**Northeastern University**

CS 5100  Foundations of Artificial Intelligence

                                        Homework and PA 4

                           NAME: **PARSHVA TIMBADIA**

                     EMAIL: [timbadia.p@northeastern.edu](mailto:timbadia.p@northeastern.edu" \t "https://mail.google.com/mail/u/0/" \l "advanced-search/from=parshwat%40gmail.com&query=parshwat%40gmail.com&isrefinement=true&fromdisplay=parshwat/_blank)

                           NUID: **001091783**

**Q1>**

Refinement: Deliver(Package, Truck)

STEPS: [Navigate(Truck, Current\_Position), Locate(Package, Destination),Route(Truck, Destination\_Position), Control(Truck)]

Refinement:Deliver(Package, Truck)

STEPS:[Reach(Current\_Position, Intermediate\_destination), Reach(Intermediate\_destination, final\_destination)]

Refinement: Control(Truck)

STEPS: [Move(Truck, Forward), Turn(Truck, Left), Turn(Truck, Right)]

**Q2>**

1>

**DURATION** should be **an effect of the action**. Lets say to perform an action you will always need certain amount of time, which is nothing but duration.

**For instance**, the beginning and the ending time of the action can be predicted by the duration, where action can be of fitting all the four wheels in the car.

ACTION(Fitwheel)

EFFECT FITTED(Wheel) ^ DURATION(100)

Let **USE** be **the separate field** as it is used as a resources for the action and not the effect.

**For example,** During wheel fitting we might need to use tools like Tightening-Driver, car-lifter and more.

ACTION(Fitwheel)

USE(Tightening-Driver)

USE(car-lifter)

3>

**CONSUME** will also be **the separate field** as it also makes use of the resources for the action and does not have any effect.

**For instance,** using few nuts out of all the available nuts for fitting the wheels.

ACTION(Fitwheel)

CONSUME(NUTS)

**Q3>**

**A>**

Naive Bayes is a classification problem that uses a **probabilistic model** based on **Bayes Theorem**. Before diving deep into how the algorithm works, I would like to highlight some of the **examples** of this algorithm.

For binary classification, one of the very common examples of this classification algorithm is **Spam Filtering.** We often receive mails that can be authorized or not authorized and this algorithm can help us to classify these mails. For instance, any normal mail may include high occurrence keywords like Dear, friend, lunch, and so one. Whereas the spam mail would have a high occurrence of Money, Loan, and so on. We calculate the probability of the occurrences and whenever new mail arrives we can judge based on the previous results using the Bayes.

Another example of the binary classification is **conversion estimation**. This means that based on the data set the model can help in the prediction that whether a certain type of people will buy a smartphone or not or such similar products.

Naive Bayes is much useful for **large data sets** and works on the principle that the features are **independent** of each other.

The **basic flow of the algorithm** is something as follows:

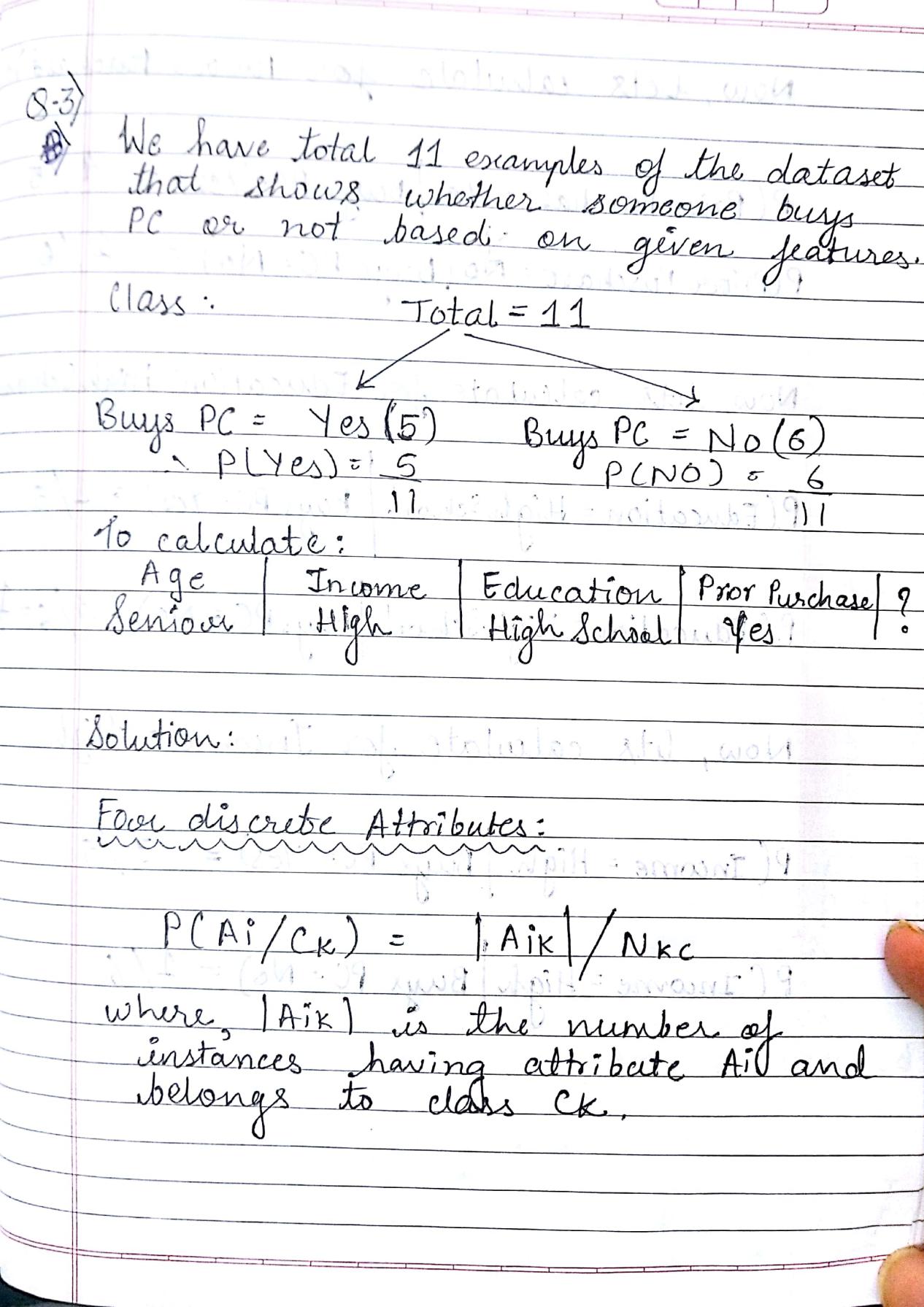
STEP1: Generate a frequency table based on the data provided.

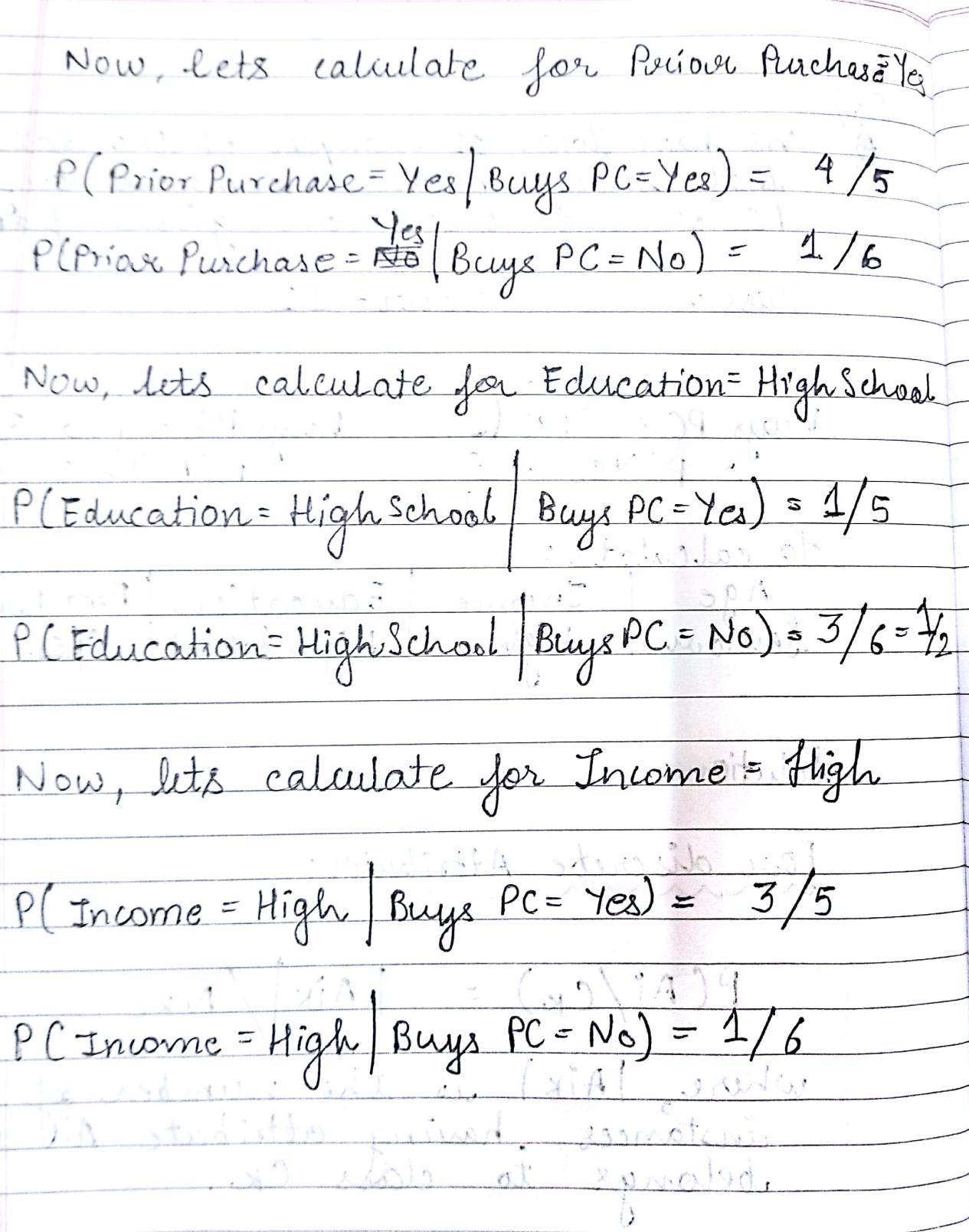
STEP2: Generation of the Likelihood table

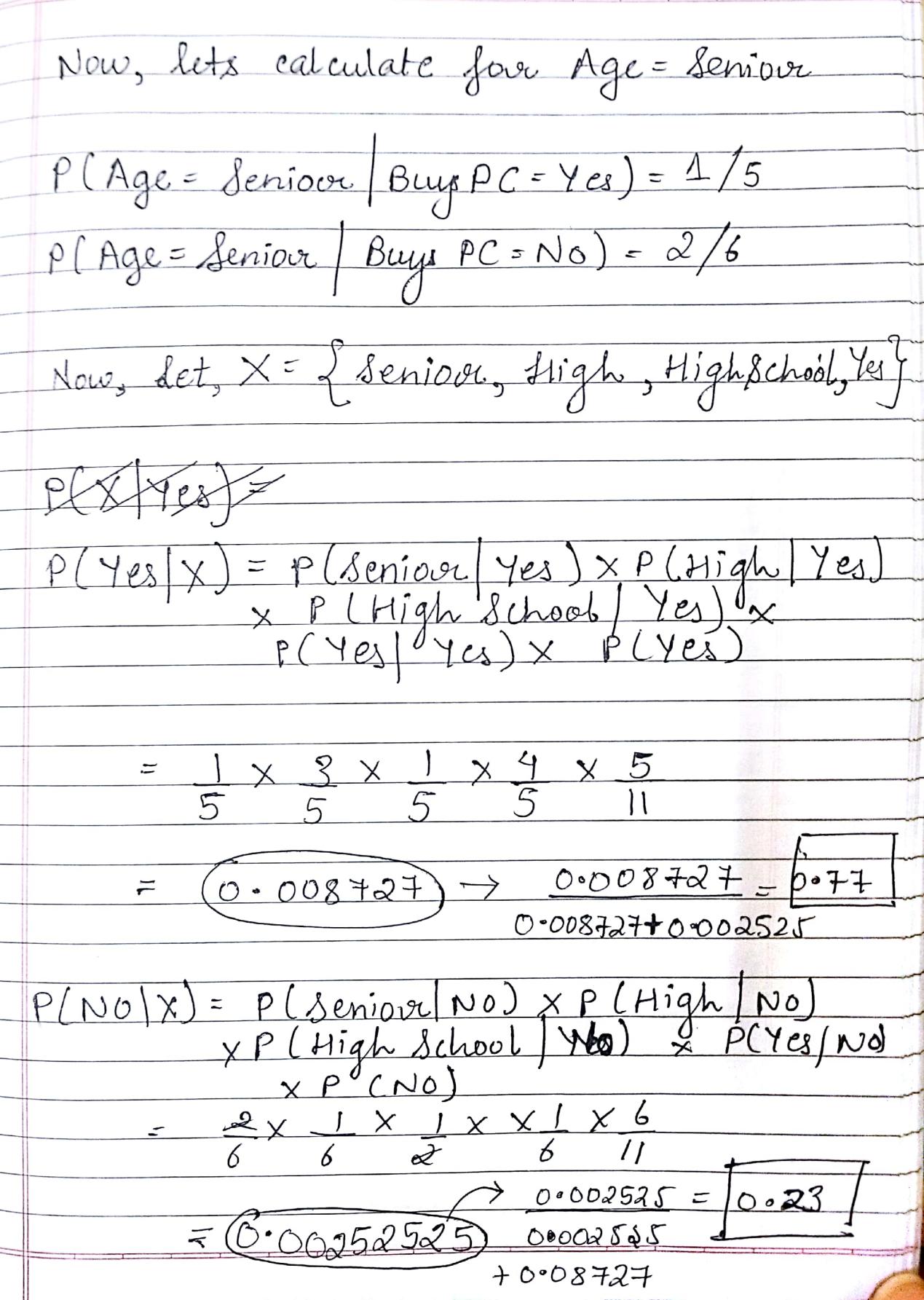
STEP3: For the posterior probability of each class make use of the Bayesian equation.

STEP4: One with the highest posterior probability is the result.

**B>**

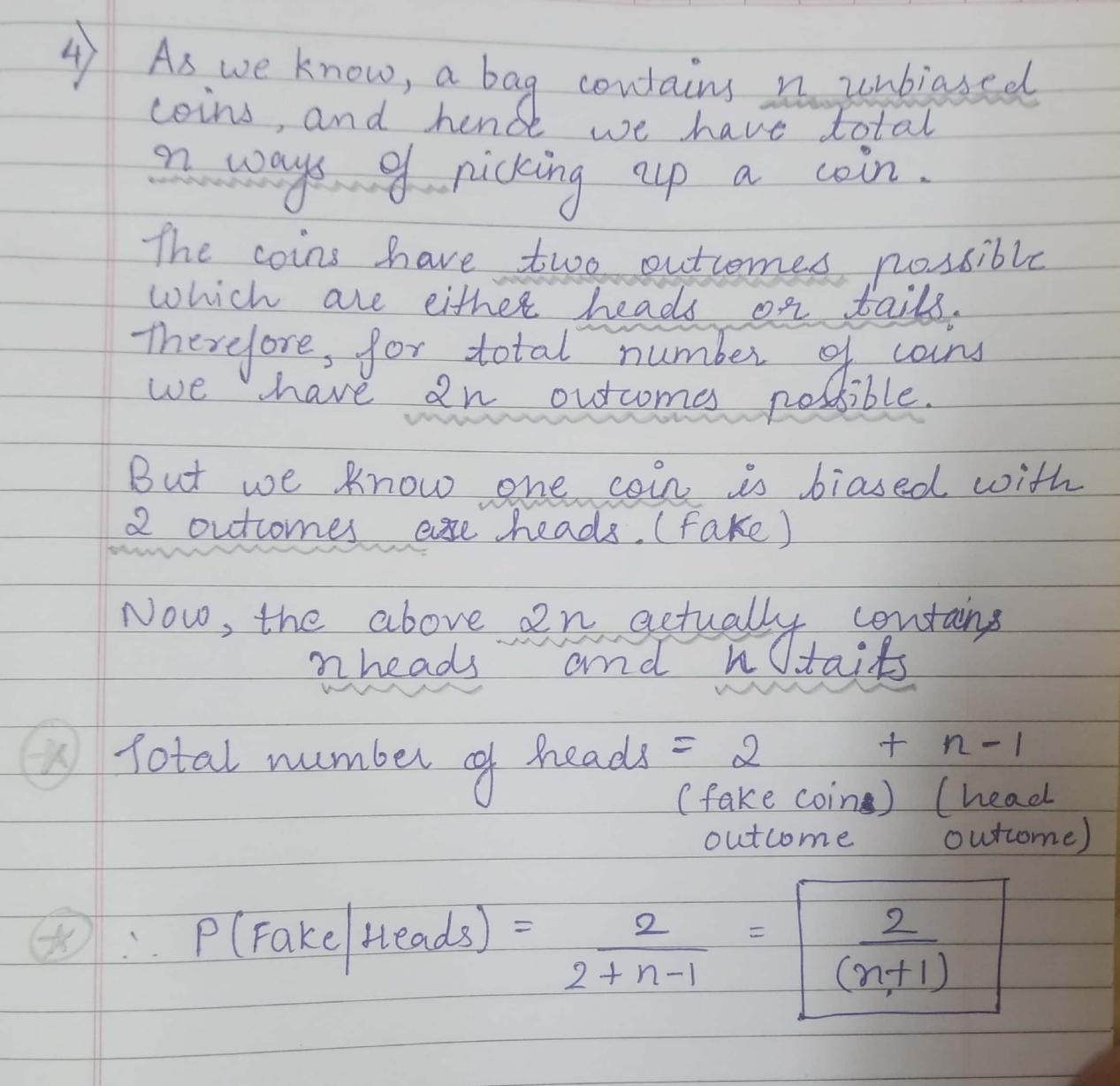






Hence, for the given query the outcome turns out to be **YES**

**Q4>**



**Q5>**

1. Let X be the event of drawing one coin at random from the bag. Also, it is mentioned that the likelihood of drawing each coin from the bag is same.Let the three coins be {a,b,c}.

So the CPT for X will be as follows:

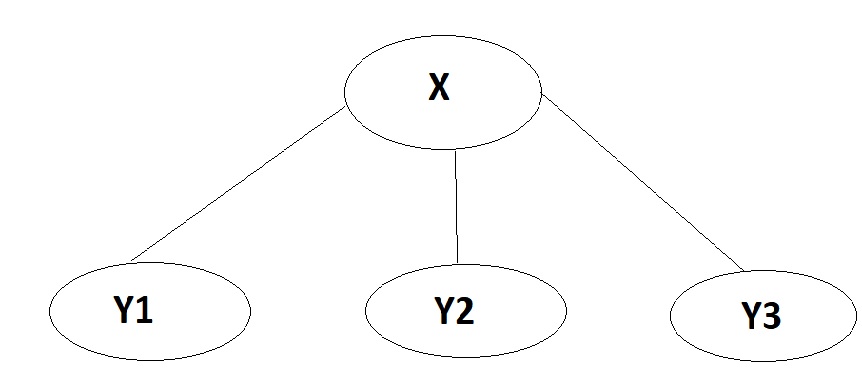
|  |  |
| --- | --- |
| X | P(X) |
| a | 1/3 |
| b | 1/3 |
| c | 1/3 |

Now, we are asked to grab a coin and flip the coin three times. So let the outcome of the coin flipped thrice such that event C(coin has already been picked) occurred be Y1, Y2, Y3.

So the CPT for Yi such that C has occurred will be equal and can be showcased as follows:

|  |  |  |
| --- | --- | --- |
| X | Yi | P(X) |
| a | head | 0.2 |
| b | head | 0.6 |
| c | head | 0.8 |

The representation of the CPT can be shown as:



2>

Now we will calculate the probability of all the coins with the event where the flip turns out to be 2 heads and 1 tails.

We know that the based the Bayesian Rule the conditional probability can be written as follows:

P(X | Y1, Y2, Y3 ) = P(Y1, Y2, Y3 | X) \*P(X) / P(Y1, Y2, Y3)

Above can be written as for our given condition:

P(X | H,H,T ) = P(H,H,T | X) \*P(X) / P(H,H,T)

As we can see the denominator on the RHS will always be a constant so we can remove that from our consideration and now calculate probabilities for {a,b,c}.

When X=a,

P(X=a | H,H,T) = P(H | X=a)\*P(H | X=a)\*P(T | X=a )\*P(X=a)

= 0.2\*0.2\*0.8\*(1/3)

= 0.01067

When X=b,

P(X=b | H,H,T) = P(H | X=b)\*P(H | X=b)\*P(T | X=b )\*P(X=b)

=0.6\*0.6\*0.4\*(1/3)

= 0.048

When X=c,

P(X=c | H,H,T) = P(H | X=c)\*P(H | X=c)\*P(T | X=c )\*P(X=c)

=0.8\*0.8\*0.2\*(1/3)

= 0.04267

Therefore, for the situation the probability of **Coin B is occur is most likely.**

**Q6>**

**A>**

If no evidence is observed then Yes, the burglary and Earthquake are **independent.**

Let us consider the joint probability of both events.

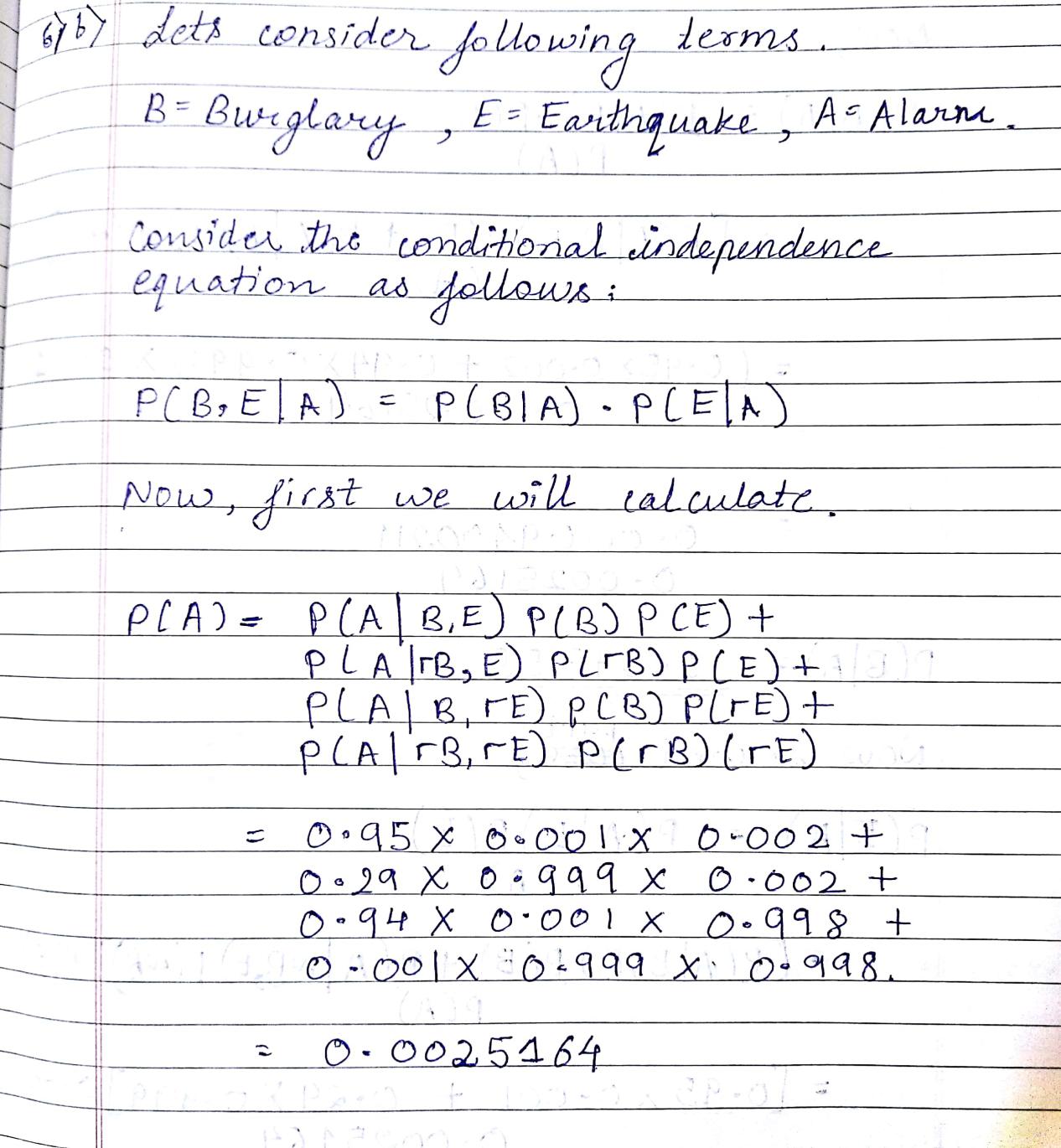
P(Burglary, Earthquake) = P(Burglary | ParentsOf(Burglary))P(Earthquake | ParentsOf(Earthquake))

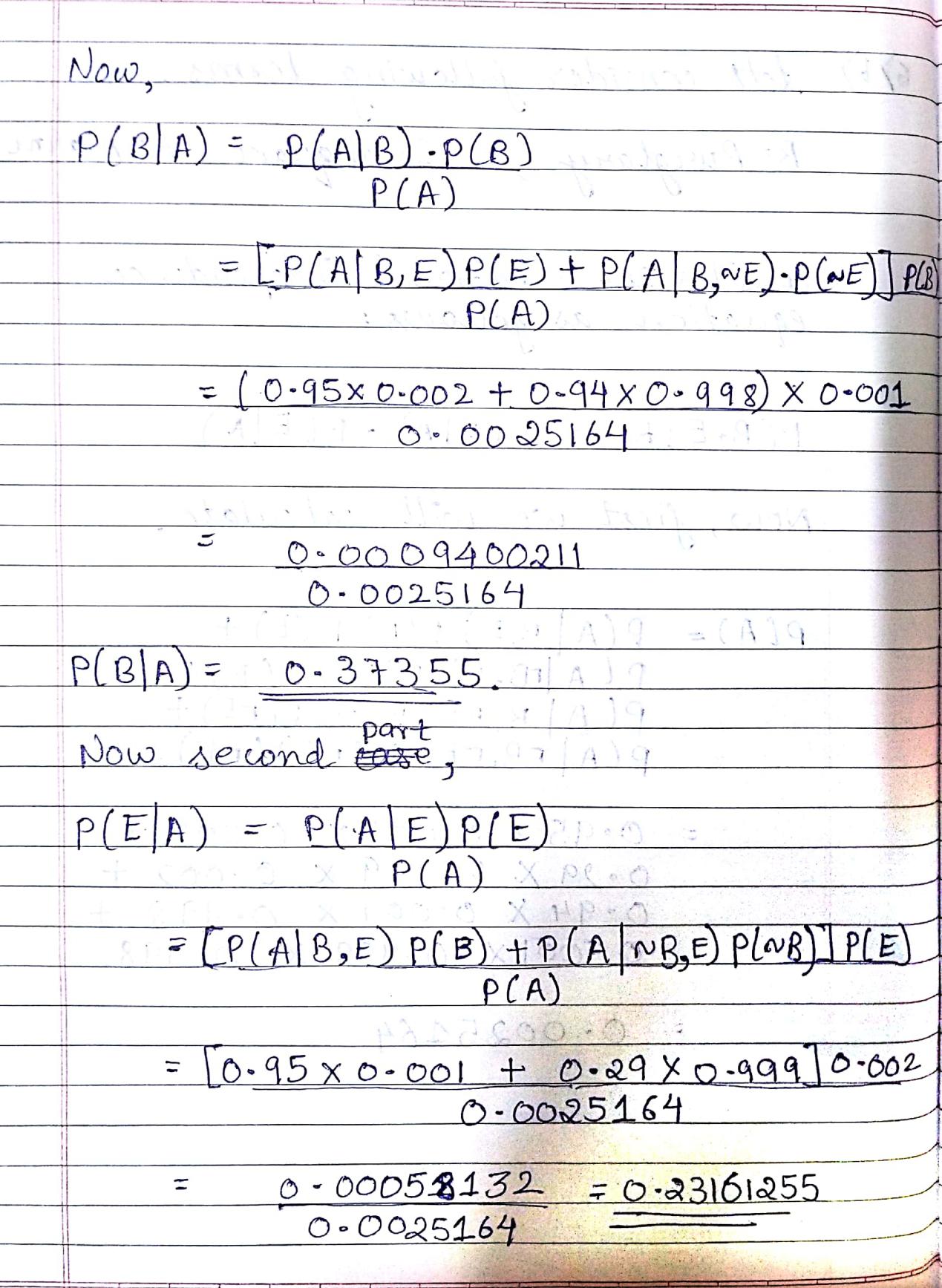
There are no parents that have successor Burglary or the Earthquake and hence we can say the following equation to be correct as far as the proof of by numerical semantics is concerned.

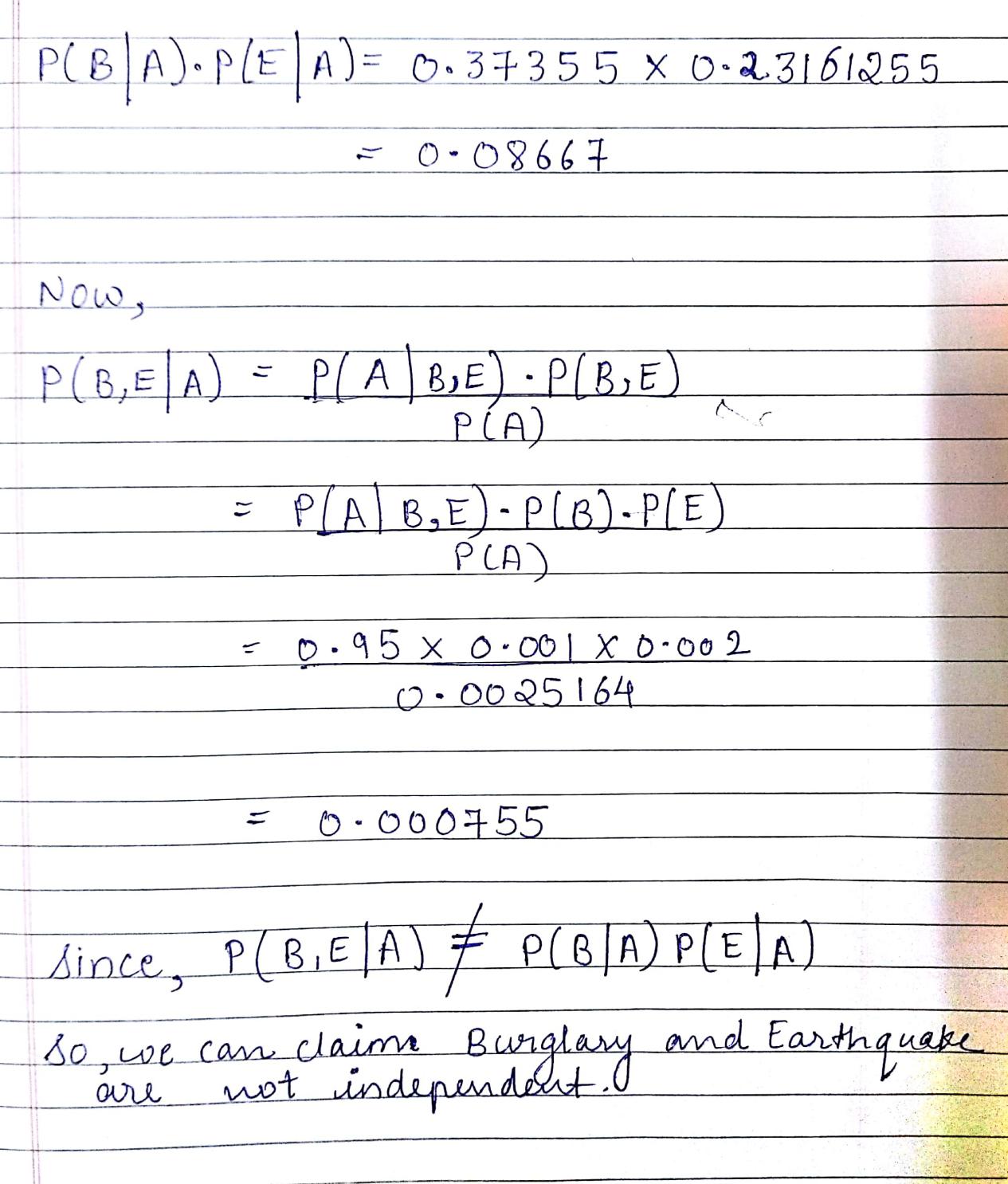
P(Burglary, Earthquake) = P(Burglary )P(Earthquake )

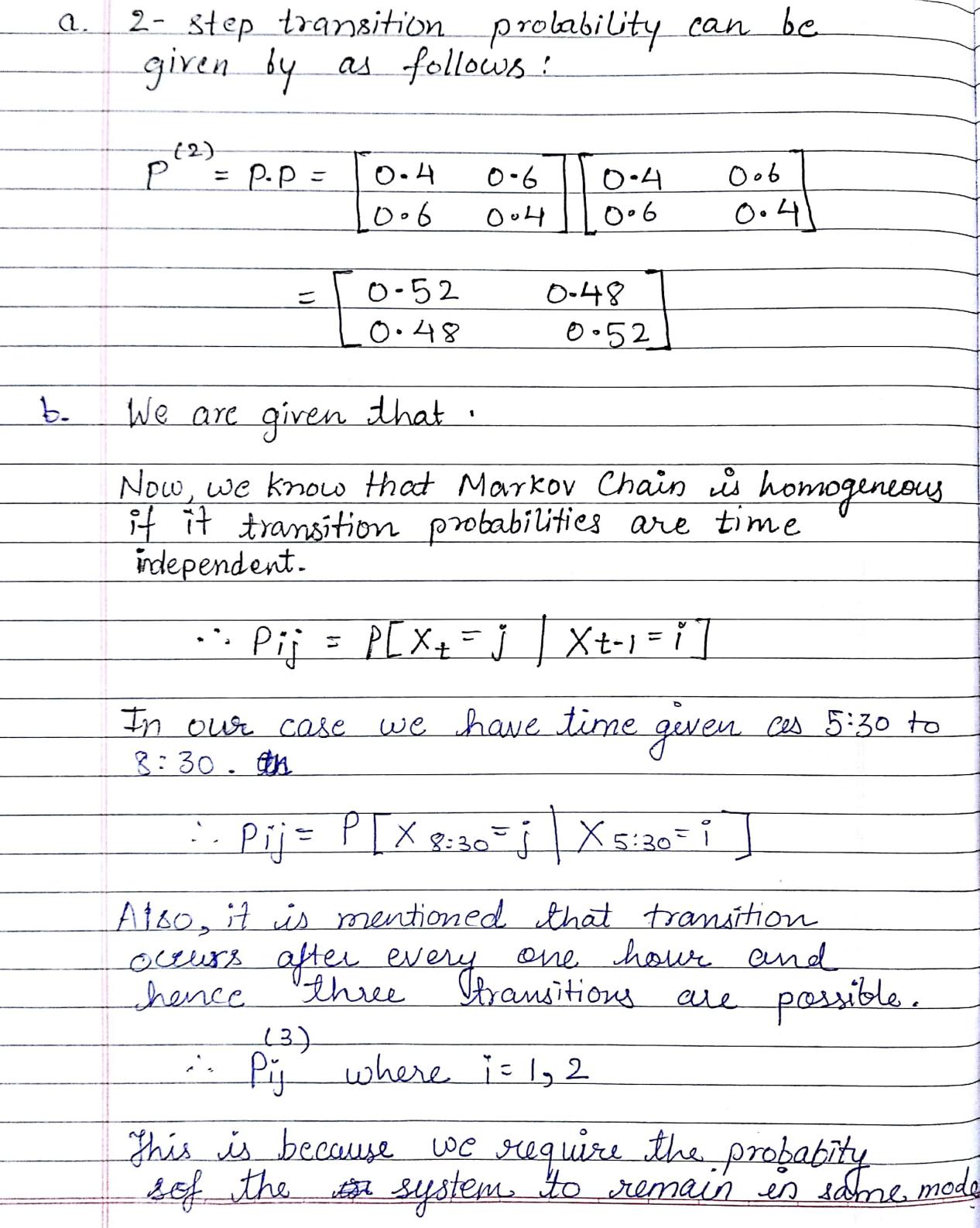
For the **topological semantics,** it is quite clear from the given Bayesian network both Burglary and Earthquake are non-descendants of each other and hence there can be no relationship derived between both. Also, we can say that Burglary and Earthquake are conditionally independent of Alarm in this case.

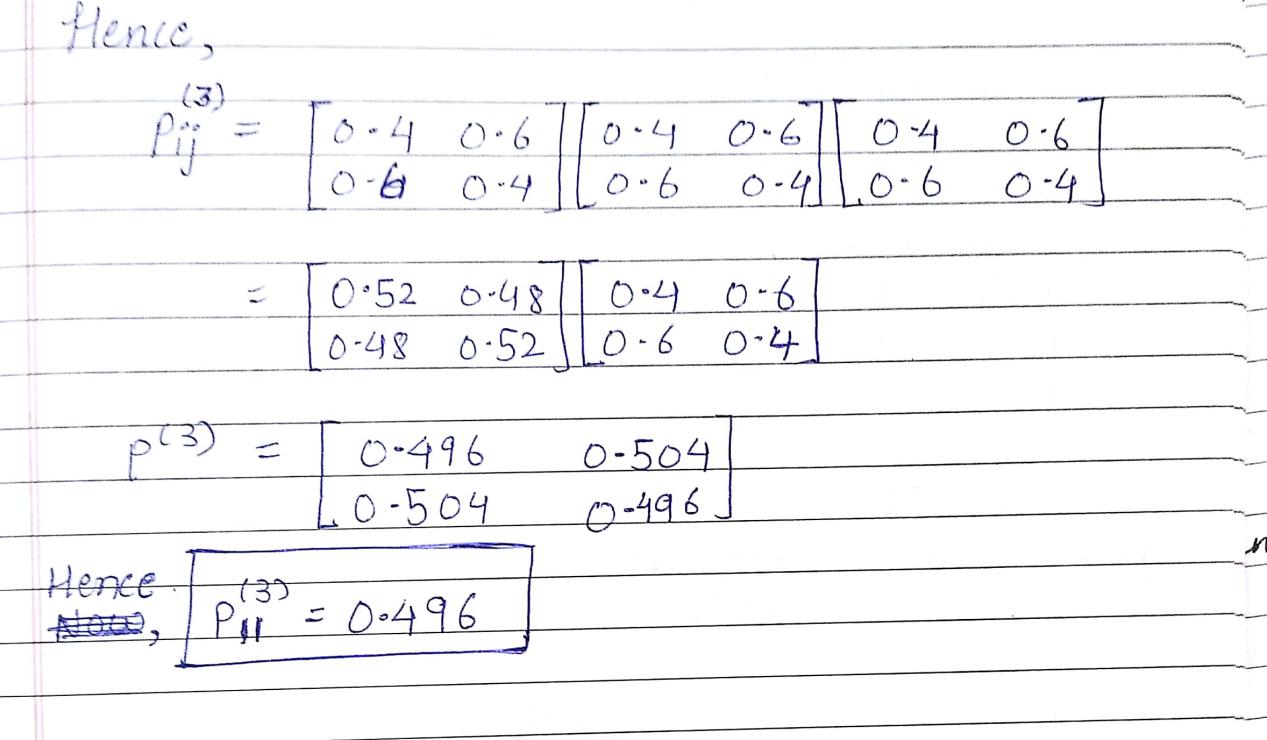
**B.**



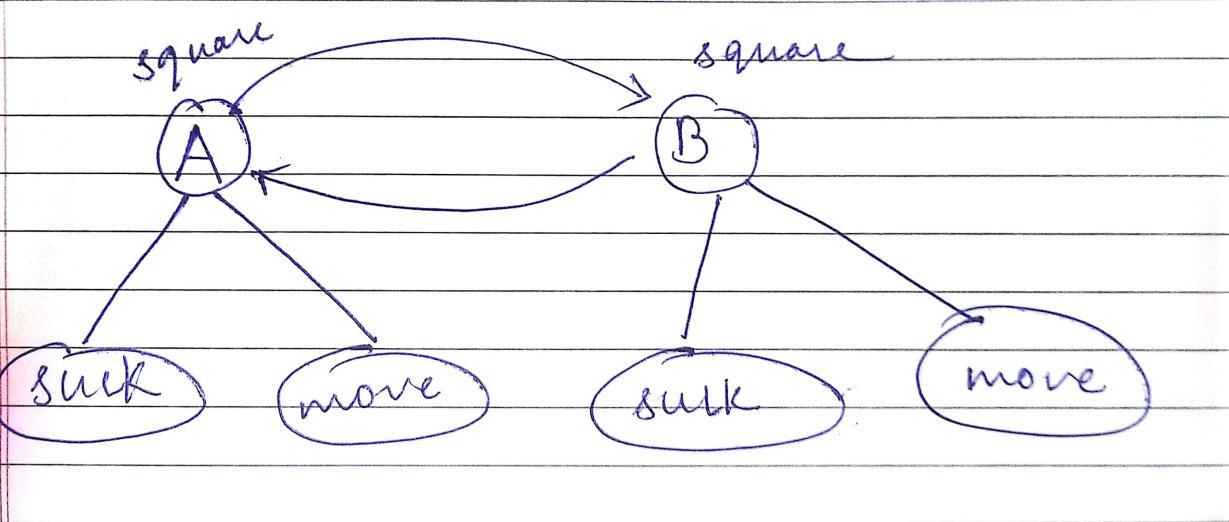




**Q7>** 



Q8>



The robot in the vacuum world doesn’t have any idea of what state is it in, as the states are hidden. The agent can be in any of the state mentioned in the diagram and then make appropriate action for accomplishing the goal. It generally uses multiple sensors like image recognition or maybe obstruction detection to identify what is the current state, after scanning the environment. The goal of the agent in the vacuum world is all the squares cleaned.

In the vacuum cleaner example which we are using has only two squares as per what is mentioned in the question. In the observable state we have suck that will clean the square in case it is found to be dirty after observation and will move in case the square is clean.

HMM are useful here because it maps between the hidden state and observable state and help the agent to estimate the future states with some transition probability.For instance, based on the observation of the current square there is and increase in the chances of having the same state in the next square.