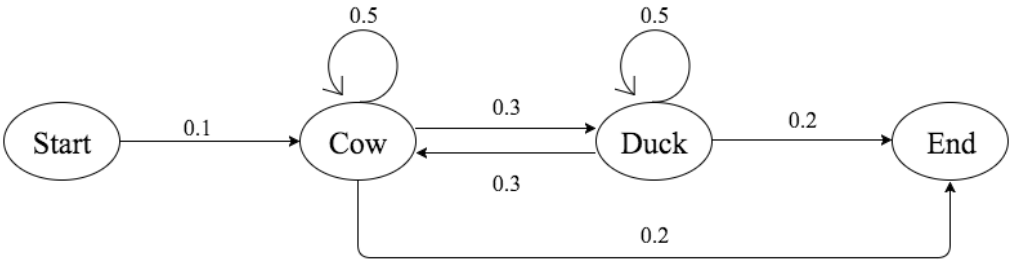
1. (a)



Emission probabilities:

- Cow
  - P (moo | Cow) = 0.9
  - P (hello | Cow) = 0.1
- Duck
  - P (quack | Duck) = 0.6
  - P (hello | Duck) = 0.4

Step1: The first token is "moo"

	0	1	2	3	4
Start	1	0			
Cow	0	0.9 0.9 * 1			
Duck	0	0 * 0			
End	0	0			

Step2: The second token is "hello"

(If "moo" is from Cow state)

	0	1	2	3	4
Start	1	0	0		
Cow	0	0.9 0.9 * 1	0.045 0.9*0.1*0.5		

			0.108
Duck	0	0 * 0	0.9*0.4*0.3
End	0	0	0

(If “moo” is from Duck state)

	0	1	2	3	4
Start	1	0	0		
		0.9	0.045		
Cow	0	0.9 * 1	0*0.1*0.5=0		
			0.108		
Duck	0	0 * 0	0*0.4*0.3=0		
End	0	0	0		

“hello” take maximum, set back pointers,

	0	1	2	3	4
Start	1	0	0		
		0.9	0.045		
Cow	0	0.9 * 1	~~0*0.1*0.5=0~~		
			0.108		
Duck	0	0 * 0	~~0*0.4*0.3=0~~		
End	0	0	0		

Step3: The third token is “quack”

(If “hello” is from Cow state)

	0	1	2	3	4
Start	1	0	0	0	
		0.9	0.045	0	
Cow	0	0.9 * 1	~~0*0.1*0.5=0~~	0.045*0*0.5	
			0.108	0.0081	

Duck	0	0 * 0	~~0*0.4*0.3=0~~	0.045*0.6*0.3
End	0	0	0	

(If "hello" is from Duck state)

	0	1	2	3	4
Start	1	0	0	0	
Cow	0	0.9 * 1	0.045 ~~0*0.1*0.5=0~~	0 0.108*0*0.3=0	
Duck	0	0 * 0	0.108 ~~0*0.4*0.3=0~~	0.0081 0.108*0.6*0.5=0.0324	
End	0	0	0		

"quack" take maximum, set back pointers

	0	1	2	3	4
Start	1	0	0	0	
Cow	0	0.9 * 1	0.045 ~~0*0.1*0.5=0~~	~~0~~ 0.108*0*0.3=0	
Duck	0	0 * 0	0.108 ~~0*0.4*0.3=0~~	~~0.0081~~ 0.108*0.6*0.5=0.0324	
End	0	0	0		

Step4: End state

(If "quack" is from Cow state)

	0	1	2	3	4
Start	1	0	0	0	
Cow	0	0.9	0.045	0	
Duck	0	0 * 0	0.108	0.0324	

End	0	0	0	-	0*0.2 = 0

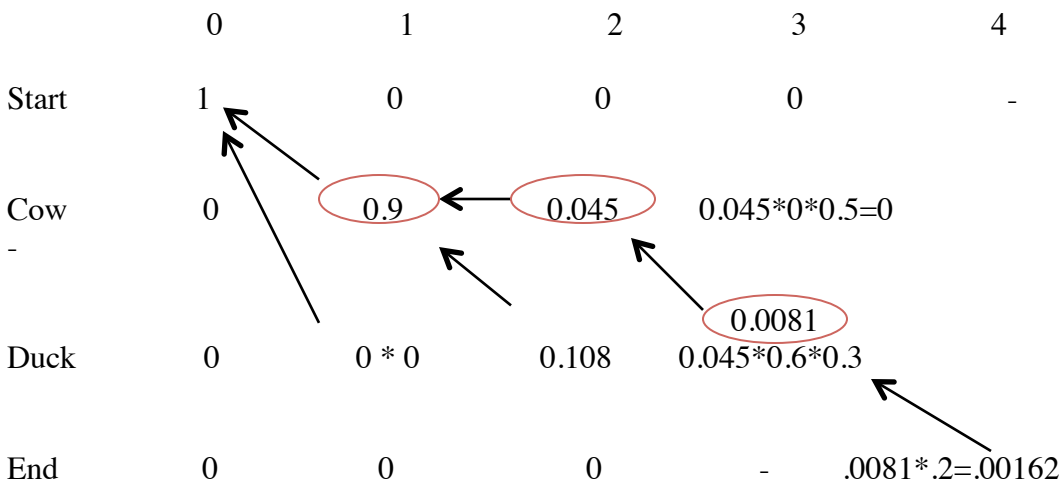
(If "quack" is from Duck state)
End takes maximum, set back pointers

	0	1	2	3	4
Start	1	0	0	0	-
Cow	0	0.9	0.045	0	-
Duck	0	0 * 0	0.108	0.0324	-
End	0	0	0	-	~~0*0.2 = 0~~ .0324*.2=.00648

Decode:
moo = Cow, hello = Duck, quack = Duck
So the probability of emitting this sentence from this state sequence is 0.00648.

	0	1	2	3	4
Start	1	0	0	0	-
Cow	0	0.9	0.045	0	-
Duck	0	0 * 0	0.108	0.0324	-
End	0	0	0	-	0. 00648

(b) There is another state sequence:
moo = Cow, hello =Cow, quack =Duck.
The probability is 0.00162



2. The output for sentence “Wu meets Jane.” (the correct one) and sentence “They have meets every day.” (the incorrect one) is shown below.

```
---------
Sentence:                                Wu meets Jane.
Annotate Wu == <constit cat="NNP">
Annotate meets == <constit cat="VBZ">
Annotate Jane == <constit cat="NNP">
Annotate . == <constit cat=".">
---------
Sentence:                                They have meets every year.
Annotate They == <constit cat="PRP">
Annotate have == <constit cat="VBP">
Annotate meets == <constit cat="VBZ">
Annotate every == <constit cat="DT">
Annotate year == <constit cat="NN">
Annotate . == <constit cat=".">
```

Sentence “They have meets every day.”

Correct situation (“meets” is NNS):
V_1 = P(They | PRP) * P (PRP | Start) * P (have | VBP) * P(VBP |PRP) * P (meets | NNS) * P (NNS | VBP) * P (every | DT) * P (DT | NNS) * P (year | NN) * P (NN | DT)

Incorrect situation (“meets” is VBZ):
V_2 = P (They | PRP) * P (PRP | Start) * P (have | VBP) * P (VBP |PRP) * P (meets | VBZ) * P (VBZ | VBP) * P (every | DT) * P (DT | VBZ) * P (year | NN) * P (NN | DT)

Since most of parts are the same, we can assume that

V_1 = Constant * P (meets | NNS) * P (NNS | VBP) * P (DT | NNS) = (1/ 71914) * (484/14955) * (1233/71914) = (484 * 1233) / (71914^2 * 14955)

V_2 = Constant * P (meets | VBZ) * P (VBZ | VBP) * P (DT | VBZ) = (10/25735) * (39/14955) * (1780/25735) = (390 * 1780) / (25735^2 * 14955)

So, V (“meets” = VBZ) is larger than V(“meets” = NNS). That is the reason why “meets” is VBZ.