RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

# Temporal Models

Edgar Granados

# Gait Modeling

## Example

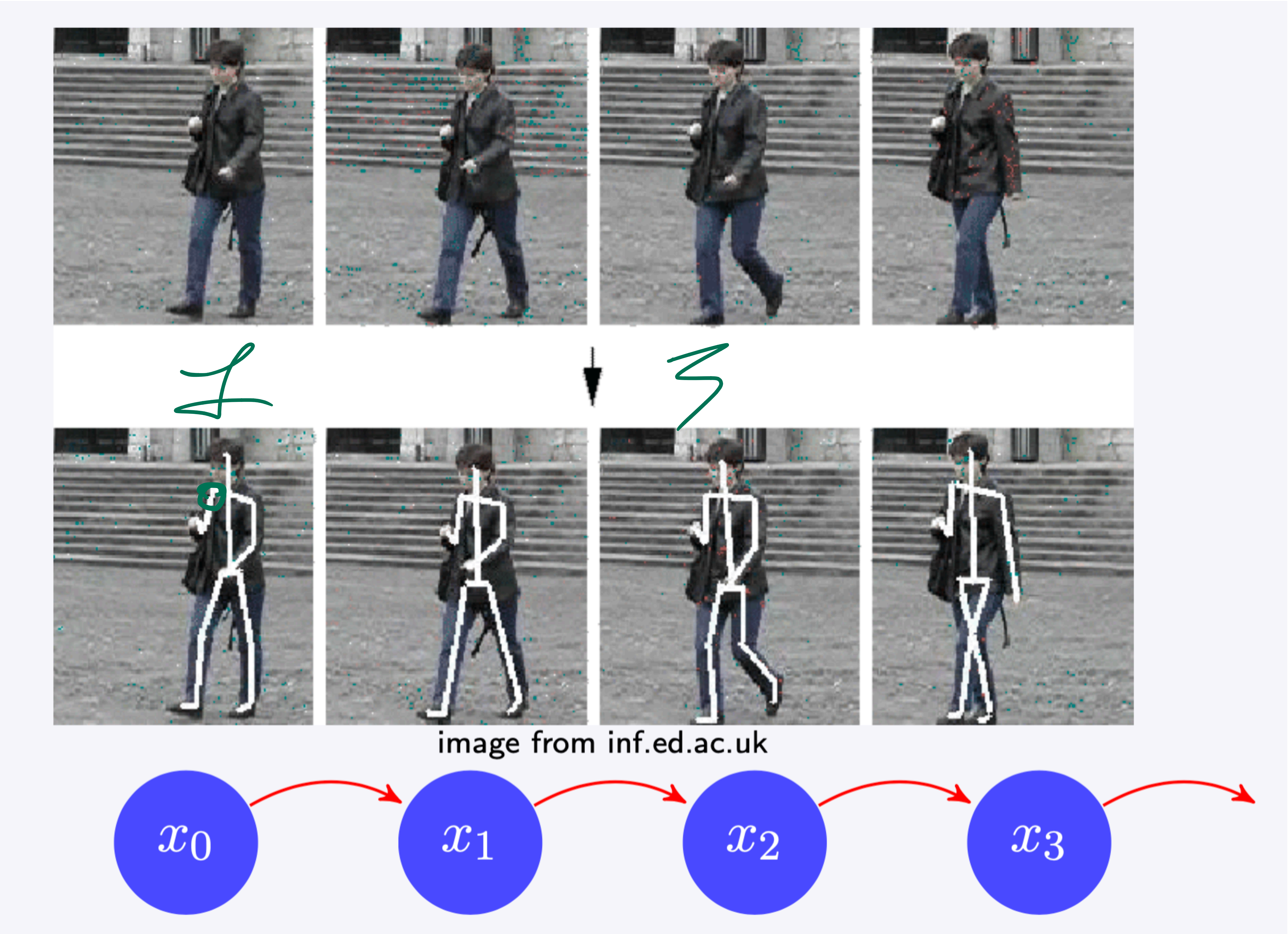
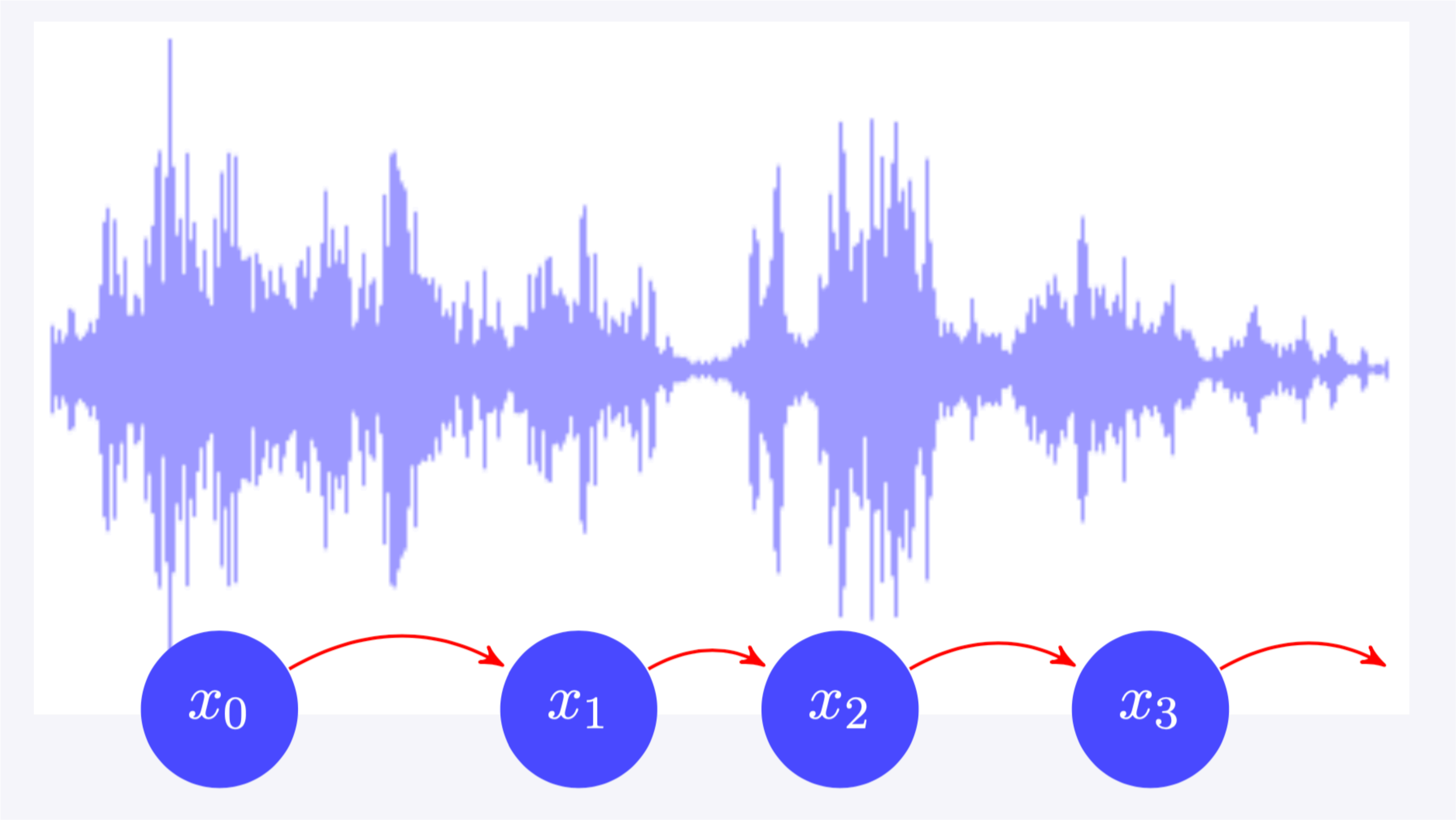



image from inf.ed.ac.uk

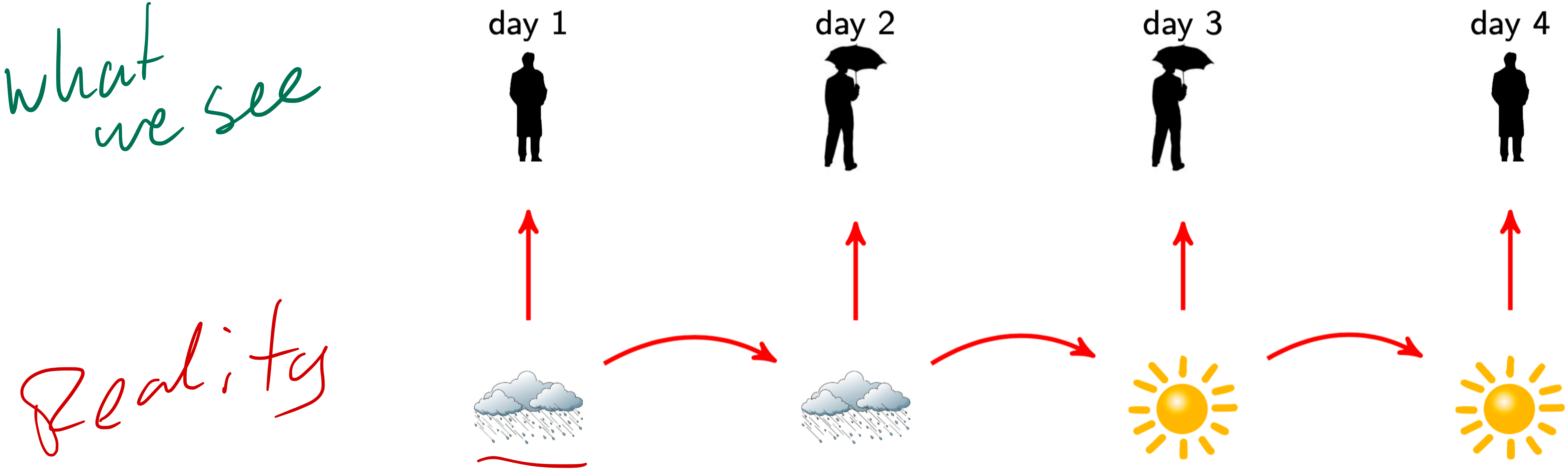
# Natural Language Processing

## Example



# Example

## Security Guard



# States and observations

- Sequence of Snapshots / Time slices
- **State:** Set of RVs $X_t$
- **Evidence:** $E_t$
- Assume discrete time

$t \in \{0, 1, 2, \ldots\}$

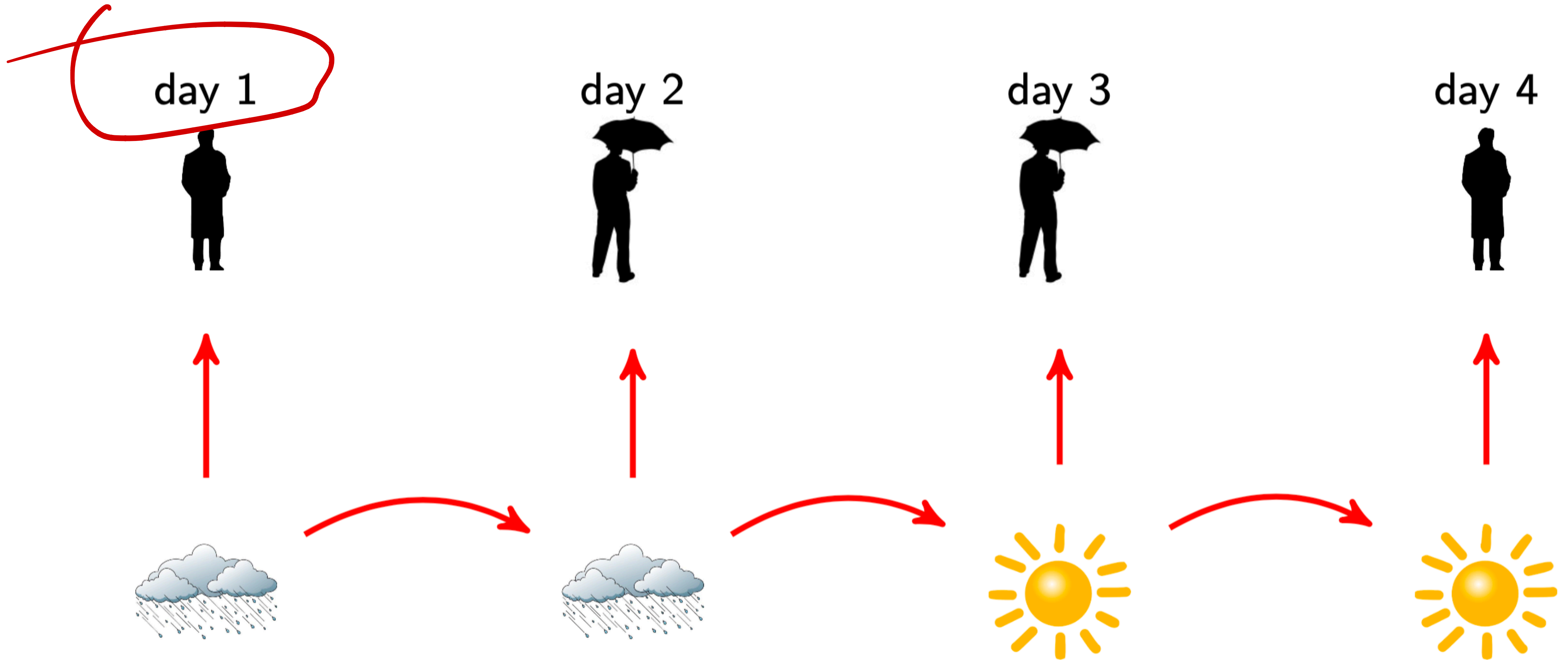
- $X_{a:b}$ denotes the variable X at times t=a to t=b

$$X_{a:b} = [X_a, X_{a+1}, (\ldots), X_b]$$

$$E_{a:b} = [E_a, E_{a+1}, (\ldots), E_b]$$

# States and observations

- **State:** $X_t = \{Rain, \neg Rain\}$
- **Evidence:** $E_t = \{Umbrella, \neg Umbrella\}$



# Transition model

- The transition model specifies the probability distribution over current state values, given all the previous states
- $\boxed{X_{0:t-1}}$ is unbounded in size as $t$ increases
- **Markov Assumption**

$$P(X_t \mid X_{0:t-1}) = P(X_t \mid X_{t-1})$$

# Markov Chains

- First order Markov chain:

$$P(X_t \mid X_{0:t-1}) = \boxed{P(X_t \mid X_{t-1})}$$

- Second order Markov chain

$$P(X_t \mid X_{0:t-1}) = P(X_t \mid \underline{X_{t-1}}, \underline{X_{t-2}})$$

- Any n-order Markov chain can be reduced to a first-order Markov chain by redefining the states

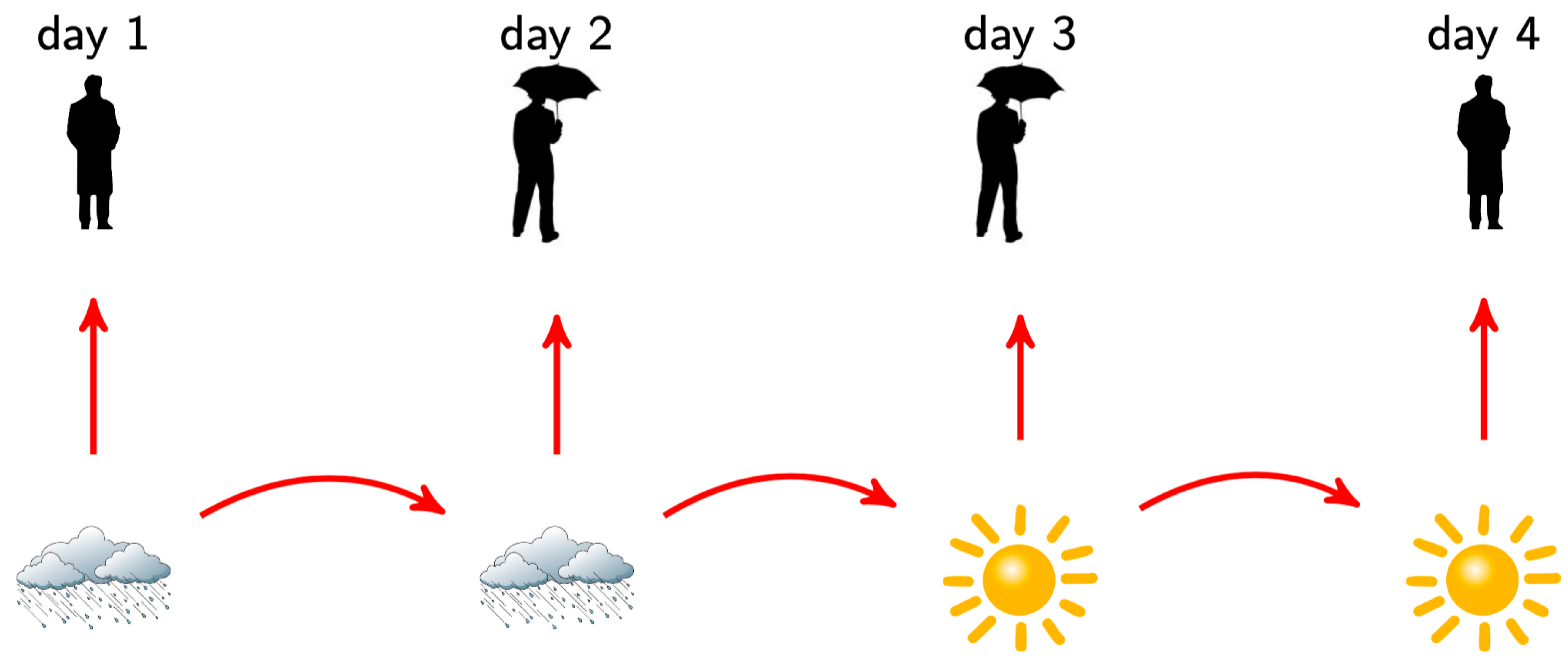
# Transition model

$t$ $\qquad$ $t-1$

$P(\text{Rain today} \mid \text{Rained Yesterday})$

$P(\text{Rain today} \mid \neg\text{Rained Yesterday})$

## Example



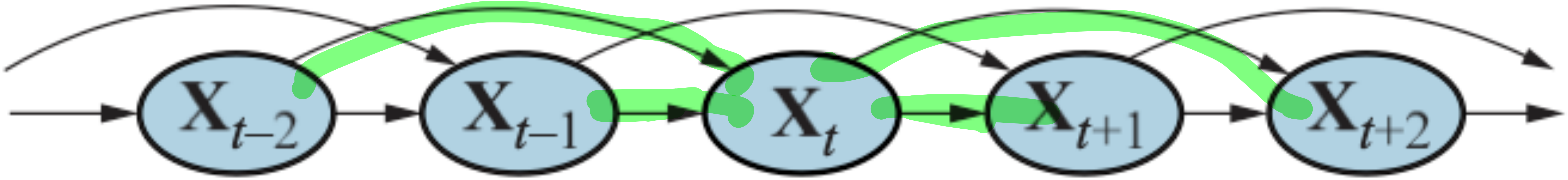
# Transition model

- The transition model is a dynamic Bayesian network. Previous states are the cause of the current state.

FO

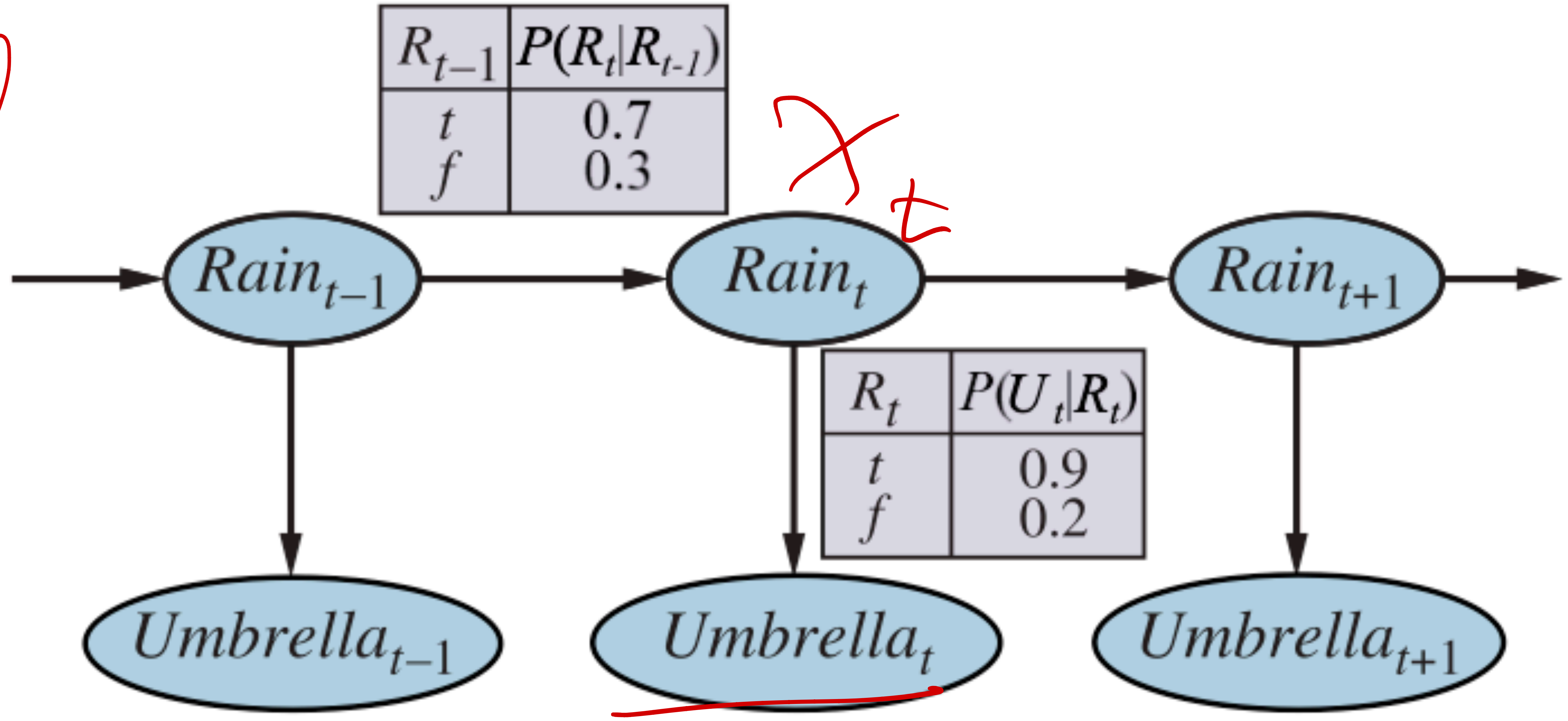
t-1 t


SO



# Observation model

- The observation model specifies the probability distribution over current observations, given the current state $P(E_t | X_t)$
- Sensor Model
- Dynamic Bayesian Network



# Temporal Models

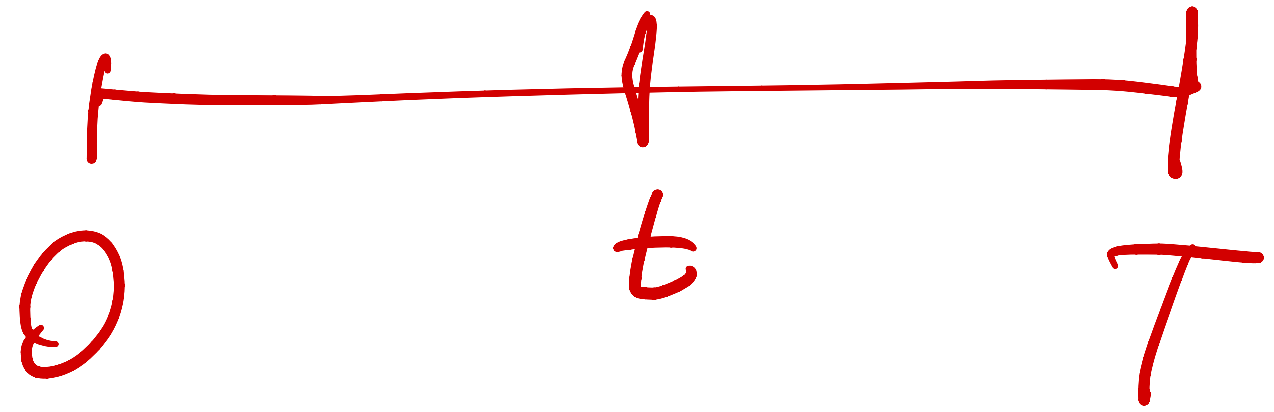
## Inference

- Joint probability of sequence of states

$$P(X_{0:t}, E_{1:t}) = P(X_0) \prod_{i=1}^{t} \underbrace{P(X_i \mid X_{i-1})}_{\text{transition}} \overbrace{P(E_i \mid X_i)}^{\text{observation}}.$$

- Filtering – state estimation $P(X_t \mid e_{1:t})$
- Prediction $P(E_{t+1:T} \mid e_{1:t})$
- Smoothing $P(X_{0:t} \mid e_{1:t})$
- Most Likely Explanation

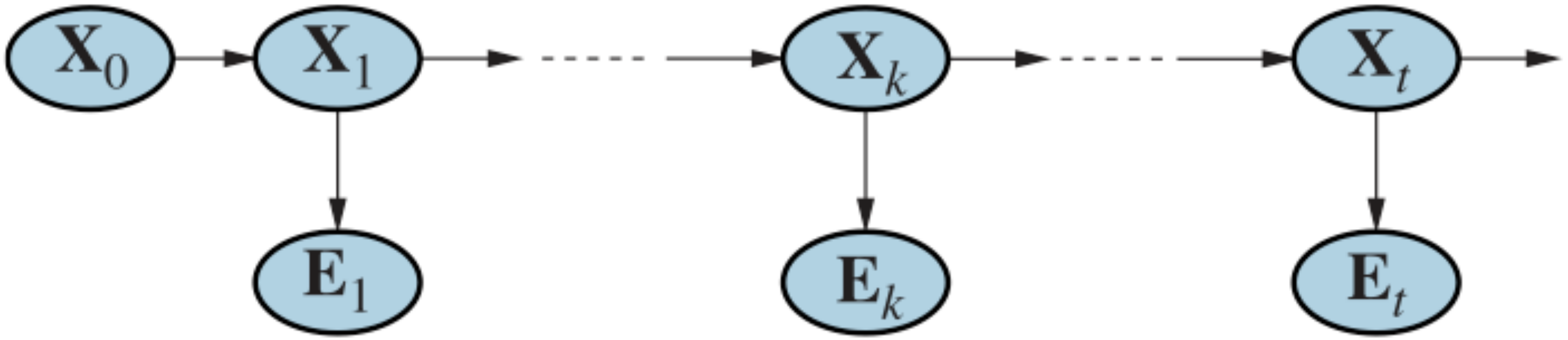
$$\underset{X_{0:t}}{\text{Argmax}}\, P(X_{0:t} \mid e_{1:t})$$



# Filtering

## State Estimation

- Given all the past and the current observations, compute a distribution on the current state $X_t$
- Belief state: $P(X_t \mid e_{1:t})$



# Filtering

## State Estimation

- $P(X_t \mid e_{1:t}) =$

# Filtering

## State Estimation

- $P(X_t | e_{1:t}) = \dfrac{P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{\sum_{x_t} P(e_t \mid x_t) \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}$

- Can be computed recursively by starting with the prior $P(X_0)$

# Prediction

- Given: $e_{1:t} = (e_1, \ldots, e_t)$
- Compute: $P(E_{t+1:T} \mid e_{1:t})$

# Prediction

- $P(E_{t+1:T} \mid x_t) =$

# Prediction

- $P(E_{t+1:T} \mid x_t) = \text{BACKWARD}(P(E_{t+2:T} \mid X_{t+1}, E_{t+1}))$

# Smoothing

- Given: $e_{1:t} = (e_1, \ldots, e_t)$
- Compute: $P(X_k \mid e_{1:t})$

# Smoothing

- $P(X_k \mid e_{1:t}) =$

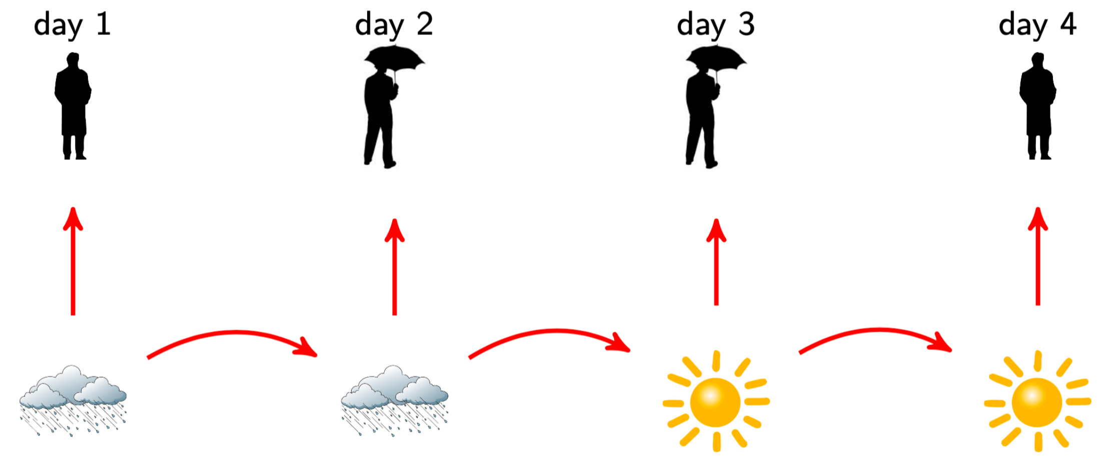
# Smoothing

## Algorithm

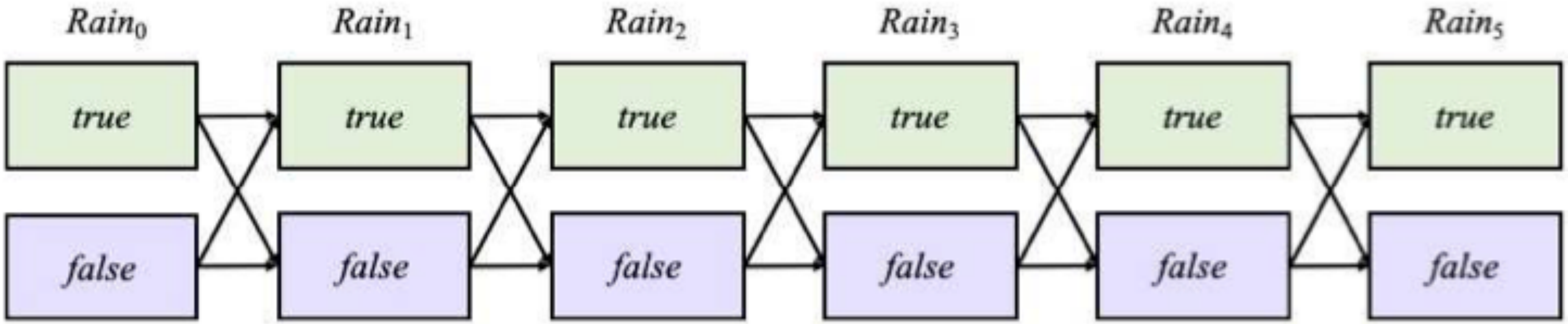
```
function FORWARD-BACKWARD(ev, prior) returns a vector of probability distributions
  inputs: ev, a vector of evidence values for steps 1, . . . , t
          prior, the prior distribution on the initial state, P(X_0)
  local variables: fv, a vector of forward messages for steps 0, . . . , t
                   b, a representation of the backward message, initially all 1s
                   sv, a vector of smoothed estimates for steps 1, . . . , t

  fv[0] ← prior
  for i = 1 to t do
     fv[i] ← FORWARD(fv[i − 1], ev[i])
  for i = t down to 1 do
     sv[i] ← NORMALIZE(fv[i] × b)
     b ← BACKWARD(b, ev[i])
  return sv
```

# Most Likely Explanation




- Umbrella sequence [true,true,false,true,true]
- What is the weather sequence that most likely explains this?



# Most Likely Explanation

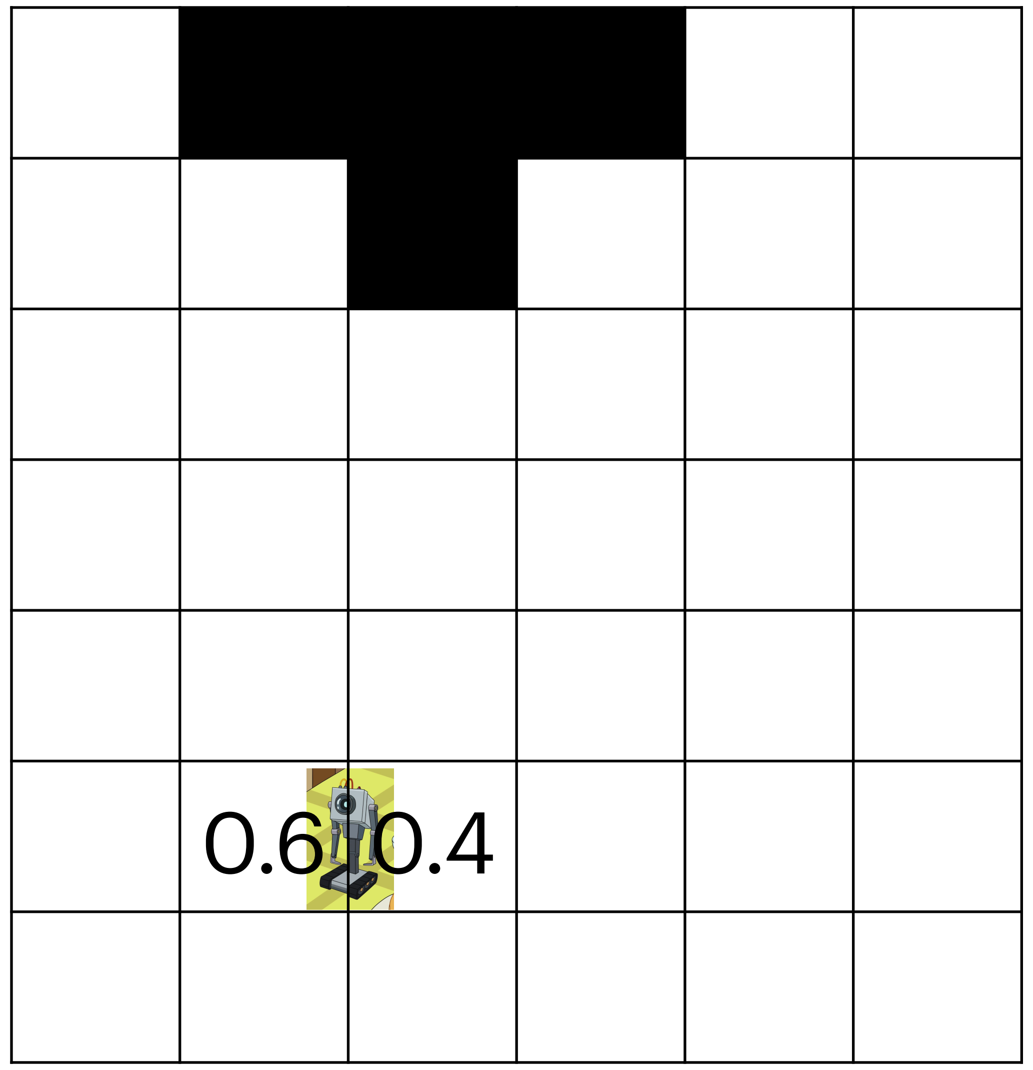
## Viterbi Algorithm

- Filering: $P(X_t \mid e_{1:t}) = \alpha P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})$

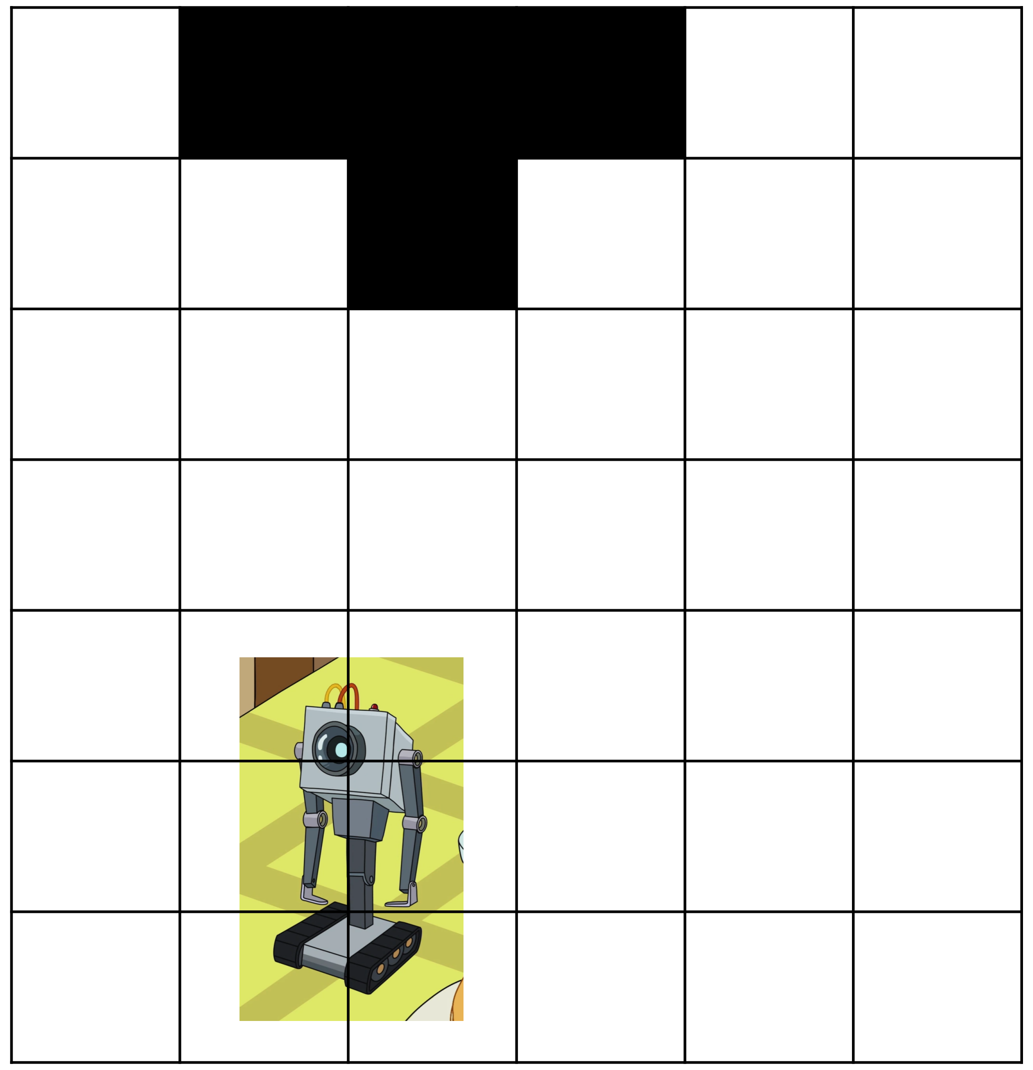
# Robot in a Discrete World

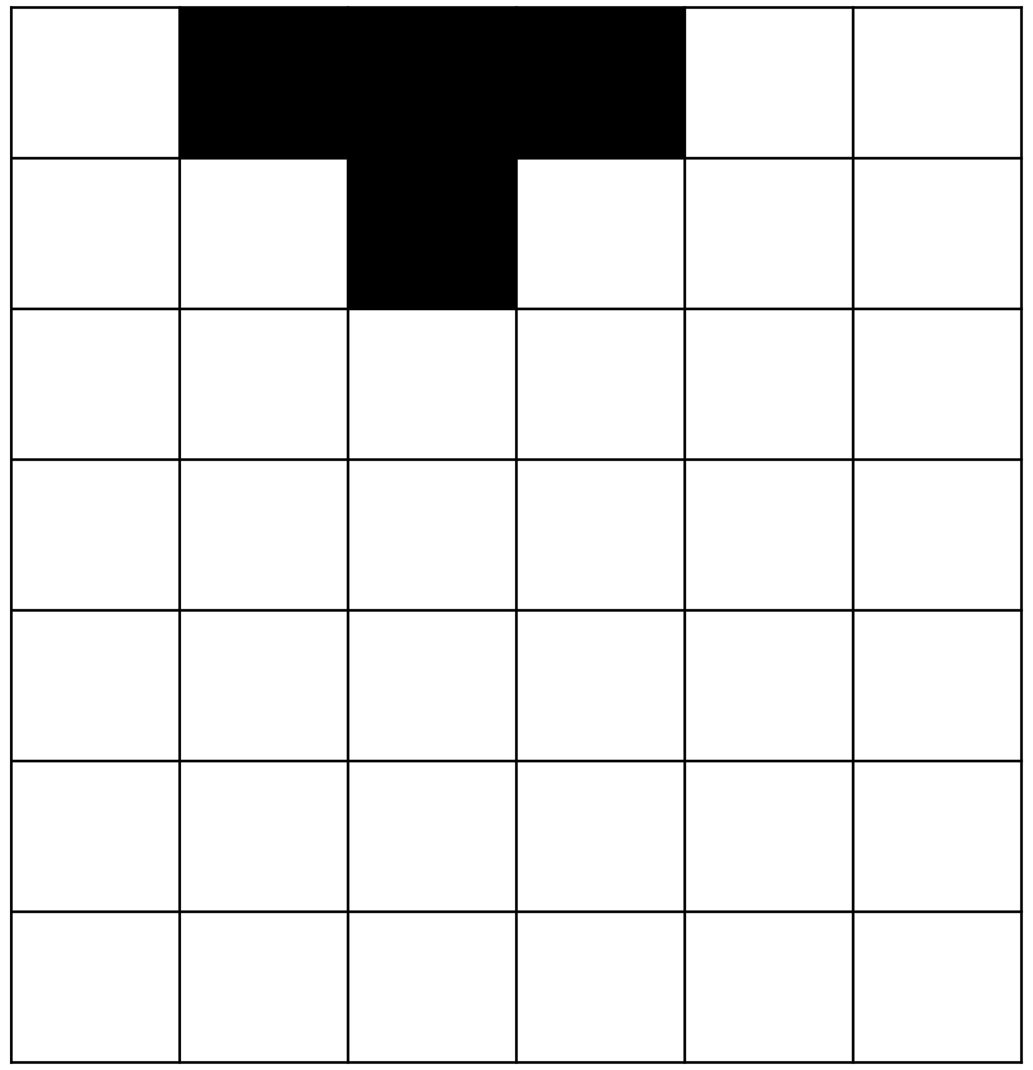
- Transition model: When the robot moves in a certain direction, there is a 10% chance of going backwards
- Observation model: The robot is able to sense how far away obstacles are in the Up direction. The robot is 80% of the time correct in its measurement, 10% percent chance to overestimate by 1 cell and 10% percent chance to underestimate by 1

- Move 1 step up
- Observation: 2 cells away



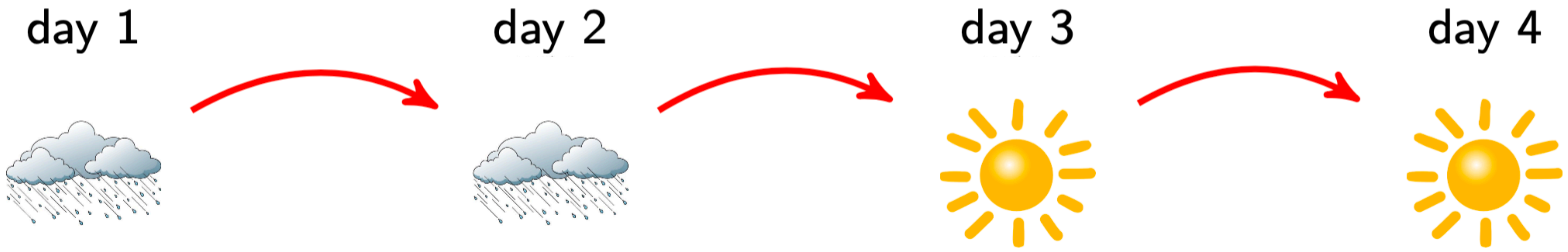




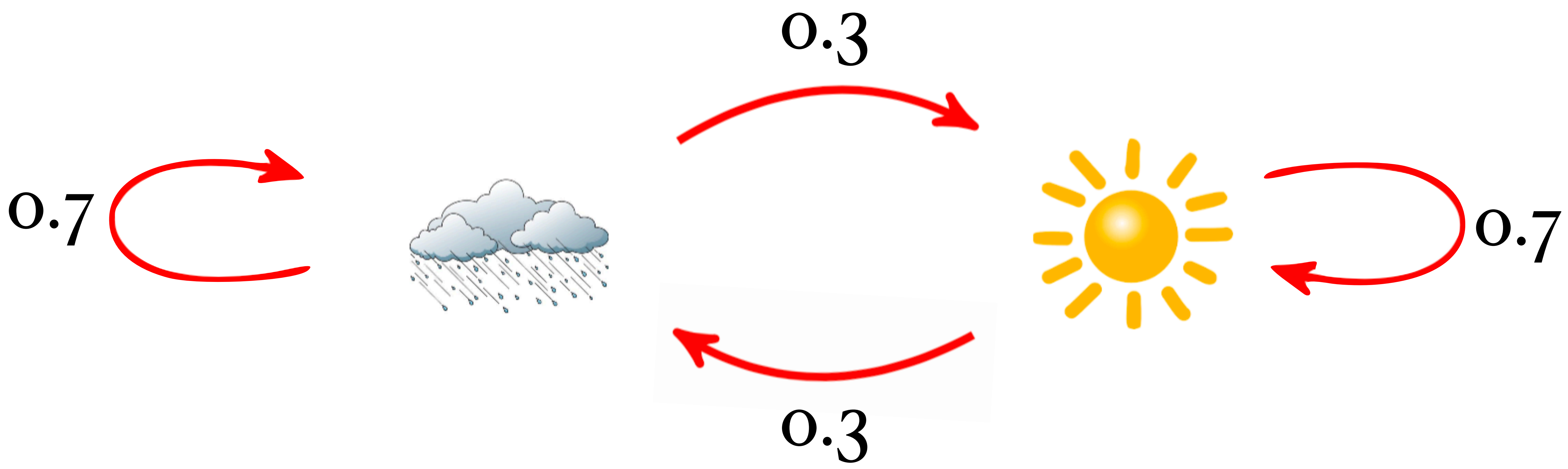


# Markov Chain

- Temporal model where the state is a single random variable that is always known.

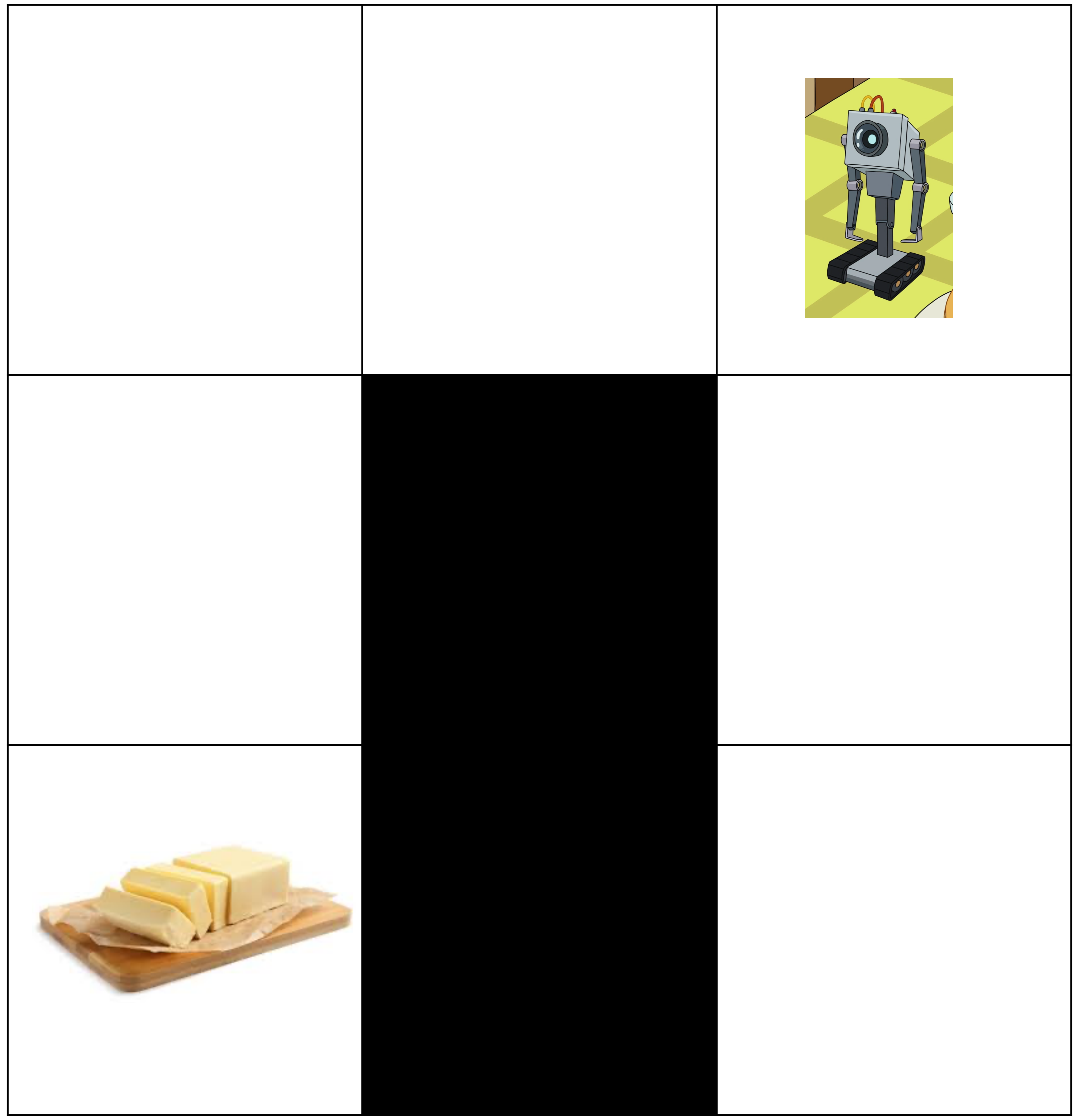



- Transition Function



# Markov Chain

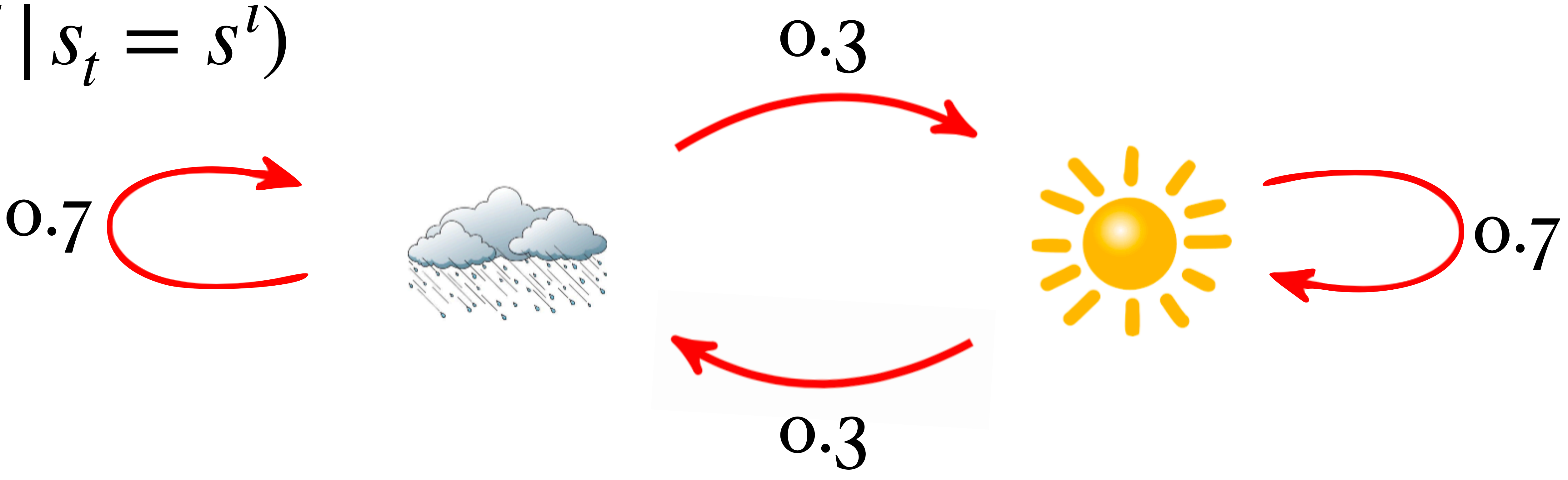
## Example



# Markov Chain

## Transition Matrix

- States: $\{s^i\}$
- Transition function: $T[i, j] = P(s_{t+1} = s^j \mid s_t = s^i)$

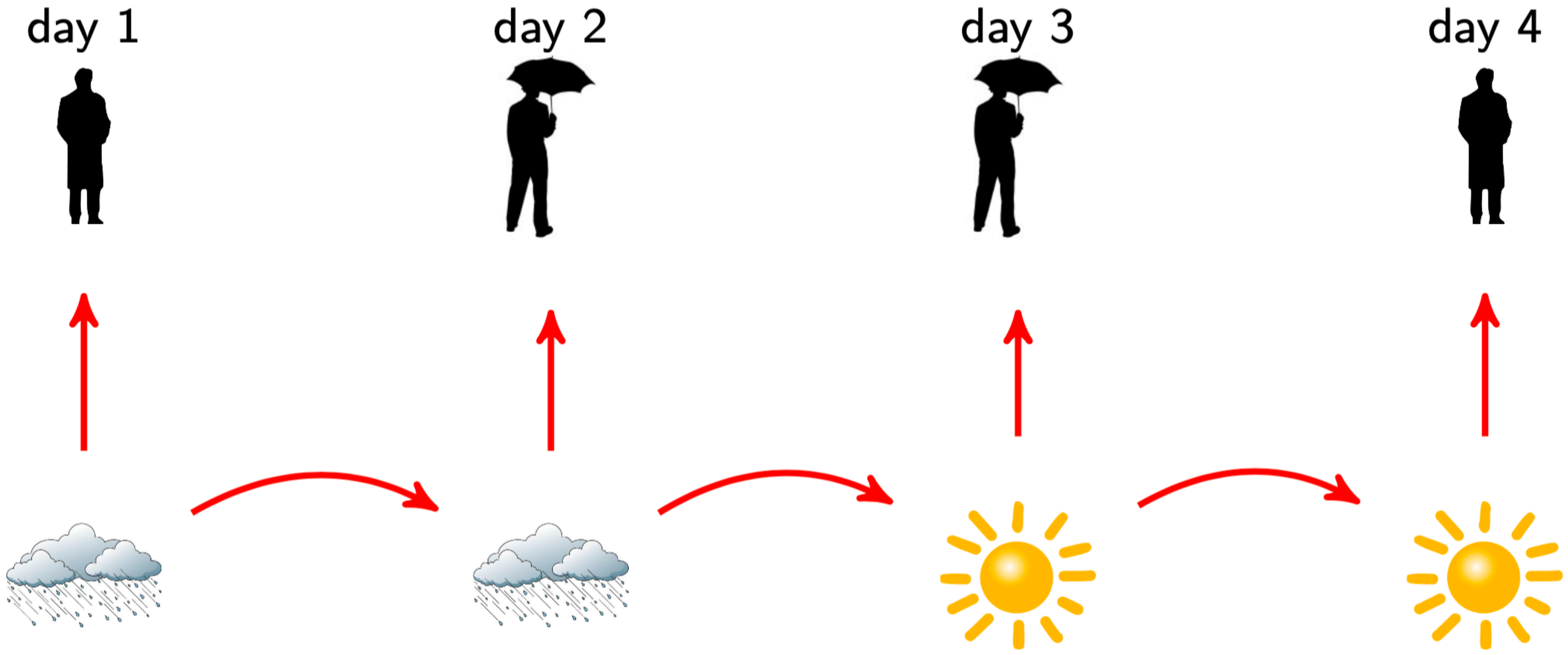


# Markov Chain

- Compute state distributions in the future
- $f_t[i] = P(s_t = s^i)$

# Hidden Markov Model

- Temporal model where the state is a single random variable that is unknown. An observable variable is used as evidence to infer the state.



# Hidden Markov Model

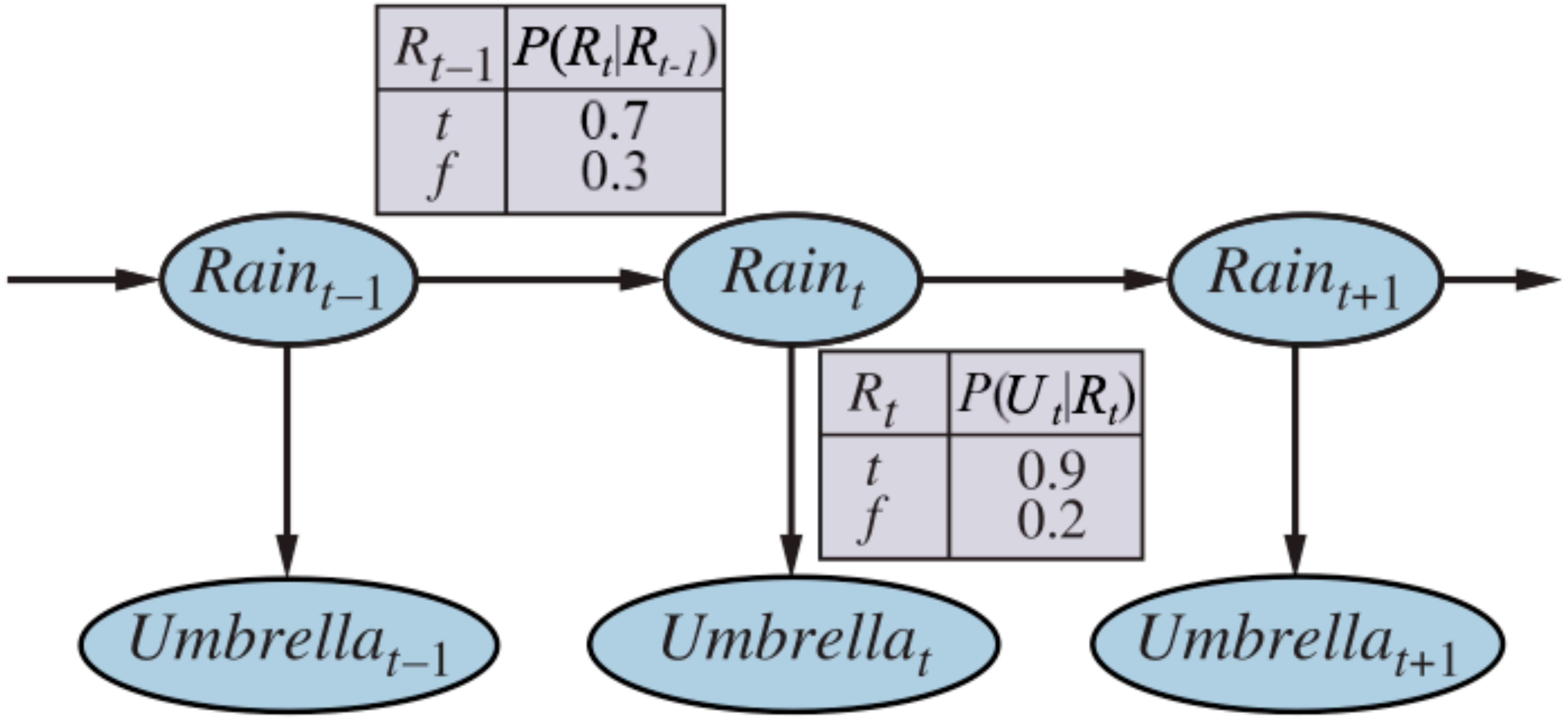
## Observation matrix

- $O_e[i, i] = P(E_t = e \mid s_t = s^i)$

$$O_e = \begin{bmatrix} P(e \mid s^0) & 0 & 0 & \cdots & 0 \\ 0 & P(e \mid s^1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & P(e \mid s^n) \end{bmatrix}$$

# Observation matrix

## Example



# Hidden Markov Model

- Forward:
  - $f_t[i] = P(s_t = s^i)$

# Hidden Markov Model

- Backward:
  - $b_t[i] = P(e_{t+1:k} \mid s_t = s^i)$