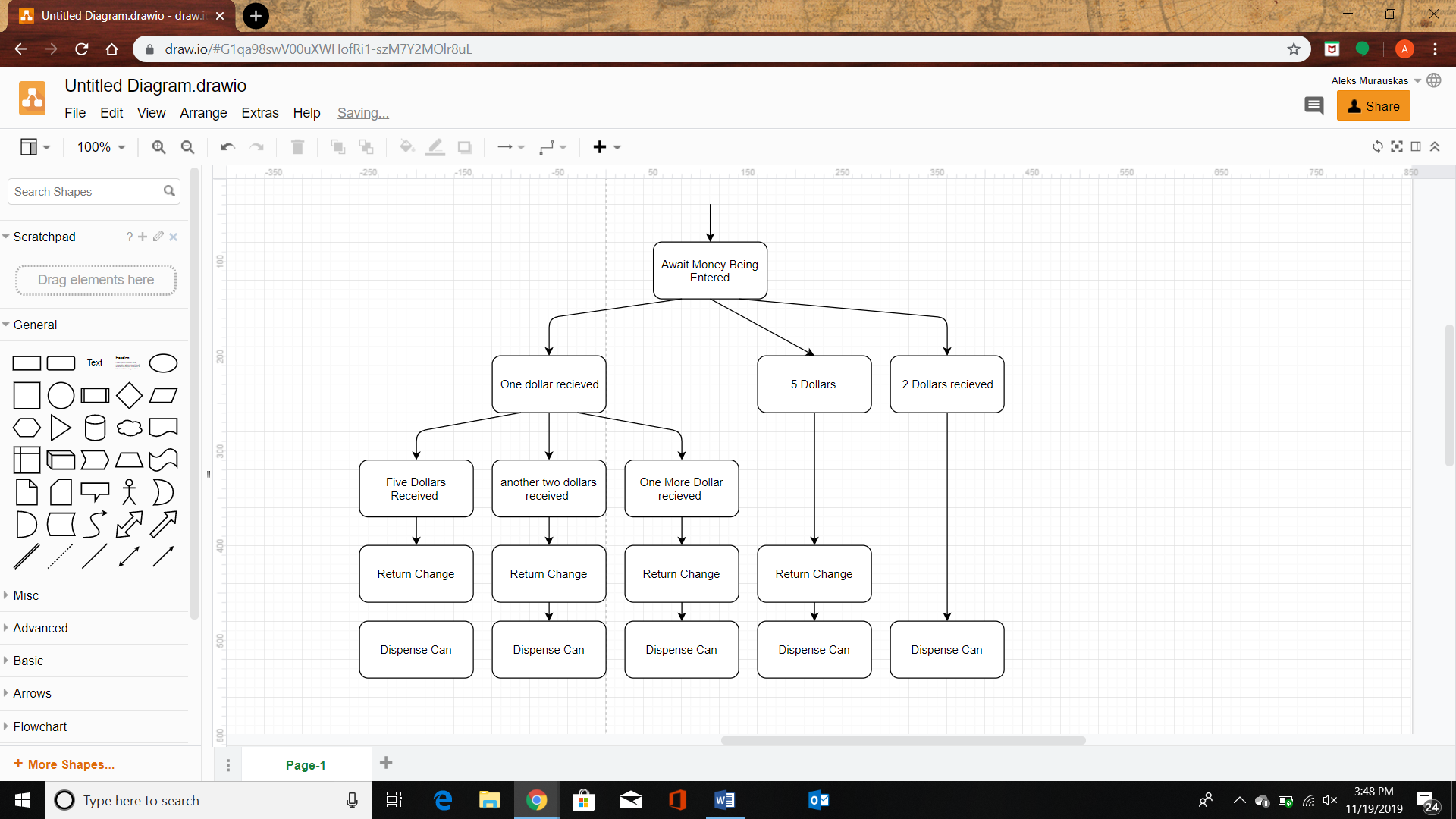
Aleksas Murauskas

ECSE 429: Assignment 3

Question 1

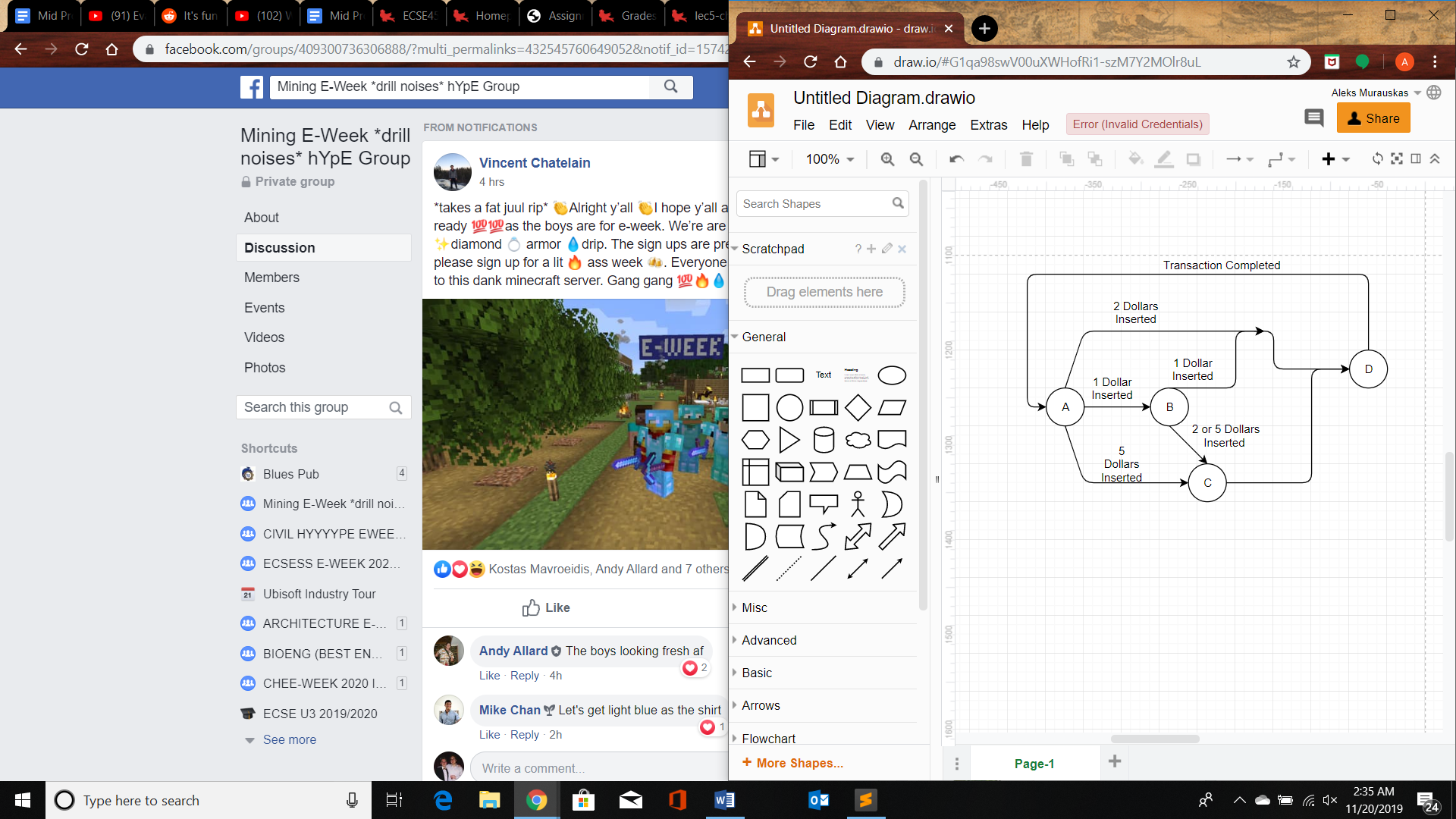
Flow Chart



States:

|  |  |
| --- | --- |
| State Name: | State Activity |
| A | No money in machine, await input |
| B | Money inserted, but not enough for transaction, await input |
| C | Too much money inserted, return Extra Change |
| D | Dispense Can |

State Machine



Pseudocode

vendingmachine(){

private int state;

private int inserted\_money=0;

initialize(){

state=0;

}

main(){

while(1){

if(state ==0){ //State A

waitForMoneyInsert();

if(money\_inserted>0){

if(money\_inserted ==1){

state=1;

}

if(money\_inserted==2){

state=3;

}

if(money\_inserted==5){

state=2;

}

}

}

else if(state==1){ //State B

waitForMoneyInsert();

if(money\_inserted==1){

state =3;

}

if(money\_inserted==2||money\_inserted==5){

state=2;

}

}

else if(state==2){ //State C

int change =money\_inserted-2;

return\_change(change);

state = 3

}

else if(state==3){ //State D

dispenseCan(); //Dispense can

state =0; //Set state to await next order

inserted\_money=0;

}

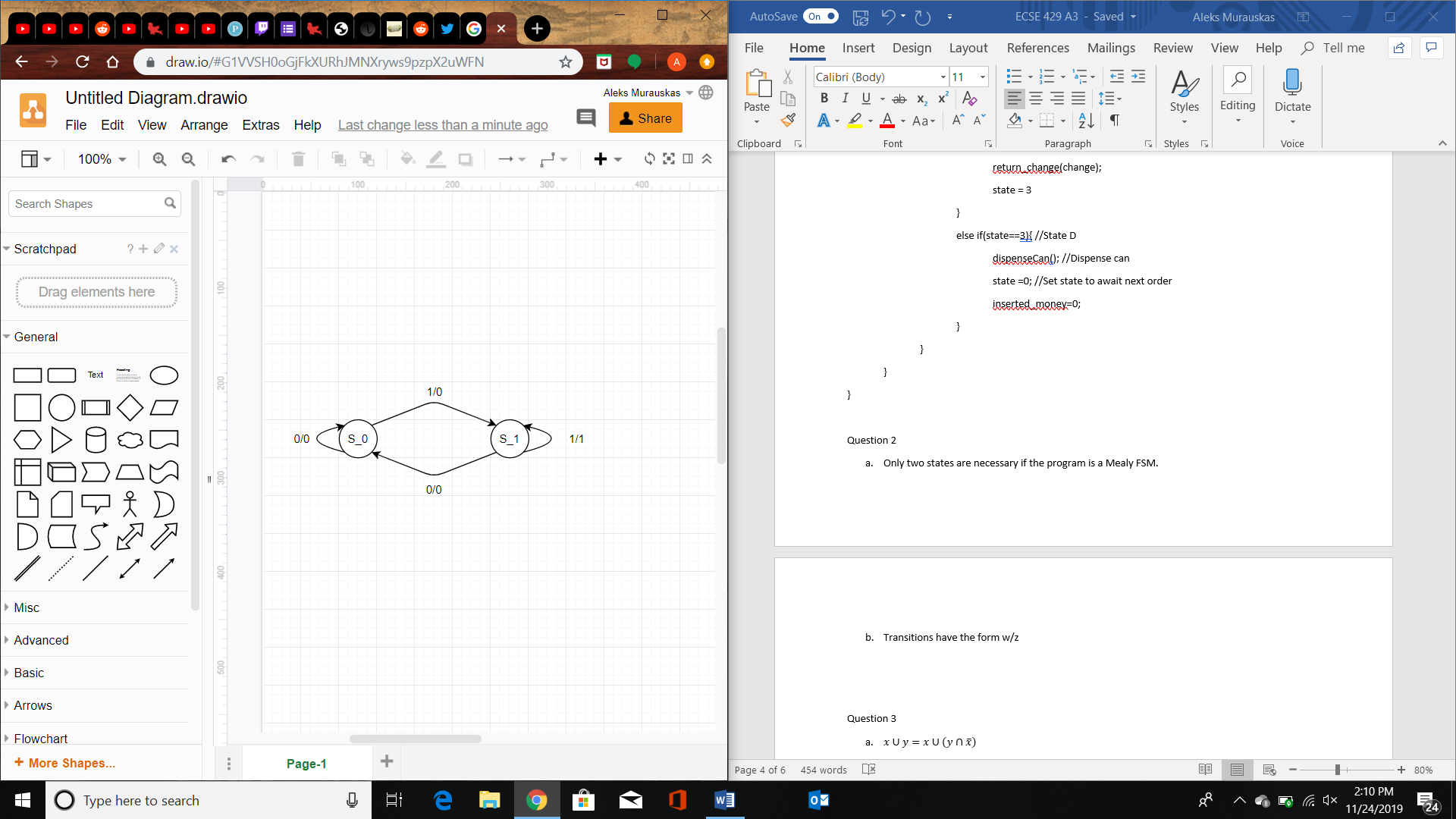
}

}

}

Question 2

1. Only two states are necessary if the FSM is a Mealy FSM.
2. Transitions have the form w/z



This is a Mealy type FSM, if it were a Moore I would need three total states to operate correctly.

Question 3

The clause(s) are:

The statement can be written in English as: The intersection of x and y is empty if and only if the intersection of x and not y is x, and vice versa

This statement is true as if the intersection of a and y is null, then not y must be the remaining possibility space outside of y, therefore x must fall into this possibility space since none of it was in y.

Below is the intersection of x and y

Below is the intersection of x and not y

The clause(s) are:

The clause(s) are:

Question 4: Simplify

* 1. C