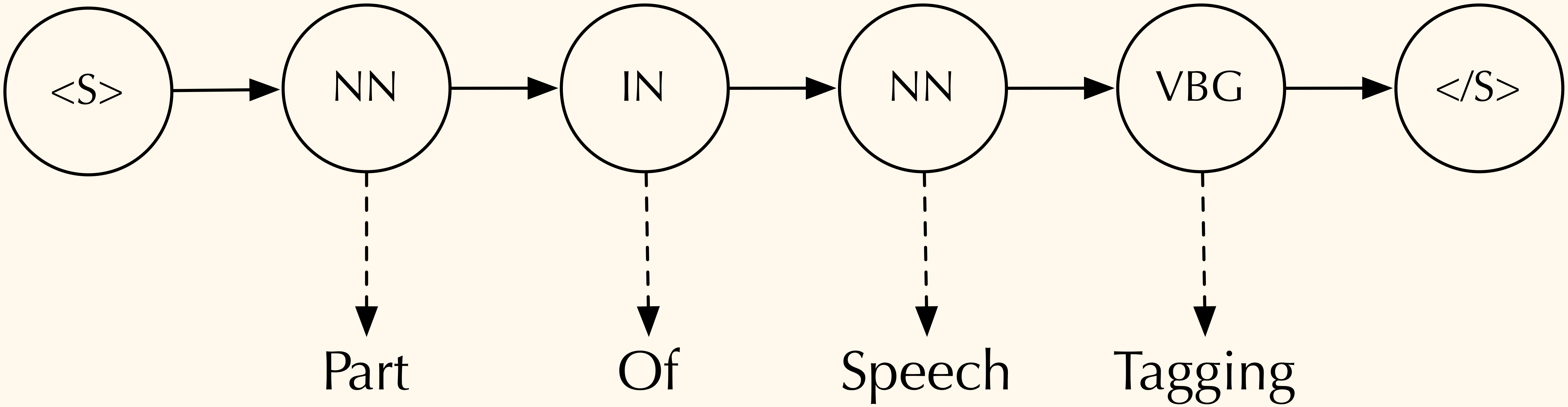


Discrete HMMs, part 2: Viterbi!

Steven Bedrick
CS/EE 5/655, 10/27/14

Quick Review:

Stochastic PoS techniques rely entirely on probability.

The goal of a stochastic PoS tagger is to find:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

Via Bayes' Rule:

$$\hat{t}_1^n = \arg\max_{t_1^n} \underbrace{P(w_1^n | t_1^n)}_{\text{Likelihood}} \underbrace{P(t_1^n)}_{\text{Prior}}$$

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

Via Bayes' Rule:

$$\hat{t}_1^n = \arg\max_{t_1^n} \underbrace{P(w_1^n | t_1^n)}_{\text{Likelihood}} \underbrace{P(t_1^n)}_{\text{Prior}}$$

After making a few simplifying assumptions:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

Via Bayes' Rule:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n | t_1^n) P(t_1^n)$$

Likelihood Prior

After making a few simplifying assumptions:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n) \approx \arg\max_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n)$$

Via Bayes′ Rule:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n | t_1^n) P(t_1^n)$$

Likelihood Prior

After making a few simplifying assumptions:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n | w_1^n) \approx \arg\max_{t_1^n} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1})$$

Probability of
word given tag

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n|w_1^n)$$

Via Bayes' Rule:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(w_1^n|t_1^n)P(t_1^n)$$

Likelihood Prior

After making a few simplifying assumptions:

$$\hat{t}_1^n = \arg\max_{t_1^n} P(t_1^n|w_1^n) \approx \arg\max_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

Probability of word given tag

Probability of tag given previous tag

скрытая марковская модель

Andrei Andreievich Markov
1856–1922

HMMs are a type of stochastic model used to examine sequential data.

The basic idea: there are two parameters changing over time, but we can only directly observe one of them. We want to know about the other.

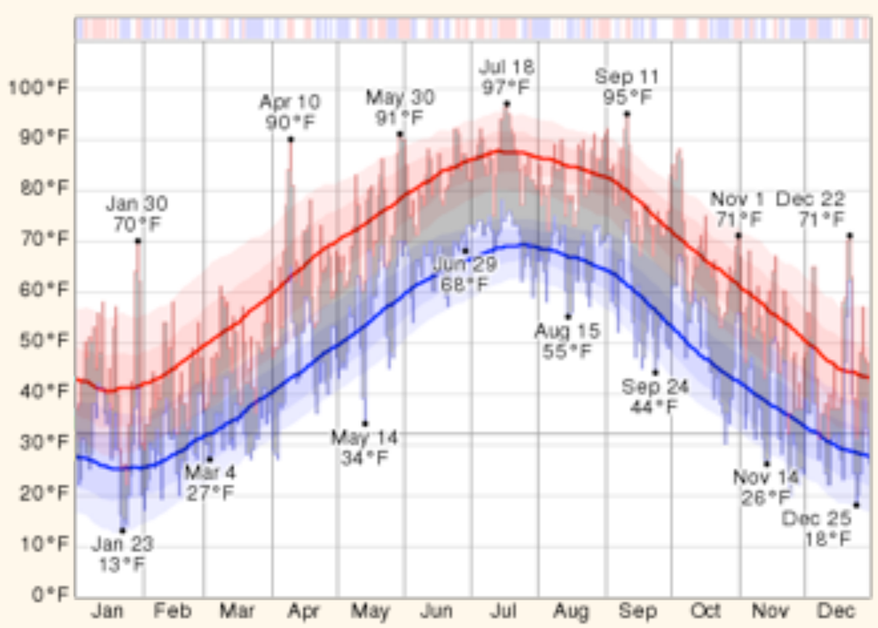
Q: { *Hot, Cold* }

A:

	Hot	Cold
Hot	0.7	0.3
Cold	0.6	0.4

B:

	1	2	3
Hot	0.2	0.4	0.4
Cold	0.5	0.4	0.1



$a_{0,Hot/Cold}$:

	Start
Hot	0.8
Cold	0.2

Note: for this demonstration, we are ignoring a_F.

There are three fundamental kinds of questions that we can ask with an HMM:

1. *Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

There are three fundamental kinds of questions that we can ask with an HMM:

1. ***Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?**

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

Let's say we have a sequence of diary entries:

$$O = 3, 1, 3$$

How likely is this sequence given the model described earlier? $P(O|\lambda)$

The *forward algorithm* uses dynamic programming to calculate this efficiently…

The *forward algorithm* uses dynamic programming to calculate this efficiently…

… by building a trellis α such that each $\alpha_t(j)$ represents:

The *forward algorithm* uses dynamic programming to calculate this efficiently...

... by building a trellis α such that each $\alpha_t(j)$ represents:

the probability of the machine being in state *j*,

The *forward algorithm* uses dynamic programming to calculate this efficiently...

... by building a trellis α such that each $\alpha_t(j)$ represents:

the probability of the machine being in state *j*,

... given the first t observations ("forward probability").

The *forward algorithm* uses dynamic programming to calculate this efficiently...

... by building a trellis α such that each $\alpha_t(j)$ represents:

the probability of the machine being in state *j*,

... given the first t observations ("forward probability").

Formally: $\alpha_t(j) = P(o_1, o_2 ... o_t, q_t = j | \lambda)$

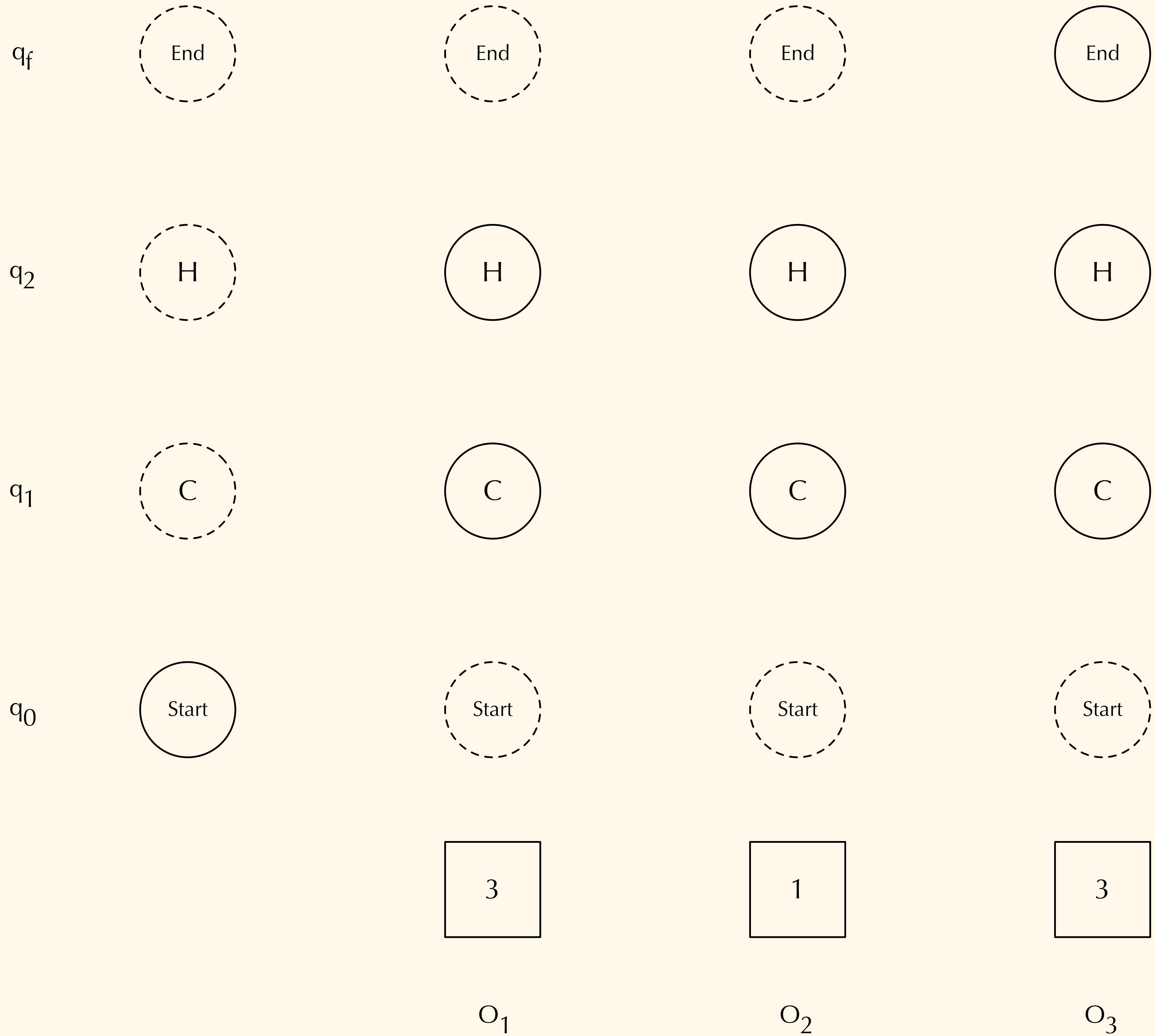
Calculating $\alpha_t(j) = P(o_1, o_2 ... o_t, q_t = j | \lambda)$ is fairly straightforward:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

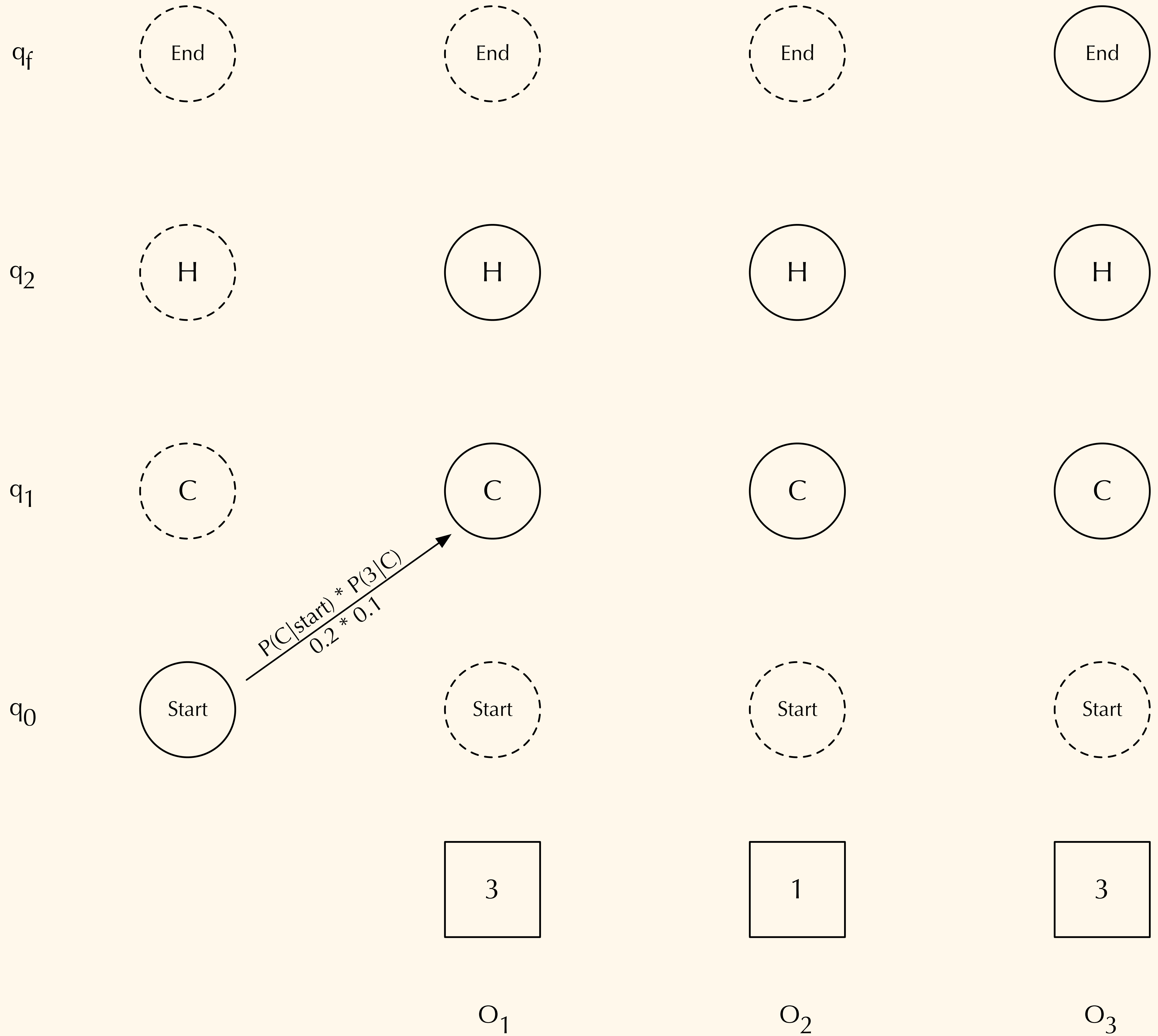
Previous time step's forward probability for state *i*

Emission likelihood for symbol o_t given current state *j*

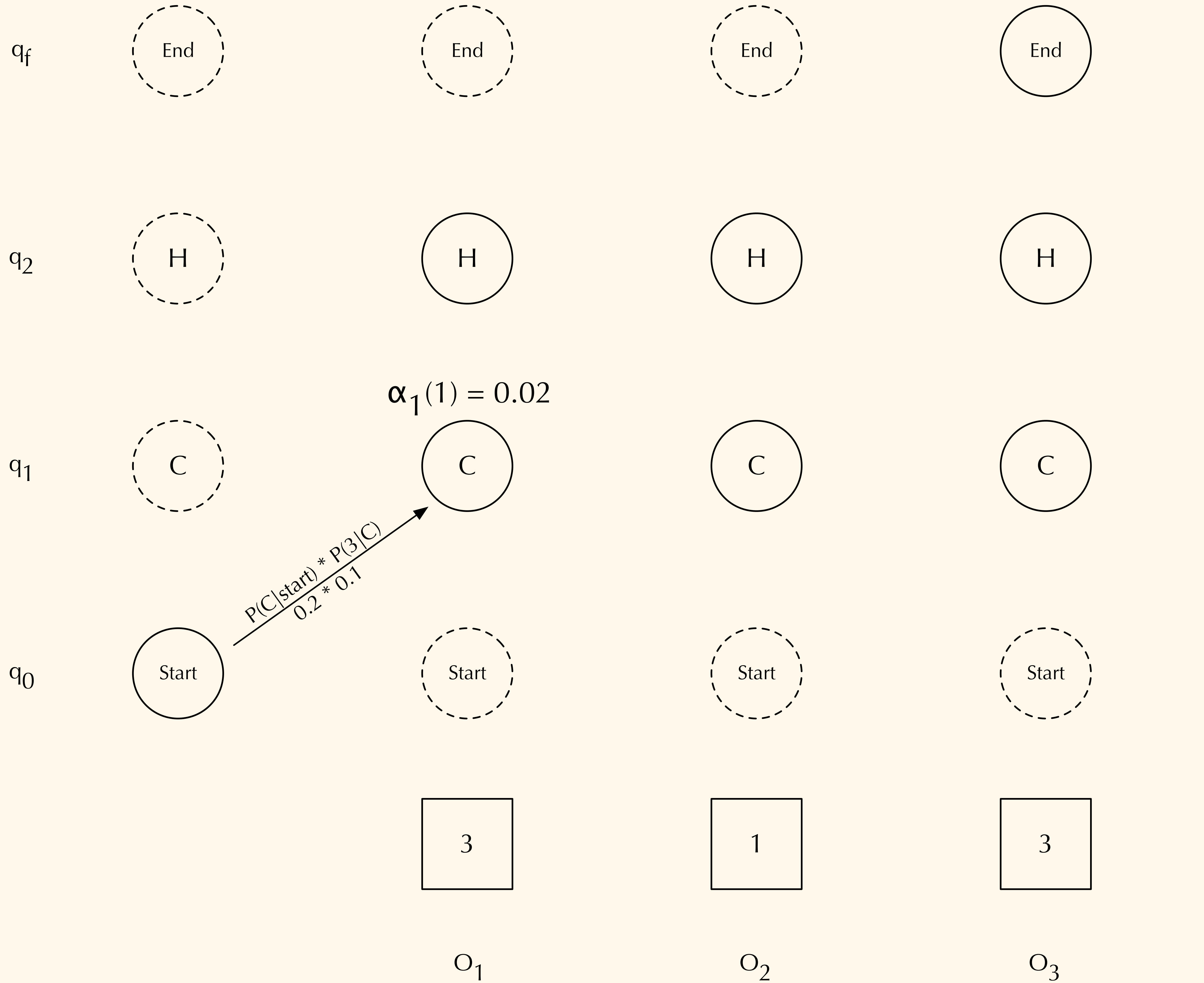
Transition prob. from previous state *i* to current state *j*



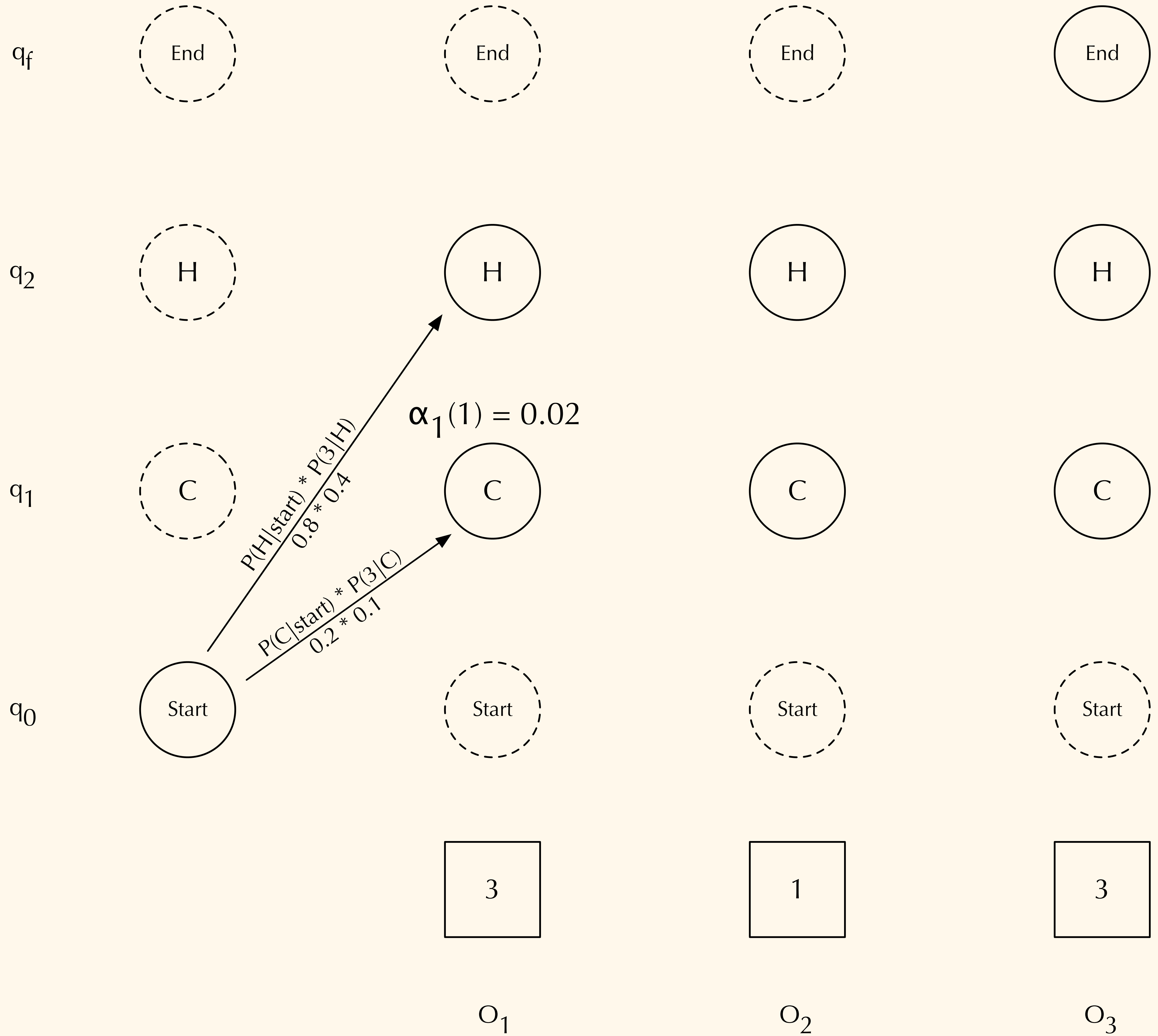
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



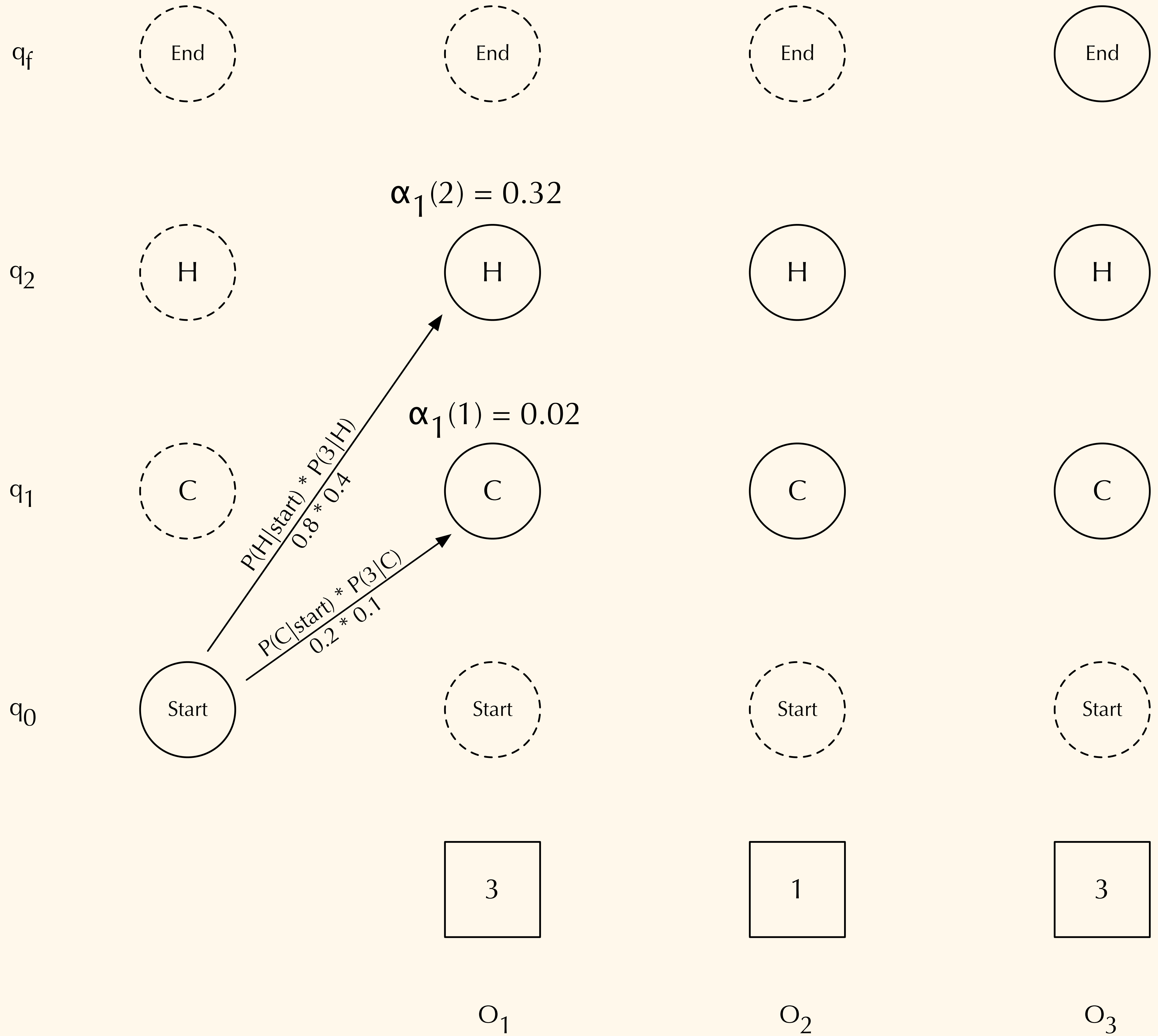
qf
End
End
End
End
q2
H
H
H
H
q1
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
q0
Start
Start
Start
Start
3
1
3
O1
O2
O3



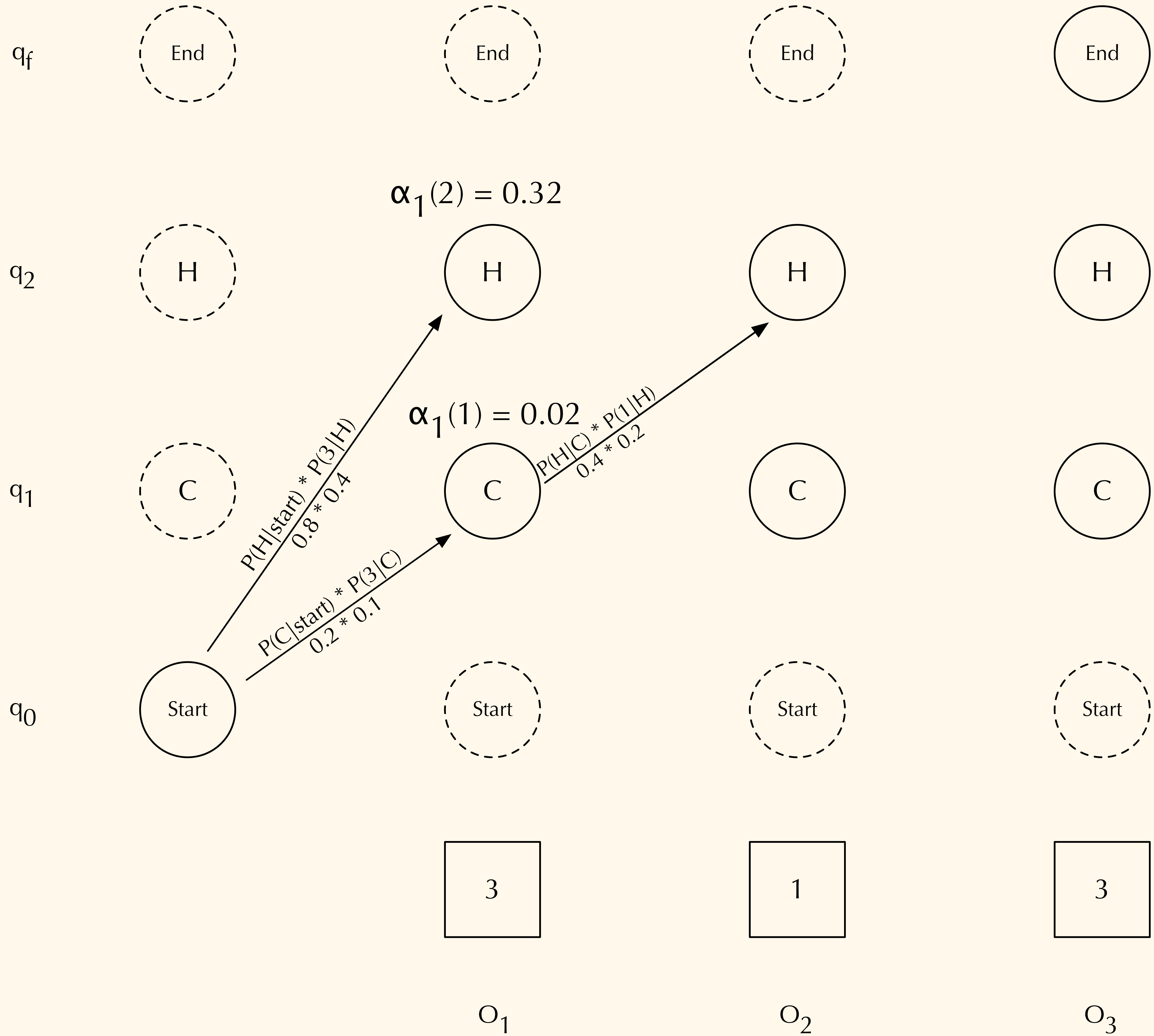
q_f
q_2
q_1
q_0
End
H
C
Start
α_1(1) = 0.02
P(C|start) * P(3|C)
0.2 * 0.1
3
1
3
O_1
O_2
O_3



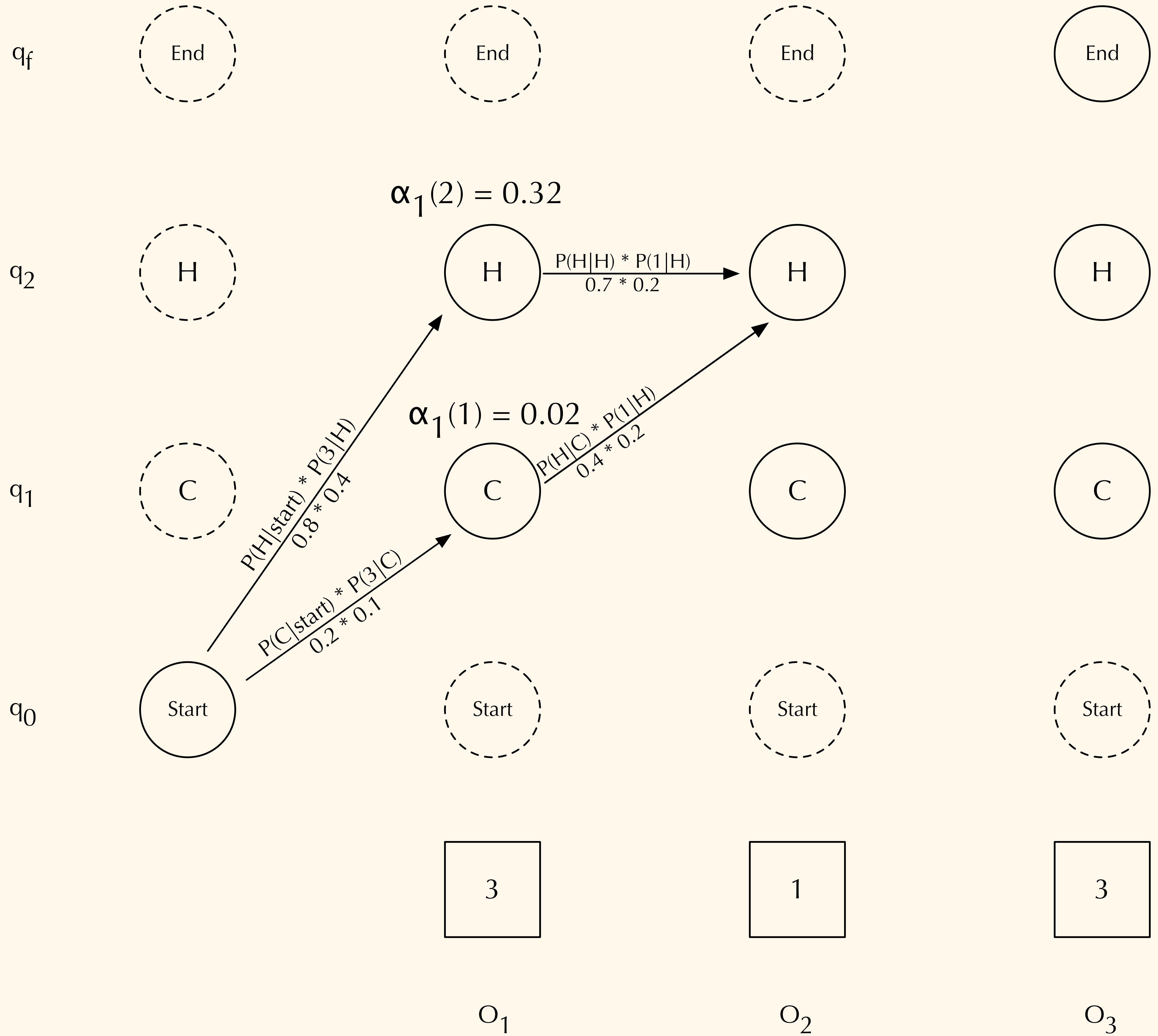
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
$\alpha_1(1) = 0.02$
3
1
3
o_1
o_2
o_3



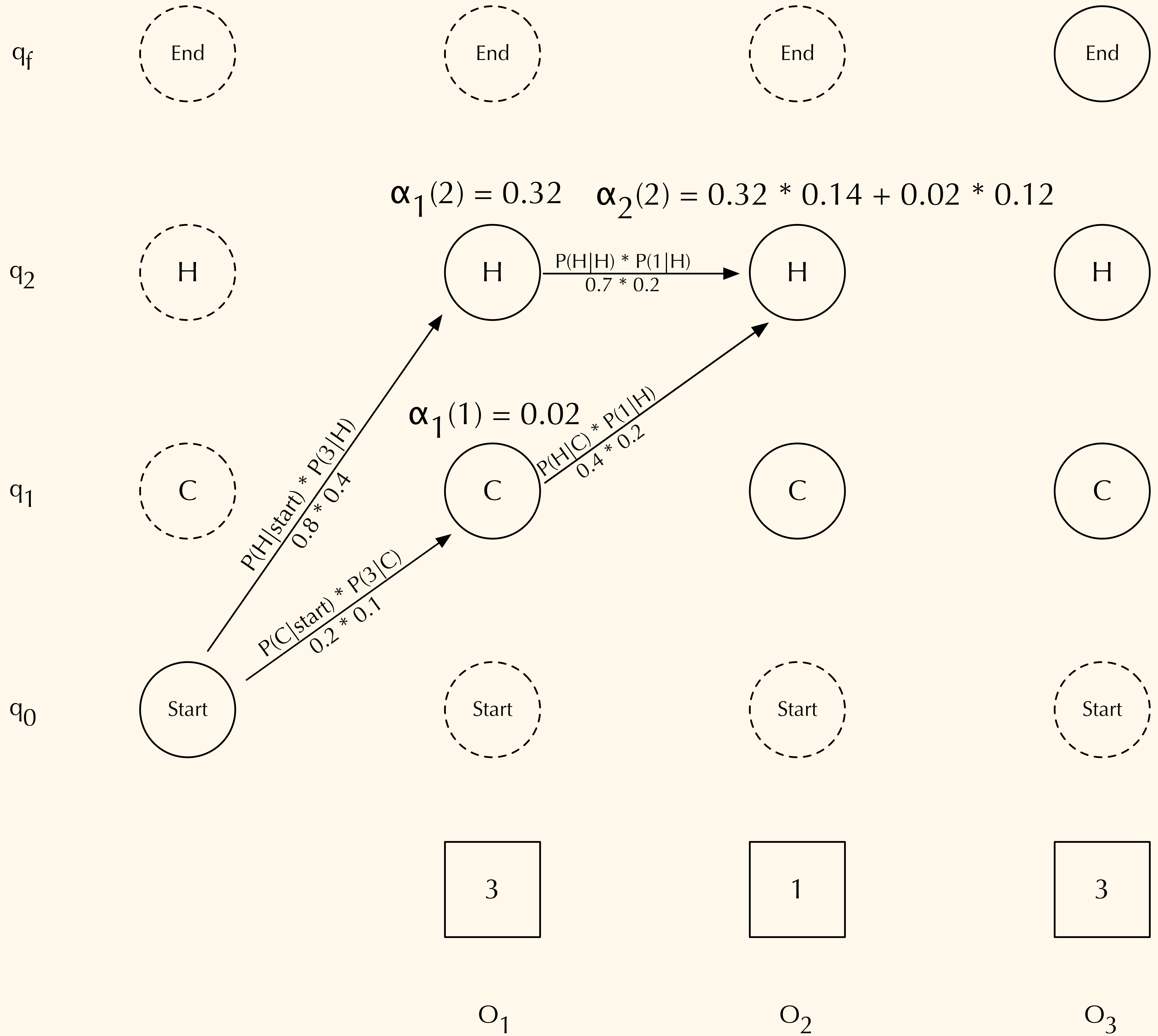
q_f
q_2
q_1
q_0
End
End
End
End
α_1(2) = 0.32
H
H
H
H
α_1(1) = 0.02
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

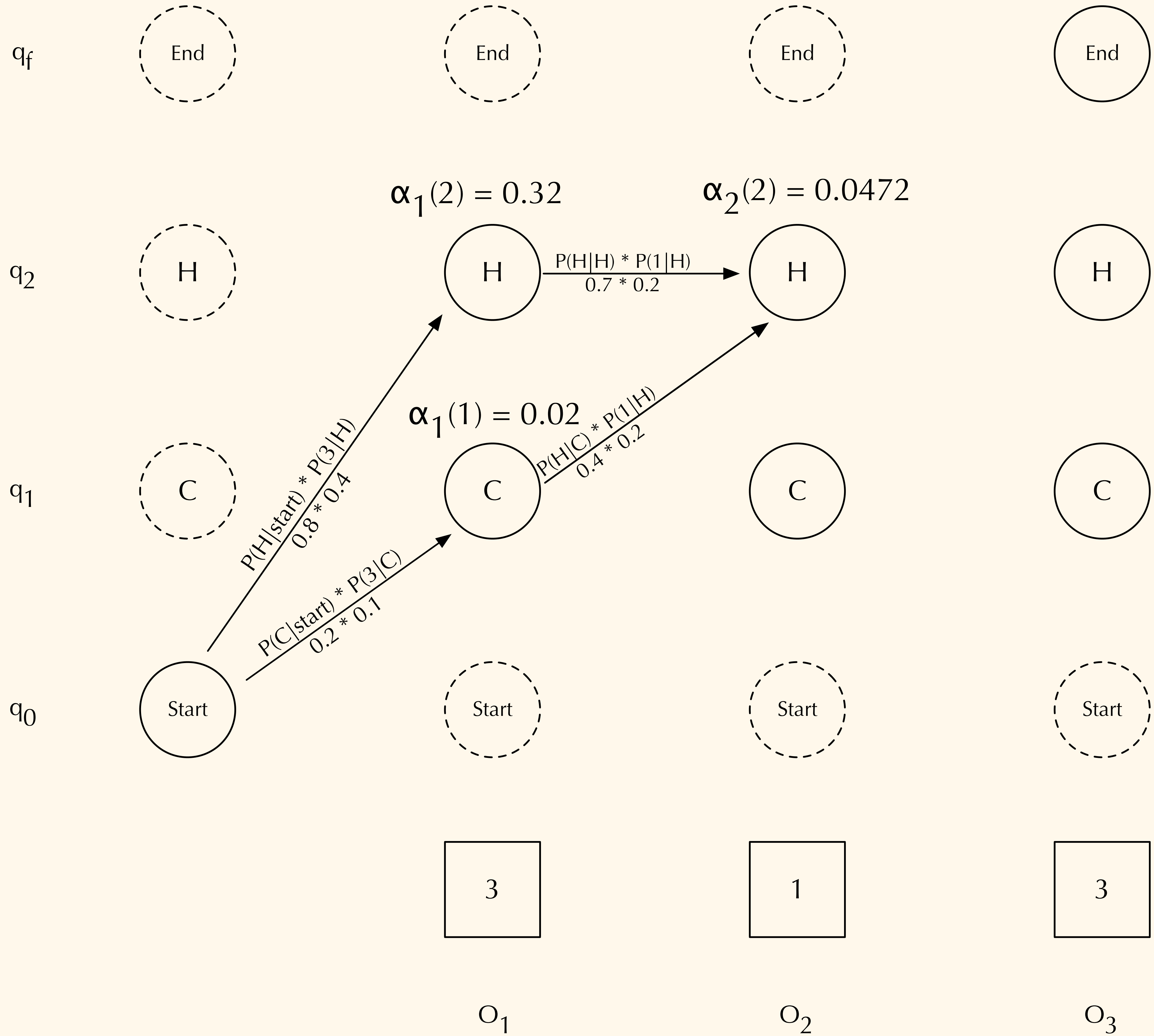


q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
H
H
H
H
$\alpha_1(1) = 0.02$
C
C
C
C
Start
Start
Start
Start
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
P(H|C) * P(1|H)
0.4 * 0.2
3
1
3
O_1
O_2
O_3

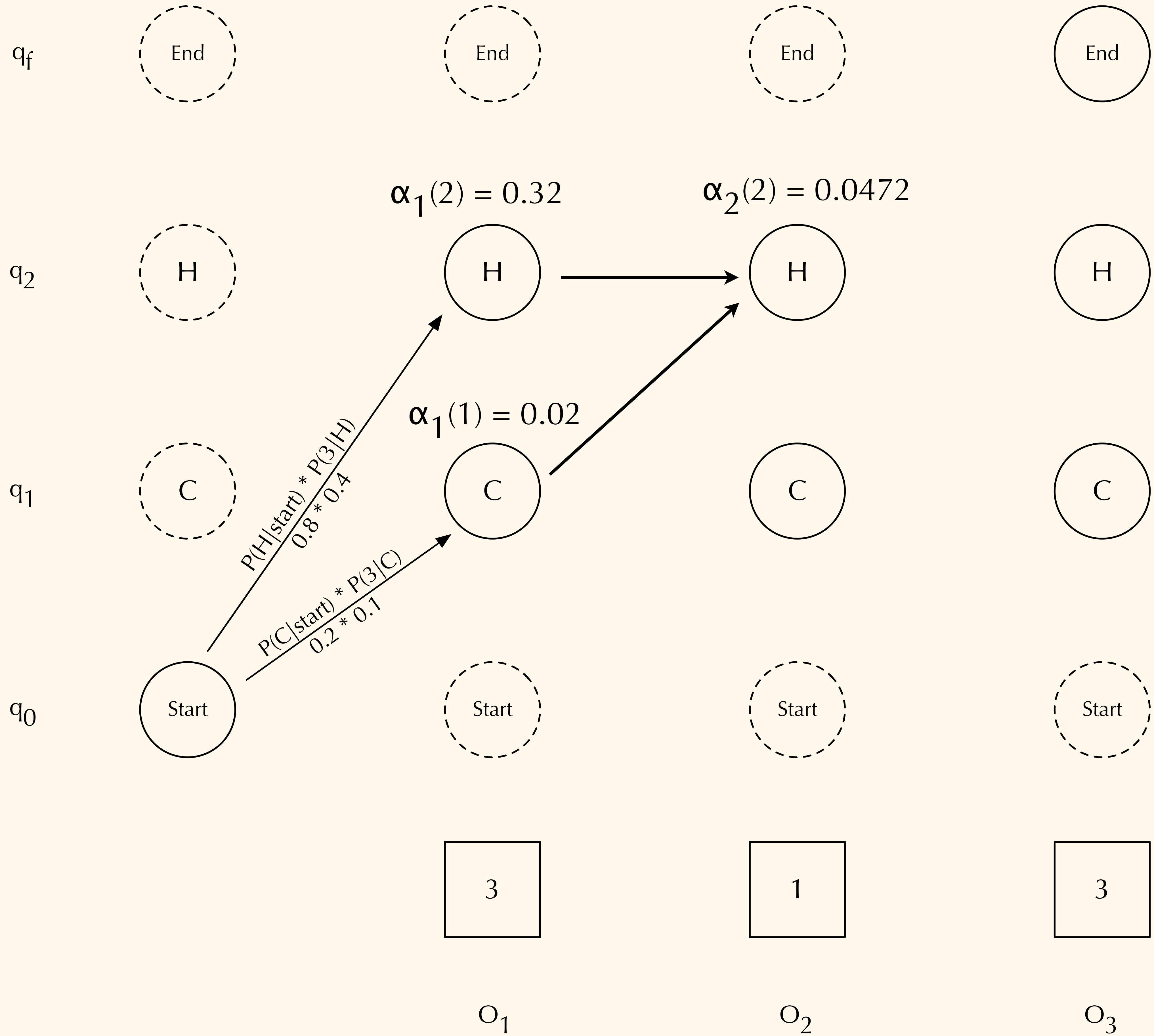


q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$\alpha_1(1) = 0.02$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
P(H|C) * P(1|H)
0.4 * 0.2
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3

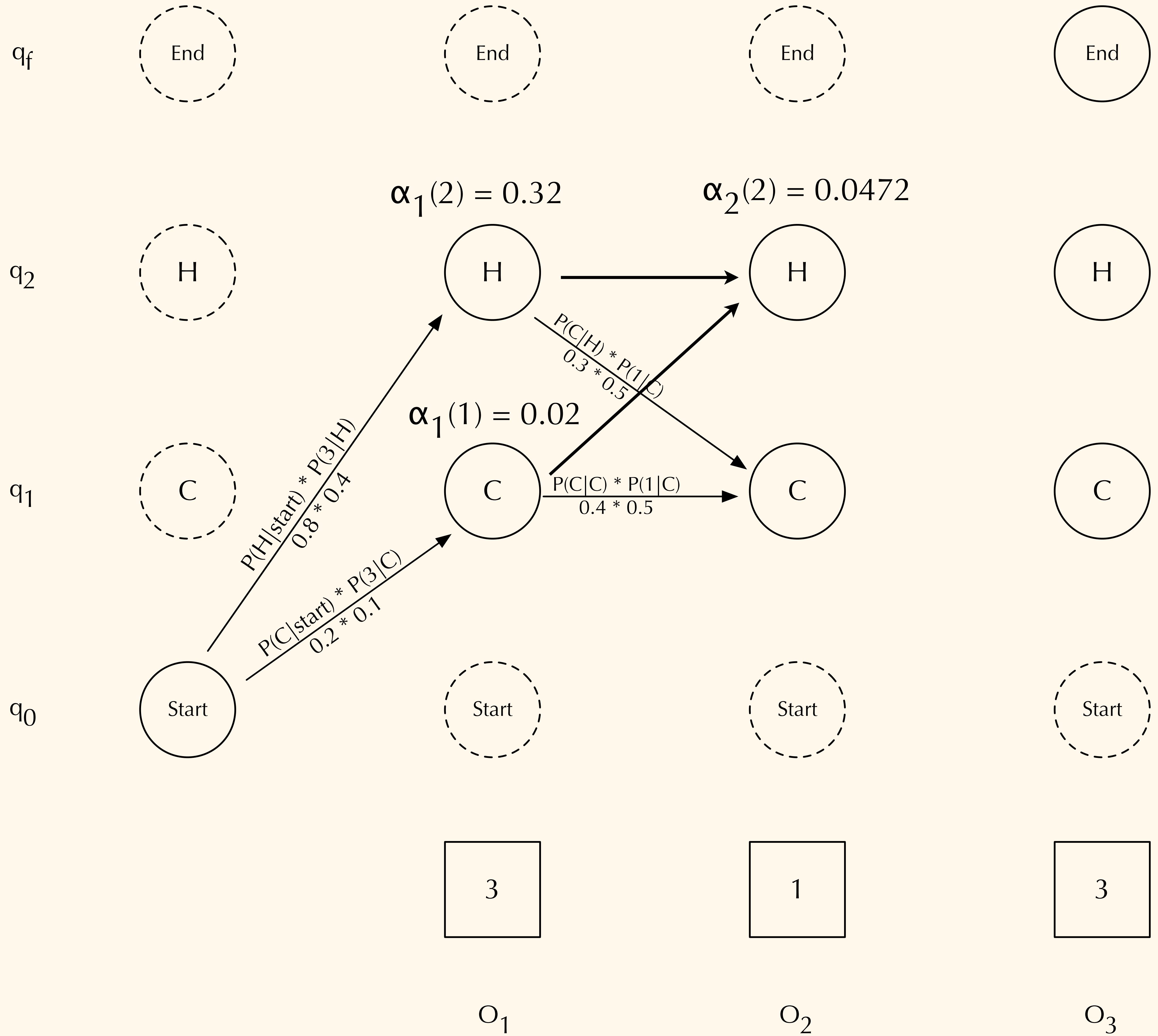

q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.32 * 0.14 + 0.02 * 0.12$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$\alpha_1(1) = 0.02$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(H|C) * P(1|H)
0.4 * 0.2
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



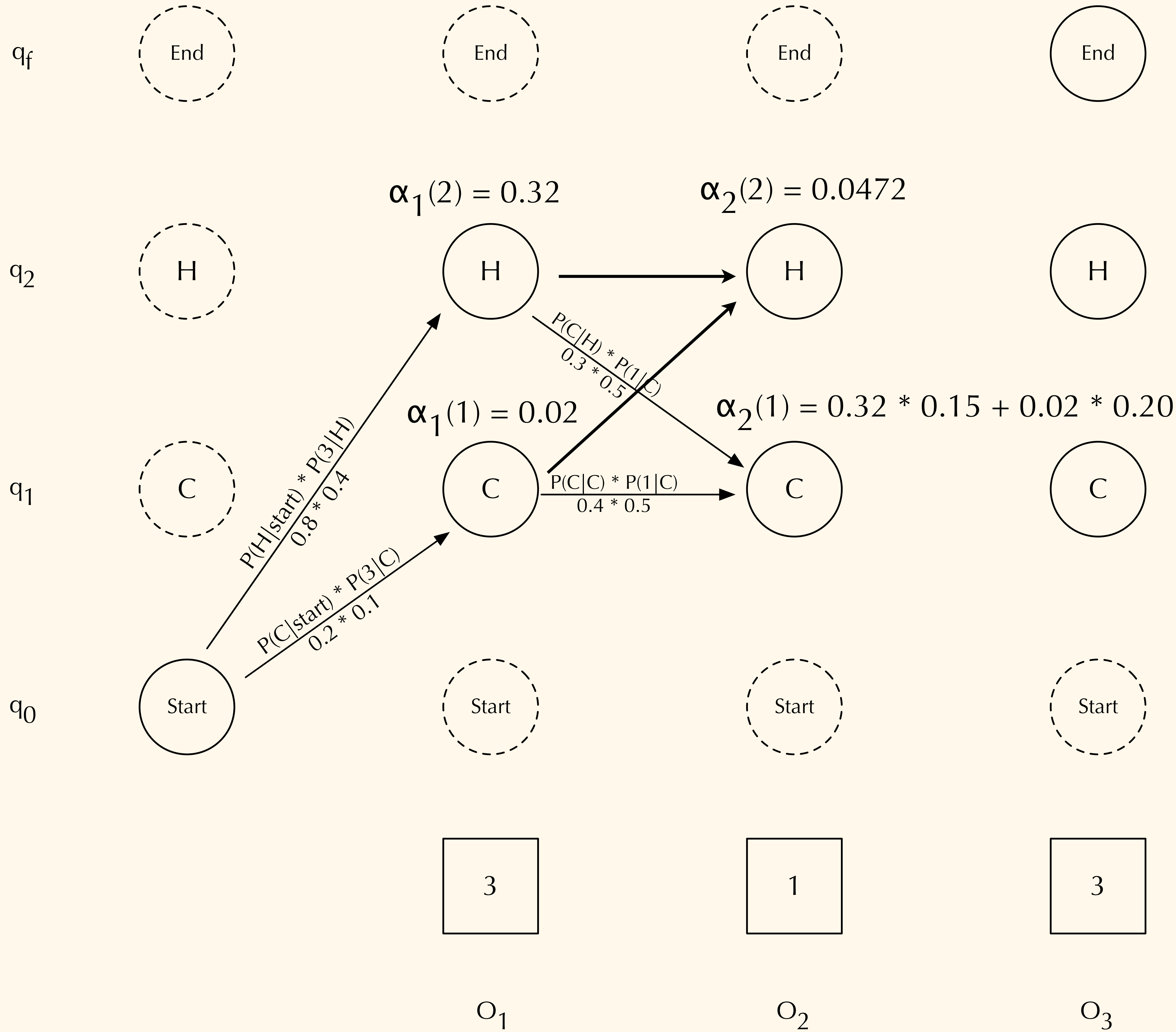
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$\alpha_1(1) = 0.02$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(H|C) * P(1|H)
0.4 * 0.2
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



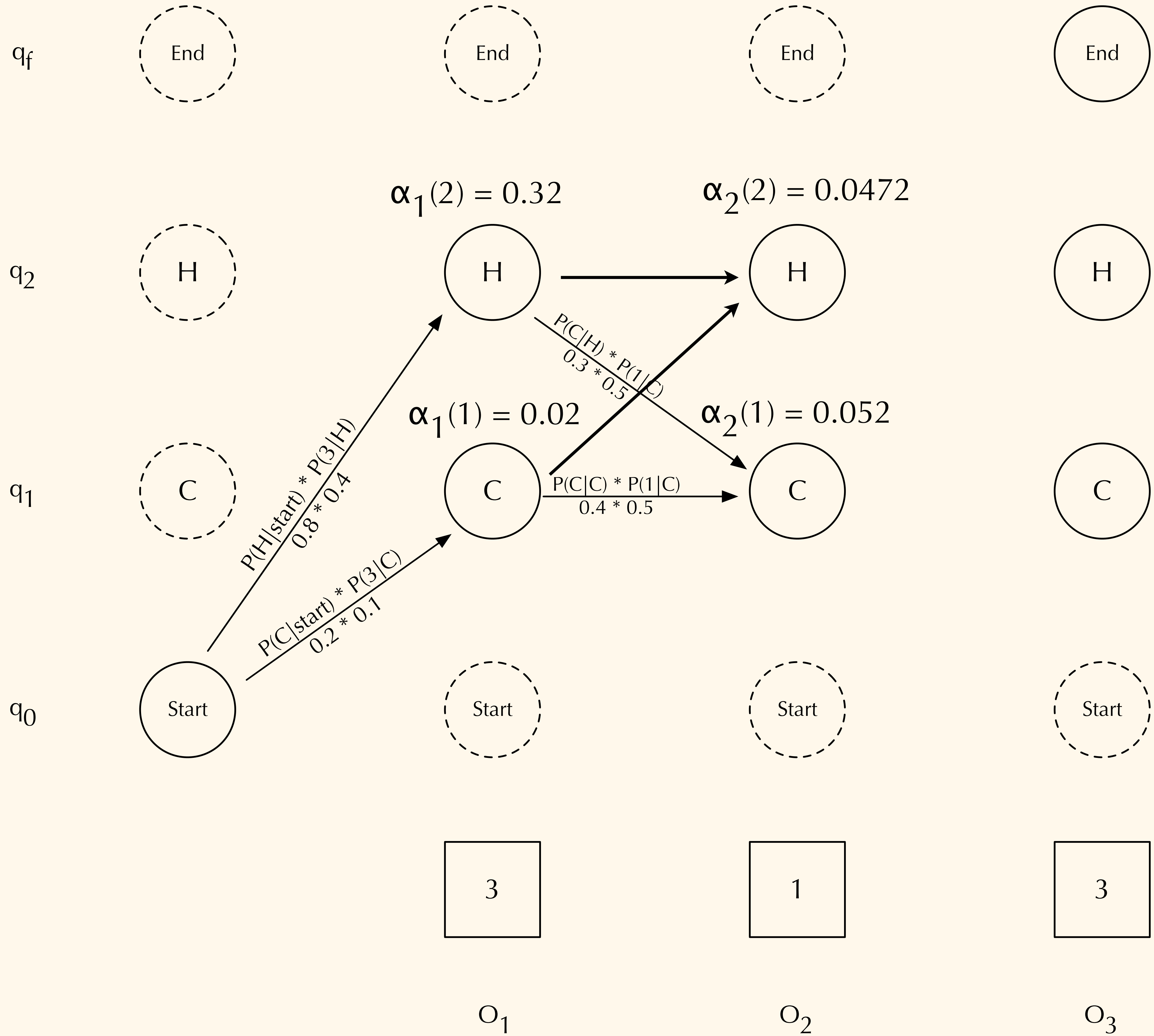
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
$\alpha_1(1) = 0.02$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



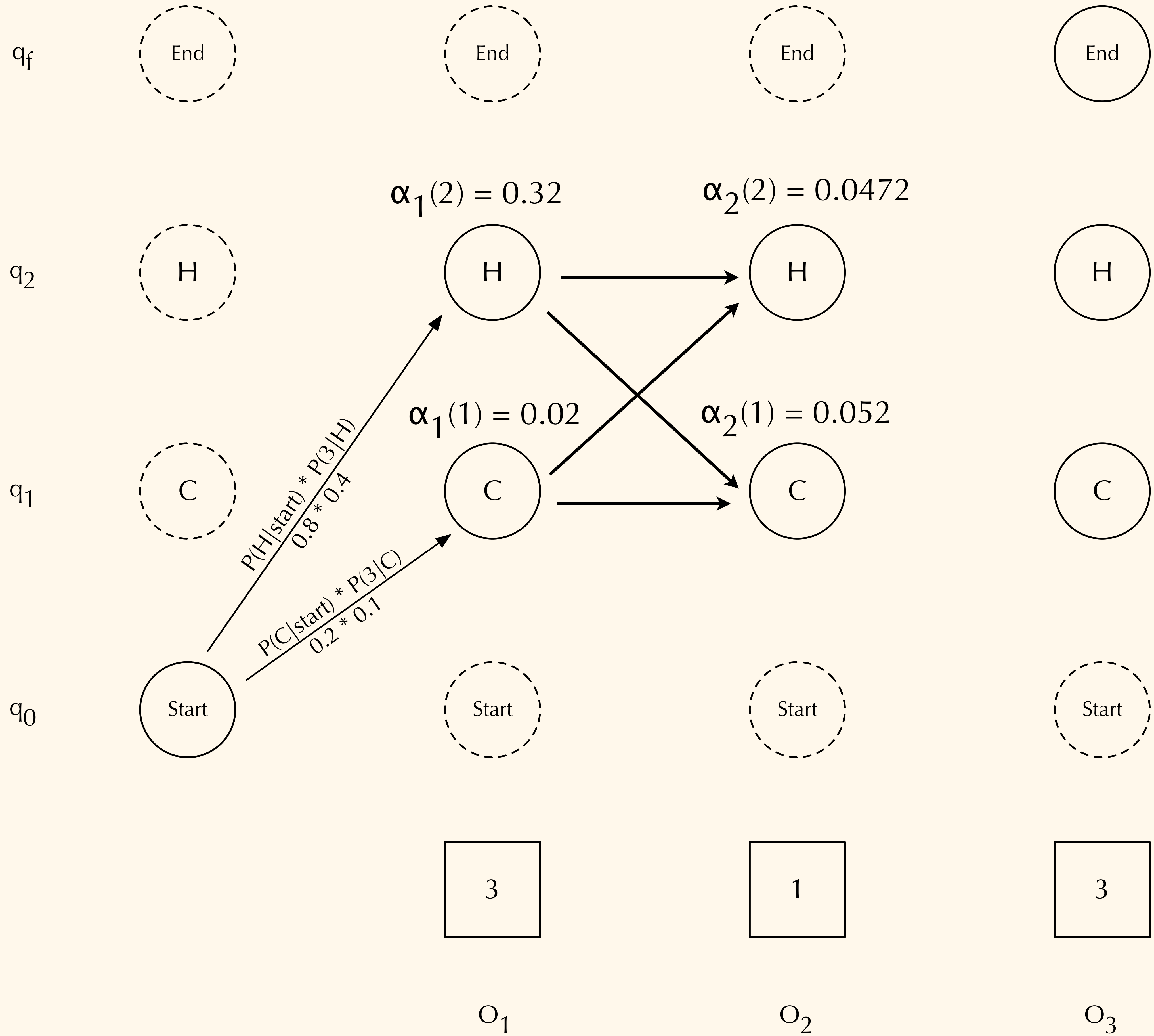
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
$\alpha_1(1) = 0.02$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|C) * P(1|C)
0.4 * 0.5
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



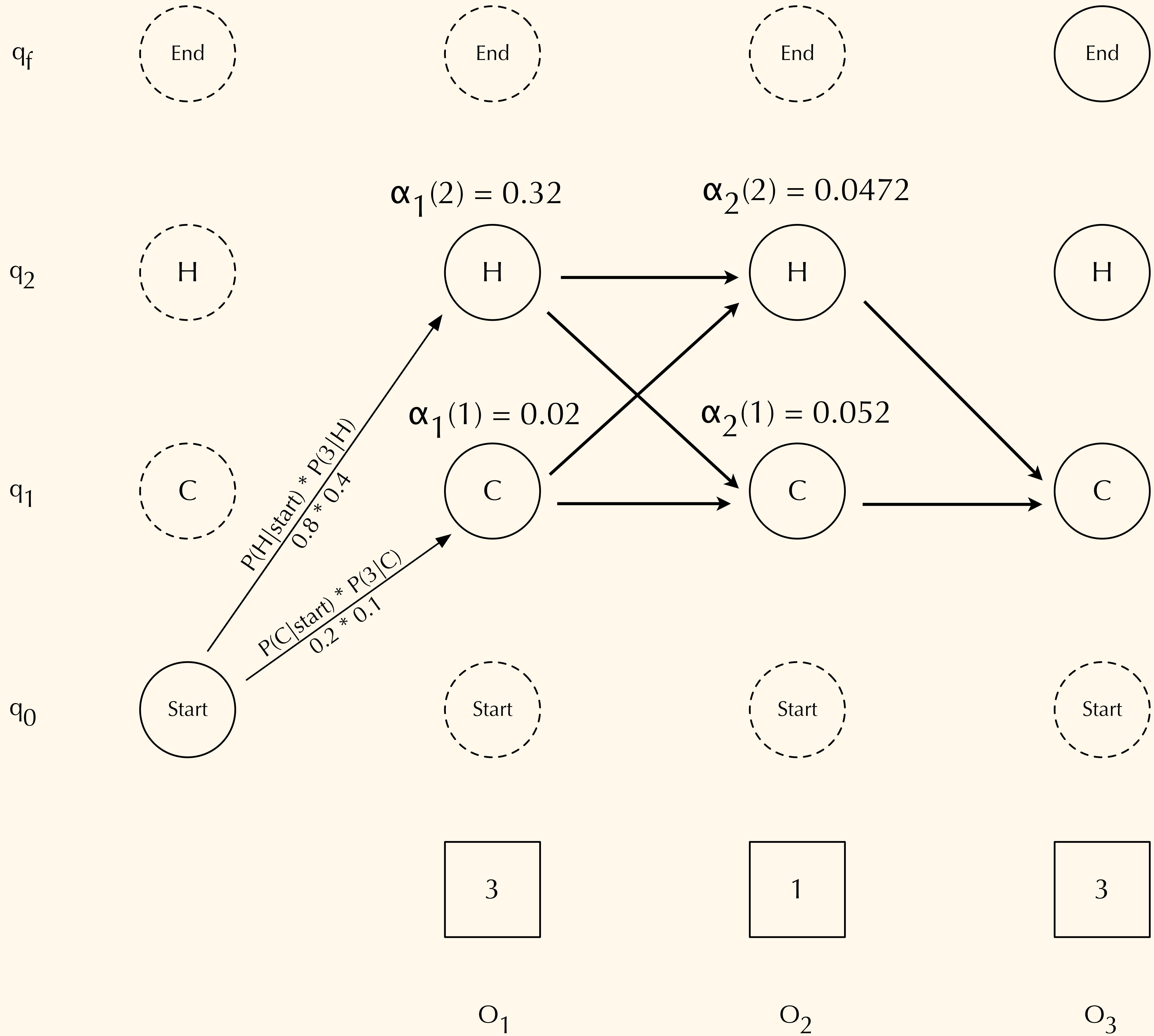
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.32 * 0.15 + 0.02 * 0.20$
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(1|C)
0.4 * 0.5
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



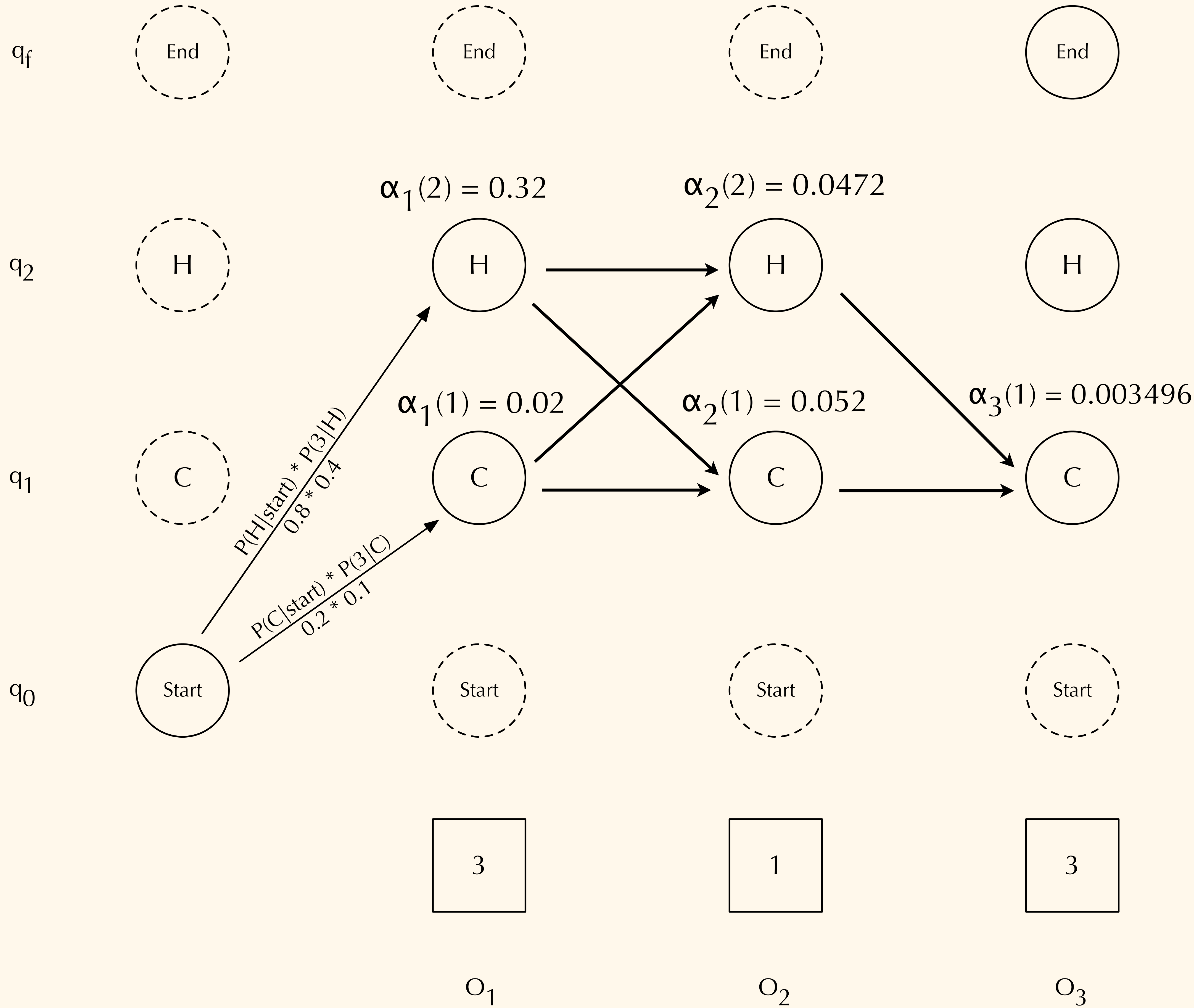
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(1|C)
0.4 * 0.5
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



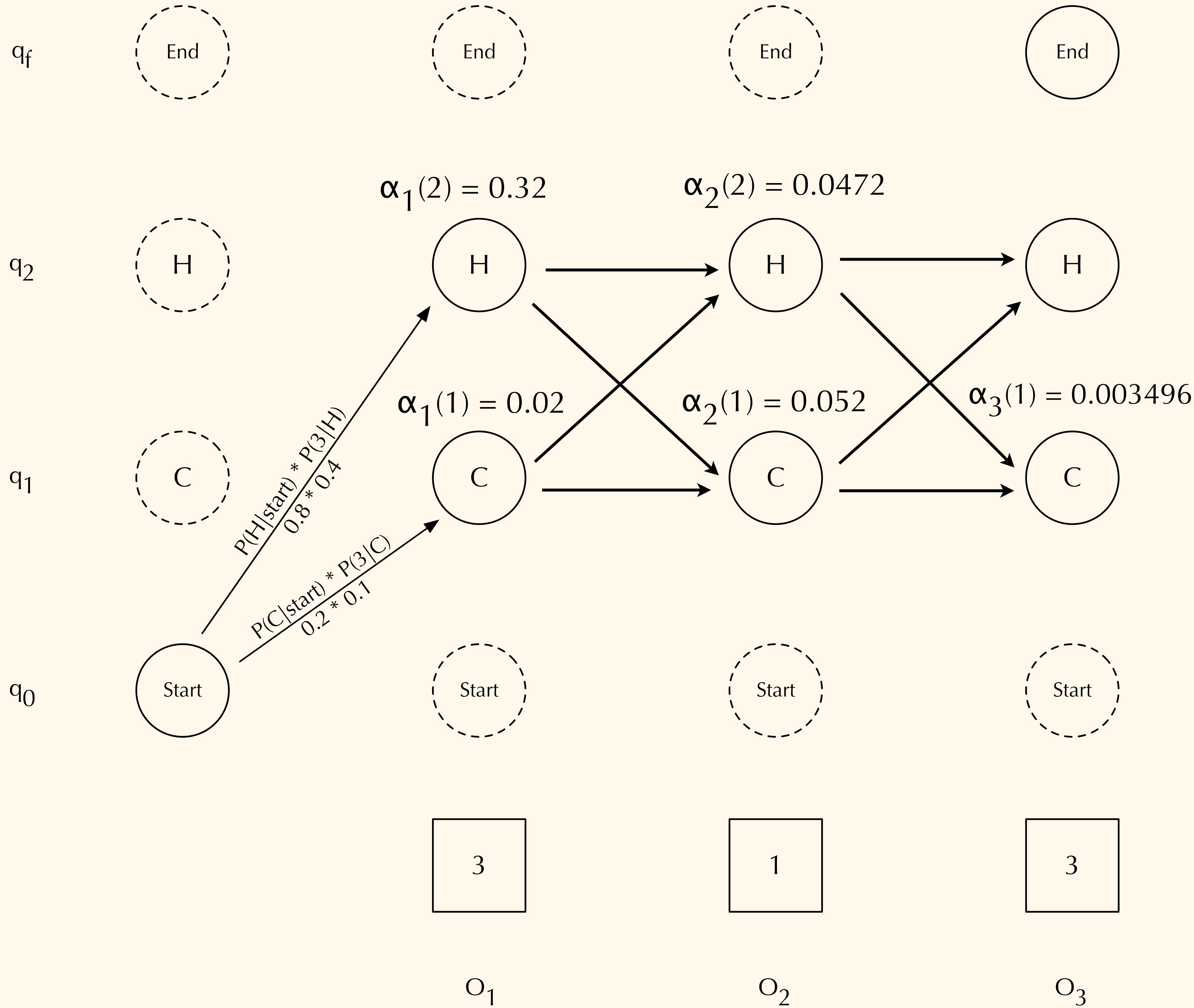
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



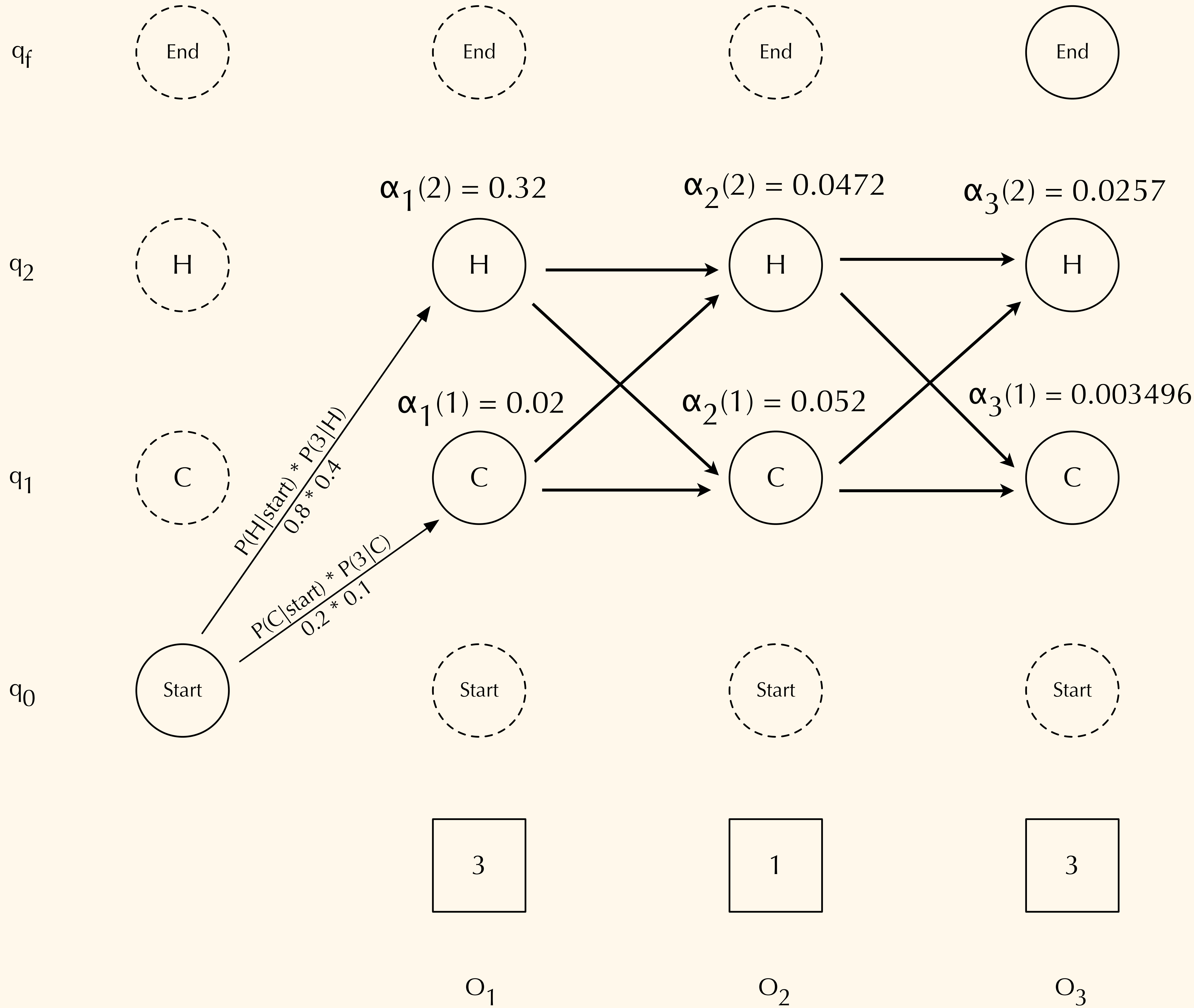
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



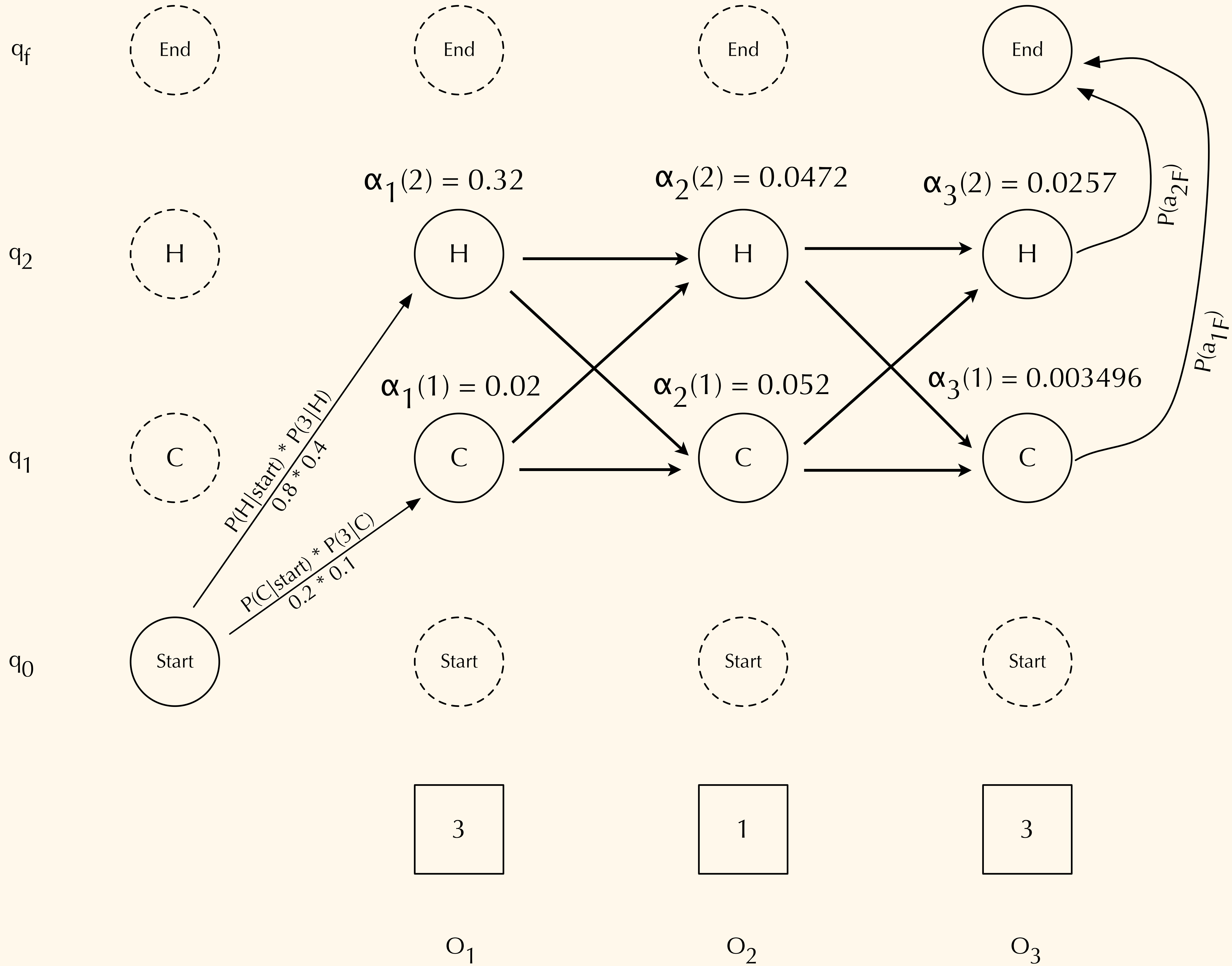
q_f
q_2
q_1
q_0
End
End
End
End
α_1(2) = 0.32
α_2(2) = 0.0472
H
H
H
H
α_1(1) = 0.02
α_2(1) = 0.052
α_3(1) = 0.003496
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



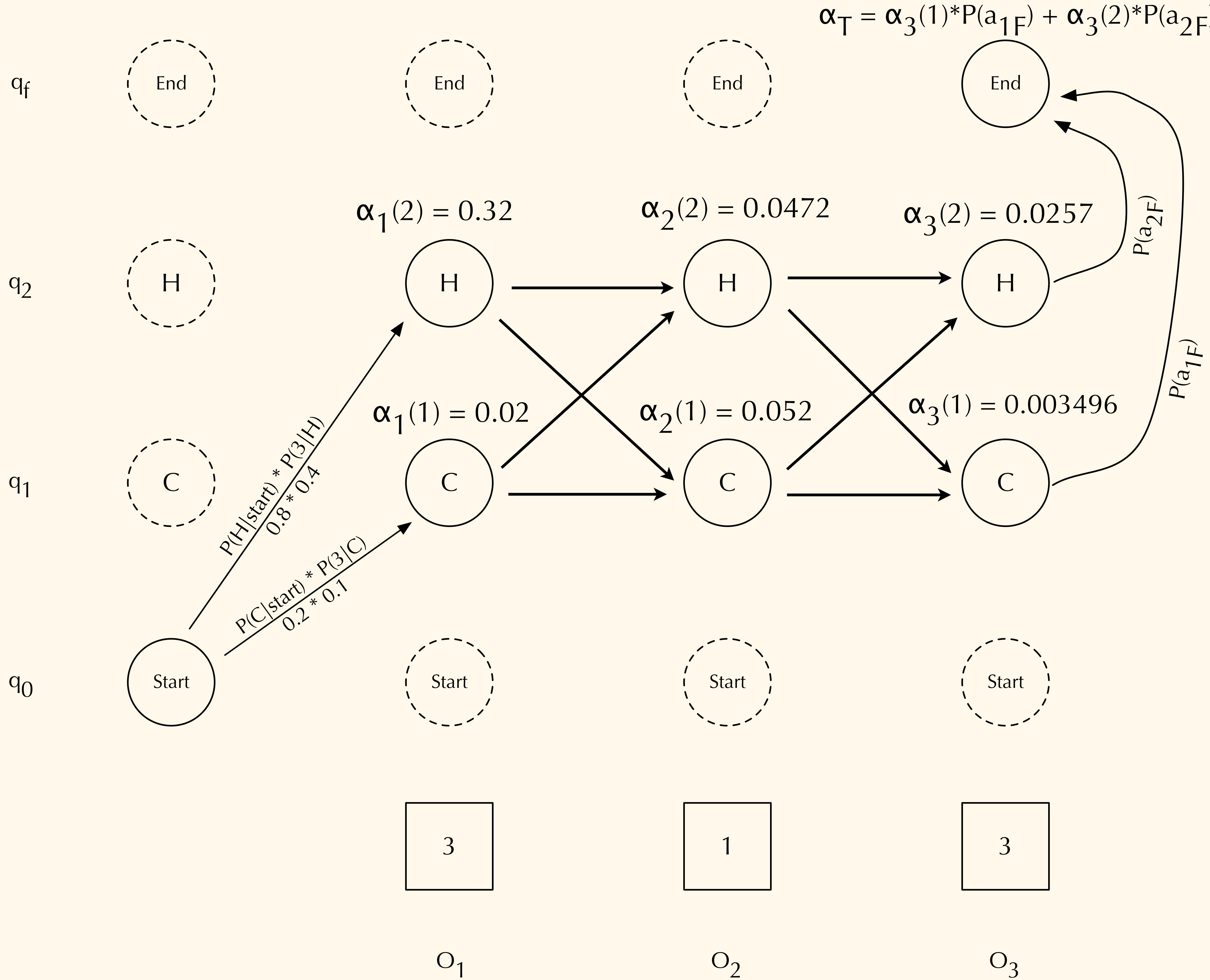
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
H
H
H
H
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
$\alpha_3(1) = 0.003496$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



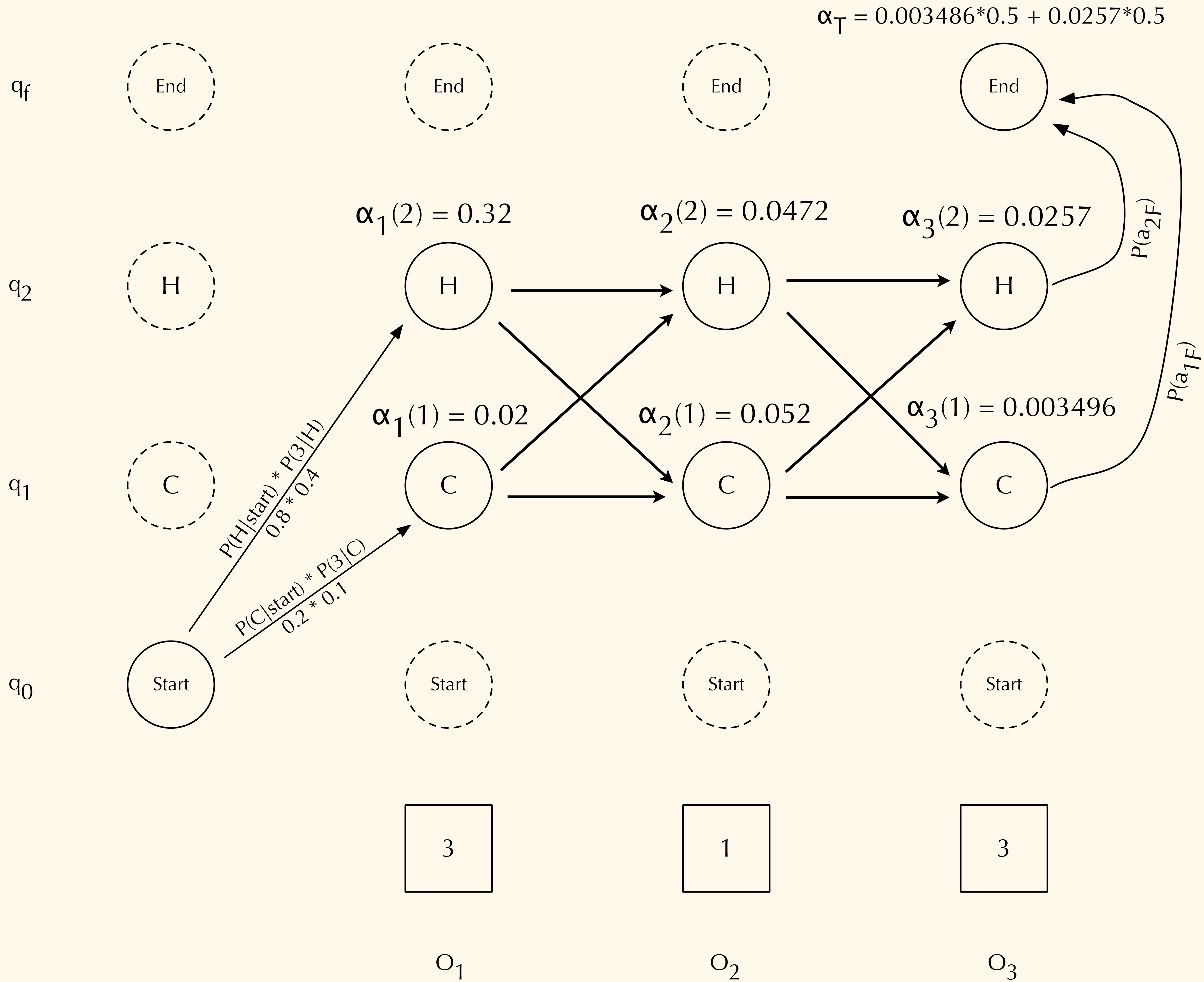
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
$\alpha_3(2) = 0.0257$
H
H
H
H
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
$\alpha_3(1) = 0.003496$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3

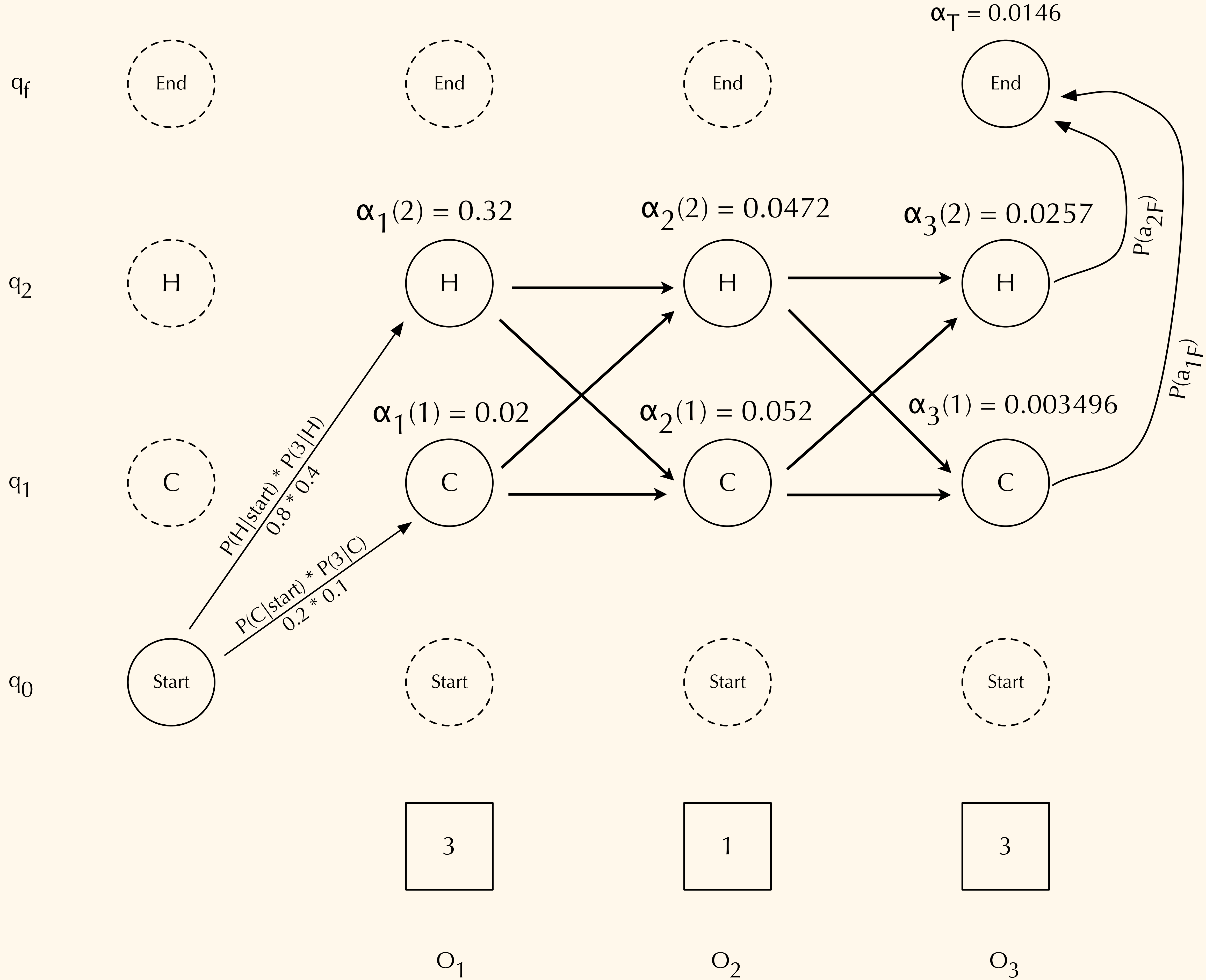


q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
$\alpha_3(2) = 0.0257$
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
$\alpha_3(1) = 0.003496$
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
$P(a_{2F})$
$P(a_{1F})$
3
1
3
O_1
O_2
O_3



$\alpha_T = \alpha_3(1)*P(a_{1F}) + \alpha_3(2)*P(a_{2F})$
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
$\alpha_3(2) = 0.0257$
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
$\alpha_3(1) = 0.003496$
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
$P(a_{2F})$
$P(a_{1F})$
3
1
3
O_1
O_2
O_3


α_T = 0.003486*0.5 + 0.0257*0.5
q_f
q_2
q_1
q_0
End
End
End
End
$\alpha_1(2)$ = 0.32
$\alpha_2(2)$ = 0.0472
$\alpha_3(2)$ = 0.0257
H
H
H
H
$P(a_{2F})$
$P(a_{1F})$
$\alpha_1(1)$ = 0.02
$\alpha_2(1)$ = 0.052
$\alpha_3(1)$ = 0.003496
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



$\alpha_T = 0.0146$
q_f
End
End
End
End
$\alpha_1(2) = 0.32$
$\alpha_2(2) = 0.0472$
$\alpha_3(2) = 0.0257$
$P(a_{2F})$
q_2
H
H
H
H
$P(a_{1F})$
$\alpha_1(1) = 0.02$
$\alpha_2(1) = 0.052$
$\alpha_3(1) = 0.003496$
q_1
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
q_0
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

There are three fundamental kinds of questions that we can ask with an HMM:

1. ***Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?**

2. *Decoding:* Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

There are three fundamental kinds of questions that we can ask with an HMM:

1. *Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?

2. ***Decoding:*** **Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?**

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

Formally: $\operatorname*{argmax}_{Q} P(Q|O)$

Formally: $\underset{Q}{\operatorname{argmax}} P(Q|O)$

"Given an observation *O*, what was the most probable sequence of states Q?"

Formally: $\underset{Q}{\operatorname{argmax}} P(Q|O)$

"Given an observation O, what was the most probable sequence of states Q?"

Decoding and likelihood estimation have certain similarities...

Formally: $\underset{Q}{\operatorname{argmax}} P(Q|O)$

"Given an observation O, what was the most probable sequence of states Q?"

Decoding and likelihood estimation have certain similarities...

One solution: run the forward algorithm over each possible state sequence...

Formally: $\underset{Q}{\operatorname{argmax}} P(Q|O)$

“Given an observation O, what was the most probable sequence of states Q?”

Decoding and likelihood estimation have certain similarities...

One solution: run the forward algorithm over each possible state sequence...

... which has the same issue as the naïve solution to the likelihood problem!

Formally: $\underset{Q}{\operatorname{argmax}} P(Q|O)$

"Given an observation *O*, what was the most probable sequence of states Q?"

Decoding and likelihood estimation have certain similarities...

One solution: run the forward algorithm over each possible state sequence...

... which has the same issue as the naïve solution to the likelihood problem!

$O(N^T)$ possible solutions...

Decoding and likelihood estimation have the same problems...

Decoding and likelihood estimation have the same problems...

... and they share a solution.

Decoding and likelihood estimation have the same problems…

… and they share a solution.

Modifying the forward algorithm slightly gives us the *Viterbi algorithm* for decoding.

Decoding and likelihood estimation have the same problems...

... and they share a solution.

Modifying the forward algorithm slightly gives us the *Viterbi algorithm* for decoding.

The main difference: instead of *summing* possible paths to each state, we take the *max*...

Decoding and likelihood estimation have the same problems...

... and they share a solution.

Modifying the forward algorithm slightly gives us the *Viterbi algorithm* for decoding.

The main difference: instead of *summing* possible paths to each state, we take the *max*...

... and *keep track* of which one it was!

Forward algorithm trellis locations:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

Forward algorithm trellis locations:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

Viterbi algorithm trellis locations:

$$v_t(j) = \max_{i=1}^{N} v_{t-1} a_{ij} b_j(o_t)$$

Forward algorithm trellis locations:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

Viterbi algorithm trellis locations:

$$v_t(j) = \max_{i=1}^{N} v_{t-1} a_{ij} b_j(o_t)$$

We also save a backtrace through the most-likely states:

Forward algorithm trellis locations:

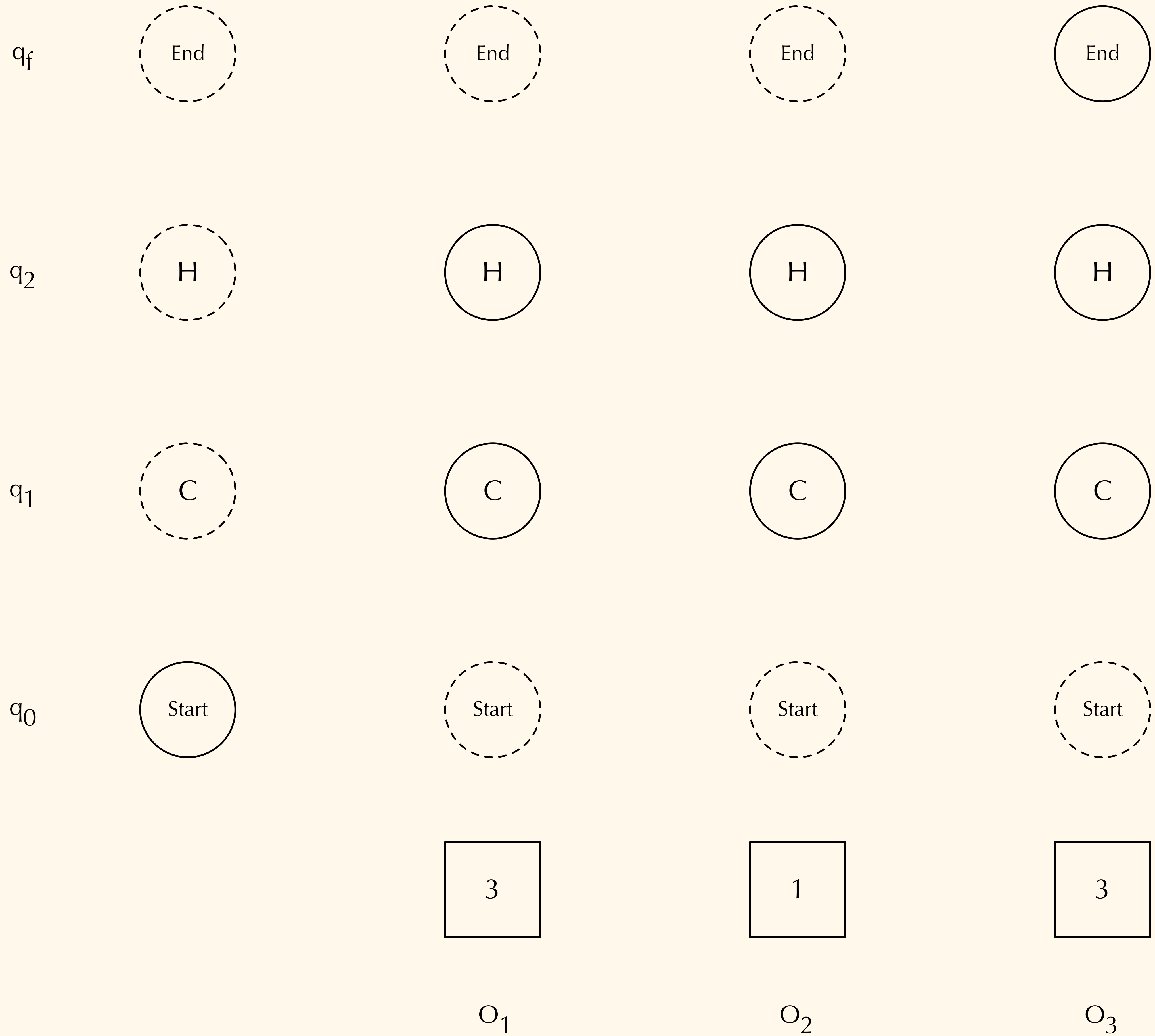
$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$$

Viterbi algorithm trellis locations:

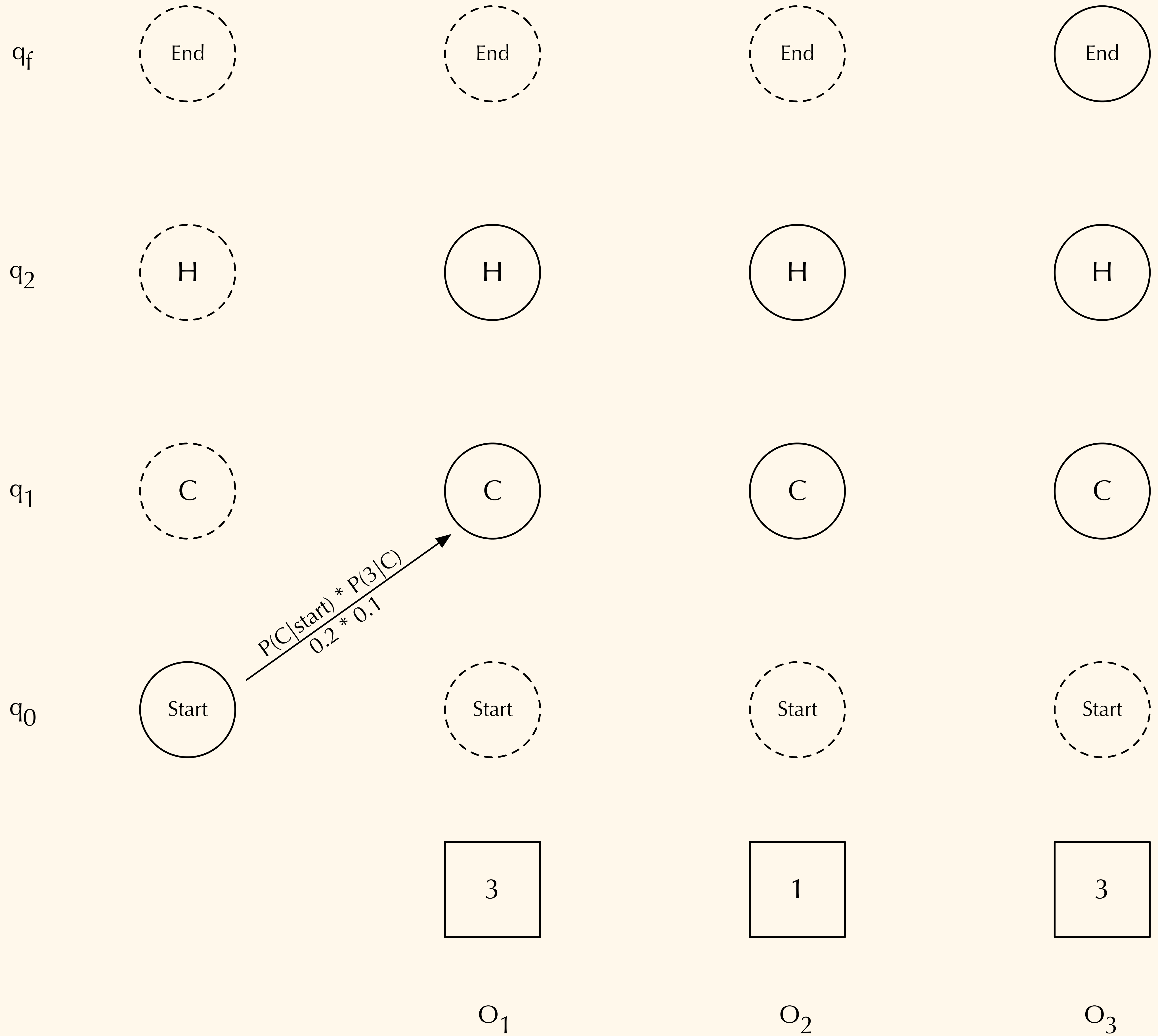
$$v_t(j) = \max_{i=1}^{N} v_{t-1} a_{ij} b_j(o_t)$$

We also save a backtrace through the most-likely states:

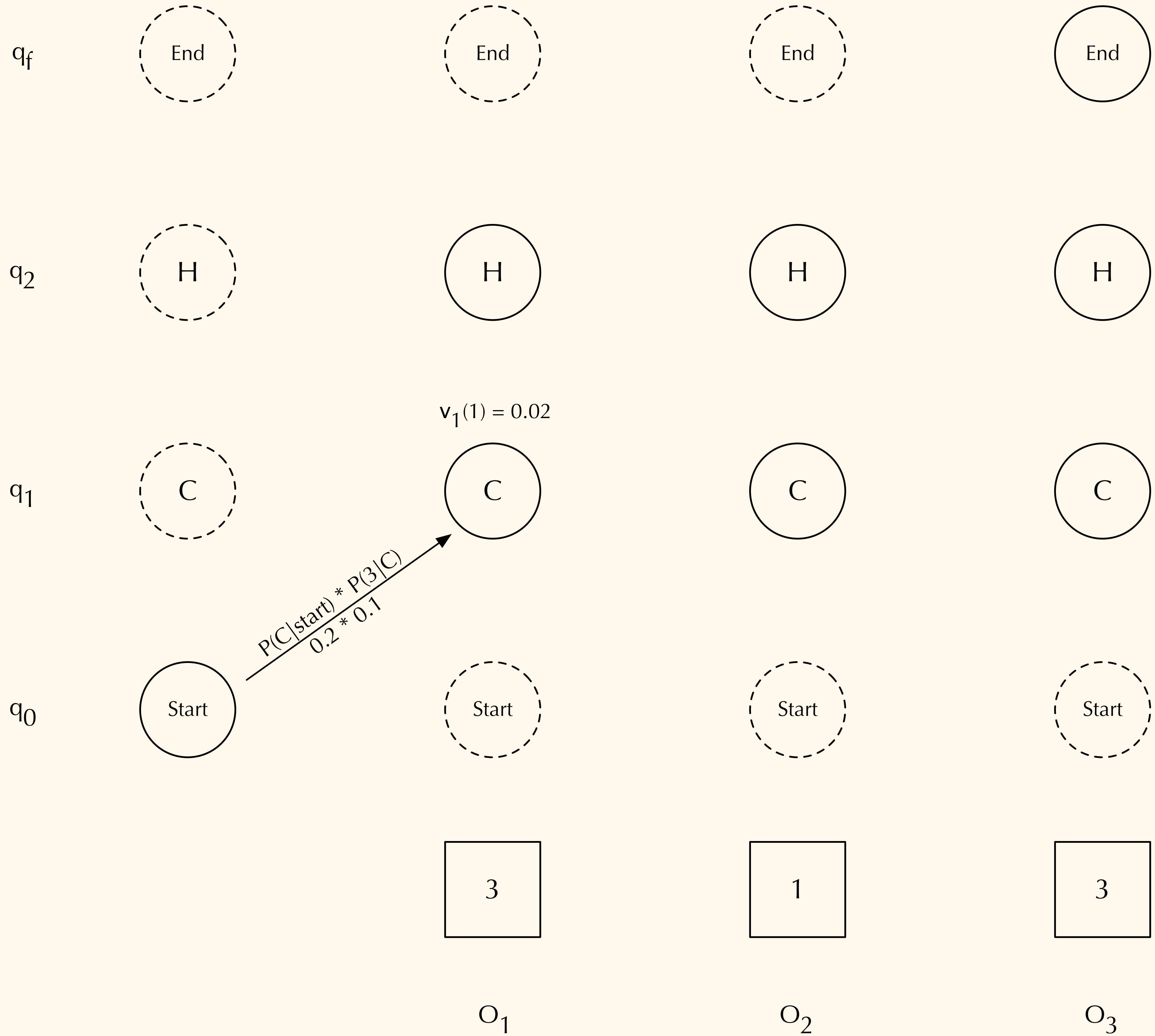
$$bt_t(j) = \operatorname*{argmax}_{i=1}^{N} v_{t-1} a_{ij} b_j(o_t)$$



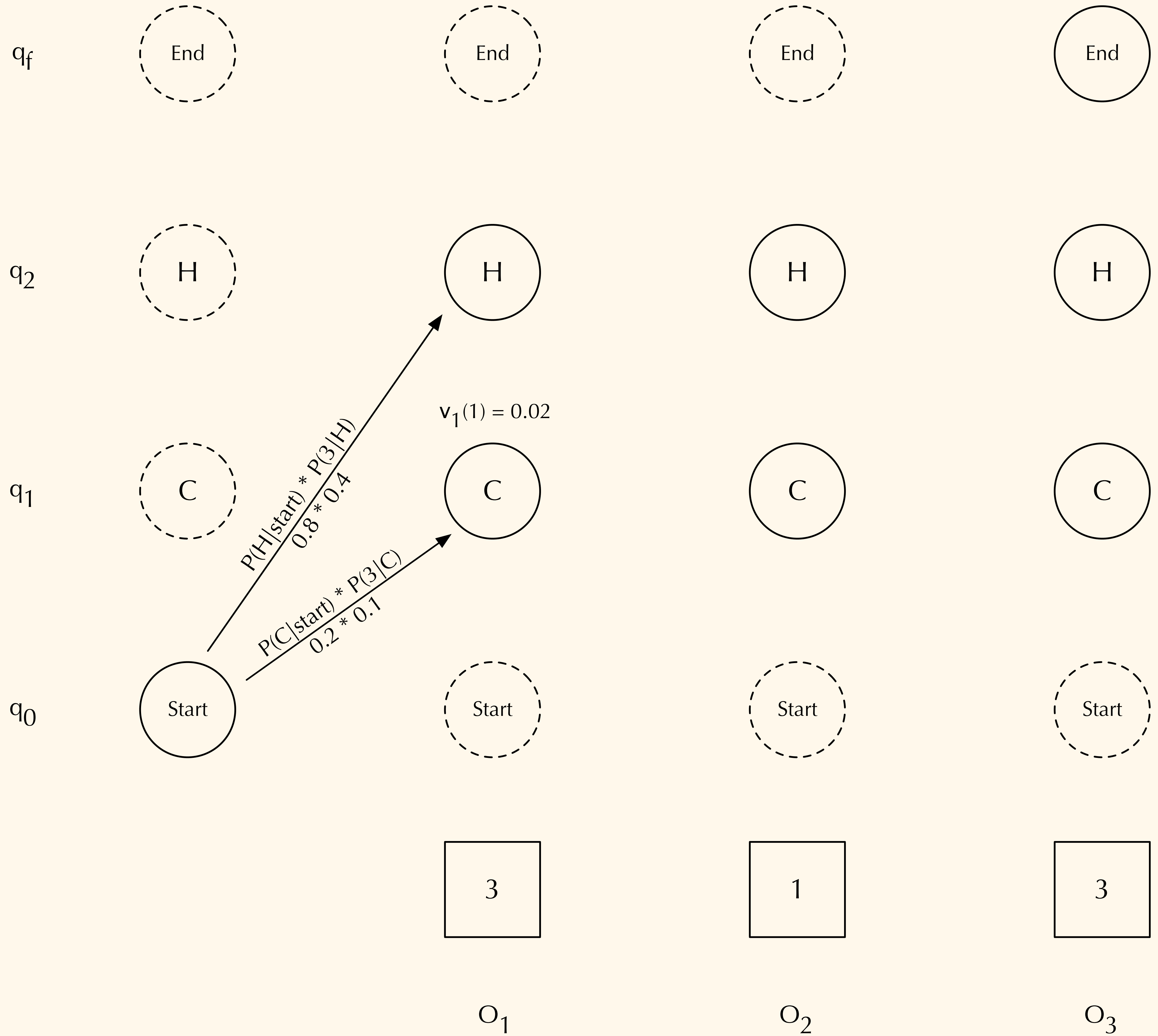
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



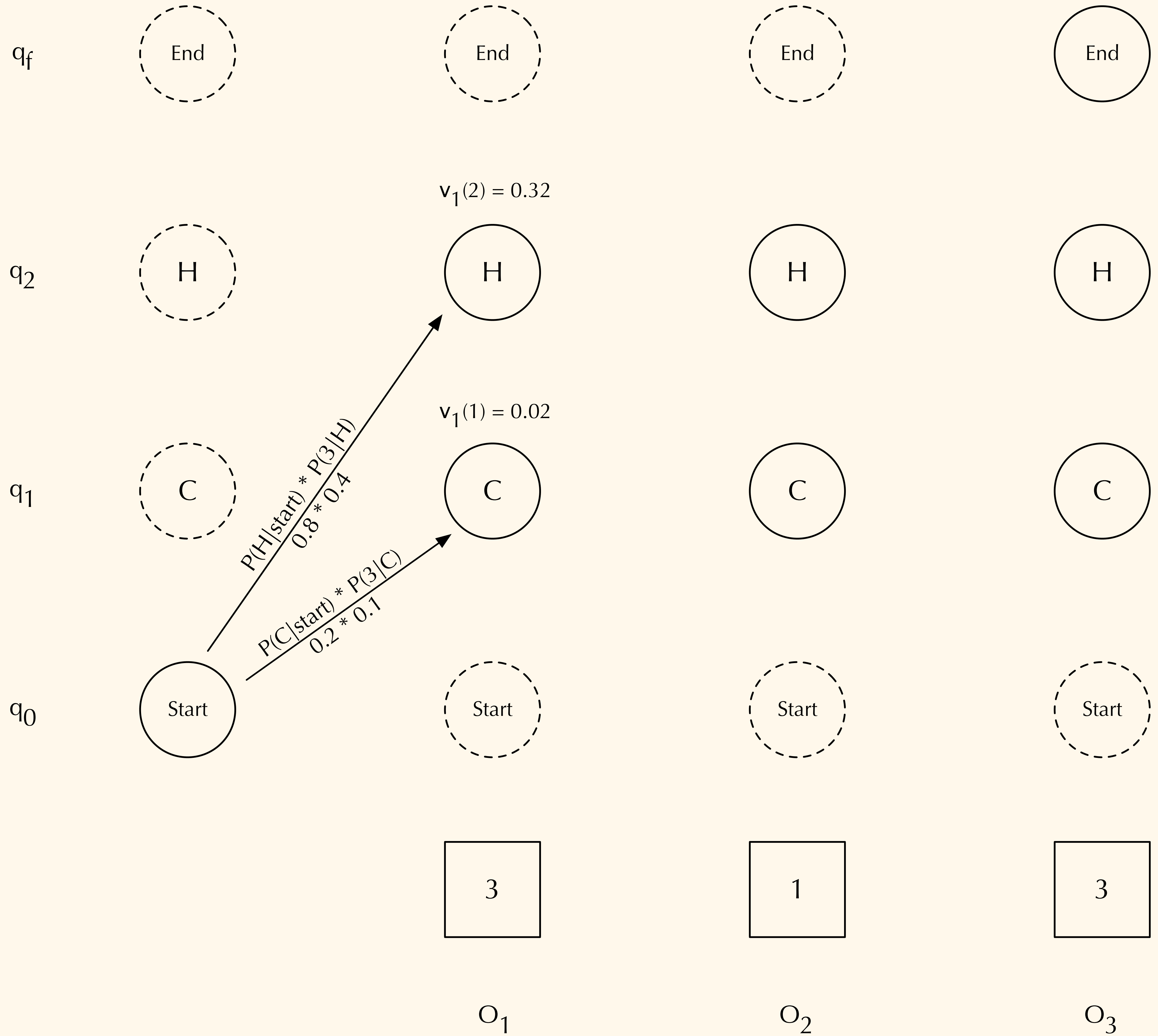
qf
q2
q1
q0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
P(C|start) * P(3|C)
0.2 * 0.1
3
1
3
O1
O2
O3



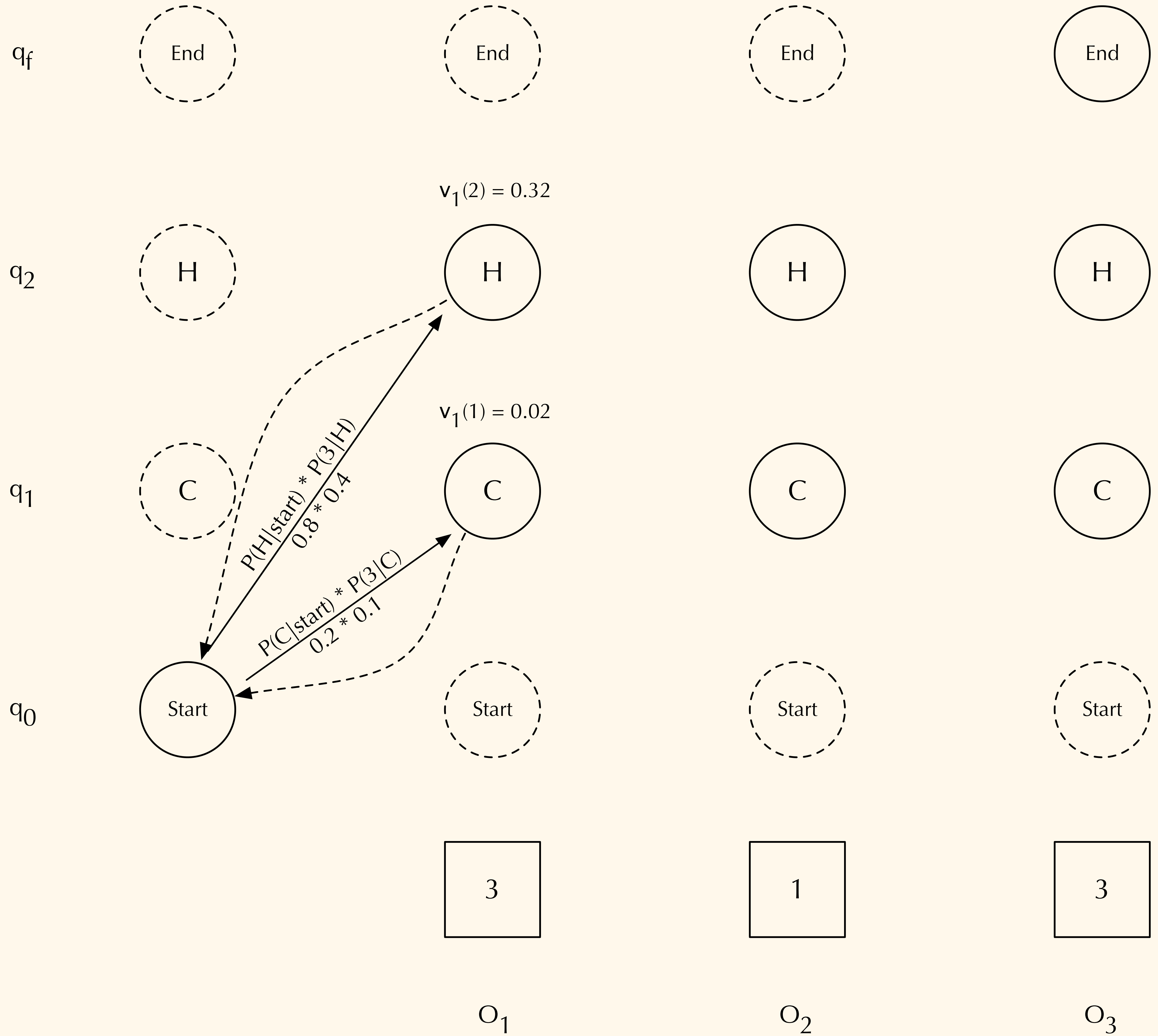
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
v_1(1) = 0.02
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



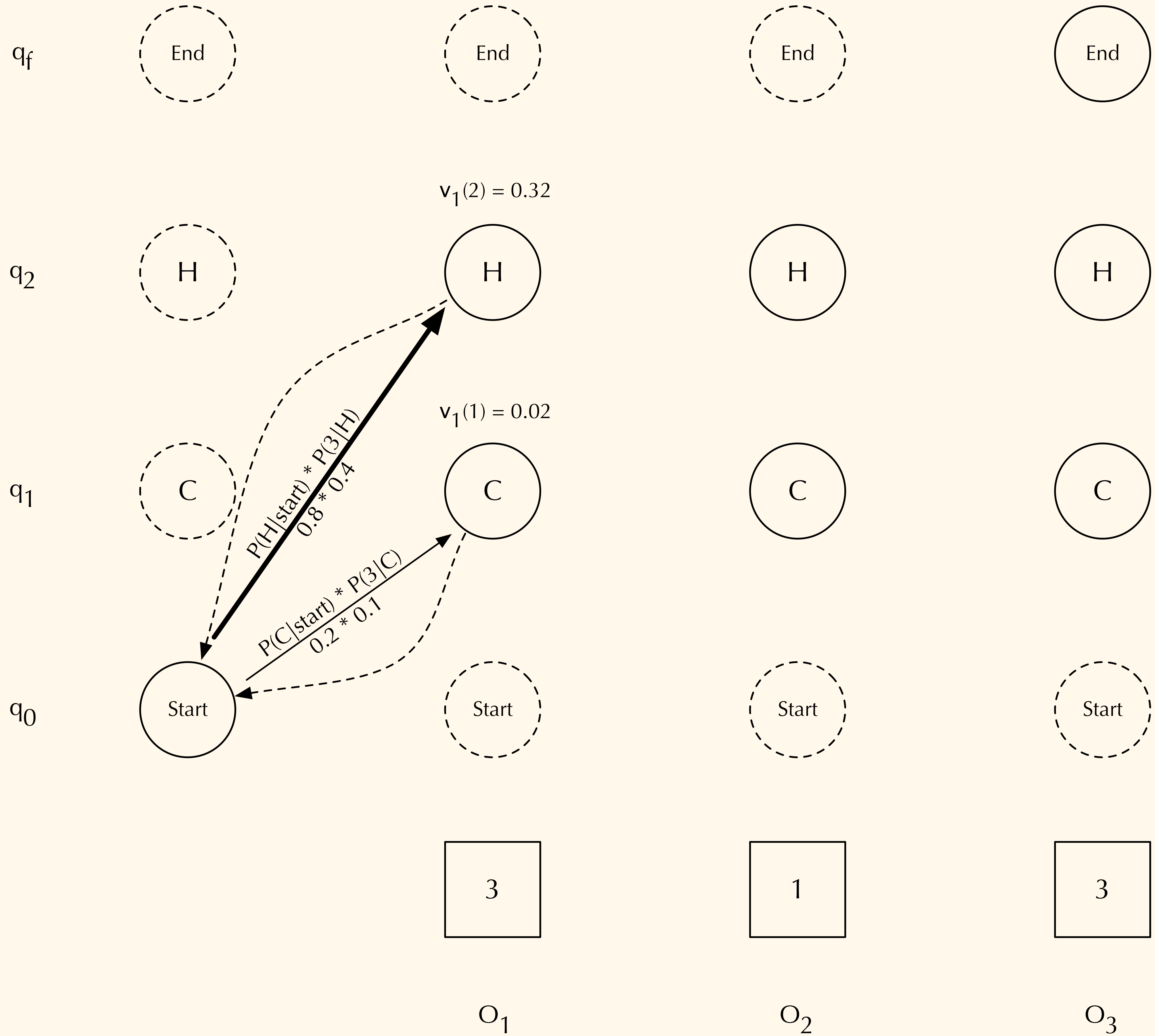
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
v_1(1) = 0.02
3
1
3
O_1
O_2
O_3

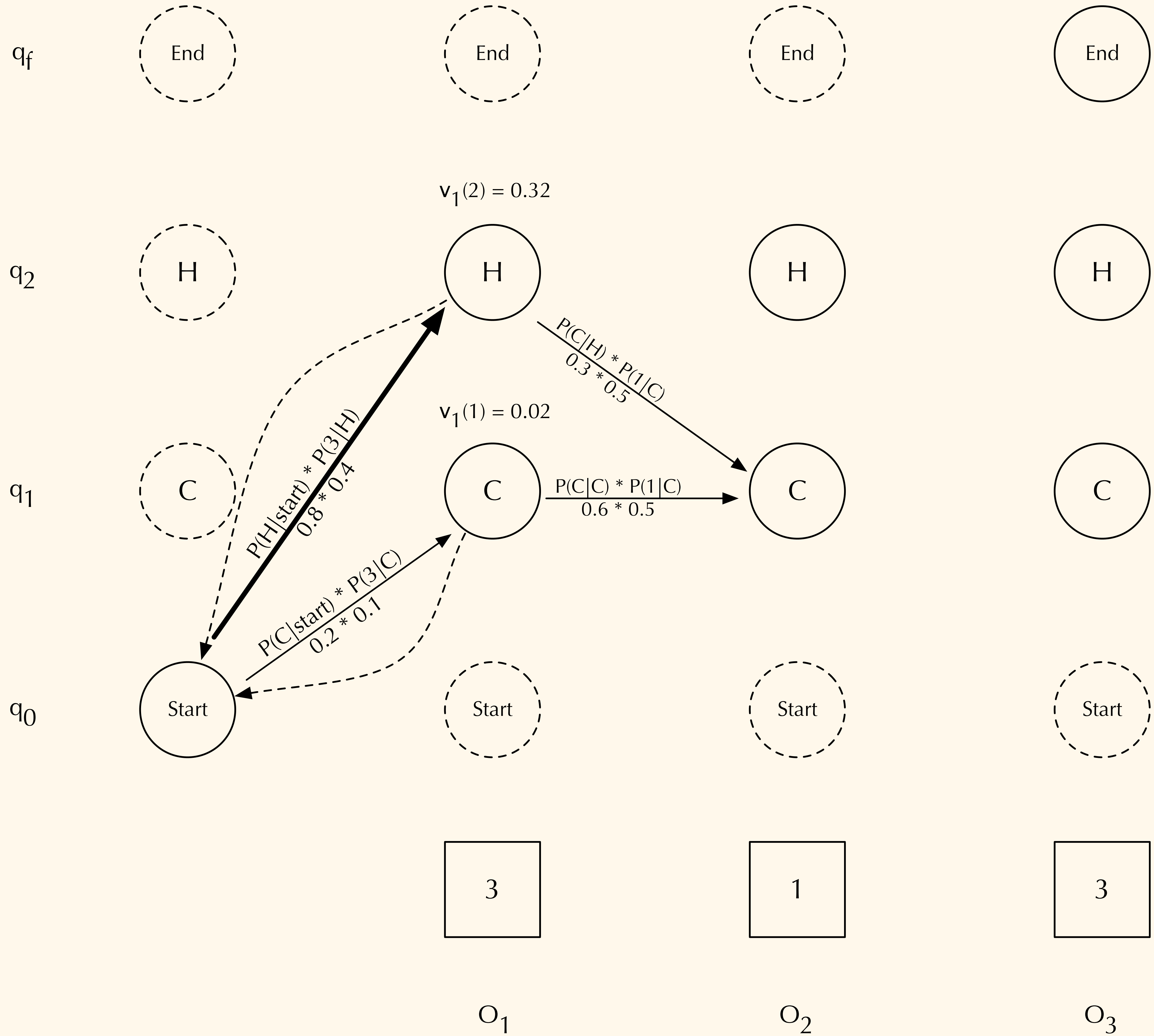


q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
H
H
H
H
$v_1(1) = 0.02$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

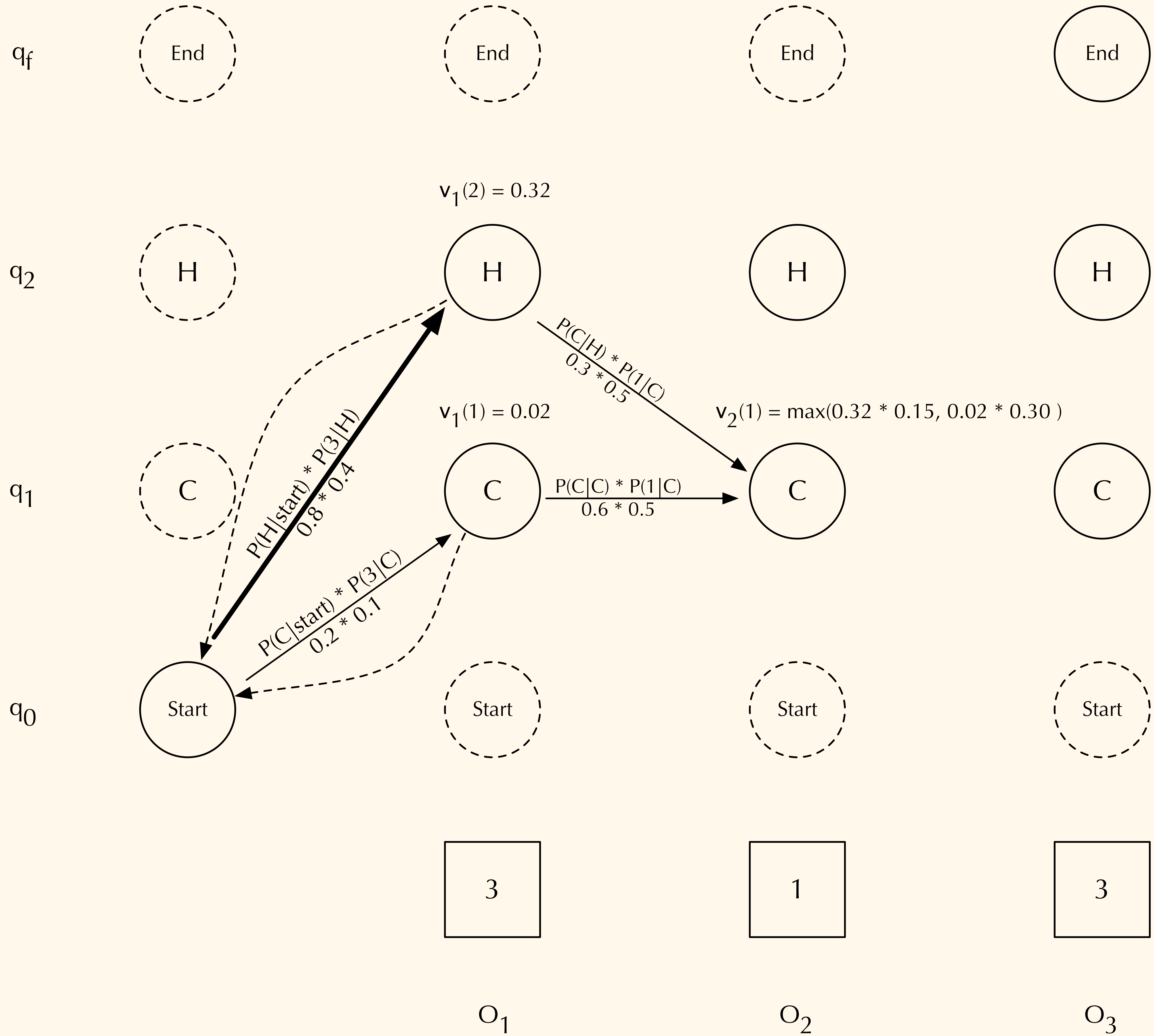


q_f
q_2
q_1
q_0
End
End
End
End
v_1(2) = 0.32
H
H
H
H
v_1(1) = 0.02
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

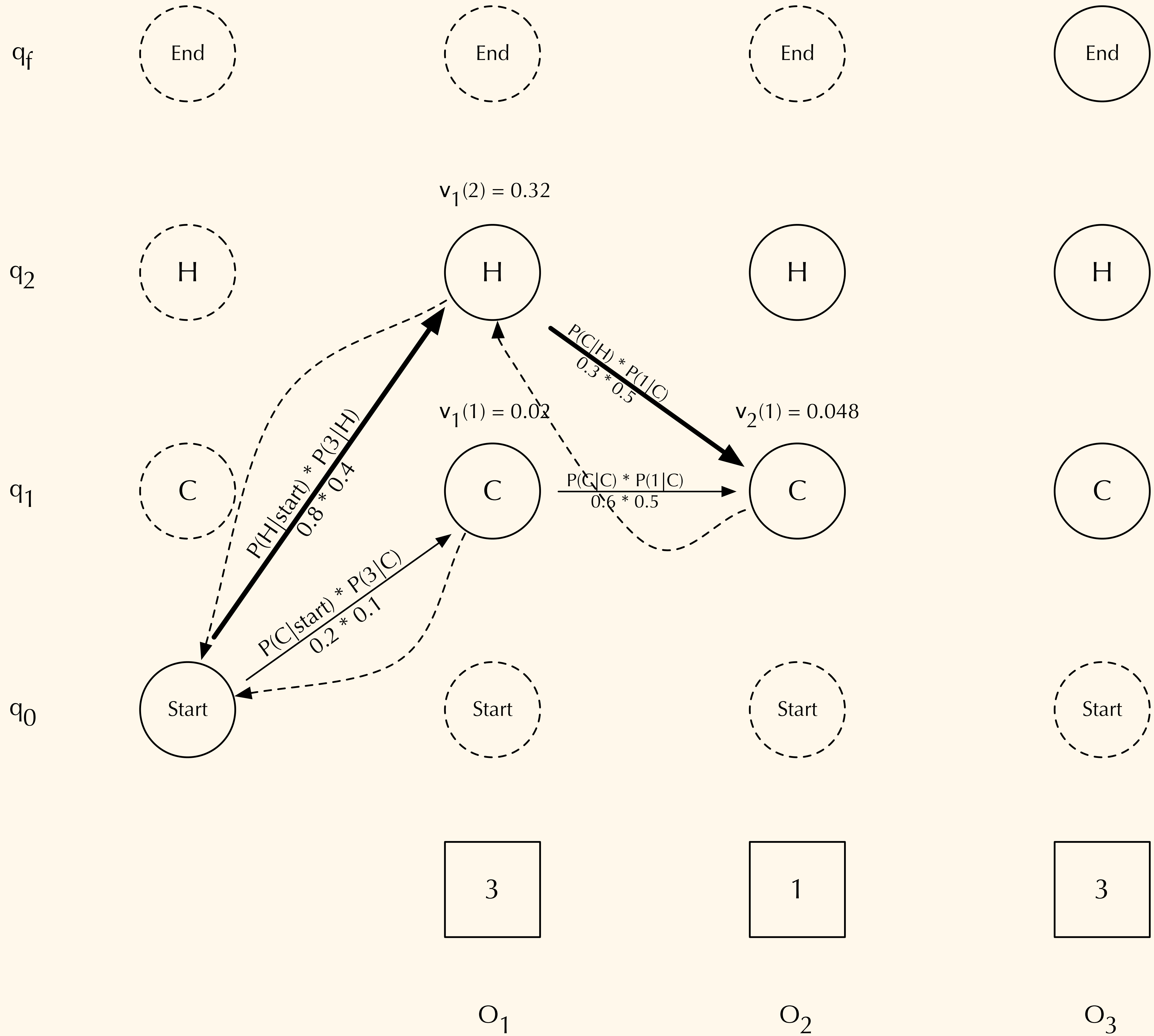

q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
H
H
H
H
$v_1(1) = 0.02$
C
C
C
C
Start
Start
Start
Start
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
3
1
3
O_1
O_2
O_3



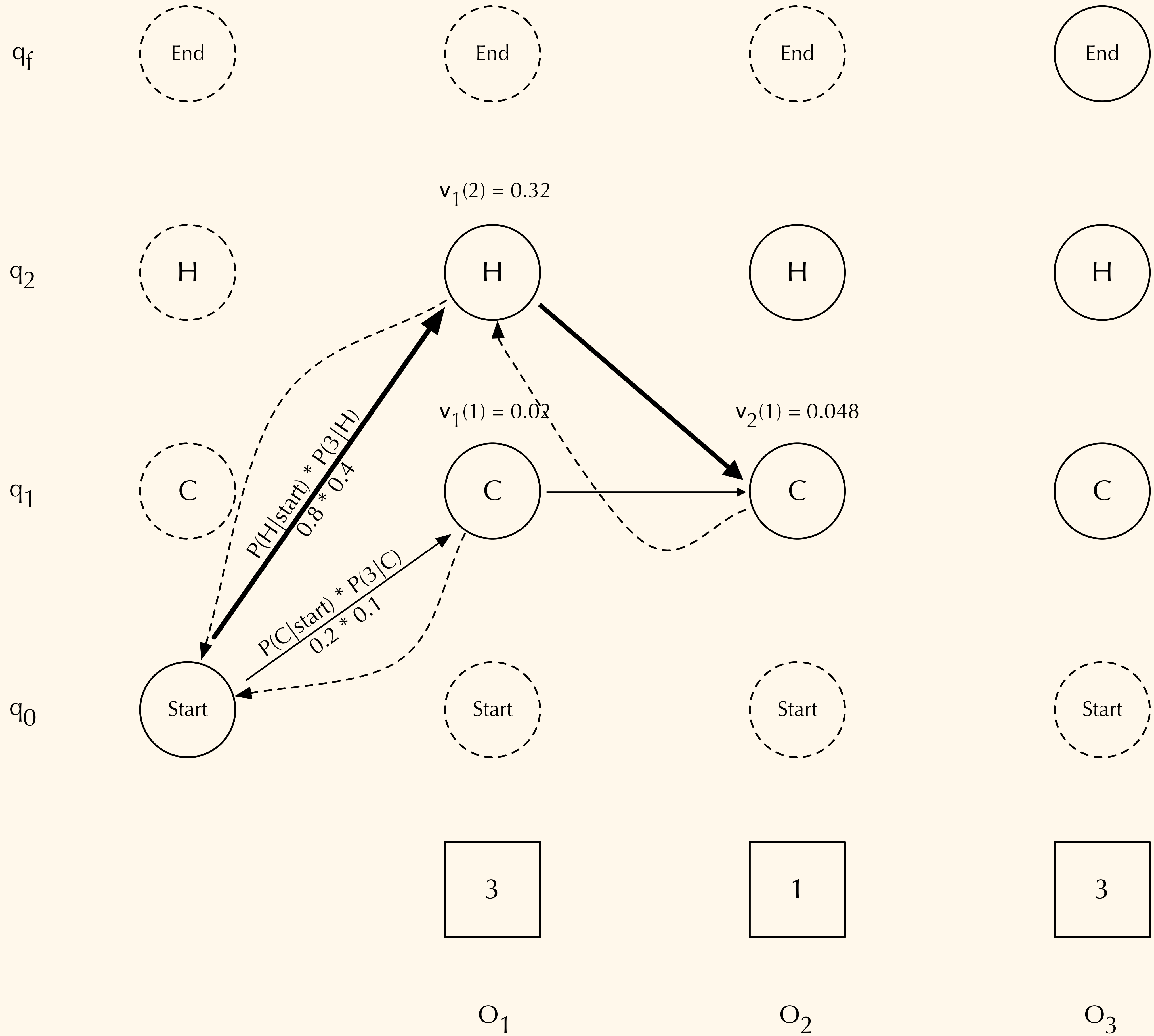
q_f
q_2
q_1
q_0
End
End
End
End
v_1(2) = 0.32
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
v_1(1) = 0.02
P(H|start) * P(3|H)
0.8 * 0.4
C
C
P(C|C) * P(1|C)
0.6 * 0.5
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



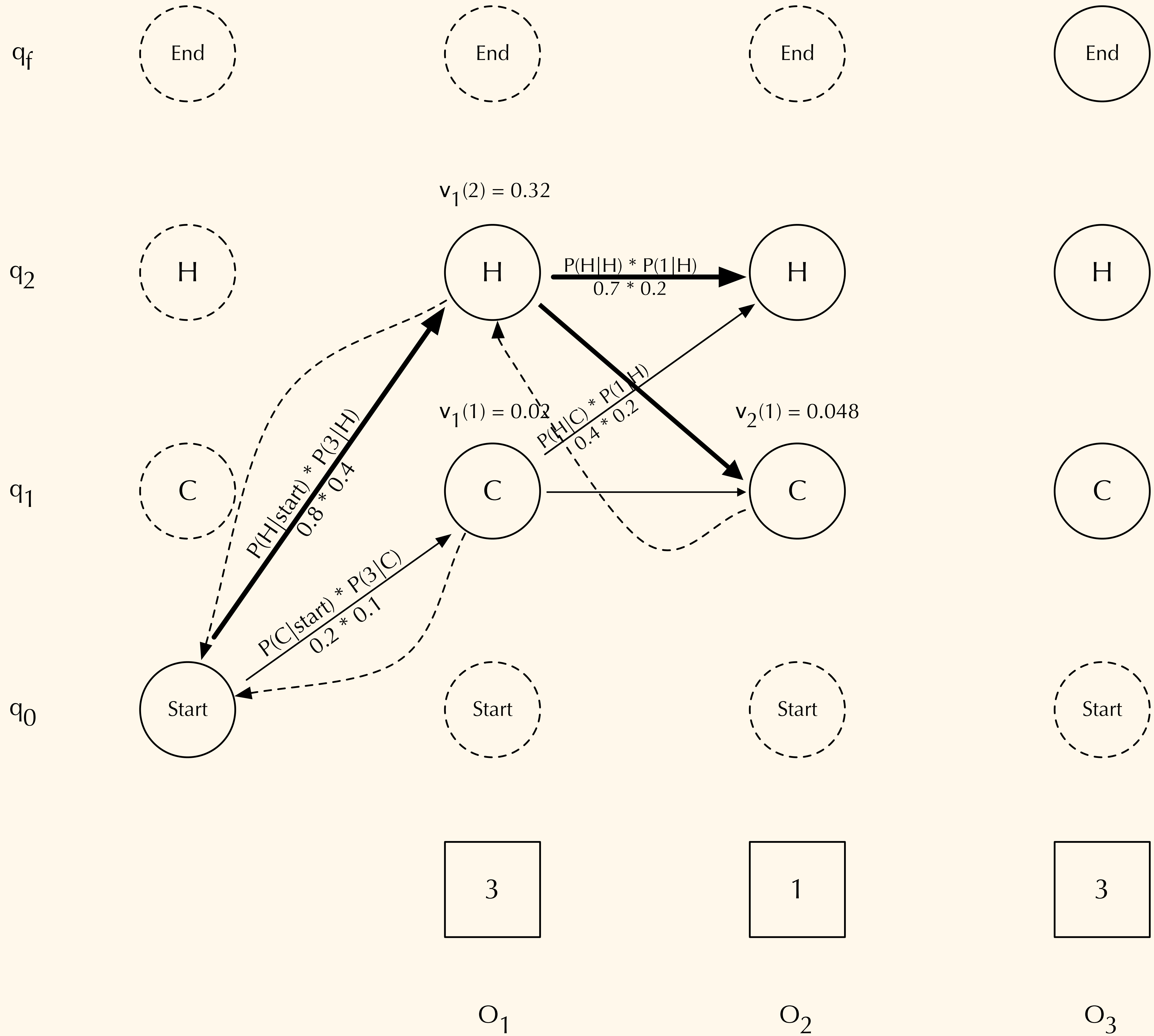
q_f
q_2
q_1
q_0
End
End
End
End
v_1(2) = 0.32
H
H
H
H
P(C|H) * P(1|C)
0.3 * 0.5
v_1(1) = 0.02
v_2(1) = max(0.32 * 0.15, 0.02 * 0.30)
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(1|C)
0.6 * 0.5
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



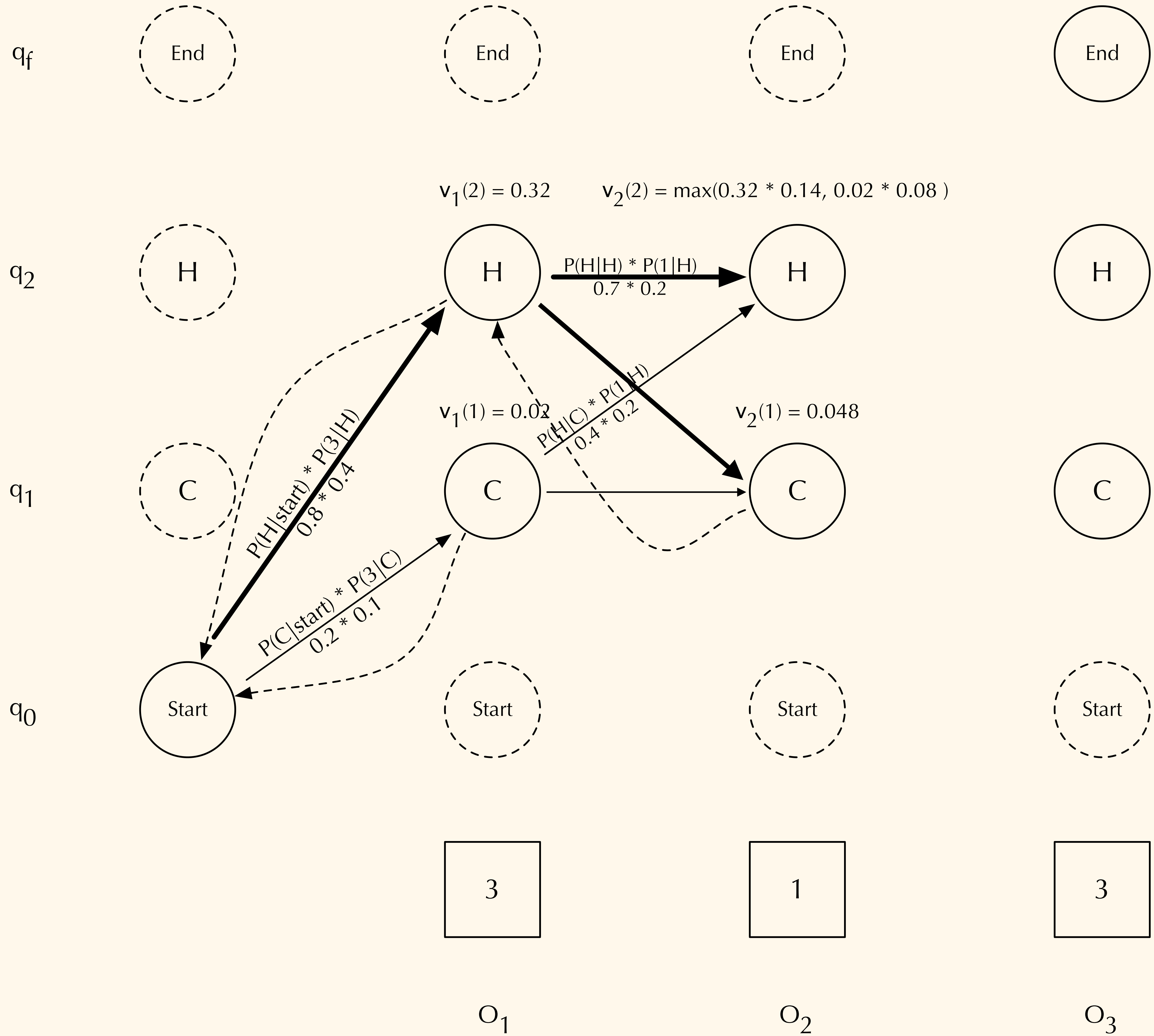
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
H
H
H
H
$v_1(1) = 0.02$
$v_2(1) = 0.048$
C
C
C
C
Start
Start
Start
Start
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
P(C|H) * P(1|C)
0.3 * 0.5
P(C|C) * P(1|C)
0.6 * 0.5
3
1
3
O_1
O_2
O_3



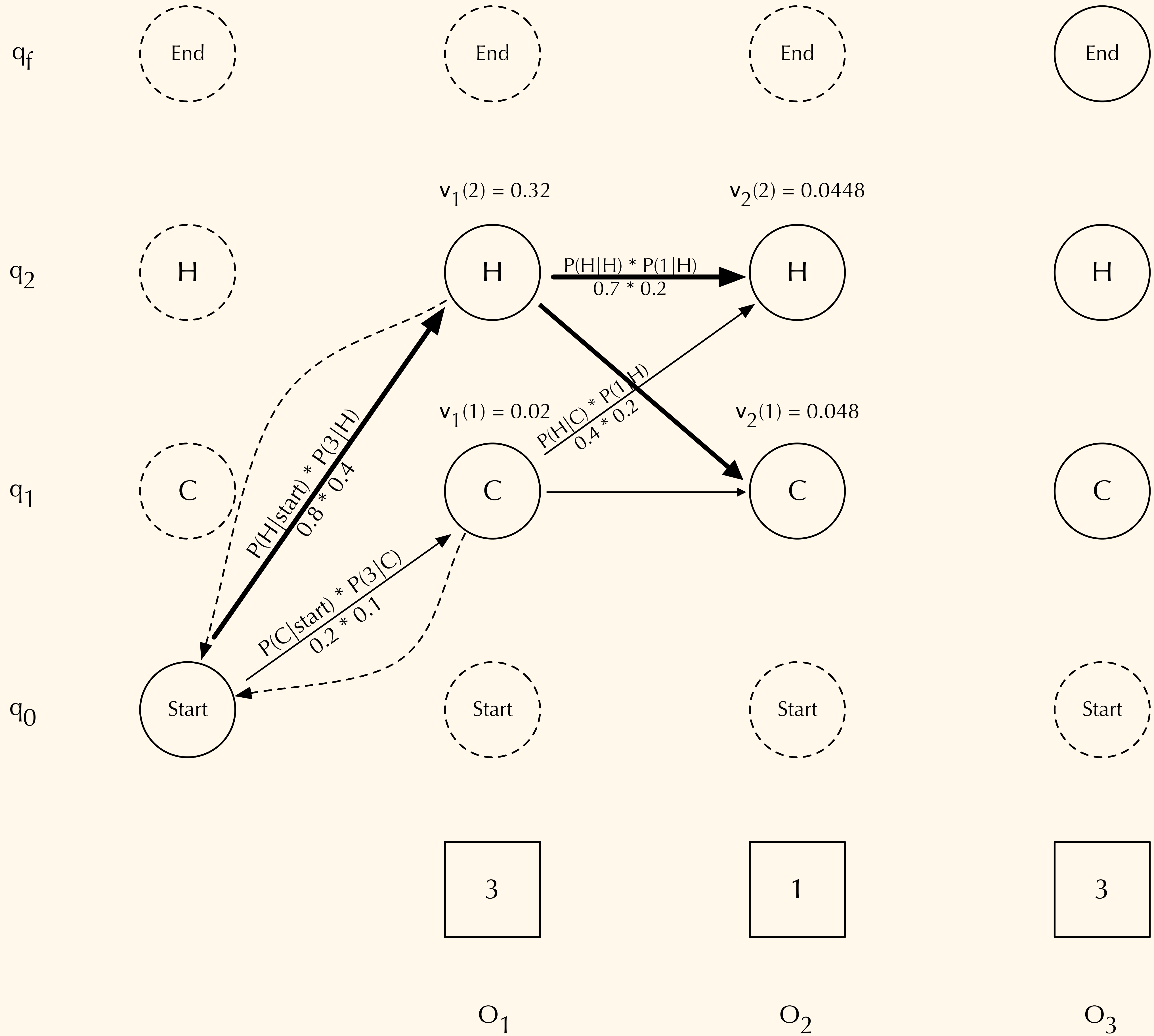
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
H
H
H
H
$v_1(1) = 0.02$
$v_2(1) = 0.048$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



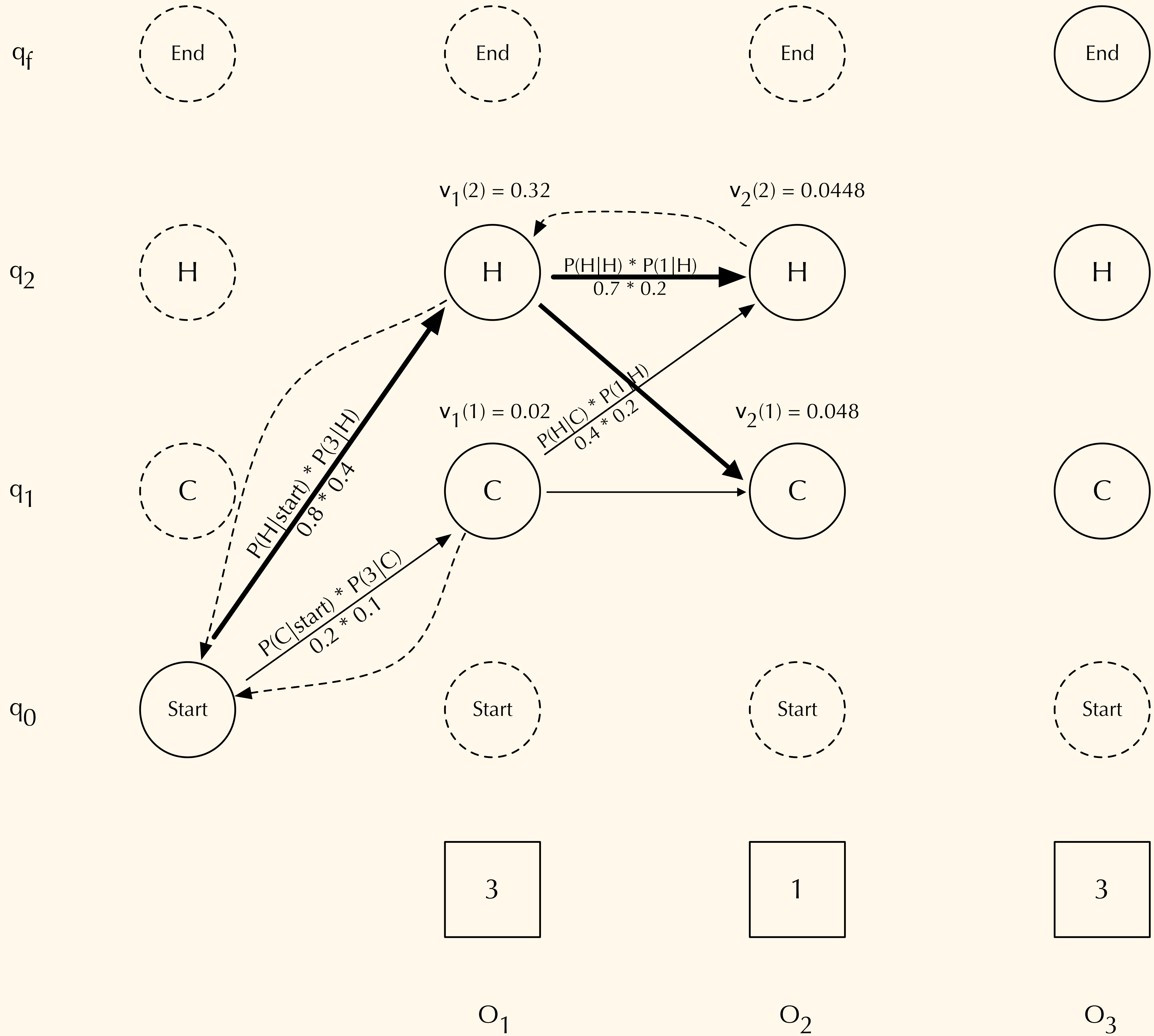
q_f
q_2
q_1
q_0
End
End
End
End
v_1(2) = 0.32
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
v_1(1) = 0.02
v_2(1) = 0.048
P(H|C) * P(1|H)
0.4 * 0.2
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



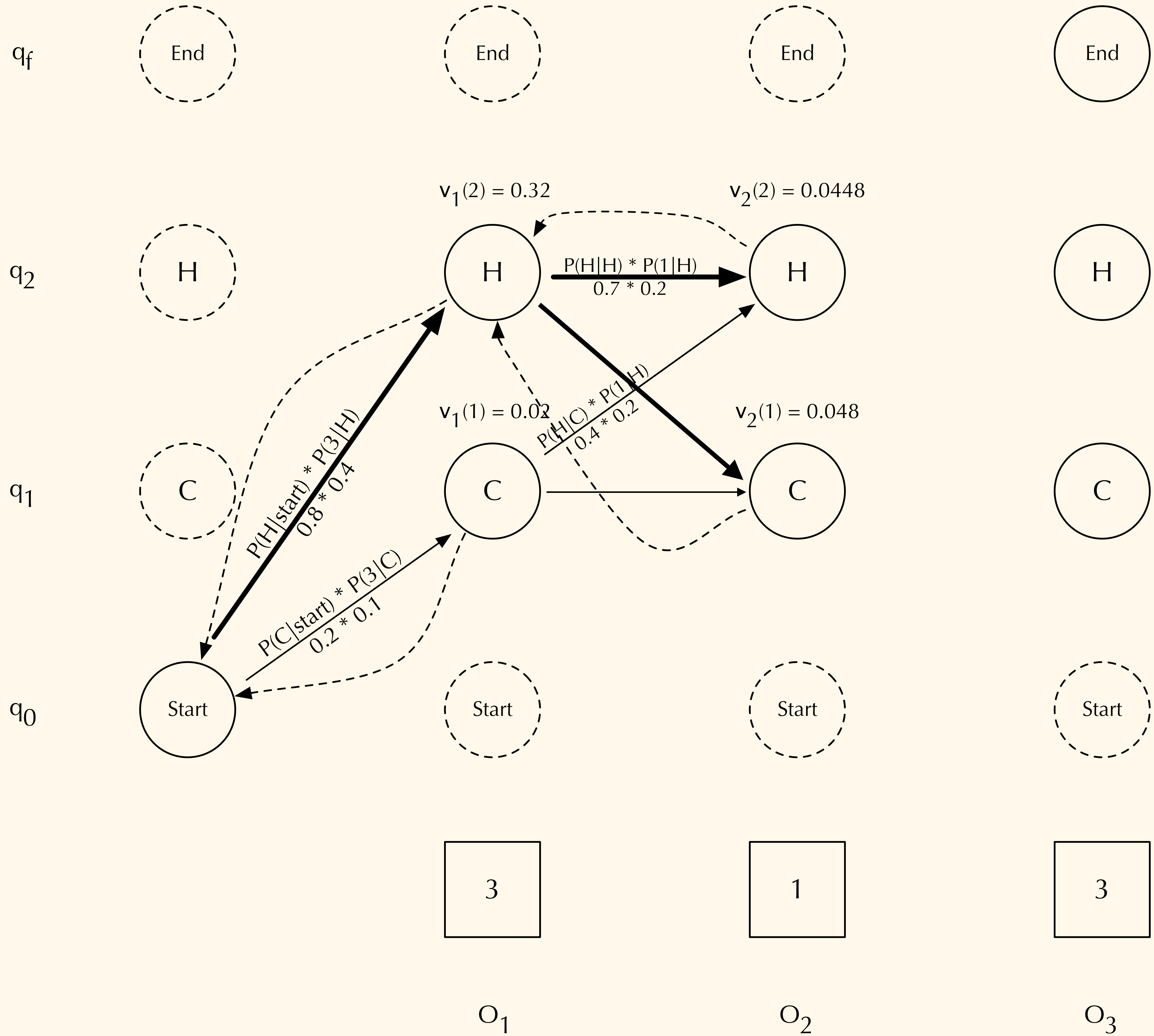
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
v_1(2) = 0.32
v_2(2) = max(0.32 * 0.14, 0.02 * 0.08)
v_1(1) = 0.02
v_2(1) = 0.048
P(H|H) * P(1|H)
0.7 * 0.2
P(H|C) * P(1|H)
0.4 * 0.2
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
3
1
3
O_1
O_2
O_3



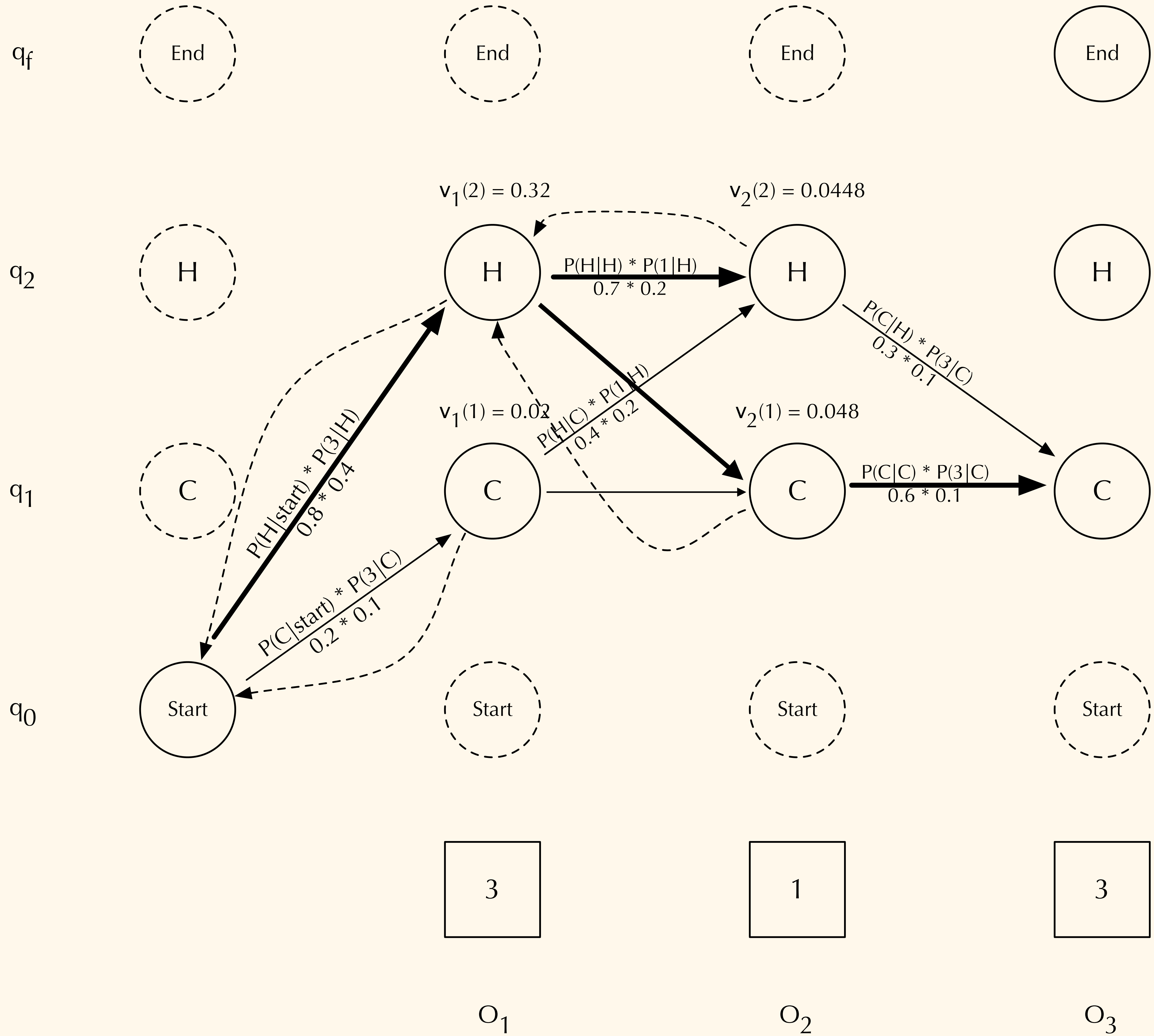
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$v_1(1) = 0.02$
$v_2(1) = 0.048$
P(H|C) * P(1|H)
0.4 * 0.2
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



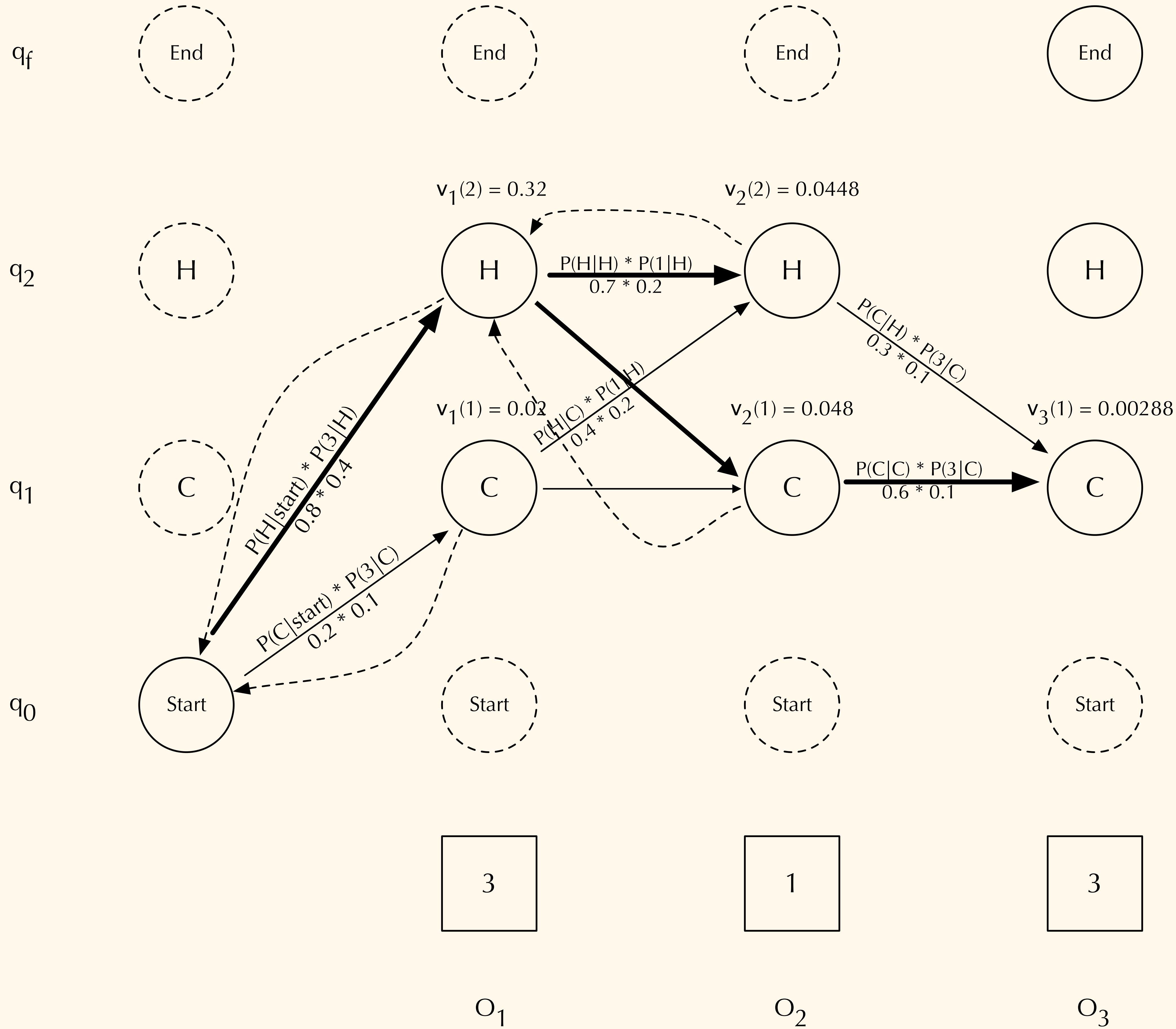
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$v_1(1) = 0.02$
$v_2(1) = 0.048$
P(H|C) * P(1|H)
0.4 * 0.2
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



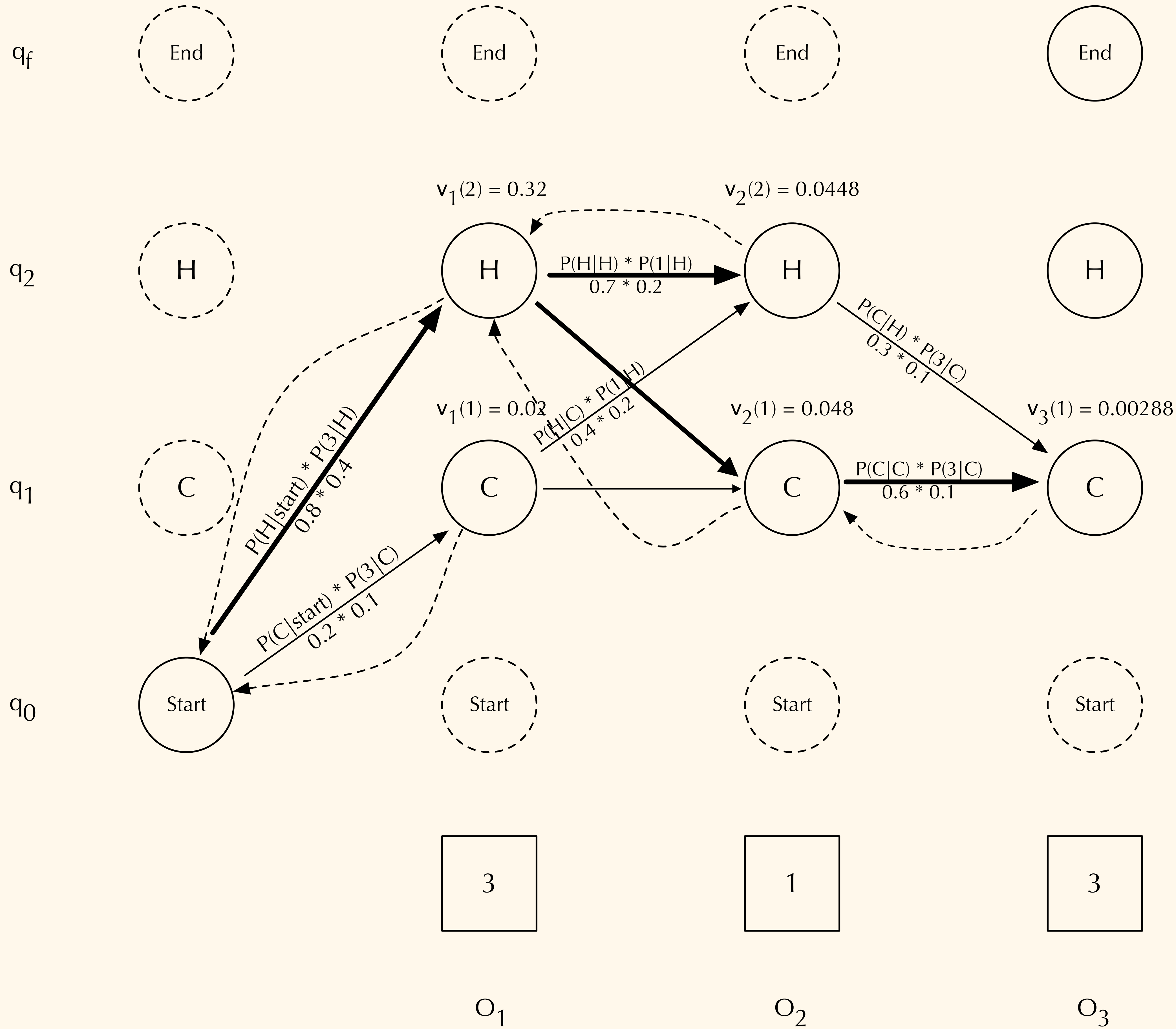
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$v_1(1) = 0.02$
$v_2(1) = 0.048$
P(H|C) * P(1|H)
0.4 * 0.2
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



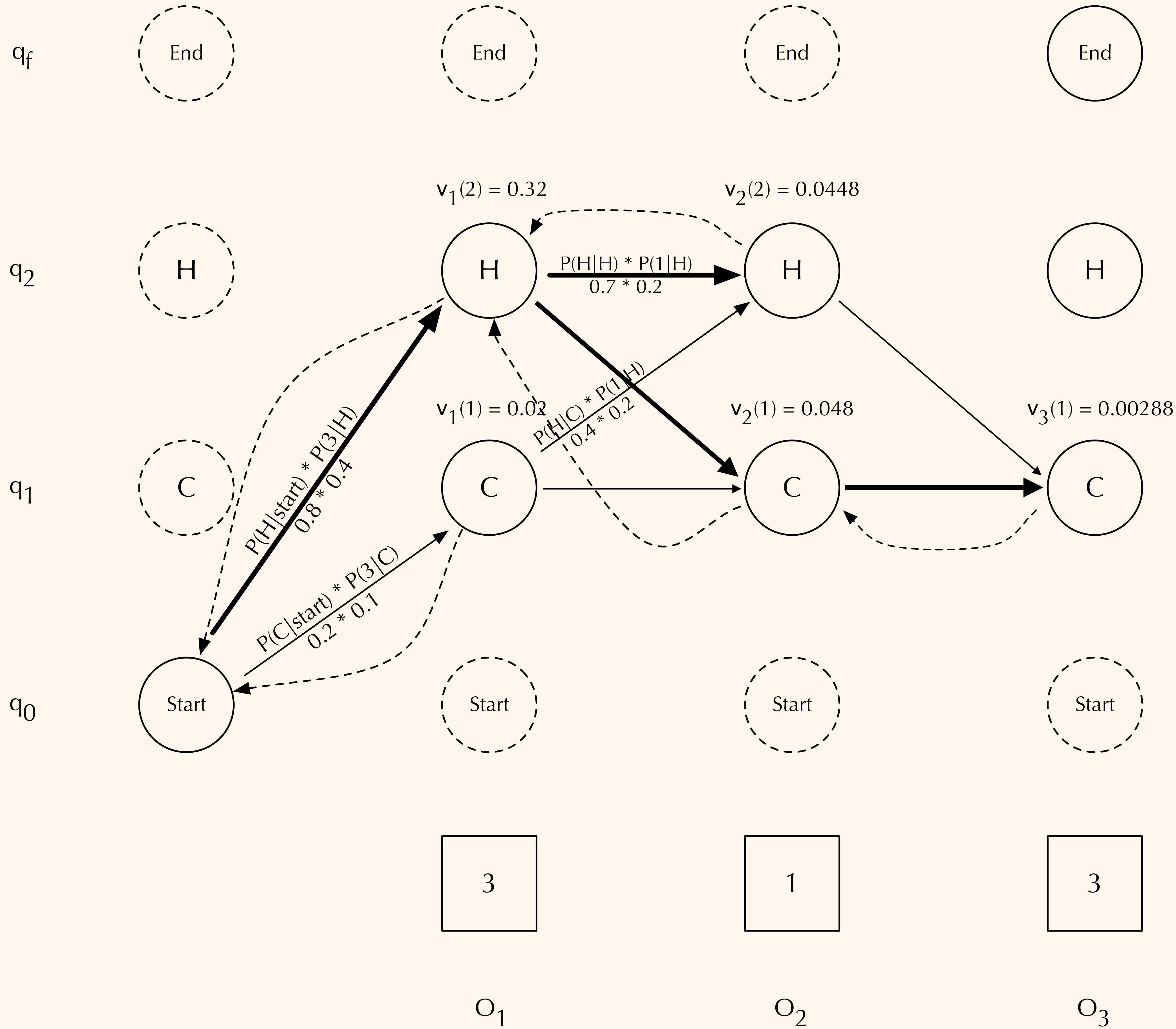
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(C|H) * P(3|C)
0.3 * 0.1
$v_1(1) = 0.02$
$v_2(1) = 0.048$
P(H|C) * P(1|H)
0.4 * 0.2
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(3|C)
0.6 * 0.1
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



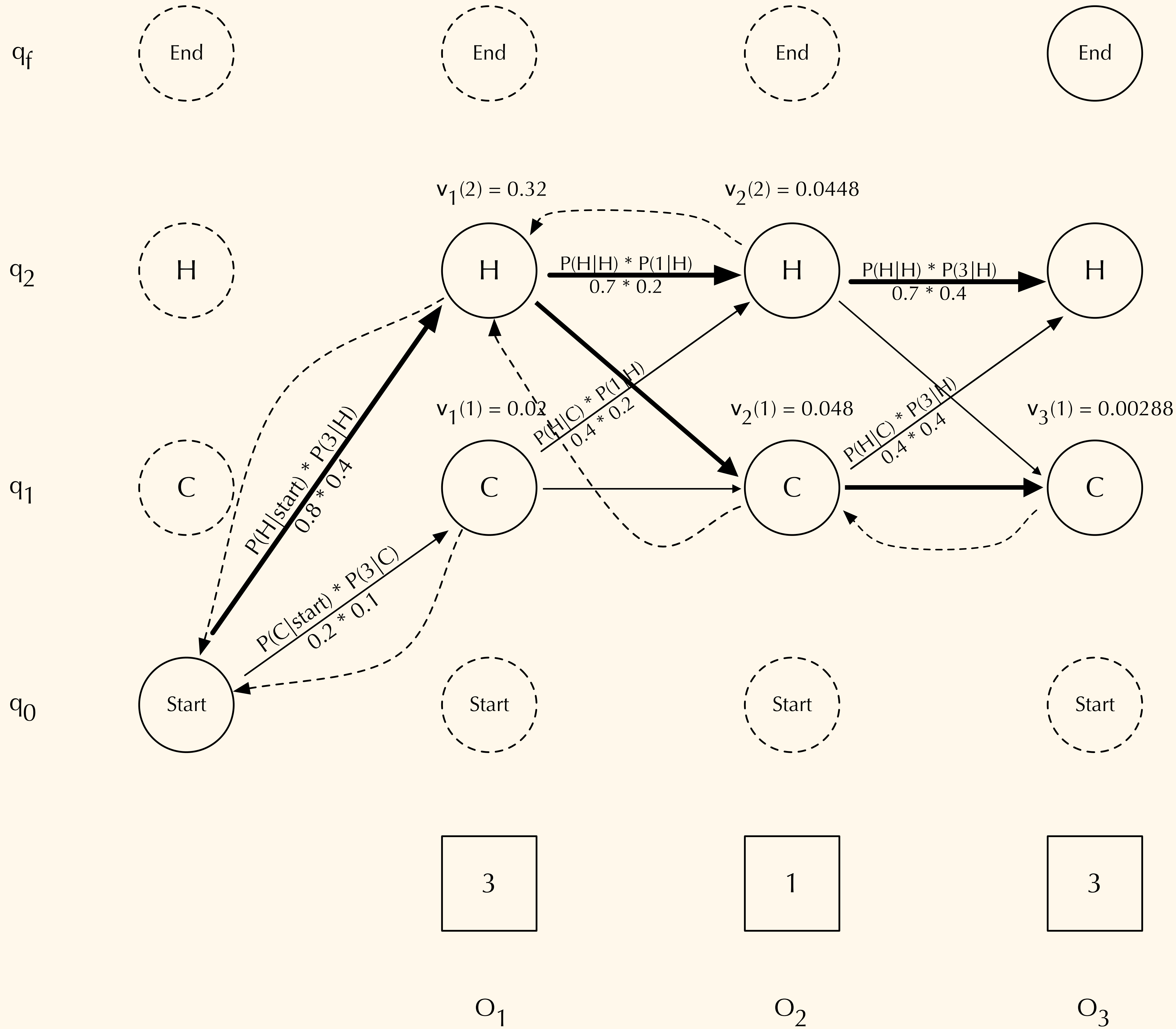
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(C|H) * P(3|C)
0.3 * 0.1
$v_1(1) = 0.02$
P(H|C) * P(1|H)
0.4 * 0.2
$v_2(1) = 0.048$
$v_3(1) = 0.00288$
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|C) * P(3|C)
0.6 * 0.1
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



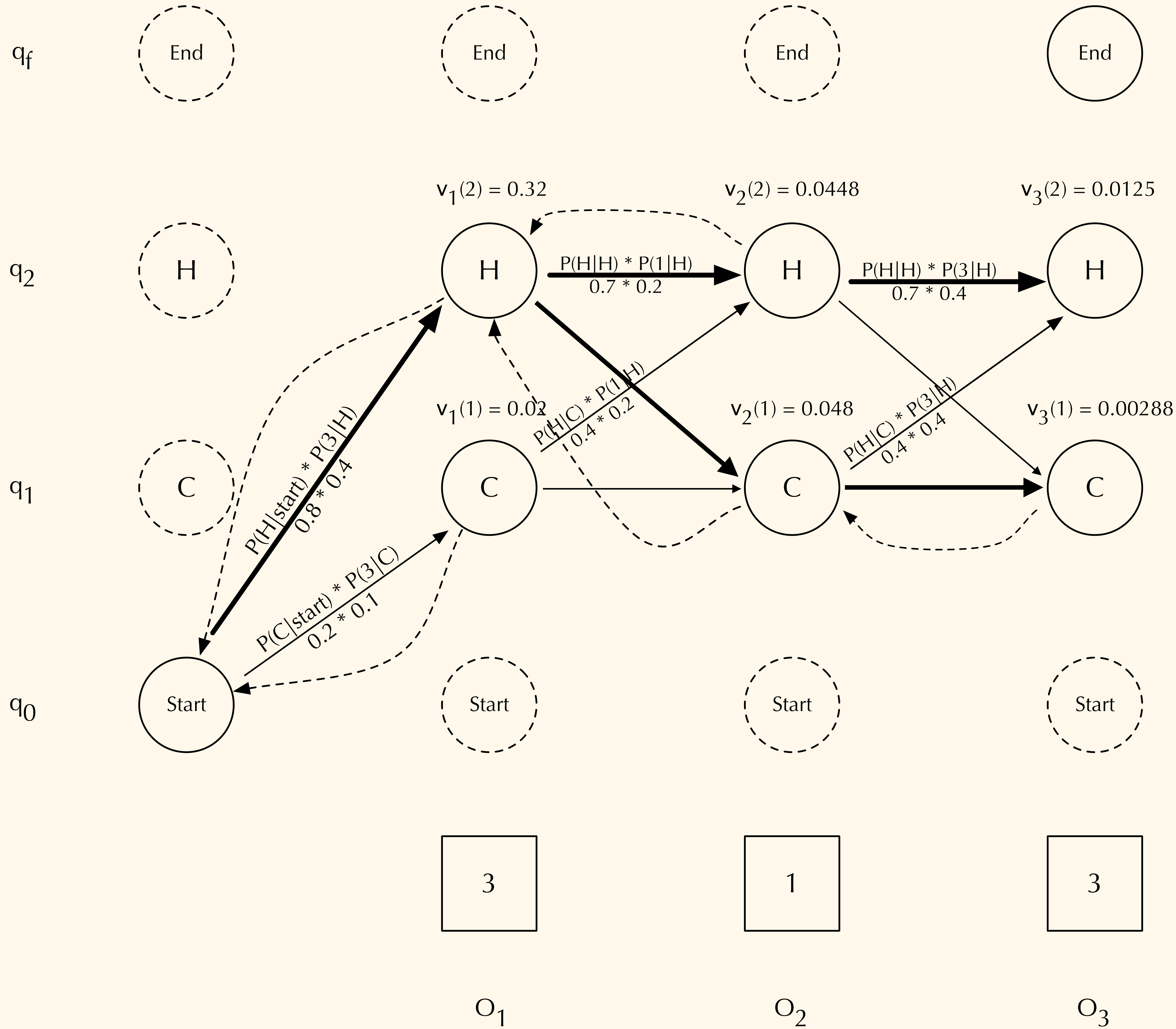
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
v_1(2) = 0.32
v_2(2) = 0.0448
v_1(1) = 0.02
v_2(1) = 0.048
v_3(1) = 0.00288
P(H|H) * P(1|H)
0.7 * 0.2
P(C|H) * P(3|C)
0.3 * 0.1
P(H|C) * P(1|H)
0.4 * 0.2
P(C|C) * P(3|C)
0.6 * 0.1
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
3
1
3
O_1
O_2
O_3



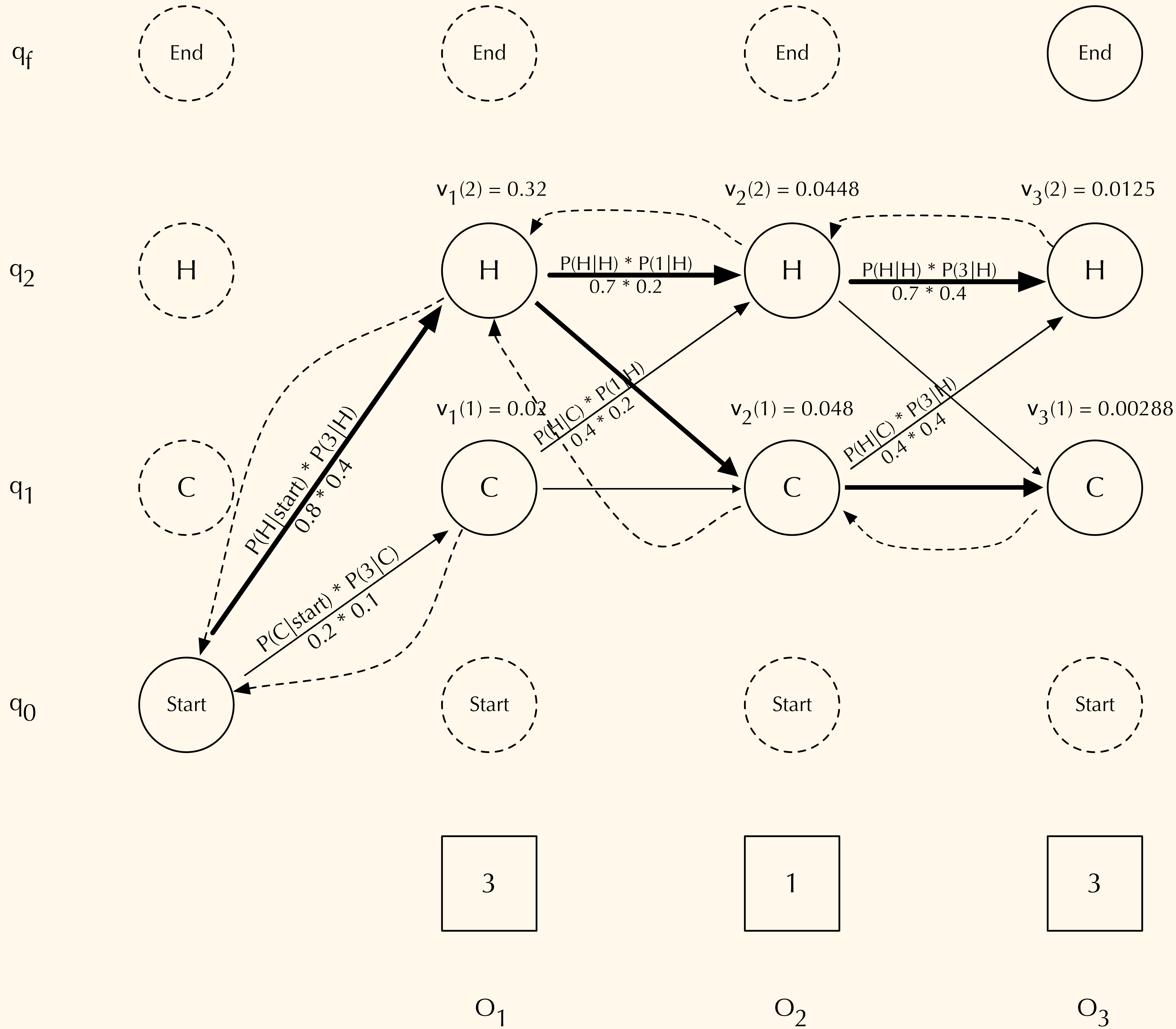
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
$v_1(1) = 0.02$
$v_2(1) = 0.048$
$v_3(1) = 0.00288$
P(H|C) * P(1|H)
0.4 * 0.2
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



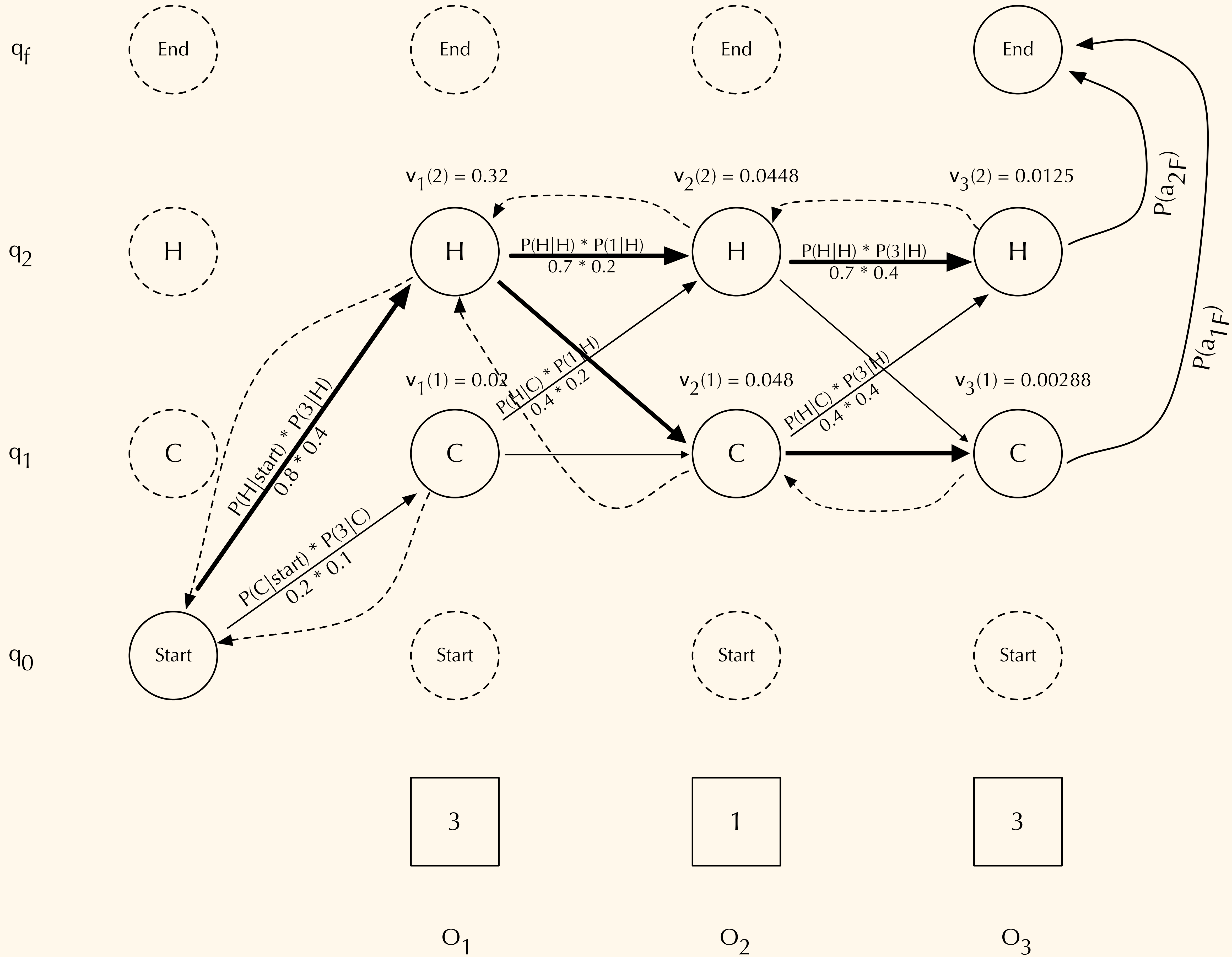
qf
q2
q1
q0
End
End
End
End
v1(2) = 0.32
v2(2) = 0.0448
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(H|H) * P(3|H)
0.7 * 0.4
v1(1) = 0.02
v2(1) = 0.048
v3(1) = 0.00288
P(H|start) * P(3|H)
0.8 * 0.4
P(H|C) * P(1|H)
0.4 * 0.2
P(H|C) * P(3|H)
0.4 * 0.4
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O1
O2
O3



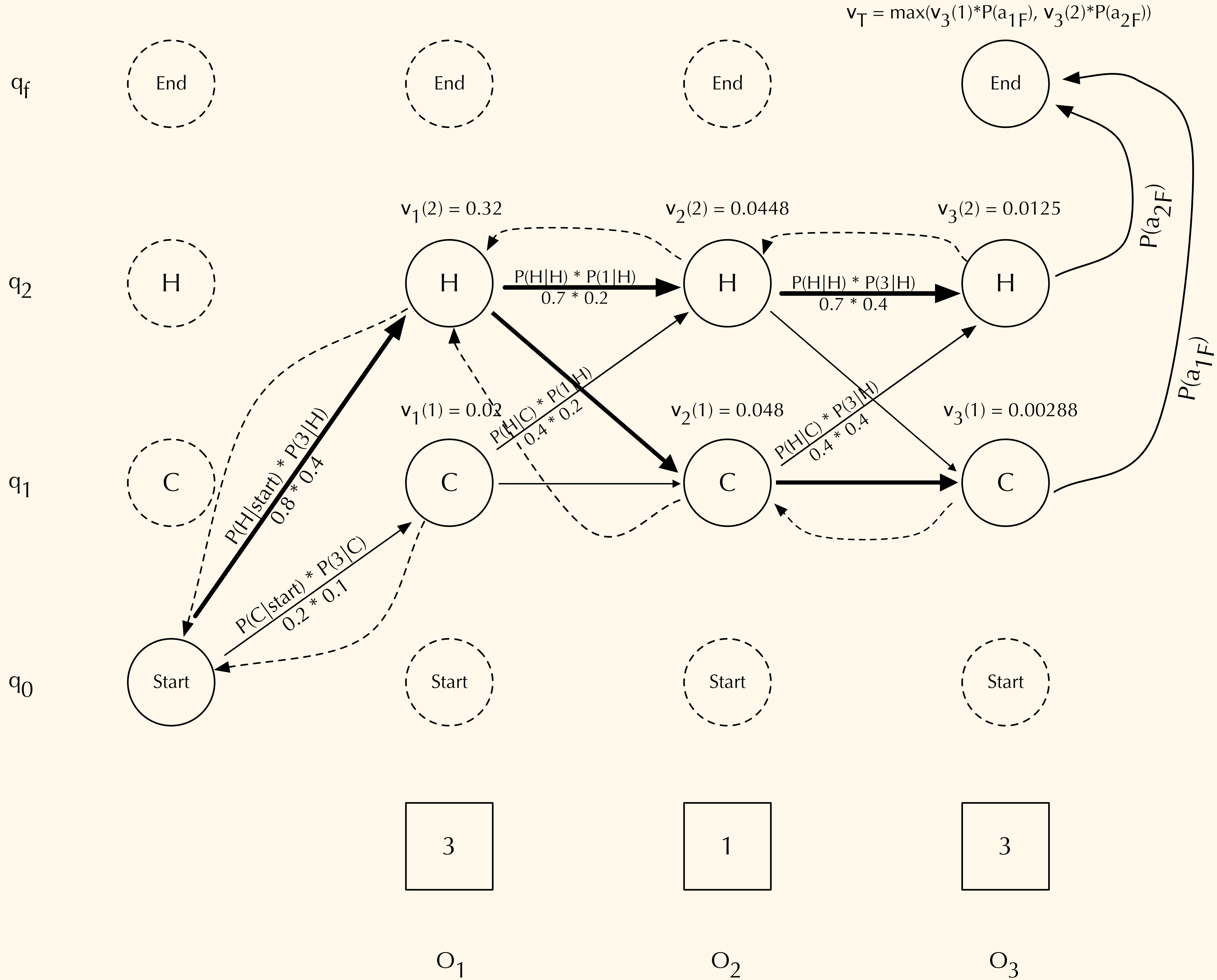
q_f
q_2
q_1
q_0
End
End
End
End
v_1(2) = 0.32
v_2(2) = 0.0448
v_3(2) = 0.0125
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(H|H) * P(3|H)
0.7 * 0.4
v_1(1) = 0.02
v_2(1) = 0.048
v_3(1) = 0.00288
P(H|C) * P(1|H)
0.4 * 0.2
P(H|C) * P(3|H)
0.4 * 0.4
P(H|start) * P(3|H)
0.8 * 0.4
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



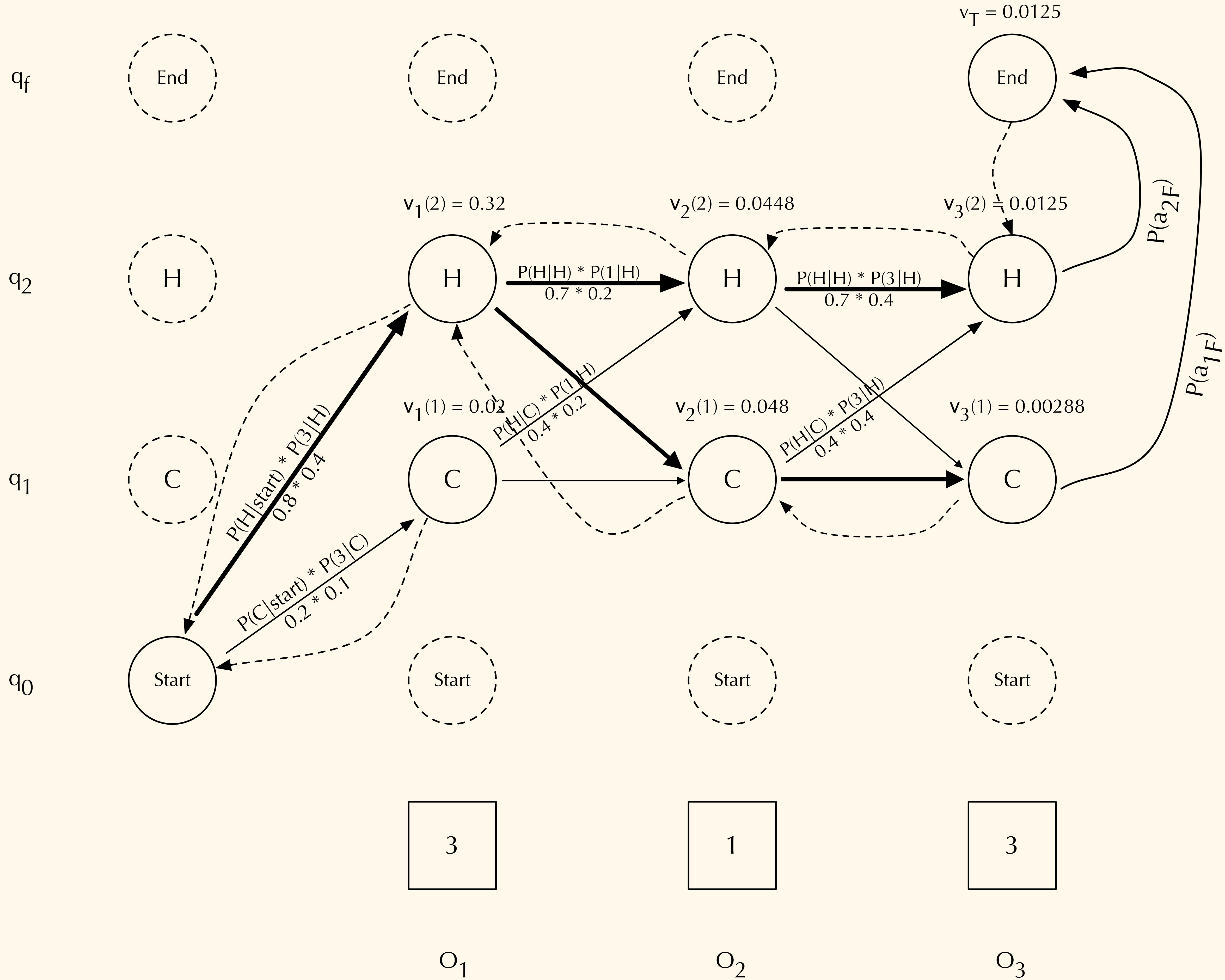
q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
$v_3(2) = 0.0125$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(H|H) * P(3|H)
0.7 * 0.4
$v_1(1) = 0.02$
$v_2(1) = 0.048$
$v_3(1) = 0.00288$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
P(H|C) * P(1|H)
0.4 * 0.2
P(H|C) * P(3|H)
0.4 * 0.4
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



q_f
q_2
q_1
q_0
End
End
End
End
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
$v_3(2) = 0.0125$
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(H|H) * P(3|H)
0.7 * 0.4
$P(a_{2F})$
$P(a_{1F})$
$v_1(1) = 0.02$
$v_2(1) = 0.048$
$v_3(1) = 0.00288$
C
C
C
C
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
P(H|C) * P(1|H)
0.4 * 0.2
P(H|C) * P(3|H)
0.4 * 0.4
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



$v_T = \max(v_3(1)*P(a_{1F}), v_3(2)*P(a_{2F}))$
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$v_1(2) = 0.32$
$v_2(2) = 0.0448$
$v_3(2) = 0.0125$
$v_1(1) = 0.02$
$v_2(1) = 0.048$
$v_3(1) = 0.00288$
P(H|H) * P(1|H)
0.7 * 0.2
P(H|H) * P(3|H)
0.7 * 0.4
P(H|C) * P(1|H)
0.4 * 0.2
P(H|C) * P(3|H)
0.4 * 0.4
P(H|start) * P(3|H)
0.8 * 0.4
P(C|start) * P(3|C)
0.2 * 0.1
$P(a_{2F})$
$P(a_{1F})$
3
1
3
o_1
o_2
o_3



v_T = 0.0125
q_f
End
End
End
End
$v_1(2)$ = 0.32
$v_2(2)$ = 0.0448
$v_3(2)$ = 0.0125
$P(a_{2F})$
q_2
H
H
H
H
P(H|H) * P(1|H)
0.7 * 0.2
P(H|H) * P(3|H)
0.7 * 0.4
$P(a_{1F})$
$v_1(1)$ = 0.02
$v_2(1)$ = 0.048
$v_3(1)$ = 0.00288
P(H|C) * P(1|H)
0.4 * 0.2
P(H|C) * P(3|H)
0.4 * 0.4
P(H|start) * P(3|H)
0.8 * 0.4
q_1
C
C
C
C
P(C|start) * P(3|C)
0.2 * 0.1
q_0
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3

There are three fundamental kinds of questions that we can ask with an HMM:

1. *Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?

2. ***Decoding:*** **Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?**

3. *Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (A and B)?

There are three fundamental kinds of questions that we can ask with an HMM:

1. *Likelihood:* Given a sequence of states, what is the most likely observed sequence? *or* How likely is a given observation sequence?

2. *Decoding*: Given an observation sequence and a fully-specified HMM, what is the most likely sequence of states to have produced that observation?

3. ***Learning:* Given an observation sequence and a set of states, what are the likely transition and emission probabilities (*A* and *B*)?**

So, we have O, and know what our state vocabulary is…

So, we have O, and know what our state vocabulary is…

… but we don't know transition or emission probabilities.

So, we have *O*, and know what our state vocabulary is…

… but we don't know transition or emission probabilities.

We can use *Baum-Welch algorithm* (a.k.a. *Forward-Backward algorithm*) to iteratively estimate *A* and *B*.

So, we have O, and know what our state vocabulary is…

… but we don't know transition or emission probabilities.

We can use *Baum-Welch algorithm* (a.k.a. *Forward-Backward algorithm*) to iteratively estimate A and B.

Leonard Baum

So, we have *O*, and know what our state vocabulary is…

… but we don't know transition or emission probabilities.

We can use *Baum-Welch algorithm* (a.k.a. *Forward-Backward algorithm*) to iteratively estimate *A* and *B*.

Leonard Baum

So, we have O, and know what our state vocabulary is...

... but we don't know transition or emission probabilities.

We can use *Baum-Welch algorithm* (a.k.a. *Forward-Backward algorithm*) to iteratively estimate A and B.

Leonard Baum

Lloyd Welch

Recall from before that the *forward probability* $\alpha_t(i)$ is the probability of ending up in state *i* given observations $O_{1:t}$.

Recall from before that the *forward probability* $\alpha_t(i)$ is the probability of ending up in state *i* given observations $O_{1:t}$.

A related property is the *backward probability* $\beta_t(i)$, which represents the probability of seeing observations $O_{t:T}$, given that we are currently in state *i* at time *t*.

Recall from before that the *forward probability* $\alpha_t(i)$ is the probability of ending up in state *i* given observations $O_{1:t}$.

A related property is the *backward probability* $\beta_t(i)$, which represents the probability of seeing observations $O_{t:T}$, given that we are currently in state *i* at time *t*.

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!).

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

$$\beta_T(i) = a_{i,F}$$

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

$$\beta_T(i) = a_{i,F}$$

Probability of finishing (i.e., reaching end state) the observed sequence from state *i*.

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

$$\beta_T(i) = a_{i,F}$$

Probability of finishing (i.e., reaching end state) the observed sequence from state *i*.

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

$$\beta_T(i) = a_{i,F}$$

Probability of finishing (i.e., reaching end state) the observed sequence from state *i*.

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

Sum of the backwards probabilities of the different paths through the model that could happen from state *i* and time *t*.

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

$$\beta_T(i) = a_{i,F}$$

Probability of finishing (i.e., reaching end state) the observed sequence from state *i*.

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

Sum of the backwards probabilities of the different paths through the model that could happen from state *i* and time *t*.

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{N} a_{0j} b_j(o_1) \beta_1(j)$$

This is calculated using the *backward algorithm*, which is very similar to the forward algorithm (but in reverse!)

$$\beta_T(i) = a_{i,F}$$

Probability of finishing (i.e., reaching end state) the observed sequence from state *i*.

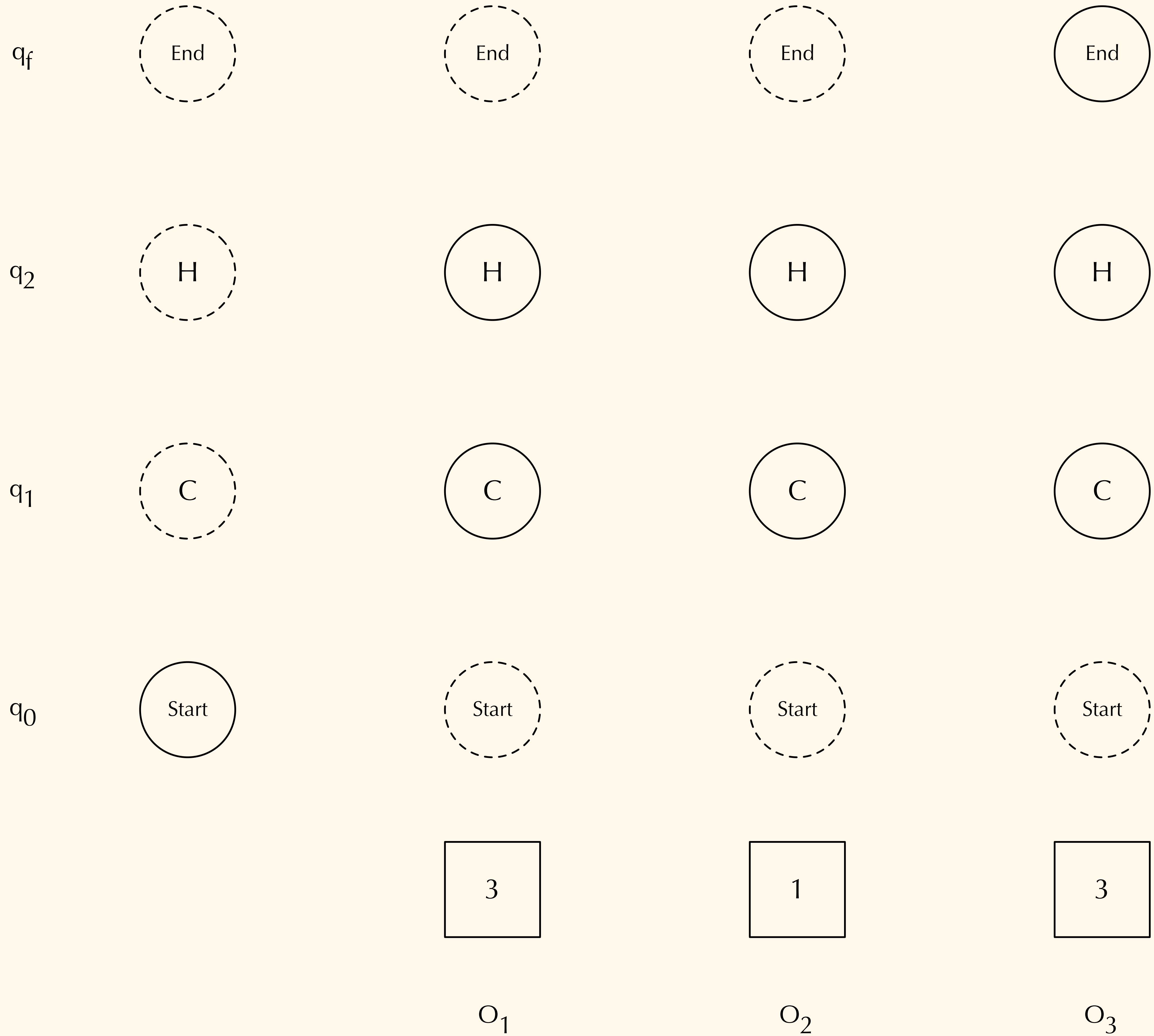
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

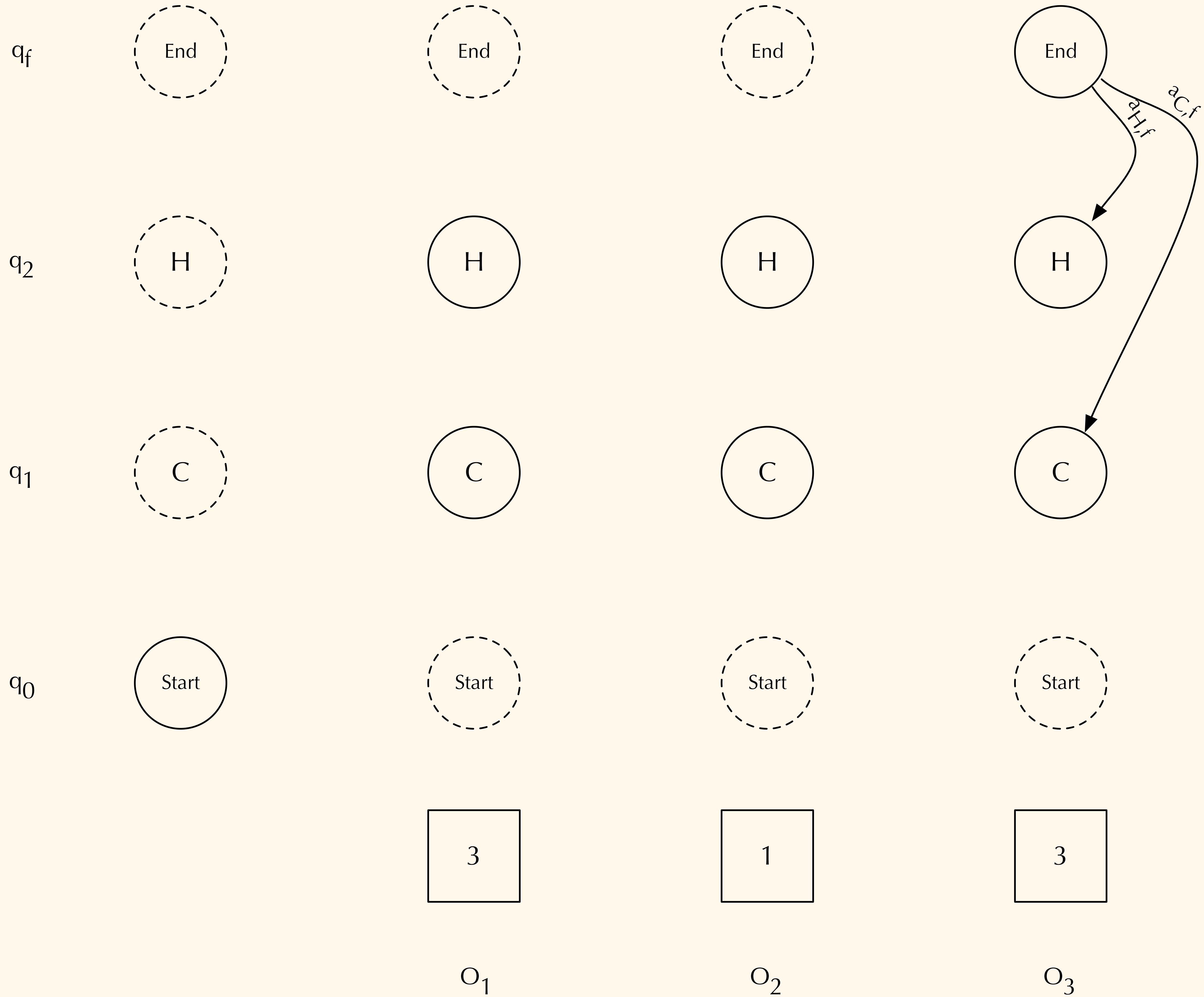
Sum of the backwards probabilities of the different paths through the model that could happen from state *i* and time *t*.

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{N} a_{0j} b_j(o_1) \beta_1(j)$$

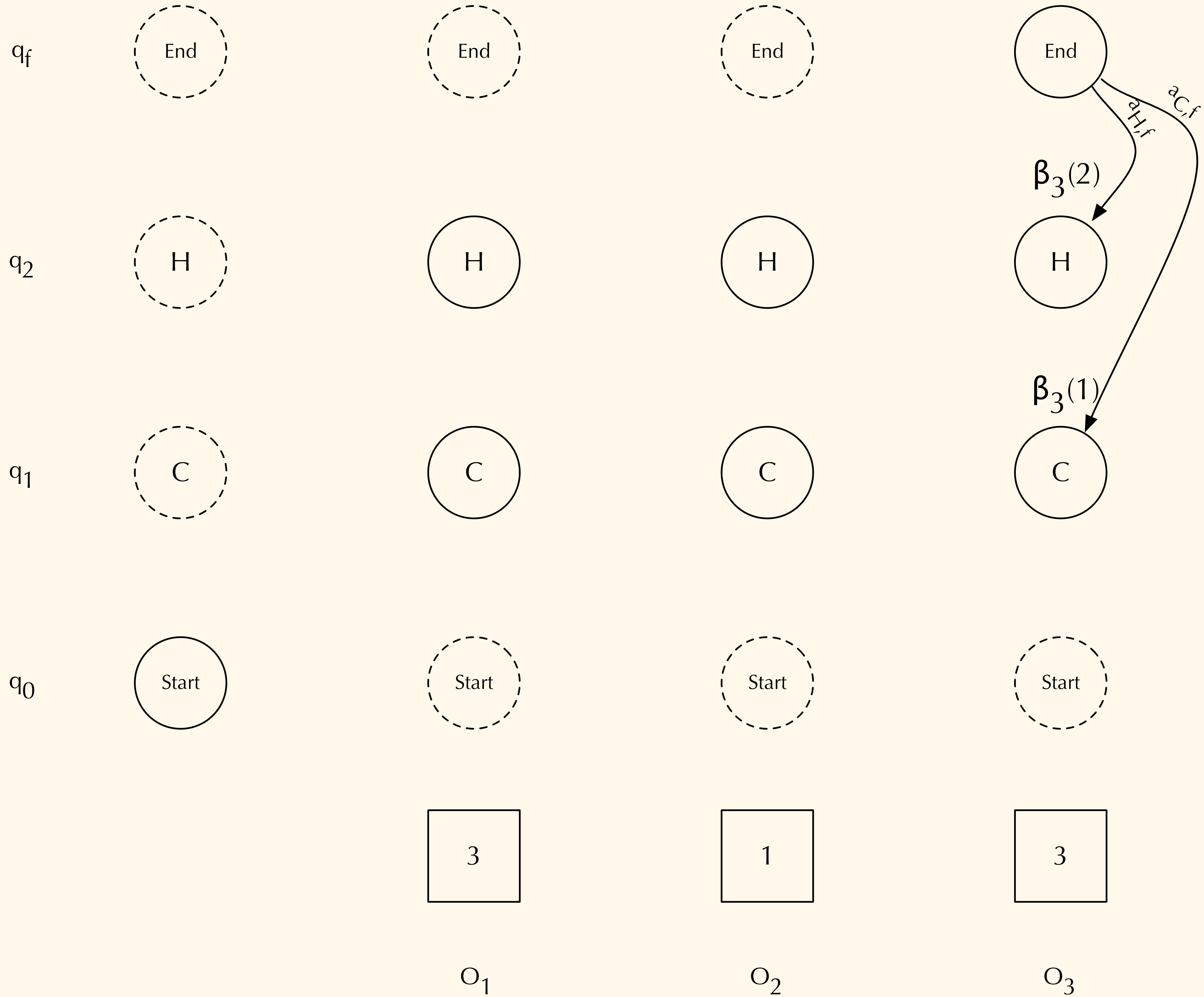
Final forward probability of observation given model.

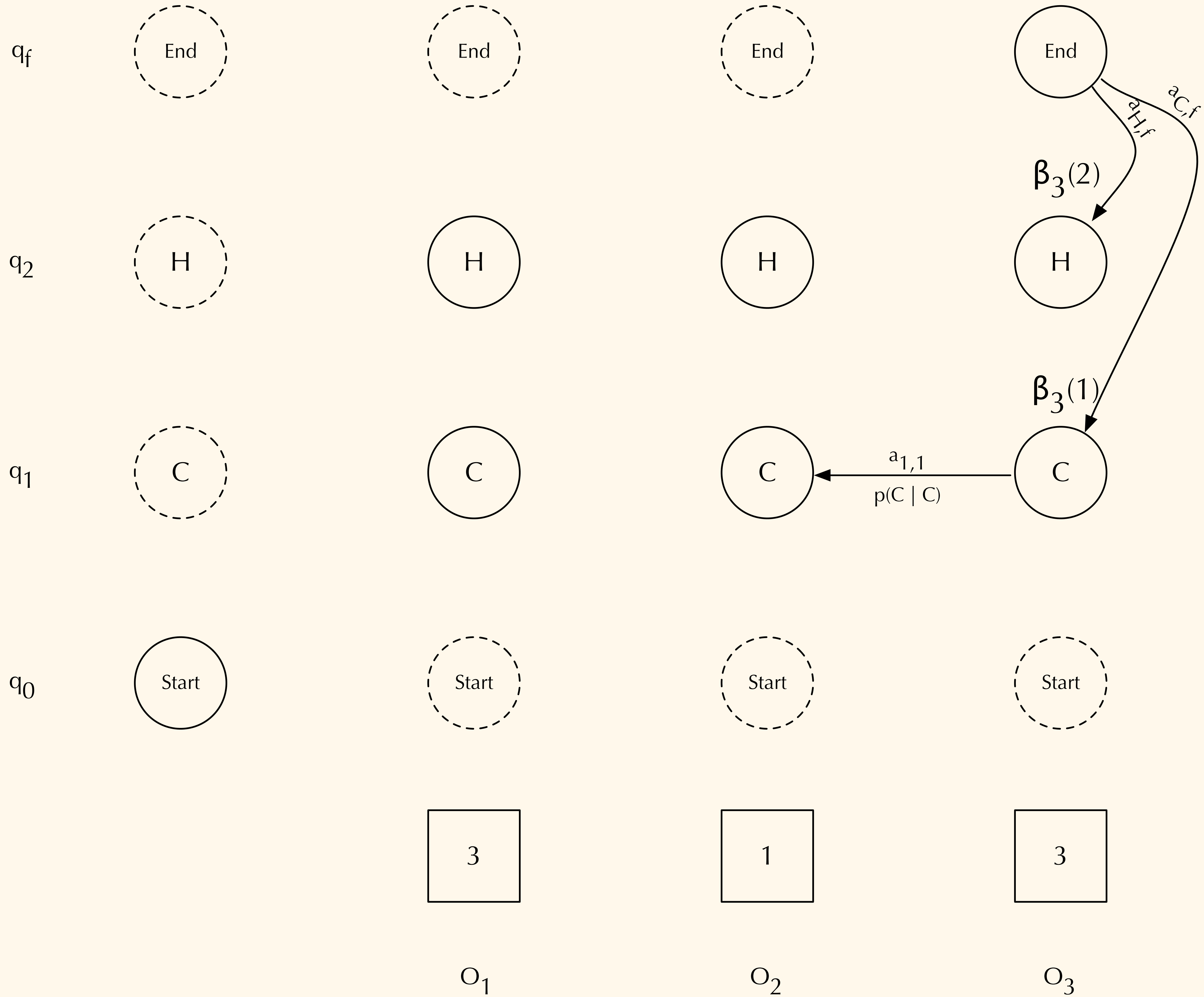
Of course, we compute all of this using the same dynamic programming approach we've already seen:


q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3

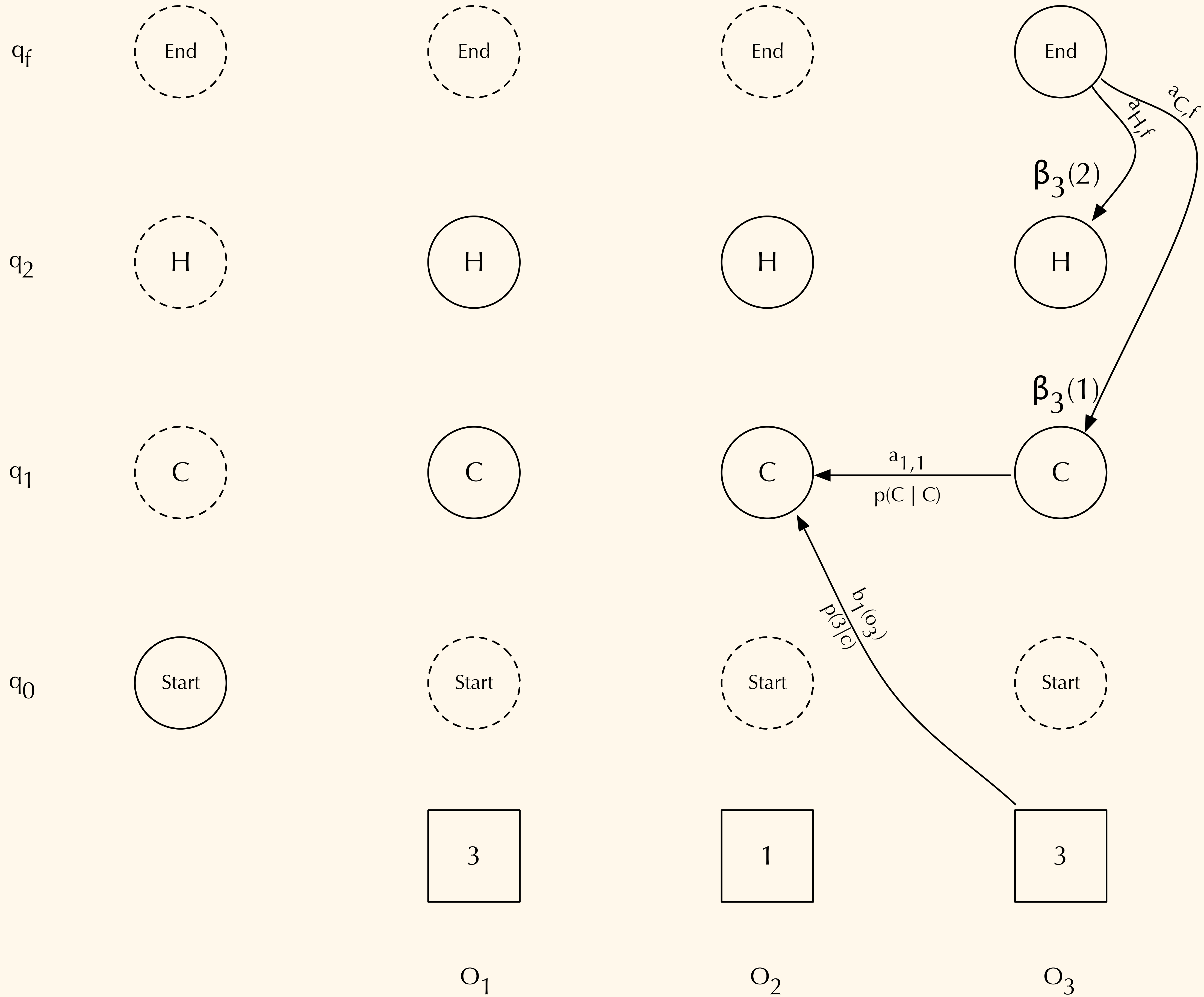


q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$a_{H,f}$
$a_{C,f}$
3
1
3
o_1
o_2
o_3

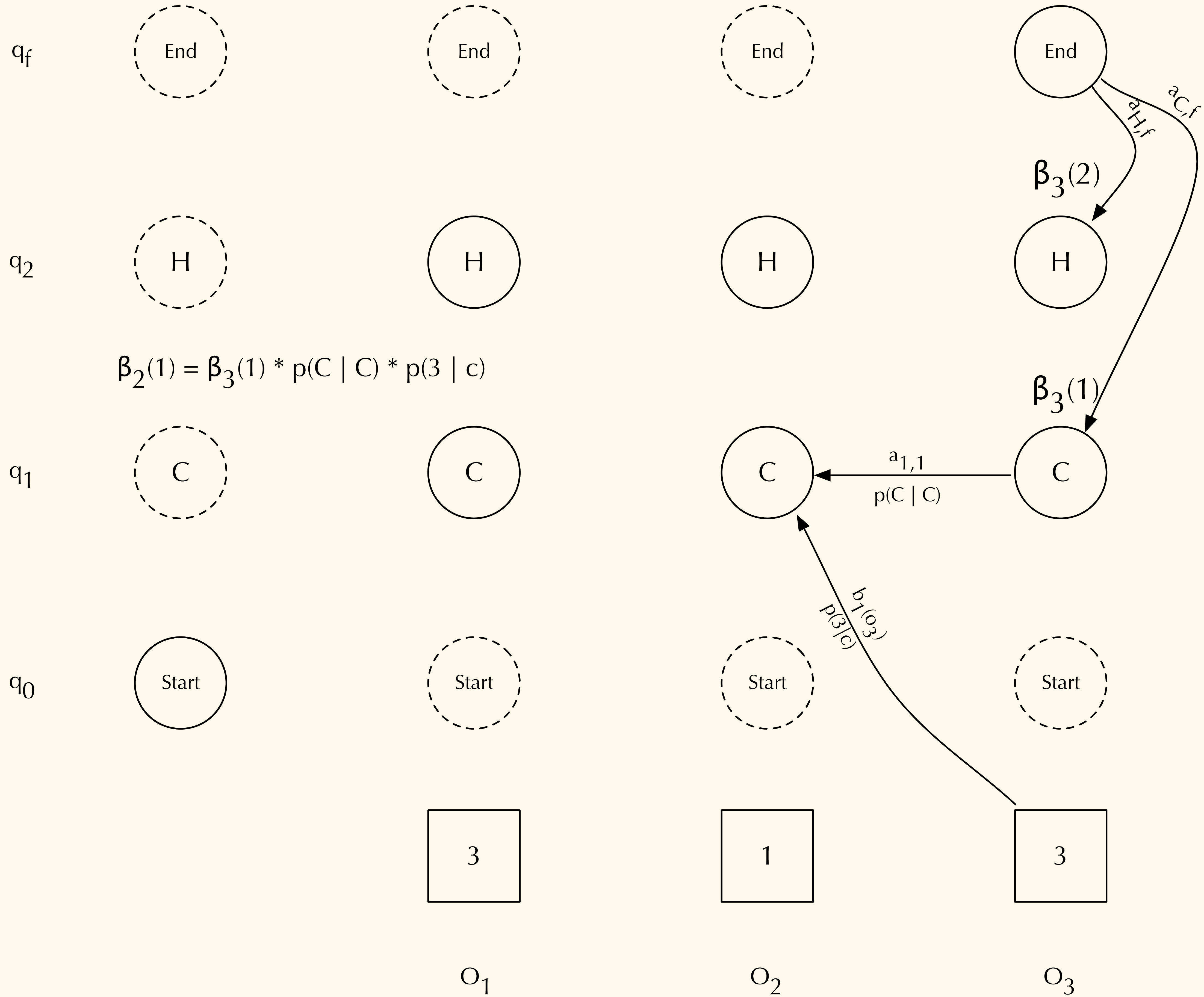

q_f
q_2
q_1
q_0
End
End
End
End
$a_{H,f}$
$a_{C,f}$
$\beta_3(2)$
H
H
H
H
$\beta_3(1)$
C
C
C
C
Start
Start
Start
Start
3
1
3
o_1
o_2
o_3



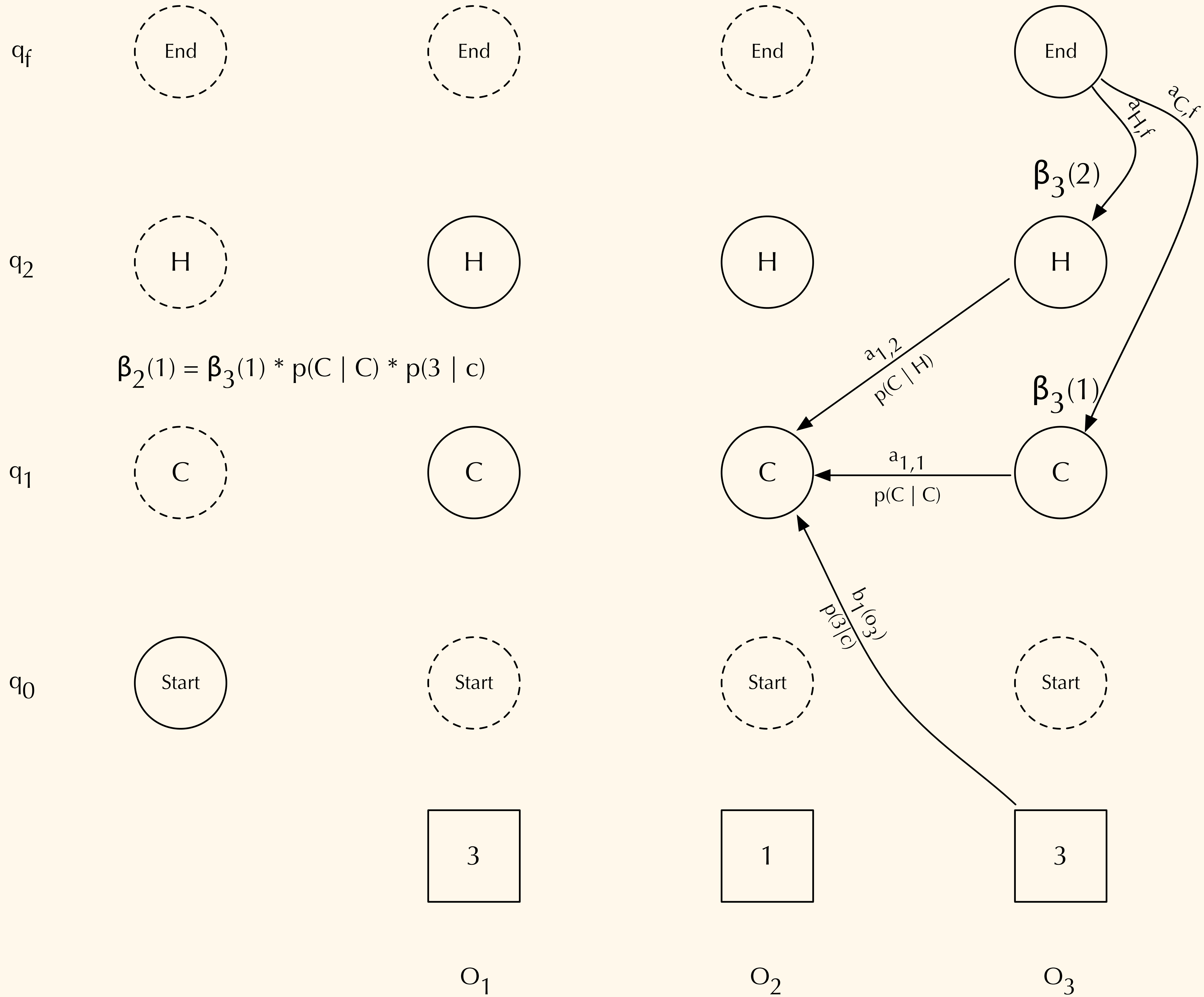
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$a_{H,f}$
$a_{C,f}$
$\beta_3(2)$
$\beta_3(1)$
$a_{1,1}$
p(C | C)
3
1
3
O_1
O_2
O_3



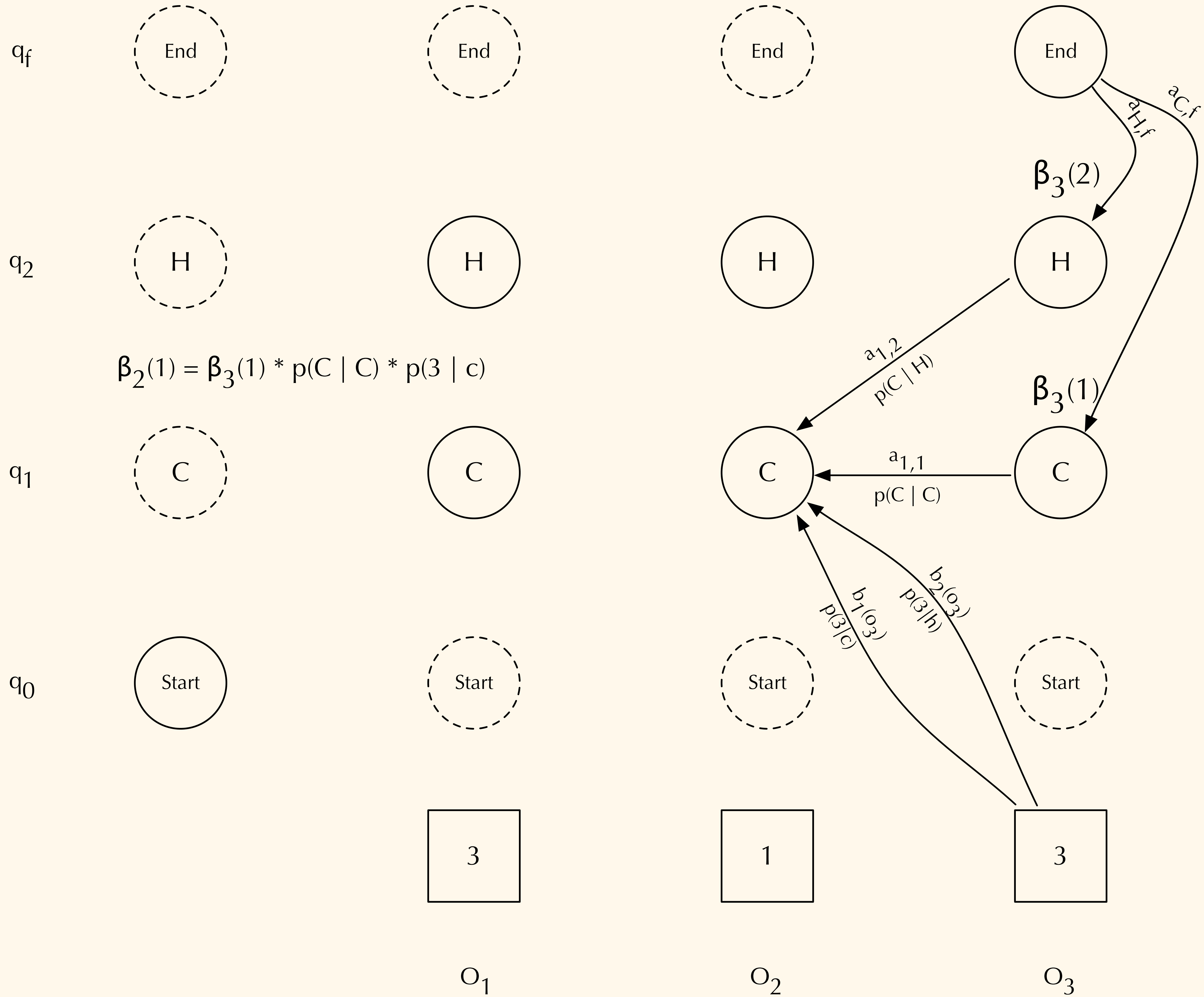
q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$a_{H,f}$
$a_{C,f}$
$\beta_3(2)$
$\beta_3(1)$
$a_{1,1}$
p(C | C)
$b_1(o_3)$
p(3|c)
3
1
3
O_1
O_2
O_3

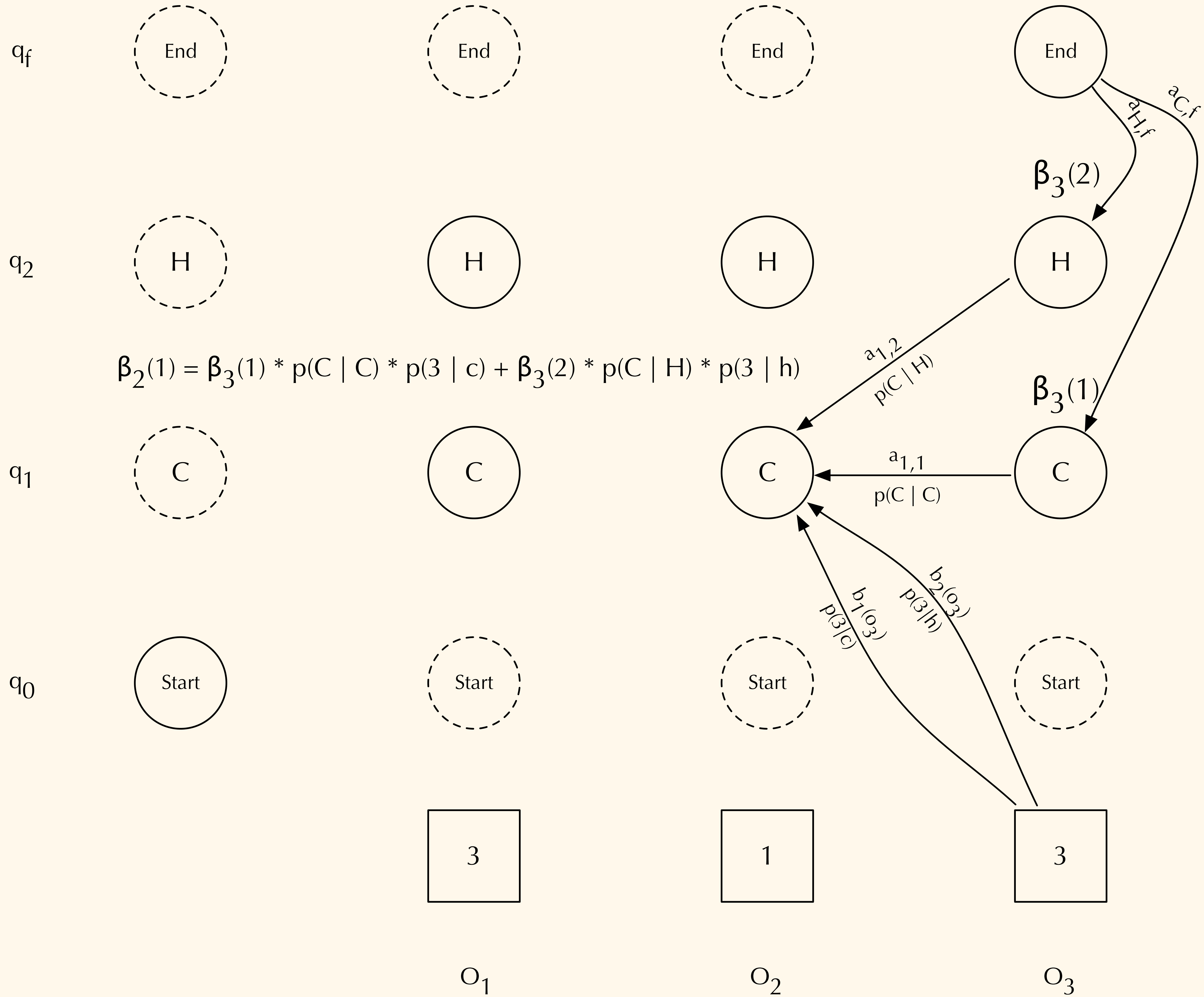


q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$a_{H,f}$
$a_{C,f}$
$\beta_3(2)$
$\beta_3(1)$
$\beta_2(1) = \beta_3(1) * p(C \mid C) * p(3 \mid c)$
$a_{1,1}$
p(C | C)
$b_1(o_3)$
p(3|c)
3
1
3
O_1
O_2
O_3



q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
$\beta_3(2)$
$a_{H,f}$
$a_{C,f}$
$\beta_2(1) = \beta_3(1) * p(C \mid C) * p(3 \mid c)$
$a_{1,2}$
$p(C \mid H)$
$\beta_3(1)$
C
C
C
C
$a_{1,1}$
$p(C \mid C)$
$b_1(o_3)$
$p(3|c)$
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3


q_f
End
End
End
End
$a_{H,f}$
$a_{C,f}$
$\beta_3(2)$
q_2
H
H
H
H
$\beta_2(1) = \beta_3(1) * p(C \mid C) * p(3 \mid c)$
$a_{1,2}$
p(C | H)
$\beta_3(1)$
q_1
C
C
C
C
$a_{1,1}$
p(C | C)
$b_1(o_3)$
p(3|c)
$b_2(o_3)$
p(3|h)
q_0
Start
Start
Start
Start
3
1
3
O_1
O_2
O_3



q_f
q_2
q_1
q_0
End
End
End
End
H
H
H
H
C
C
C
C
Start
Start
Start
Start
$a_{H,f}$
$a_{C,f}$
$\beta_3(2)$
$\beta_3(1)$
$\beta_2(1) = \beta_3(1) * p(C \mid C) * p(3 \mid c) + \beta_3(2) * p(C \mid H) * p(3 \mid h)$
$a_{1,2}$
$p(C \mid H)$
$a_{1,1}$
$p(C \mid C)$
$b_1(o_3)$
$p(3|c)$
$b_2(o_3)$
$p(3|h)$
3
1
3
O_1
O_2
O_3

Now, how to use these to estimate A and B?

Now, how to use these to estimate A and B?

Baum-Welch is a variation on *Expectation-Maximization*...

Now, how to use these to estimate A and B?

Baum-Welch is a variation on *Expectation-Maximization*…

… as such, we start with a "guess" for A and B, and iteratively improve it.

Now, how to use these to estimate A and B?

Baum-Welch is a variation on *Expectation-Maximization*…

… as such, we start with a “guess” for A and B, and iteratively improve it.

We begin by attempting to find:

Now, how to use these to estimate *A* and *B*?

Baum-Welch is a variation on *Expectation-Maximization*...

... as such, we start with a "guess" for A and B, and iteratively improve it.

We begin by attempting to find:

$$\hat{a}_{ij} = \frac{\text{expected \# transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

Now, how to use these to estimate A and B?

Baum-Welch is a variation on *Expectation-Maximization*…

… as such, we start with a "guess" for A and B, and iteratively improve it.

We begin by attempting to find:

$$\hat{a}_{ij} = \frac{\text{expected \# transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$

If we had an estimate of the probability of transition $i \rightarrow j$ occurring at each time t, we could sum them to get the total count for $i \rightarrow j$.

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

We can't quite calculate this, but we *can* calculate something similar:

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

We can't quite calculate this, but we *can* calculate something similar:

$$\tilde{\xi}_t(i,j) = P(q_t = i, q_{t+1} = j, O | \lambda)$$

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

We can't quite calculate this, but we *can* calculate something similar:

$$\tilde{\xi}_t(i,j) = P(q_t = i, q_{t+1} = j, O | \lambda)$$

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

We can't quite calculate this, but we *can* calculate something similar:

$$\tilde{\xi}_t(i,j) = P(q_t = i, q_{t+1} = j, O | \lambda)$$

We have all the pieces we need to get this:

More formally: let $\xi_t(i,j)$ be the probability of being in state *i* at time *t* and in state *j* at time *t+1*.

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda)$$

We can't quite calculate this, but we *can* calculate something similar:

$$\tilde{\xi}_t(i,j) = P(q_t = i, q_{t+1} = j, O | \lambda)$$

We have all the pieces we need to get this:

$$\tilde{\xi}_t(i,j) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

We have all the pieces we need to get this:

$$\tilde{\xi}_t(i,j) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

We have all the pieces we need to get this:

$$\tilde{\xi}_t(i,j) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

Forward probability of
observations up to this arc

We have all the pieces we need to get this:

$$\tilde{\xi}_t(i,j) = \underbrace{\alpha_t(i)}_{\text{Forward probability of observations up to this arc}} \underbrace{a_{ij}}_{\text{Transition probability between states } i \text{ and } j} b_j(o_{t+1})\beta_{t+1}(j)$$

We have all the pieces we need to get this:

$$\tilde{\xi}_t(i,j) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

Forward probability of observations up to this arc

Transition probability between states *i* and *j*

Emission probability of the next symbol

We have all the pieces we need to get this:

$$\tilde{\xi}_t(i,j) = \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

Forward probability of observations up to this arc

Transition probability between states *i* and *j*

Emission probability of the next symbol

Backward probability after this arc

Furthermore, because of $P(X|Y,Z) = \dfrac{P(X,Y|Z)}{P(Y|Z)}$

Furthermore, because of $P(X|Y,Z) = \dfrac{P(X,Y|Z)}{P(Y|Z)}$

We can transform $\tilde{\xi}_t(i,j)$, or $P(q_t = i, q_{t+1} = j, O|\lambda)$

into $\xi_t(i,j)$, or $P(q_t = i, q_{t+1} = j|O,\lambda)$, simply by dividing by $P(O|\lambda)$.

Furthermore, because of $P(X|Y,Z) = \dfrac{P(X,Y|Z)}{P(Y|Z)}$

We can transform $\tilde{\xi}_t(i,j)$, or $P(q_t = i, q_{t+1} = j, O|\lambda)$

into $\xi_t(i,j)$, or $P(q_t = i, q_{t+1} = j|O,\lambda)$, simply by dividing by $P(O|\lambda)$.

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{N} a_{0j} b_j(o_1) \beta_1(j)$$

Furthermore, because of $P(X|Y,Z) = \dfrac{P(X,Y|Z)}{P(Y|Z)}$

We can transform $\tilde{\xi}_t(i,j)$, or $P(q_t = i, q_{t+1} = j, O|\lambda)$

into $\xi_t(i,j)$, or $P(q_t = i, q_{t+1} = j|O,\lambda)$, simply by dividing by $P(O|\lambda)$.

$$P(O|\lambda) = \alpha_T(q_F) = \beta_1(0) = \sum_{j=1}^{N} a_{0j} b_j(o_1)\beta_1(j)$$

So, the final equation is:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(N)}$$

Remember:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\alpha_T(N)}$$

"The probability of going from state *i* to state *j* at time *t*." (given a current estimate of the model).

Remember:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(N)}$$

"The probability of going from state *i* to state *j* at time *t*." (given a current estimate of the model).

Summing over all times *t* gives us $\hat{a}_{ij}$:

Remember:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\alpha_T(N)}$$

"The probability of going from state *i* to state *j* at time *t*." (given a current estimate of the model).

Summing over all times *t* gives us $\hat{a}_{ij}$:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)}$$

Remember:

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\alpha_T(N)}$$

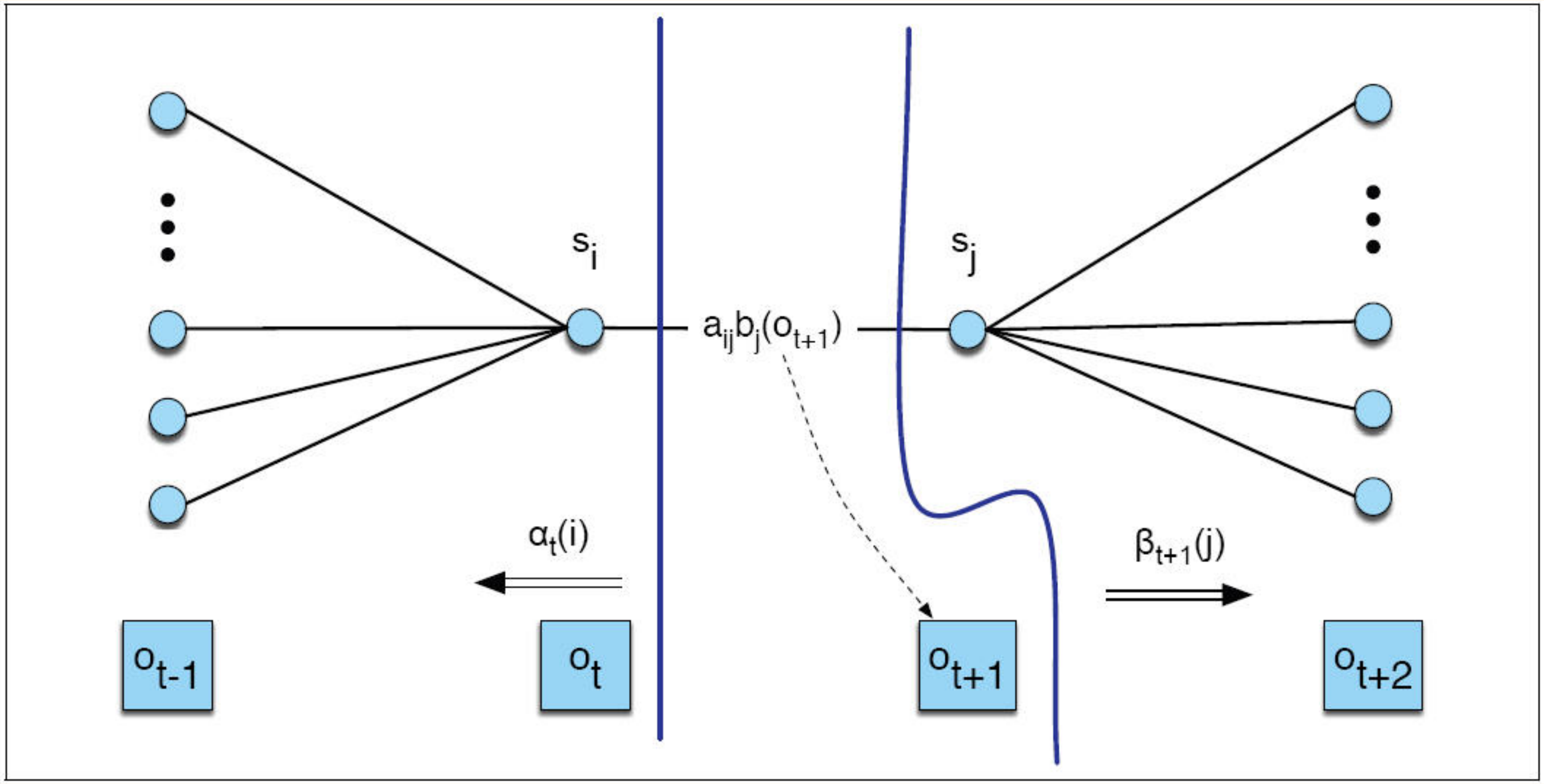
"The probability of going from state *i* to state *j* at time *t*." (given a current estimate of the model).

Summing over all times *t* gives us $\hat{a}_{ij}$:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)}$$

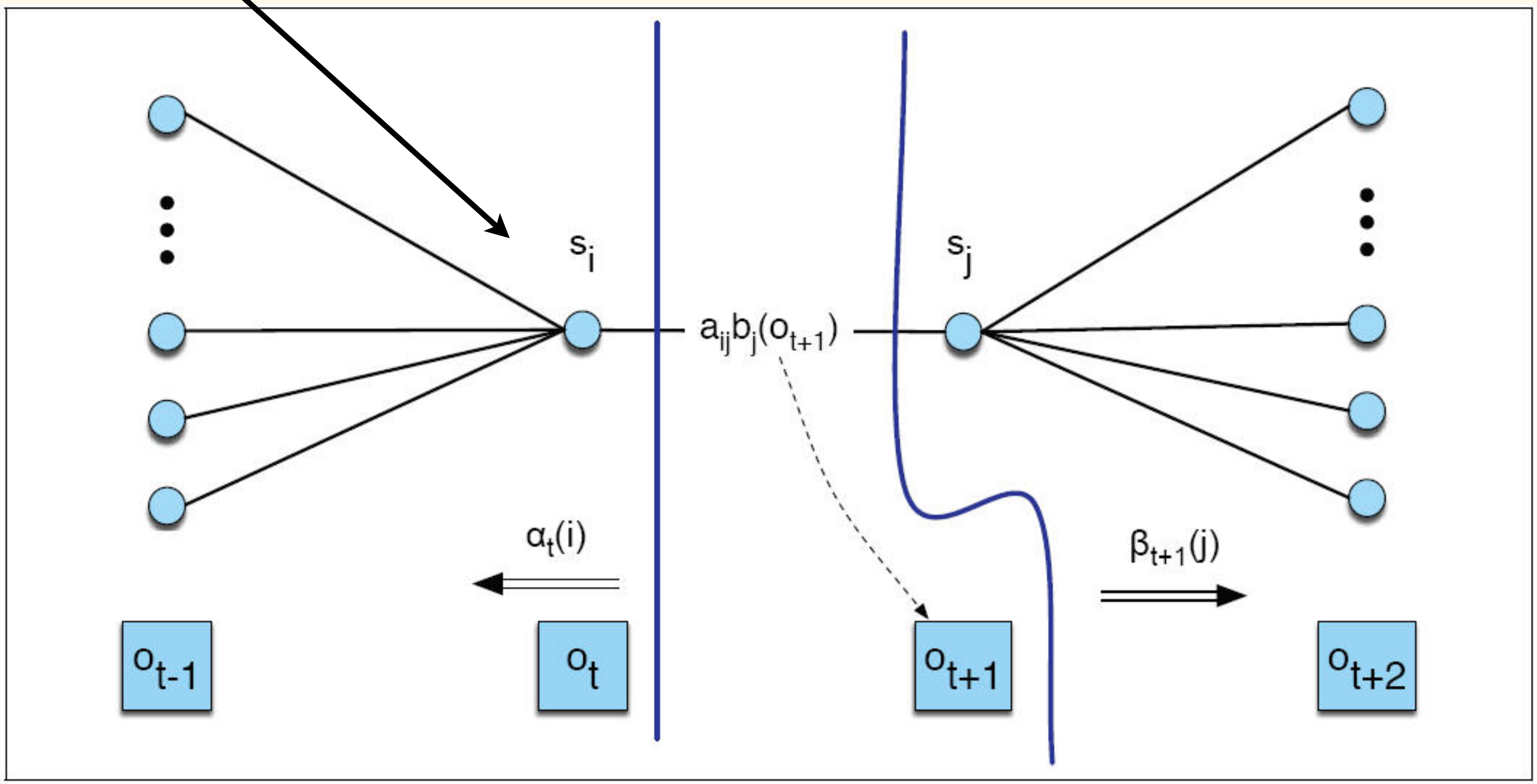
Which looks a lot like:

$$\hat{a}_{ij} = \frac{\text{expected } \# \text{ transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}$$



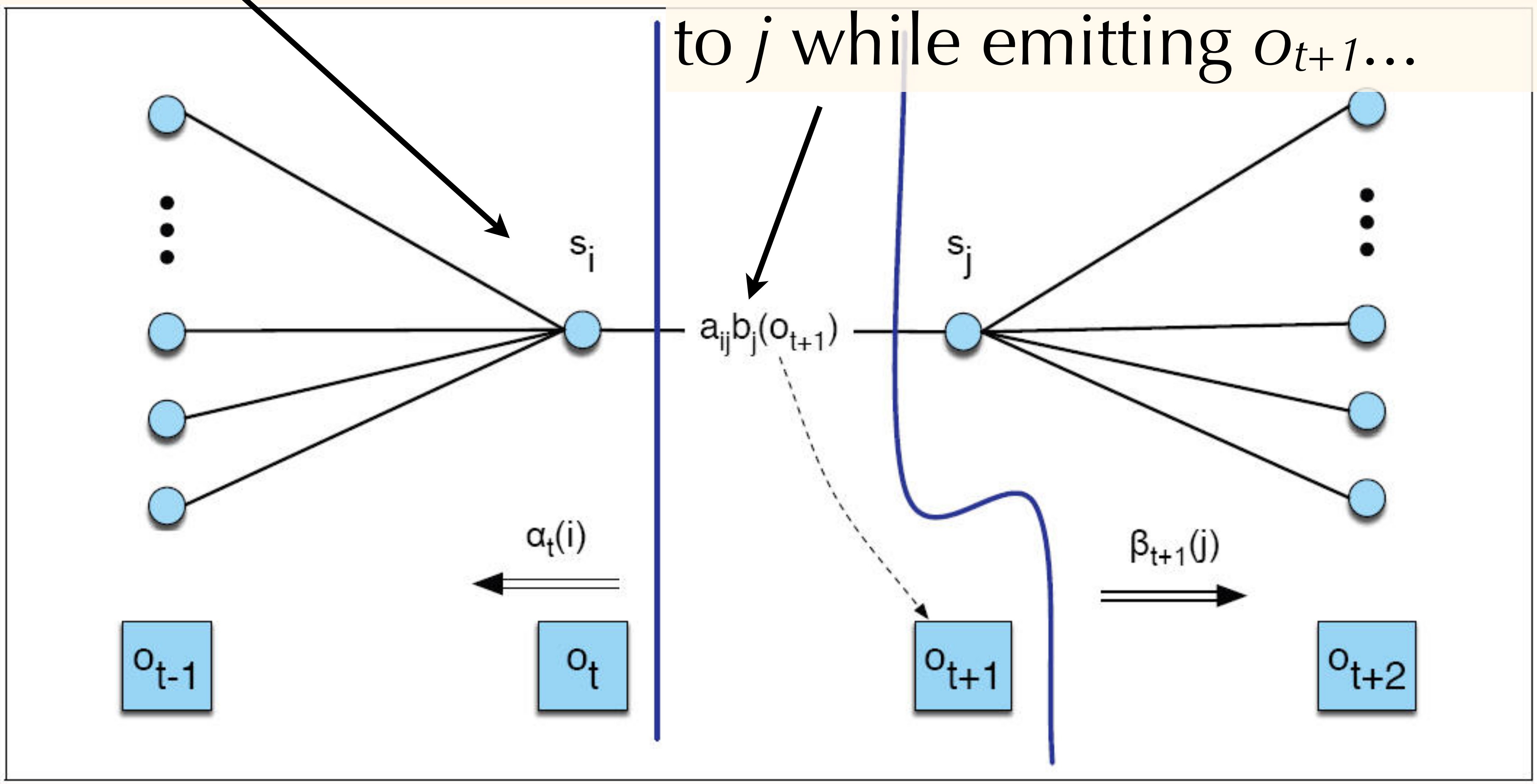
s_i
s_j
a_ij b_j(o_t+1)
α_t(i)
β_t+1(j)
o_t-1
o_t
o_t+1
o_t+2

All of the ways the model could have gotten into state *i* at time *t*...



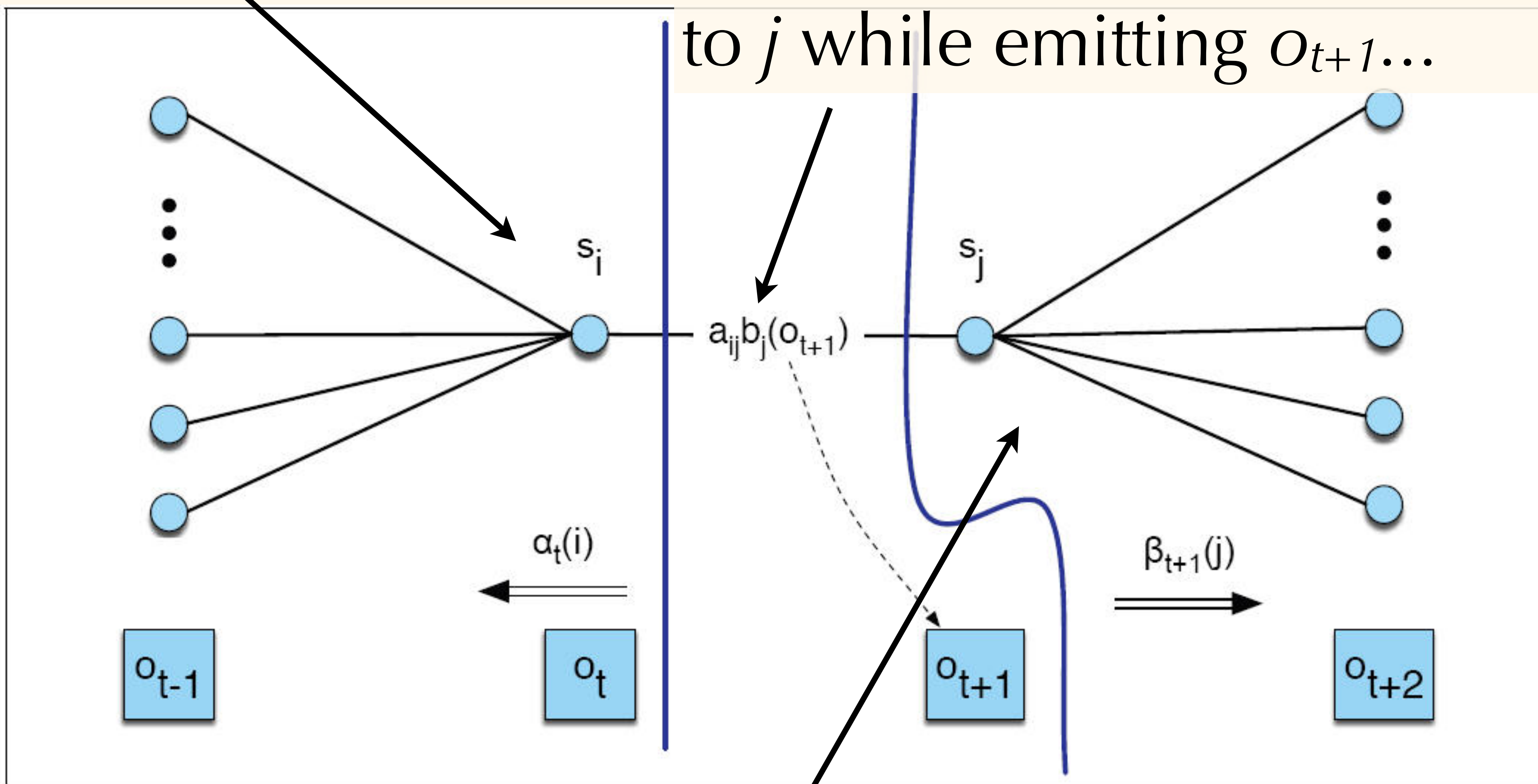
All of the ways the model could have gotten into state *i* at time *t*...

... the likelihood of going from *i* to *j* while emitting o_{t+1}...



All of the ways the model could have gotten into state *i* at time *t*...

... the likelihood of going from *i* to *j* while emitting o_{t+1}...



... all of the ways the model could finish from state *j* at time *t+1*.

We follow a similar process to estimate B.

We follow a similar process to estimate B.

$$\hat{b}_j(v_k) = \frac{\text{expected \# times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

We follow a similar process to estimate B.

$$\hat{b}_j(v_k) = \frac{\text{expected \# times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

We'll need to know the probability of being in state j at time t:

We follow a similar process to estimate B.

$$\hat{b}_j(v_k) = \frac{\text{expected \# times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

We'll need to know the probability of being in state *j* at time *t*:

$$\gamma_t(j) = P(q_t = j | O, \lambda)$$

We follow a similar process to estimate B.

$$\hat{b}_j(v_k) = \frac{\text{expected \# times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

We'll need to know the probability of being in state *j* at time t:

$$\gamma_t(j) = P(q_t = j | O, \lambda)$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)}$$

We follow a similar process to estimate B.

$$\hat{b}_j(v_k) = \frac{\text{expected \# times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

We'll need to know the probability of being in state *j* at time *t*:

$$\gamma_t(j) = P(q_t = j | O, \lambda)$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

We follow a similar process to estimate B.

$$\hat{b}_j(v_k) = \frac{\text{expected \# times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j}$$

We'll need to know the probability of being in state *j* at time *t*:

$$\gamma_t(j) = P(q_t = j | O, \lambda)$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

"Probability of getting to this state at this time point, times the probability of the rest of the observations given this state and this time point"

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)}$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)}$$

"Probability of getting to this state at this time point, times the probability of the rest of the observations given this state and this time point"

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)}$$

"Probability of getting to this state at this time point, times the probability of the rest of the observations given this state and this time point"

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 \text{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)}$$

"Probability of getting to this state at this time point, times the probability of the rest of the observations given this state and this time point"

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 \text{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Using the same trick as before:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \qquad P(q_t = j, O|\lambda) = \alpha_t(j)\beta_t(j)$$

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)}$$

"Probability of getting to this state at this time point, times the probability of the rest of the observations given this state and this time point"

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 \mathrm{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

"Only count observations where the observed emission was v_k."

Now, that we have new estimates for *A* and *B*...

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)} \quad \hat{b}_j(v_k) = \frac{\sum_{t=1 \mathrm{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Now, that we have new estimates for *A* and *B*...

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)} \quad \hat{b}_j(v_k) = \frac{\sum_{t=1 \mathrm{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

We can go back, calculate *new* forward and backward trellises, and re-compute *A* and *B*.

Now, that we have new estimates for *A* and *B*...

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)} \quad \hat{b}_j(v_k) = \frac{\sum_{t=1 \mathrm{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

We can go back, calculate *new* forward and backward trellises, and re-compute *A* and *B*.

Rinse, wash, and repeat until things converge or we get bored.

Now, that we have new estimates for A and B...

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)} \quad \hat{b}_j(v_k) = \frac{\sum_{t=1 \mathrm{s.t.} O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

We can go back, calculate *new* forward and backward trellises, and re-compute A and B.

Rinse, wash, and repeat until things converge or we get bored.

In practice, much depends on our initial estimates, and so we often use additional information when possible (e.g., encoding impossible transitions, etc.).