

26:198:722 Expert Systems

- Pearl's Bayesian Networks
- Lauritzen & Spiegelhalter
- Aalborg Architecture
- Shenoy-Shafer Architecture

Pearl's Bayesian Networks

Section 1

Goal:

A computational model for humans' inferential reasoning, namely, the mechanism by which people integrate data from multiple sources and generate coherent interpretations of that data.

Pearl's Bayesian Networks

Section 1.1

Belief networks formed by decomposition of the joint probability distribution

Also called Bayesian networks, causal networks

Strictly speaking, Bayesian networks form hypergraphs not graphs

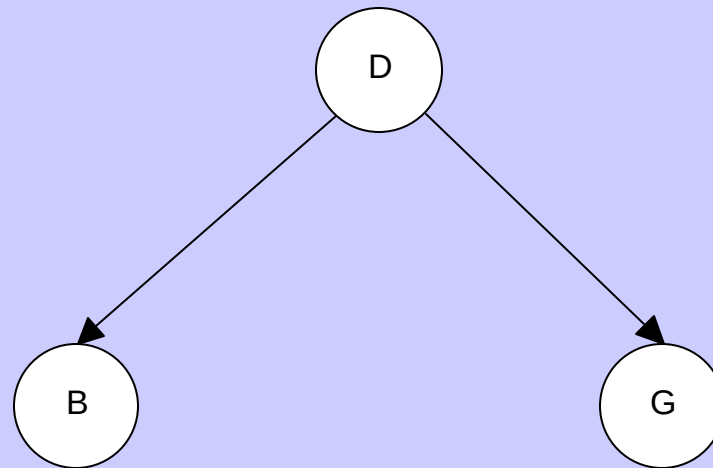
Pearl's Bayesian Networks

Section 1.2

Conditional independence and graph separation

- ◆ Two links meeting head-to-tail or tail-to-tail at node X are blocked by S if X is in S
- ◆ Two links meeting head-to-head at node X are blocked by S if neither X nor any of its descendants are in S
- ◆ A path is separated by S if at least one pair of links along the path is blocked by S
- ◆ S separates Y and Z if all paths between Y and Z are separated by S

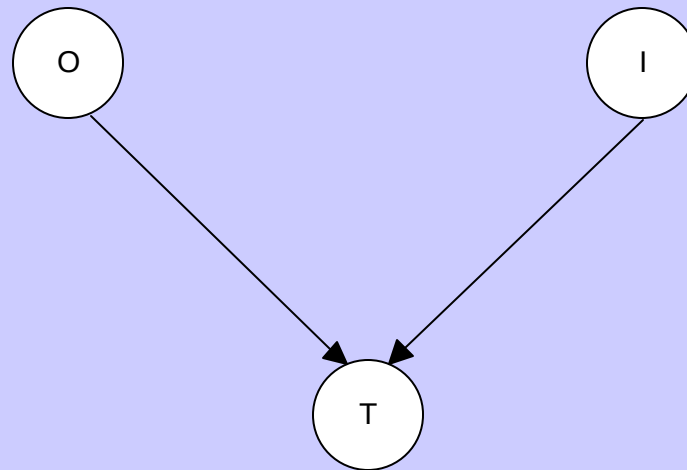
Pearl's Bayesian Networks



Does D separate B and G ?

Yes (D blocks B and G)

Pearl's Bayesian Networks



Does T separate O and I ?

No (T unblocks O and I)

Pearl's Bayesian Networks

Section 2.1

A local propagation mechanism

Distributed message passing paradigm

Pearl's Bayesian Networks

Section 2.2

Belief propagation in trees

Specific evidence and virtual evidence

Generalization of odds-likelihood version of Bayes' Rule

Pearl's Bayesian Networks

Section 2.2.1

Data fusion is based on the belief induced on some B by the data $D = D_B^- \cup D_B^+$

$$Bel(B_i) = a P(D_B^- | B_i) \cdot P(B_i | D_B^+) = a \cdot l(B_i) \cdot p(B_i)$$

where

$$l(B_i) = P(D_B^- | B_i) = \prod_k P(D^{k-} | B_i)$$

Pearl's Bayesian Networks

Section 2.2.2

There is no need to normalize the π -messages prior to transmission. This is done solely for the purpose of retaining the probabilistic meaning of these messages.

Pearl's Bayesian Networks

Section 2.2.2

When a processor B is activated it inspects the messages $p_B(A)$ from its father and $l_k(B)$ from its sons, computes new parameters

$$l(B_i) = \prod_k l_k(B_i) \text{ and}$$

$$p(B_i) = b \sum_j P(B_i | A_j) \cdot p_B(A_j)$$

and computes new messages for its father and sons:

Pearl's Bayesian Networks

Section 2.2.2

Bottom-up propagation

$$l_B(A_j) = \sum_i P(B_i | A_j) \cdot l(B_i)$$

Top-down propagation

$$p_k(B_i) = ap(B_i) \cdot \prod_{m \neq k} l_m(B_i)$$

Pearl's Bayesian Networks

Section 2.2.2

Special cases:

Anticipatory (uninstantiated) nodes:

$$l = (1, 1, \dots, 1)$$

Data (instantiated) nodes:

$$l = (0, \dots, 0, 1, 0, \dots, 0)$$

Dummy (virtual evidence) nodes:

$$l_B(A_i) = K \cdot P(\text{observation} | A_i)$$

Root nodes:

$$p(\text{root}) = \text{prior probability}$$

Pearl's Bayesian Networks

Section 2.3

Extension to singly-connected networks
(polytrees)

“When a physician discovers evidence in favor of one disease, it reduces the probability of other diseases” - this is the phenomenon of “explaining away”

Pearl's Bayesian Networks

Section 2.3

For a node A which is a common child of B and C and has children X and Y , propagation uses:

$$l_A(B_i) = a \sum_j \left[p_A(C_j) \cdot \sum_k l_X(A_k) \cdot l_Y(A_k) \cdot P(A_k | B_i, C_j) \right]$$

and

$$p_X(A_i) = a l_Y(A_i) \cdot \left[\sum_{jk} P(A_i | B_j, C_k) \cdot p_A(B_j) \cdot p_A(C_k) \right]$$

Pearl's Bayesian Networks

Section 2.4

Multiply-connected networks
cannot be handled by this mechanism

Proposals:

- Local interpretation by node collapsing

- Stochastic relaxation

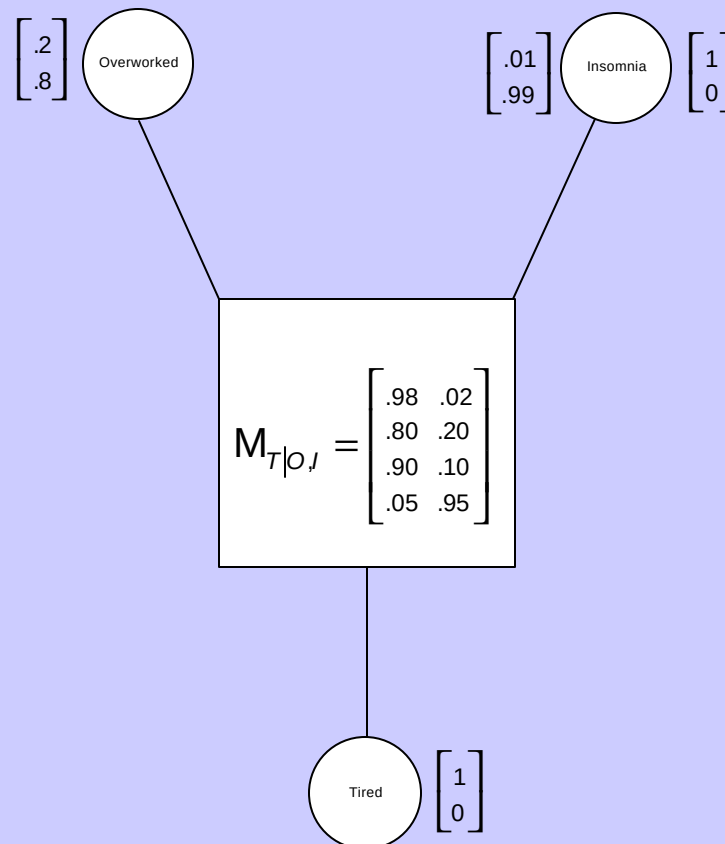
- Conditioning

- Pre-processing using auxiliary variables:

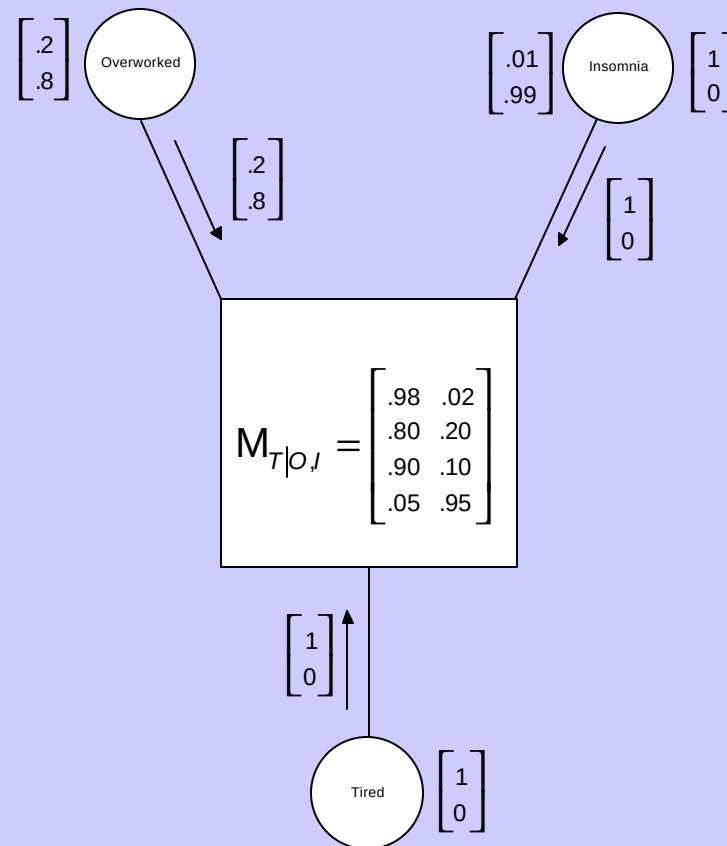
 - Star distribution

 - Star decomposition

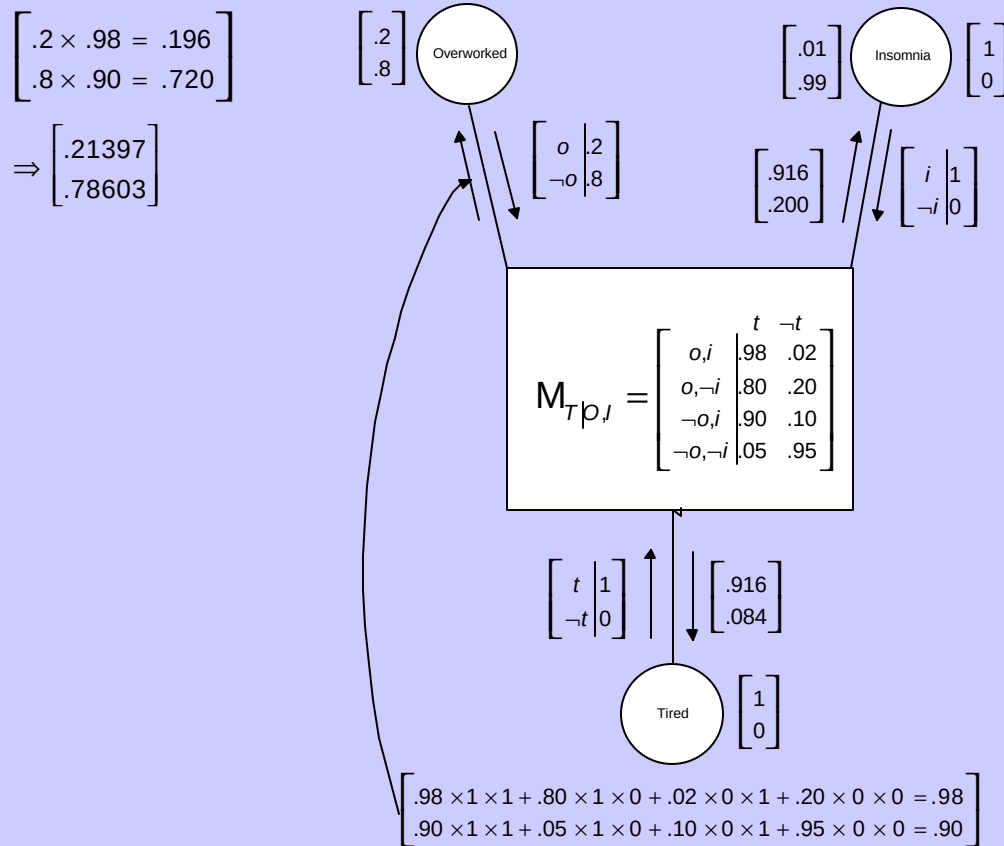
Assignment 4 - Question 1



Assignment 4 - Question 1



Assignment 4 - Question 1



Lauritzen & Spiegelhalter

- Emphasis on efficient computational schemes
- Assumes a fixed model currently being entertained and the numerical assessments are precisely specified
- Based on development of MUNIN
- Acknowledges much prior work

Lauritzen & Spiegelhalter

■ Main Issues:

- * Initialization
- * Absorption of evidence
- * Global propagation
- * Hypothesizing
- * Planning
- * Influential findings

Lauritzen & Spiegelhalter

- What is the significance of the comment “In particular, the event ‘the patient presented to the clinic’ is implicitly conditioned upon, and perhaps should be explicitly represented.”?

Lauritzen & Spiegelhalter

- As in Pearl's scheme, there is a fundamental assumption that the joint distribution is given as a product of priors and conditionals; e.g.,

$$P(a, t, x, e, d, l, b, s) =$$

$$P(a) \cdot P(t|a) \cdot P(x|e) \cdot P(e|t, d) \cdot P(d|e, b) \cdot P(l|s) \cdot P(b|s) \cdot P(s)$$

- Or, working with proportionality, as potentials:

$$\psi(a) \cdot \psi(t, a) \cdot \psi(x, e) \cdot \psi(e, t, l) \cdot \psi(d, e, b) \cdot \psi(l, s) \cdot \psi(b, a) \cdot \psi(s)$$

Lauritzen & Spiegelhalter

- Evidence potentials are not uniquely determined
- They may not, in general, be easy to interpret other than through the joint probability being proportional to their product
- Initially, however, we may set the potentials equal to the probability factors

Lauritzen & Spiegelhalter

- *Building the tree of cliques for propagation:*
 - * Form the moral graph by:
 - ◆ dropping directions on the links; and
 - ◆ marrying parents
 - * Add fill-ins to create a triangulated graph
 - * Identify the cliques of the triangulated graph
 - * Label nodes using maximal cardinality search, rank cliques according to the highest labeled node, to form a set chain with the running intersection property
 - * This is, of course, a join tree; arbitrarily select a root

Lauritzen & Spiegelhalter

- A subset of nodes of a graph is *complete* if every member is connected to every other member by an edge
- A *clique* is a maximal complete subset

Lauritzen & Spiegelhalter

- Maximum cardinality search
 - * Assign 1 to an arbitrary node
 - * Number the nodes consecutively, choosing as the next a node with a maximum number of previously numbered neighbors
 - * Break ties arbitrarily

Lauritzen & Spiegelhalter

■ Maximum cardinality fill-in

- * Assume the nodes are numbered as for maximum cardinality search
- * Working backwards from the last node to the first, consider the set $bd(i) \cap \{1, \dots, i-1\}$ where $bd(i)$ is the set of numbers of the neighbors of the i -th node
- * If this set is not complete, add whatever fill-in edges are necessary to make it so

Lauritzen & Spiegelhalter

■ Running intersection property

- * Suppose we have an ordered set of cliques C_i
- * Then the set of cliques has the *running intersection property* if, for each i , all the nodes of C_i that are in any earlier cliques are all members of one particular earlier clique; i.e.,

$$\forall i \geq 2, \exists j < i : C_i \cap (C_1 \cup \dots \cup C_{i-1}) \subseteq C_j$$

Lauritzen & Spiegelhalter

■ Rules for propagation

* Inward pass

- ◆ Each node waits to send its message to its inward neighbor until it has received messages from all its outward neighbors
- ◆ When a node is ready to send its message to its inward neighbor, it computes the message by marginalizing its current potential to its intersection with its inward neighbor; it sends this marginal to the inward neighbor, and then divides its own current potential by it
- ◆ When a node receives a message from its outward neighbor, it replaces its own current potential with the product of that potential and the message

Lauritzen & Spiegelhalter

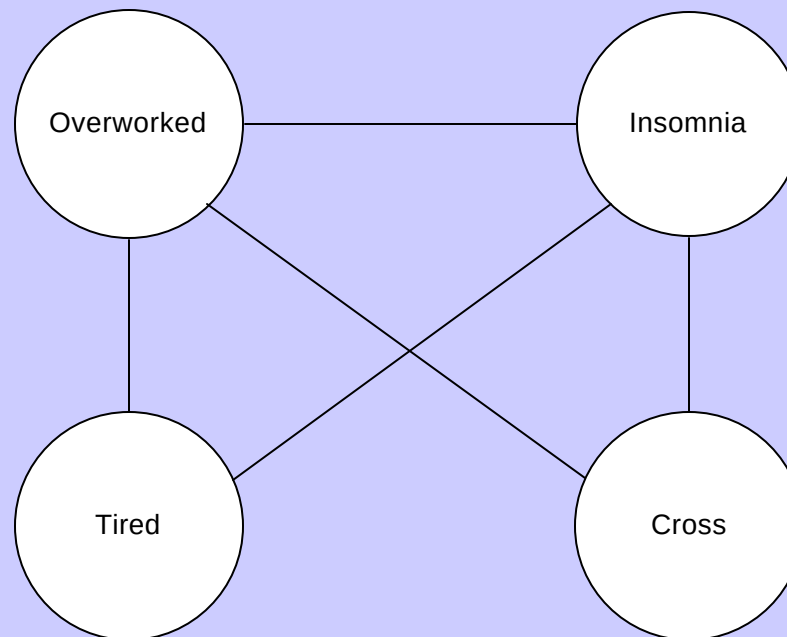
■ Rules for propagation

* Outward pass

- ◆ Each node waits to send its message to its outward neighbors until it has received the message from its inward neighbor
- ◆ When a node is ready to send its message to its outward neighbor, it computes the message by marginalizing its current potential to its intersection with its outward neighbor; it sends this marginal to the outward neighbor
- ◆ When a node receives a message from its inward neighbor, it replaces its own current potential with the product of that potential and the message

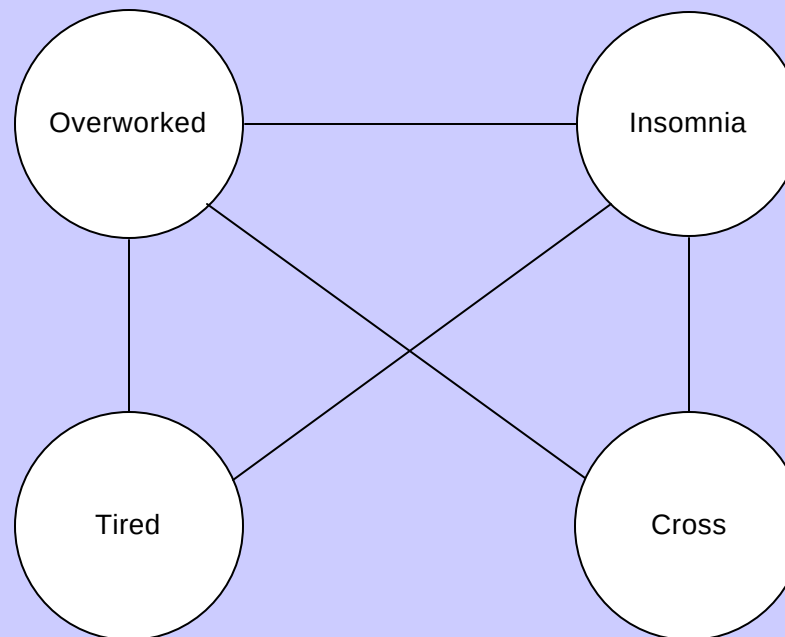
Assignment 4 - Question 2

■ Moral graph



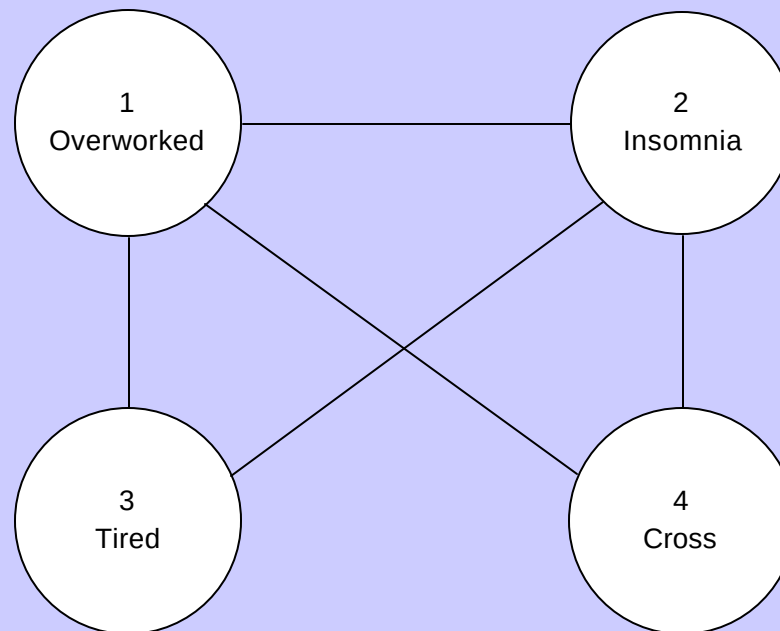
Assignment 4 - Question 2

- This is clearly triangulated . . .



Assignment 4 - Question 2

■ Maximum cardinality



Assignment 4 - Question 2

- No fill-ins needed because

$$bd(4) \cap \{1, 2, 3\} = \{1, 2\}$$

$$bd(3) \cap \{1, 2\} = \{1, 2\}$$

$$bd(2) \cap \{1\} = \{1\}$$

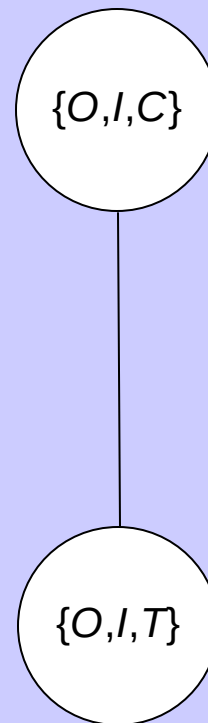
$$bd(1) \cap \{ \} = \{ \}$$

Assignment 4 - Question 2

- Cliques ordered by highest numbered node:
 $\{1, 2, 3\}, \{1, 2, 4\} = \{O, I, T\}, \{O, I, C\}$
- Since there are only two cliques, they vacuously have the running intersection property!

■	i	Clique C_i	Residual R_i	Separator S_i	Parent(s)
	1	$\{O, I, T\}$	$\{O, I, T\}$	$\emptyset$	
	2	$\{O, I, C\}$	$\{C\}$	$\{O, I\}$	1

Assignment 4 - Question 2



- It would be natural to take $\{O,I,T\}$ as root, but since this is a *tree* I can in fact choose any node, and I have chosen $\{O,I,C\}$ as root in the next slide!

Assignment 4 - Question 2

Clique	Configuration	Potentials from conditionals	Inward Pass Set Chain	Outward Pass Clique Marginals	Potentials after absorption	Inward Pass Set Chain	Outward Pass Clique Marginals	Normalized
{O,I,C}	o,i,c	0.00192	0.00192	0.00192	0.00192	0.00188	0.00188	0.01261
	o,i,~c	0.00008	0.00008	0.00008	0.00000	0.00000	0.00000	0.00000
	o,~i,c	0.17820	0.17820	0.17820	0.17820	0.14256	0.14256	0.95577
	o,~i,~c	0.01980	0.01980	0.01980	0.00000	0.00000	0.00000	0.00000
	~o,i,c	0.00480	0.00480	0.00480	0.00480	0.00432	0.00432	0.02896
	~o,i,~c	0.00320	0.00320	0.00320	0.00000	0.00000	0.00000	0.00000
	~o,~i,c	0.00792	0.00792	0.00792	0.00792	0.00040	0.00040	0.00265
	~o,~i,~c	0.78408	0.78408	0.78408	0.00000	0.00000	0.00000	0.00000
{O,I,T}	o,i,t	0.98000	0.98000	0.00196	0.98000	1.00000	0.00188	0.01261
	o,i,~t	0.02000	0.02000	0.00004	0.00000	0.00000	0.00000	0.00000
	o,~i,t	0.80000	0.80000	0.15840	0.80000	1.00000	0.14256	0.95577
	o,~i,~t	0.20000	0.20000	0.03564	0.00000	0.00000	0.00000	0.00000
	~o,i,t	0.90000	0.90000	0.00720	0.90000	1.00000	0.00432	0.02896
	~o,i,~t	0.10000	0.10000	0.00080	0.00000	0.00000	0.00000	0.00000
	~o,~i,t	0.05000	0.05000	0.03960	0.05000	1.00000	0.00040	0.00265
	~o,~i,~t	0.95000	0.95000	0.75240	0.00000	0.00000	0.00000	0.00000
							P(o)	0.96838
							P(~o)	0.03162
							P(i)	0.04158
							P(~i)	0.95842

Aalborg Architecture

■ *Building the Junction Tree:*

- * Construct a tree of cliques with the running intersection property as for the Lauritzen & Spiegelhalter architecture
 - ◆ Note that more efficient techniques for fill-in such as a one-step look-ahead may be used instead of maximum cardinality
 - ◆ Alternatively, a tree of cliques may be constructed by building a Join Tree (Markov Tree) using the Shenoy-Shafer architecture, and then dropping all nodes that are subsets of adjacent nodes
- * Place a separator between each two adjacent cliques containing their intersection
- * This is, of course, a join tree; arbitrarily select a root

Aalborg Architecture

■ Rules for propagation

- * If separators initially contain tables of 1s, then for both the Inward pass and Outward pass:
 - ◆ Each non-root node waits to send a message to a given neighbor until it has received messages from all its other neighbors
 - ◆ The root waits to send messages to its neighbors until it has received messages from them all
 - ◆ When a node is ready to send its message to a particular neighbor, it computes the message by marginalizing its current potential to its intersection with this neighbor, and then sends the message to the separator between it and the neighbor

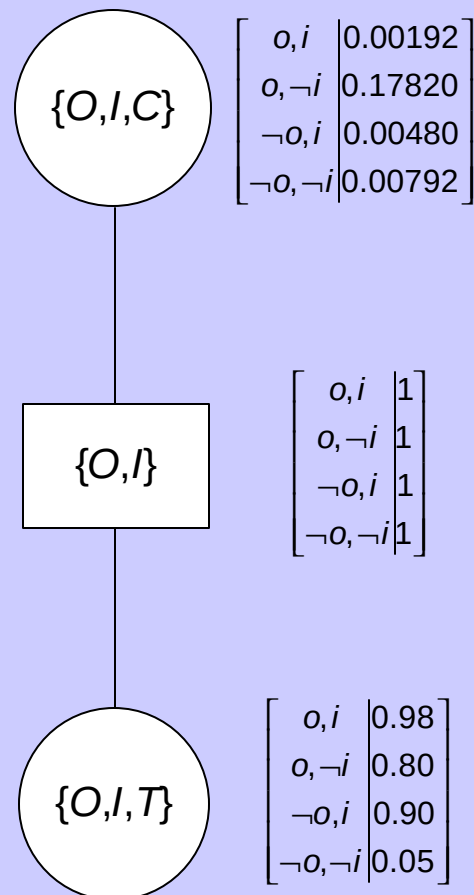
Aalborg Architecture

■ Rules for propagation

- * If separators initially contain tables of 1s, then for both the Inward pass and Outward pass:
 - ◆ When a separator receives a message from one of its two nodes, it divides the message by its current potential, sends the quotient on to the other node, and replaces its current potential with the new message
 - ◆ When a node receives a message, it replaces its current potential with the product of that potential and the message

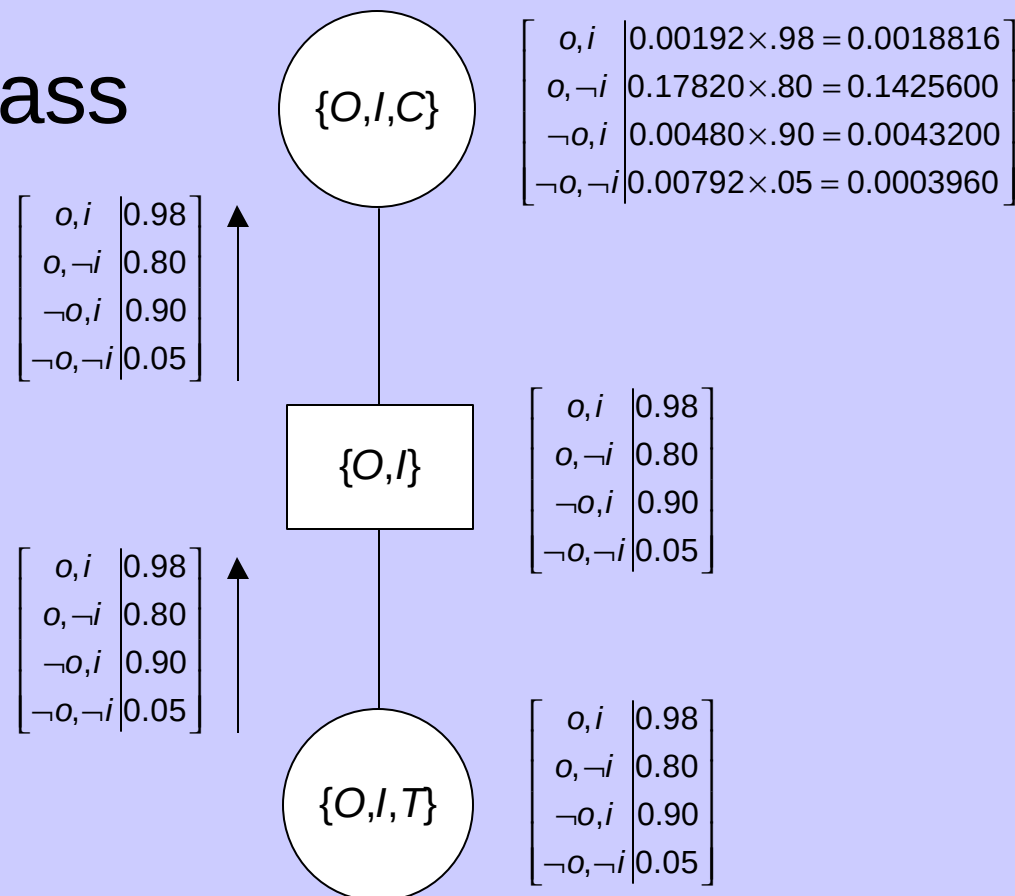
Assignment 4 - Question 3

■ Junction Tree



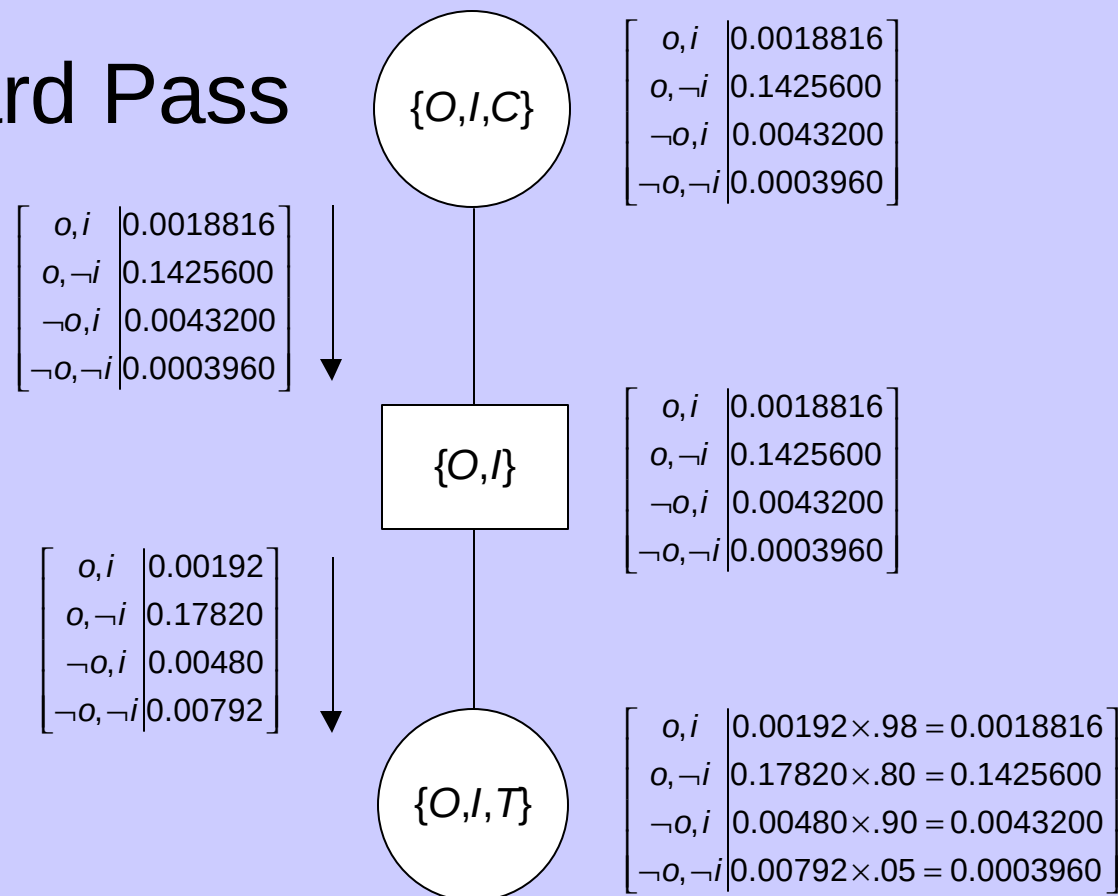
Assignment 4 - Question 3

■ Inward Pass



Assignment 4 - Question 3

■ Outward Pass



Assignment 4 - Question 3

■ Hence, after normalization:

$$P(o) = \frac{(0.0018816 + 0.1425600)}{(0.0018816 + 0.1425600 + 0.0043200 + 0.0003960)} = 0.96838$$

and:

$$P(i) = \frac{(0.0018816 + 0.0043200)}{(0.0018816 + 0.1425600 + 0.0043200 + 0.0003960)} = 0.04158$$

Shenoy-Shafer Architecture

■ *Building the Join Tree:*

- * Detailed algorithms were provided in writing last week for:
 - ◆ Shenoy-Shafer Join Trees
 - ◆ Binary Join Trees

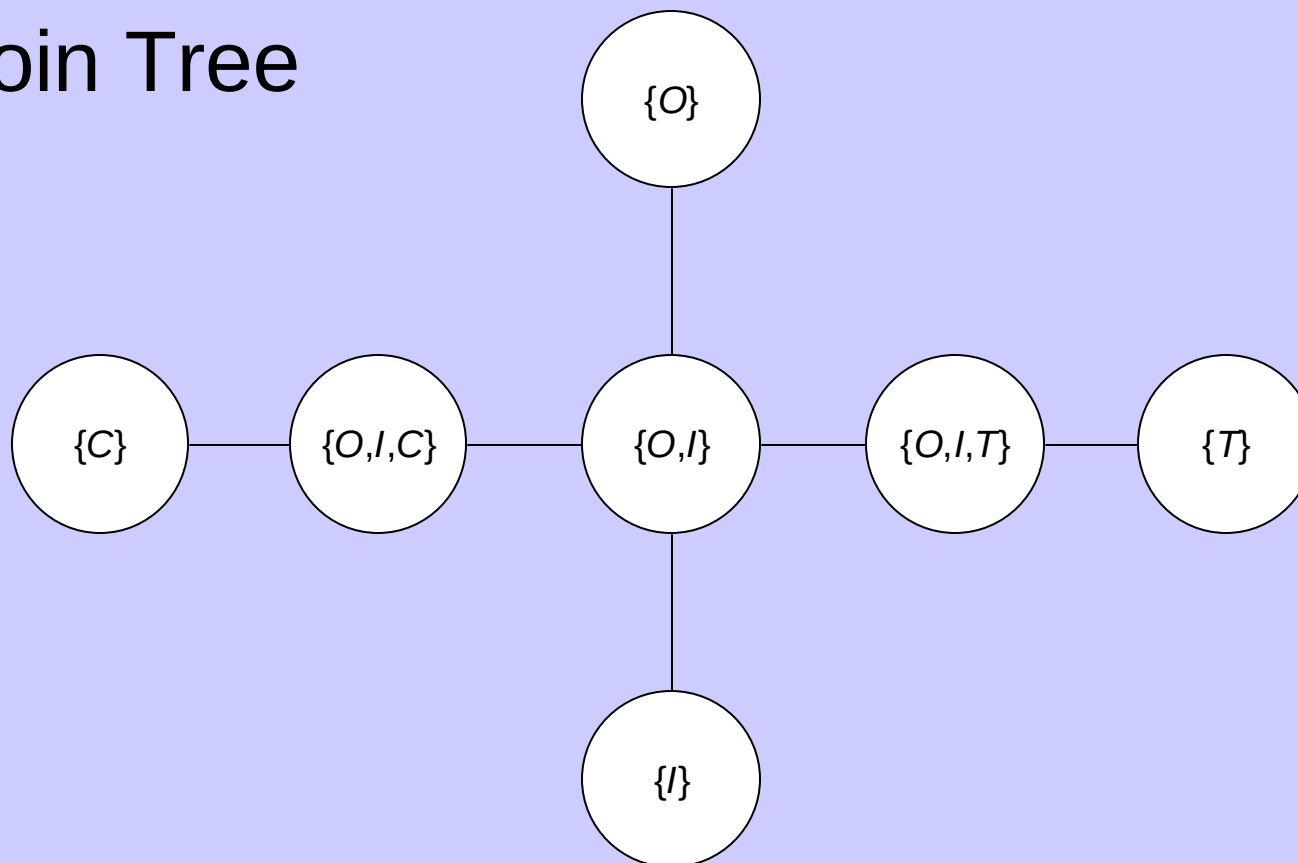
Shenoy-Shafer Architecture

■ Rules for propagation

- * There is no distinction between inward and outward passes
 - ◆ Each node waits to send its message to a given neighbor until it has received messages from all its other neighbors
 - ◆ When a node is ready to send its message to a particular neighbor, it computes the message by collecting all its messages from other neighbors, multiplying its own potential by these messages, and marginalizing the product to its intersection with the neighbor to which it is sending

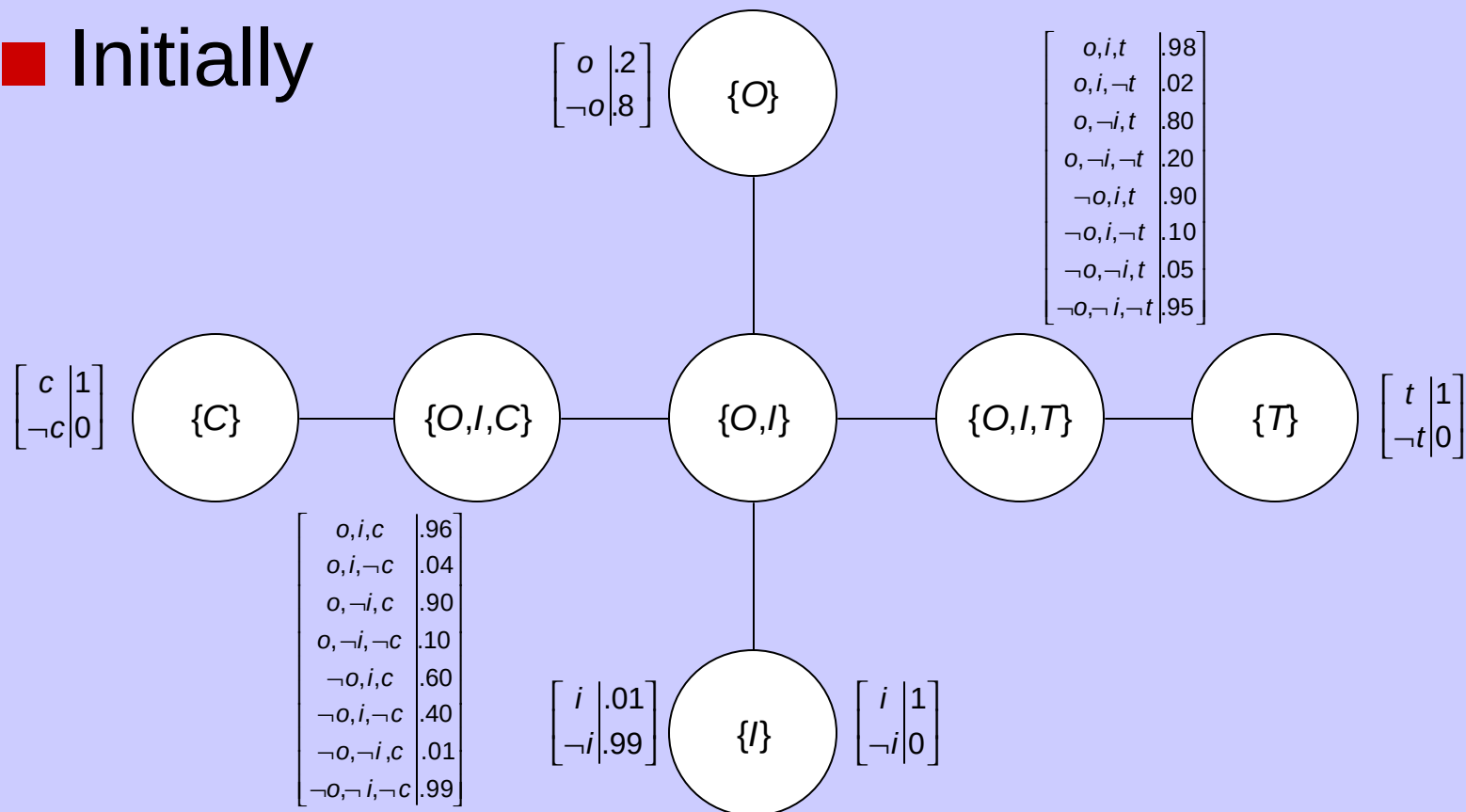
Assignment 4 - Question 4

■ Join Tree



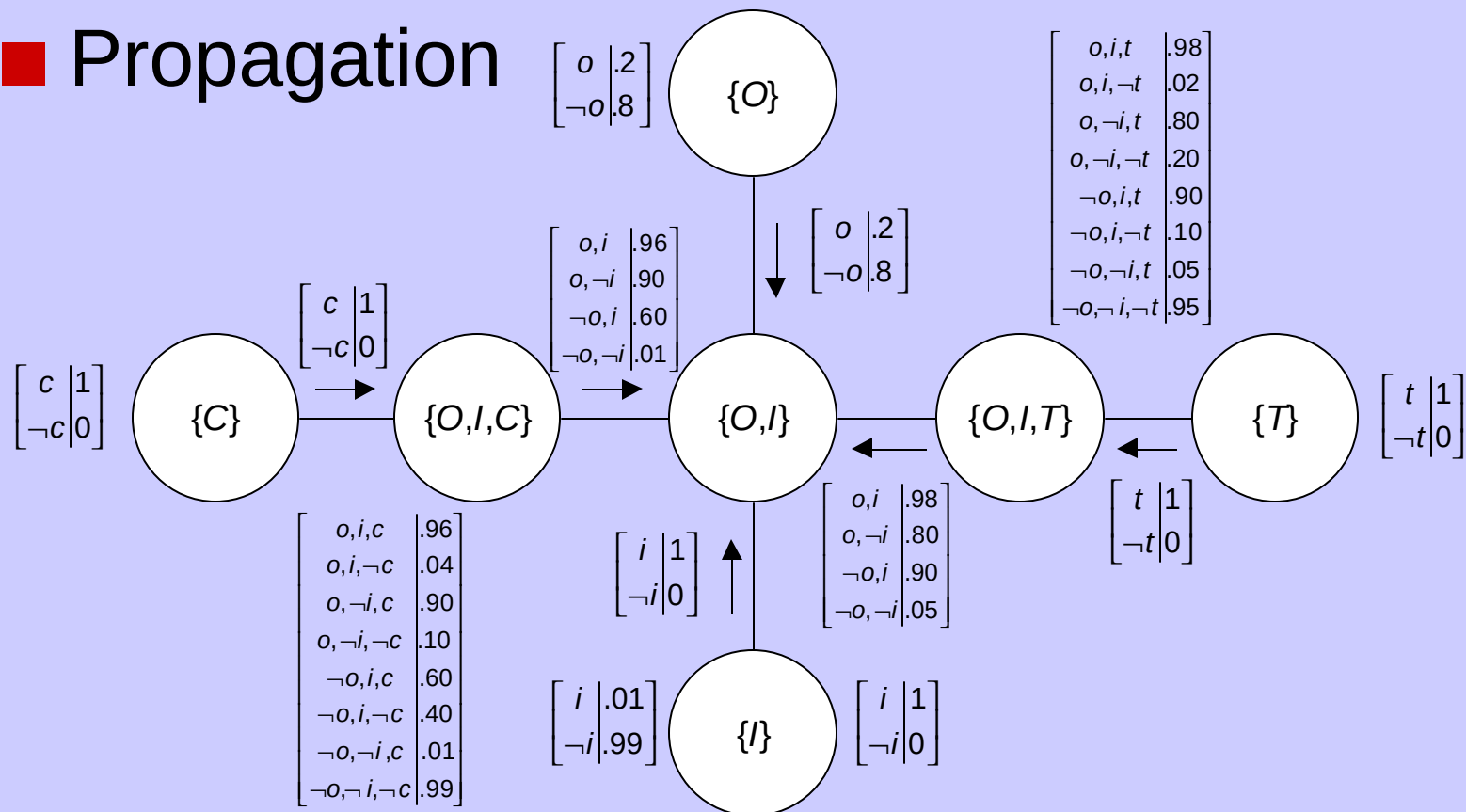
Assignment 4 - Question 4

Initially



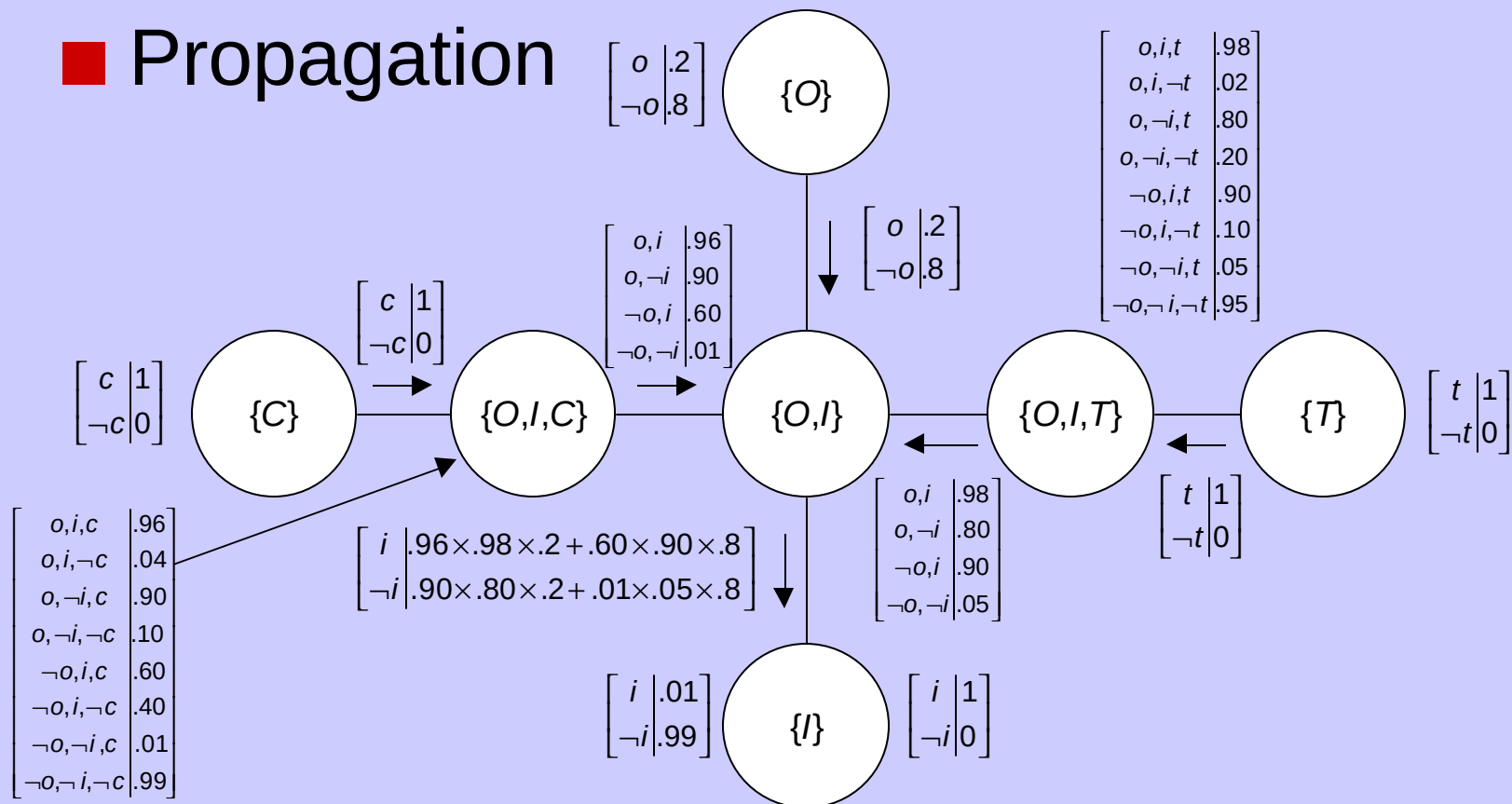
Assignment 4 - Question 4

■ Propagation



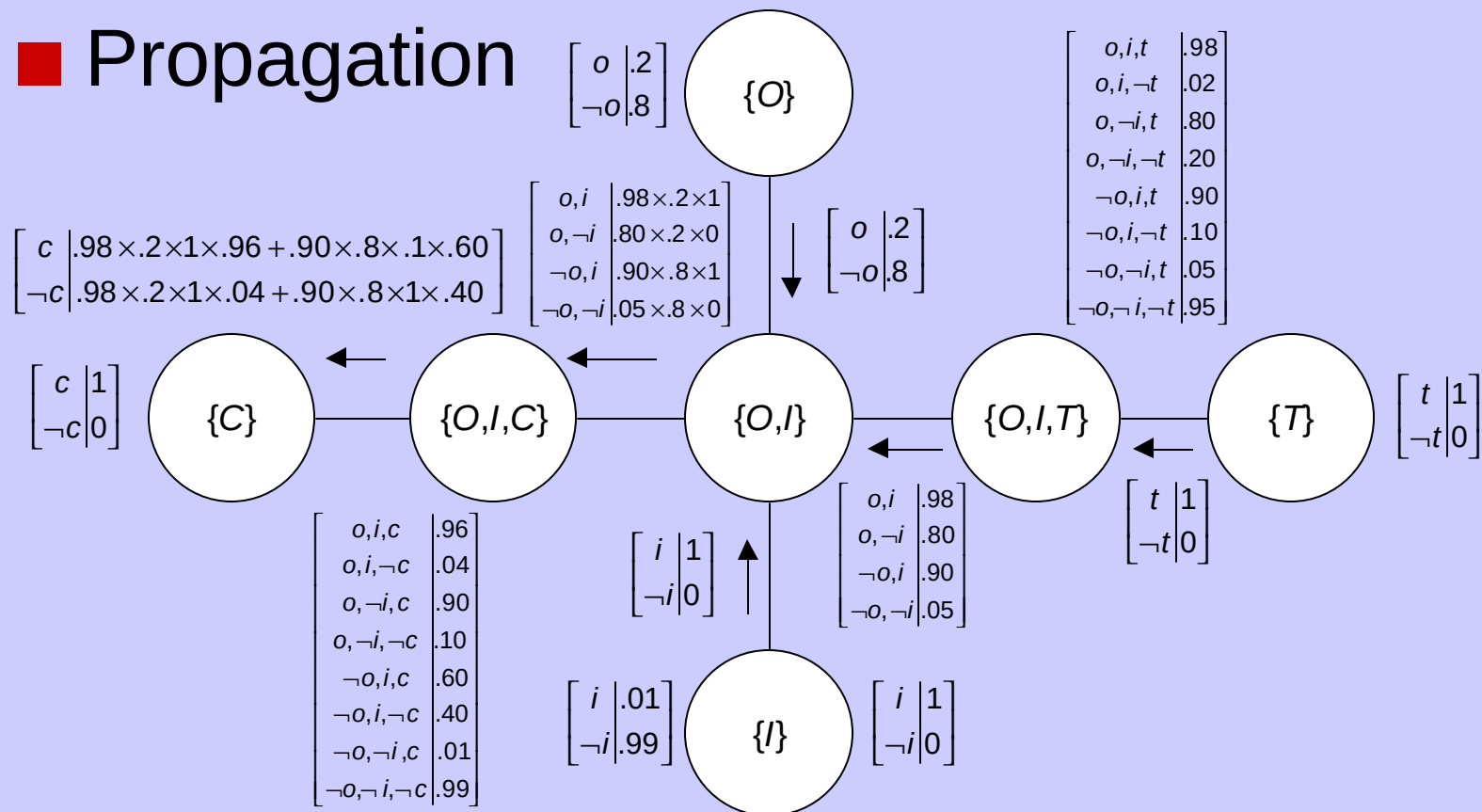
Assignment 4 - Question 4

■ Propagation



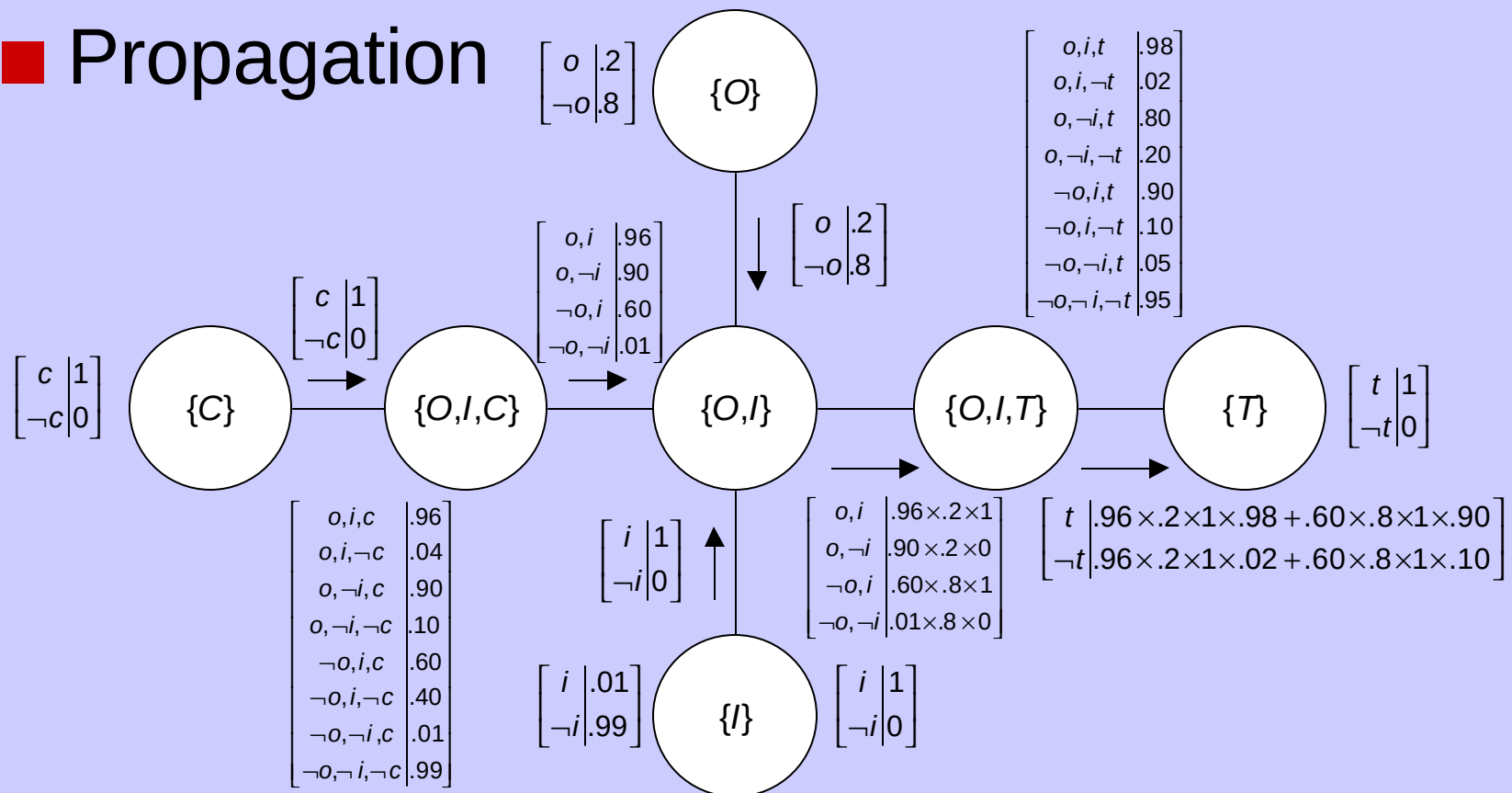
Assignment 4 - Question 4

■ Propagation



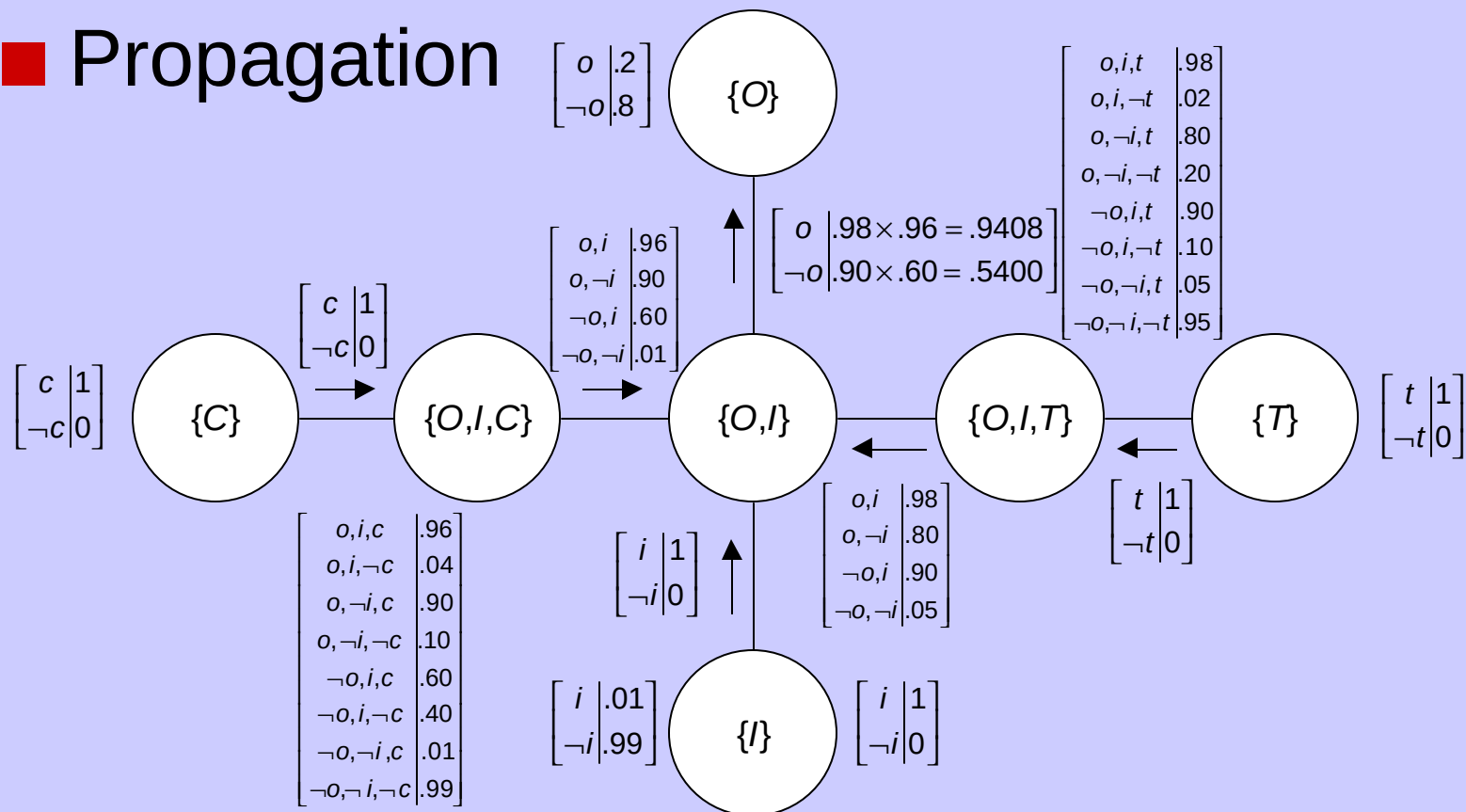
Assignment 4 - Question 4

■ Propagation



Assignment 4 - Question 4

■ Propagation



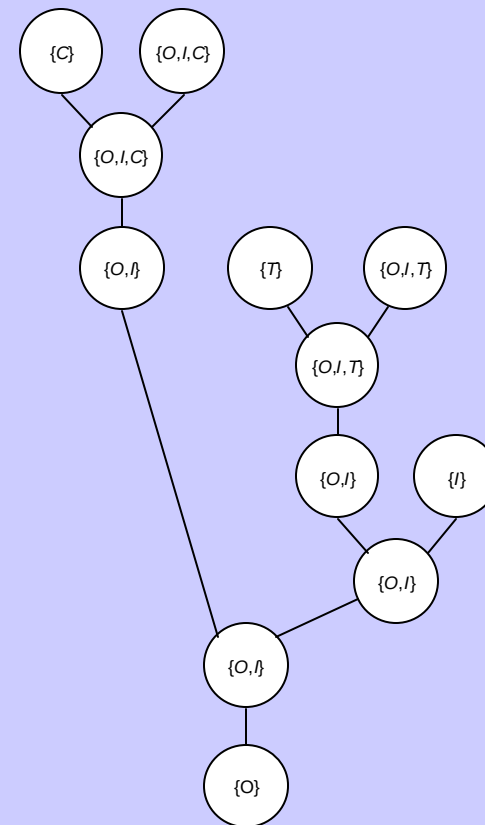
Assignment 4 - Question 4

- Hence the posterior for O is given by

$$\begin{bmatrix} .2 \\ .8 \end{bmatrix} \cdot \begin{bmatrix} .9408 \\ .5400 \end{bmatrix} = \begin{bmatrix} .18816 \\ .43200 \end{bmatrix} \Rightarrow \begin{bmatrix} o & .30341 \\ \sim o & .69659 \end{bmatrix}$$

Assignment 4 - Question 4

■ Binary Join Tree (1)



Assignment 4 - Question 4

■ Binary Join Tree (2)

