In [1]:
```python
import numpy as np
import pandas as pd

# ----------------------------------------
# HMM Definition
# ----------------------------------------

states = ['s', 'p', 'ie', 'tS']  # /s/, /p/, /ie:/, /tʃ/
observations = ['lowE', 'highE', 'highPitch', 'longDur']

# Start probability: starting at /s/ = 1
start_prob = np.array([1.0, 0.0, 0.0, 0.0])

# Transition probabilities (your table exactly)
trans_prob = np.array([
    [0.1, 0.8, 0.1, 0.0],  # from /s/
    [0.0, 0.1, 0.8, 0.1],  # from /p/
    [0.0, 0.0, 0.2, 0.8],  # from /ie:/
    [0.2, 0.0, 0.0, 0.8]   # from /tS/
])

# Example emission probabilities (replace with yours)
emit_prob = np.array([
    [0.1, 0.6, 0.2, 0.1],  # /s/
    [0.05, 0.7, 0.1, 0.15],  # /p/
    [0.05, 0.2, 0.3, 0.45],  # /ie/
    [0.1, 0.6, 0.15, 0.15]   # /tS/
])


# ----------------------------------------
# Sampling helpers
# ----------------------------------------

def sample_from_dist(p):
    return np.searchsorted(np.cumsum(p), np.random.rand())


# ----------------------------------------
# Simulate hidden + observed sequence
# ----------------------------------------

def simulate_sequence(max_steps=12):
    hidden = []
    observed = []

    current = sample_from_dist(start_prob)
    hidden.append(current)
    observed.append(sample_from_dist(emit_prob[current]))

    for _ in range(max_steps - 1):
        current = sample_from_dist(trans_prob[current])
        hidden.append(current)
```

```
        observed.append(sample_from_dist(emit_prob[current]))

    return hidden, observed


# ----------------------------------------
# Viterbi Algorithm
# ----------------------------------------

def viterbi(obs_seq):
    N = len(states)
    T = len(obs_seq)

    log_start = np.log(start_prob + 1e-12)
    log_trans = np.log(trans_prob + 1e-12)
    log_emit = np.log(emit_prob + 1e-12)

    dp = np.full((N, T), -np.inf)
    backpointer = np.zeros((N, T), dtype=int)

    # Initialization
    dp[:, 0] = log_start + log_emit[:, obs_seq[0]]

    # Recursion
    for t in range(1, T):
        for s in range(N):
            probs = dp[:, t-1] + log_trans[:, s] + log_emit[s, obs_seq[t]]
            backpointer[s, t] = np.argmax(probs)
            dp[s, t] = np.max(probs)

    # Termination
    last_state = np.argmax(dp[:, -1])

    path = [last_state]
    for t in range(T - 1, 0, -1):
        last_state = backpointer[last_state, t]
        path.append(last_state)

    return path[::-1]


# ----------------------------------------
# Forward Algorithm
# ----------------------------------------

def forward(obs_seq):
    N = len(states)
    T = len(obs_seq)

    alpha = np.zeros((N, T))
    alpha[:, 0] = start_prob * emit_prob[:, obs_seq[0]]

    for t in range(1, T):
```

```
        for j in range(N):
            alpha[j, t] = emit_prob[j, obs_seq[t]] * np.sum(alpha[:, t-1] * tr

    return alpha, np.sum(alpha[:, -1])


# ----------------------------------------
# Run a few example simulations
# ----------------------------------------

for i in range(3):
    h, o = simulate_sequence()
    decoded = viterbi(o)
    alpha, likelihood = forward(o)

    print(f"\n--- Example {i+1} ---")
    print("Hidden states:    ", [states[x] for x in h])
    print("Observations:     ", [observations[x] for x in o])
    print("Viterbi Decoded:  ", [states[x] for x in decoded])
    print("Likelihood:       ", likelihood)
```

```
--- Example 1 ---
Hidden states:     ['s', 'p', 'p', 'ie', 'ie', 'tS', 'tS', 'tS', 'tS', 'tS',
's', 'p']
Observations:      ['highE', 'longDur', 'highE', 'longDur', 'longDur', 'longDu
r', 'highE', 'highPitch', 'highPitch', 'highPitch', 'highE', 'highPitch']
Viterbi Decoded:   ['s', 'p', 'p', 'ie', 'tS', 'tS', 'tS', 'tS', 'tS', 'tS', 't
S', 'tS']
Likelihood:        6.655344016270932e-08

--- Example 2 ---
Hidden states:     ['s', 'p', 'p', 'ie', 'tS', 's', 'p', 'ie', 'tS', 'tS', 't
S', 'tS']
Observations:      ['highE', 'highE', 'highE', 'longDur', 'highE', 'highE', 'lo
wE', 'highPitch', 'highPitch', 'highE', 'highE', 'longDur']
Viterbi Decoded:   ['s', 'p', 'p', 'ie', 'tS', 'tS', 'tS', 'tS', 'tS', 'tS', 't
S', 'tS']
Likelihood:        1.4642070353951677e-06

--- Example 3 ---
Hidden states:     ['s', 'p', 'ie', 'tS', 'tS', 'tS', 'tS', 's', 'p', 'ie', 't
S', 's']
Observations:      ['highPitch', 'highE', 'highPitch', 'highPitch', 'highE', 'h
ighE', 'longDur', 'highE', 'longDur', 'highE', 'lowE', 'highE']
Viterbi Decoded:   ['s', 'p', 'ie', 'tS', 'tS', 'tS', 'tS', 'tS', 'tS', 'tS',
'tS', 'tS']
Likelihood:        9.695883708597841e-07
```

In [ ]: