A meteorological observing station in Ithaca uses a wireless network link to transmit data from a moisture sensor to a central computer. Each day the sensor sends a `yes` message to the computer if it has rained that day, or a `no` message if it has not rained. Unfortunately, the wireless link is noisy, so that 40% of the `yes` messages sent by the sensor are incorrectly received as `no` messages by the computer, and 20% of the `no` messages transmitted by the sensor are incorrectly received as `yes` messages by the computer. Suppose that the weather in Ithaca can be modeled by a simple Markov chain: if it rains one day, the probability of it raining the next day is 65%; if it does not rain one day, the probability of rain the next day is 25%.

During the first week of operation, the computer receives the following sequence of messages from the sensor:

```
yes yes no yes no no yes
```

Use the Viterbi algorithm to estimate whether or not it was actually raining on each day of the week. Assume that on the first day of the week, there was a 50% chance of rain in Ithaca.

This problem can be modeled as an HMM with the following parameters:

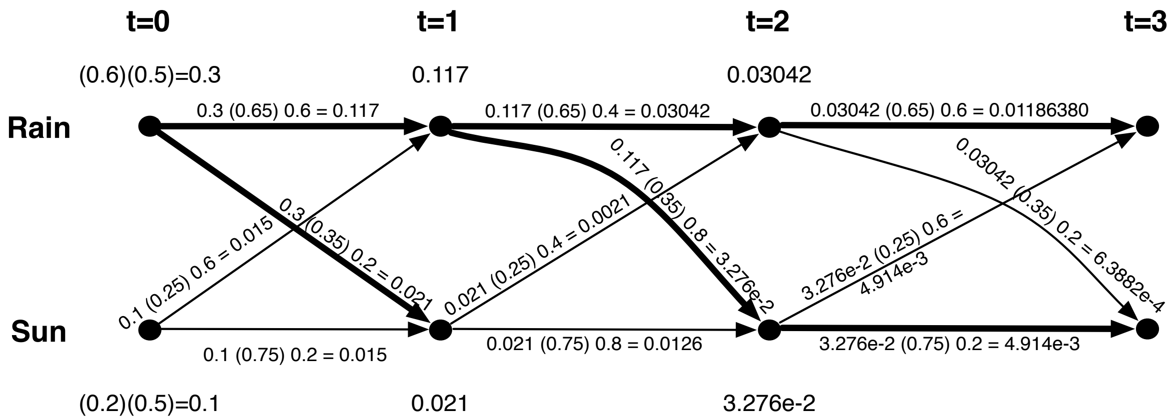
- **Initial distribution:** $P(X_0 = R) = 0.5$, $P(X_0 = S) = 0.5$ (where R and S represent rain and sun, respectively).
- **Transition probabilities:**

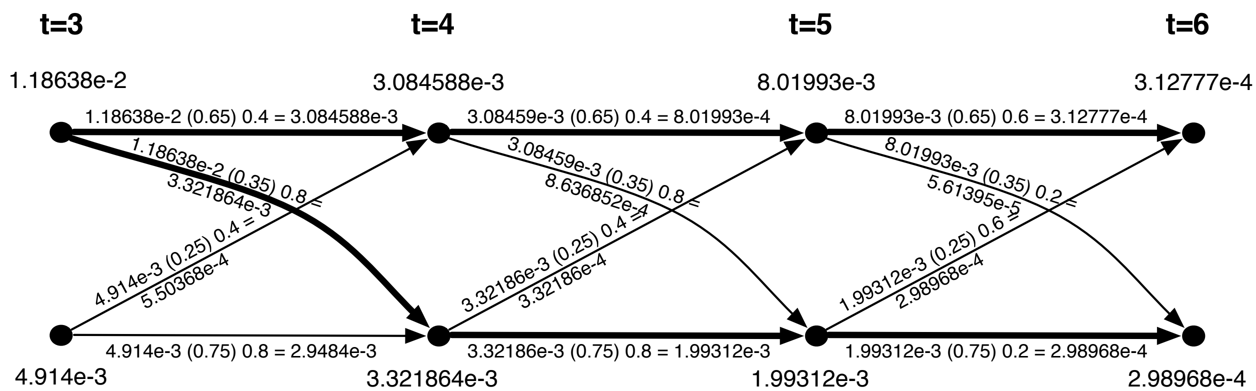
$$P(X_{t+1} = R|X_t = R) = 0.65$$
$$P(X_{t+1} = S|X_t = R) = 0.35$$
$$P(X_{t+1} = R|X_t = S) = 0.25$$
$$P(X_{t+1} = S|X_t = S) = 0.75$$

- **Emission probabilities:**

$$E_R(Y) = 0.6, E_R(N) = 0.4$$
$$E_S(Y) = 0.2, E_S(N) = 0.8$$

Then the Viterbi algorithm can be used to find the most likely state sequence:





From the result at $t = 6$, we see that the most likely state sequence ends with rain. Following the bolded arrows backwards, we find the mostly likely sequence: RRRRRRR.