# Hidden Markov Model

Lili Mou
lmou@ualberta.ca
lili-mou.github.io

UNIVERSITY OF ALBERTA

# Drawbacks of LR/Softmax

- Classification is non-linear
  - May not even represented as fixed-dimensional features
- Do not consider the relationship of labels within one data sample

| The | lecture | is | really | boring |
|---|---|---|---|---|
| `determiner` | `?` | `verb` | `adverb` | `adjective` |

| Three | professors | lecture | IntroNLP |
|---|---|---|---|
| `CardinalNumber` | `Noun` | `?` | `ProperNoune` |

https://www.merriam-webster.com/dictionary/lecture

**lecture** noun

lec·ture | \ ˈlek-chər , -shər\

**Definition of *lecture* (Entry 1 of 2)**

1 : a discourse given before an audience or c

2 : a formal reproof

**lecture** verb

**lectured**; **lecturing** \ ˈlek-chə-riŋ , ˈlek-shriŋ\

**Definition of *lecture* (Entry 2 of 2)**

*intransitive verb*

# Motivation

- One data sample may have different labels, e.g.,
  - POS tagging
  - Parsing
  - Sentence generation
  - etc.

# Markov Model

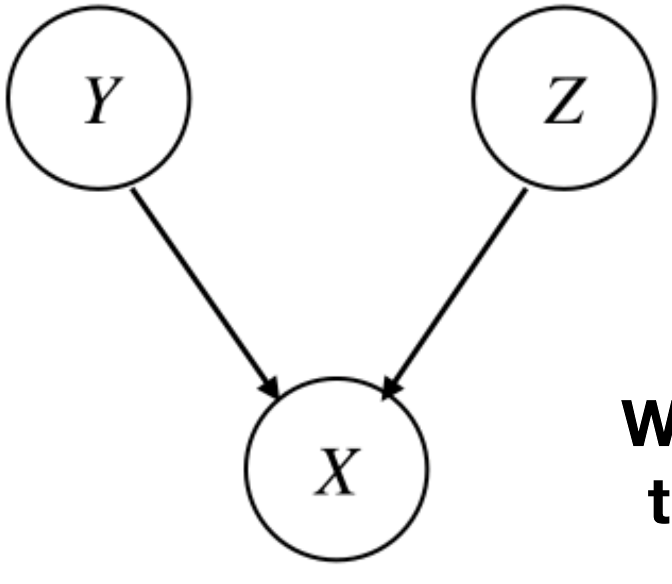
- Finite states $S = \{s_1, s_2, \cdots, s_n\}$
- You start from a state following the distribution $\boldsymbol{\pi} = [\pi_1, \pi_2, \cdots, \pi_n]$
- Transition only depends on the current state $\mathbb{P}[S^{(t)} = s_i \,|\, S^{(t-1)} = s_j]$
- Examples
  - Weather
  - N-gram model

# Bayesian Network in General

- Directed Acyclic Graph $G = \langle V, E \rangle$
  - Each node is a random variable
  - Each edge $a \rightarrow b$ represents that $a$ is a direct "cause" of $b$
  - The joint probability can be represented as

$$p(x_1, \cdots, x_n) = \prod_{i=1}^{n} p(x_i \,|\, \mathrm{Par}(x_i))$$

**All parents**

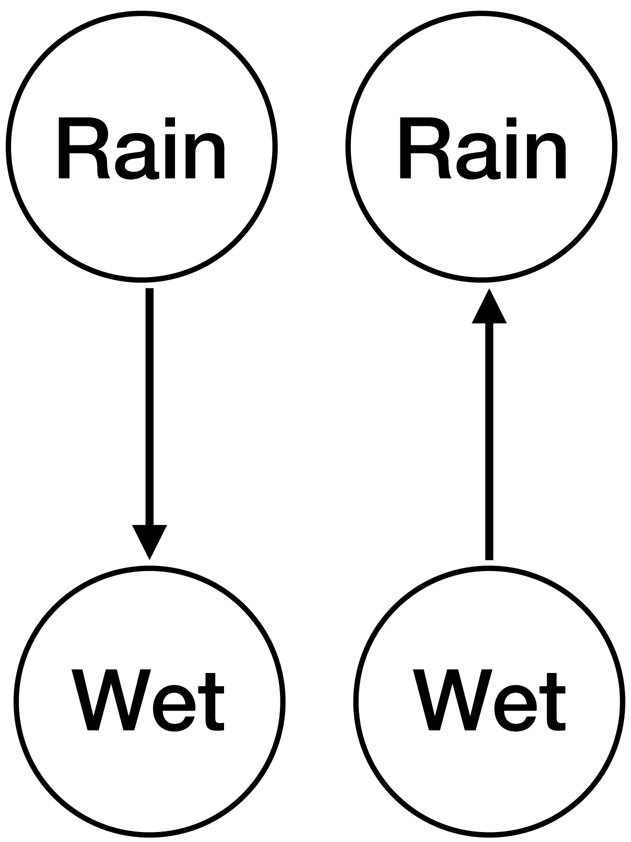



$$p(X, Y, Z) = p(X \,|\, Y) p(X \,|\, Z)$$

**Wrong. Factorization only happens to the LHS of the conditional bar.**

UNIVERSITY OF ALBERTA

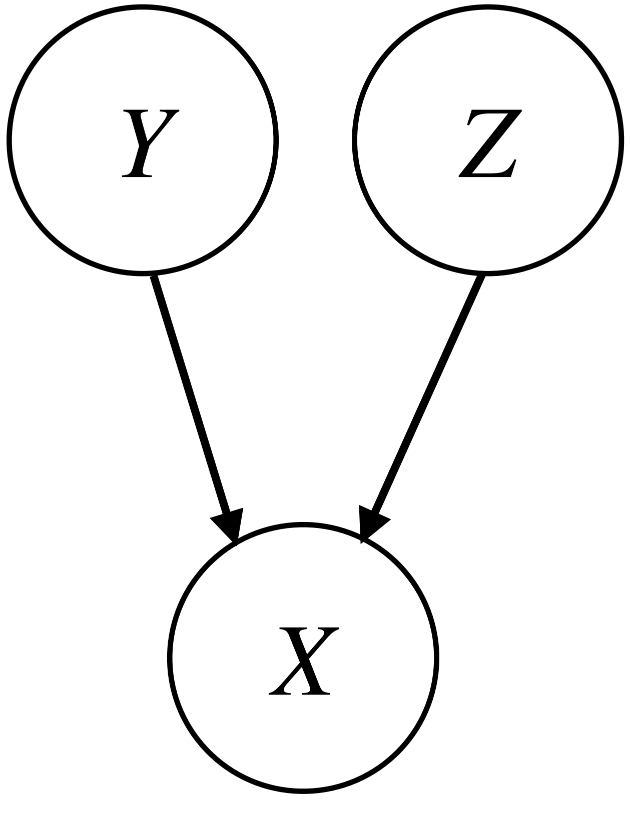
# Can we reverse cause & effect?

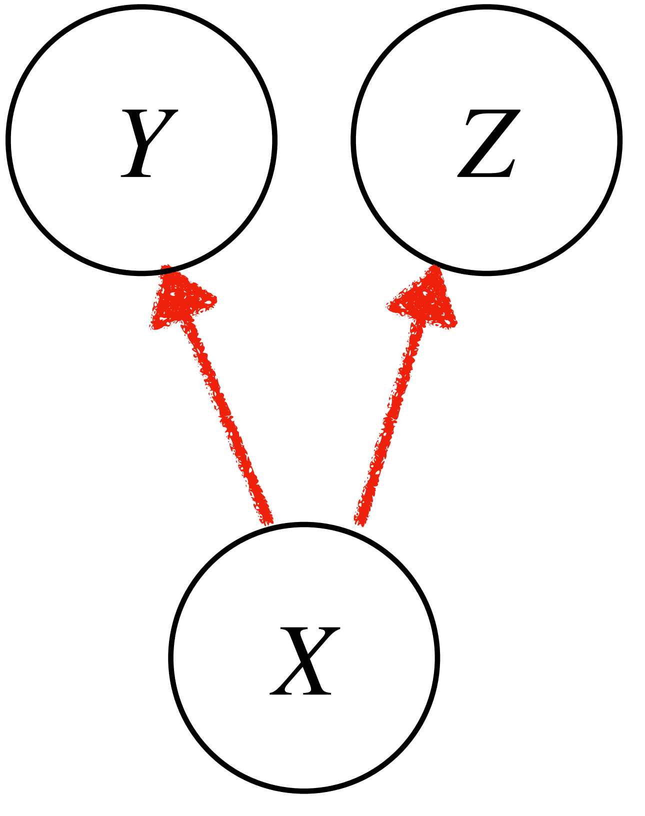



$$p(R, W) = p(R)p(W \mid R)$$

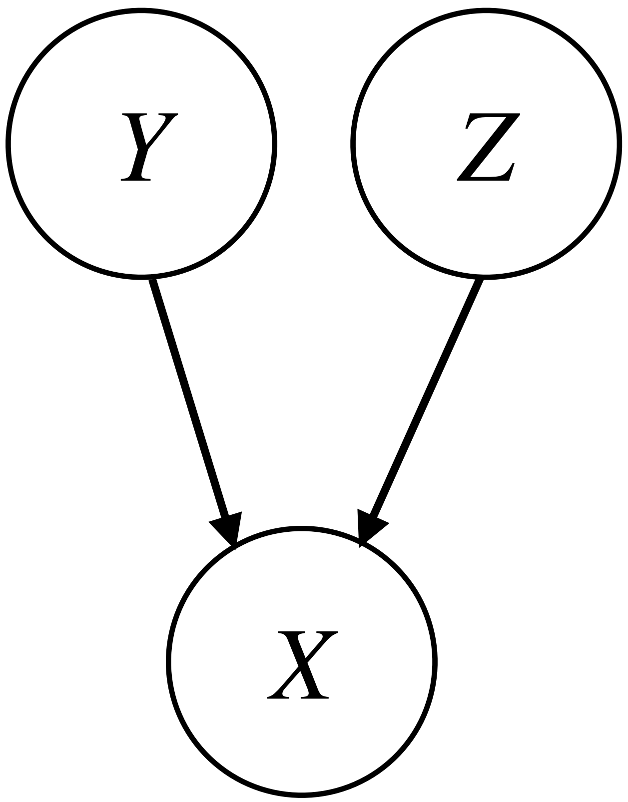
$$p(R, W) = p(W)p(W \mid R)$$

# Can we reverse cause & effect?



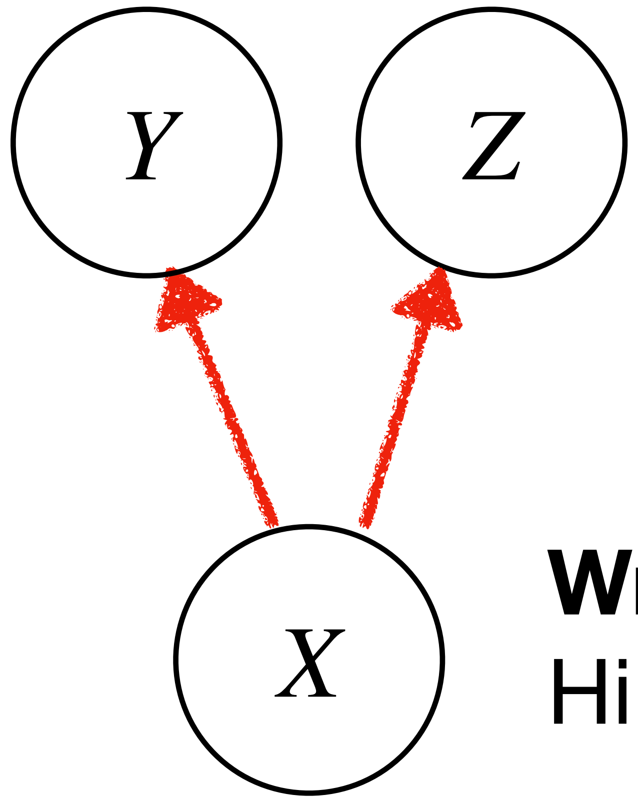



# Can we reverse cause & effect?




$$Y \perp Z \,|\, X$$

**does not hold in general**

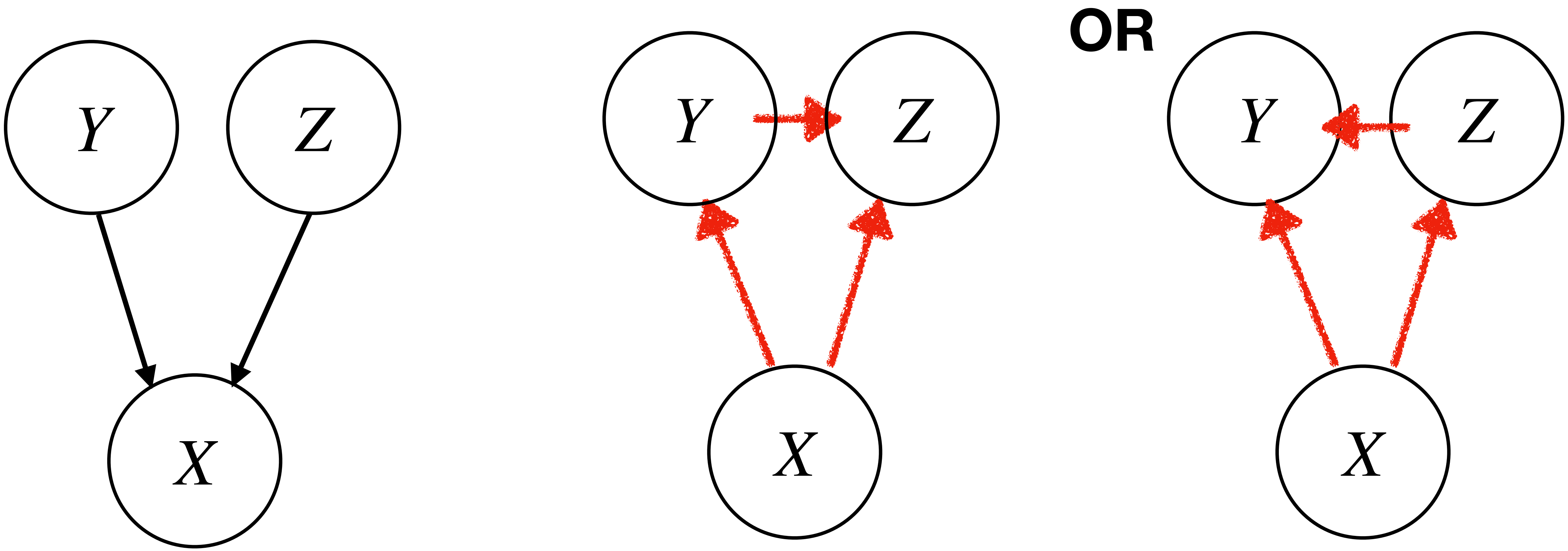



$$Y \perp Z \,|\, X$$

**Written assignment: Prove.**
Hint: By definition.

By the property of BNs,
$Y \perp Z \,|\, X. \implies 0$ mark

# Can we reverse cause & effect?




**Cause and effect cannot be formally defined.**

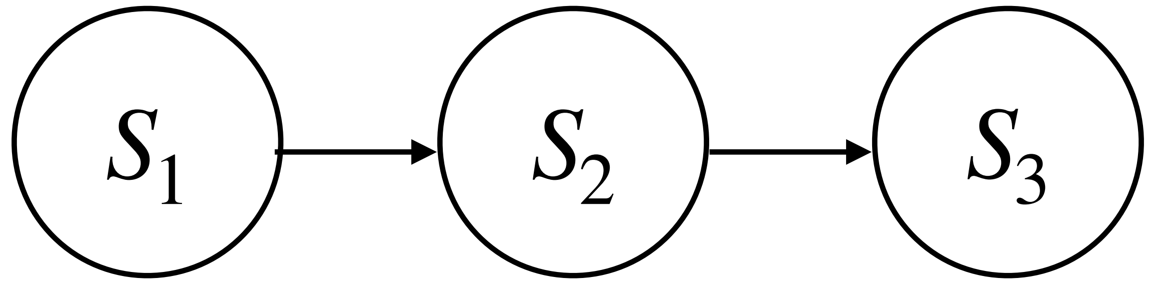
- In BN, "→" refers to conditional probability
- In logics, "→" refers to entailment

Cause and effect cannot be formally defined.

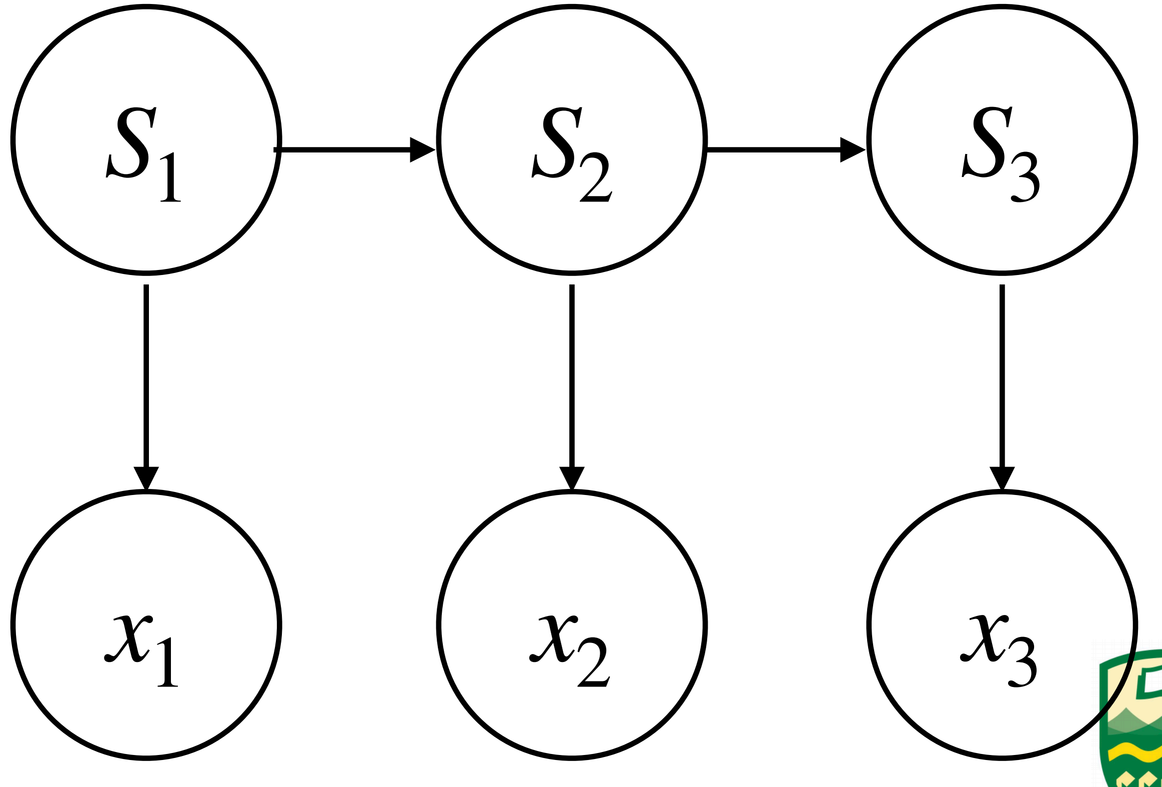
**But with our intuition of cause and effect, we can simplify our model.**

UNIVERSITY OF ALBERTA

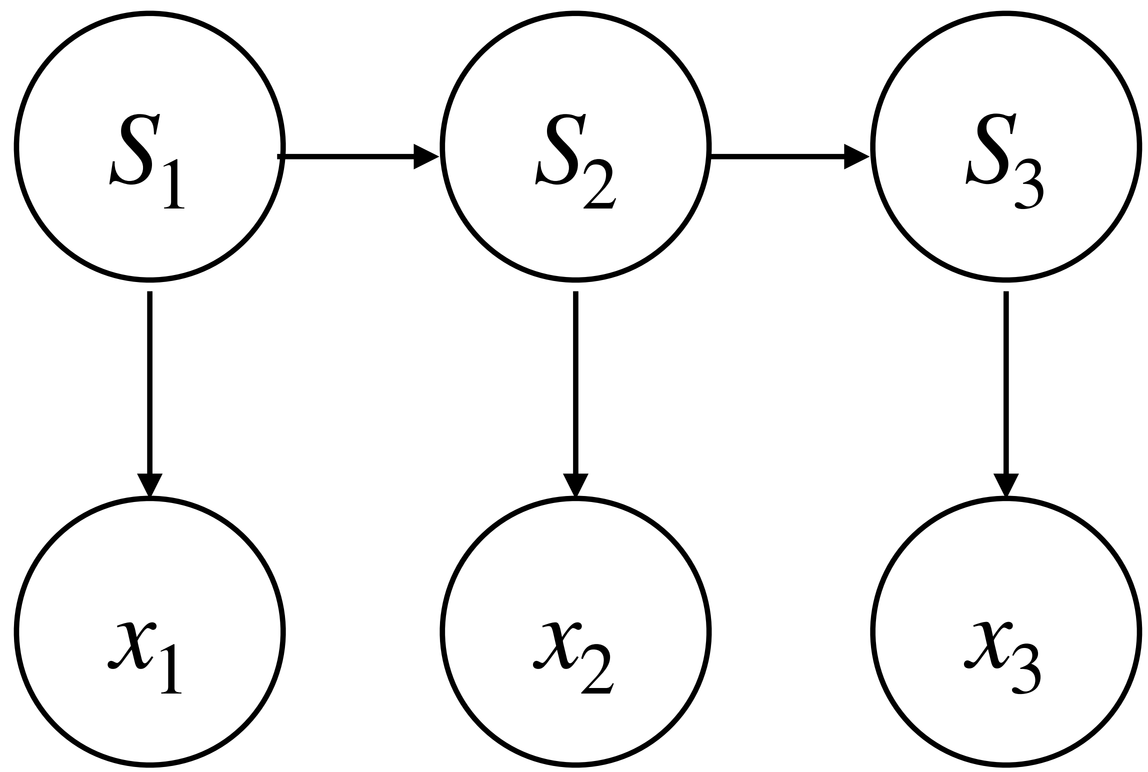
# Markov Model




# Hidden Markov Model



# Hidden Markov Model



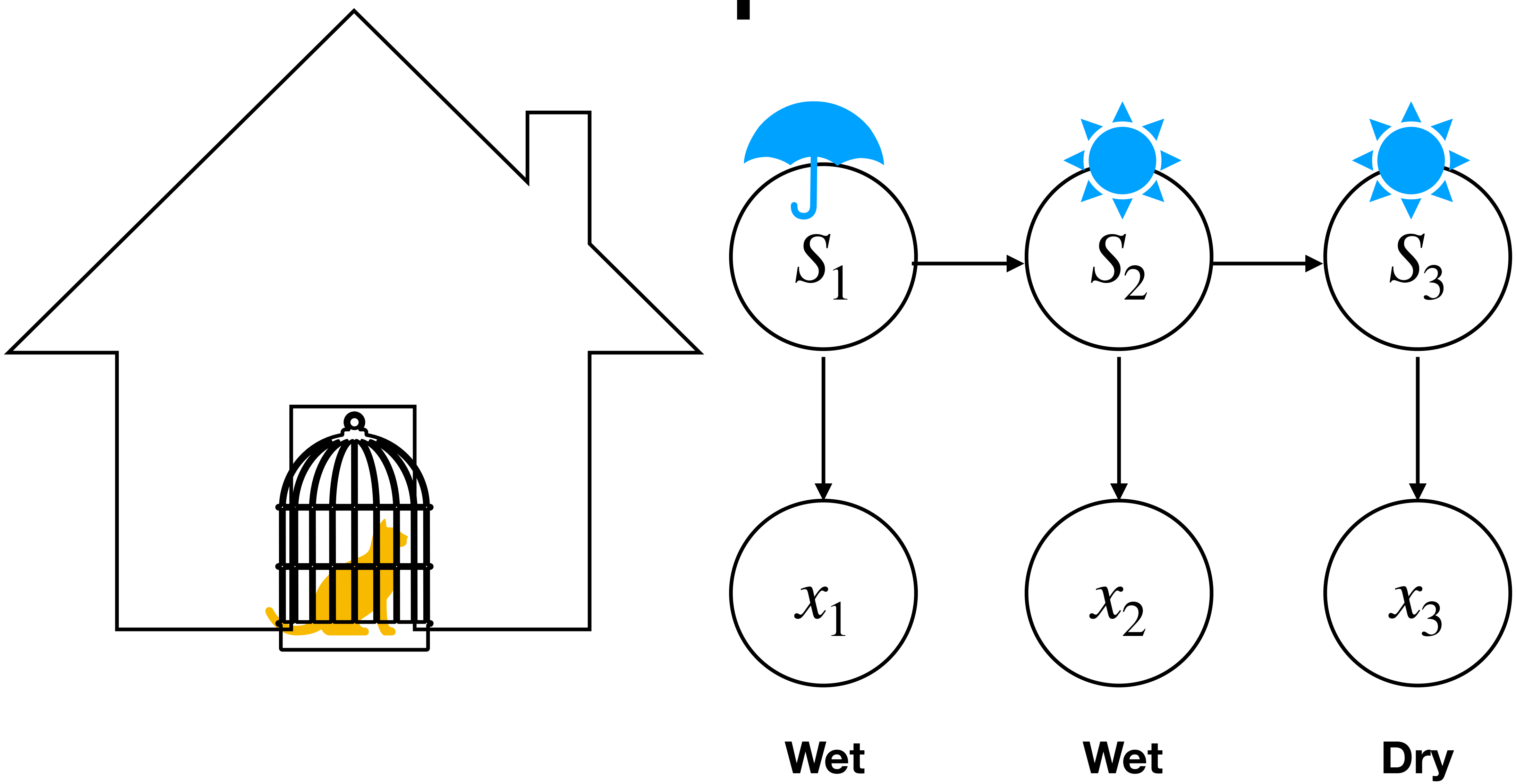

$$p(s_1, \cdots, s_T, x_1, \cdots x_T) = p(s_1) \prod_{t=2}^{T} p(s_t \mid s_{t-1}) \prod_{t=1}^{T} p(x_t \mid s_t)$$

**Initial State Prob.** $n$

**Transition Prob.** $n^2$

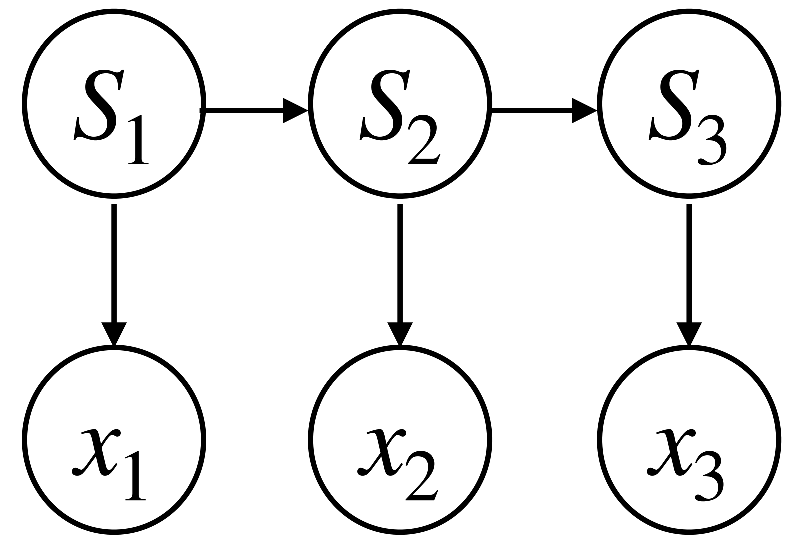
**Emission Prob.** $v \cdot n$

# Example of HMM



# Maximum Likelihood Estimation

- Training if fully observable
  - E.g., annotated by experts




$$p(s_1, \cdots, s_T, x_1, \cdots x_T) = p(s_1) \prod_{t=2}^{T} p(s_t \mid s_{t-1} \prod_{t=1}^{T} p(x_t \mid s_t)$$

$$\log p(\,\cdot\,) = \boxed{\log p(s_1)} + \boxed{\sum_{t=2}^{T} \log p(s_t \mid s_{t-1})} + \boxed{\sum_{t=1}^{T} \log p(x_t \mid s_t)}$$

**Parameters factorize**

# MLE for Multinomial Distribution

- Counting
  - With one constraint $\pi_1 + \cdots \pi_n = 1$
  - You need to explicitly represent $\pi_n = 1 - \pi_1 - \cdots - \pi_{n-1}$
  - Or, you apply the Lagrangian multiplier method

$$\log p(\cdot) = \boxed{\log p(s_1)} + \sum_{t=2}^{T} \log p(s_t | s_{t-1}) + \sum_{t=1}^{T} \log p(x_t | s_t)$$

$$\pi_i = \frac{\sum_{i=1}^{M} \mathbb{I}\{S_1 = i\}}{M} = \frac{\text{\# of samples that start with stae } i}{\text{\# of all samples}}$$

UNIVERSITY OF ALBERTA

# MLE for Multinomial Distribution

- Counting
  - With one constraint $\pi_1 + \cdots \pi_n = 1$
  - You need to explicitly represent $\pi_n = 1 - \pi_1 - \cdots - \pi_{n-1}$
  - Or, you apply the Lagrangian multiplier method

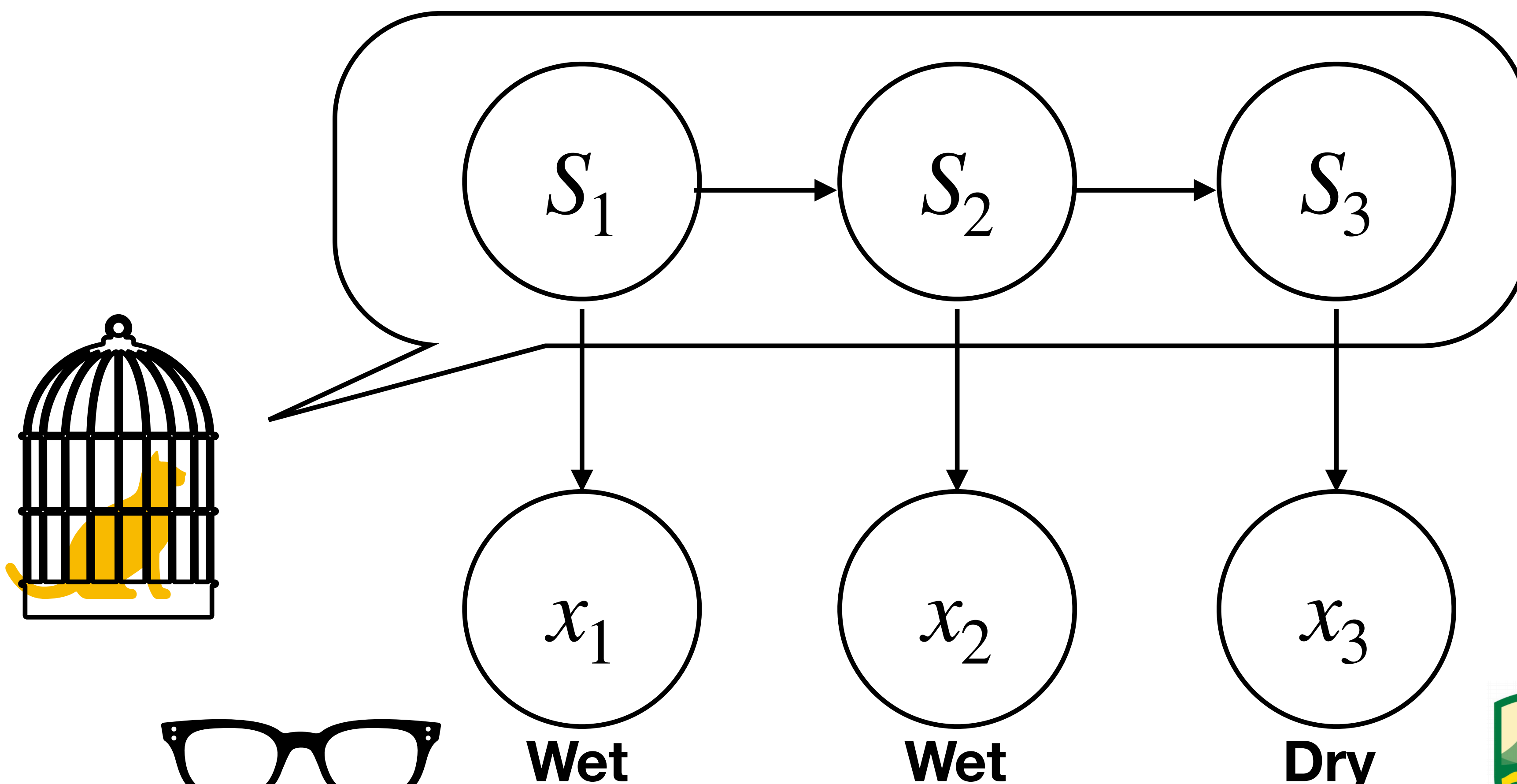
$$\log p(\cdot) = \log p(s_1) + \sum_{t=2}^{T} \log p(s_t | s_{t-1}) + \sum_{t=1}^{T} \log p(x_t | s_t)$$

**Written assignment**

UNIVERSITY OF ALBERTA

# Inference

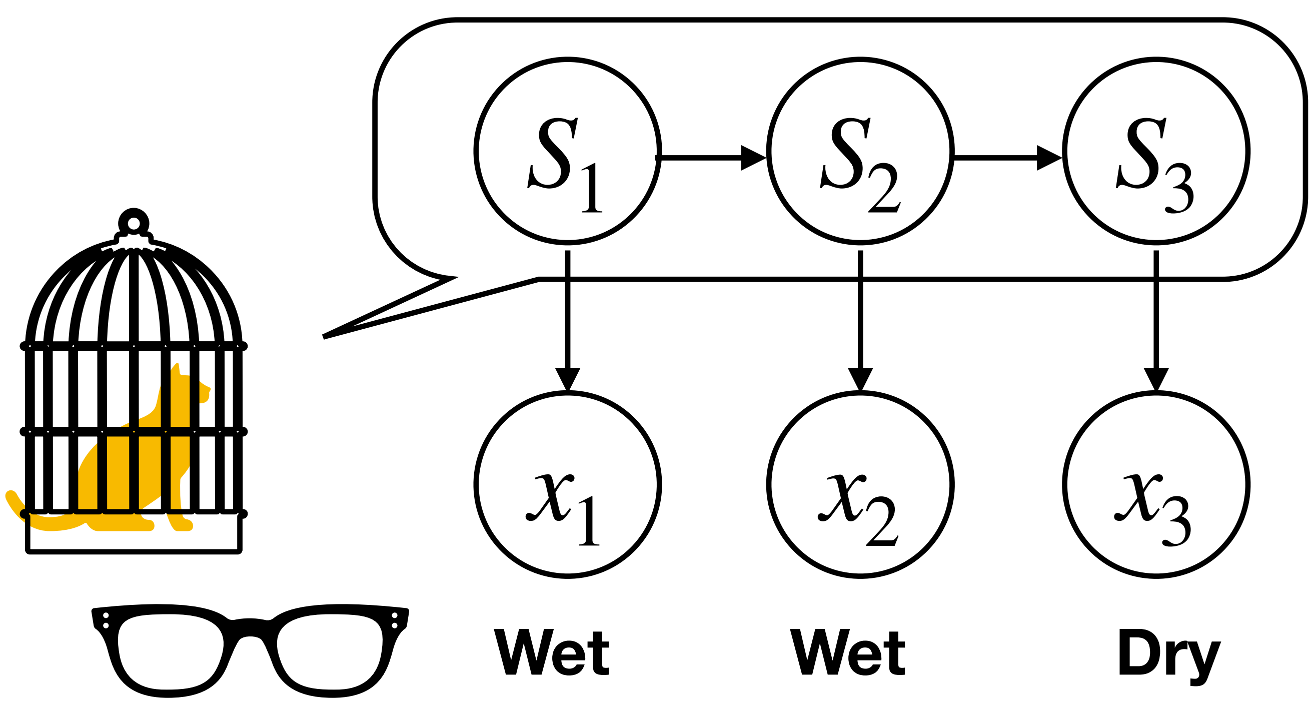
- Suppose the model is full trained
- During prediction, we observe $x_1, \cdots, x_T$
  - How can we know the states $s_1, \cdots, s_T$ that best explain $x_1, \cdots, x_T$?



# Inference Criteria

- We would like to predict the best (most probable) states
- Max *a posteriori* inference

$$
\begin{aligned}
s_1, \cdots, s_T &= \operatorname*{argmax}_{s_1, \cdots, s_T} p(s_1, \cdots, s_T \,|\, x_1, \cdots, x_T) \\
&= \operatorname*{argmax}_{s_1, \cdots, s_T} p(s_1, \cdots, s_T, x_1, \cdots, x_T)
\end{aligned}
$$

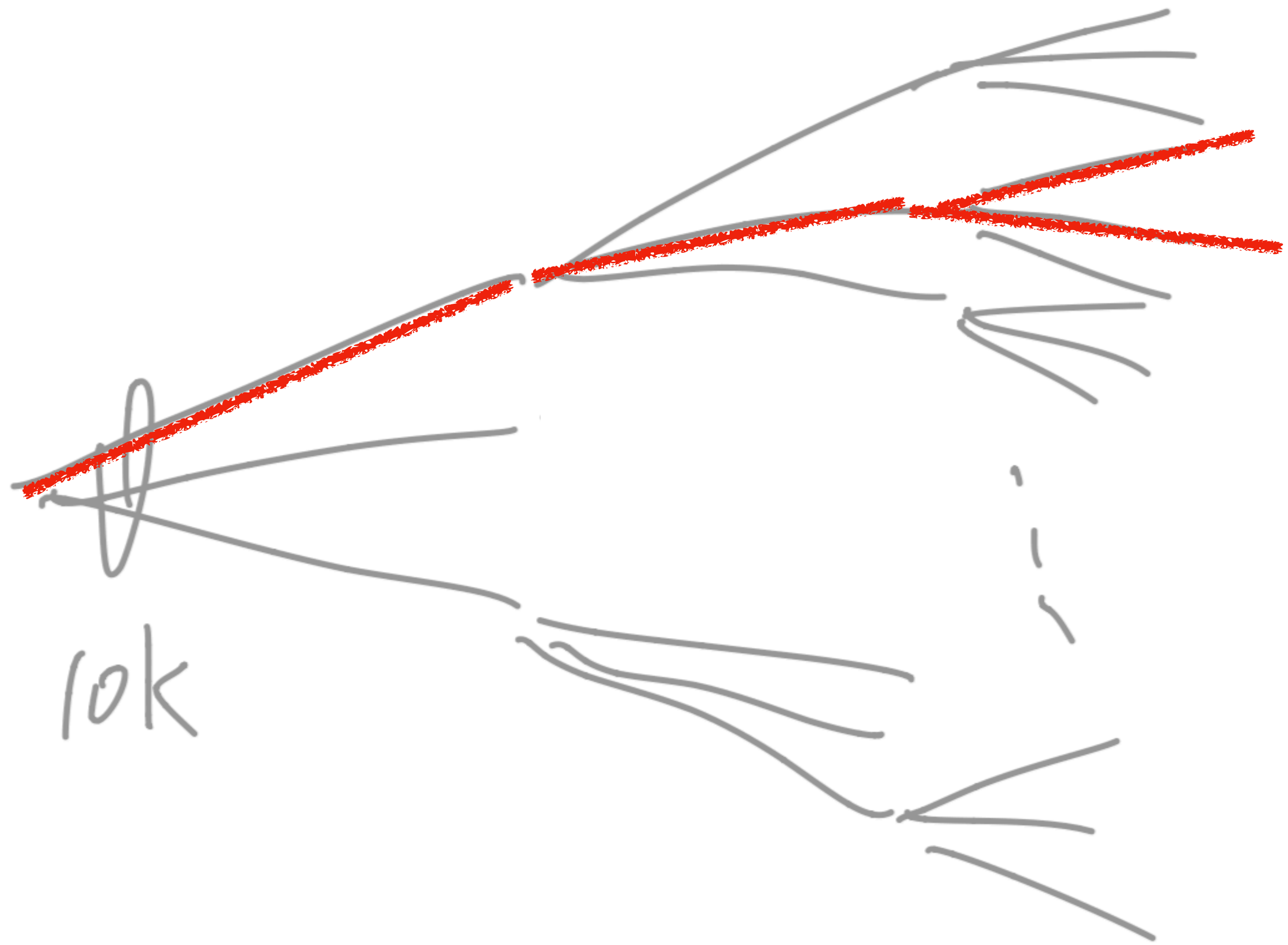


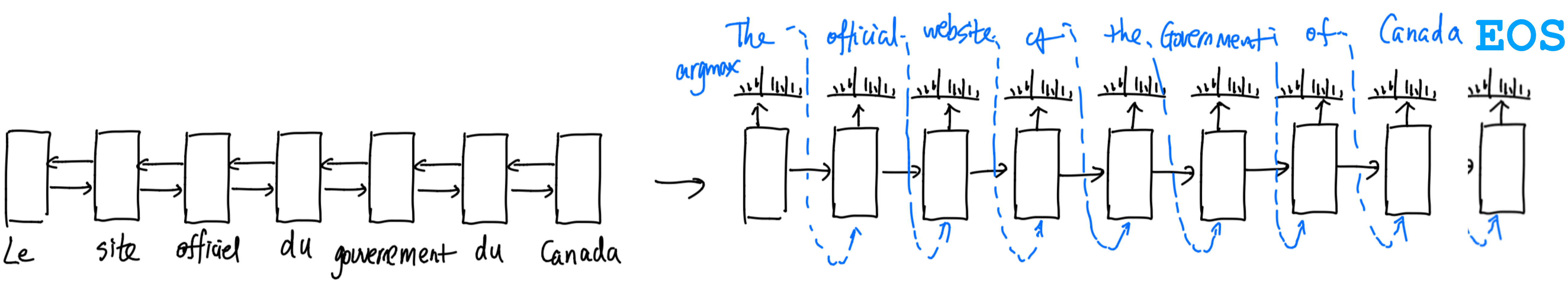

**Simplified notation may be used:**

$x_{1:t}, \quad x_1^t$

# Recall Beam Search

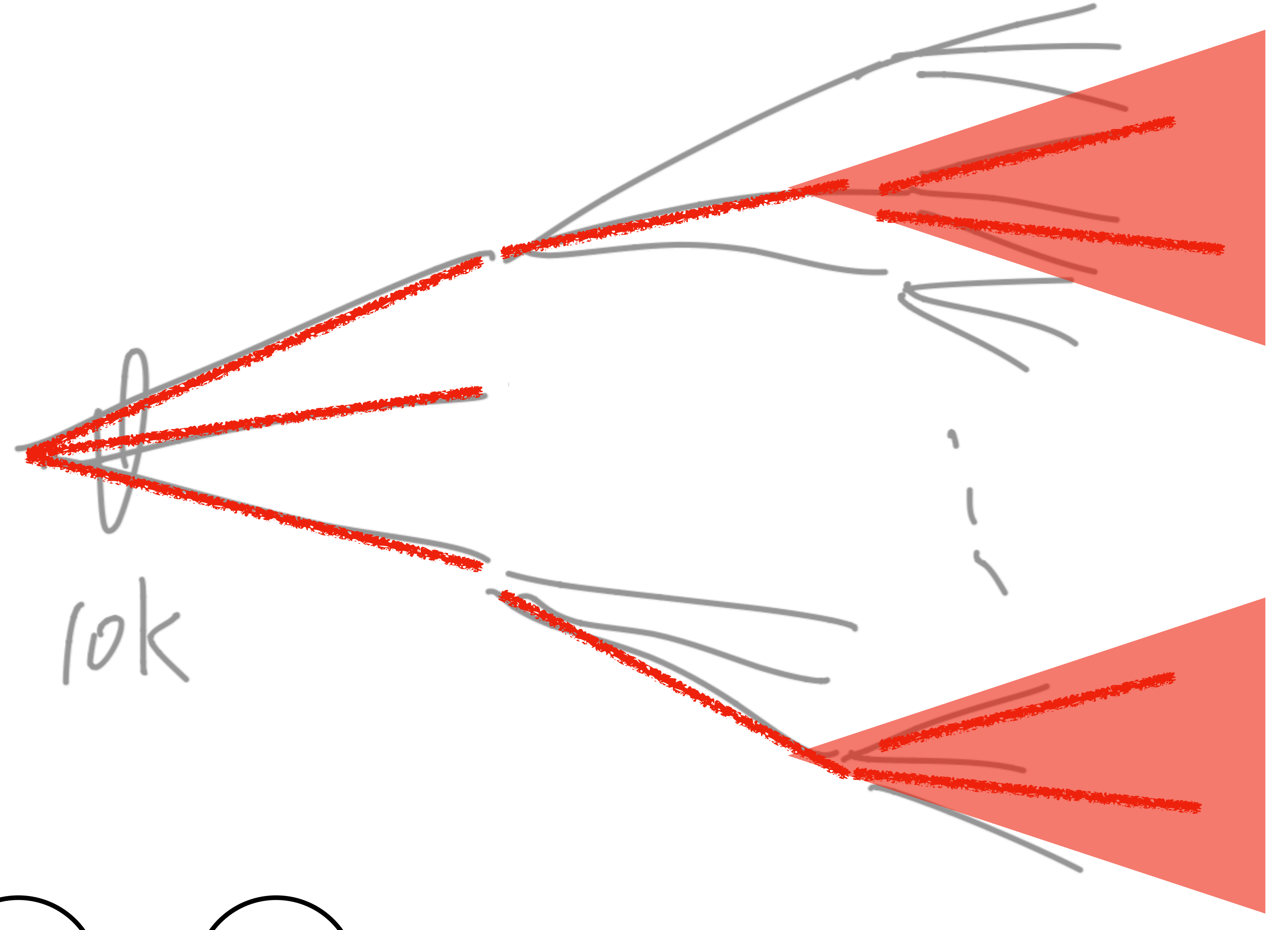
**B=2**

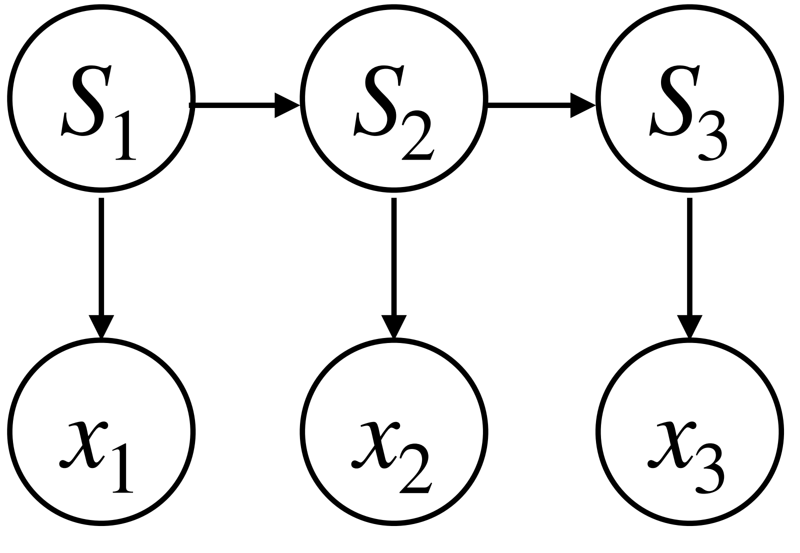


# Search in HMM

Some sub-structures are shared in different paths



# Markov Blanket

$$p(s_{1:T}, x_{1:T}) = \prod_{i=1}^{n} \Big[ p(s_i \mid s_{i-1}) p(x_i \mid s_i) \Big]$$

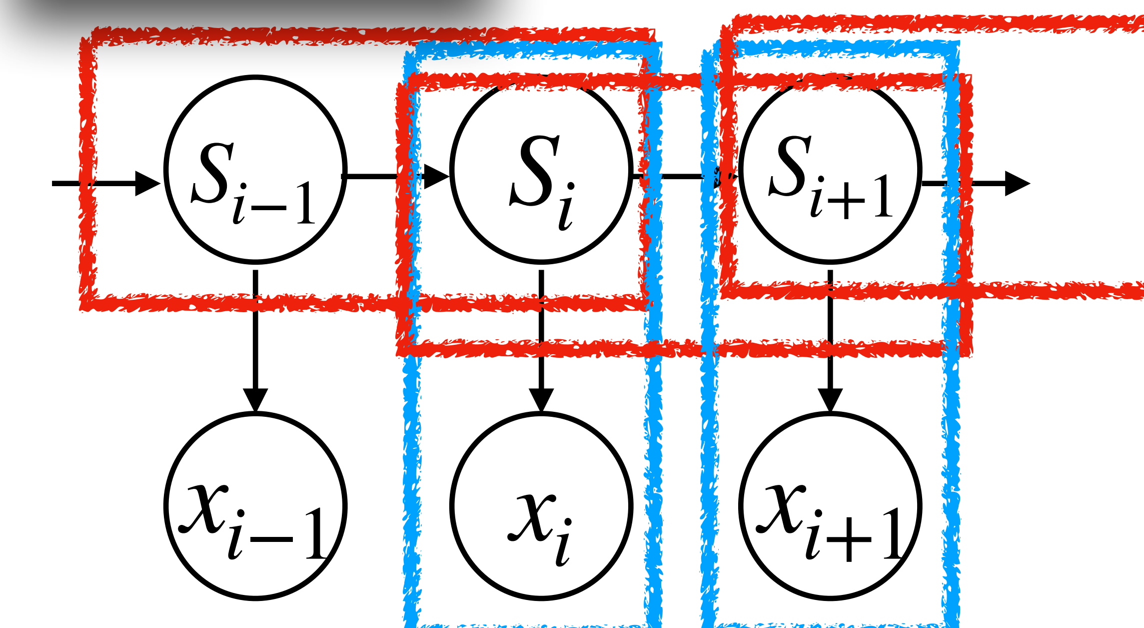
For simplicity, the first state's probability is denoted as

$$\mathbb{P}[s_1] \triangleq p(s_1 \mid s_0)$$

**Key observation:**

Factorized probability is local.

- $s_{i:T}, x_{i:T}$ only depends on $s_{i-1}$
- but not $s_{\le i-2}, x_{\le i-1}$



# Recursion Variable

$$s_1, \cdots, s_T = \underset{s_1, \cdots, s_T}{\operatorname{argmax}}\, p(s_1, \cdots, s_T, x_1, \cdots, x_T)$$

$$p(s_{1:T}, x_{1:T}) = \prod_{i=1}^{n} \left[ p(s_i \mid s_{i-1}) p(x_i \mid s_i) \right]$$

- Attempt#1: $\max_{s_{1:t}} p(x_1, \cdots, x_i, s_t)$
  - But best choice for every step $\neq$ best choice globally
- Attempt#2: $\max_{s_{1:t-1}} p(x_1, \cdots, x_t, s_t)$, for $s_t$ being any state

$$M[t][j] \triangleq \max_{s_{1:t-1}} p(x_{1:t}, S_t = j)$$

# Dynamic Programming

$$M[t][j] \triangleq \max_{1:t-1} p(x_{1:t}, S_t = j)$$

**Initialization**

$$M[1][j] = \max_{\varnothing} p(x_1, S_1 = j)$$ [nothing to choose for "max"]

$$= p(x_1, S_1 = j)$$

$$= p(S_1 = j) p(x_1 \mid S_1 = j)$$

$$= \pi_j \cdot p(x_1 \mid s_1 = j)$$ [both are model parameters]

# Dynamic Programming

$$M[t][j] \triangleq \max_{1:t-1} p(x_{1:t}, S_t = j)$$

## Recursion Step

- Assume $M[t-1][j] = \max_{s_{1:t-2}} p(x_{1:t-1}, S_{t-1} = j)$ known $(\forall j)$
- Goal: Figure out $M[t][j]$

$$M[t][j] = \max_{s_{1:t-1}} p(x_1, \cdots, x_t, S_t = j)$$

$$= \max_{s_{1:t-1}} p(x_1, \cdots, x_{t-1}, s_{t-1}) p(s_t = j \mid s_{t-1}) p(x_t \mid s_j)$$

$$= \max_{s_t} \max_{s_{1:t-2}} p(x_1, \cdots, x_{t-1}, s_{t-1}) p(s_t = j \mid s_{t-1}) p(x_t \mid s_j)$$

Known by recursion assumption $M[t-1][s_t]$

# Dynamic Programming

$$M[t][j] \triangleq \max_{1:t-1} p(x_{1:t}, S_t = j)$$

## Recursion Step

- Assume $M[t-1][j] = \max_{s_{1:t-2}} p(x_{1:t-1}, S_{t-1} = j)$ known $(\forall j)$
- Goal: Figure out $M[t][j]$

$$M[t][j] = \max_{s_{1:t-1}} p(x_1, \cdots, x_t, S_t = j)$$

$$= \max_{s_{1:t-1}} p(x_1, \cdots, x_{t-1}, s_{t-1}) p(S_t = j \mid s_{t-1}) p(x_t \mid S_t = j)$$

$$= \max_{s_t} \max_{s_{1:t-2}} p(x_1, \cdots, x_{t-1}, s_{t-1}) p(S_t = j \mid s_{t-1}) p(x_t \mid S_t = j)$$

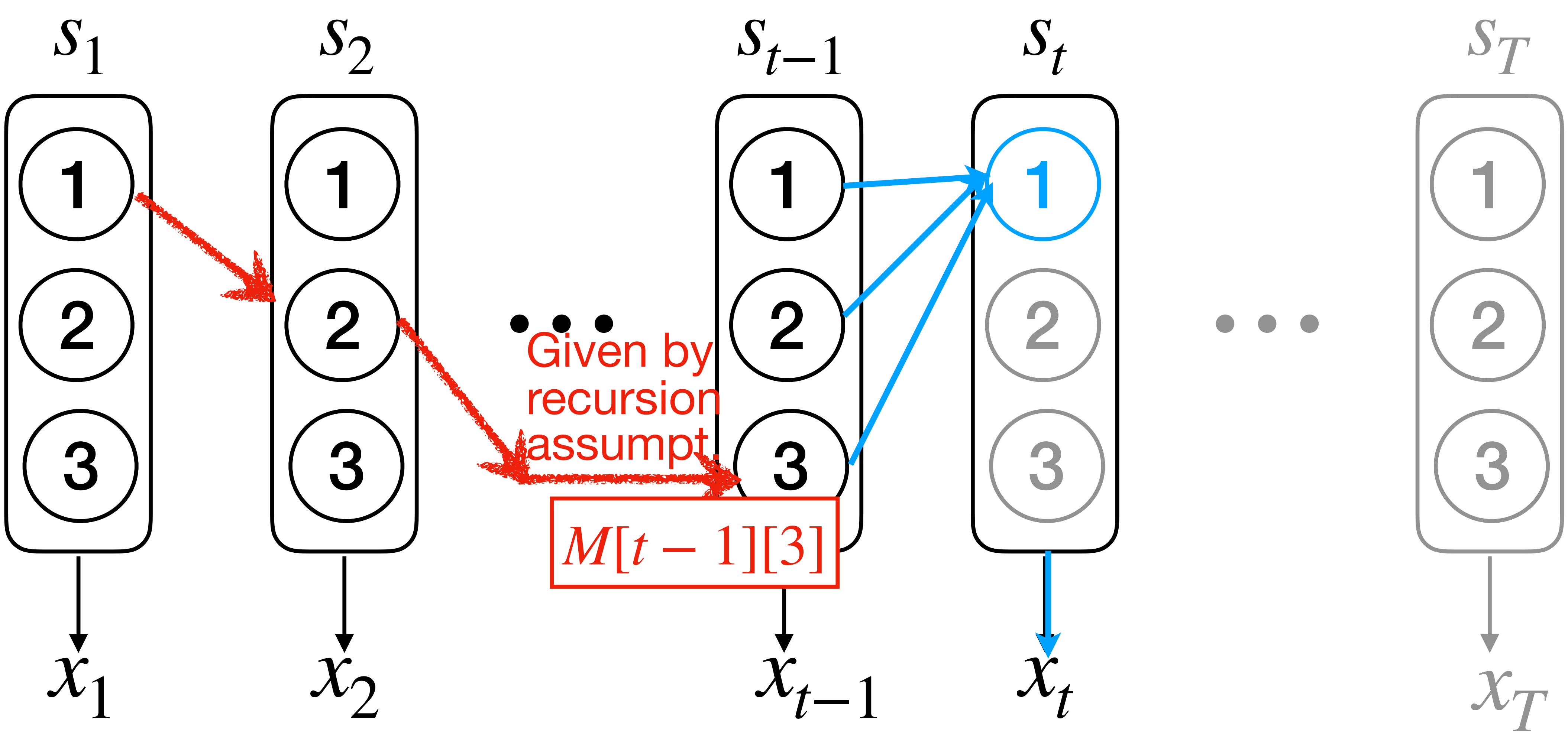
Known by recursion assumption $M[t-1][s_t]$

# Illustration

$$M[t][j] \triangleq \max_{1:t-1} p(x_{1:t}, S_t = j)$$

**Recursion Step**

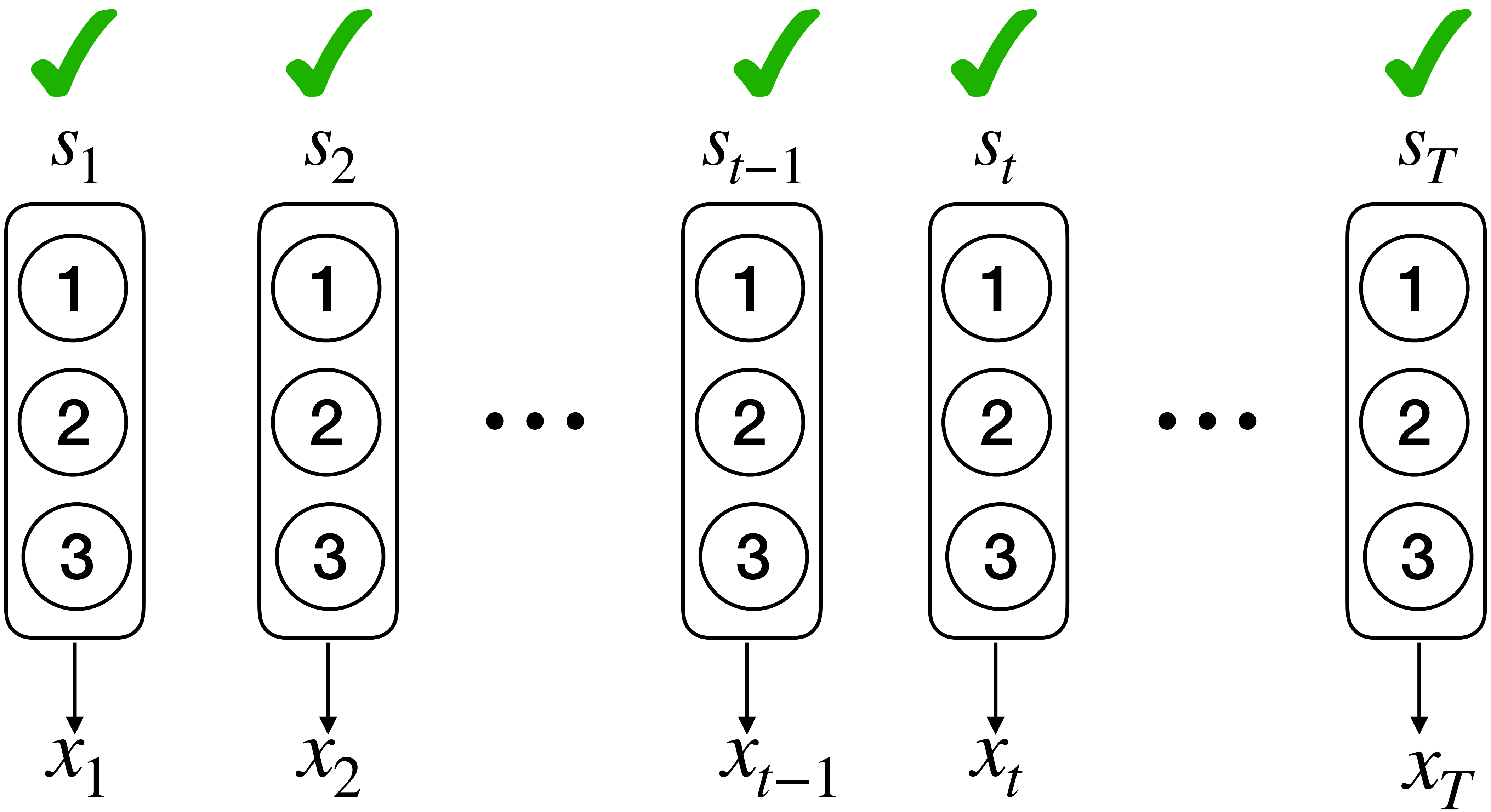
- Assume $M[t-1][j] = \max_{s_{1:t-2}} p(x_{1:t-1}, S_{t-1} = j)$ known
- Goal: Figure out $M[t][j]$ $(\forall j)$




$M[t][j] = \max_{s_{t-1}} \{ \rightarrow \rightarrow \rightarrow \;\; \nearrow \downarrow \}$

# Dynamic Programming

**Termination:** $M[T][j]$ is done $(\forall j)$

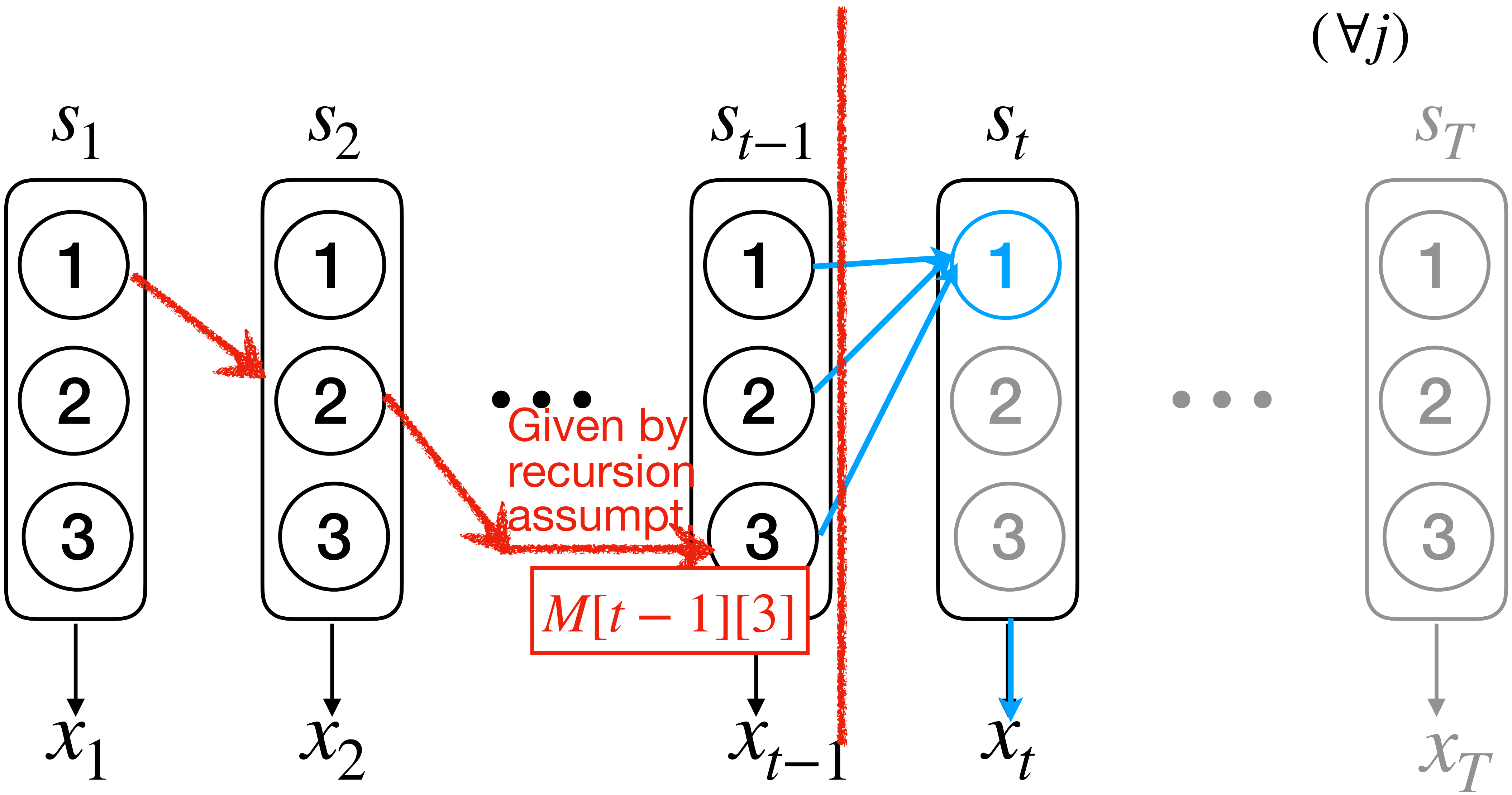


# Backtracking the States

$$M[t][j] \triangleq \max_{1:t-1} p(x_{1:t}, S_t = j)$$

$$B[t][j] = \text{argmax}_{i}\{M[t-1][i] \cdot P(S_t = j \mid S_{t-1} = i) \cdot P(x_t \mid S_j)\}$$

$(\forall j)$




$M[t][j] = \max_{s_{t-1}} \{ \rightarrow \rightarrow \rightarrow \nearrow \downarrow \}$

# Written Assignment

- Suppose an HMM is given
  - States $S = \{1, \cdots, n\}$
  - Parameters $\pi_j$, $P(S_{t-1} = j \mid S_t = i)$, $P(x_t \mid S_t = j)$ known
- Goal
  - To find the state and output sequences of length $T$ that have the highest jointly probability

$$s_{1:T}, x_{1:T} = \underset{s_{1:T}, x_{1:T}}{\operatorname{argmax}}\, p(s_{1:T}, x_{1:T})$$

  - Think of the problem $x_{1:T} = \operatorname{argmax}_{x_{1:T}} p(x_{1:T})$ [optional]

UNIVERSITY OF ALBERTA

# Written Assignment

- Requirements
  - Design a DP algorithm, stating the initialization, recursion, and termination of the algorithm

    (don't forget backpointers)
  - For any recursion variable, a clear definition is needed
  - The recursion step should be supported by derivation
  - Give pseudo code that generates $s_{1:T}, x_{1:T}$

UNIVERSITY OF ALBERTA

# Written Assignments

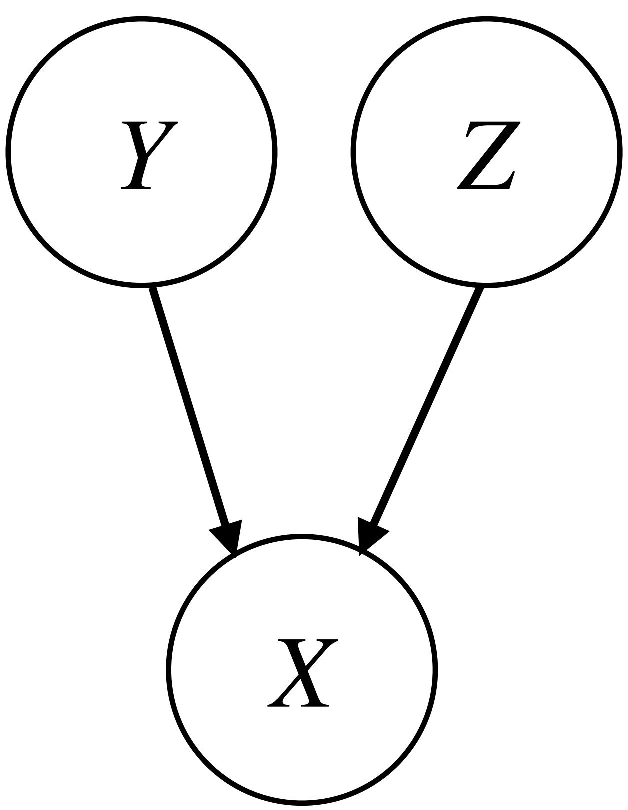
- Every week, we solve problems that have been mentioned in Monday's and Wednesday’s lectures.
- Every assignment is due on next Monday
- Automatically extended to next Wednesday [**before class**]
- Further extensions require good reasons (self-approved extension may result in 0 mark).

# Problem 1

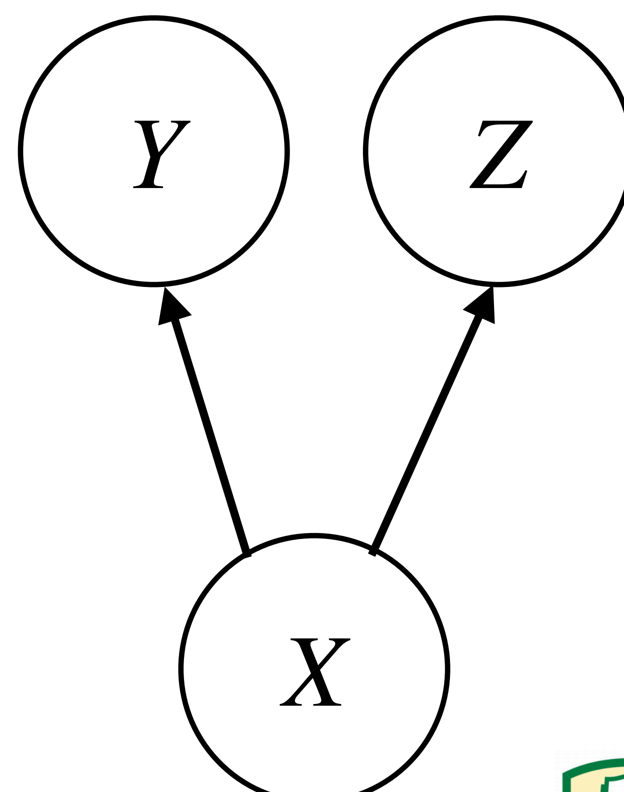
Show that $Y \perp Z | X$ does not hold in general for BN (1), but $Y \perp Z | X$ must be true for BN (2).

**Note:** If your solution involves showing some example, please provide your own example.

**(1)**




**(2)**




UNIVERSITY OF ALBERTA

# Problem 2

Give the MLE estimation for HMM transition and emission probabilities

- Figure out what are the parameters
- Give the formula to estimate these parameters (either by indicator functions or natural language expressions)

It's strongly recommended to derive MLE for multinomial distributions, but is optional for this assignment.

$$\log p(\cdot) = \log p(s_1) + \boxed{\sum_{t=2}^{T} \log p(s_t \mid s_{t-1})} + \boxed{\sum_{t=1}^{T} \log p(x_t \mid s_t)}$$

$$\pi_i = \frac{\sum_{i=1}^{M} \mathbb{I}\{S_1 = i\}}{M} = \frac{\text{\# of samples that start with state } i}{\text{\# of all samples}}$$

# Problem 3

- Suppose an HMM is given
  - States $S = \{1, \cdots, n\}$
  - Parameters $\pi_j$, $P(S_{t-1} = j \mid S_t = i)$, $P(x_t \mid S_t = j)$ known
- Goal
  - To find the state and output sequences of length $T$ that have the highest jointly probability

$$s_{1:T}, x_{1:T} = \underset{s_{1:T}, x_{1:T}}{\operatorname{argmax}}\, p(s_{1:T}, x_{1:T})$$

  - Think of the problem $x_{1:T} = \operatorname{argmax}_{x_{1:T}} p(x_{1:T})$ [optional]

# Problem 3

- Requirements
  - Design a DP algorithm, stating the initialization, recursion, and termination of the algorithm

    (don't forget back pointers)
  - For any recursion variable, a clear definition is needed
  - The recursion step should be supported by derivation
  - Give pseudo code that generates $s_{1:T}, x_{1:T}$

# Thank you!

Q&A