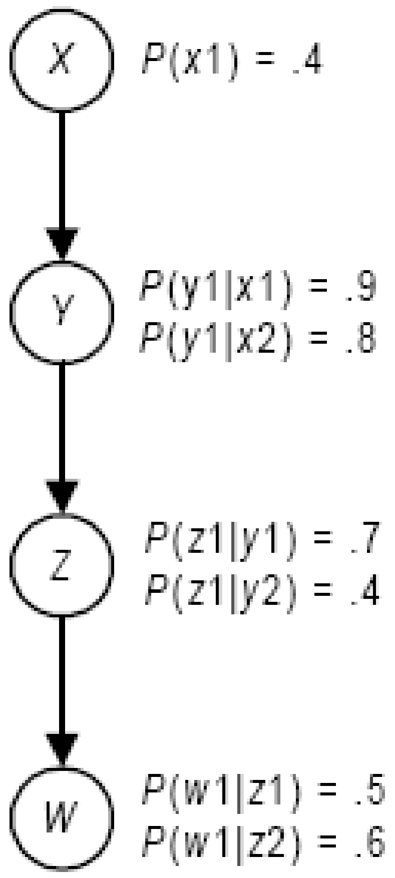
PROBABILISTIC REASONING

Unit # 08-1: Inference in Bayesian Networks

ACKNOWLEDGEMENTS

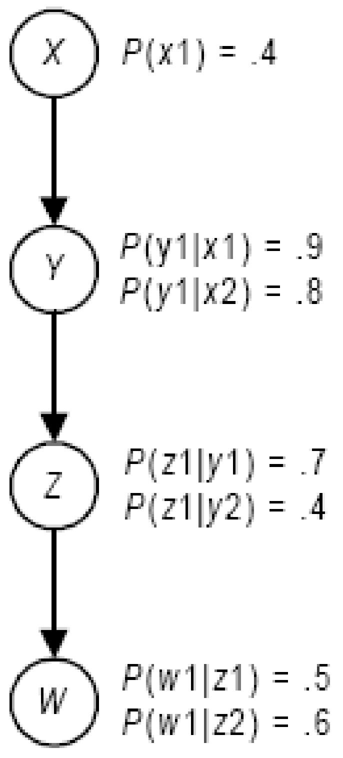
The material in this presentation is taken from Neapolitan’s book “Learning Bayesian Networks” (Chapter 3).

INFERENCE IN SINGLY CONNECTED NETWORKS

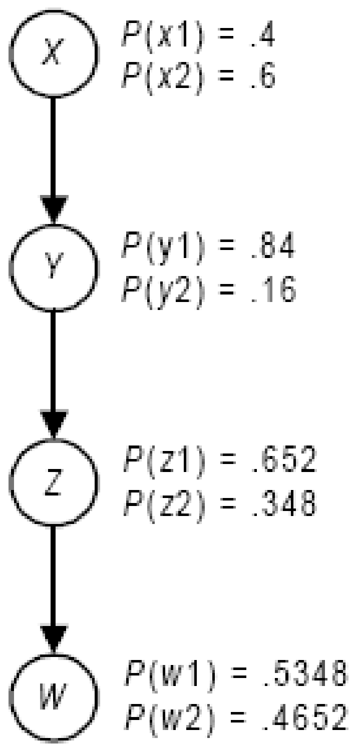


(a)

INFERENCE IN SINGLY CONNECTED NETWORKS



(a)



(b)

$$P(y1) = P(y1 \mid x1)\,P(x1) + P(y1 \mid x2)\,P(x2)$$

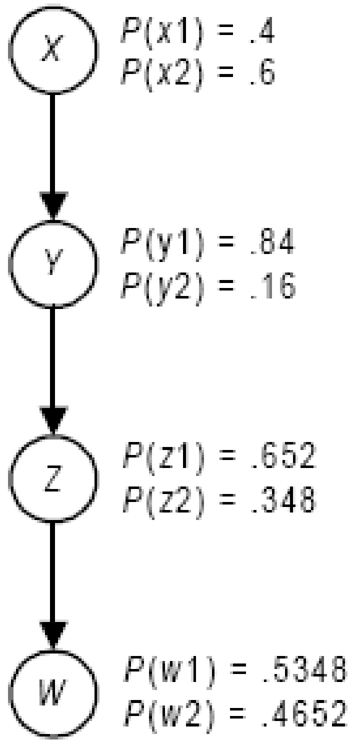
$$P(y2) = 1 - P(y1)$$

$$P(z1) = P(z1 \mid y1)\,P(y1) + P(z1 \mid y2)\,P(y2)$$

$$P(z2) = 1 - P(z1)$$

P(w1) = ???

INFERENCE IN SINGLY CONNECTED NETWORKS



Suppose we get evidence that w1 is true, i.e., P(w1) = 1.

Now compute the posterior probabilities:
P*(z1), P*(y1), P*(x1)

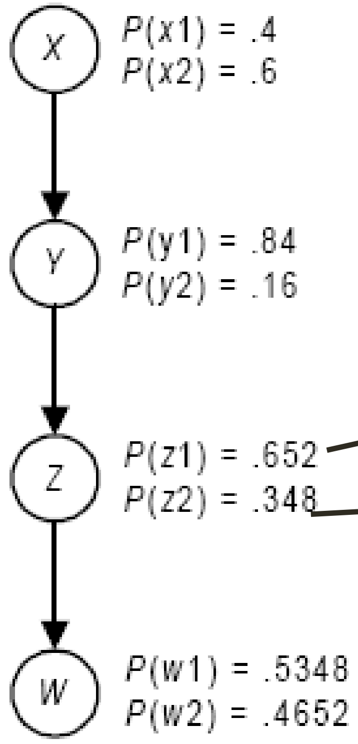
$$P^*(z1) = P(z1 \mid w1)\, P(w1) \longleftarrow P(w1)=1$$
$$\quad + P(z1 \mid w2)\, P(w2) \longleftarrow P(w2)=0$$

Computing P(z1 | w1) using Bayes theorem:
$P(z1 \mid w1) = P(w1 \mid z1)\, P(z1) / P(w1)$
$P(z1 \mid w1) = 0.5 \times 0.652 / 0.5348 = 0.61$

=>
$P^*(z1) = 0.61 * 1 + 0 = 0.61$

Computation of P*(y1) on the next slide

INFERENCE IN SINGLY CONNECTED NETWORKS



Now update the probability of Y.

$P^*(y1) = P(y1 \mid z1)\,P(z1)$
$\quad + P(y1 \mid z2)\,P(z2)$

$P(z1)=0.61$
$P(z2)=0.39$ } Computed on the previous slide

Computing $P(y1 \mid z1)$ and $P(y1 \mid z2)$ using Bayes theorem:

$P(y1 \mid z1) = P(z1 \mid y1)\,P(y1) / P(z1)$

$P(y1 \mid z1) = 0.7 \times 0.84 / 0.652 = 0.90$

$P(y1 \mid z2) = P(z2 \mid y1)\,P(y1) / P(z2)$

$P(y1 \mid z2) = 0.3 \times 0.84 / 0.348 = 0.72$

Finally plug the values in the equation at the top

$P^*(y1) = P(y1 \mid z1)\,P(z1) + P(y1 \mid z2)\,P(z2)$
$\quad = 0.90 \times 0.61 + 0.72 \times 0.39$
$\quad = 0.83$

Confirm the results using Genie.

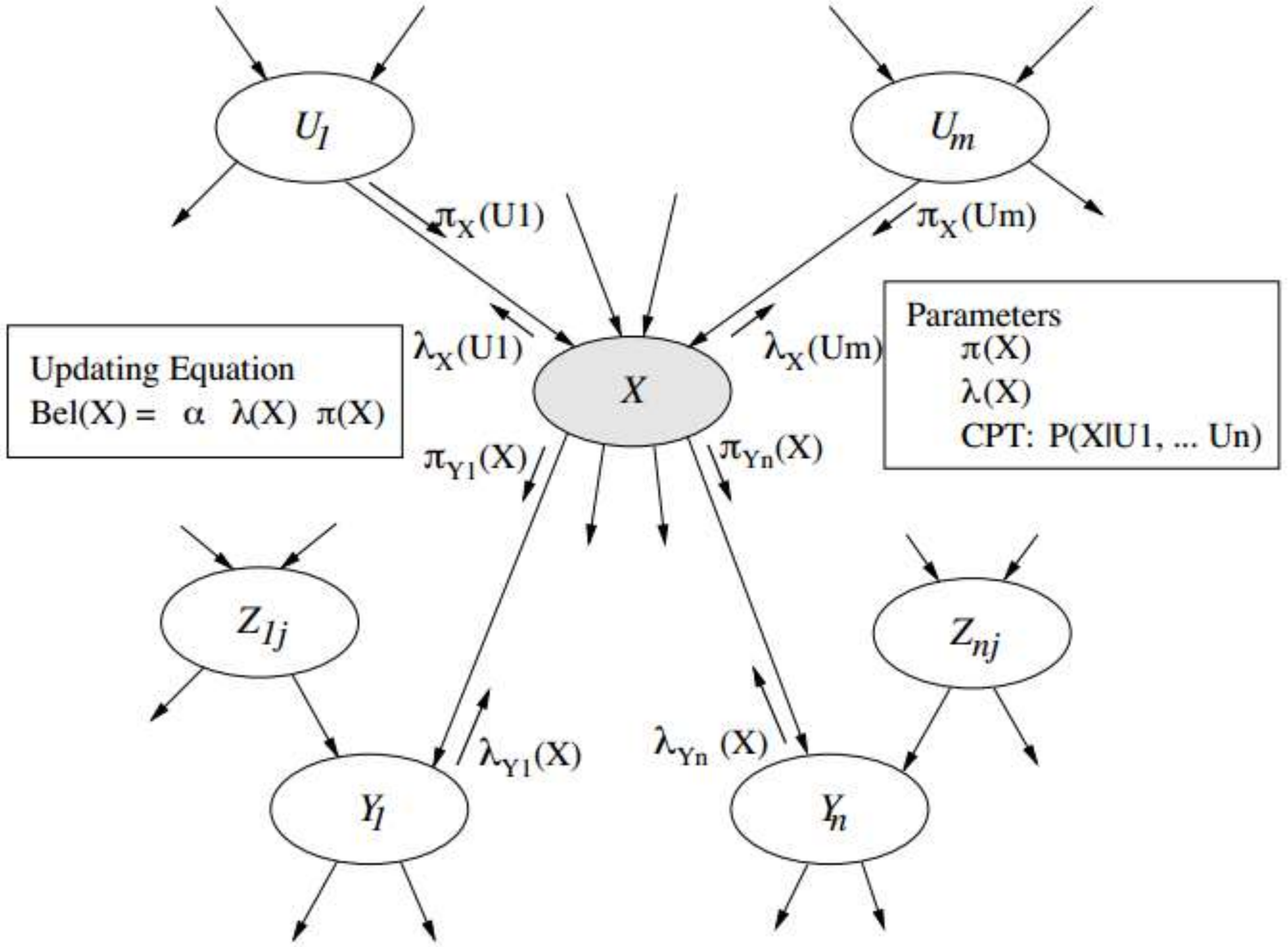
Finally compute $P^*(x1)$

INFERENCE IN SINGLY CONNECT NETWORKS

Singly connected networks (also called polytrees) have at most one path between any pair of nodes.

The figure on the next slide shows a diagram of a node X in a SCN, with all its connections to parents (the Ui), children (the Yj), and the children's other parents (the Zij).

INFERENCE IN SINGLY CONNECT NETWORKS (CONT'D)



PREDICTIVE AND DIAGNOSTIC SUPPORT

An evidence can be divided into two components:

The predictive support for X , from evidence nodes connected to X through its parents, U1,….Um; and

The diagnostic support for X , from evidence nodes connected to X through its children Y1,….Yn.

KIM AND PEARL'S MESSAGE PASSING ALGORITHM

- *The current strength of the predictive support π contributed by each incoming link $U_i \rightarrow X$:*

  $\pi_X(U_i) = P(U_i | E_{U_i \backslash X})$

  where $E_{U_i \backslash X}$ is all evidence connected to U_i except via X.
- *The current strength of the diagnostic support λ contributed by each outgoing link $X \rightarrow Y_j$:*

  $\lambda_{Y_j}(X) = P(E_{Y_j \backslash X} | X)$

  where $E_{Y_j \backslash X}$ is all evidence connected to Y_j through its parents except via X.
- *The fixed CPT $P(X|U_i, \ldots, U_n)$ (relating X to its immediate parents).*

These parameters are used to do local belief updating in the following three steps, which can be done in any order.

INITIALIZATION

The algorithm requires the following initializations (i.e., before any evidence is entered).

- Set all λ values, λ messages and π messages to 1.
- Root nodes: If node W has no parents, set $\pi(W)$ to the prior, $P(W)$.

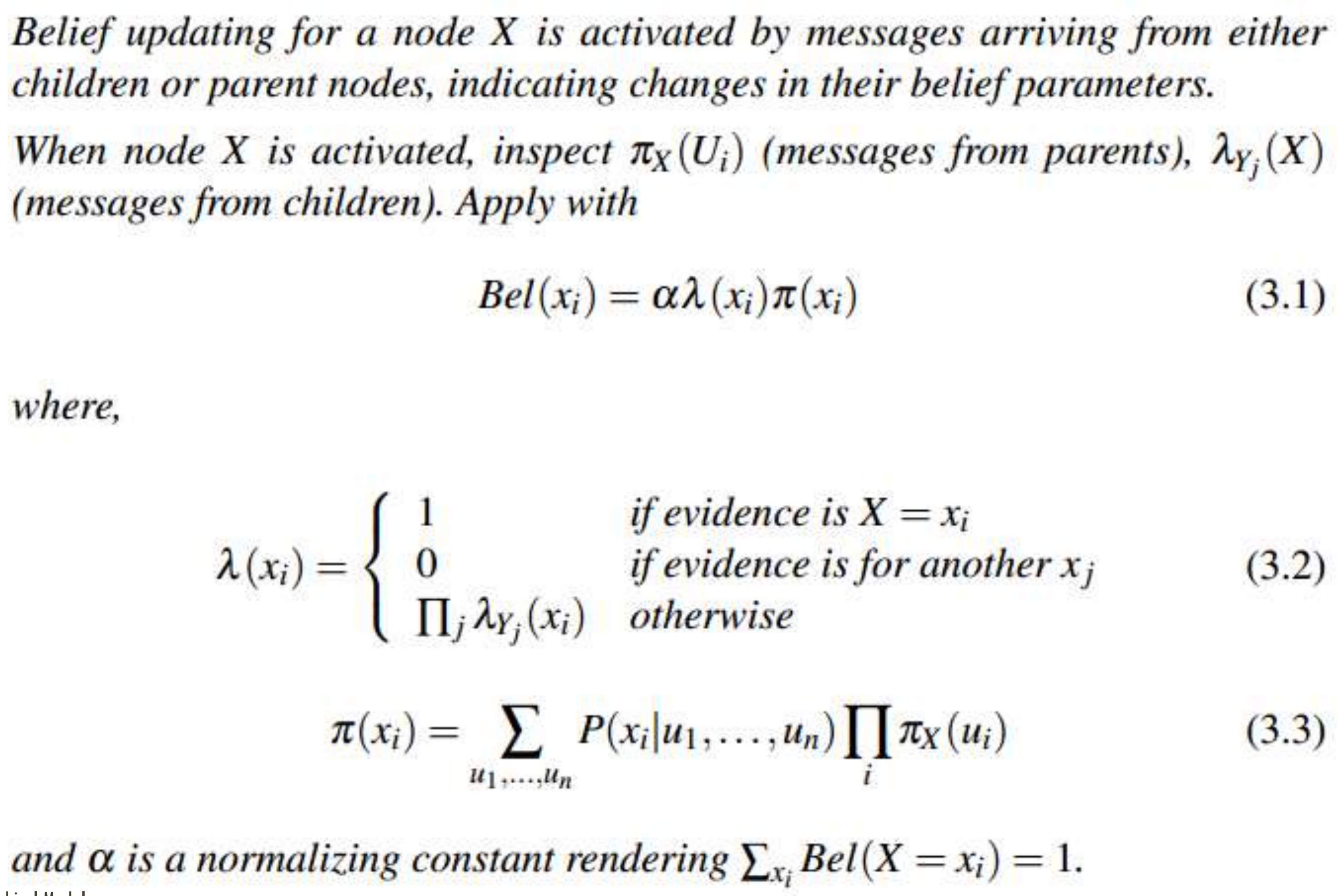
PROPAGATION FLOW

The format for both types of messages is π_{Child}(Parent) and λ_{Child}(Parent).

- So, π messages are sent in the direction of the arc, from parent to child, hence the notation is $\pi_{Receiver}$(Sender);
- λ messages are sent from child to parent, against the direction of the arc, hence the notation is λ_{Sender} (Receiver).

π plays the role of prior and λ the likelihood in Bayes' Theorem.

STEP 1: BELIEF UPDATING



Belief updating for a node X is activated by messages arriving from either children or parent nodes, indicating changes in their belief parameters.

When node X is activated, inspect $\pi_X(U_i)$ (messages from parents), $\lambda_{Y_j}(X)$ (messages from children). Apply with

$$Bel(x_i) = \alpha\lambda(x_i)\pi(x_i) \tag{3.1}$$

where,

$$\lambda(x_i) = \begin{cases} 1 & \text{if evidence is } X = x_i \\ 0 & \text{if evidence is for another } x_j \\ \prod_j \lambda_{Y_j}(x_i) & \text{otherwise} \end{cases} \tag{3.2}$$

$$\pi(x_i) = \sum_{u_1,\ldots,u_n} P(x_i|u_1,\ldots,u_n)\prod_i \pi_X(u_i) \tag{3.3}$$

and α is a normalizing constant rendering $\sum_{x_i} Bel(X = x_i) = 1$.

STEP 2: BOTTOM-UP PROPAGATION

Node X computes new λ messages to send to its parents.

$$\lambda_X(u_i) = \sum_{x_i} \lambda(x_i) \sum_{u_k : k \neq i} P(x_i | u_1, \ldots, u_n) \prod_{k \neq i} \pi_X(u_k) \qquad (3.4)$$

The λ message to one parent combines

- (i) information that has come from children via λ messages and been summarized in the λ (X) parameter,
- (ii) the values in the CPT and
- (iii) any π messages that have been received from any other parents.

STEP 3: TOP-DOWN PROPAGATION

Node X computes new π messages to send to its children.

$$\pi_{Y_j}(x_i) = \begin{cases} 1 & \textit{if evidence value } x_i \textit{ is entered} \\ 0 & \textit{if evidence is for another value } x_j \\ \alpha[\prod_{k\neq j}\lambda_{Y_k}(x_i)]\sum_{u_1,\ldots,u_n} P(x_i|u_1,\ldots,u_n)\prod_i \pi_X(u_i) & \\ = \frac{\alpha Bel(x_i)}{\lambda_{Y_j}(x_i)} & \end{cases} \qquad (3.5)$$

The $\pi_{Y_j}(x_i)$ message down to child Y_j is 1 if x_i is the evidence value and 0 if the evidence is for some other value x_j .

If no evidence is entered for X , then it combines

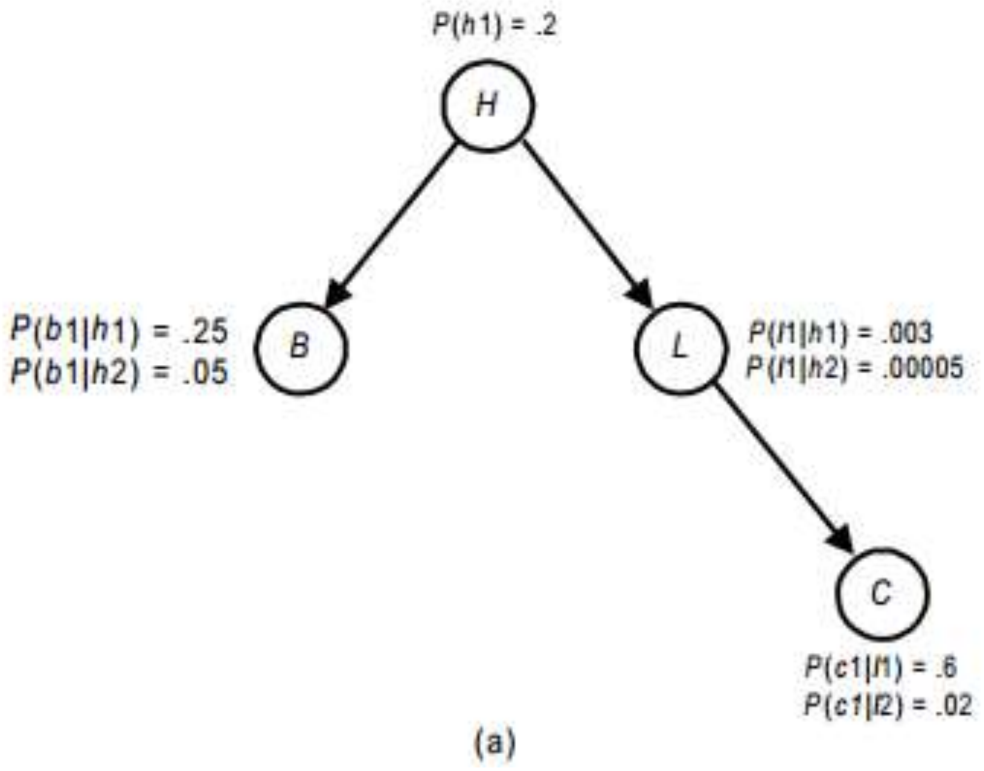
- (i) information from children other than Y_j
- (ii) the CPT and
- (iii) the π messages it has received from its parents.

INITIALIZATION

The algorithm requires the following initializations (i.e., before any evidence is entered).

- Set all λ values, λ messages and π messages to 1.
- Root nodes: If node W has no parents, set π(W) to the prior, P(W).

EXAMPLE NETWORK (NEAPOLITAN 2003)



```
$\lambda(h1) = 1; \lambda(h2) = 1;$          // Compute $\lambda$ values.
$\lambda(b1) = 1; \lambda(b2) = 1;$
$\lambda(l1) = 1; \lambda(l2) = 1;$
$\lambda(c1) = 1; \lambda(c2) = 1;$

$\lambda_B(h1) = 1; \lambda_B(h2) = 1;$      // Compute $\lambda$ messages.
$\lambda_L(h1) = 1; \lambda_L(h2) = 1;$
$\lambda_C(l1) = 1; \lambda_C(l2) = 1;$

$P(h1|\varnothing) = P(h1) = .2;$          // Compute $P(h|\varnothing)$.
$P(h2|\varnothing) = P(h2) = .8;$

$\pi(h1) = P(h1) = .2;$                // Compute $H$'s $\pi$ values.
$\pi(h2) = P(h2) = .8;$

send_$\pi$_msg(H, B);
send_$\pi$_msg(H, L);
```

MARGINAL OF B

The call

$send\_\pi\_msg(H, B);$

results in the following steps:

$$\pi_B(h1) = \pi(h1)\lambda_L(h1) = (.2)(1) = .2; \qquad // \textit{ H sends B a } \pi \textit{ message.}$$
$$\pi_B(h2) = \pi(h2)\lambda_L(h2) = (.8)(1) = .8;$$

$$\begin{aligned}\pi(b1) &= P(b1|h1)\pi_B(h1) + P(b1|h2)\pi_B(h2); \qquad // \textit{ Compute B's } \pi \textit{ values.}\\ &= (.25)(.2) + (.05)(.8) = .09;\end{aligned}$$

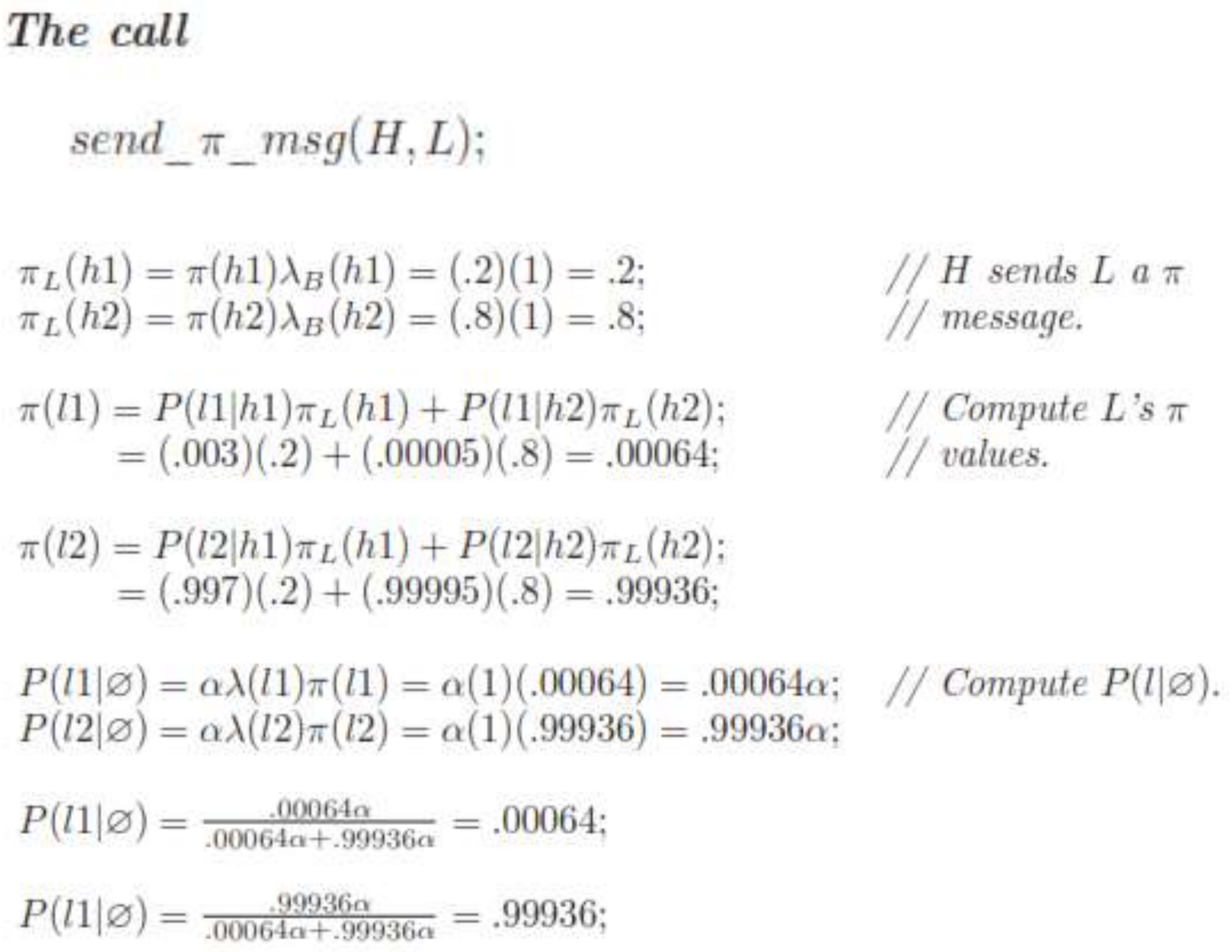
$$\begin{aligned}\pi(b2) &= P(b2|h1)\pi_B(h1) + P(b2|h2)\pi_B(h2);\\ &= (.75)(.2) + (.95)(.8) = .91;\end{aligned}$$

$$P(b1|\varnothing) = \alpha\lambda(b1)\pi(b1) = \alpha(1)(.09) = .09\alpha; \qquad // \textit{ Compute } P(b|\varnothing).$$
$$P(b2|\varnothing) = \alpha\lambda(b2)\pi(b2) = \alpha(1)(.91) = .91\alpha;$$

$$P(b1|\varnothing) = \frac{.09\alpha}{.09\alpha + .91\alpha} = .09;$$

$$P(b1|\varnothing) = \frac{.91\alpha}{.09\alpha + .91\alpha} = .91;$$

MARGINAL OF L



The call

$$send\_\pi\_msg(H, L);$$

$$\begin{aligned}
\pi_L(h1) &= \pi(h1)\lambda_B(h1) = (.2)(1) = .2; && \textit{// H sends L a } \pi \\
\pi_L(h2) &= \pi(h2)\lambda_B(h2) = (.8)(1) = .8; && \textit{// message.}
\end{aligned}$$

$$\begin{aligned}
\pi(l1) &= P(l1|h1)\pi_L(h1) + P(l1|h2)\pi_L(h2); && \textit{// Compute L's } \pi \\
&= (.003)(.2) + (.00005)(.8) = .00064; && \textit{// values.}
\end{aligned}$$

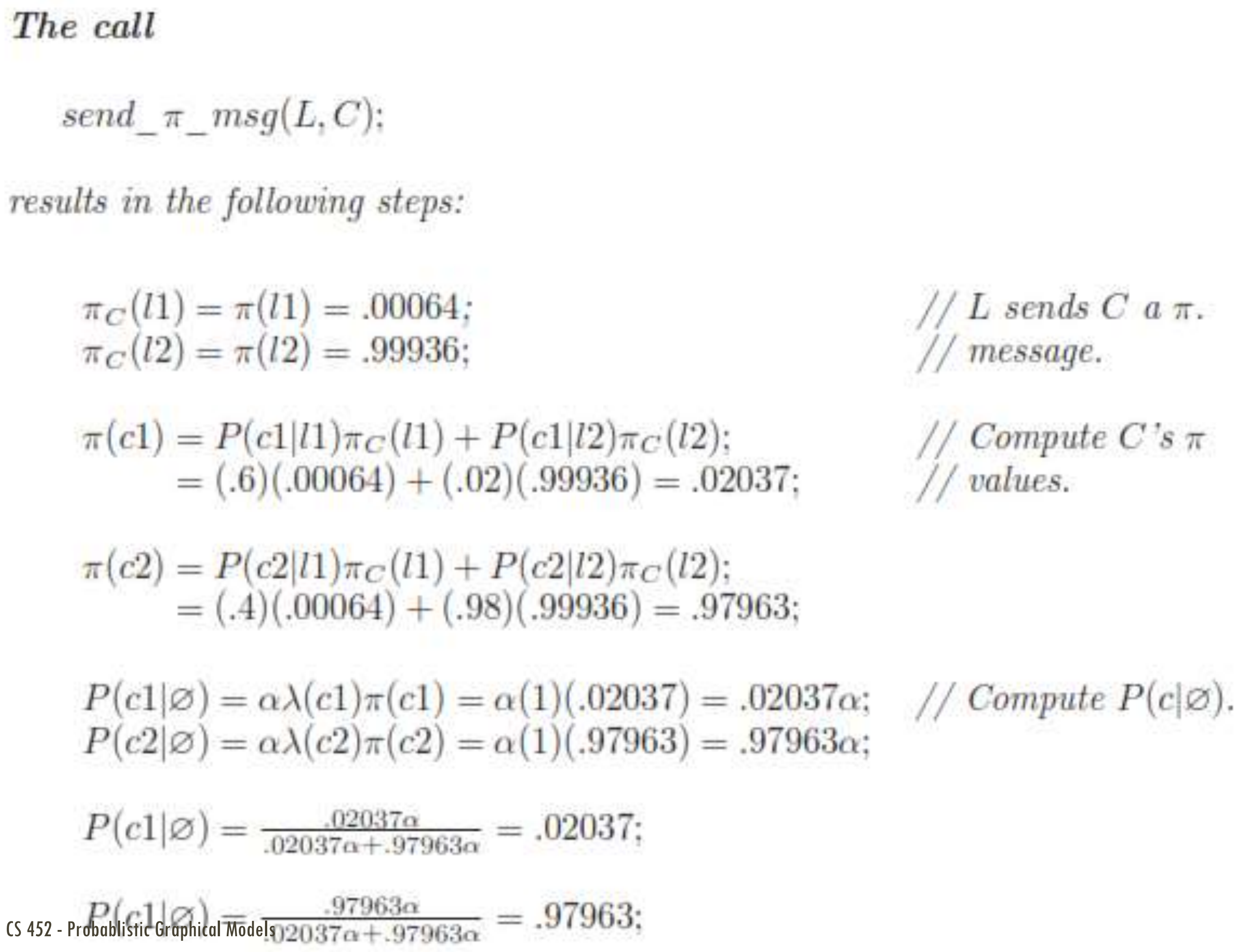
$$\begin{aligned}
\pi(l2) &= P(l2|h1)\pi_L(h1) + P(l2|h2)\pi_L(h2); \\
&= (.997)(.2) + (.99995)(.8) = .99936;
\end{aligned}$$

$$\begin{aligned}
P(l1|\varnothing) &= \alpha\lambda(l1)\pi(l1) = \alpha(1)(.00064) = .00064\alpha; && \textit{// Compute } P(l|\varnothing). \\
P(l2|\varnothing) &= \alpha\lambda(l2)\pi(l2) = \alpha(1)(.99936) = .99936\alpha;
\end{aligned}$$

$$P(l1|\varnothing) = \frac{.00064\alpha}{.00064\alpha + .99936\alpha} = .00064;$$

$$P(l1|\varnothing) = \frac{.99936\alpha}{.00064\alpha + .99936\alpha} = .99936;$$

MARGINAL OF C



The call

$send\_\pi\_msg(L, C);$

results in the following steps:

$\pi_C(l1) = \pi(l1) = .00064;$ // *L sends C a* π.
$\pi_C(l2) = \pi(l2) = .99936;$ // *message.*

$\pi(c1) = P(c1|l1)\pi_C(l1) + P(c1|l2)\pi_C(l2);$ // *Compute C's* π
$\quad = (.6)(.00064) + (.02)(.99936) = .02037;$ // *values.*

$\pi(c2) = P(c2|l1)\pi_C(l1) + P(c2|l2)\pi_C(l2);$
$\quad = (.4)(.00064) + (.98)(.99936) = .97963;$

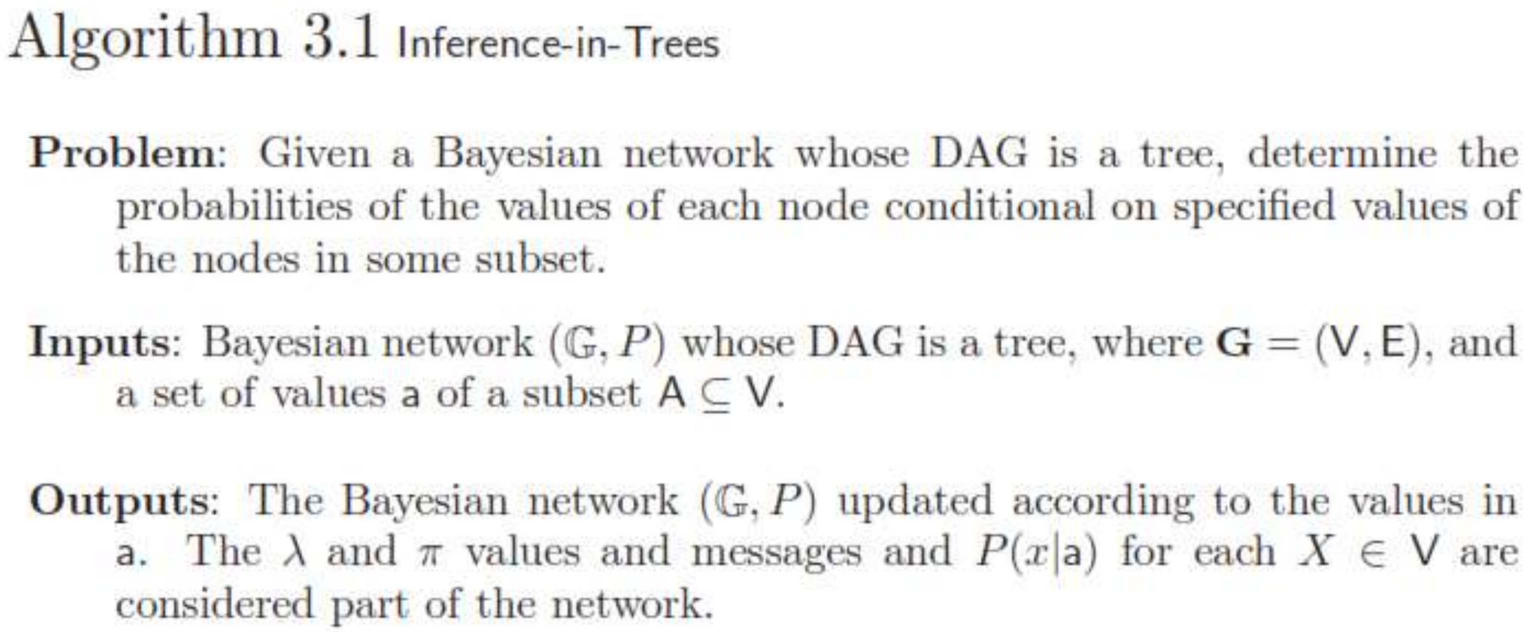
$P(c1|\varnothing) = \alpha\lambda(c1)\pi(c1) = \alpha(1)(.02037) = .02037\alpha;$ // *Compute* $P(c|\varnothing)$.
$P(c2|\varnothing) = \alpha\lambda(c2)\pi(c2) = \alpha(1)(.97963) = .97963\alpha;$

$P(c1|\varnothing) = \frac{.02037\alpha}{.02037\alpha + .97963\alpha} = .02037;$

$P(c1|\varnothing) = \frac{.97963\alpha}{.02037\alpha + .97963\alpha} = .97963;$

ALGORITHM

KIM AND PEARL MESSAGE PASSING ALGORITHM



Algorithm 3.1 Inference-in-Trees

Problem: Given a Bayesian network whose DAG is a tree, determine the probabilities of the values of each node conditional on specified values of the nodes in some subset.

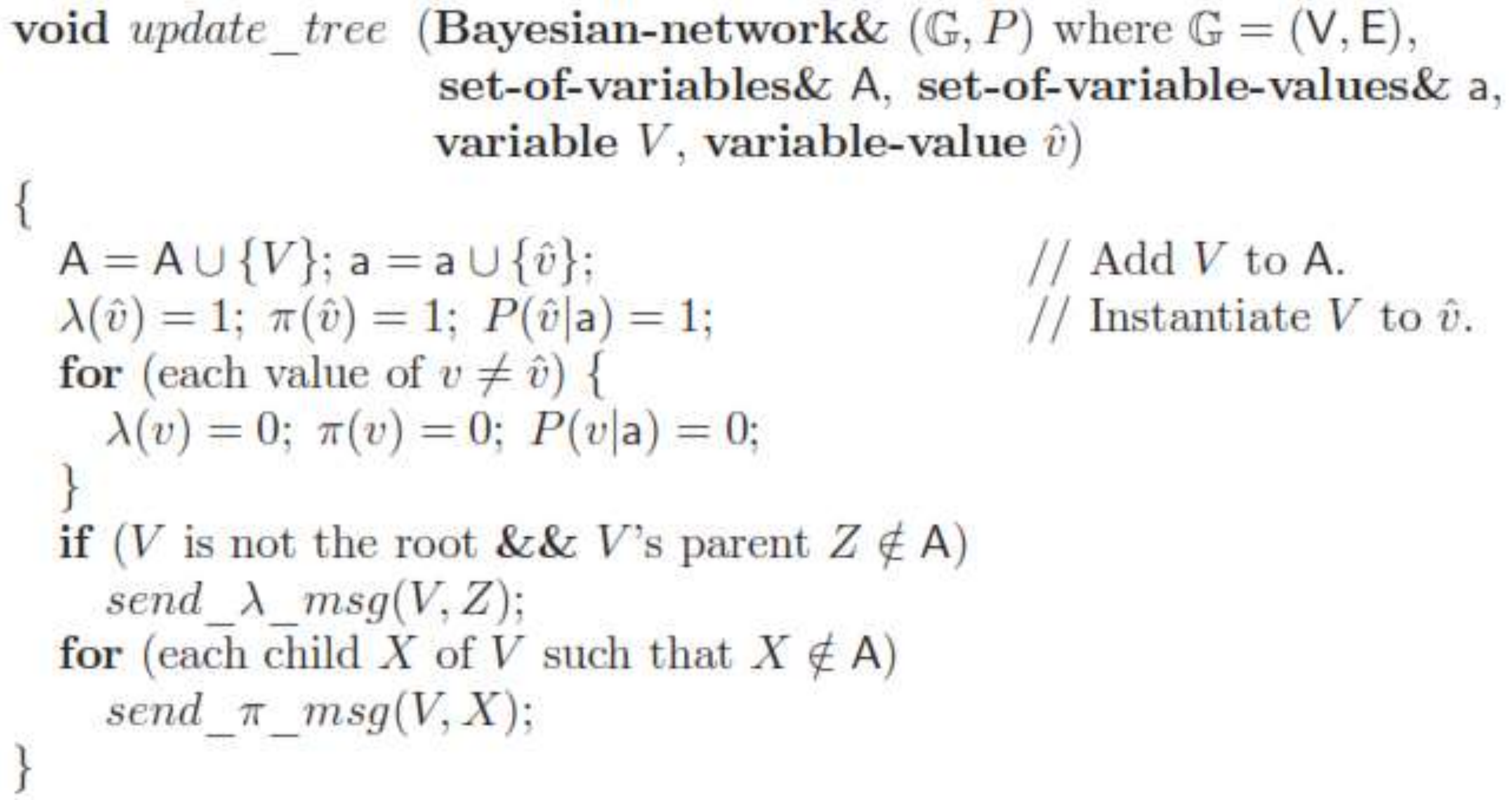
Inputs: Bayesian network $(\mathbb{G}, P)$ whose DAG is a tree, where $\mathbf{G} = (\mathsf{V}, \mathsf{E})$, and a set of values a of a subset $\mathsf{A} \subseteq \mathsf{V}$.

Outputs: The Bayesian network $(\mathbb{G}, P)$ updated according to the values in a. The λ and π values and messages and $P(x|\mathsf{a})$ for each $X \in \mathsf{V}$ are considered part of the network.

KIM AND PEARL MESSAGE PASSING ALGORITHM

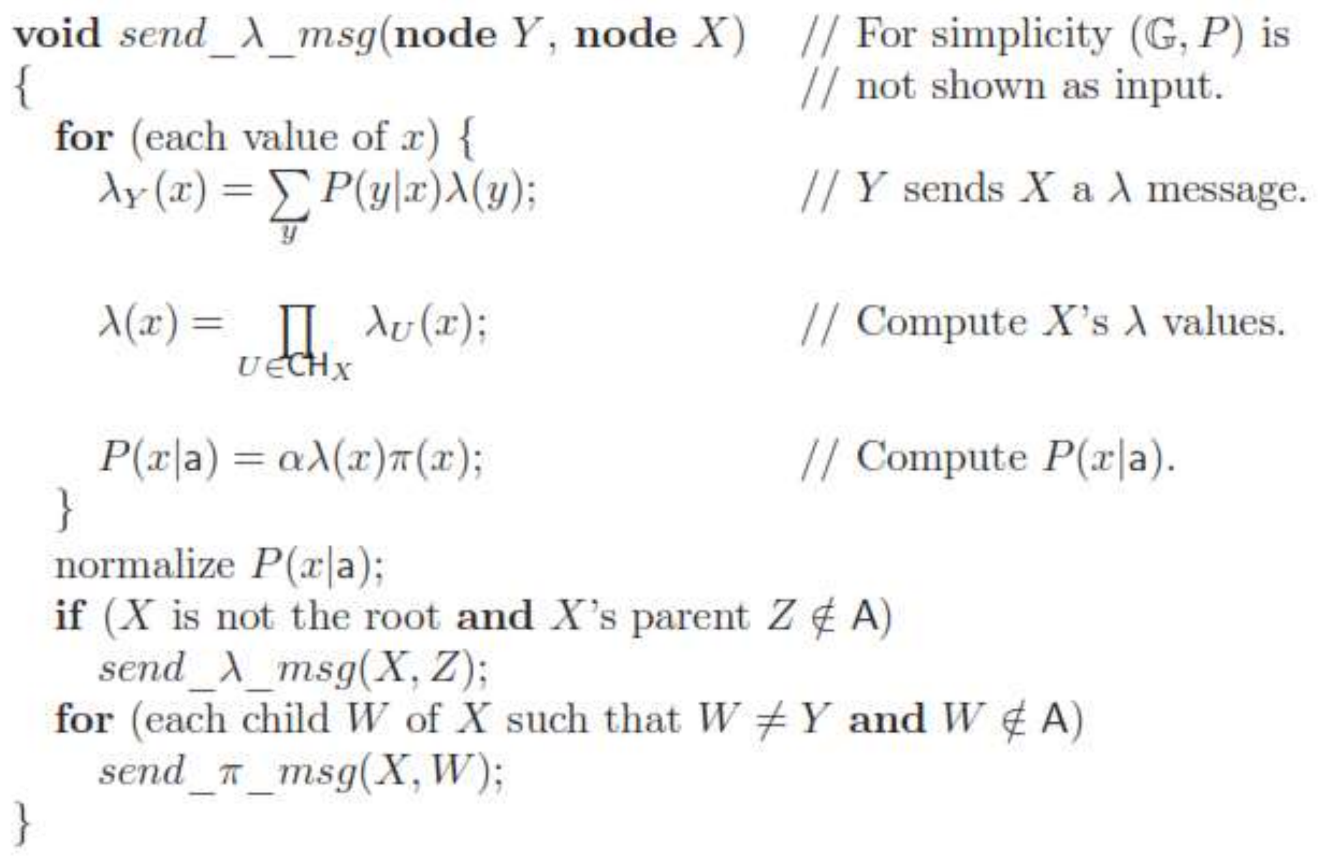
```
void $initial\_tree$   (Bayesian-network& $(\mathbb{G}, P)$ where $\mathbb{G} = (\mathsf{V}, \mathsf{E})$,
                        set-of-variables& $\mathsf{A}$, set-of-variable-values& $\mathsf{a}$)
{
  $\mathsf{A} = \varnothing; \mathsf{a} = \varnothing;$
  for (each $X \in \mathsf{V}$) {
    for (each value $x$ of $X$)
      $\lambda(x) = 1;$                  // Compute $\lambda$ values.
    for (the parent $Z$ of $X$)          // Does nothing if $X$ is the a root.
      for (each value $z$ of $Z$)
        $\lambda_X(z) = 1;$              // Compute $\lambda$ messages.
  }
  for (each value $r$ of the root $R$) {
    $P(r|\mathsf{a}) = P(r);$            // Compute $P(r|\mathsf{a})$.
    $\pi(r) = P(r);$                     // Compute $R$'s $\pi$ values.
  }
  for (each child $X$ of $R$)
    $send\_\pi\_msg(R, X);$
}
```

KIM AND PEARL MESSAGE PASSING ALGORITHM



```
void $update\_tree$ (Bayesian-network& $(\mathbb{G}, P)$ where $\mathbb{G} = (\mathsf{V}, \mathsf{E})$,
                     set-of-variables& $\mathsf{A}$, set-of-variable-values& $\mathsf{a}$,
                     variable $V$, variable-value $\hat{v}$)
{
  $\mathsf{A} = \mathsf{A} \cup \{V\}$; $\mathsf{a} = \mathsf{a} \cup \{\hat{v}\}$;           // Add $V$ to $\mathsf{A}$.
  $\lambda(\hat{v}) = 1$; $\pi(\hat{v}) = 1$; $P(\hat{v}|\mathsf{a}) = 1$;     // Instantiate $V$ to $\hat{v}$.
  for (each value of $v \neq \hat{v}$) {
    $\lambda(v) = 0$; $\pi(v) = 0$; $P(v|\mathsf{a}) = 0$;
  }
  if ($V$ is not the root && $V$'s parent $Z \notin \mathsf{A}$)
    $send\_\lambda\_msg(V, Z)$;
  for (each child $X$ of $V$ such that $X \notin \mathsf{A}$)
    $send\_\pi\_msg(V, X)$;
}
```

KIM AND PEARL MESSAGE PASSING ALGORITHM

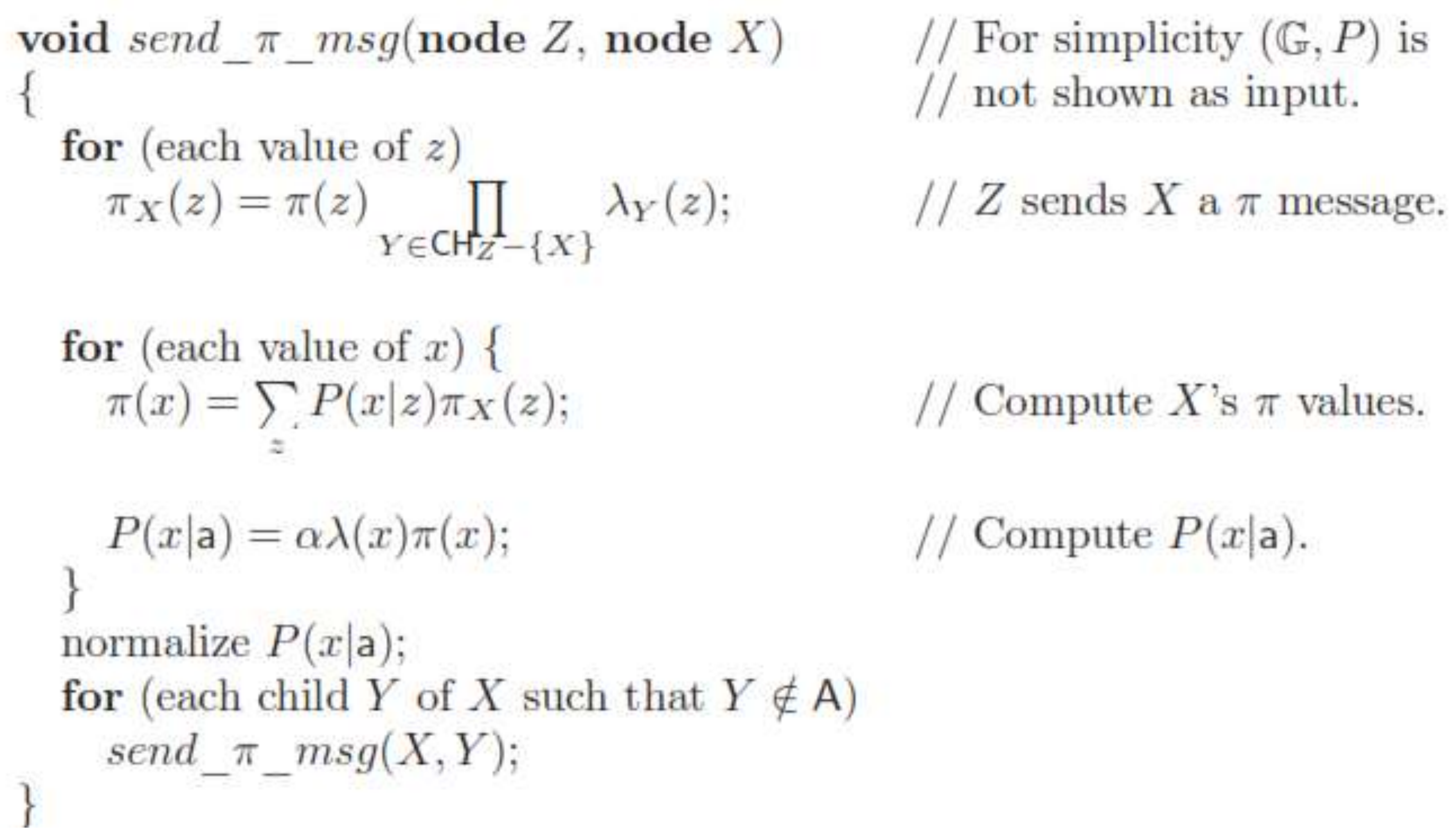


```
void $send\_\lambda\_msg$(node $Y$, node $X$)      // For simplicity $(\mathbb{G}, P)$ is
{                                                 // not shown as input.
   for (each value of $x$) {
      $\lambda_Y(x) = \sum_y P(y|x)\lambda(y)$;          // $Y$ sends $X$ a $\lambda$ message.

      $\lambda(x) = \prod_{U \in \mathsf{CH}_X} \lambda_U(x)$;          // Compute $X$'s $\lambda$ values.

      $P(x|\mathsf{a}) = \alpha\lambda(x)\pi(x)$;           // Compute $P(x|\mathsf{a})$.
   }
   normalize $P(x|\mathsf{a})$;
   if ($X$ is not the root and $X$'s parent $Z \notin \mathsf{A}$)
      $send\_\lambda\_msg(X, Z)$;
   for (each child $W$ of $X$ such that $W \neq Y$ and $W \notin \mathsf{A}$)
      $send\_\pi\_msg(X, W)$;
}
```

KIM AND PEARL MESSAGE PASSING ALGORITHM



```
void $send\_\pi\_msg$(node $Z$, node $X$)          // For simplicity $(\mathbb{G}, P)$ is
{                                                  // not shown as input.
   for (each value of $z$)
      $\pi_X(z) = \pi(z) \prod_{Y \in \mathsf{CH}_Z - \{X\}} \lambda_Y(z)$;     // $Z$ sends $X$ a $\pi$ message.

   for (each value of $x$) {
      $\pi(x) = \sum_z P(x|z)\pi_X(z)$;           // Compute $X$'s $\pi$ values.

      $P(x|\mathsf{a}) = \alpha\lambda(x)\pi(x)$;   // Compute $P(x|\mathsf{a})$.
   }
   normalize $P(x|\mathsf{a})$;
   for (each child $Y$ of $X$ such that $Y \notin \mathsf{A}$)
      $send\_\pi\_msg(X, Y)$;
}
```

HAVING AN EVIDENCE AND REVISING BELIEFS

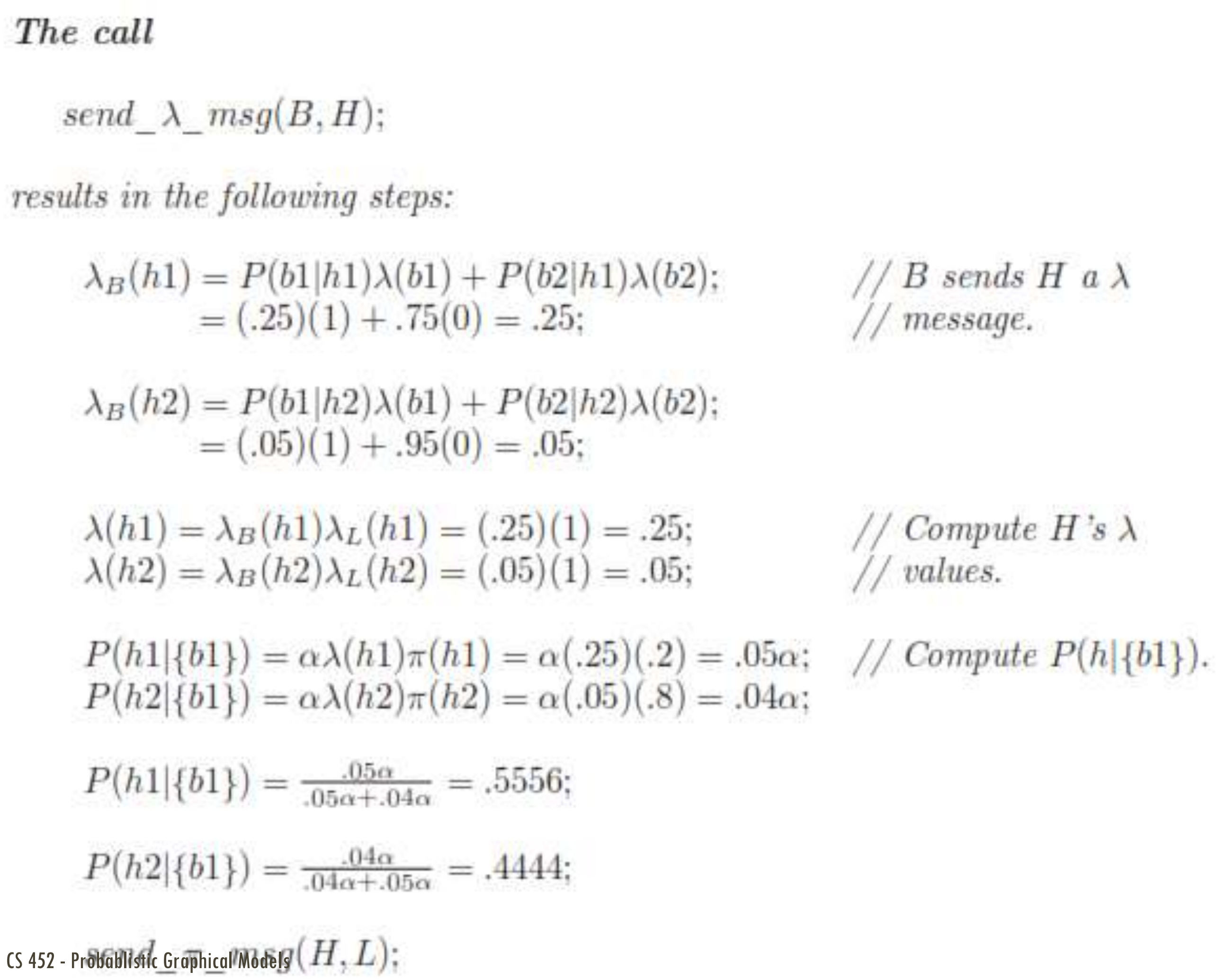
AFTER GETTING EVIDENCE ON B1

```
$\mathsf{A} = \varnothing \cup \{B\} = \{B\};$
$\mathsf{a} = \varnothing \cup \{b1\} = \{b1\};$

$\lambda(b1) = 1;\ \pi(b1) = 1;\ P(b1|\{b1\}) = 1;$        $//\ Instantiate\ B\ for\ b1.$
$\lambda(b2) = 0;\ \pi(b2) = 0;\ P(b2|\{b1\}) = 0;$

$send\_\lambda\_msg(B, H);$
```

POSTERIOR OF ‘H’



The call

send_λ_msg(B, H);

results in the following steps:

$$\begin{aligned}\lambda_B(h1) &= P(b1|h1)\lambda(b1) + P(b2|h1)\lambda(b2); && \textit{// B sends H a } \lambda\\ &= (.25)(1) + .75(0) = .25; && \textit{// message.}\end{aligned}$$

$$\begin{aligned}\lambda_B(h2) &= P(b1|h2)\lambda(b1) + P(b2|h2)\lambda(b2);\\ &= (.05)(1) + .95(0) = .05;\end{aligned}$$

$$\begin{aligned}\lambda(h1) &= \lambda_B(h1)\lambda_L(h1) = (.25)(1) = .25; && \textit{// Compute H's } \lambda\\ \lambda(h2) &= \lambda_B(h2)\lambda_L(h2) = (.05)(1) = .05; && \textit{// values.}\end{aligned}$$

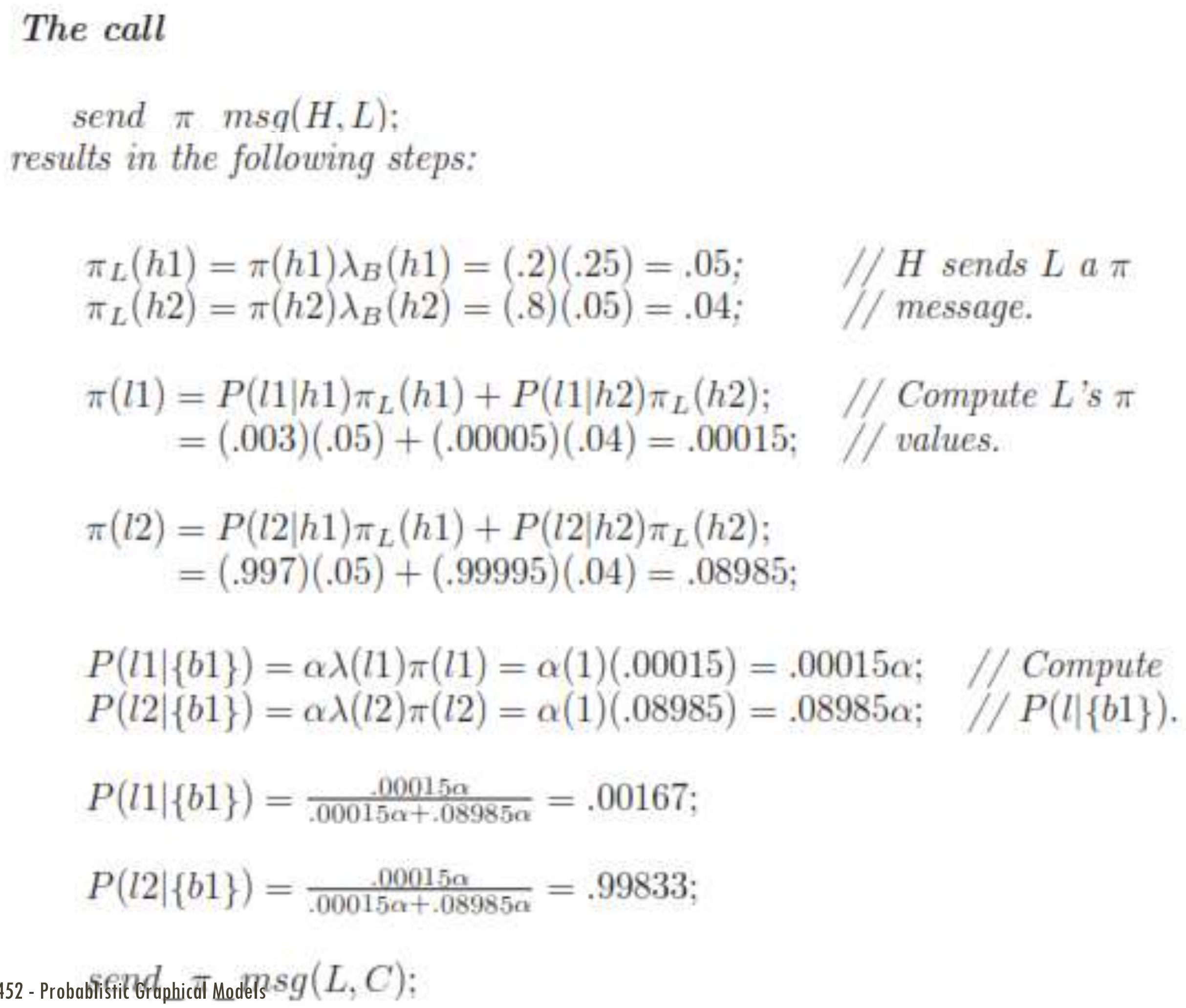
$$\begin{aligned}P(h1|\{b1\}) &= \alpha\lambda(h1)\pi(h1) = \alpha(.25)(.2) = .05\alpha; && \textit{// Compute } P(h|\{b1\}).\\ P(h2|\{b1\}) &= \alpha\lambda(h2)\pi(h2) = \alpha(.05)(.8) = .04\alpha;\end{aligned}$$

$$P(h1|\{b1\}) = \tfrac{.05\alpha}{.05\alpha + .04\alpha} = .5556;$$

$$P(h2|\{b1\}) = \tfrac{.04\alpha}{.04\alpha + .05\alpha} = .4444;$$

send_π_msg(H, L);

POSTERIOR OF 'L'



The call

send π *msg*(H, L);
results in the following steps:

$$\pi_L(h1) = \pi(h1)\lambda_B(h1) = (.2)(.25) = .05; \qquad \text{// } H \text{ sends } L \text{ a } \pi$$
$$\pi_L(h2) = \pi(h2)\lambda_B(h2) = (.8)(.05) = .04; \qquad \text{// message.}$$

$$\pi(l1) = P(l1|h1)\pi_L(h1) + P(l1|h2)\pi_L(h2); \qquad \text{// Compute } L\text{'s } \pi$$
$$= (.003)(.05) + (.00005)(.04) = .00015; \qquad \text{// values.}$$

$$\pi(l2) = P(l2|h1)\pi_L(h1) + P(l2|h2)\pi_L(h2);$$
$$= (.997)(.05) + (.99995)(.04) = .08985;$$

$$P(l1|\{b1\}) = \alpha\lambda(l1)\pi(l1) = \alpha(1)(.00015) = .00015\alpha; \qquad \text{// Compute}$$
$$P(l2|\{b1\}) = \alpha\lambda(l2)\pi(l2) = \alpha(1)(.08985) = .08985\alpha; \qquad \text{// } P(l|\{b1\}).$$

$$P(l1|\{b1\}) = \frac{.00015\alpha}{.00015\alpha+.08985\alpha} = .00167;$$

$$P(l2|\{b1\}) = \frac{.00015\alpha}{.00015\alpha+.08985\alpha} = .99833;$$

send π *msg*(L, C);

POSTERIOR OF ‘C’

The call

*send_*π*_msg*(L, C);

results in the following steps:

$$\begin{aligned}\pi_C(l1) &= \pi(l1) = .00015; && \textit{// L sends C a } \pi\\ \pi_C(l2) &= \pi(l2) = .08985; && \textit{// message.}\end{aligned}$$

$$\begin{aligned}\pi(c1) &= P(c1|l1)\pi_C(l1) + P(c1|l2)\pi_C(l2); && \textit{// Compute C's } \pi\\ &= (.6)(.00015) + (.02)(.08985) = .00189; && \textit{// values.}\end{aligned}$$

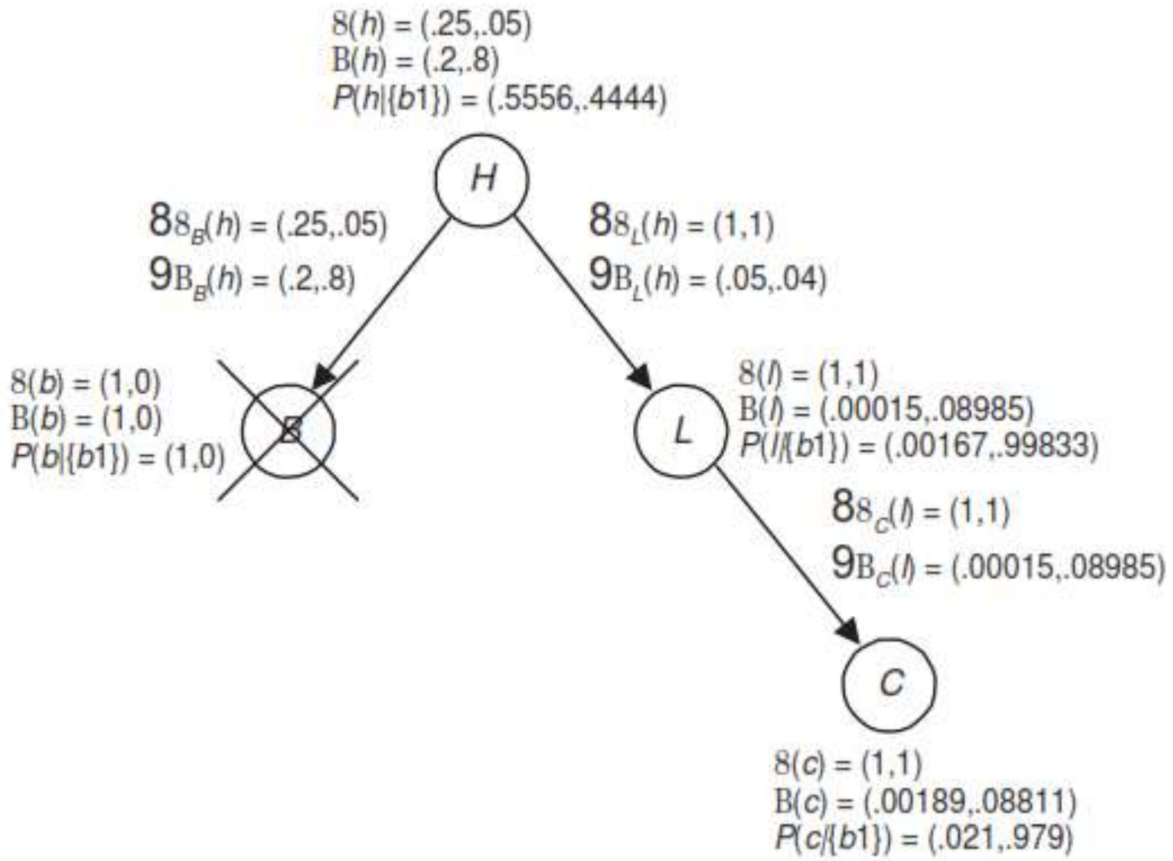
$$\begin{aligned}\pi(c2) &= P(c2|l1)\pi_C(l1) + P(c2|l2)\pi_C(l2);\\ &= (.4)(.00015) + (.98)(.08985) = .08811;\end{aligned}$$

$$\begin{aligned}P(c1|\{b1\}) &= \alpha\lambda(c1)\pi(c1) = \alpha(1)(.00189) = .00189\alpha; && \textit{// Compute}\\ P(c2|\{b1\}) &= \alpha\lambda(c2)\pi(c2) = \alpha(1)(.08811) = .08811\alpha; && \textit{// } P(c|\{b1\}).\end{aligned}$$

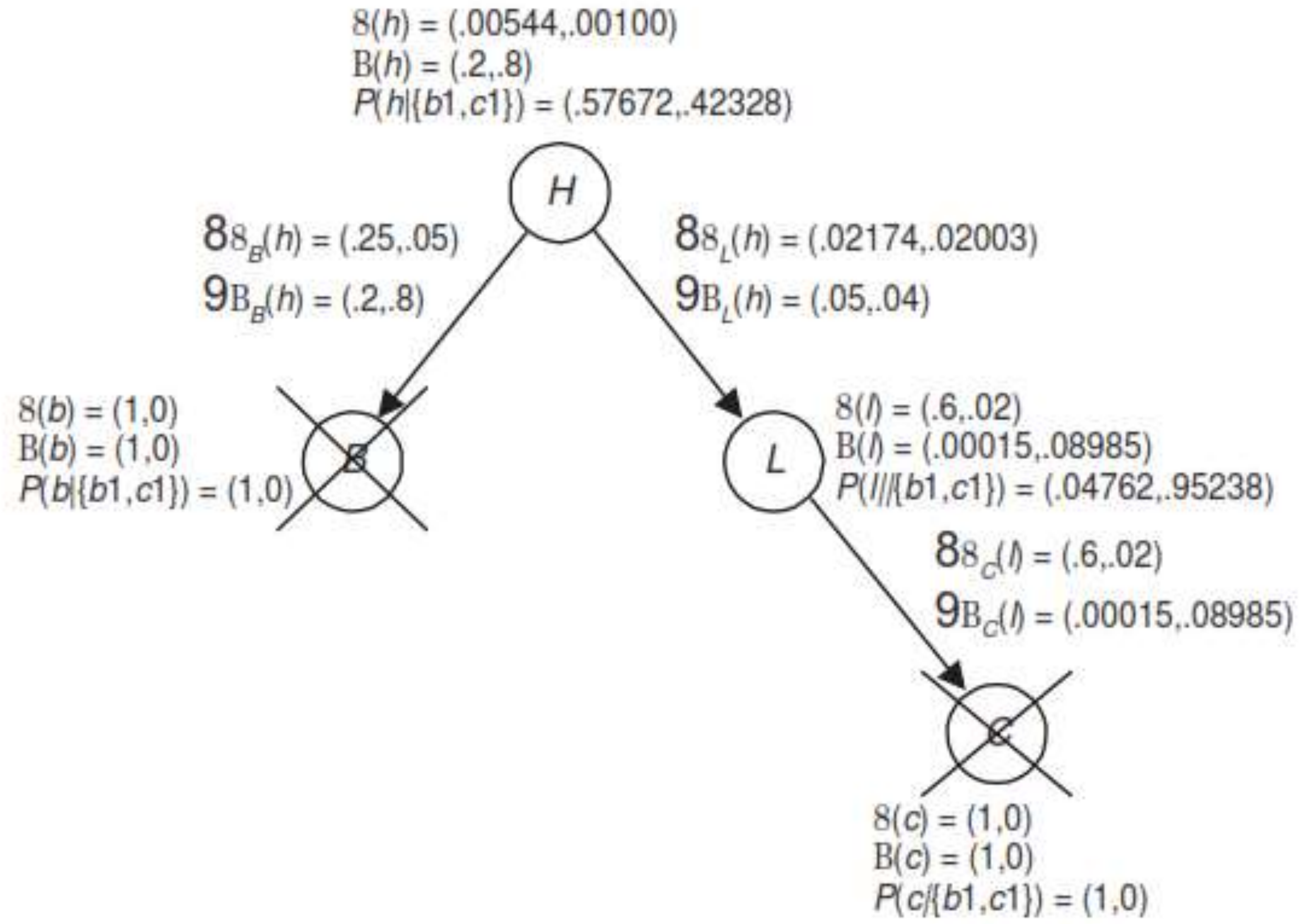
$$P(c1|\{b1\}) = \frac{.00189\alpha}{.00189\alpha + .08811\alpha} = .021;$$

$$P(c2|\{b1\}) = \frac{.08811\alpha}{.00189\alpha + .08811\alpha} = .979;$$

POSTERIOR PROBABILITIES | B1



EVIDENCE ON C1 AFTER B1



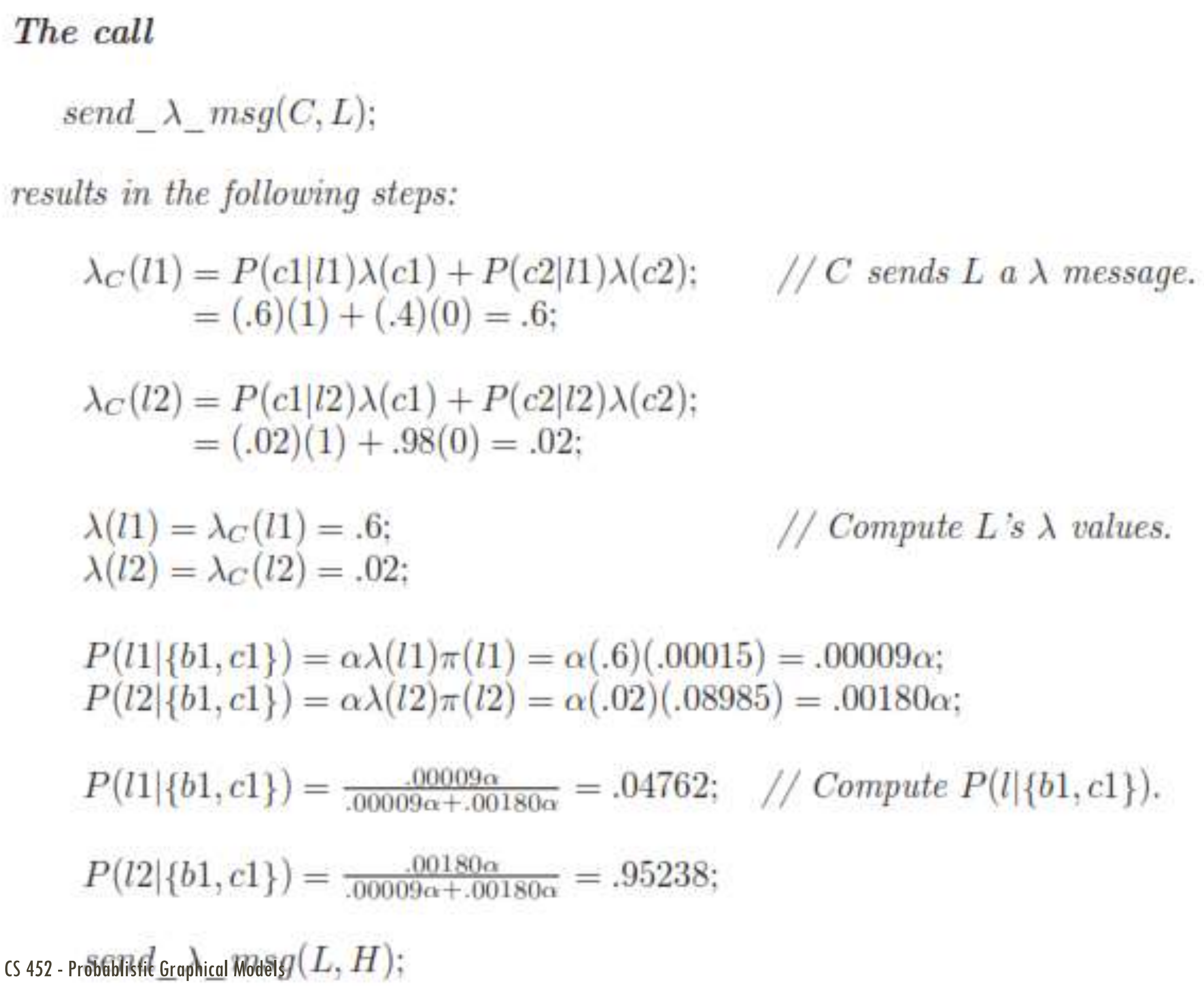
EVIDENCE ON C1 AFTER B1

```
$\mathsf{A} = \{B\} \cup \{C\} = \{B, C\};$
$\mathsf{a} = \{b1\} \cup \{c1\} = \{b1, c1\};$

$\lambda(c1) = 1;\ \pi(c1) = 1;\ P(c1|\{b1, c1\}) = 1;$        $//\ Instantiate\ C\ for\ c1.$
$\lambda(c2) = 0;\ \pi(c2) = 0;\ P(c2|\{b1, c1\}) = 0;$

$send\_\lambda\_msg(C, L);$
```

POSTERIOR OF 'L' | {B1,C1}



The call

$send\_\lambda\_msg(C, L);$

results in the following steps:

$$\begin{aligned}\lambda_C(l1) &= P(c1|l1)\lambda(c1) + P(c2|l1)\lambda(c2); \qquad \textit{// C sends L a } \lambda \textit{ message.}\\ &= (.6)(1) + (.4)(0) = .6;\end{aligned}$$

$$\begin{aligned}\lambda_C(l2) &= P(c1|l2)\lambda(c1) + P(c2|l2)\lambda(c2);\\ &= (.02)(1) + .98(0) = .02;\end{aligned}$$

$$\lambda(l1) = \lambda_C(l1) = .6; \qquad \textit{// Compute L's } \lambda \textit{ values.}$$
$$\lambda(l2) = \lambda_C(l2) = .02;$$

$$P(l1|\{b1, c1\}) = \alpha\lambda(l1)\pi(l1) = \alpha(.6)(.00015) = .00009\alpha;$$
$$P(l2|\{b1, c1\}) = \alpha\lambda(l2)\pi(l2) = \alpha(.02)(.08985) = .00180\alpha;$$

$$P(l1|\{b1, c1\}) = \frac{.00009\alpha}{.00009\alpha + .00180\alpha} = .04762; \qquad \textit{// Compute } P(l|\{b1, c1\}).$$

$$P(l2|\{b1, c1\}) = \frac{.00180\alpha}{.00009\alpha + .00180\alpha} = .95238;$$

$send\_\lambda\_msg(L, H);$

POSTERIOR OF 'H' | {B1,C1}

$send\_\lambda\_msg(L, H);$

results in the following steps:

$$\begin{aligned}\lambda_L(h1) &= P(l1|h1)\lambda(l1) + P(l2|h1)\lambda(l2); \\ &= (.003)(.6) + .997(.02) = .02174;\end{aligned}$$ *// L sends H a λ // message.*

$$\begin{aligned}\lambda_L(h2) &= P(l1|h2)\lambda(l1) + P(l2|h2)\lambda(l2); \\ &= (.00005)(.6) + .99995(.02) = .02003;\end{aligned}$$

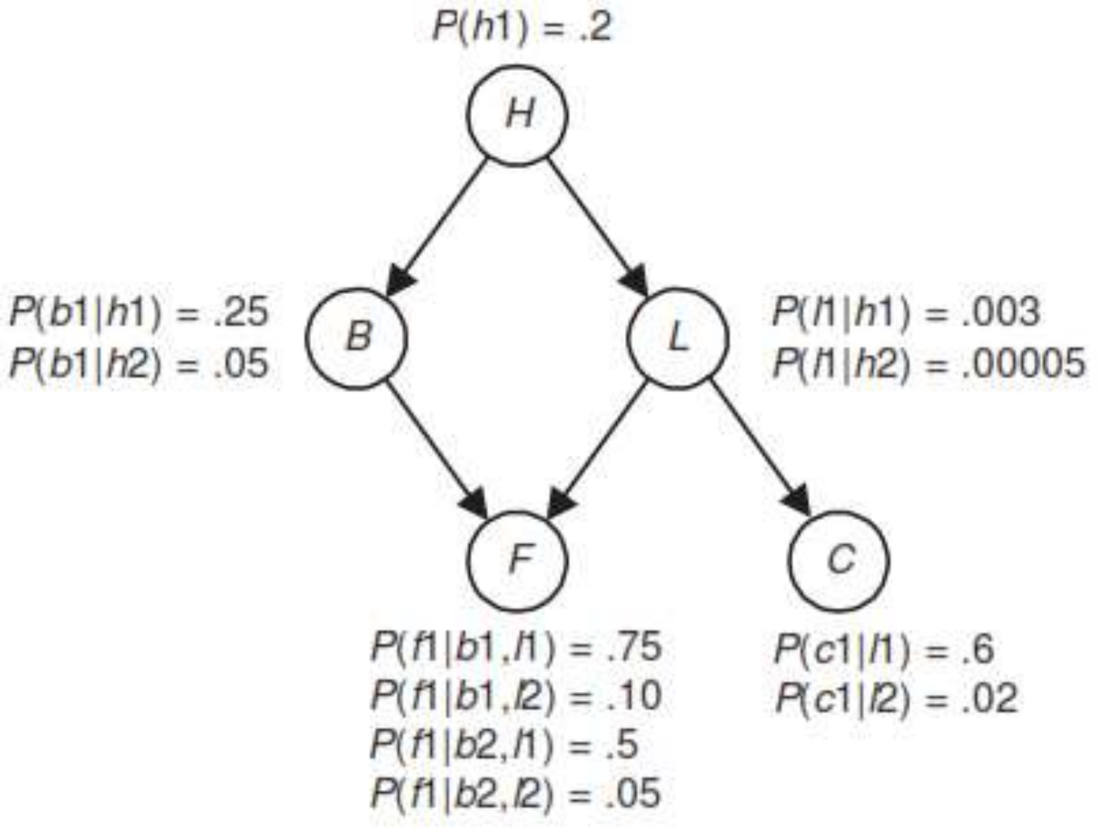
$$\lambda(h1) = \lambda_B(h1)\lambda_L(h1) = (.25)(.02174) = .00544;$$ *// Compute H's λ*
$$\lambda(h2) = \lambda_B(h2)\lambda_L(h2) = (.05)(.02003) = .00100;$$ *// values.*

$$P(h1|\{b1,c1\}) = \alpha\lambda(h1)\pi(h1) = \alpha(.00544)(.2) = .00109\alpha;$$
$$P(h2|\{b1,c1\}) = \alpha\lambda(h2)\pi(h2) = \alpha(.00100)(.8) = .00080\alpha;$$

$$P(h1|\{b1,c1\}) = \frac{.00109\alpha}{.00109\alpha + .00080\alpha} = .57672;$$ *// Compute $P(h|\{b1,c1\})$.*

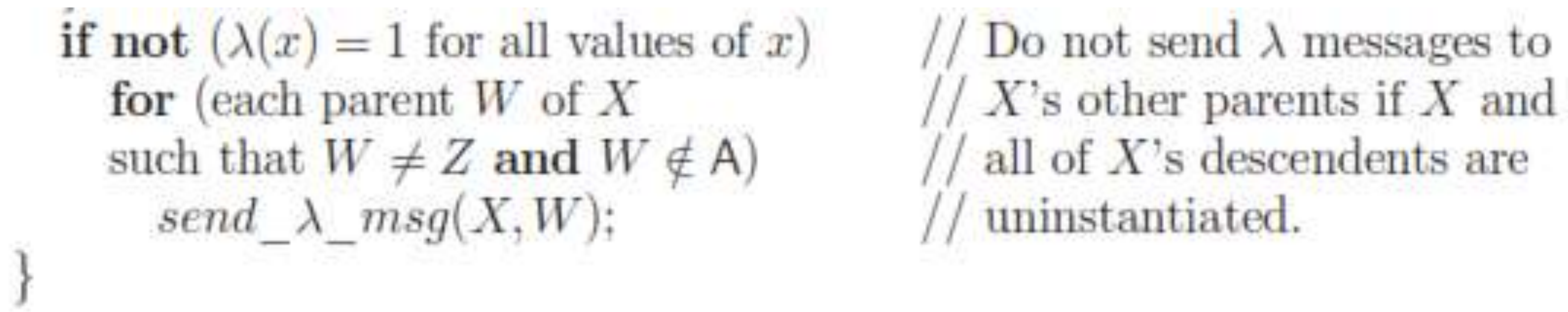
$$P(h2|\{b1,c1\}) = \frac{.0008\alpha}{.00109\alpha + .00080\alpha} = .42328;$$

MULTI-CONNECTION BN



ADAPTING FOR SINGLY CONNECTED BN

There can be nodes with more than one parents:



```
if not ($\lambda(x) = 1$ for all values of $x$)          // Do not send $\lambda$ messages to
    for (each parent $W$ of $X$                         // $X$'s other parents if $X$ and
    such that $W \neq Z$ and $W \notin$ A)              // all of $X$'s descendents are
        $send\_\lambda\_msg(X, W)$;                     // uninstantiated.
}
```

THANKS