Edit distance dynamic programming algorithm

- Given two strings S_1 and S_2 of length m and n respectively
- Let $F(i, j)$ be the fewest edits mapping $S_1[1, i]$ to $S_2[1, j]$
- Let $F(0, j) = j$ and $F(i, 0) = i$ for all i, j
- Let $M[x, y]$ be the cost of mapping from symbol x to symbol y

$$M[x, y] = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{otherwise} \end{cases}$$

- Then

$$F(i, j) = \min \left\{ \begin{array}{l} F(i, j-1) + 1, \\ F(i-1, j) + 1, \\ F(i-1, j-1) + M[S_1(i), S_2(j)] \end{array} \right\}$$

Tabular representation: ‘perambulate’ $\rightarrow$ ‘preamble’

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow} \; {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0									
p	1									
e	2									
r	3									
a	4									
m	5									
b	6									
u	7									
l	8									
a	9									
t	10									
e	11									

Initialize zero positions

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ _{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1								
e	2	2								
r	3	3								
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

Fill cell, $i = 1$, $j = 1$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	$\searrow\downarrow$ $\rightarrow\cdot$							
e	2	2								
r	3	3								
a	4	4								
m	5	5								
b	6	6								
u	7	7				
l	8	8				
a	9	9				
t	10	10				
e	11	11				

$$F(1,1) \; = \; \min \left\{ \begin{array}{l} F(1,0)+1, \\ F(0,1)+1, \\ F(0,0)+M[p,p] \end{array} \right\}$$

Fill cell, $i = 2$, $j = 1$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0							
e	2	2	$\overset{\searrow\,\downarrow}{\rightarrow\cdot}$							
r	3	3								
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(2,1) \;=\; \min \left\{ \begin{array}{l} F(2,0)+1, \\ F(1,1)+1, \\ F(1,0)+M[e,p] \end{array} \right\}$$

Fill cell, $i = 1$, $j = 2$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	$\searrow\downarrow$ $\rightarrow\cdot$						
e	2	2	1							
r	3	3								
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(1,2) \;=\; \min \left\{ \begin{array}{l} F(1,1)+1, \\ F(0,2)+1, \\ F(0,1)+M[p,r] \end{array} \right\}$$

Fill cell, $i = 2$, $j = 2$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1						
e	2	2	1	$\overset{\searrow\,\downarrow}{\rightarrow\cdot}$						
r	3	3								
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(2,2) \;=\; \min\left\{\begin{array}{l} F(2,1)+1, \\ F(1,2)+1, \\ F(1,1)+M[e,r] \end{array}\right\}$$

Fill cell, $i = 3$, $j = 1$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ j $\rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1						
e	2	2	1	1						
r	3	3	$\searrow\downarrow$ $\rightarrow\cdot$							
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(3,1) = \min \left\{ \begin{array}{l} F(3,0) + 1, \\ F(2,1) + 1, \\ F(2,0) + M[r,p] \end{array} \right\}$$

Fill cell, $i = 3$, $j = 2$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $_j$ $\rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1						
e	2	2	1	1						
r	3	3	2	$\searrow\downarrow$ $\rightarrow\cdot$						
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(3,2) = \min \left\{ \begin{array}{l} F(3,1)+1, \\ F(2,2)+1, \\ F(2,1)+M[r,r] \end{array} \right\}$$

Fill cell, $i = 1$, $j = 3$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	$\searrow\downarrow$ $\rightarrow\cdot$					
e	2	2	1	1						
r	3	3	2	1						
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(1,3) \;=\; \min\left\{\begin{array}{l} F(1,2)+1, \\ F(0,3)+1, \\ F(0,2)+M[p,e] \end{array}\right\}$$

Fill cell, $i = 2$, $j = 3$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2					
e	2	2	1	1	$\begin{smallmatrix}\searrow & \downarrow\\ \rightarrow & \cdot\end{smallmatrix}$					
r	3	3	2	1						
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(2,3) = \min\left\{\begin{array}{l} F(2,2)+1,\\ F(1,3)+1,\\ F(1,2)+M[e,e] \end{array}\right\}$$

Fill cell, $i = 3$, $j = 3$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $_j$ $\rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2					
e	2	2	1	1	1					
r	3	3	2	1	↘ ↓ → •					
a	4	4								
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

$$F(3,3) \;=\; \min \left\{ \begin{array}{l} F(3,2)+1, \\ F(2,3)+1, \\ F(2,2)+M[r,e] \end{array} \right\}$$

Fill cell, $i = 4$, $j = 4$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ _{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3				
e	2	2	1	1	1	2				
r	3	3	2	1	2	2				
a	4	4	3	2	2	$\begin{matrix}\searrow & \downarrow \\ \rightarrow & \cdot\end{matrix}$				
m	5	5								
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

Fill cell, $i = 5$, $j = 5$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4			
e	2	2	1	1	1	2	3			
r	3	3	2	1	2	2	3			
a	4	4	3	2	2	2	3			
m	5	5	4	3	3	3	$\overset{\searrow\downarrow}{\rightarrow\bullet}$			
b	6	6								
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

Fill cell, $i = 6$, $j = 6$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5		
e	2	2	1	1	1	2	3	4		
r	3	3	2	1	2	2	3	4		
a	4	4	3	2	2	2	3	4		
m	5	5	4	3	3	3	2	3		
b	6	6	5	4	4	4	3	↘↓ →•		
u	7	7								
l	8	8								
a	9	9								
t	10	10								
e	11	11								

Fill cell, $i = 7$, $j = 7$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $_j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5	6	
e	2	2	1	1	1	2	3	4	5	
r	3	3	2	1	2	2	3	4	5	
a	4	4	3	2	2	2	3	4	5	
m	5	5	4	3	3	3	2	3	4	
b	6	6	5	4	4	4	3	2	3	
u	7	7	6	5	5	5	4	3	$\overset{\searrow\downarrow}{\rightarrow\cdot}$	
l	8	8								
a	9	9								
t	10	10								
e	11	11								

Fill cell, $i = 8$, $j = 8$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $_j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5	6	7
e	2	2	1	1	1	2	3	4	5	6
r	3	3	2	1	2	2	3	4	5	6
a	4	4	3	2	2	2	3	4	5	6
m	5	5	4	3	3	3	2	3	4	5
b	6	6	5	4	4	4	3	2	3	4
u	7	7	6	5	5	5	4	3	3	4
l	8	8	7	6	6	6	5	4	3	$\searrow\downarrow$ $\rightarrow\cdot$
a	9	9								
t	10	10								
e	11	11								

Fill cell, $i = 9$, $j = 8$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5	6	7
e	2	2	1	1	1	2	3	4	5	6
r	3	3	2	1	2	2	3	4	5	6
a	4	4	3	2	2	2	3	4	5	6
m	5	5	4	3	3	3	2	3	4	5
b	6	6	5	4	4	4	3	2	3	4
u	7	7	6	5	5	5	4	3	3	4
l	8	8	7	6	6	6	5	4	3	4
a	9	9	8	7	7	6	6	5	4	$\searrow\downarrow$ $\rightarrow\bullet$
t	10	10								
e	11	11								

Fill cell, $i = 10$, $j = 8$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ _{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5	6	7
e	2	2	1	1	1	2	3	4	5	6
r	3	3	2	1	2	2	3	4	5	6
a	4	4	3	2	2	2	3	4	5	6
m	5	5	4	3	3	3	2	3	4	5
b	6	6	5	4	4	4	3	2	3	4
u	7	7	6	5	5	5	4	3	3	4
l	8	8	7	6	6	6	5	4	3	4
a	9	9	8	7	7	6	6	5	4	4
t	10	10	9	8	8	7	7	6	5	$\overset{\searrow\ \downarrow}{\rightarrow\cdot}$
e	11	11								

Fill cell, $i = 11$, $j = 8$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ $_j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5	6	7
e	2	2	1	1	1	2	3	4	5	6
r	3	3	2	1	2	2	3	4	5	6
a	4	4	3	2	2	2	3	4	5	6
m	5	5	4	3	3	3	2	3	4	5
b	6	6	5	4	4	4	3	2	3	4
u	7	7	6	5	5	5	4	3	3	4
l	8	8	7	6	6	6	5	4	3	4
a	9	9	8	7	7	6	6	5	4	4
t	10	10	9	8	8	7	7	6	5	5
e	11	11	10	9	8	8	8	7	6	$\searrow\downarrow$ $\rightarrow\cdot$

Minimal edit distance: cell $i = 11$, $j = 8$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ _{j} \rightarrow$	0	1	2	3	4	5	6	7	8
	0	0	1	2	3	4	5	6	7	8
p	1	1	0	1	2	3	4	5	6	7
e	2	2	1	1	1	2	3	4	5	6
r	3	3	2	1	2	2	3	4	5	6
a	4	4	3	2	2	2	3	4	5	6
m	5	5	4	3	3	3	2	3	4	5
b	6	6	5	4	4	4	3	2	3	4
u	7	7	6	5	5	5	4	3	3	4
l	8	8	7	6	6	6	5	4	3	4
a	9	9	8	7	7	6	6	5	4	4
t	10	10	9	8	8	7	7	6	5	5
e	11	11	10	9	8	8	8	7	6	**5**

Find the optimal alignment

- Now we know that the lowest cost of aligning ‘perambulate’ to ‘preamble’ is 5
  - This is called the Levenshtein distance
- Just knowing this cost might be useful in some cases
- But in general, we want to know *which* edits led to the optimal alignment
- Thus, backtrace to find the path(s) corresponding to the score in bottom-right cell ($\boldsymbol{i = 11, j = 8}$)
  - (Why might we have more than one optimal path?)

Find path(s) corresponding to score in $i = 11$, $j = 8$

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}\ {}_j \rightarrow$	0	1	2	3	4	5	6	7	8
	0	**0**	1	2	3	4	5	6	7	8
p	1	1	**0**	**1**	2	3	4	5	6	7
e	2	2	**1**	**1**	**1**	2	3	4	5	6
r	3	3	2	**1**	**2**	2	3	4	5	6
a	4	4	3	2	2	**2**	3	4	5	6
m	5	5	4	3	3	3	**2**	3	4	5
b	6	6	5	4	4	4	3	**2**	3	4
u	7	7	6	5	5	5	4	**3**	3	4
l	8	8	7	6	6	6	5	4	**3**	4
a	9	9	8	7	7	6	6	5	**4**	4
t	10	10	9	8	8	7	7	6	**5**	5
e	11	11	10	9	8	8	8	7	6	**5**

Backtrace

- Can find the path(s) corresponding to final score in $O(n+m)$
- While filling in the matrix, keep a backpointer $B(i,j)$ for each cell such that

$$B(i,j) = \text{argmin} \left\{ \begin{array}{l} F(i, j-1) + 1, \\ F(i-1, j) + 1, \\ F(i-1, j-1) + M[S_1(i), S_2(j)] \end{array} \right\}$$

  - On a match/substitution, $B(i,j)$ will point to cell $(i-1, j-1)$
  - On an insertion, $B(i,j)$ will point to cell $(i, j-1)$
  - On a deletion, $B(i,j)$ will point to cell $(i-1, j)$
  - On a tie, $B(i,j)$ may point to multiple cells

Backpointers along optimal path(s)

			p	r	e	a	m	b	l	e
	$\overset{i}{\downarrow}$ j →	0	1	2	3	4	5	6	7	8
	0	↖								
p	1		↖	←						
e	2		↑	↖	↖					
r	3			↖	↖ ↑ ←					
a	4					↖				
m	5						↖			
b	6							↖		
u	7							↑		
l	8								↖	
a	9								↑	
t	10								↑	
e	11									↖

Paths correspond to alignments

- Three different alignments result in edit distance of 5:

1.

p	r	e	a	m	b	-	l	-	-	e
p	e	r	a	m	b	u	l	a	t	e

2.

p	-	r	e	a	m	b	-	l	-	-	e
p	e	r	-	a	m	b	u	l	a	t	e

3.

p	r	e	-	a	m	b	-	l	-	-	e
p	-	e	r	a	m	b	u	l	a	t	e

- Can choose to slightly skew costs to avoid such ambiguities
  - e.g., score substitutions at cost 0.99

Substitution models

- For natural language sequences, typically looking for full approximate matches (e.g., spell checking)
- For protein and DNA/RNA sequences, more often looking to match subsequences (e.g., for similarity across species)
- Need some way to find “likely” related subsequences, i.e., approximate matches that probably didn’t arise by chance
  - Build “random” model, whereby two sequences are modeled independently
  - Build joint model, whereby two sequences are modeled together
  - Compare likelihoods via log likelihood or log odds ratio
- This is a principled way to capture the fact that particular symbols tend to substitute for each other
  - i.e., are evolutionarily related

Substitution likelihood

- Let $q(a)$ be the probability of observing symbol a
- Let $p(ab)$ be the probability that symbols a and b are substituted
- Then, for a given ungapped alignment between S_1 and S_2, the *odds ratio* between the joint and random models is

$$\text{odds}(S_1, S_2) = \frac{\prod_i p(S_1(i)S_2(i))}{\prod_i q(S_1(i)) \prod_i q(S_2(i))} = \prod_i \frac{p(S_1(i)S_2(i))}{q(S_1(i))q(S_2(i))}$$

- Taking the log, we get

$$\text{log-odds}(S_1, S_2) = \sum_i L[S_1(i), S_2(i)]$$

where $$L[a, b] = \log p(ab) - \log q(a) - \log q(b)$$

- $L[a, b]$ will be positive for symbols with high probability of substitution
- Note that we now switch from min to max for dynamic programming

Substitution likelihood

- Let $q(a)$ be the probability of observing symbol a
- Let $p(ab)$ be the probability that symbols a and b are substituted
- Then, for a given ungapped alignment between S_1 and S_2, the *odds ratio* between the joint and random models is

$$\text{odds}(S_1, S_2) = \frac{\prod_i p(S_1(i)S_2(i))}{\prod_i q(S_1(i)) \prod_i q(S_2(i))} = \prod_i \frac{p(S_1(i)S_2(i))}{q(S_1(i))q(S_2(i))}$$

- Taking the log, we get

$$\text{log-odds}(S_1, S_2) = \sum_i L[S_1(i), S_2(i)]$$

$$\text{where} \qquad L[a, b] = \log p(ab) - \log q(a) - \log q(b)$$

- $L[a, b]$ will be positive for symbols with high probability of substitution
- Note that we now switch from min to max for dynamic programming

PAM and Blosum matrices

- PAM (Point Accepted Mutation) amino acid substitution matrices
  - Developed by M. Dayhoff from explicit models of evolution
- PAM1 matrix estimates expected substitution rates if 1% of the amino acids had changed
- Can calculate expected rates over longer durations by taking $\boldsymbol{M^k}$
- Most widely used is PAM250, scaled by $\boldsymbol{\frac{3}{\log 2}}$
- BLOSUM (block substitution matrix) are preferred for evolutionarily divergent sequences
  - Repeated small changes poorly estimates large differences

PAM250 substitution matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V	7	4	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	72	4	17

Blosum50 substitution matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5

Gap penalties

- Not just substitution to consider – also insertion and deletion
- These are penalized as “gaps” of a certain length g
- Linear gap penalties give the same cost d to every single symbol gap
  - Thus, the penalty for a gap of length g is $\gamma(g) = -gd$
- Also, commonly, an “affine” gap penalty is used
  - A penalty for starting a gap d
  - Another penalty for continuing an already started gap e
  - Thus, the penalty for a gap of length g is $\gamma(g) = -d - (g - 1)e$
- For affine gap penalties, need to keep track of whether gap is started or not
  - slightly different dynamic programming (stay tuned . . .)

Protein sequence alignment

- Will use example from Durbin et al., section 2.3
  - Strings S_1 = 'HEAGAWGHEE' and S_2 = 'PAWHEAE'
  - Use BLOSUM50 substitution matrix
  - Linear gap penalty with $d = 8$
- Let $F(0, j) = -jd$ and $F(i, 0) = -id$ for all i, j
- Alignment scores are calculated

$$F(i, j) = \max \left\{ \begin{array}{l} F(i, j-1) - d, \\ F(i-1, j) - d, \\ F(i-1, j-1) + M[S_1(i), S_2(j)] \end{array} \right\}$$

Initialize zero positions

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}$ $_j$ $\rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8							
E	2	-16							
A	3	-24							
G	4	-32							
A	5	-40							
W	6	-48							
G	7	-56							
H	8	-64							
E	9	-72							
E	10	-80							

Fill cell, $i = 1$, $j = 1$

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ {}_j \rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8	$\searrow\downarrow$ $\rightarrow\cdot$						
E	2	-16							
A	3	-24							
G	4	-32							
A	5	-40							
W	6	-48							
G	7	-56							
H	8	-64							
E	9	-72							
E	10	-80							

$$\boldsymbol{F(1,1) = \max \left\{ \begin{array}{l} F(1,0) - 8, \\ F(0,1) - 8, \\ F(0,0) + M[H,P] \end{array} \right\}}$$

$$\boldsymbol{M[H,P] = -2}$$

Fill cell, $i = 2$, $j = 2$

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}$ j $\rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8	-2	-10					
E	2	-16	-9	$\overset{\searrow\downarrow}{\rightarrow\cdot}$					
A	3	-24							
G	4	-32							
A	5	-40							
W	6	-48							
G	7	-56							
H	8	-64							
E	9	-72							
E	10	-80							

$$F(2,2) = \max\left\{\begin{array}{l} F(2,1)-8, \\ F(1,2)-8, \\ F(1,1)+M[E,A] \end{array}\right\}$$

$$M[E,A] = -1$$

(skip to interesting bits) **Fill cell, $i = 5$, $j = 2$**

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}$ j $\rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8	-2	-10					
E	2	-16	-9	-3					
A	3	-24	-17	-4					
G	4	-32	-25	-12					
A	5	-40	-33	↘ ↓ → ·					
W	6	-48							
G	7	-56							
H	8	-64							
E	9	-72							
E	10	-80							

$$F(5,2) \; = \; \max \left\{ \begin{array}{l} F(5,1) - 8, \\ F(4,2) - 8, \\ F(4,1) + M[A,A] \end{array} \right\}$$

$$M[A,A] \; = \; 5$$

Fill cell, $i = 6$, $j = 3$

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}$ $j \rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8	-2	-10	-18				
E	2	-16	-9	-3	-11				
A	3	-24	-17	-4	-6				
G	4	-32	-25	-12	-7				
A	5	-40	-33	-20	-15				
W	6	-48	-41	-28	$\searrow\downarrow$ $\rightarrow\cdot$				
G	7	-56							
H	8	-64							
E	9	-72							
E	10	-80							

$$F(6,3) \;=\; \max\left\{\begin{array}{l} F(6,2) - 8, \\ F(5,3) - 8, \\ F(5,2) + M[W,W] \end{array}\right\}$$

$$M[W,W] \;=\; 15$$

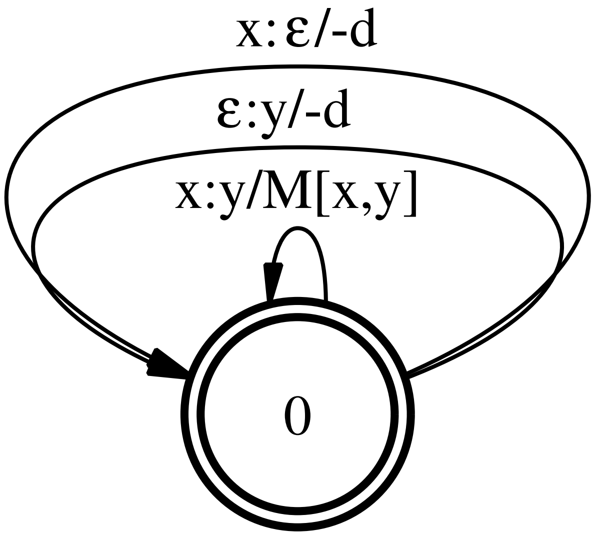
Fill cell, $i = 9$, $j = 5$

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8	-2	-10	-18	-14	-22		
E	2	-16	-9	-3	-11	-18	-8		
A	3	-24	-17	-4	-6	-13	-16		
G	4	-32	-25	-12	-7	-8	-16		
A	5	-40	-33	-20	-15	-9	-9		
W	6	-48	-41	-28	-5	-13	-12		
G	7	-56	-49	-36	-13	-7	-15		
H	8	-64	-57	-44	-21	-3	-7		
E	9	-72	-65	-52	-29	-11	$\searrow \downarrow$ $\rightarrow \bullet$		
E	10	-80							

Best path (one among many)

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ j \rightarrow$	0	1	2	3	4	5	6	7
	0	**0**	-8	-16	-24	-32	-40	-48	-56
H	1	**-8**	-2	-10	-18	-14	-22	-30	-38
E	2	-16	**-9**	-3	-11	-18	-8	-16	-24
A	3	-24	-17	**-4**	-6	-13	-16	-3	-11
G	4	-32	-25	**-12**	-7	-8	-16	-11	-6
A	5	-40	-33	**-20**	-15	-9	-9	-11	-12
W	6	-48	-41	-28	**-5**	-13	-12	-12	-14
G	7	-56	-49	-36	**-13**	-7	-15	-12	-15
H	8	-64	-57	-44	-21	**-3**	-7	-15	-12
E	9	-72	-65	-52	-29	-11	**3**	**-5**	-9
E	10	-80	-73	-60	-37	-19	-5	2	**1**

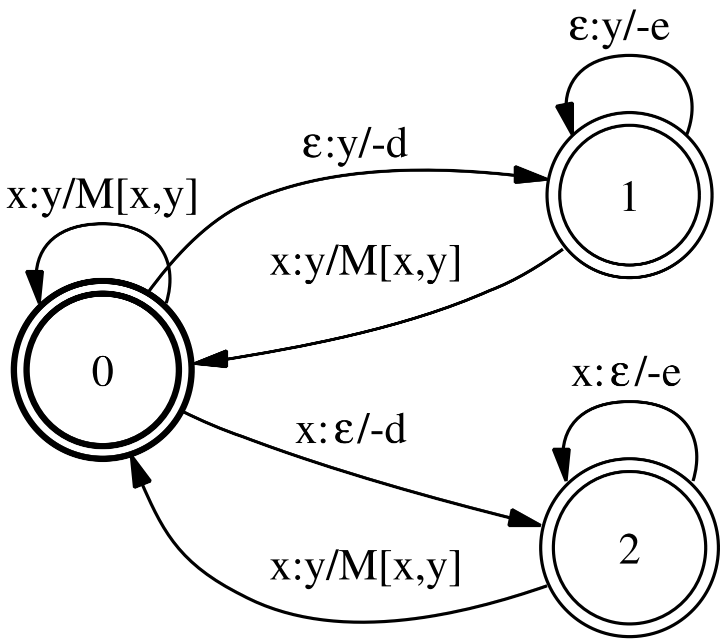
Finite-state transducer: linear gaps



	ϵ	P	A	ϵ	ϵ	W	ϵ	H	E	A	E
	H	E	A	G	A	W	G	H	E	ϵ	E
state: 0	0	0	0	0	0	0	0	0	0	0	0

- Only one state required; add scores together
- ϵ represents a gap of length 1
- gaps receive $-d$ cost for each symbol in gap
- Mapping input symbol x to output symbol y gets substitution matrix score for that pair

Finite-state transducer: affine gaps



	ϵ	P	A	ϵ	ϵ	W	ϵ	H	E	A	E
	H	E	A	G	A	W	G	H	E	ϵ	E
state: 0	1	0	0	1	1	0	1	0	0	2	0

- Three states required; add scores together
- Initial gap on input goes to state 1; initial gap on output to state 2
- gaps receive $-\boldsymbol{d}$ cost to start; plus $-\boldsymbol{e}$ for each additional symbol in gap
- Mapping input symbol $\boldsymbol{x}$ to output symbol $\boldsymbol{y}$ gets substitution matrix score for that pair

Larger chart required for dynamic programming

					P			A			W			H			E			A			
	$i \downarrow$ $j \rightarrow$	0			1			2			3			4			5			6			
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	0	0	·	·	·	-8	·	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·	·
H	1	·	·	-8	↘	→	↓																
E	2	·	·	-12																			
A	3	·	·	-16																			
G	4	·	·	-20																			
A	5	·	·	-24																			
W	6	·	·	-28																			
G	7	·	·	-32																			
H	8	·	·	-36																			
E	9	·	·	-40																			
E	10	·	·	-44																			

Larger chart required for dynamic programming

					P			A			W			H			E			A			
	i ↓ j →	0			1			2			3			4			5			6			
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	0	$\boxed{0}$	$\boxed{\cdot}$	$\boxed{\cdot}$	$\boxed{\cdot}$	-8	$\boxed{\cdot}$	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·	·
H	1	$\boxed{\cdot}$	$\boxed{\cdot}$	-8	↘	→	↓																
E	2	·	·	-12																			
A	3	·	·	-16																			
G	4	·	·	-20																			
A	5	·	·	-24																			
W	6	·	·	-28																			
G	7	·	·	-32																			
H	8	·	·	-36																			
E	9	·	·	-40																			
E	10	·	·	-44																			

State 1 only from states 0,1; State 2 from 0,2

					P			A			W			H			E			A			
	$\overset{i}{\downarrow}$ j $\rightarrow$	0			1			2			3			4			5			6			
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	0	0	·	·	·	-8	·	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·	·
H	1	·	·	-8	-2	·	·	↘	→	↓													
E	2	·	·	-12																			
A	3	·	·	-16																			
G	4	·	·	-20																			
A	5	·	·	-24																			
W	6	·	·	-28																			
G	7	·	·	-32																			
H	8	·	·	-36																			
E	9	·	·	-40																			
E	10	·	·	-44																			

State 1 only from states 0,1; State 2 from 0,2

					P			A			W			H			E			A			
	i ↓ j →	0			1			2			3			4			5			6			
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	0	0	·	·	$\boxed{\cdot}$	$\boxed{-8}$	$\boxed{\cdot}$	$\boxed{\cdot}$	-12	$\boxed{\cdot}$	·	-16	·	·	-20	·	·	-24	·	·	-28	·	·
H	1	·	·	-8	$\boxed{-2}$	$\boxed{\cdot}$	·	↘	→	↓													
E	2	·	·	-12																			
A	3	·	·	-16																			
G	4	·	·	-20																			
A	5	·	·	-24																			
W	6	·	·	-28																			
G	7	·	·	-32																			
H	8	·	·	-36																			
E	9	·	·	-40																			
E	10	·	·	-44																			

State 1 costs $-d$ from state 0; only $-e$ from state 1

					P			A			W			H			E			A			
	$i \downarrow$ $j \rightarrow$	0			1			2			3			4			5			6			
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	0	0	·	·	·	-8	·	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·	·
H	1	·	·	-8	-2	·	·	-10	-10	·	↘	→	↓										
E	2	·	·	-12																			
A	3	·	·	-16																			
G	4	·	·	-20																			
A	5	·	·	-24																			
W	6	·	·	-28																			
G	7	·	·	-32																			
H	8	·	·	-36																			
E	9	·	·	-40																			
E	10	·	·	-44																			

State 1 only from states 0,1; State 2 from 0,2

					P			A			W			H			E			A			
	$\overset{i}{\downarrow}$ j →	0			1			2			3			4			5			6			
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0
	0	0	·	·	·	-8	·	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·	·
H	1	·	·	-8	-2	·	·	-10	-10	·	-15	-14	·										
E	2	·	·	-12	↘	→	↓																
A	3	·	·	-16																			
G	4	·	·	-20																			
A	5	·	·	-24																			
W	6	·	·	-28																			
G	7	·	·	-32																			
H	8	·	·	-36																			
E	9	·	·	-40																			
E	10	·	·	-44																			

State 2 costs $-d$ from state 0; only $-e$ from state 2

					P			A			W			H			E			A		
	$\overset{i}{\downarrow}$ $j \rightarrow$	0			1			2			3			4			5			6		
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
	0	0	·	·	·	-8	·	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·
H	1	·	·	-8	-2	·	·	-10	-10	·	-15	-14	·									
E	2	·	·	-12	-9	·	-10															
A	3	·	·	-16	↘	→	↓															
G	4	·	·	-20																		
A	5	·	·	-24																		
W	6	·	·	-28																		
G	7	·	·	-32																		
H	8	·	·	-36																		
E	9	·	·	-40																		
E	10	·	·	-44																		

And so on – same dynamic programming

					P			A			W			H			E			A		
	i ↓ j →	0			1			2			3			4			5			6		
	state:	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2
	0	0	·	·	·	-8	·	·	-12	·	·	-16	·	·	-20	·	·	-24	·	·	-28	·
H	1	·	·	-8	-2	·	·	-10	-10	·	-15	-14	·									
E	2	·	·	-12	-9	·	-10															
A	3	·	·	-16	-13	·	-14															
G	4	·	·	-20																		
A	5	·	·	-24																		
W	6	·	·	-28																		
G	7	·	·	-32																		
H	8	·	·	-36																		
E	9	·	·	-40																		
E	10	·	·	-44																		

Finite-state transducers for alignment

- Can move to arbitrarily complex finite-state transducer models
  - Durbin et al. discuss a 4 state model, with two match states corresponding to low and high fidelity regions
- Must keep track of scores at each state in dynamic programming

Local alignment

- Simple idea: allow resetting alignment at any point
- Get high quality local alignments, rather than global alignments
- Same algorithm, except now:

$$\boldsymbol{F(i,j) = \max \left\{ \begin{array}{l} 0, \\ F(i,j{-}1) - d, \\ F(i{-}1,j) - d, \\ F(i{-}1,j{-}1) + M[S_1(i), S_2(j)] \end{array} \right\}}$$

- Similar modification for multi-state models
- Note: assumes scores less than zero
  - PAM250 won't work unmodified

Initialize zero positions (Global)

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7
	0	0	-8	-16	-24	-32	-40	-48	-56
H	1	-8							
E	2	-16							
A	3	-24							
G	4	-32							
A	5	-40							
W	6	-48							
G	7	-56							
H	8	-64							
E	9	-72							
E	10	-80							

Initialize zero positions (Local)

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ {}_j \rightarrow$	0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
H	1	0							
E	2	0							
A	3	0							
G	4	0							
A	5	0							
W	6	0							
G	7	0							
H	8	0							
E	9	0							
E	10	0							

P no matches; H 1 match

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ _j \rightarrow$	0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
H	1	0	0	0	0	10	0	0	0
E	2	0	0						
A	3	0	0						
G	4	0	0						
A	5	0	0						
W	6	0	0						
G	7	0	0						
H	8	0	0						
E	9	0	0						
E	10	0	0						

4 non-zero cells in next row

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}\ {}_{j} \rightarrow$	0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
H	1	0	0	0	0	10	0	0	0
E	2	0	0	0	0	2	16	8	6
A	3	0	0						
G	4	0	0						
A	5	0	0						
W	6	0	0						
G	7	0	0						
H	8	0	0						
E	9	0	0						
E	10	0	0						

Great local match – not in global solutions

			P	A	W	H	E	A	E
	$\overset{i}{\downarrow}$ $j \rightarrow$	0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
H	1	0	0	0	0	**10**	0	0	0
E	2	0	0	0	0	2	**16**	8	6
A	3	0	0	5	0	0	8	**21**	13
G	4	0	0						
A	5	0	0						
W	6	0	0						
G	7	0	0						
H	8	0	0						
E	9	0	0						
E	10	0	0						

Sequence processing tasks using HMMs

- Gene prediction
  - Non-hierarchical bracketing task:

    are nucleotides inside an exon, intron or outside?
  - Complicated graph structures for multi-exon genes
- Named-entity extraction
  - Non-hierarchical bracketing task:

    are words inside a named-entity (possibly of different types) or outside?
  - Different graph structures for different kinds of entities

CpG Islands

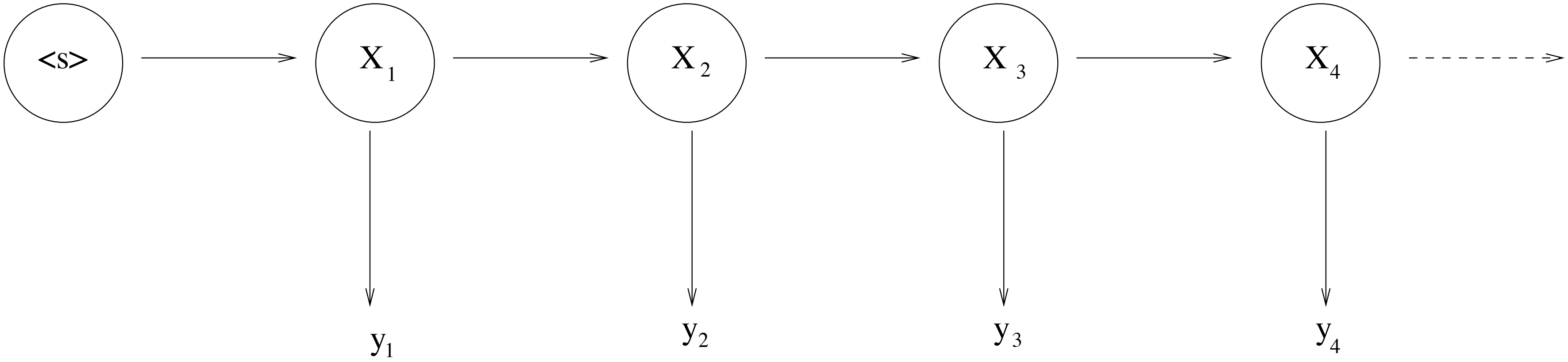
- Some parts of the nucleotide sequence are more resistant to change
  - Functionally critical regions, e.g., promoter regions
- Some local configurations are prone to change
  - 'methylation': CG $\rightarrow$ TG
- If we find many change-prone local configurations in a particular region, this is evidence of regional change resistance
- Useful evidence of functional importance
- Call areas with lots of CG pairs called 'CpG Islands'

HMM alignment

- When tagging, one state transition per symbol
- When aligning, that will generally not be the case
  - Deletions and insertions require variable number of state transitions
- Each transition is labeled with a symbol pair
  - substitution: x:y
  - deletion: x:ϵ
  - insertion: ϵ:y

HMMs

- Sequence of hidden states, representing variables $\boldsymbol{X}$
  - e.g., whether or not in a CpG Island
- States output the observed values $\boldsymbol{y}$
  - in this case, the particular nucleotide
- Typical graphical model representation:

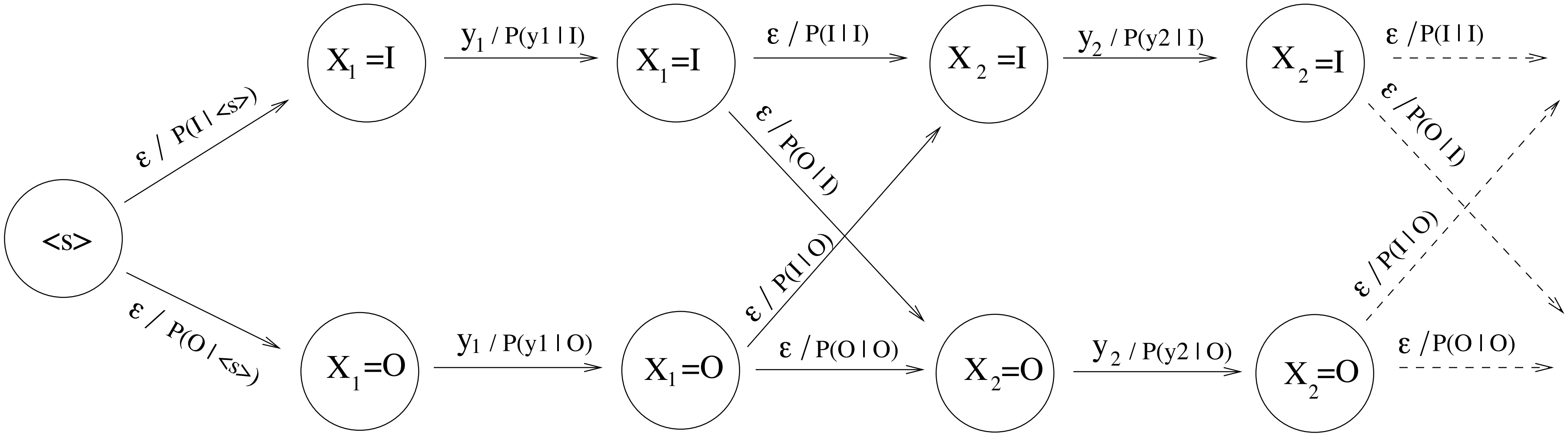


HMM parameterization

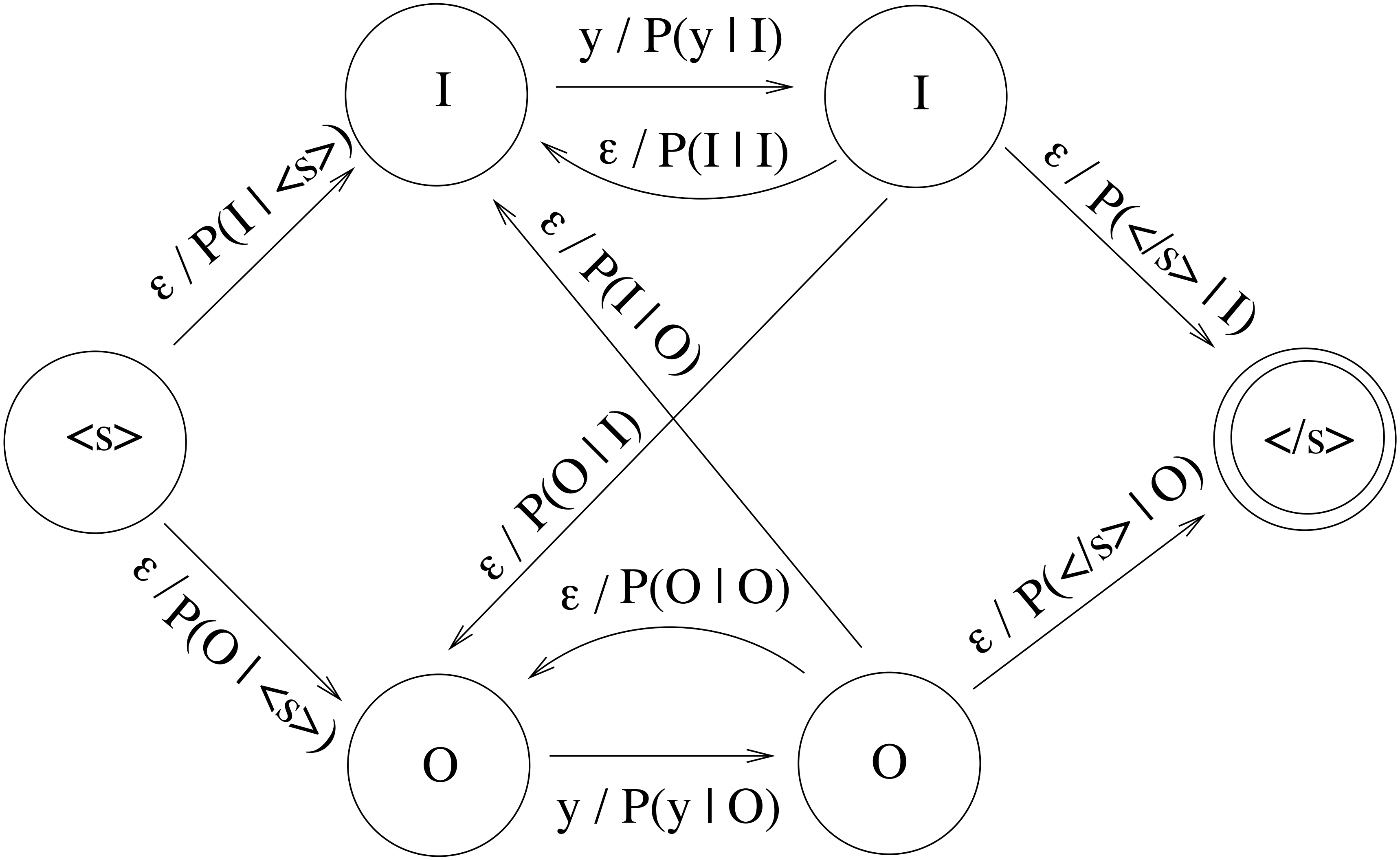
- Model consists of two kinds of parameters
  - Transition probabilities between states: $\mathrm{P}(\boldsymbol{X}_i = \boldsymbol{x} \mid \boldsymbol{X}_{i-1} = \boldsymbol{x}')$, for some instantiated values $\boldsymbol{x}, \boldsymbol{x}'$
  - Emission probabilities from states to observations: $\mathrm{P}(\boldsymbol{y}_i \mid \boldsymbol{X}_i = \boldsymbol{x})$
- When we include start and end states, this defines a probability distribution over joint state/observation sequences
  - Can use it to infer the “best” state sequence for a given observation sequence

Explicitly breaking out states in HMM

- Transitions with ϵ output
  - Carrying the HMM state transition probabilites
- Transitions with y_i output
  - Carrying the HMM emission probabilities



Weighted finite-state automaton representation



Larger state space

- This model will not do a good job of modeling CpG islands
- Why not?
  - CpG islands are regions with CG neighbors
  - In the current model, the probability of outputting a G depends only on whether the hidden state is I or O
  - The model forgets whether the previous observation was C or not
- The solution is to stop the model from forgetting about C
- We will split the states of our HMM to encode the previous symbol

General HMM notation

- Let $a_{x,x'}$ denote the transition probability:

$$\begin{aligned} a_{x,x'} &= \mathrm{P}(X_i{=}x' \mid X_{i-1}{=}x) \\ &= \mathrm{P}(x' \mid x) \end{aligned}$$

- Let $a_{\overline{x}}$ be shorthand for $a_{\text{<s>},x}$
- Let $a_{\underline{x}}$ be shorthand for $a_{x,\text{</s>}}$
- Let $b_{x,y}$ denote the emission probability:

$$\begin{aligned} b_{x,y} &= \mathrm{P}(Y_i{=}y \mid X_i{=}x) \\ &= \mathrm{P}(y \mid x) \end{aligned}$$

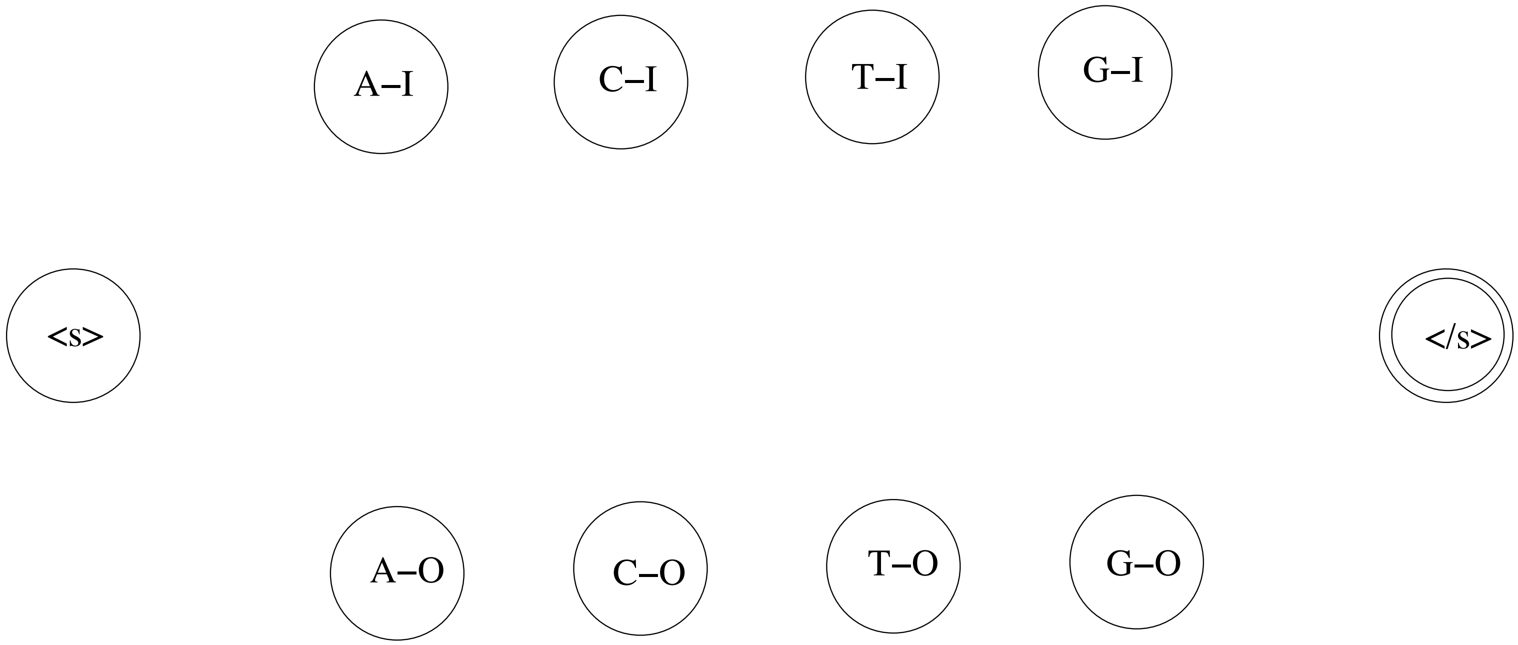
Larger state space HMM

- Need to remember previous symbol, and whether I or O
- Hence, since $\Sigma = \{$A,C,T,G$\}$, there are 10 states: <s>, </s>, A-I, C-I, T-I, G-I, A-O, C-O, T-O, G-O
- Note: for any non-start/stop state, only one possible observation:

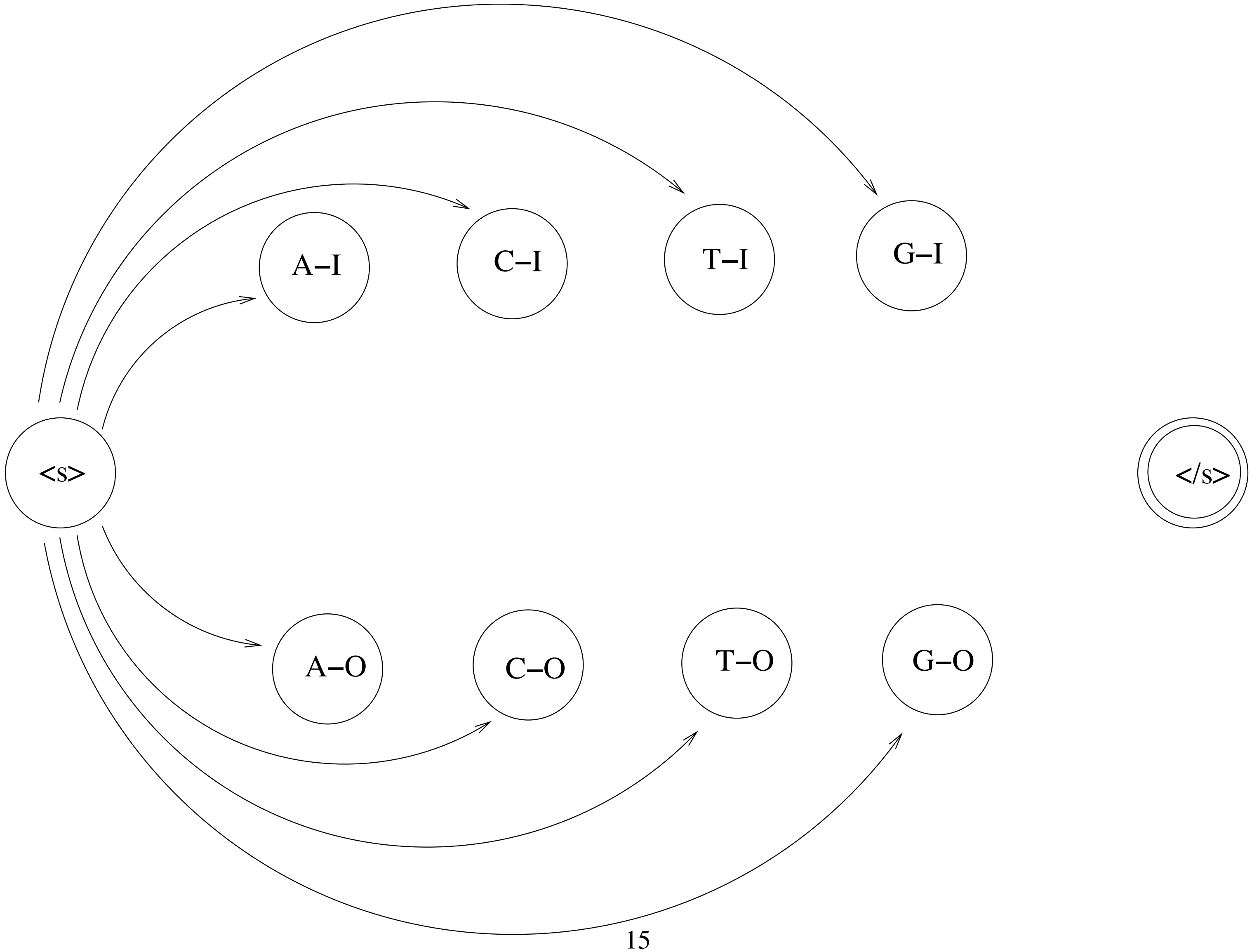
$$\boldsymbol{b}_{\boldsymbol{X}\text{-}\boldsymbol{A},\boldsymbol{X}} = \mathbf{1} \qquad \text{for } \boldsymbol{X} \in \Sigma \text{ and } \boldsymbol{A} \in \{\text{I,O}\}$$

- Many more transition probabilities
  - 64 transitions between $\boldsymbol{X}$-$\boldsymbol{A}$ symbols
  - 8 start and 8 stop transitions
- Hopefully P(G-I | C-I) $\gg$ P(G-O | C-O), i.e., $\boldsymbol{a}_{\text{C-I,G-I}} \gg \boldsymbol{a}_{\text{C-O,G-O}}$

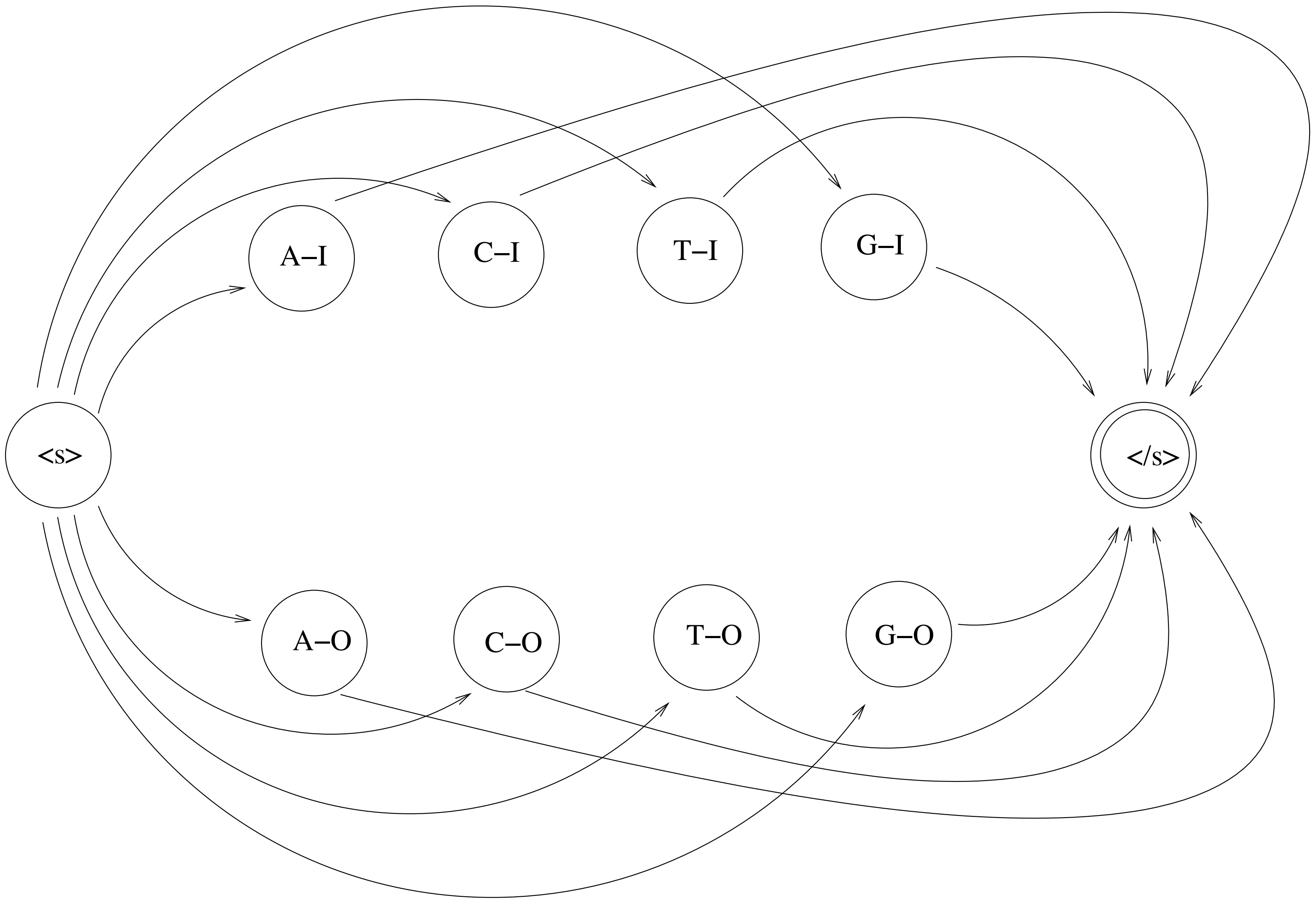
Larger state space FSA



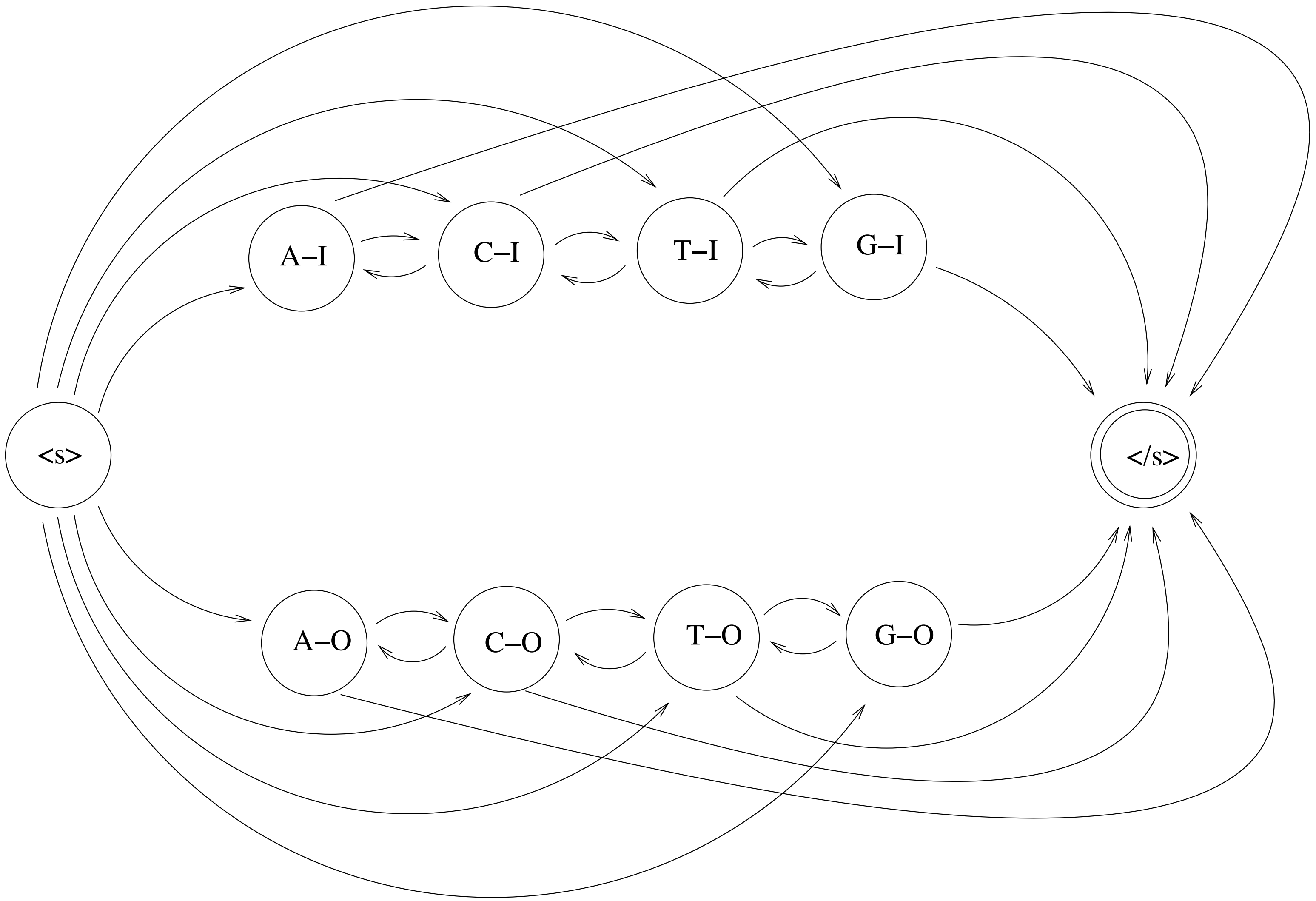
Add start transitions to larger state space FSA



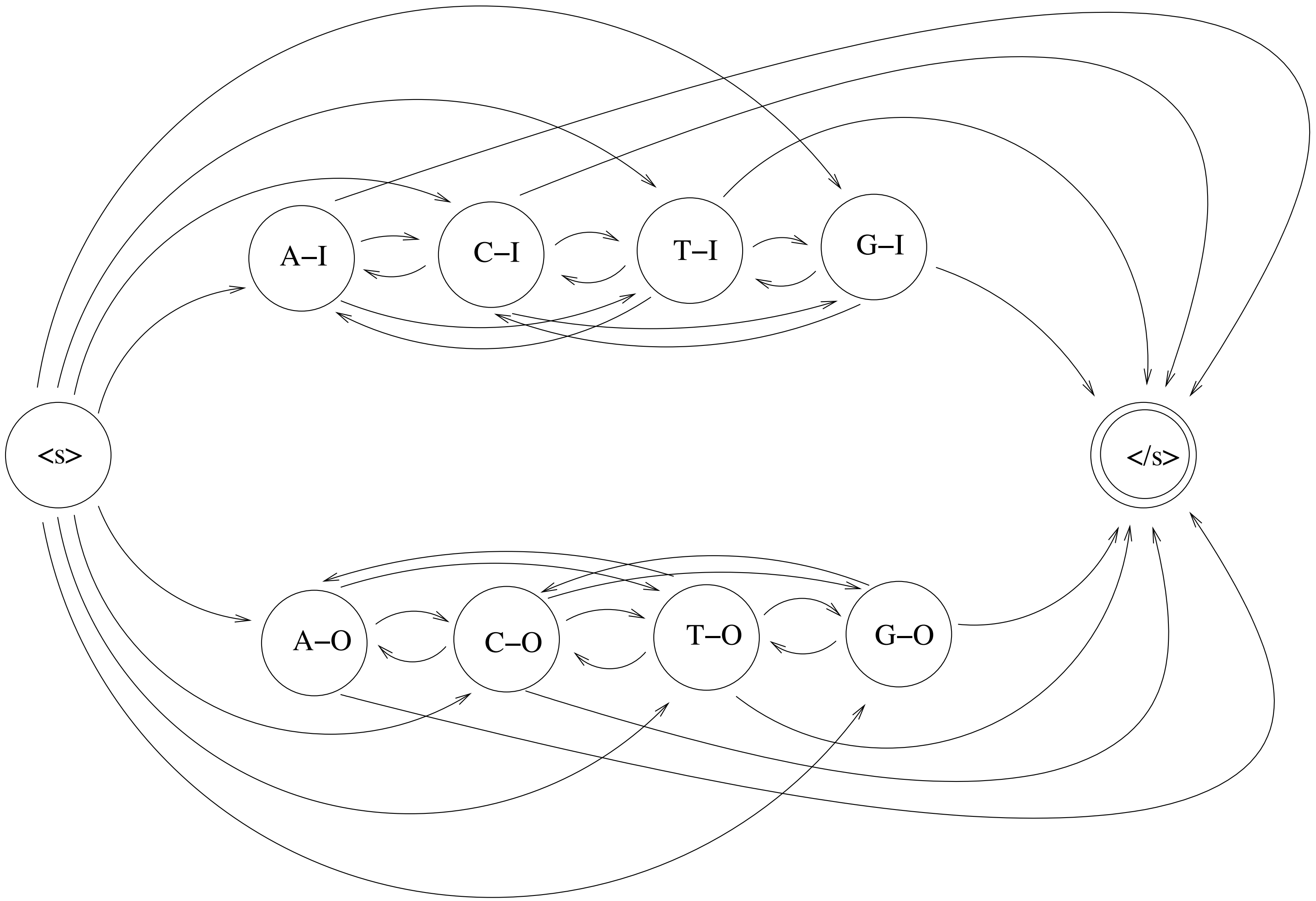
Add final transitions to larger state space FSA



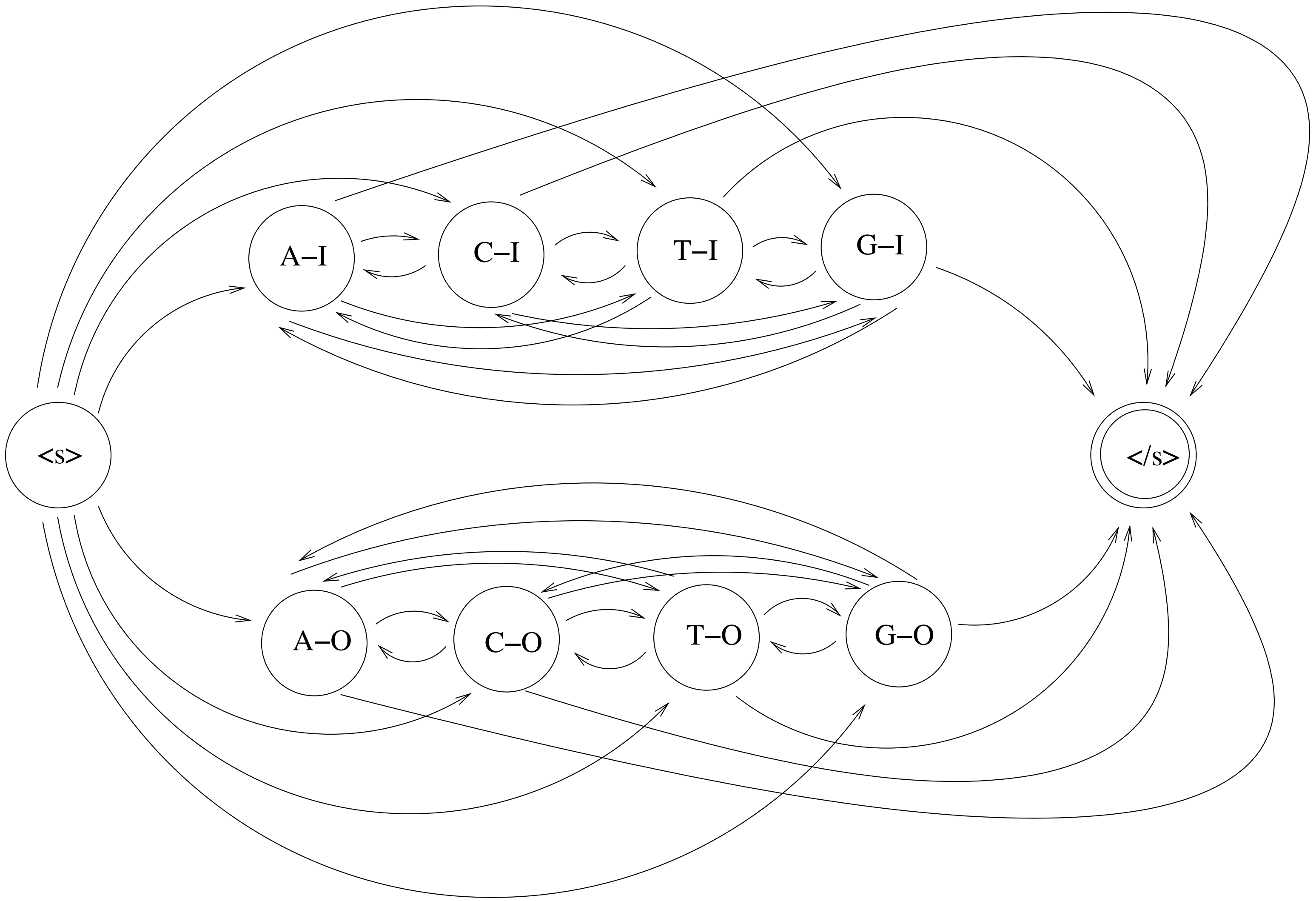
Add adjacent state arcs to larger state space FSA



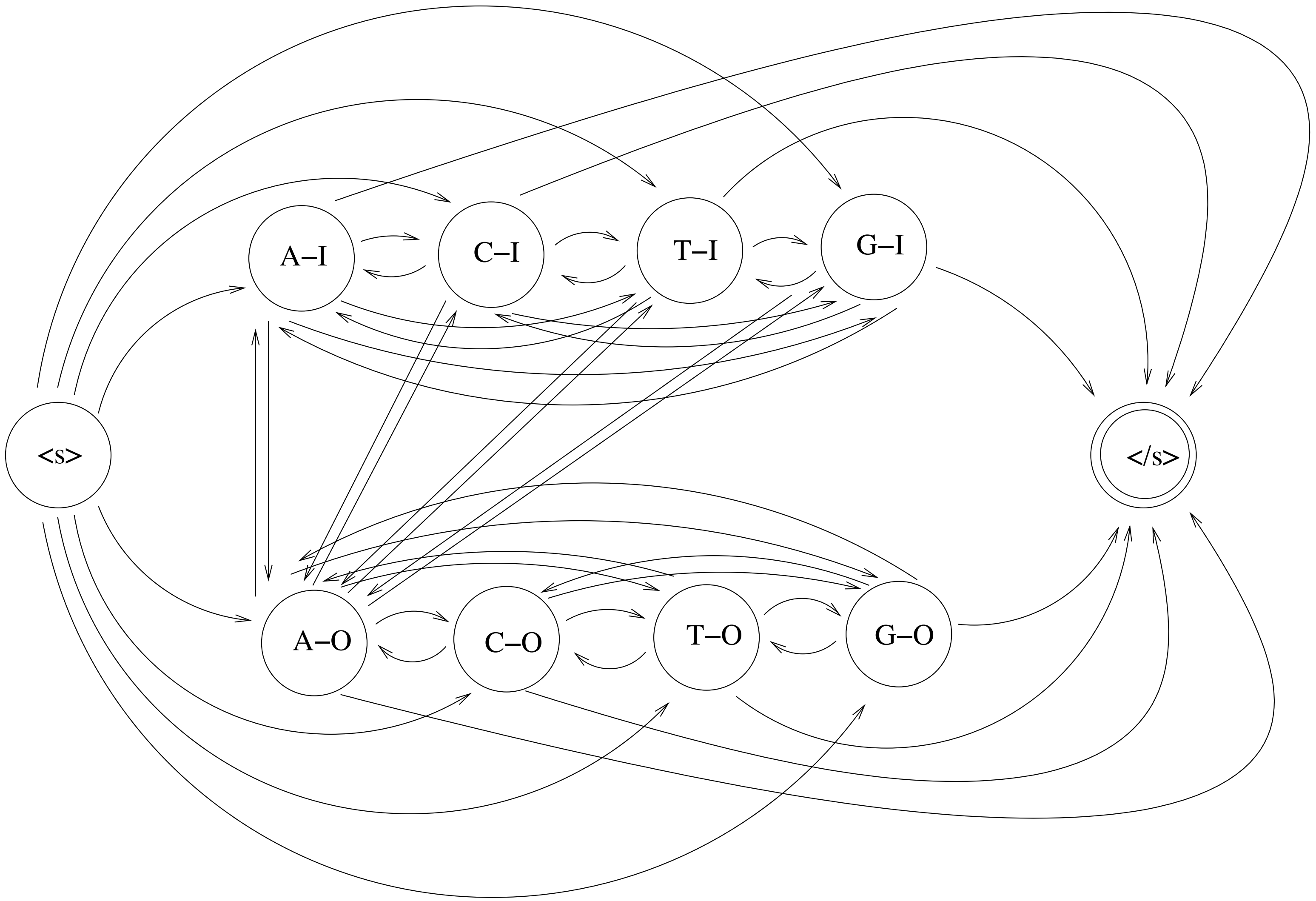
Add longer distance arcs to larger state space FSA



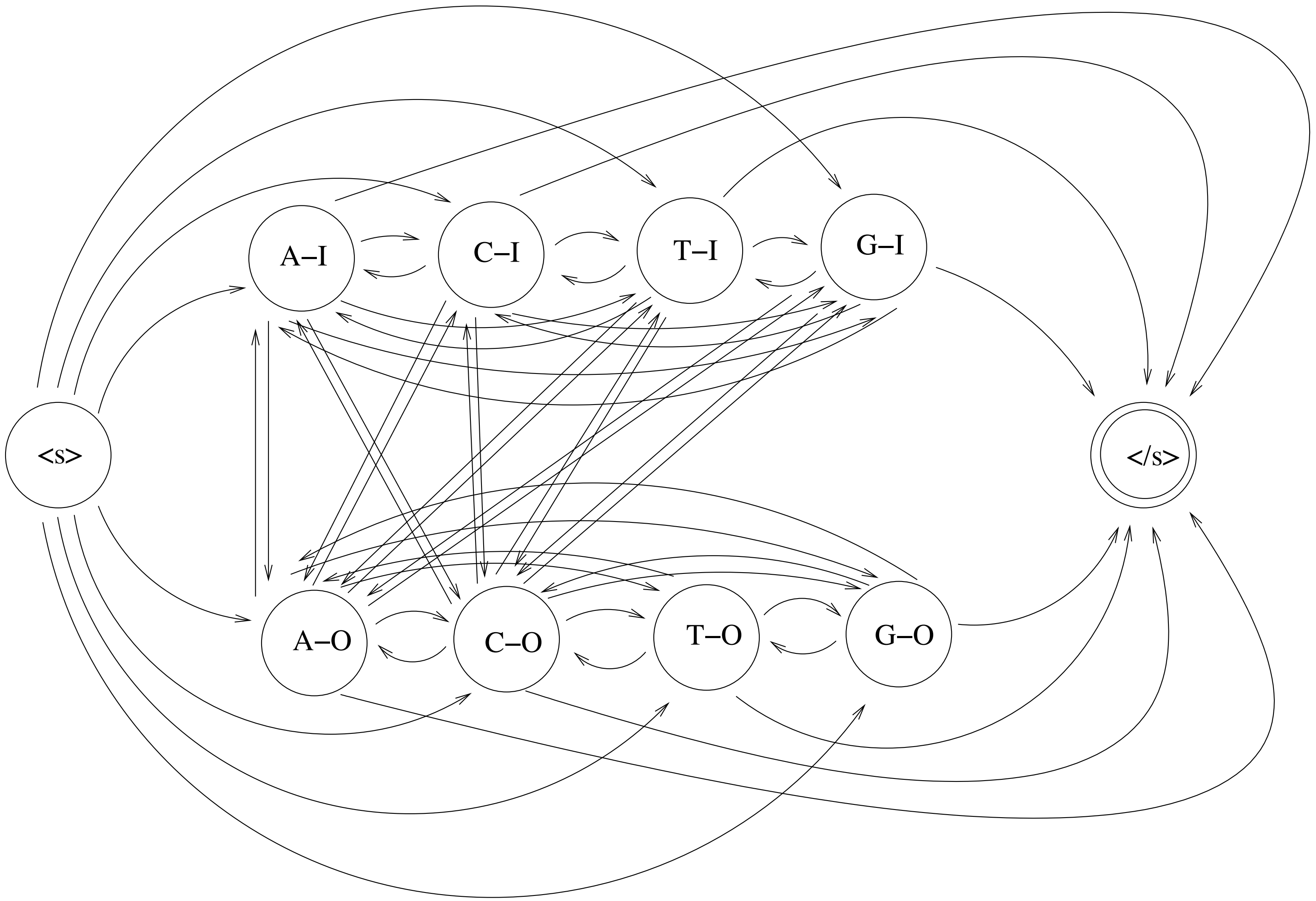
Add longer distance arcs to larger state space FSA



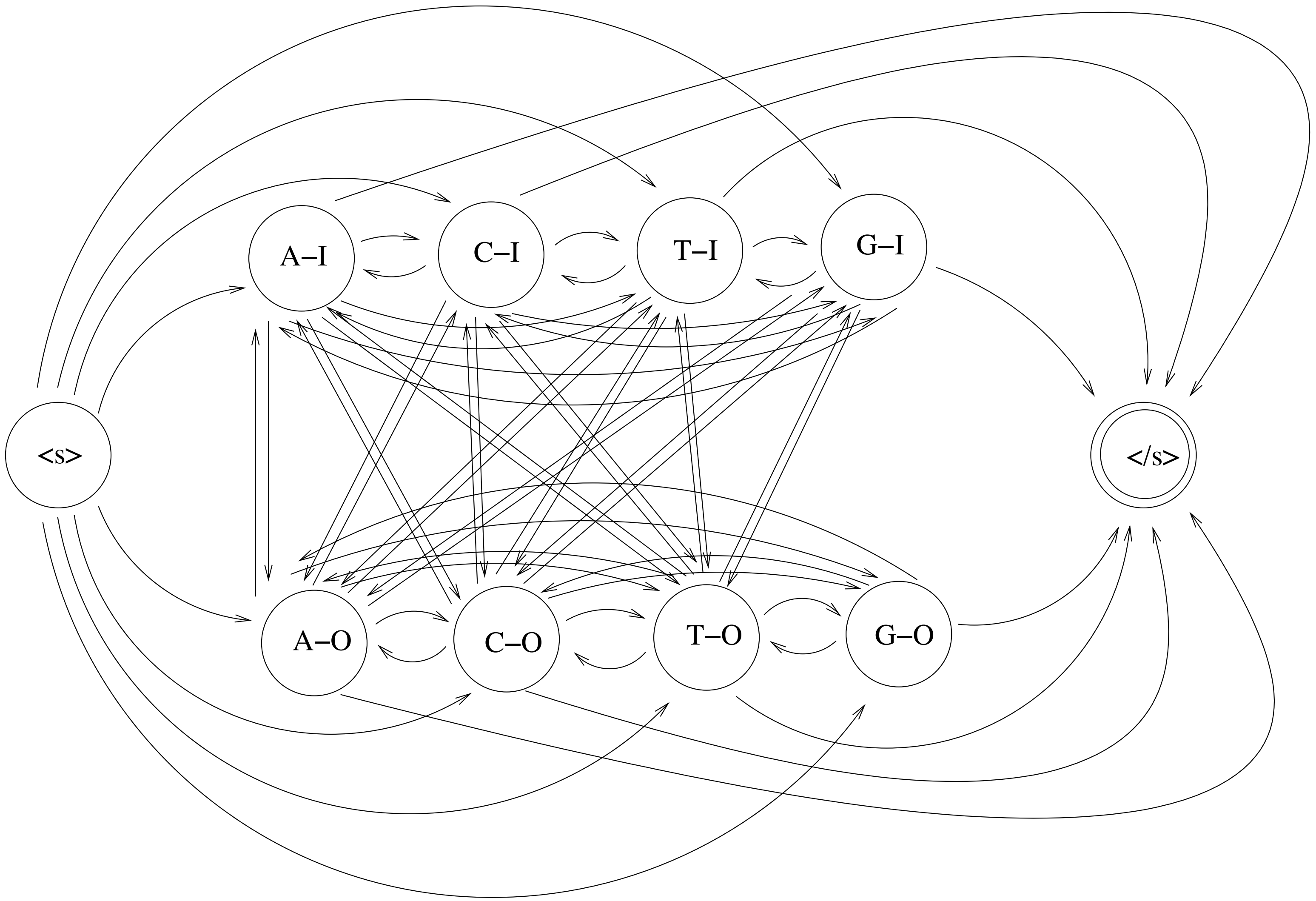
Add longer distance arcs to larger state space FSA



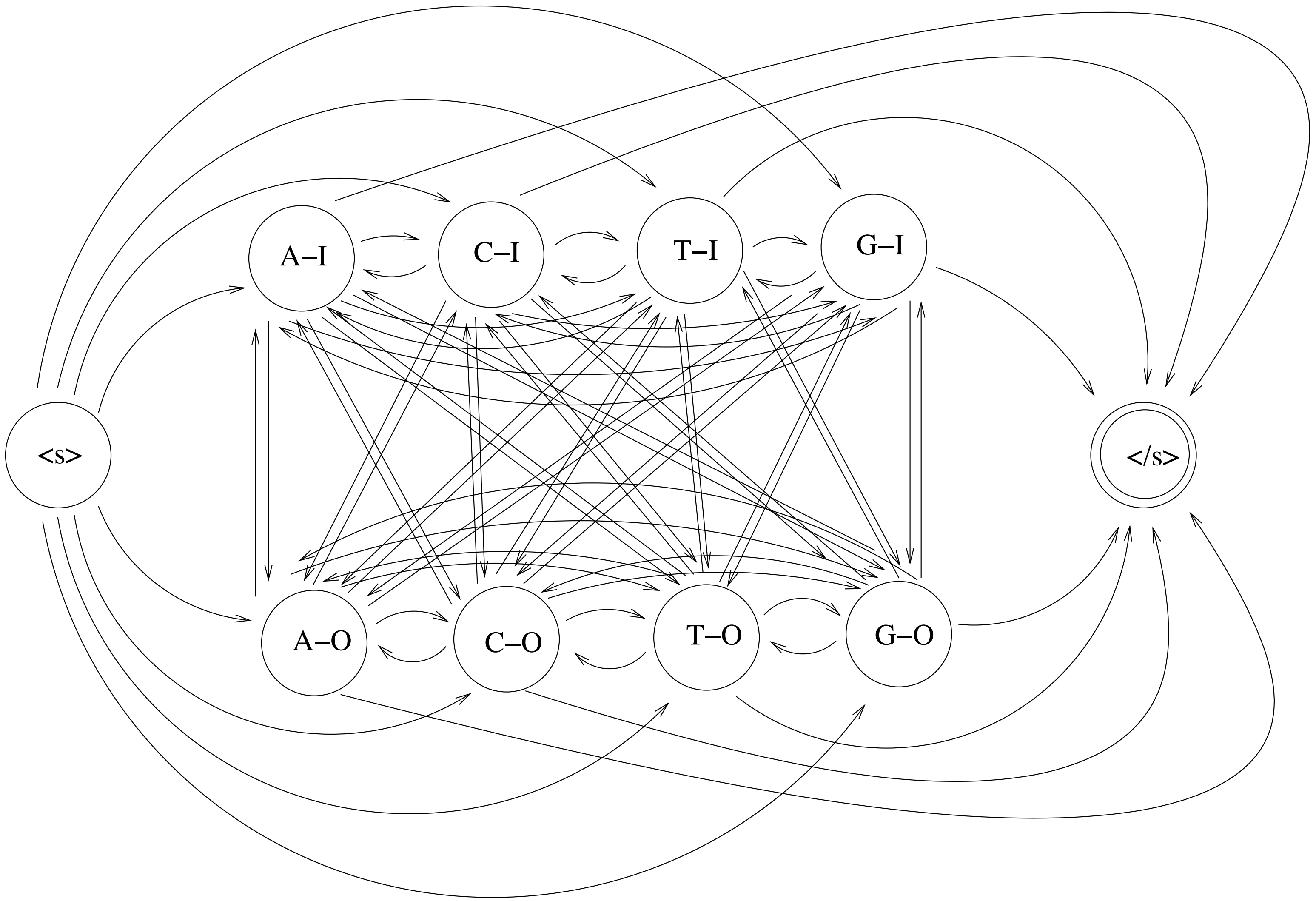
Add longer distance arcs to larger state space FSA



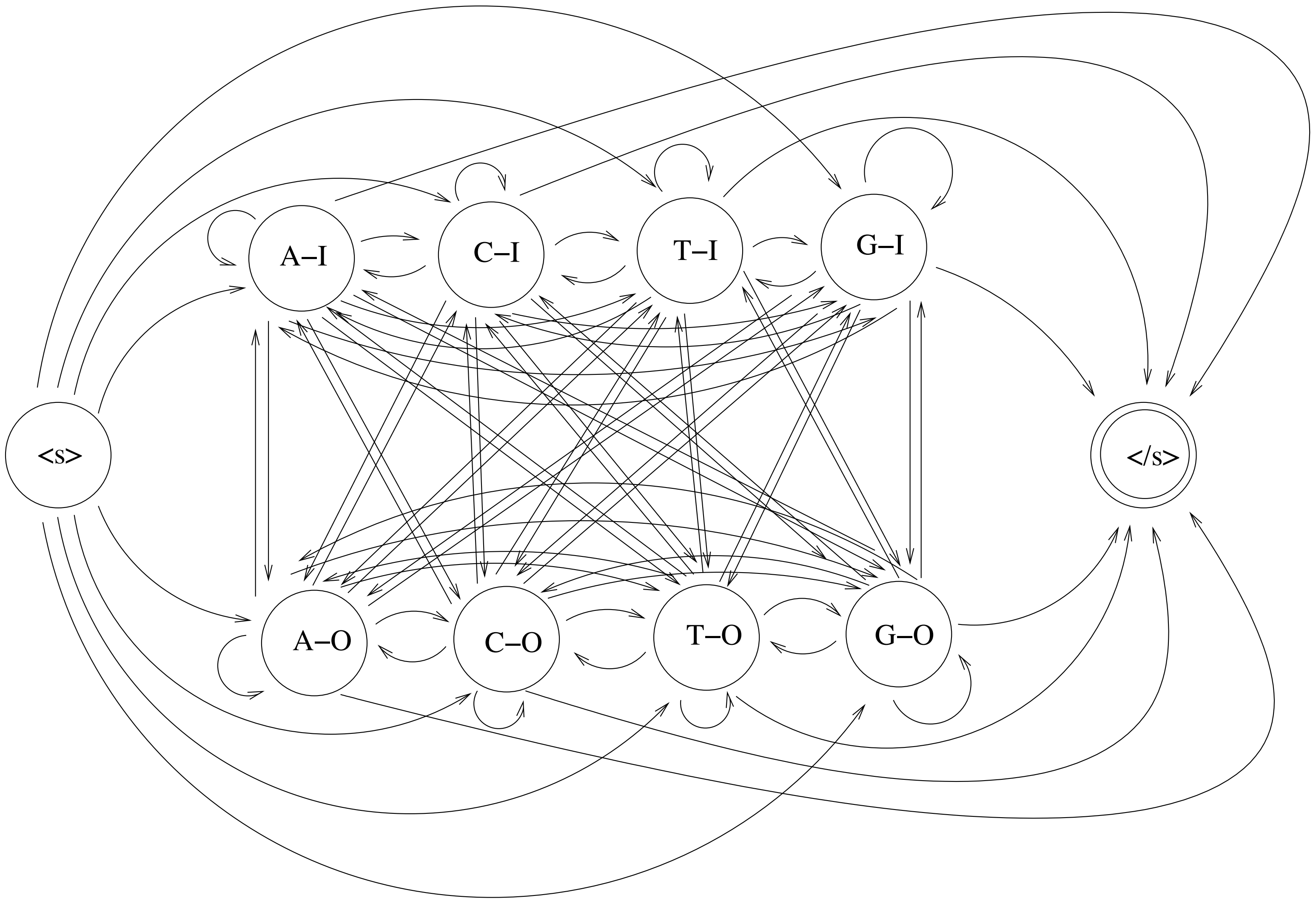
Add longer distance arcs to larger state space FSA



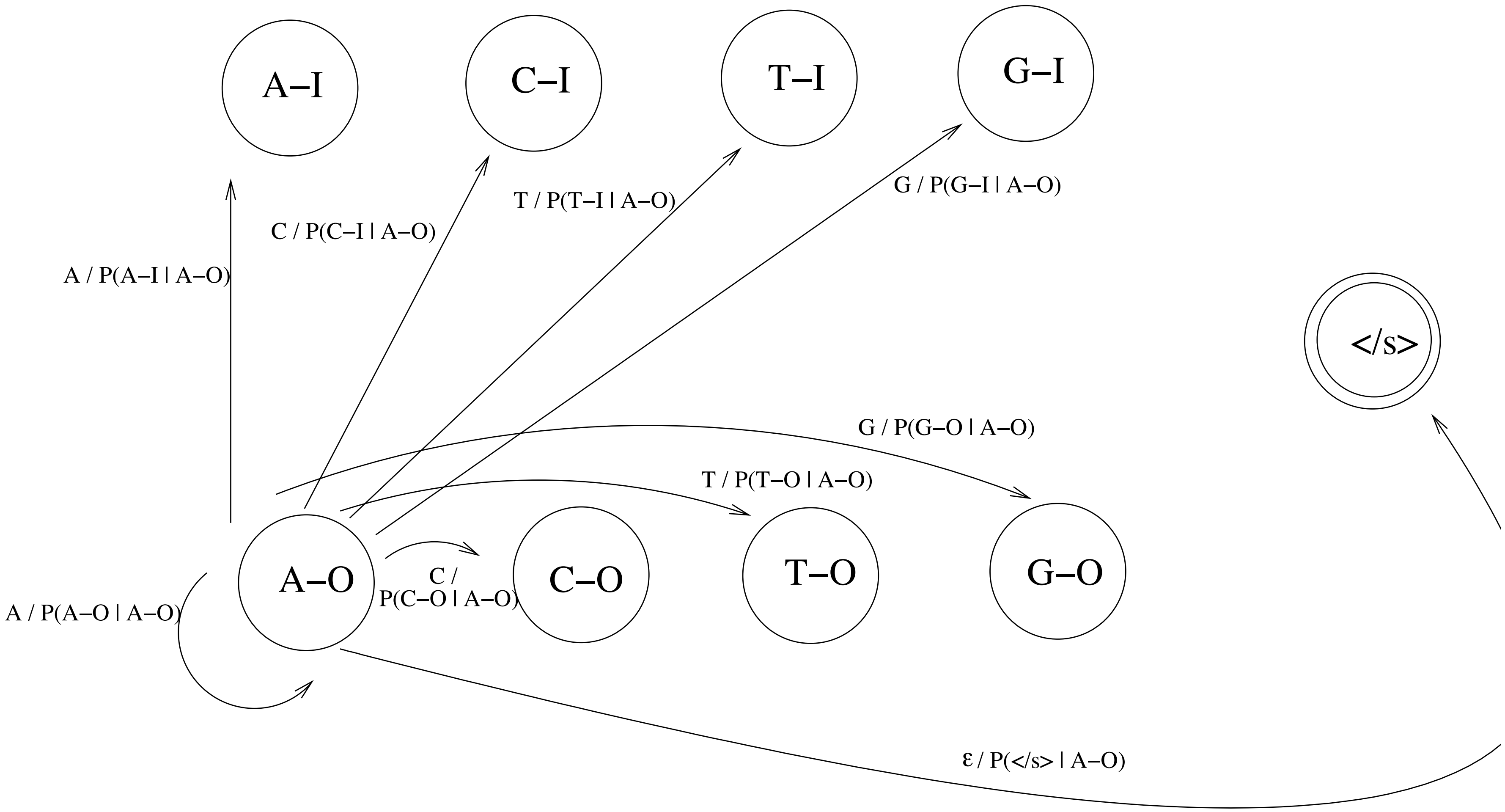
Add longer distance arcs to larger state space FSA



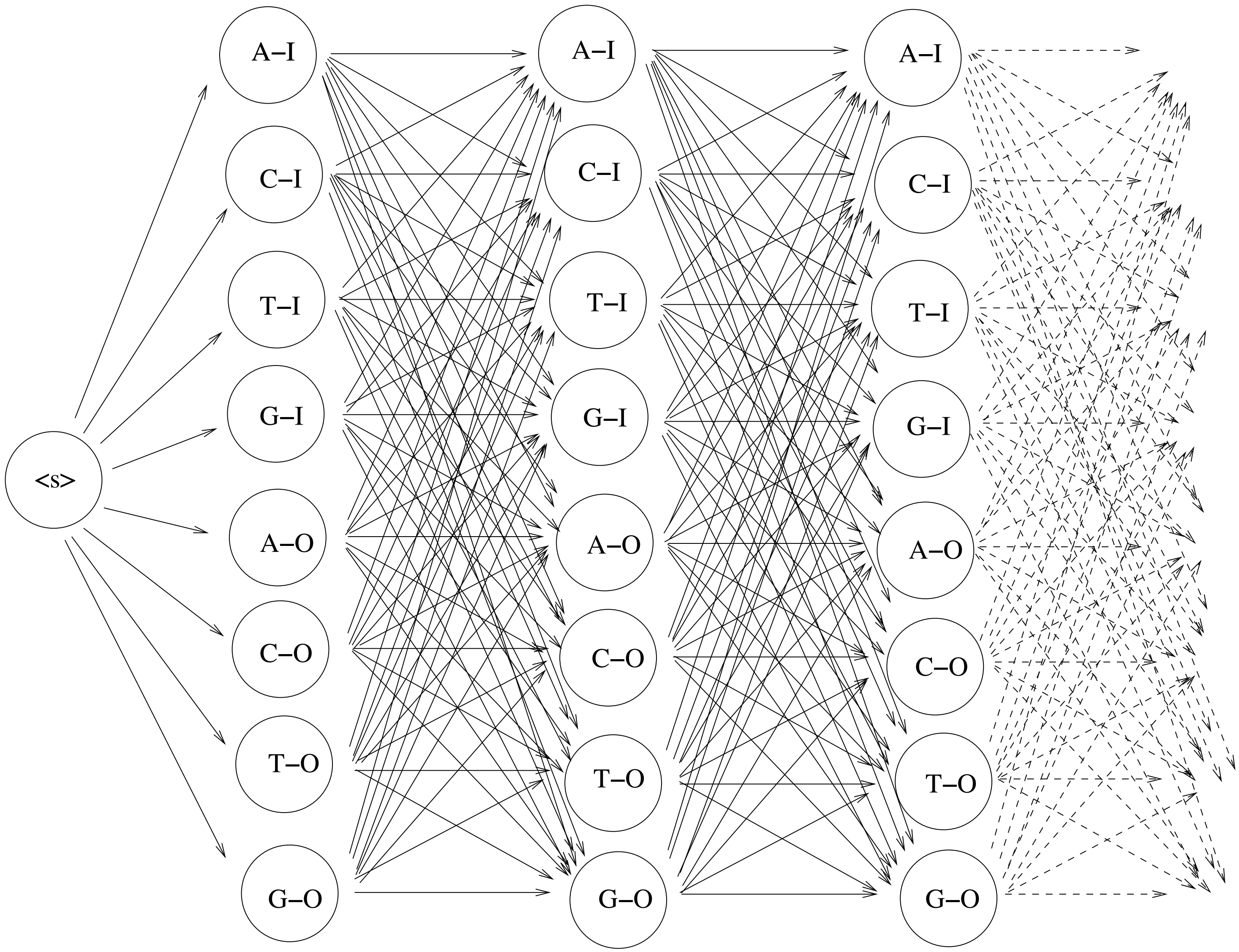
Add self loops to larger state space FSA



Don't forget labels (and probs) on transitions



Full Decoding graph



Sequence processing tasks using HMMs

- Gene prediction
  - Non-hierarchical bracketing task:
    are nucleotides inside an exon, intron or outside?
  - Complicated graph structures for multi-exon genes
- Named-entity extraction
  - Non-hierarchical bracketing task:
    are words inside a named-entity (possibly of different types) or outside?
  - Different graph structures for different kinds of entities

HMM alignment

- When tagging, one state transition per symbol
- When aligning, that will generally not be the case
    - Deletions and insertions require variable number of state transitions
- Each transition is labeled with a symbol pair
    - substitution: x:y
    - deletion: x:ϵ
    - insertion: ϵ:y

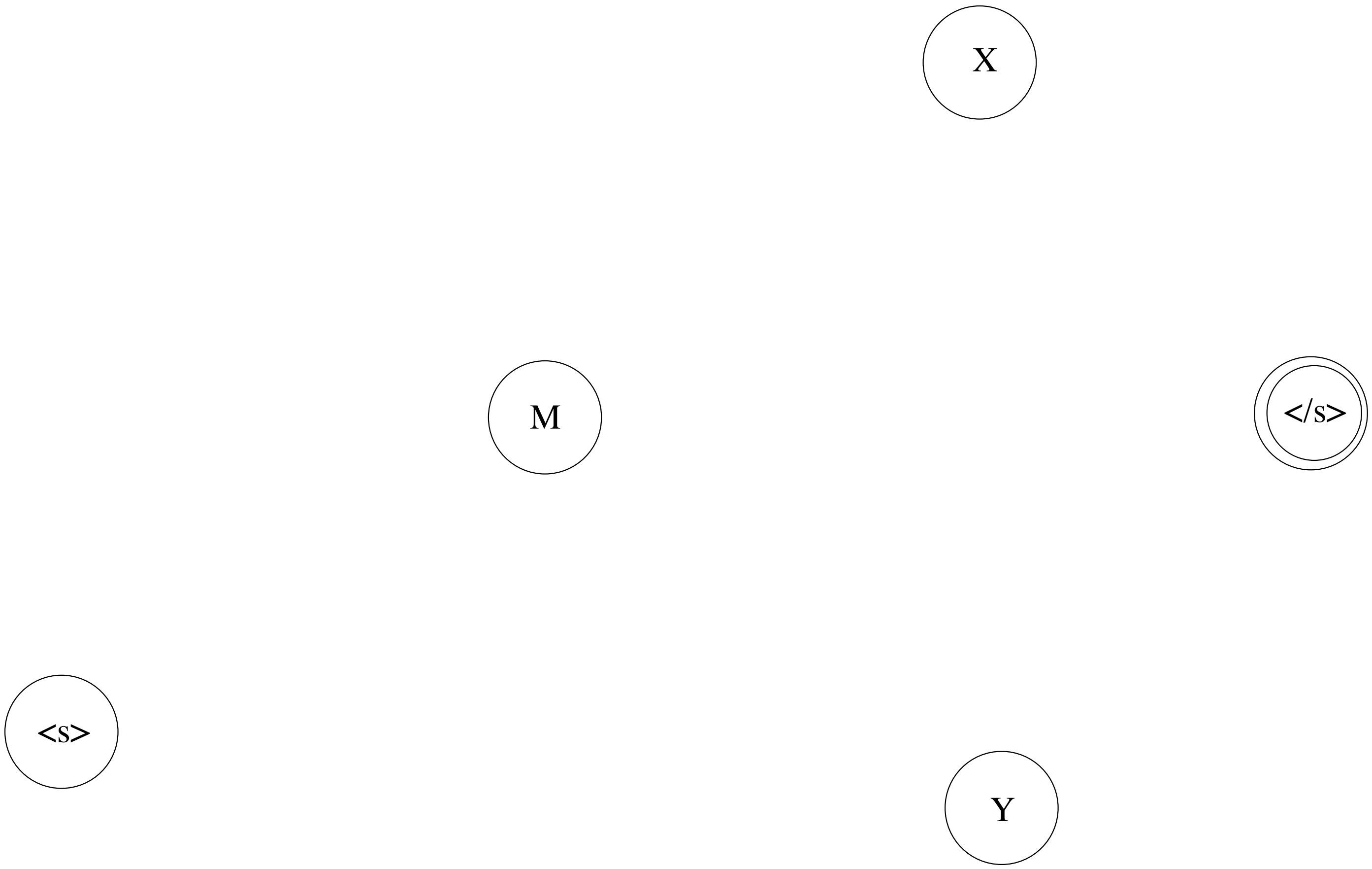
Affine gap alignment as HMM

- To define an HMM, we first need to define the states
- Second, the transition and emission probabilities
  - which we denote $\boldsymbol{a_{x,x'}}$ and $\boldsymbol{b_{x,y}}$
  - (Recall, last example, $\boldsymbol{b_{x,y}}$ was always 0 or 1, hence ignored)
- Then, let's look at the graph

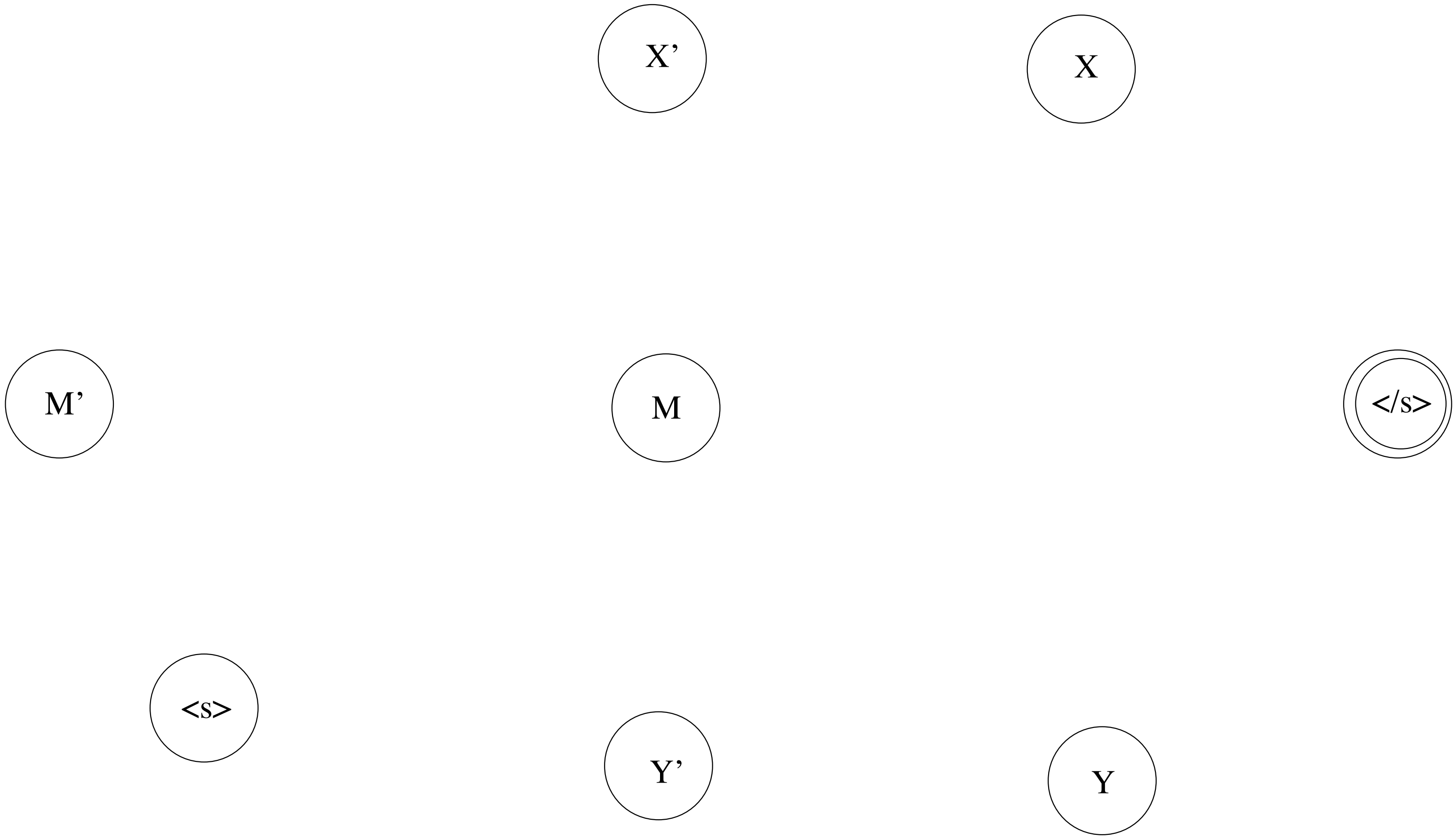
States in affine gap model

- Start (<s>) and stop (</s>) states
- State after zero deletions or insertions (M)
- State after one or more deletion (X)
- State after one or more insertion (Y)

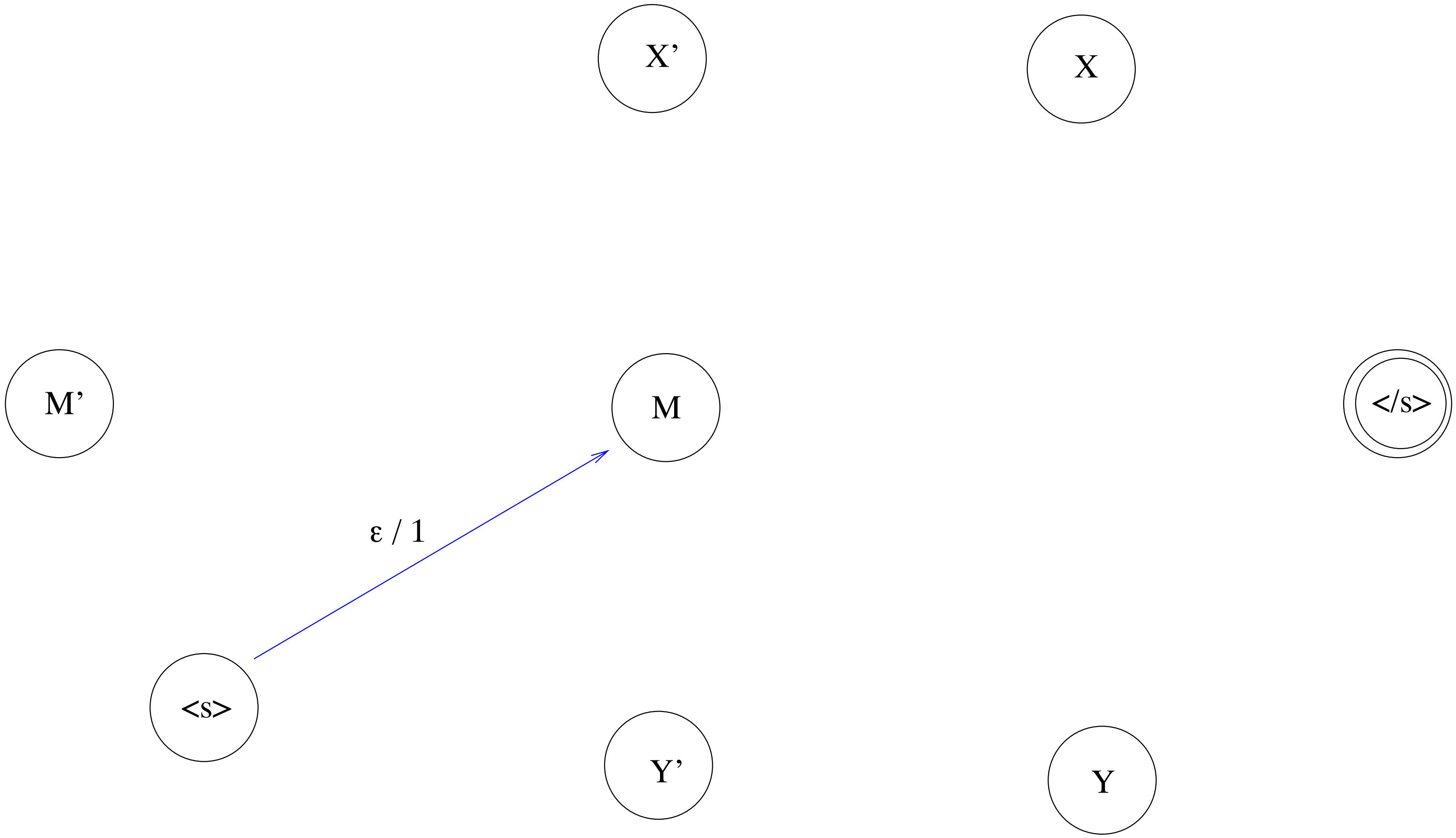
Affine gap HMM transducer states



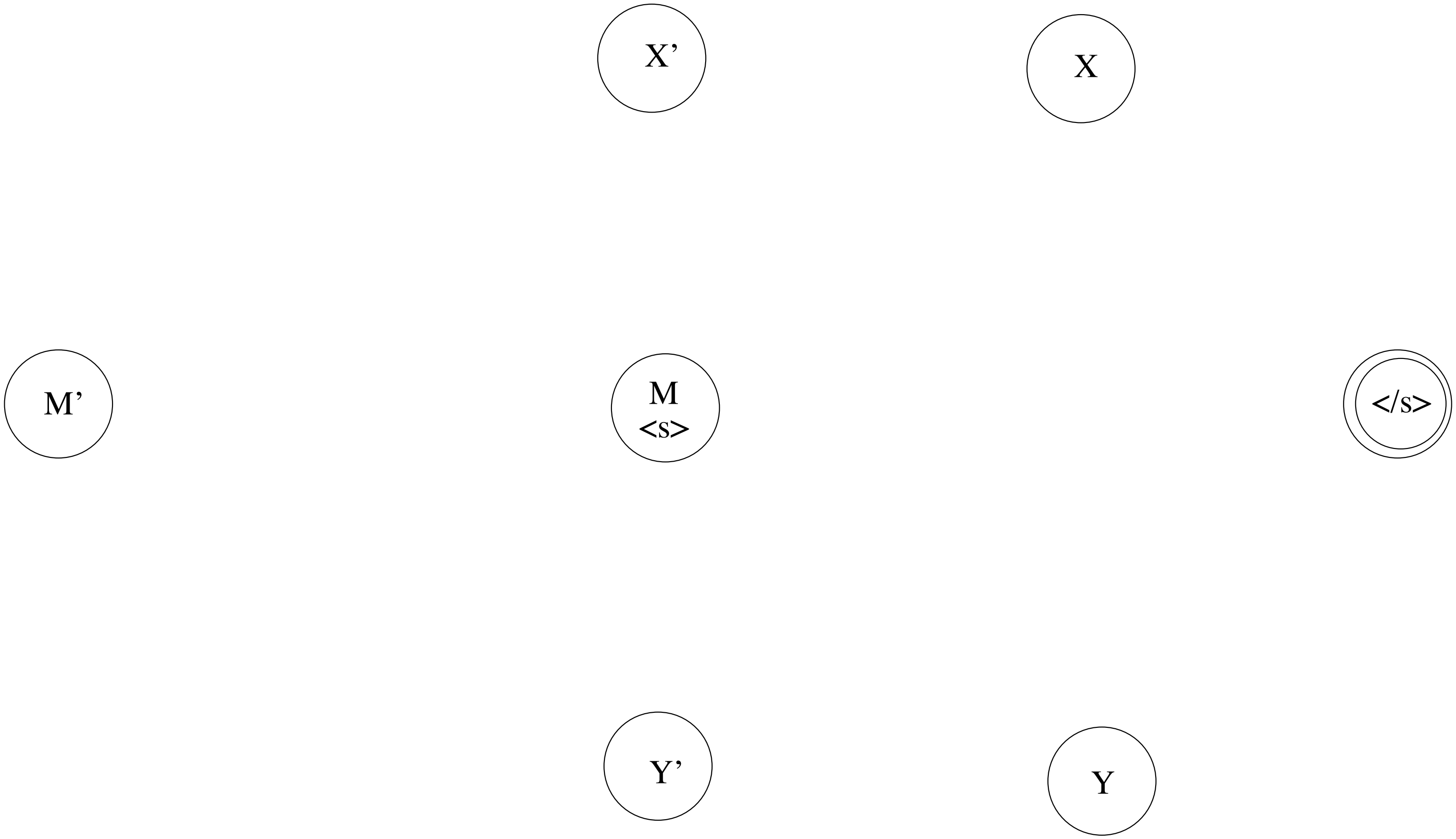
Affine gap HMM transducer states



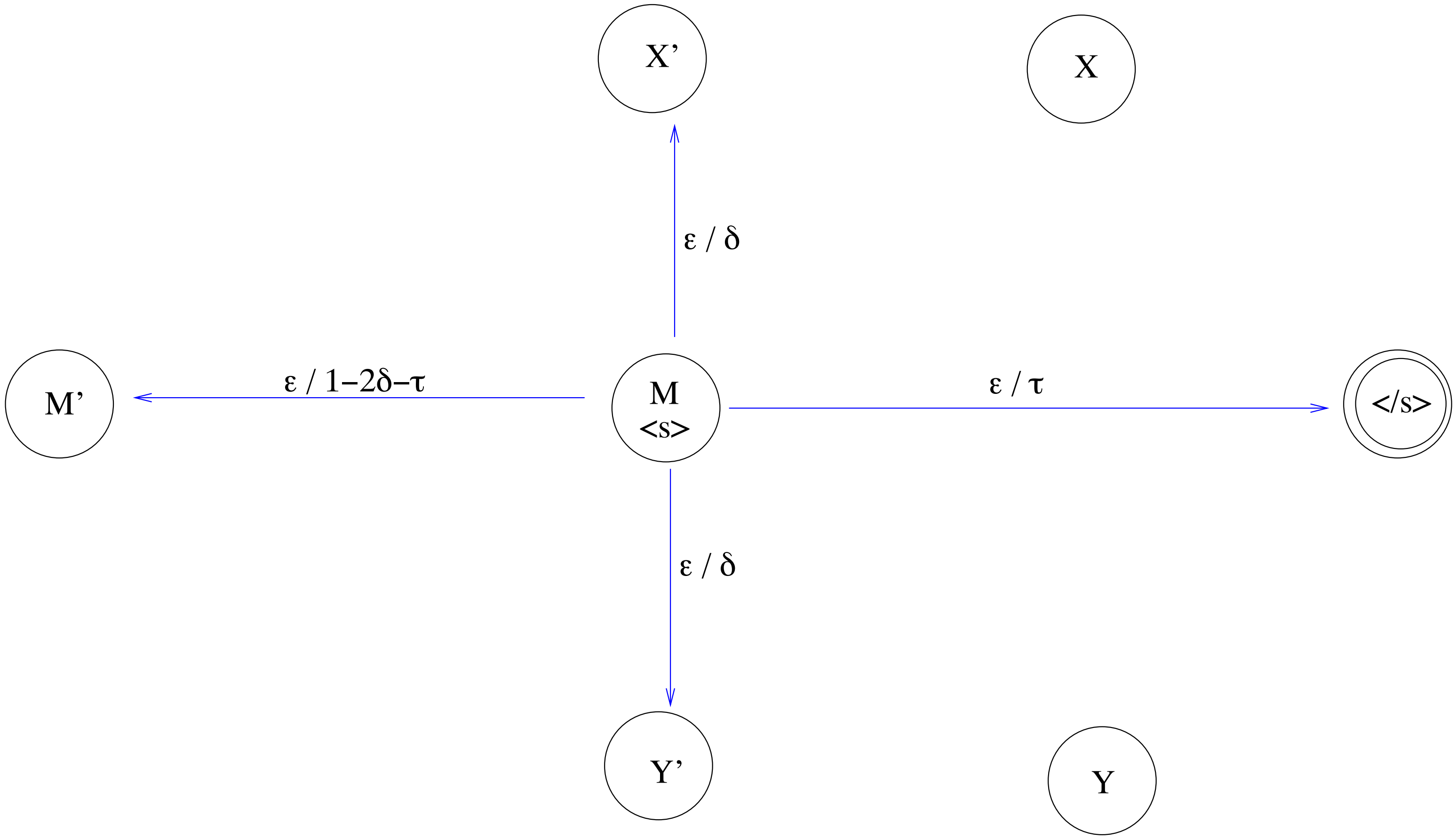
Affine gap HMM transducer states



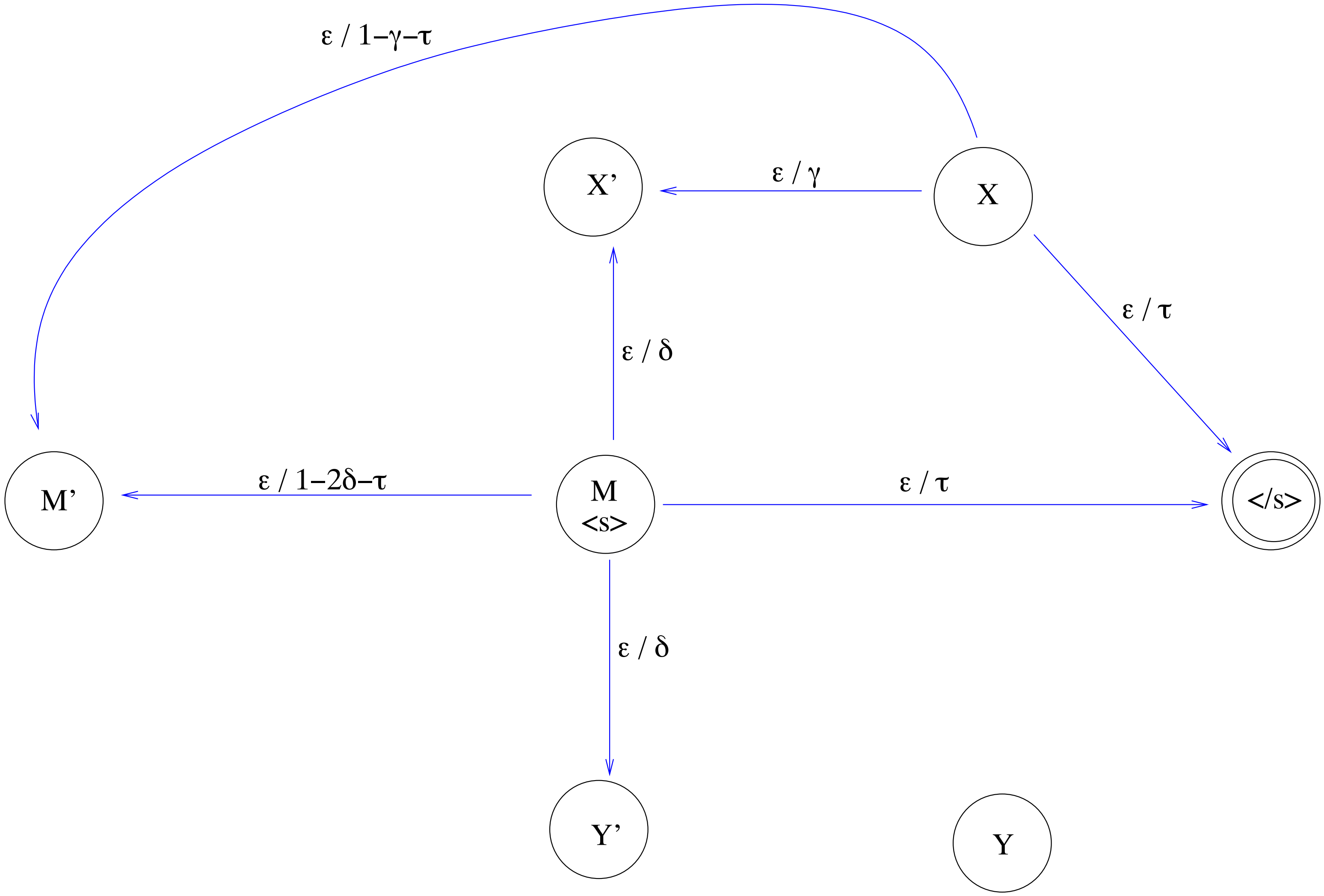
Affine gap HMM transducer states



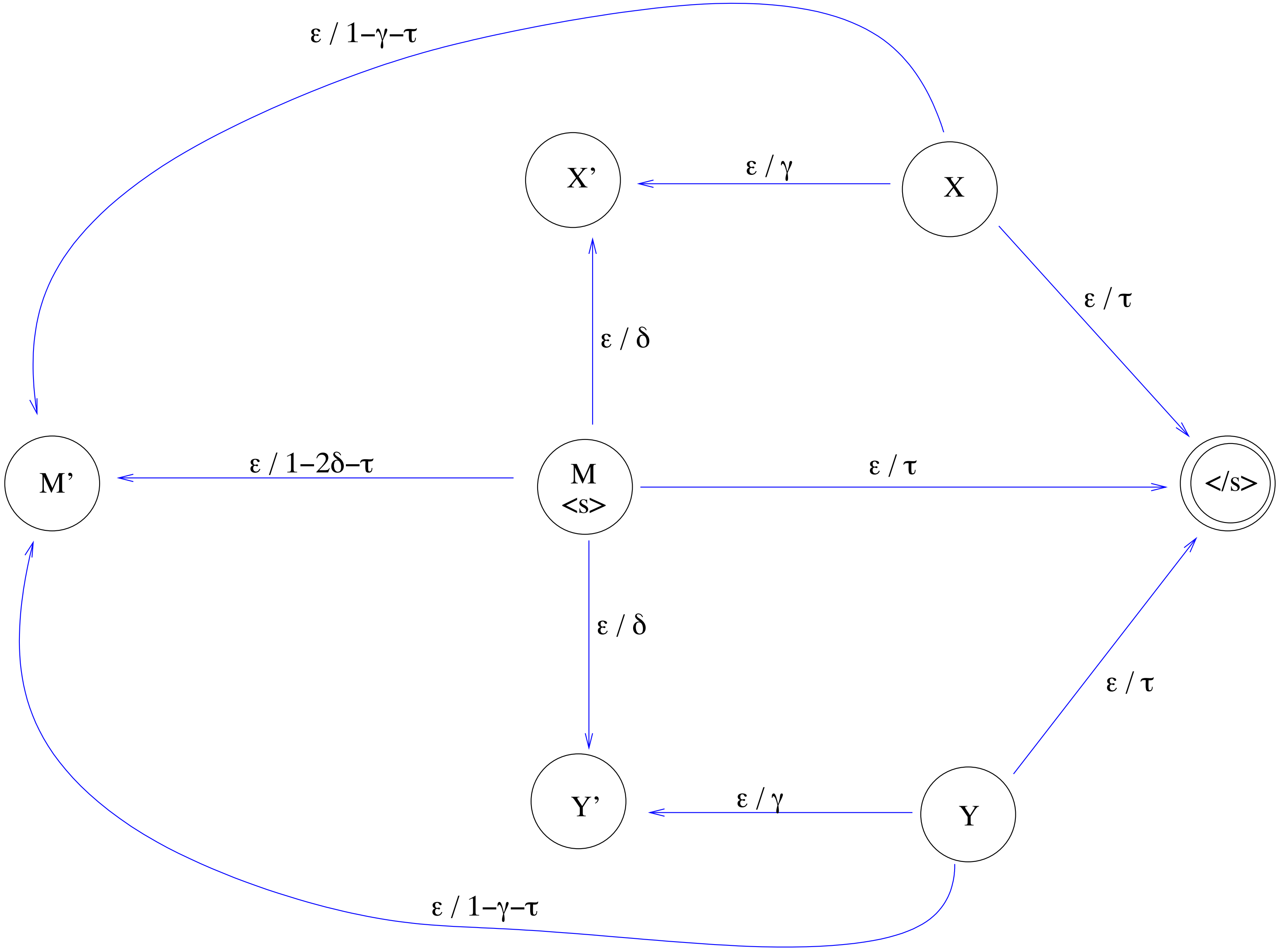
Affine gap HMM transducer states



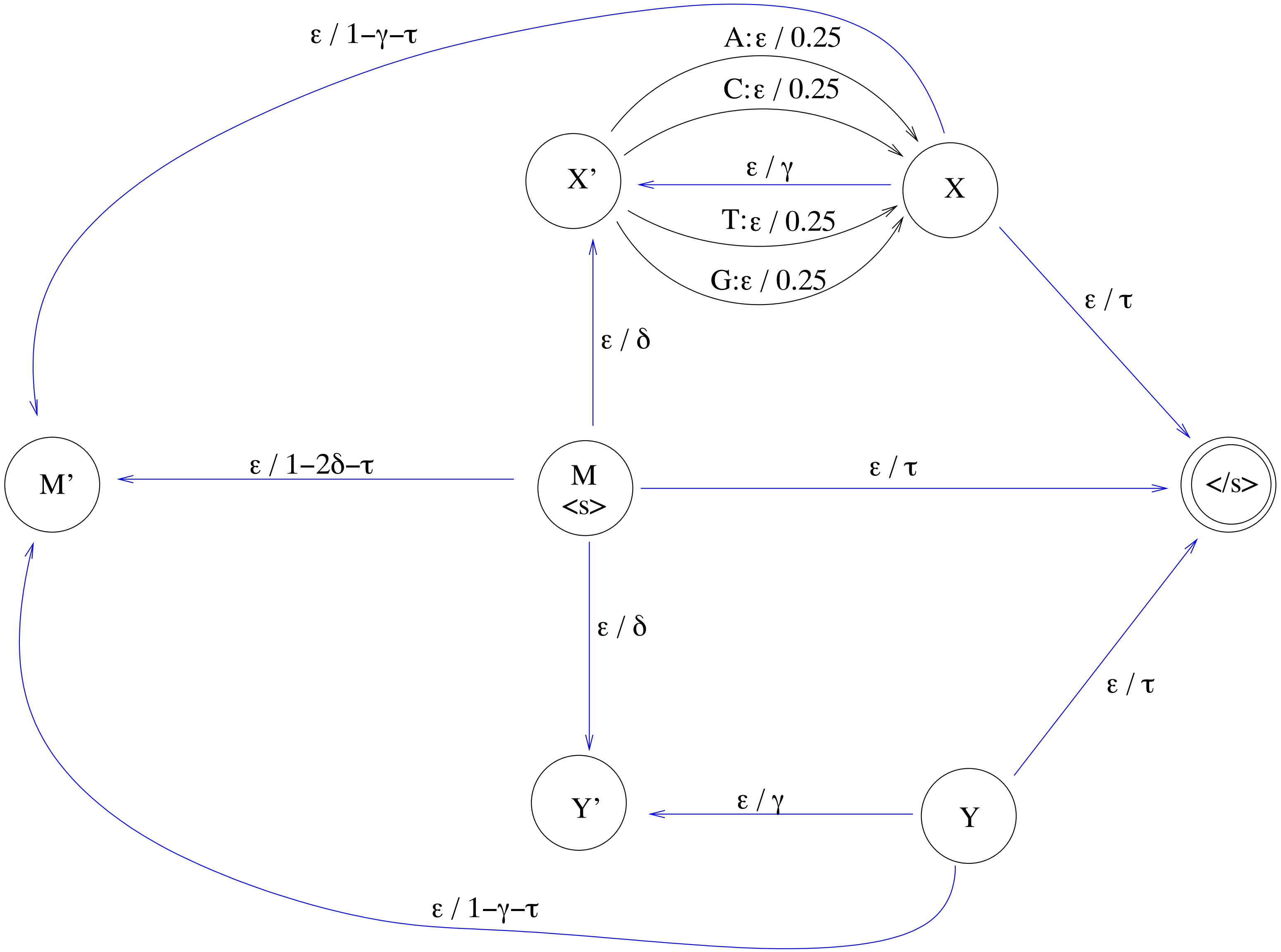
Affine gap HMM transducer states



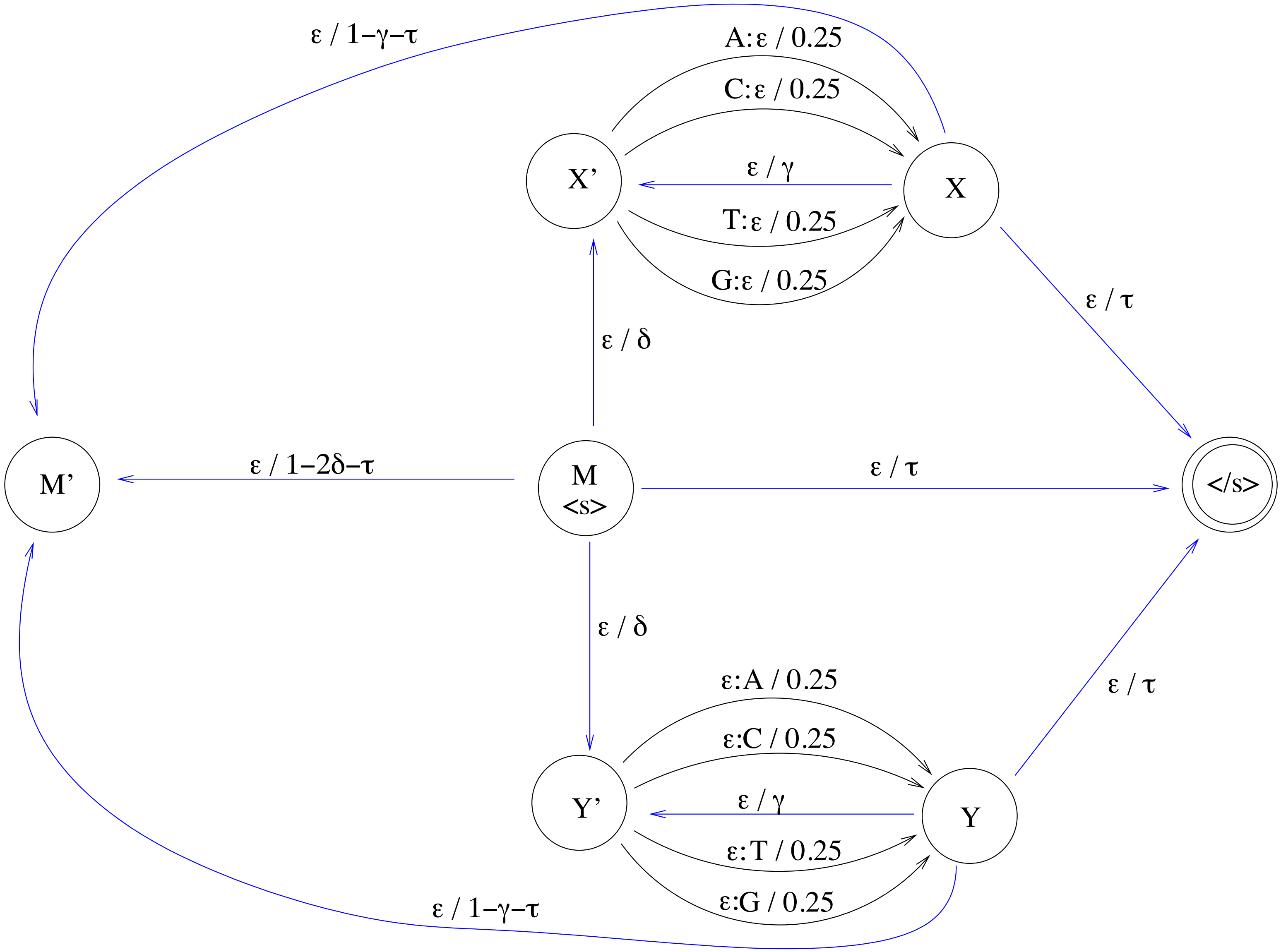
Affine gap HMM transducer states



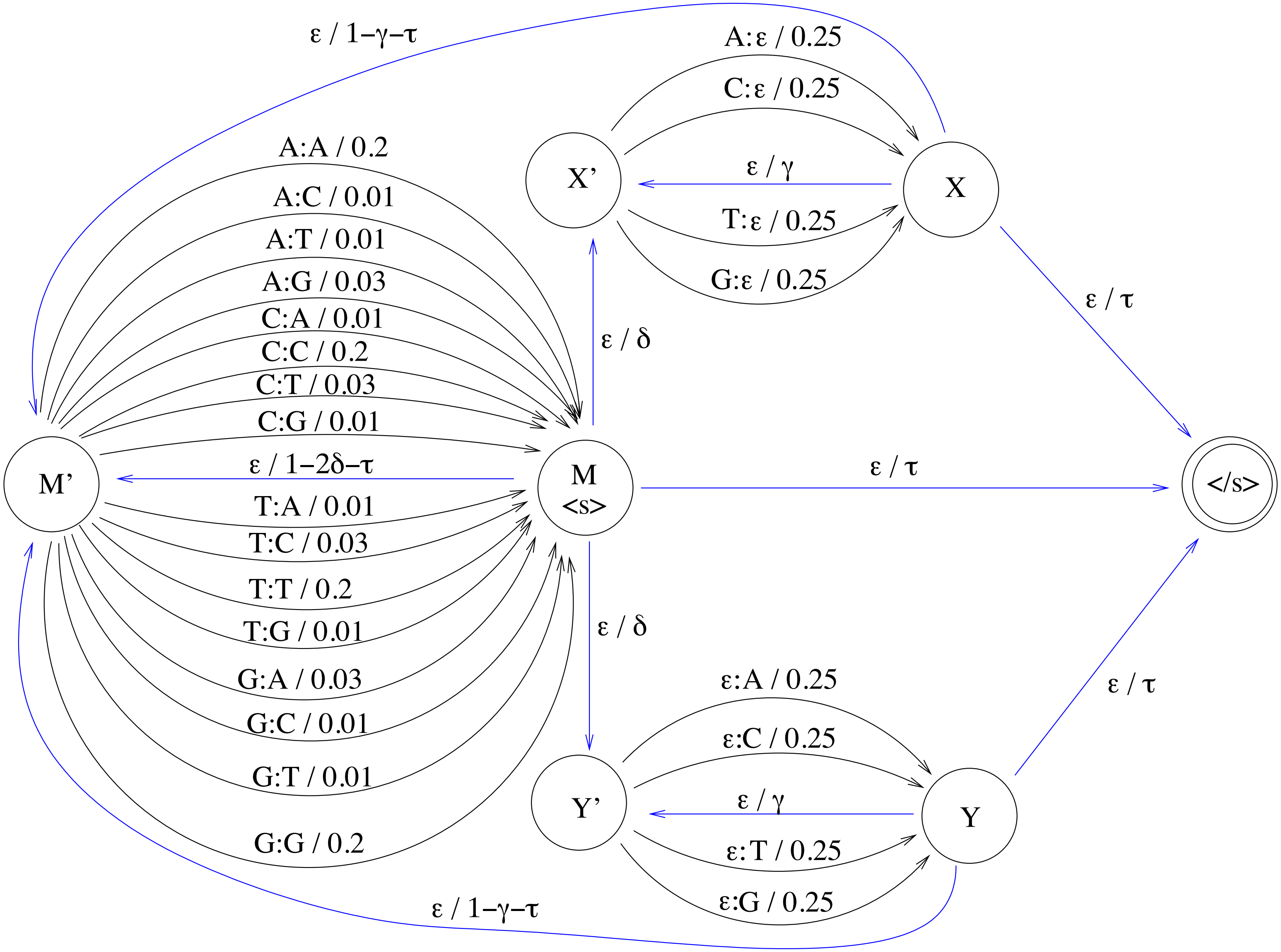
Affine gap HMM transducer states



Affine gap HMM transducer states



Affine gap HMM transducer



Alignment with transducers

- Composition: $\boldsymbol{X} \circ \boldsymbol{Y}$
  - Match ouptut labels in $\boldsymbol{X}$ with input labels in $\boldsymbol{Y}$
  - When matched, keep input in $\boldsymbol{X}$ and output in $\boldsymbol{Y}$
  - Multiply probabilities (in the real semiring)
  - States in resulting transducer represent pairs of states, one from $\boldsymbol{X}$ and one from $\boldsymbol{Y}$
  - Both must be final for the resulting state to be final
- One key complication: $\boldsymbol{\epsilon}$
  - Advisable to use an *epsilon filter*

Learning alignment models

- Some insertions, deletions and substitutions are more likely than others
  - A/G and C/T form functional pairs more likely to substitute
  - In spoken language, some sounds are more likely to be inserted, others more likely to be deleted
- How can we go about learning to better predict such patterns?
- Answer: start with a model, use EM to improve model
- As with a tagging task, HMM alignment model can be used with forward-backward and EM

Ristad and Yianilos (1997)

- Each word in a speech recognition system has a canonical pronunciation (or three)
  - e.g., nuclear: N UW K L IY ER
- Actual utterances may depart from this
  - e.g., Bush: N UW K Y AA L ER
  - or New York: N UW K L IY AH
- May want to learn common edits from canonical pronunciations
- Ristad and Yianilos show that trained edit distance is far superior to standard Levenshtein distance

Ristad and Yianilos (1997) task

- Training and testing corpus of utterances phonetically labeled
- Given
  - Pronunciation lexicon with canonical pronunciations
  - Alignment model
- Find the word string that best matches phonetic label string
- With HMM alignment models, can use EM to re-train alignment model
  - For this paper, they used 10 iterations

Basic findings

- Levenshtein distance not particularly effective
- Generally untied parameters were best
  - With large amount of training and small vocabulary, usually enough observations for parameters
- In at least one scenario parameter tying was helpful
- Reached reasonable performance
  - Probably could get even better performance with more states in HMM model

Larger state space

- Consider earlier motivating example:

  N UW K L IY ER → N UW K Y AA L ER

- Insertion of ‘Y AA’ probably has a lot to do with having ‘K’ and ‘L’ together
- Encoding the local context in the HMM alignment model states will do a better job of capturing such regularities
- With more states and transitions, fewer observations per parameter
  - Sparse data: parameter tying probably a good idea