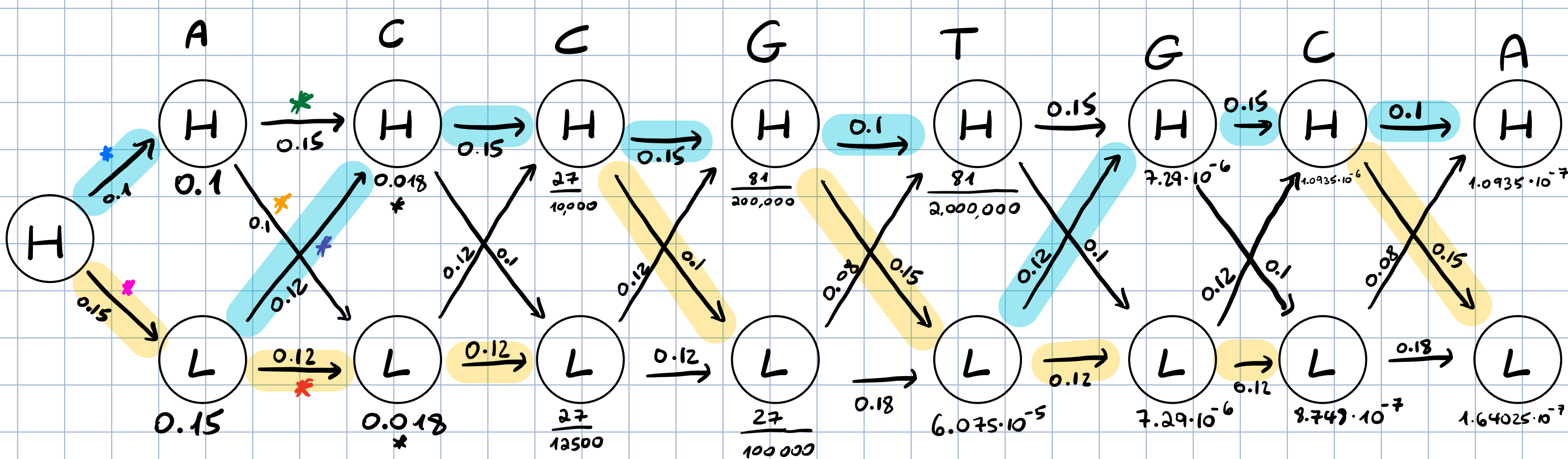
1. (10 pts) Consider this (toy) biological setup:
A cell can be in one of two states - H, for high GC-content, and L for low GC. On each time step the cell produces one nucleotide, A,C,T or G, and might also change its state. The probability of changing from state H to L is 0.5, and from state L to H is 0.4.
In state H the probabilities for producing nucleotides are 0.2 for A, 0.3 for C, 0.3 for G and 0.2 for T. In L the probabilities are 0.3 for A, 0.2 for C, 0.2 for G and 0.3 for T.
Consider the nucleotide sequence $S = ACCGTGCA$. Use the Viterbi algorithm to find the best state-sequence and calculate the probability of S given this state-sequence. Assume the previous state before S was H.

1. נרשום ראשית את הנתונים:

$q(H|L)=0.4 \quad q(L|L)=0.6$

$q(H|H)=0.5 \quad q(L|H)=0.5$

- $e(A|H)=0.2 \quad e(A|L)=0.3$ · $e(G|H)=0.3 \quad e(G|L)=0.2$
- $e(C|H)=0.3 \quad e(C|L)=0.2$ · $e(T|H)=0.2 \quad e(T|L)=0.3$



דוג׳ לחישוב עבור המקרה הראשון:

$*\, q(H|H)e(A|H)=0.1 \qquad *\, q(H|L)e(C|H)\cdot 0.12$

$*\, q(L|H)e(A|L)=0.15 \qquad *\, q(L|H)e(C|L)=0.1$

$*\, q(H|H)e(C|H)=0.15 \qquad *\, q(L|L)e(C|L)=0.12$

סדרת המצבים עם ההסתברות הגבוהה ביותר היא: LHHHLHHL

ההסתברות של S בהינתן רצף המצבים שמצאנו היא:

$$e(A|L)e(C|H)e(C|H)e(G|H)e(T|L)e(G|H)e(C|H)e(A|L)=6.561\cdot 10^{-5}$$

2. (10 pts) In class we saw the trigram HMM model and the corresponding Viterbi algorithm. We will now make two main changes. First, we will consider a four-gram tagger, where p takes the form:

$$p(x_1 \cdots x_n, y_1 \cdots y_{n+1}) = \prod_{i=1}^{n+1} q(y_i | y_{i-3}, y_{i-2}, y_{i-1}) \prod_{i=1}^{n} e(x_i | y_i) \tag{1}$$

We assume in this definition that $y_0 = y_{-1} = y_{-2} = *$, where $*$ is the START symbol, $y_{n+1} = STOP$, and $y_i \in \mathcal{K}$ for $i = 1 \cdots n$, where $\mathcal{K}$ is the set of possible tags in the HMM.
Second, we consider a version of the Viterbi algorithm that takes as input **an integer** n (and not a sentence $x_1 \cdots x_n$ as we saw in class) and finds

$$\max_{y_1 \cdots y_{n+1}, x_1 \cdots x_n} p(x_1 \cdots x_n, y_1 \cdots y_{n+1})$$

for a four-gram tagger, as defined in Equation 1. $x_1 \cdots x_n$ may range over the values of some fixed vocabulary $\mathcal{V}$. Complete the following pseudo-code of this version of the Viterbi algorithm for this model . The pseudo-code must be efficient.

Input: An integer n, parameters $q(w|t, u, v)$ and $e(x|s)$.
Definitions: Define $\mathcal{K}$ to be the set of possible tags. Define $\mathcal{K}_{-2} = \mathcal{K}_{-1} = \mathcal{K}_0 = \{*\}$, and $\mathcal{K}_k = \mathcal{K}$ for $k = 1 \cdots n$. Define $\mathcal{V}$ to be the set of possible words.
Initialization: $\cdots$
Algorithm: $\cdots$
Return: $\cdots$

אתחול: נאתחל טבלה π בגודל $(n+1)\times(|K|+1)\times(|K|+1)\times(|K|+1)$.

נאתחל $\pi(0,*,*,*)=1$. לכל תג v נמצא $\arg\max_{x\in V} e(x|v)=x_v$.

for $k=1,\dots,n$

  for $w\in K_{k-2}$ $u\in K_{k-1}$ $v\in K_k$

$$\pi(k,w,u,v)=\max_{z\in K_{k-3}}\left(\pi(k-1,z,w,u)\times q(v|z,w,u)\times e(x_v|v)\right)$$

Return $\displaystyle\max_{\substack{w\in K_{k-2}\\ u\in K_{k-1}\\ v\in K_k}}\left(\pi(n,w,u,v)\times q(STOP|w,u,v)\right)$

באלגוריתם עבור כל תג אנו עוברים על שלשת התגים הקיימים לפניו, עם המילה שמצאנו שממקסמת את $e(x|v)$.

באתחול אנו בונים טבלה בגודל $O(n|K|^3)$ - $O(n|K|^3)$
ועבור כל תא אנו מוצאים את המילה מקבוצת המילים V
שממקסמת את $e(x|v)$ - $O(|K||V|)$.
נשים לב כי גודל הטבלה הוא $O(n|K|^3)$ ומילוי של תא בטבלה
ייקח $O(|K|)$ משום שאנו מסתכלים על כלל הוורציות של כל התגים
- $O(|K|^4 \cdot n)$.
בהחזרה אנו עוברים על $O(|K|^3)$ תאים.
סה"כ: זמן הריצה של האלגוריתם הינו $O(|K|^4 \cdot n)$.

תשובות חלק פרקטי:

b ii)

```
Error rate for known words is 0.07044
Error rate for unknown words is 0.74346
Total error rate is 0.14731
```

c iii)

```
Error rate for known words viterbi is 0.19838
Error rate for unknown words viterbi is 0.74346
Total error rate viterbi is 0.26064
```

d ii)

```
Error rate for known words viterbi add one is 0.16699
Error rate for unknown words viterbi add one is 0.73386
Total error rate viterbi add one is 0.23174
```

e ii)

```
Error rate for known words viterbi pseudo is 0.21738
Error rate for unknown words viterbi pseudo is 0
Total error rate viterbi pseudo is 0.21738
```

e iii)

```
Error rate for known words viterbi add one pseudo is 0.20941
Error rate for unknown words viterbi add one pseudo is 0
Total error rate viterbi add one pseudo is 0.20941
```

	START	AT	NP	NN	JJ	VBD	...	DTX	BEM	WP	UH	QLP	
START	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
AT	0.0	856.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
NP	0.0	8.0	257.0	248.0	9.0	1.0	...	0.0	0.0	0.0	0.0	0.0	0.0
NN	0.0	18.0	27.0	1377.0	27.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
JJ	0.0	12.0	8.0	180.0	321.0	2.0	...	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
BEM	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
WP	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
UH	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
QLP	0.0	0.0	0.0	4.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0

מהתבוננות מעמיקה במטריצה שמנו לב כי השגיאות הנפוצות ביותר שלנו קורות עבור מילים שחזינו להן תיוג "MM" על אף שבהן אינו תיוגן המקורי.