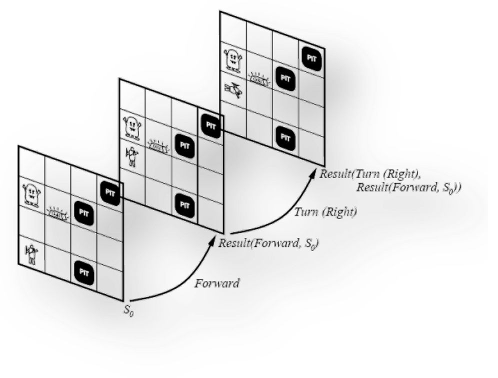
# Introduction to Artificial Intelligence

**Roman Barták**
Department of Theoretical Computer Science and Mathematical Logic

In **situation calculus**, we view the world as a series of snapshots (**time slices**). A similar approach can be applied in **probabilistic reasoning about time**.



Each time slice (**state**) is described as a set of random variables:

- **hidden** (not observable) random **variables $\mathbf{X}_t$** *describe the actual state*
- **observable** random **variables $\mathbf{E}_t$** (with observed values $\mathbf{e}_t$) *describe what we observe about the state*

**t** is an identification of the time slice (we assume **discrete time** with uniform time steps)

*Notation:*

$X_{a:b}$ denotes a set of variables from $X_a$ to $X_b$

## *Formal model*

We need to describe evolution of states and how observations depend on states.

### Transition model

specifies the probability distribution over the latest state variables given the previous values $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{0:t-1})$

*Simplifying assumptions:*

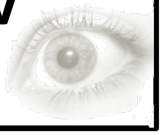
- state depends on previous state only (**Markov assumption**): $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$
- all transitions tables $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ are identical for all t (**stationary process**)

### Sensor (observation) model

describes how the evidence (observed) variables $\mathbf{E}_t$ depend on other variables $\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_{0:t}, \mathbf{E}_{1:t-1})$

*Simplifying assumption:*

- observation depends on current state only (**sensor Markov assumption**): $\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_{0:t}, \mathbf{E}_{1:t-1}) = \mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)$



## *Working example: umbrella world*

You are the security guard stationed at a secret underground installation and you want to know whether it is **raining today**:

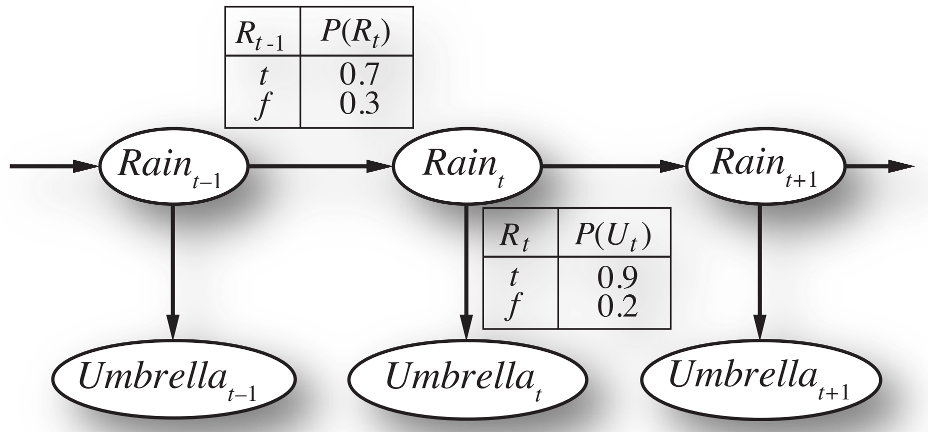
– hidden random variable $\mathbf{R}_t$

But your only access to the outside world occurs each morning when you see the the director coming in **with, or without, an umbrella**.

– observable random variable $\mathbf{U}_t$

The transition and sensor models can be described using a **Bayesian network**.

In addition to $\mathbf{P}(\mathbf{X}_t \mid \mathbf{X}_{t-1})$ and $\mathbf{P}(\mathbf{E}_t \mid \mathbf{X}_t)$ we need to say how everything gets started $\mathbf{P}(\mathbf{X}_0)$ (= $\langle 0.5, 0.5 \rangle$ , for example).



| $R_{t-1}$ | $P(R_t)$ |
|---|---|
| $t$ | 0.7 |
| $f$ | 0.3 |

| $R_t$ | $P(U_t)$ |
|---|---|
| $t$ | 0.9 |
| $f$ | 0.2 |

We have a specification of the complete joint distribution:

$$\mathbf{P}(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = \mathbf{P}(\mathbf{X}_0)\, \Pi_i\, \mathbf{P}(\mathbf{X}_i \mid \mathbf{X}_{i-1})\, \mathbf{P}(\mathbf{E}_i \mid \mathbf{X}_i)$$

## *Basic inference tasks*

**Filtering**: the task of computing the posterior distribution over the *most recent state,* given all evidence to date

$P(X_t \mid \mathbf{e}_{1:t})$

Where am I now?

**Prediction**: the task of computing the posterior distribution over the *future state*, given all evidence to date

$P(X_{t+k} \mid \mathbf{e}_{1:t})$ for $k>0$

Where will I be in future?

**Smoothing**: the task of computing posterior distribution over a *past state*, given all evidence up to the present

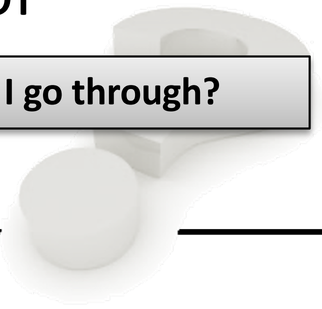
$P(X_k \mid \mathbf{e}_{1:t})$ for $k: 0 \leq k < t$

Where was I in past?

**Most likely explanation**: the task to find the sequence of states that is most likely generated a given sequence of observations

$\text{argmax}_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t} \mid \mathbf{e}_{1:t})$

What path did I go through?



The task of computing the posterior distribution over the *most recent state,* given all evidence to date – $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$.

A useful filtering algorithm needs to maintain a current state estimate and update it, rather than going back over (**recursive estimation**):

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = f(\mathbf{e}_{t+1},\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}))$$

How to define the function f?

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t},\mathbf{e}_{t+1})\\ &= \alpha\, \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1},\mathbf{e}_{1:t})\, \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})\\ &= \alpha\, \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\, \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})\\ &= \alpha\, \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\, \Sigma_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t,\mathbf{e}_{1:t})\, P(\mathbf{x}_t|\mathbf{e}_{1:t})\\ &= \alpha\, \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\, \Sigma_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)\, P(\mathbf{x}_t|\mathbf{e}_{1:t})\end{aligned}$$

**Bayes rule**

**sensor Markov assumption**

**conditioning**

A message $\mathbf{f}_{1:t}$ is propagated forward over the sequence:

$$\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}) = \mathbf{f}_{1:t}$$

$$\mathbf{f}_{1:t+1} = \alpha\, \text{FORWARD}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$$

$$\mathbf{f}_{1:0} = \mathbf{P}(\mathbf{X}_0)$$

The task of computing the posterior distribution over the *future state*, given all evidence to date – $\mathbf{P}(\mathbf{X}_{t+k} \mid \mathbf{e}_{1:t})$ for some k>0.

We can see this task as filtering without the addition of new evidence:

$$\mathbf{P}(\mathbf{X}_{t+k+1} | \mathbf{e}_{1:t}) = \Sigma_{\mathbf{x}_{t+k}} \mathbf{P}(\mathbf{X}_{t+k+1} | \mathbf{x}_{t+k})\ P(\mathbf{x}_{t+k} | \mathbf{e}_{1:t})$$

After some time (**mixing time**) the predicted distribution converges to the **stationary distribution** of the Markov process and remains constant.

The task of computing posterior distribution over a *past state*, given all evidence up to the present – $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for k: $0 \leq k < t$.

We again exploit a recursive message-passing approach, now in two directions.

$$\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t}) = \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k},\mathbf{e}_{k+1:t})$$
$$= \alpha\, \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\, \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{e}_{1:k})$$
$$= \alpha\, \mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:k})\, \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k)$$
$$= \alpha\, \mathbf{f}_{1:k} \times \mathbf{b}_{k+1:t}$$

**Bayes rule**

**conditional independence**

$$\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) = \Sigma_{\mathbf{x}_{k+1}} \mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k,\mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)$$
$$= \Sigma_{\mathbf{x}_{k+1}} P(\mathbf{e}_{k+1:t}|\mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)$$
$$= \Sigma_{\mathbf{x}_{k+1}} P(e_{k+1},\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)$$
$$= \Sigma_{\mathbf{x}_{k+1}} P(e_{k+1}|\mathbf{x}_{k+1})\, P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)$$

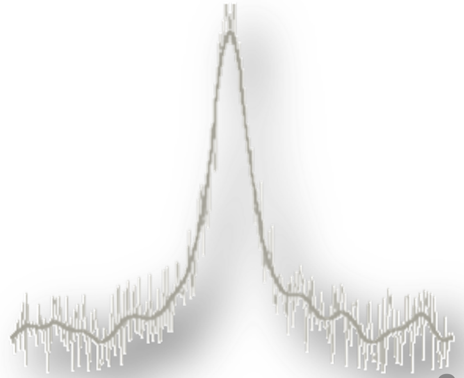
**conditioning**

**conditional independence**

**conditional independence**

Using the backward message-passing notation:

$$\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) = \mathbf{b}_{k+1:t}$$
$$\mathbf{b}_{k+1:t} = \text{BACKWARD}(\mathbf{b}_{k+2:t}, \mathbf{e}_{k+1})$$
$$\mathbf{b}_{t+1:t} = \mathbf{P}(\mathbf{e}_{t+1:t}|\mathbf{X}_t) = \mathbf{P}(\,|\mathbf{X}_t) = \mathbf{1}$$

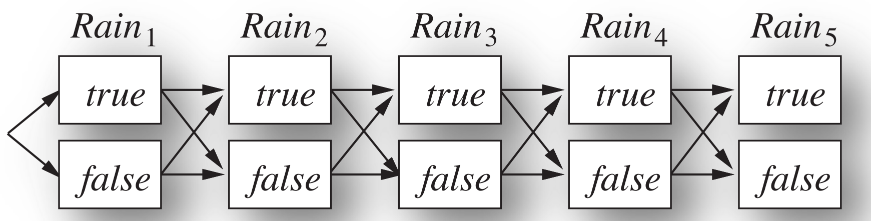


## *Most likely explanation/sequence*

The task to find the *sequence of states that is most likely* generated a given sequence of observations – $\operatorname{argmax}_{x_{1:t}} P(x_{1:t} \mid \mathbf{e}_{1:t})$.

Note: This is different from smoothing for each past state and taking the sequence of most probable states!

We can see each sequence as a **path through a graph** whose nodes are possible states at each time step.



The most likely path to a given state consists of the most likely path to some previous state followed by a transition to that state.

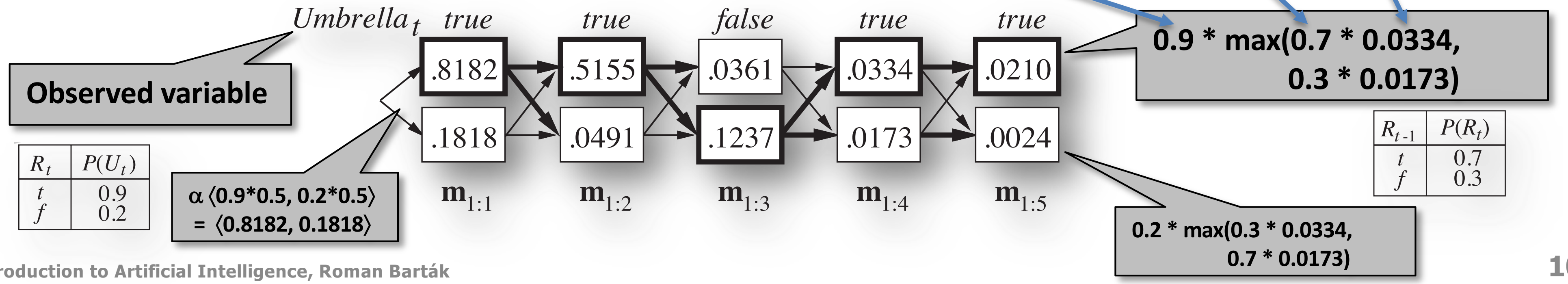
This can be described using a **recursive formula** (**Viterbi algorithm**):

$$\max_{\mathbf{x}_1,\ldots,\mathbf{x}_t} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_t,\mathbf{X}_{t+1} \mid \mathbf{e}_{1:t+1})$$

$$= \alpha\, \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} (\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t) \max_{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}} P(\mathbf{x}_1,\ldots,\mathbf{x}_t \mid \mathbf{e}_{1:t}))$$

Again, we use an approach of forward message passing:

$m_{1:t} = \max_{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}} \mathbf{P}(\mathbf{x}_1,\ldots,\mathbf{x}_{t-1},\mathbf{X}_t \mid \mathbf{e}_{1:t})$, $\quad m_{1:t+1} = \mathbf{P}(\mathbf{e}_{t+1} \mid \mathbf{X}_{t+1}) \max_{\mathbf{x}_t} (\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{x}_t)\, m_{1:t})$



| $R_t$ | $P(U_t)$ |
|---|---|
| $t$ | 0.9 |
| $f$ | 0.2 |

| $R_{t-1}$ | $P(R_t)$ |
|---|---|
| $t$ | 0.7 |
| $f$ | 0.3 |

## Hidden Markov models

Assume that the state of process is described by a single discrete random variable $X_t$ (there is also a single evidence variable $E_t$).

This is called a **hidden Markov model** (HMM).

This restricted model allows for a simple and elegant **matrix implementation** of all the basic algorithms.

Assume that variable $X_t$ takes values from the set {1,…S}, where S is the number of possible states.

The **transition model P**$(X_t \mid X_{t-1})$ becomes an S×S matrix **T**, where:

$$\mathbf{T}_{(i,j)} = P(X_t = j \mid X_{t-1}=i)$$

We also put the **sensor model** in matrix form. Now we know the value of the evidence variable $e_t$ so we describe $P(E_t = e_t \mid X_t=i)$, using a diagonal matrix $\mathbf{O}_t$, where:

$$\mathbf{O}_{t\,(i,i)} = P(E_t = e_t \mid X_t=i)$$

**The forward message propagation** (from filtering)

$\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}) = \mathbf{f}_{1:t}$

$\mathbf{f}_{1:t+1} = \alpha\, \mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}) \sum_{\mathbf{x}_t} \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)\, P(\mathbf{x}_t|\mathbf{e}_{1:t})$

can be reformulated using matrix operations (message $\mathbf{f}_{1:t}$ is modelled as a one-column matrix) as follows:

$\mathbf{T}_{(i,j)} = P(X_t = j \mid X_{t-1}=i)$

$\mathbf{O}_{t\,(i,i)} = P(E_t = e_t \mid X_t=i)$

$\mathbf{f}_{1:t+1} = \alpha\, \mathbf{O}_{t+1}\, \mathbf{T}^{T}\, \mathbf{f}_{1:t}$

**The backward message propagation** (from smoothing)

$\mathbf{P}(\mathbf{e}_{k+1:t}|\mathbf{X}_k) = \mathbf{b}_{k+1:t}$

$\mathbf{b}_{k+1:t} = \sum_{\mathbf{x}_{k+1}} P(e_{k+1}|\mathbf{x}_{k+1})\, P(\mathbf{e}_{k+2:t}|\mathbf{x}_{k+1})\, \mathbf{P}(\mathbf{x}_{k+1}|\mathbf{X}_k)$

can be reformulated using matrix operations (message $\mathbf{b}_{k:t}$ is modelled as a one-column matrix) as follows:

$\mathbf{b}_{k+1:t} = \mathbf{T}\, \mathbf{O}_{k+1}\, \mathbf{b}_{k+2:t}$

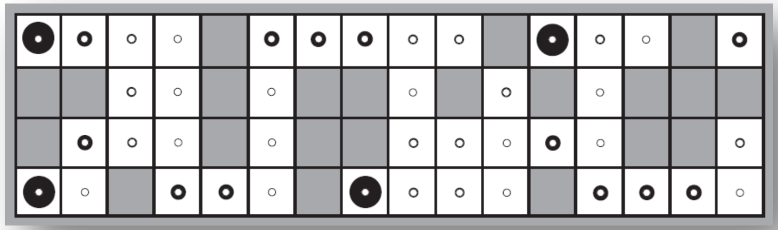
## Localization (an example of HMM)

Assume a robot that moves randomly in a grid world, has a map of the world and (noisy) sensors reporting obstacles laying immediately to the north, south, east, and west. The robot needs to find its location.

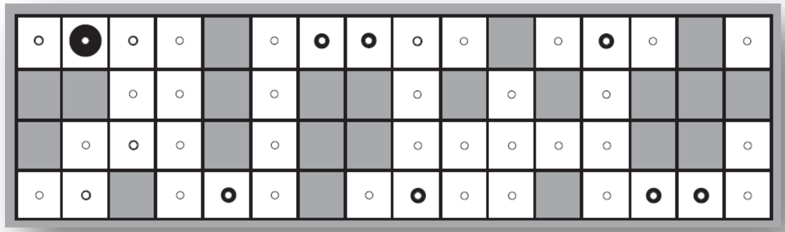
**A possible model:**

- **random variables** $X_t$ describe robot's location at times t
  - possible values are 1,..,n for n locations
  - Nb(i) – a set of neighboring locations for location i
- **transition tables** (random move)
  - $P(X_{t+1}=j \mid X_t=i) = 1/\,|Nb(i)|$, if $j \in Nb(i)$,
    0, otherwise
- **sensor variables** $E_t$ describe observations (evidence) at times t (four sensor for four directions NSEW)
  - values indicate detection of obstacle at a given direction NSEW (16 values for all directions)
  - assume that sensor's error rate is $\varepsilon$
- **sensor tables**
  - $P(E_t=e_t \mid X_t=i) = (1-\varepsilon)^{4-d_{it}} \varepsilon^{d_{it}}$
    where $d_{it}$ is the number of deviations of observation $e_t$ from the true values for square i

**$P(X_1 \mid E_1=NSW)$**



**$P(X_2 \mid E_1=NSW, E_2=NS)$**

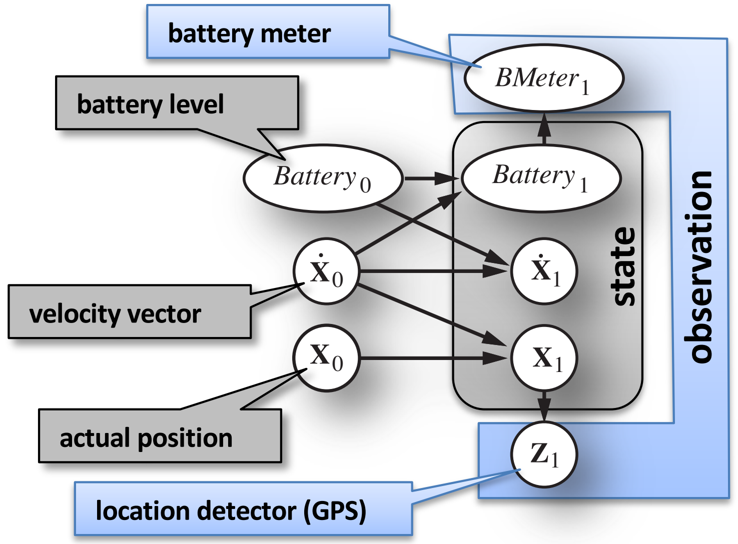


**Dynamic Bayesian network** (DBN) is a Bayesian network that represents a temporal probability model.

the variables and links are exactly replicated from slice to slice

It is enough to describe one slice.

- prior distribution $\mathbf{P}(\mathbf{X}_0)$
- transition model $\mathbf{P}(\mathbf{X}_1 \mid \mathbf{X}_0)$
- sensor model $\mathbf{P}(\mathbf{E}_1 \mid \mathbf{X}_1)$

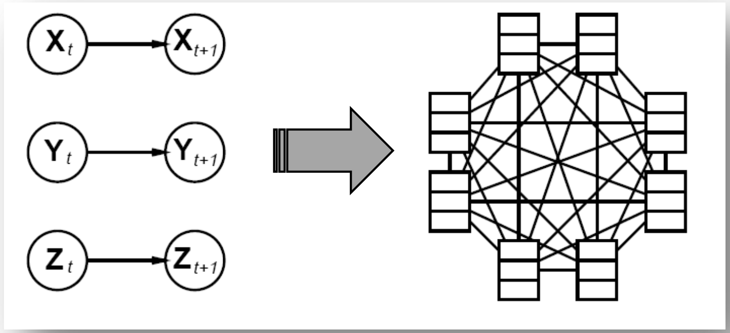


Each state variable has parents either at the same slice or in the previous slice (Markov assumption).

A hidden Markov model is a special case of a dynamic Bayesian network.

Similarly, a dynamic Bayesian network can be encoded as a hidden Markov model

one random variable in HMM whose values are n-tuples of values of state variables in DBN



**What is the difference?**

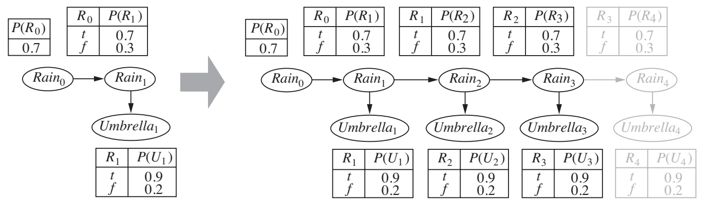
The relationship between DBN and HMM is roughly analogous to the relationship between ordinary Bayesian networks and full tabulated joint distribution.

- DBN with 20 Boolean state variables, each of which has three parents
  - the transition model has $20 \times 2^3 = 160$ probabilities
- Hidden Markov model has one random variable with $2^{20}$ values
  - the transition model has $2^{20} \times 2^{20} \approx 10^{12}$ probabilities
  - HMM requires much more space and inference is much more expensive

## Inference in DBN

Dynamic Bayesian networks are Bayesian networks and we already have algorithms for inference in Bayesian networks.

We can construct the full Bayesian network representation of a DBN by replicating slices to accommodate the observations (**unrolling**).

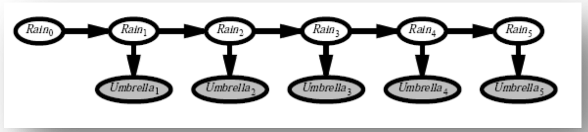


**Exact inference:**

- If applied naively, its complexity will increase with time (due to more slices).
- We can use **variable elimination** and keep in memory only last two slices (via summing out the variables from previous slices).
- The bad news are that "constant" space to represent the largest factor will **be exponential** in the number of state variables.

**Approximate inference:**

- We sample non-evidence nodes of the network in topological order, weighting each sample by the likelihood in accords to the observed evidence variables.
- But samples are generated completely independently of the evidence!
- Hence, the weights of samples will decrease so to keep accuracy we need to **increase the number of samples exponentially** with t.



We can exploit probability theory when reasoning about time. Specifically, when **transitions are uncertain,** and environment is **partially observable** via sensors.

We use **transition and observation models** with **Markov assumptions**.

**Basic inference tasks** (exploit recursive formulas):

- **filtering** (where am I now?)
- **prediction** (where will I be in future?)
- **smoothing** (where was I in past?)
- **most likely explanation** (what path did I go through?)

**Hidden Markov Model**

- one state variable and one observation variable
- simplified inference using matrix operations

**Dynamic Bayesian Network**

- compact representation via more CPTs