

Inference in Belief Trees

Inference in Belief Networks

A Bayesian Network is a complete model for the variables and their relationships.

It can be used to answer queries about them.

Typical question:

Given observed variables, what is the updated knowledge about the other variables.

There are exact inference methods for this task.

Motivation

Choice of universe of discourse

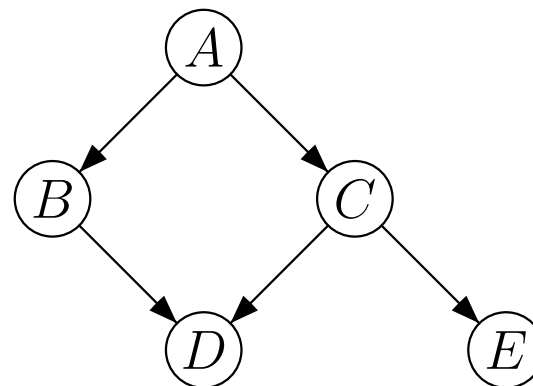
	Variable	Domain
A	metastatic cancer	$\{a_1, a_2\}$
B	increased serum calcium	$\{b_1, b_2\}$
C	brain tumor	$\{c_1, c_2\}$
D	coma	$\{d_1, d_2\}$
E	headache	$\{e_1, e_2\}$

(\cdot_1 — present, \cdot_2 — absent)

$$\Omega = \{a_1, a_2\} \times \cdots \times \{e_1, e_2\}$$

$$|\Omega| = 32$$

Analysis of dependencies



Motivation

$$\left. \begin{array}{l} P(e_1 | c_1) = 0.8 \\ P(e_1 | c_2) = 0.6 \end{array} \right\} \text{headaches common, but more common if tumor present}$$

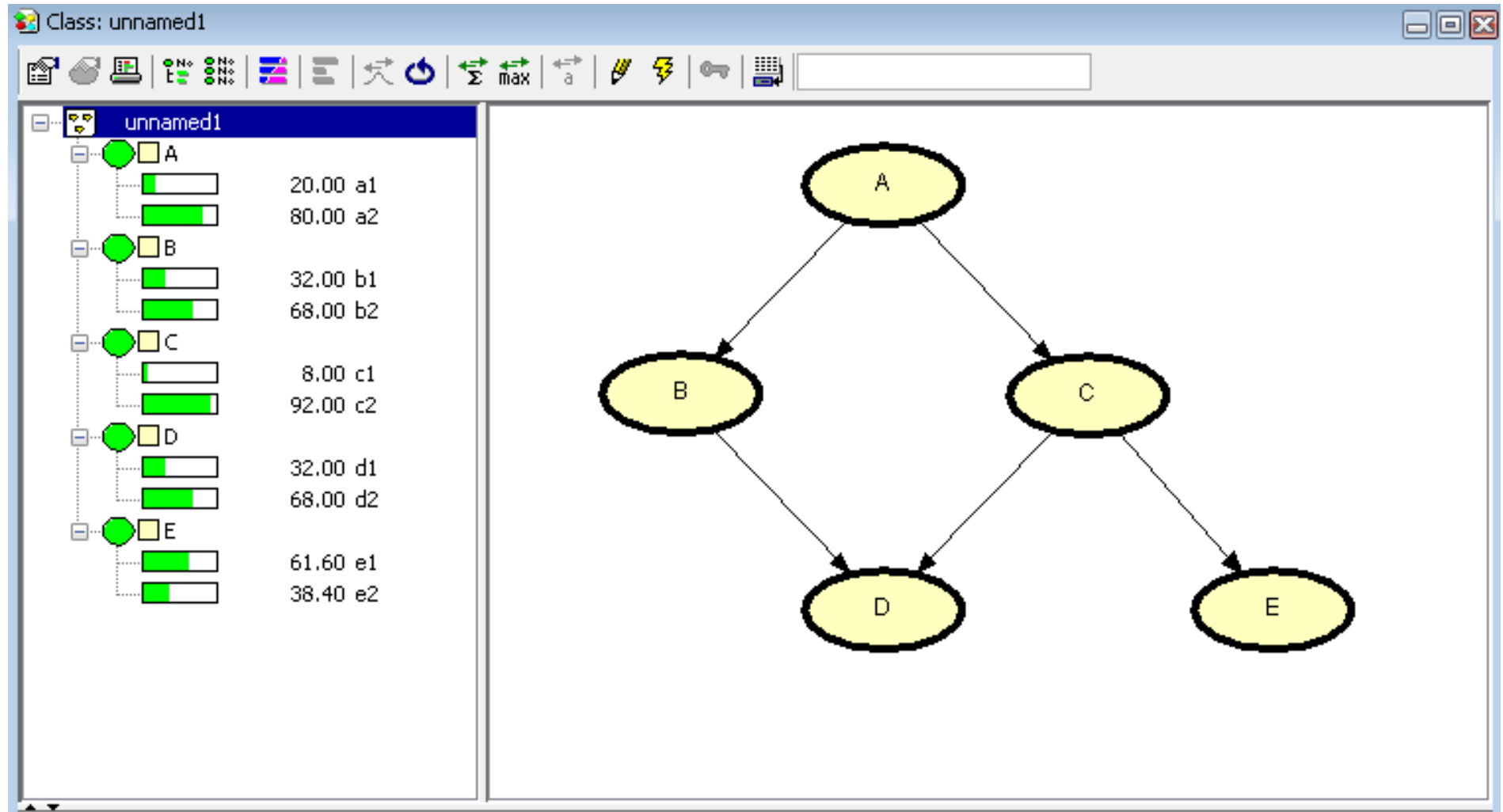
$$\left. \begin{array}{l} P(d_1 | b_1, c_1) = 0.8 \\ P(d_1 | b_1, c_2) = 0.8 \\ P(d_1 | b_2, c_1) = 0.8 \\ P(d_1 | b_2, c_2) = 0.05 \end{array} \right\} \text{coma rare but common, if either cause is present}$$

$$\left. \begin{array}{l} P(b_1 | a_1) = 0.8 \\ P(b_1 | a_2) = 0.2 \end{array} \right\} \begin{array}{l} \text{increased calcium uncommon,} \\ \text{but common consequence of metastases} \end{array}$$

$$\left. \begin{array}{l} P(c_1 | a_1) = 0.2 \\ P(c_1 | a_2) = 0.05 \end{array} \right\} \text{brain tumor rare, and uncommon consequence of metastases}$$

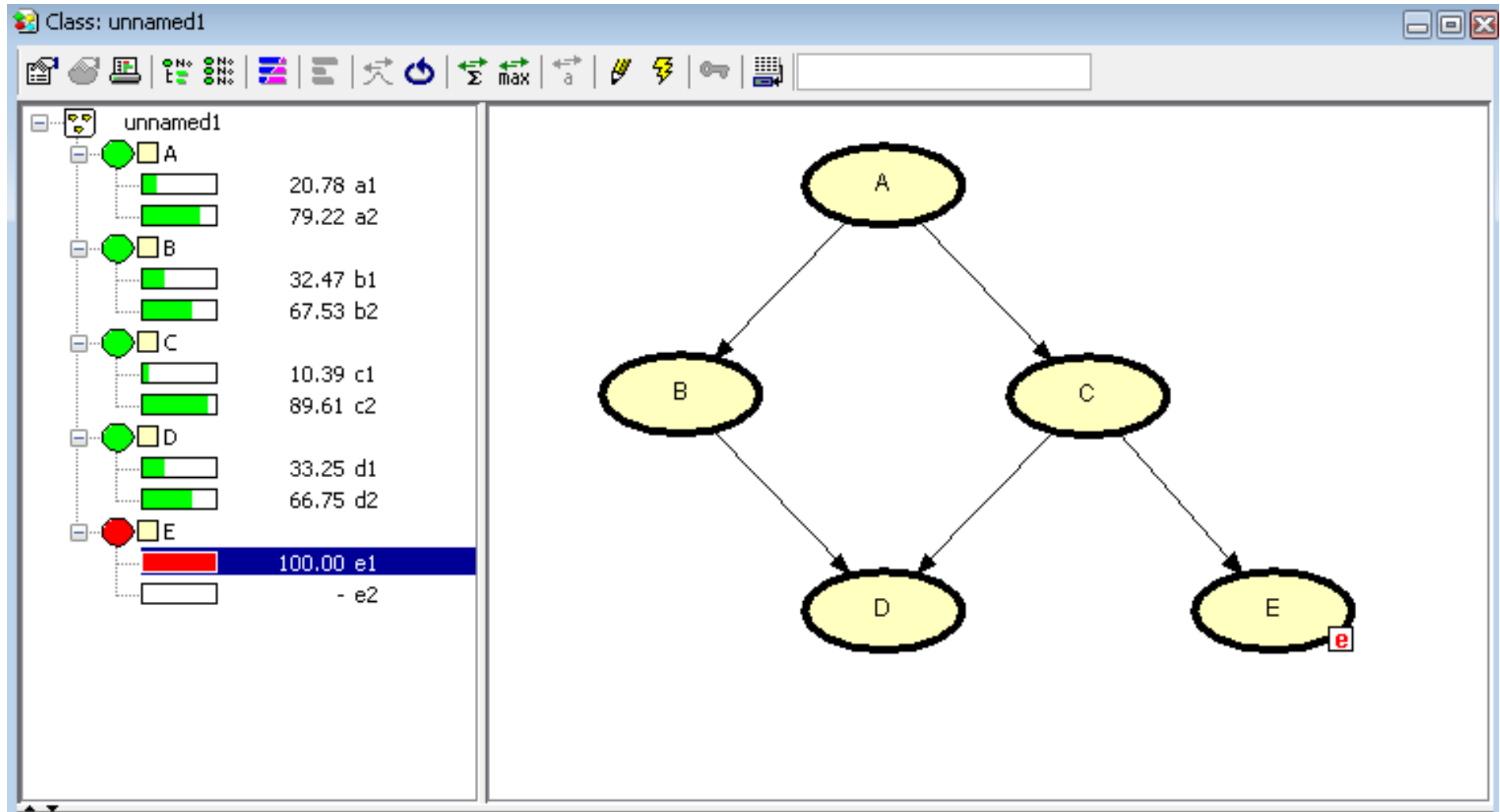
$$P(a_1) = 0.2 \quad \left. \right\} \text{incidence of metastatic cancer in relevant clinic}$$

Motivation



Marginal distributions in the HUGIN tool.

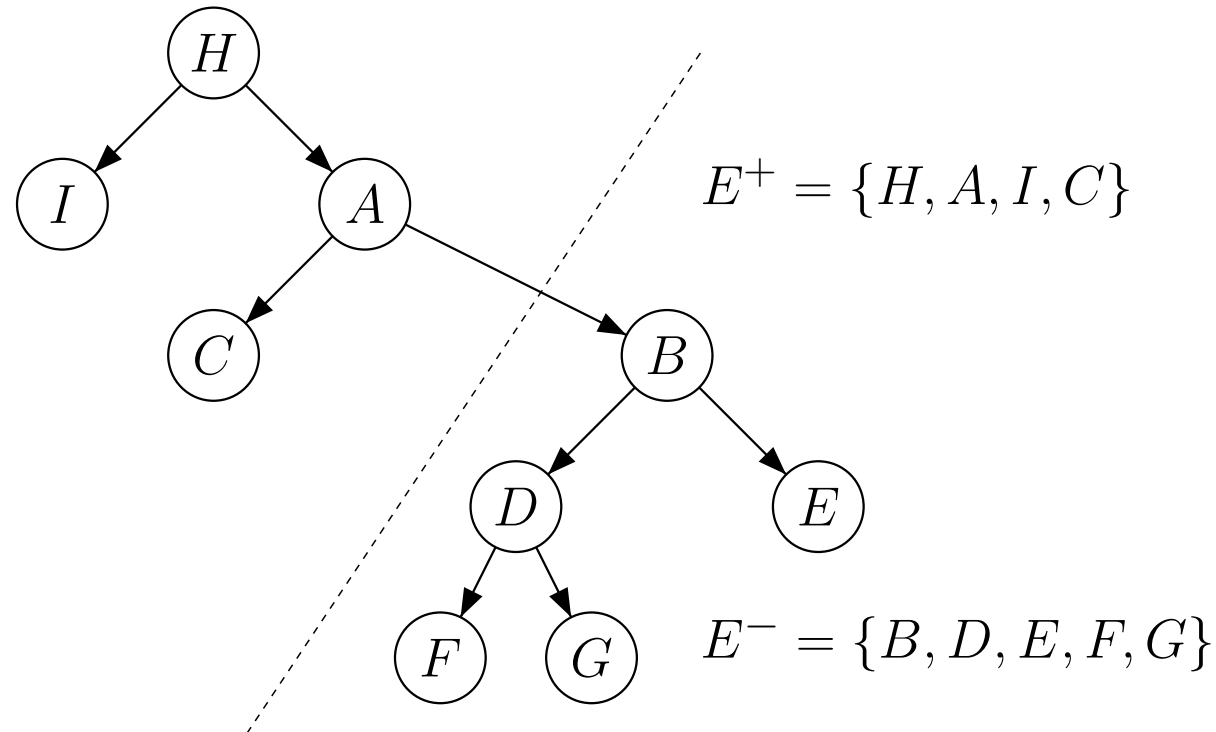
Motivation



Conditional marginal distributions with evidence $E = e_1$

Probability Propagation in Trees

Given that the BN has a tree structure, any node divides the network into two independent subtrees



Basic Equations

Given certain evidence, E (subset of instantiated variables), the posterior probability for a value i of any variable B , can be obtained by applying the Bayes rule:

$$P(B_i|E) = P(B_i)P(E|B_i)/P(E)$$

We can separate the evidence into:

- E^- : Evidence in the tree rooted in B .
- E^+ : All other evidence.

Then:

$$P(B_i|E) = P(B_i)P(E^-, E^+|B_i)/P(E)$$

Basic Equations

Given that E^+ and E^- are independent, by applying the Bayes rule again, we obtain.

$$P(B_i|E) = \alpha P(B_i|E^+)P(E^-|B_i)$$

Where α is a normalization constant.

We define the following terms:

$$\lambda(B_i) = P(E^-|B_i)$$

$$\pi(B_i) = P(B_i|E^+)$$

Then:

$$P(B_i|E) = \alpha \pi(B_i) \lambda(B_i)$$

Propagation Algorithm

The computation of the posterior probability of any node B is decomposed into two parts:

- the evidence coming from the sons of B in the tree (λ)
- and the evidence coming from the parent of B , (π)

Each node B in the tree is like a simple processor that stores

- its vectors $\pi(B)$ and $\lambda(B)$
- its conditional probability table, $P(B|A)$

Evidence is propagated via a message passing mechanism

- each node sends the corresponding messages to its parent and sons in the tree

Messages

A message sent from node B to its parent A

$$\lambda_B(A_i) = \sum_j P(B_j|A_i)\lambda(B_j)$$

A message sent from node B to its son S_k :

$$\pi_k(B_i) = \alpha\pi(B_j) \prod_{l \neq k} \lambda_l(B_j)$$

where l refers to each one of the sons of B

Combination and Propagation

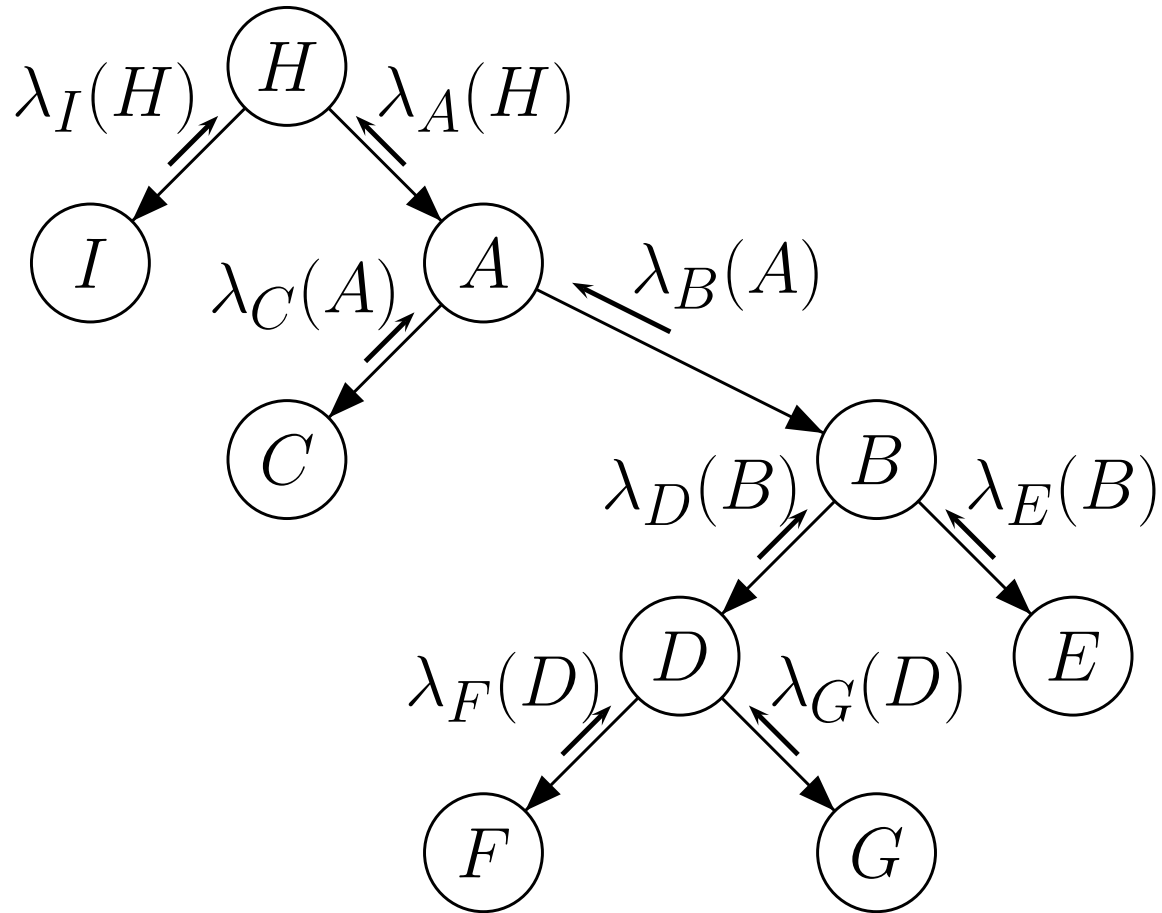
Each node can receive several λ messages, which are combined via a term by term multiplication for the λ messages received from each son (S_k):

$$\lambda(B_i) = \prod_k \lambda_{S_k}(B_i)$$

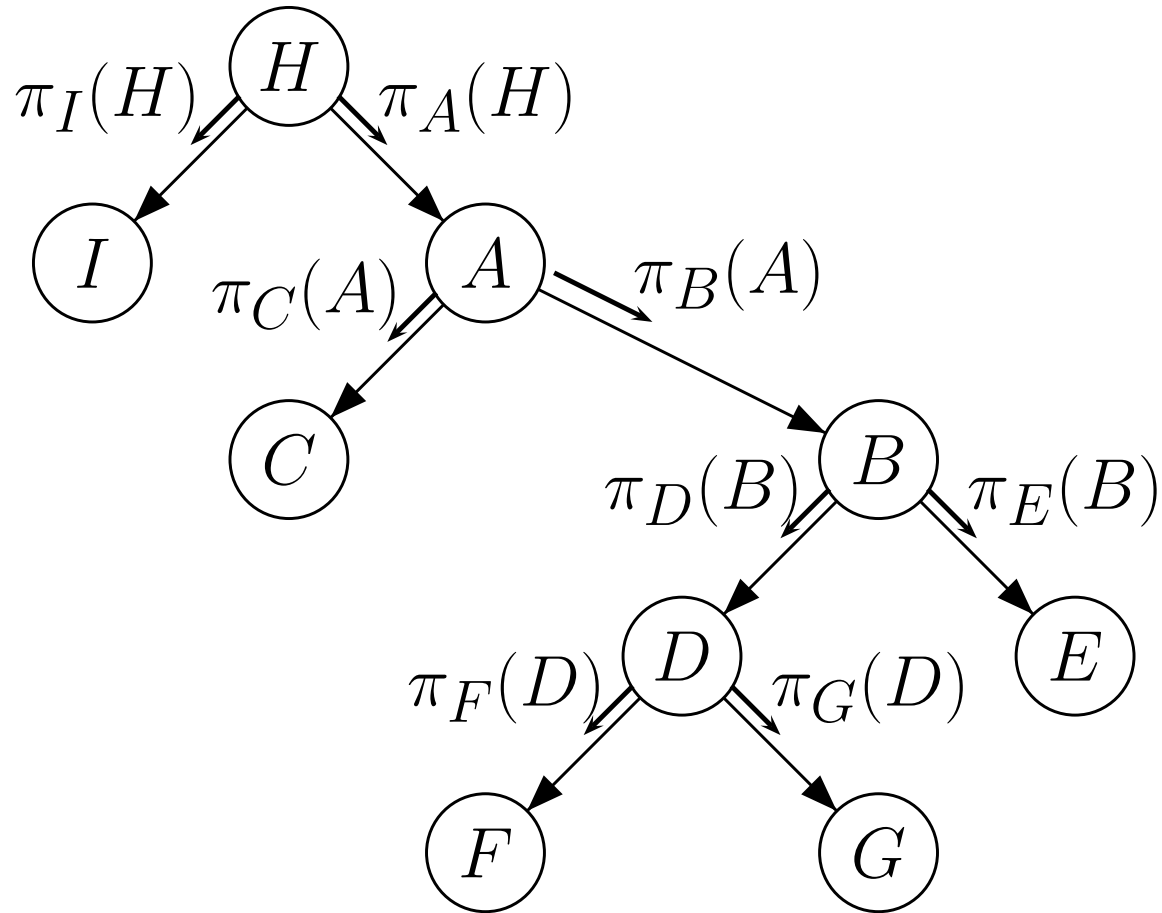
The propagation algorithm starts by assigning the evidence to the known variables, and then propagating it through the message passing mechanism until:

- the root of the tree is reached for the λ messages
- the leaves are reached for the π messages

Bottom-up propagation



Top-down propagation



Initial Conditions

Leaf nodes:

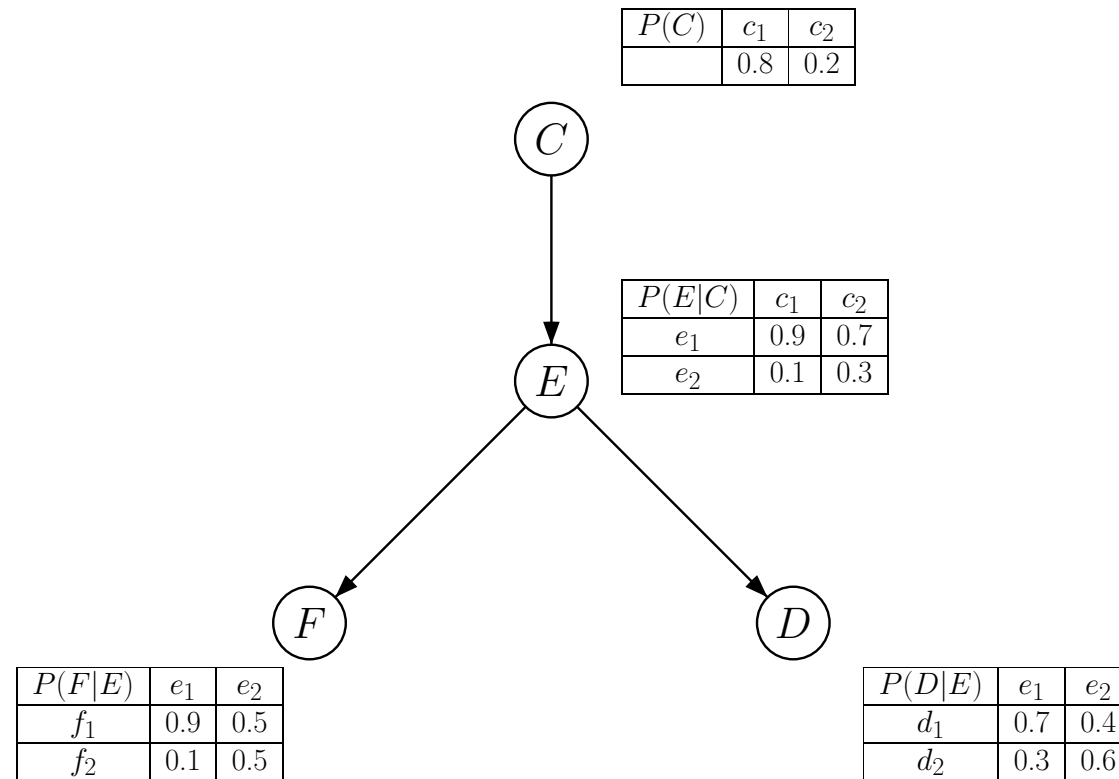
- If not known, $\lambda = [1, 1, \dots, 1]$ (a uniform distribution)
- If known, $\lambda = [0, 0, \dots, 1, \dots, 0]$ (one for the assigned value and zero for all other values).

.

Root node:

- If not known, $\pi = P(A)$ (prior marginal probability vector)
- If known, $\pi = [0, 0, \dots, 1, \dots, 0]$ (one for the assigned value and zero for all other values).

Top-down propagation



Consider that the only evidence is $F = \text{false}$ - initial conditions for the leaf nodes are: $\lambda_F = [1, 0]$ and $\lambda_D = [1, 1]$ (no evidence)

Example - λ propagation

Multiplying the λ vectors by the corresponding CPTs

$$\lambda_F(E) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 0.9 & 0.5 \\ 0.1 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.5 \end{bmatrix}$$

$$\lambda_D(E) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Then, $\lambda(E)$ is obtained by combining the messages from its two sons:

$$\lambda(E) = \begin{bmatrix} 0.9 & 0.5 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.9 & 0.5 \end{bmatrix}$$

Propagation to its parent, C:

$$\lambda_E(C) = \begin{bmatrix} 0.9 & 0.5 \end{bmatrix} \begin{bmatrix} 0.9 & 0.7 \\ 0.1 & 0.3 \end{bmatrix} = \begin{bmatrix} 0.86 & 0.78 \end{bmatrix}$$

Example - π propagation

Given that C is not instantiated, $\pi(C) = [0.8 \ 0.2]$

Propagate to its son, E , which also corresponds to multiplying the π vector by the corresponding CPT:

$$\pi(E) = [0.8 \ 0.2] \begin{bmatrix} 0.9 & 0.7 \\ 0.1 & 0.3 \end{bmatrix}^T = [0.86 \ 0.14]$$

We now propagate to its son D ; however, given that E has another son, F , we also need to consider the λ message from this other son, thus:

$$\pi(D) = [0.86 \ 0.14] \times [0.9 \ 0.5] \begin{bmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{bmatrix}^T = [0.57 \ 0.27]$$

Example - Posterior Probabilities

Given the λ and π vectors for each unknown variable, we just multiply them term by term and then normalize to obtain the posterior probabilities:

$$\begin{aligned} P(C) &= \begin{bmatrix} 0.8 & 0.2 \end{bmatrix} \times \begin{bmatrix} 0.86 & 0.78 \end{bmatrix} \approx \alpha \begin{bmatrix} 0.69 & 0.16 \end{bmatrix} \\ &\approx \begin{bmatrix} 0.815 & 0.185 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} P(E) &= \begin{bmatrix} 0.86 & 0.14 \end{bmatrix} \times \begin{bmatrix} 0.9 & 0.5 \end{bmatrix} \approx \alpha \begin{bmatrix} 0.77 & 0.07 \end{bmatrix} \\ &\approx \begin{bmatrix} 0.917 & 0.083 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} P(D) &= \begin{bmatrix} 0.57 & 0.27 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \end{bmatrix} \approx \alpha \begin{bmatrix} 0.57 & 0.27 \end{bmatrix} \\ &\approx \begin{bmatrix} 0.675 & 0.325 \end{bmatrix} \end{aligned}$$

Analysis

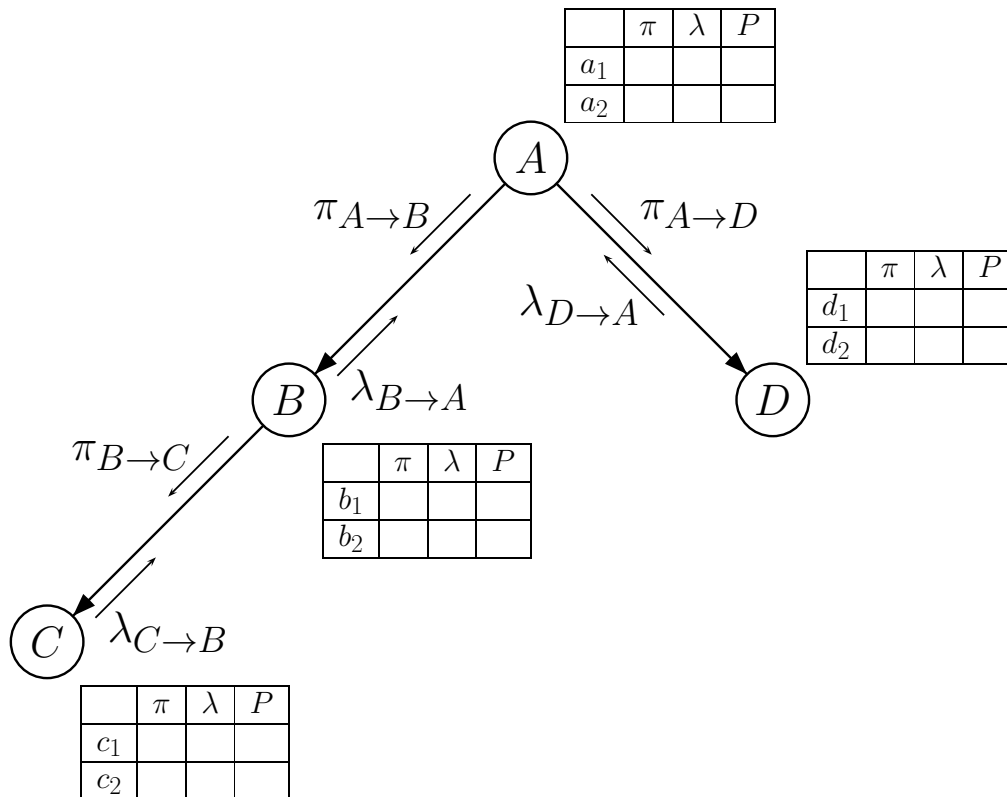
The time complexity to obtain the posterior probability of all the variables in the tree is proportional to the diameter of the network (the number of arcs in the trajectory from the root to the most distant leaf).

The message passing mechanism can be directly extended to polytrees, as these are also singly connected networks. In this case, a node can have multiple parents, so the λ messages should be sent from a node to all its parents

The propagation algorithm only applies to singly connected network

Implementation of Belief Trees

Belief Tree:



Parameters:

$$P(a_1) = 0.1 \quad P(b_1 | a_1) = 0.7$$

$$P(b_1 | a_2) = 0.2$$

$$P(d_1 | a_1) = 0.8 \quad P(c_1 | b_1) = 0.4$$

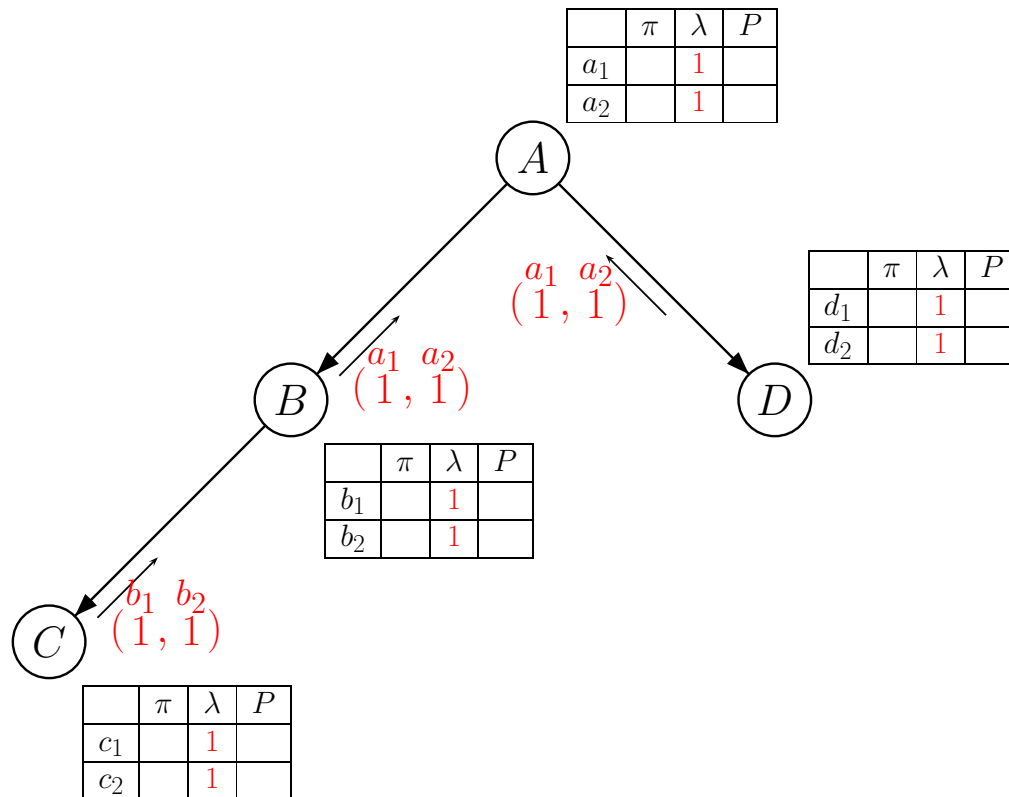
$$P(d_1 | a_2) = 0.4 \quad P(c_1 | b_2) = 0.001$$

Desired:

$$P(A) \quad P(B) \quad P(C) \quad P(D)$$

Implementation of Belief Trees (2)

Belief Tree:

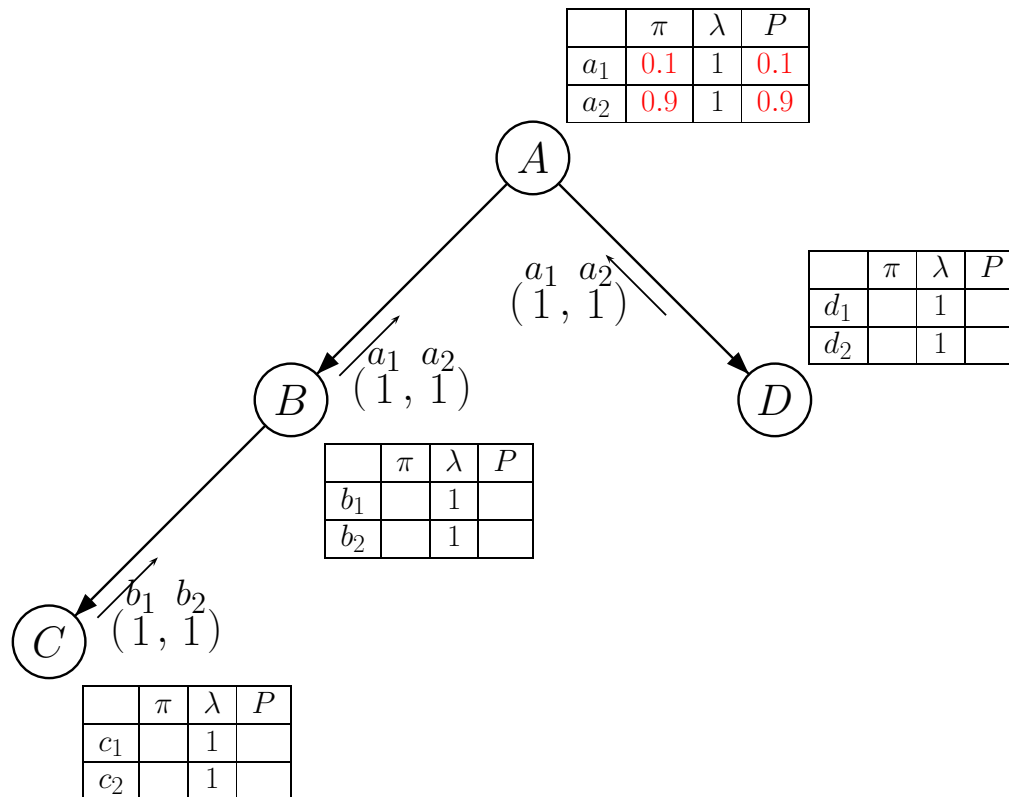


Initialization Phase:

Set all λ -messages and λ -values to 1.

Implementation of Belief Trees (3)

Belief Tree:



Initialization Phase:

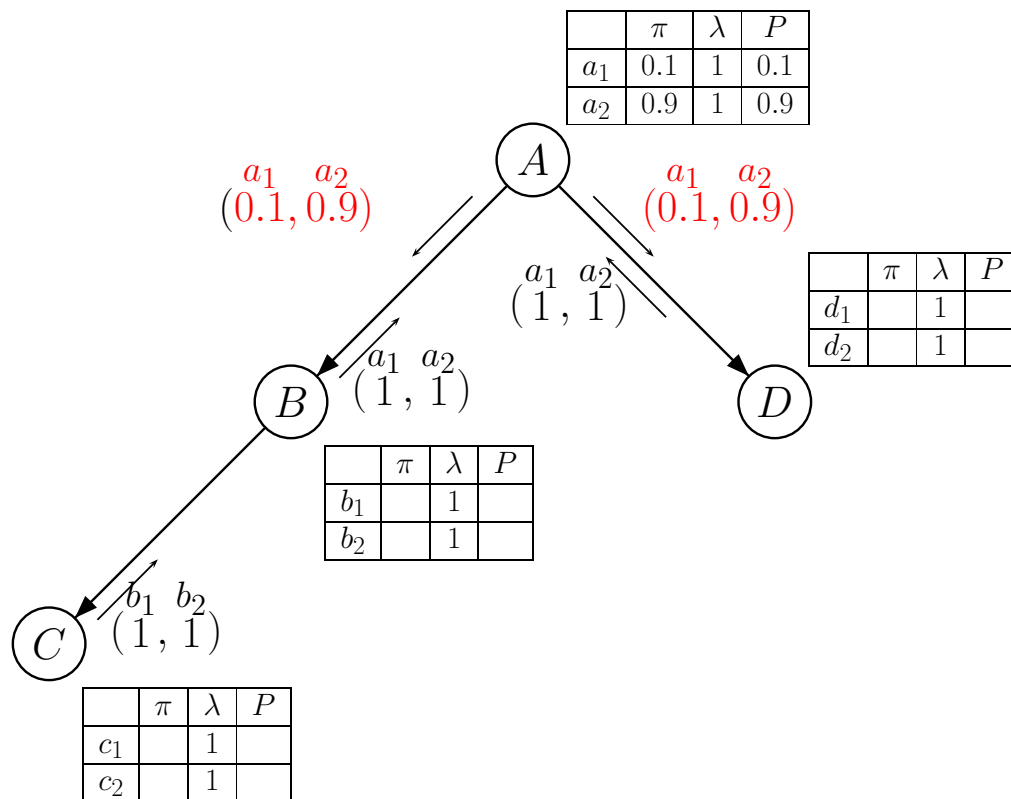
Set all λ -messages and λ -values to 1.

$\pi(a_1) = P(a_1)$ and

$\pi(a_2) = P(a_2)$

Implementation of Belief Trees (4)

Belief Tree:



Initialization Phase:

Set all λ -messages and λ -values to 1.

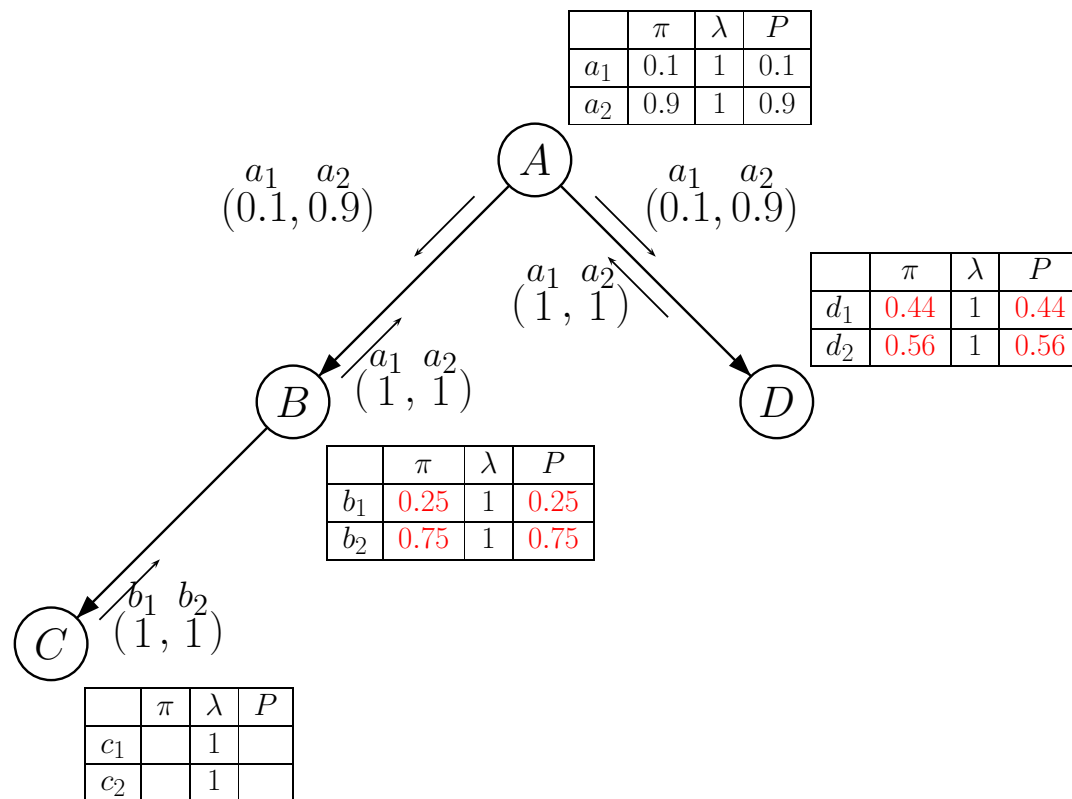
$\pi(a_1) = P(a_1)$ and

$\pi(a_2) = P(a_2)$.

A sends π -messages to B and D.

Implementation of Belief Trees (5)

Belief Tree:



Initialization Phase:

Set all λ -messages and λ -values to 1.

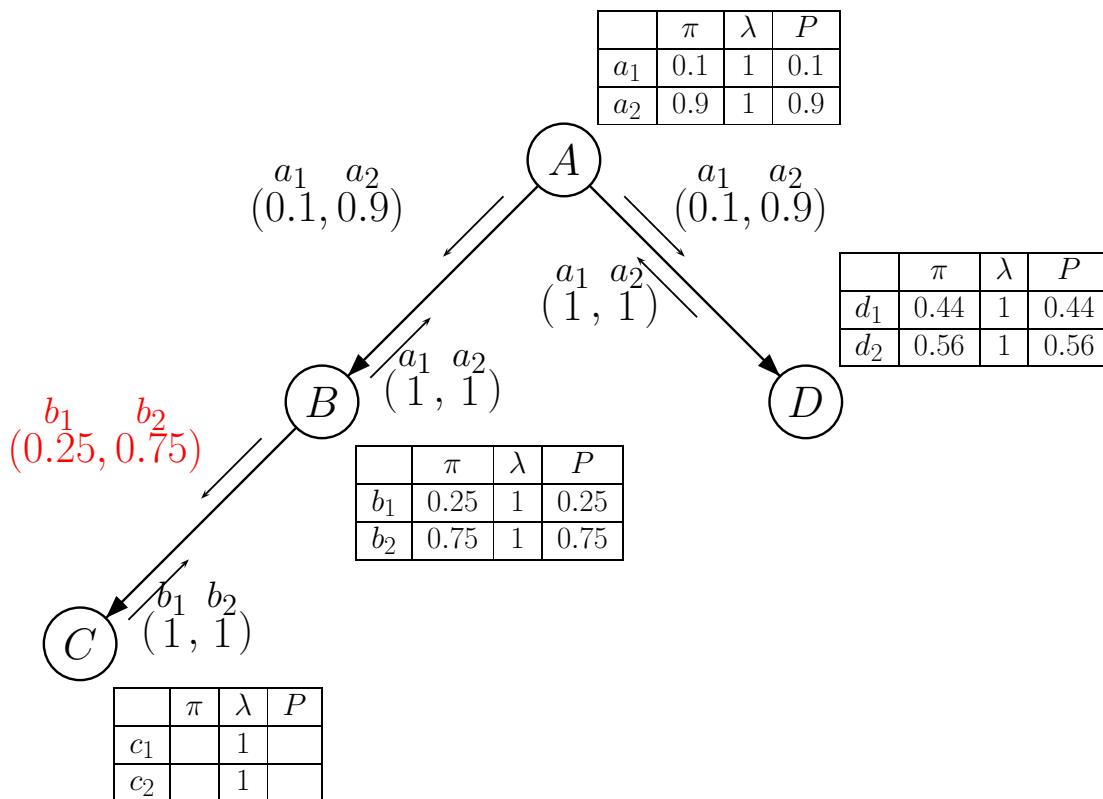
$\pi(a_1) = P(a_1)$ and
 $\pi(a_2) = P(a_2)$.

A sends π -messages to B and D.

B and D update their π -values.

Implementation of Belief Trees (6)

Belief Tree:



Initialization Phase:

Set all λ -messages and λ -values to 1.

$\pi(a_1) = P(a_1)$ and
 $\pi(a_2) = P(a_2)$.

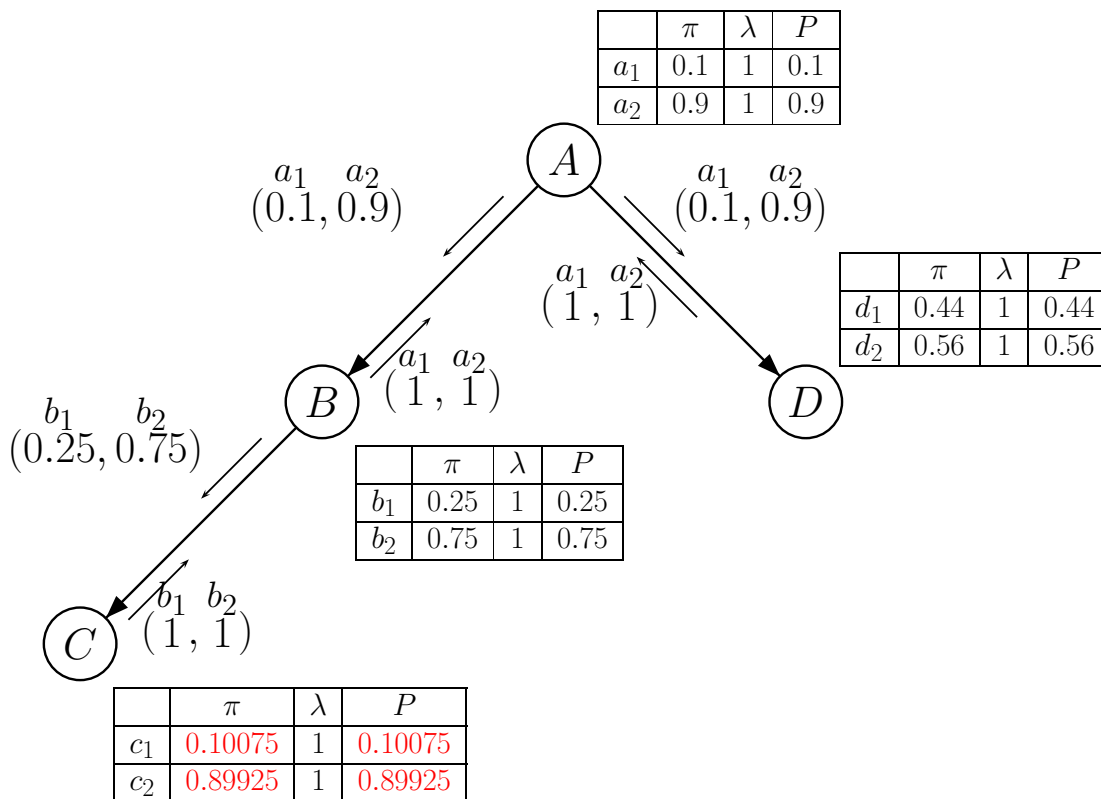
A sends π -messages to B and D.

B and D update their π -values.

B sends π -message to C.

Implementation of Belief Trees (7)

Belief Tree:



Initialization Phase:

Set all λ -messages and λ -values to 1.

$\pi(a_1) = P(a_1)$ and
 $\pi(a_2) = P(a_2)$.

A sends π -messages to B and D.

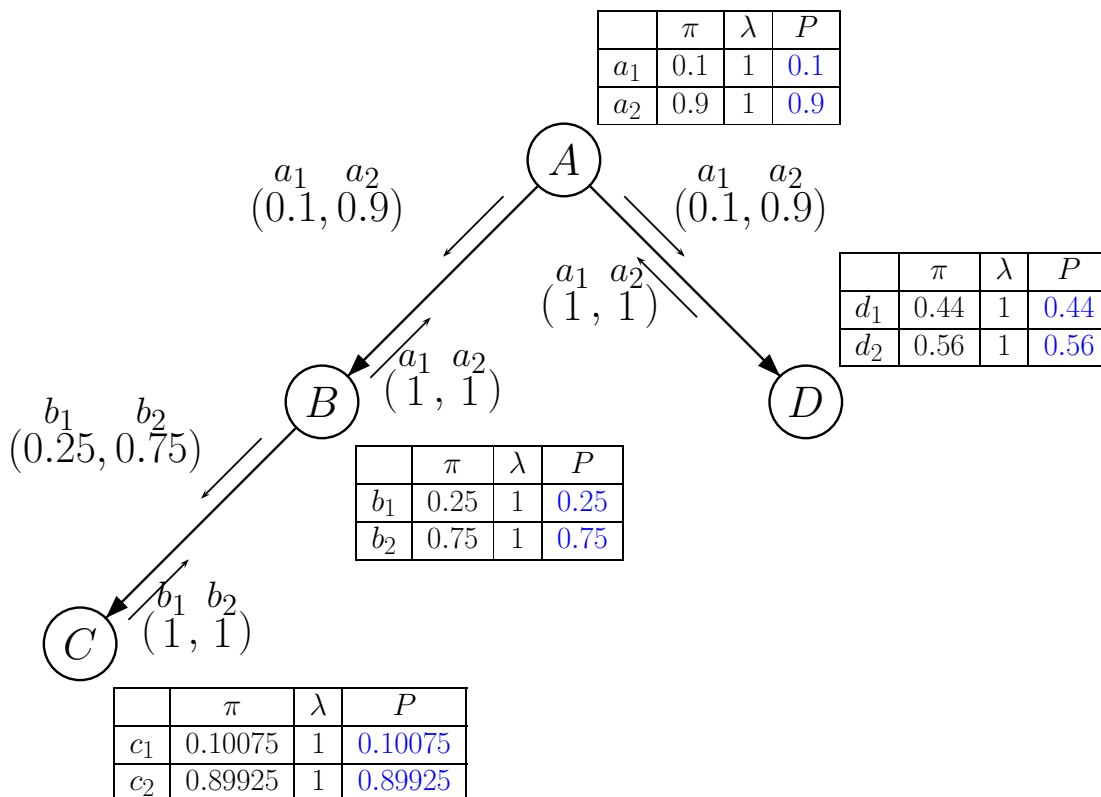
B and D update their π -values.

B sends π -message to C.

C updates its π -value.

Implementation of Belief Trees (8)

Belief Tree:



Initialization Phase:

Set all λ -messages and λ -values to 1.

$\pi(a_1) = P(a_1)$ and
 $\pi(a_2) = P(a_2)$.

A sends π -messages to B and D.

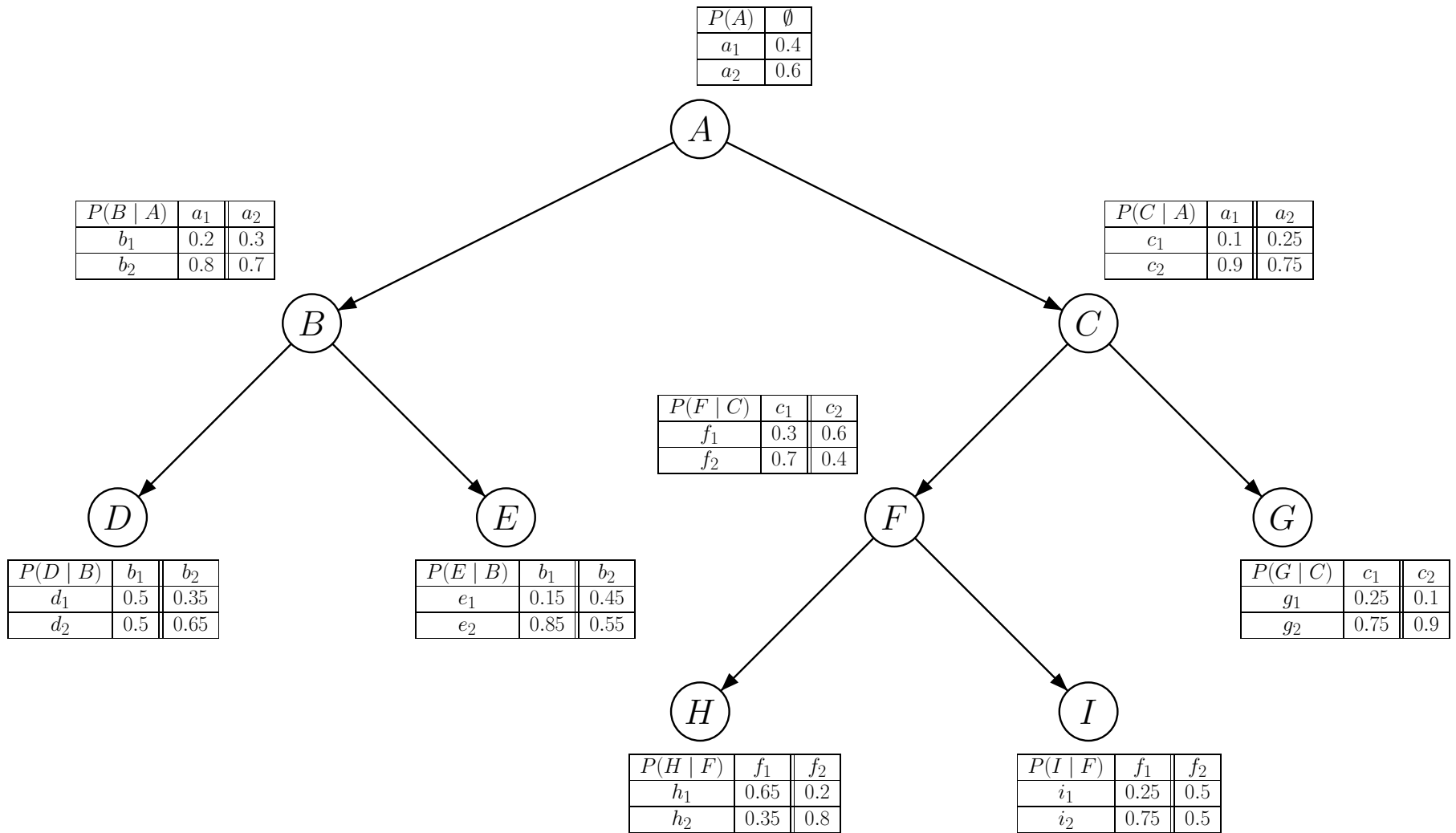
B and D update their π -values.

B sends π -message to C.

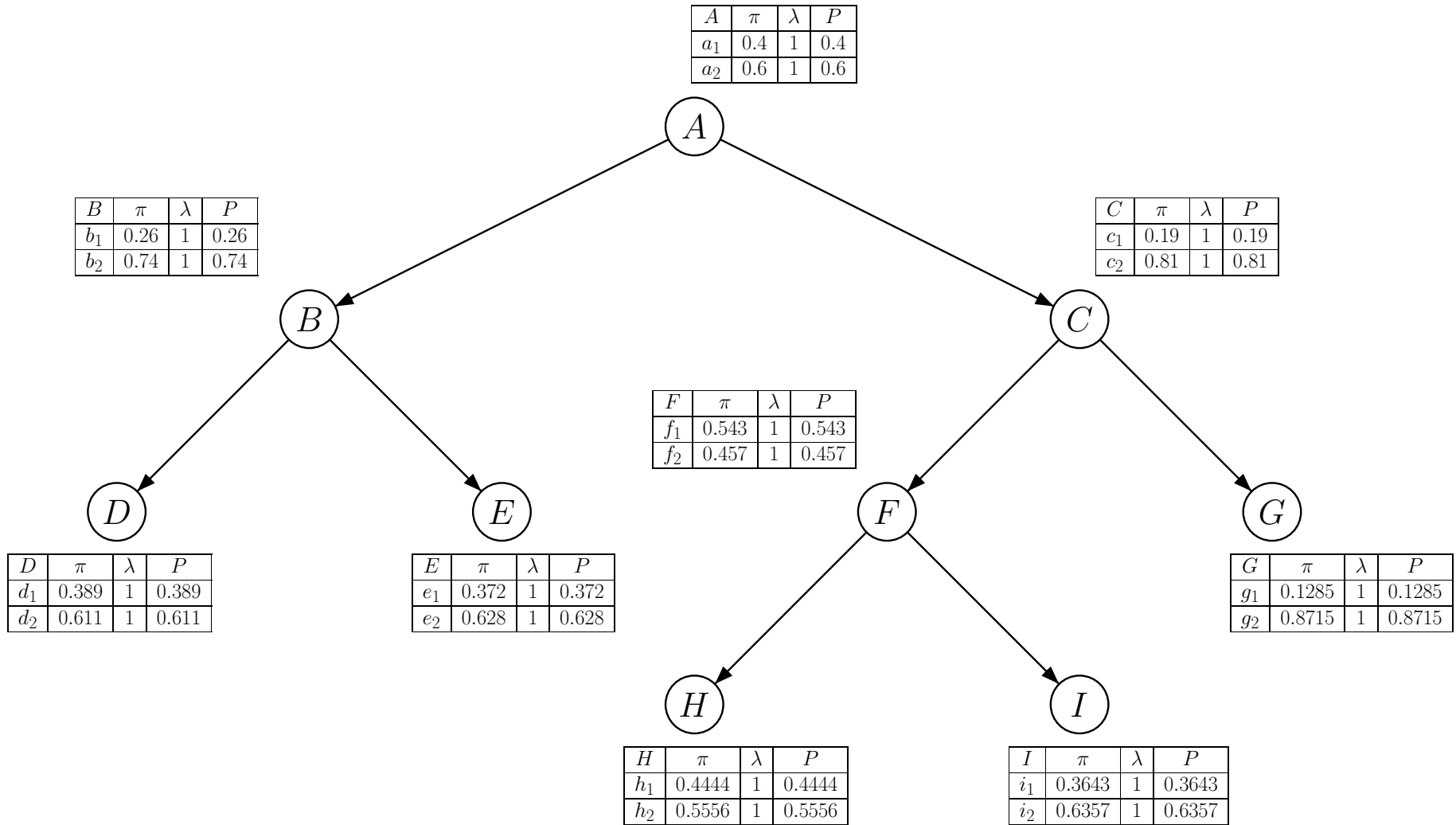
C updates its π -value.

Initialization finished.

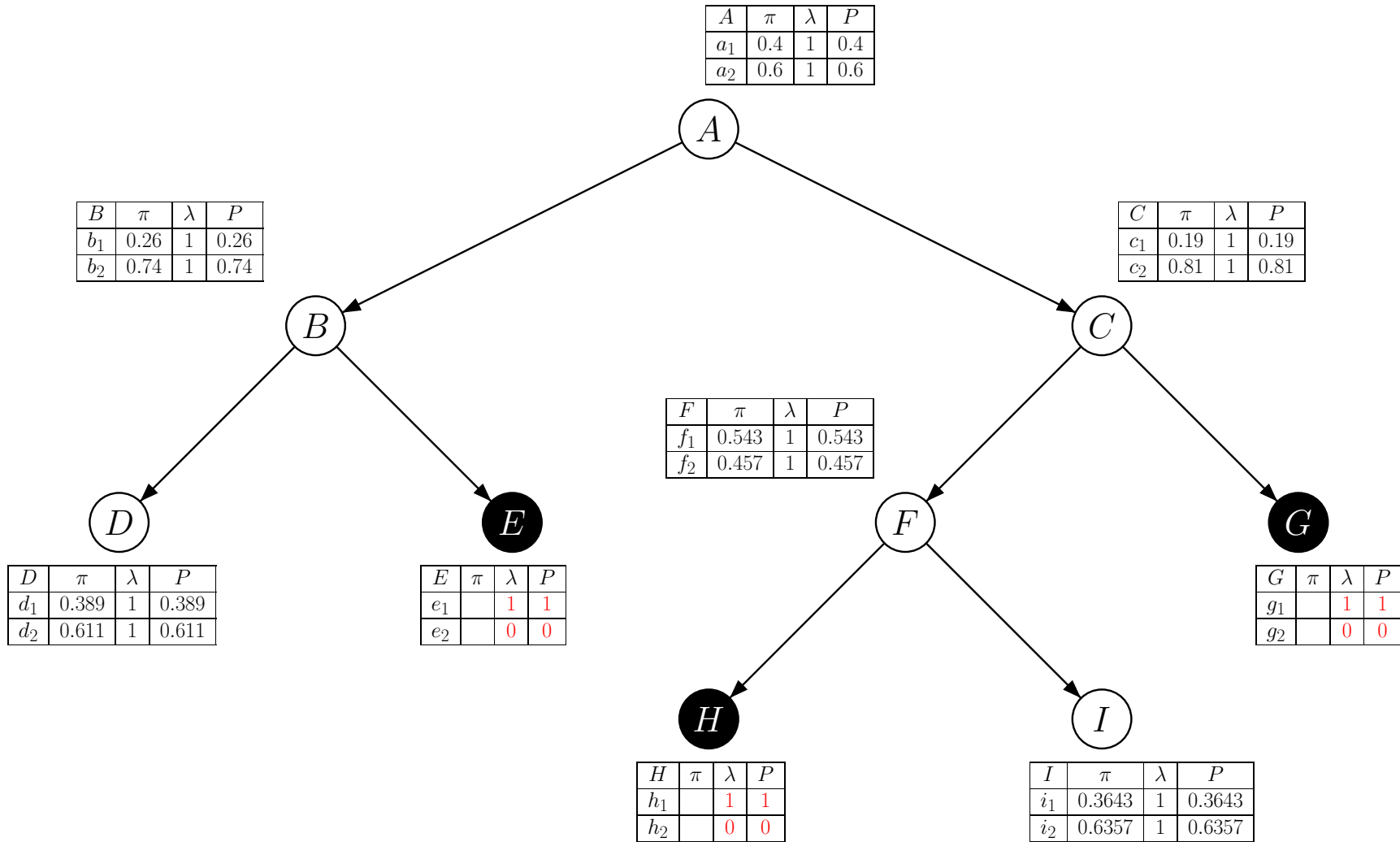
Larger Network (1): Parameters



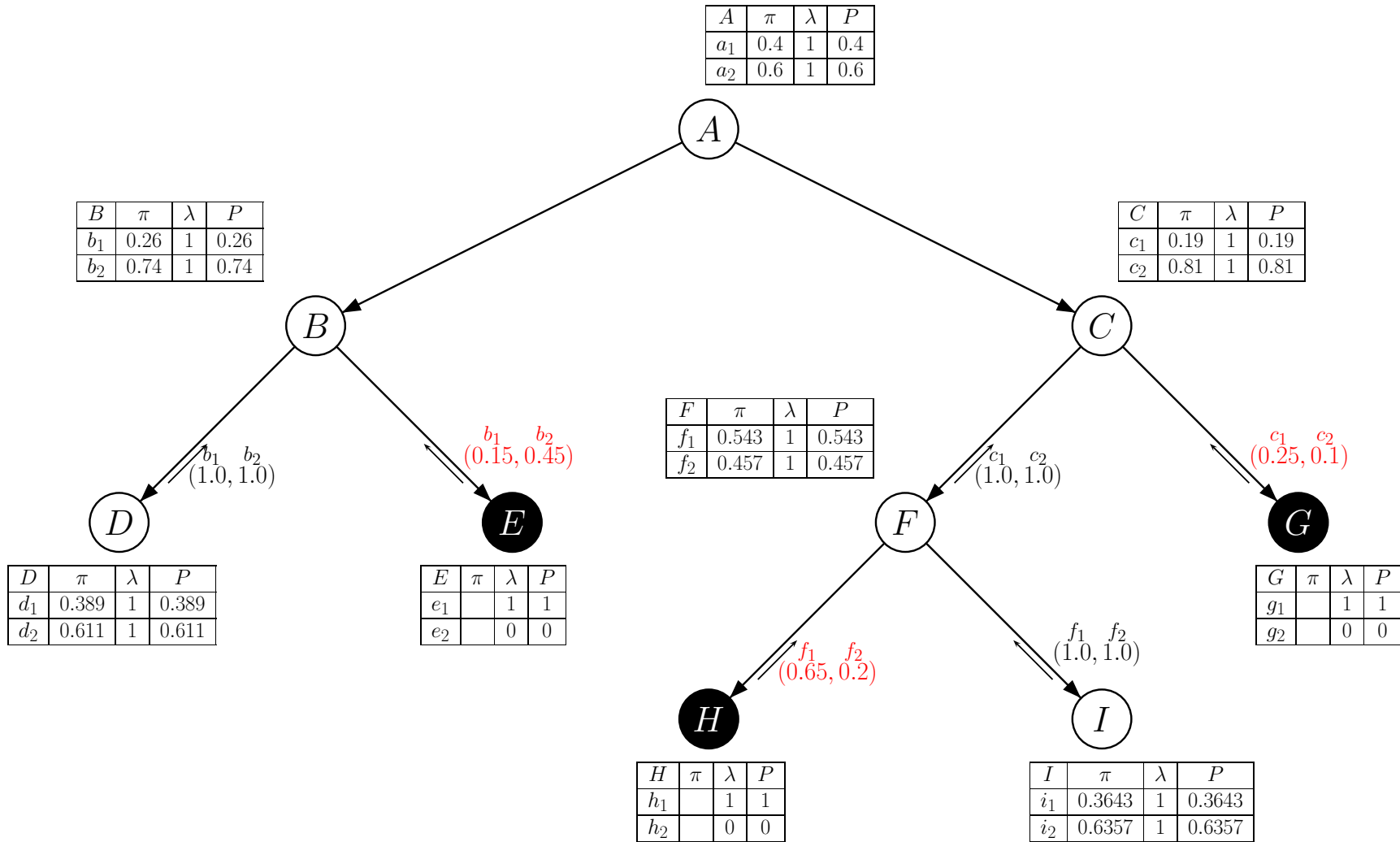
Larger Network (2): After Initialization



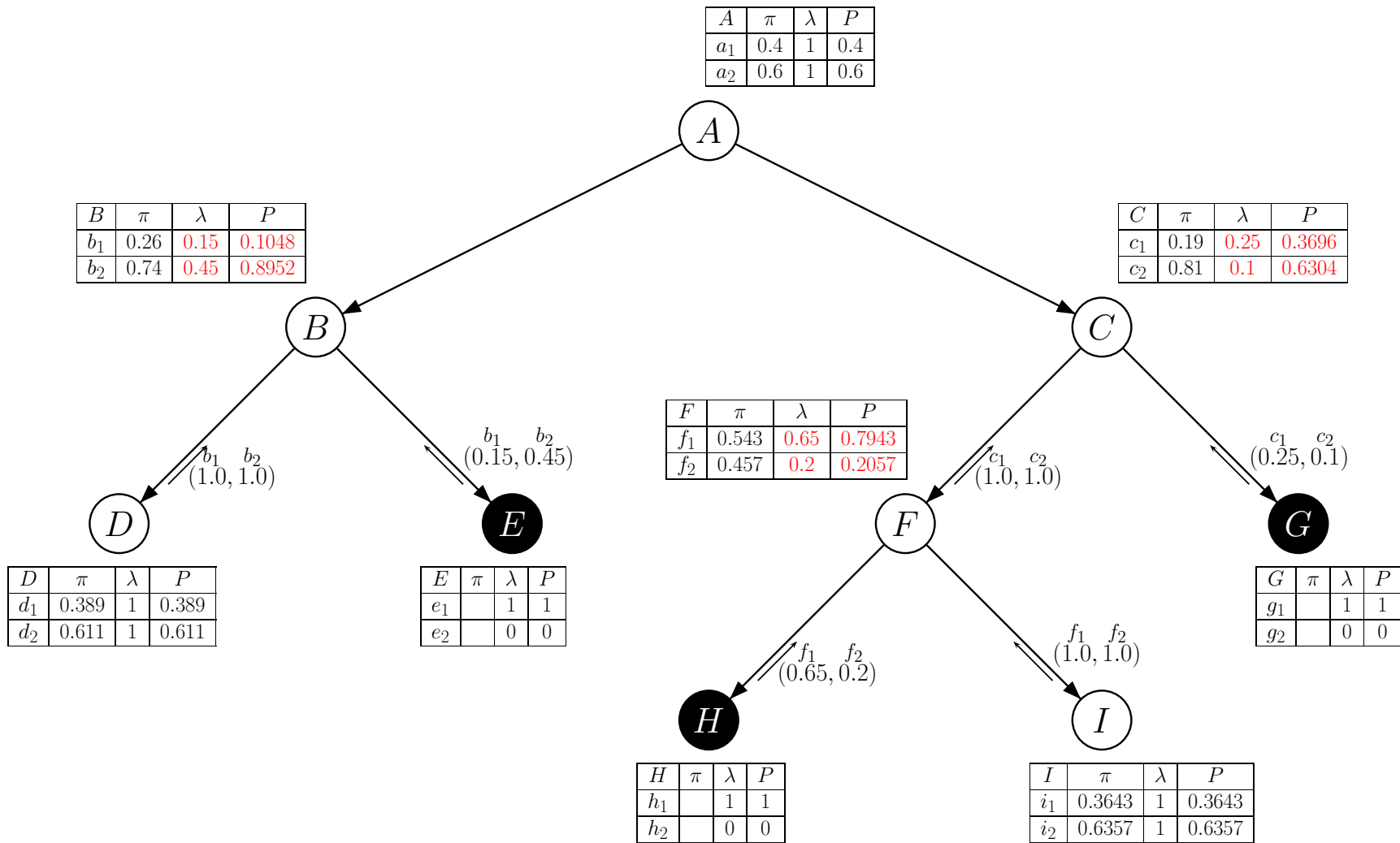
Larger Network (3): Set Evidence e_1, g_1, h_1



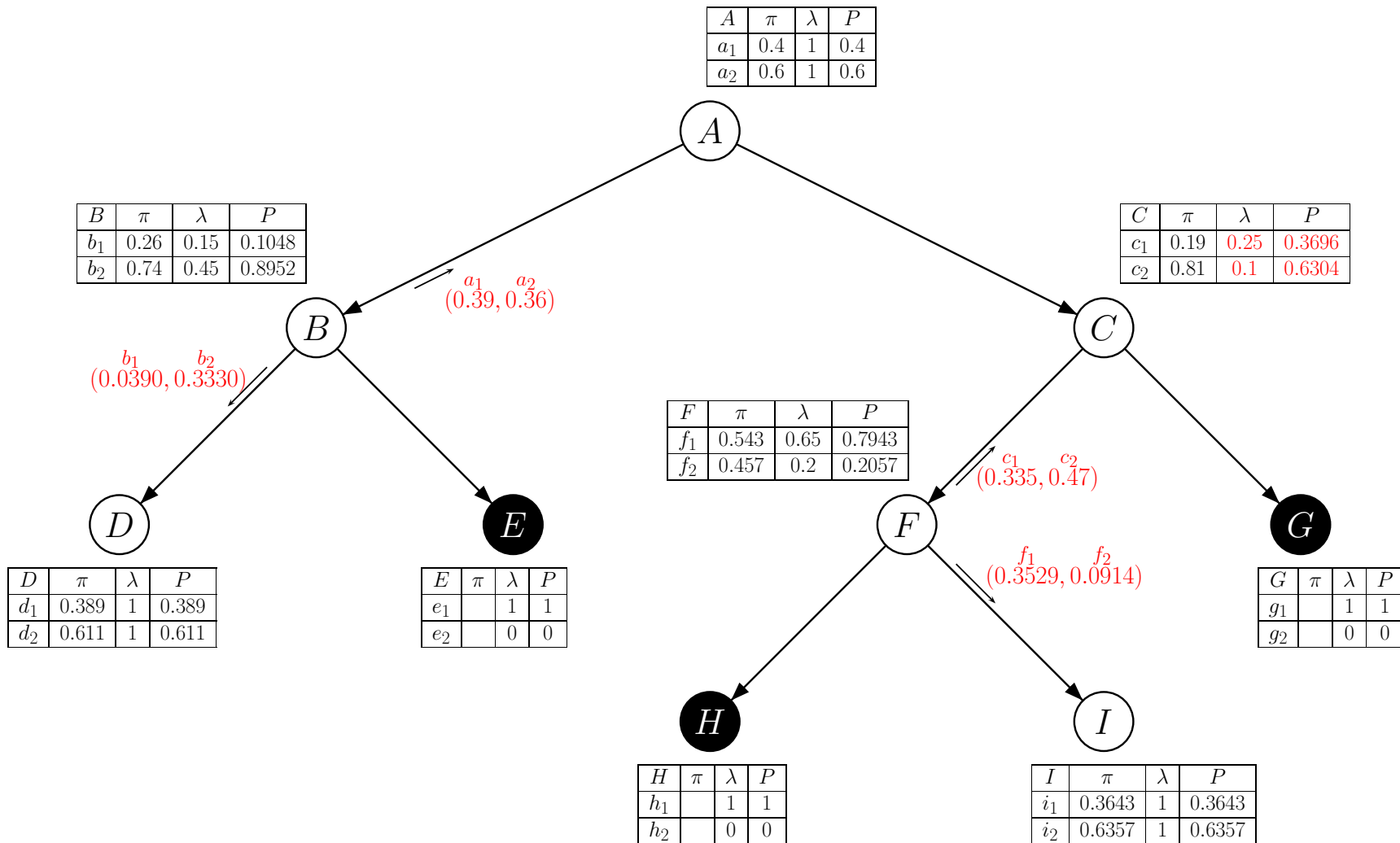
Larger Network (4): Propagate Evidence



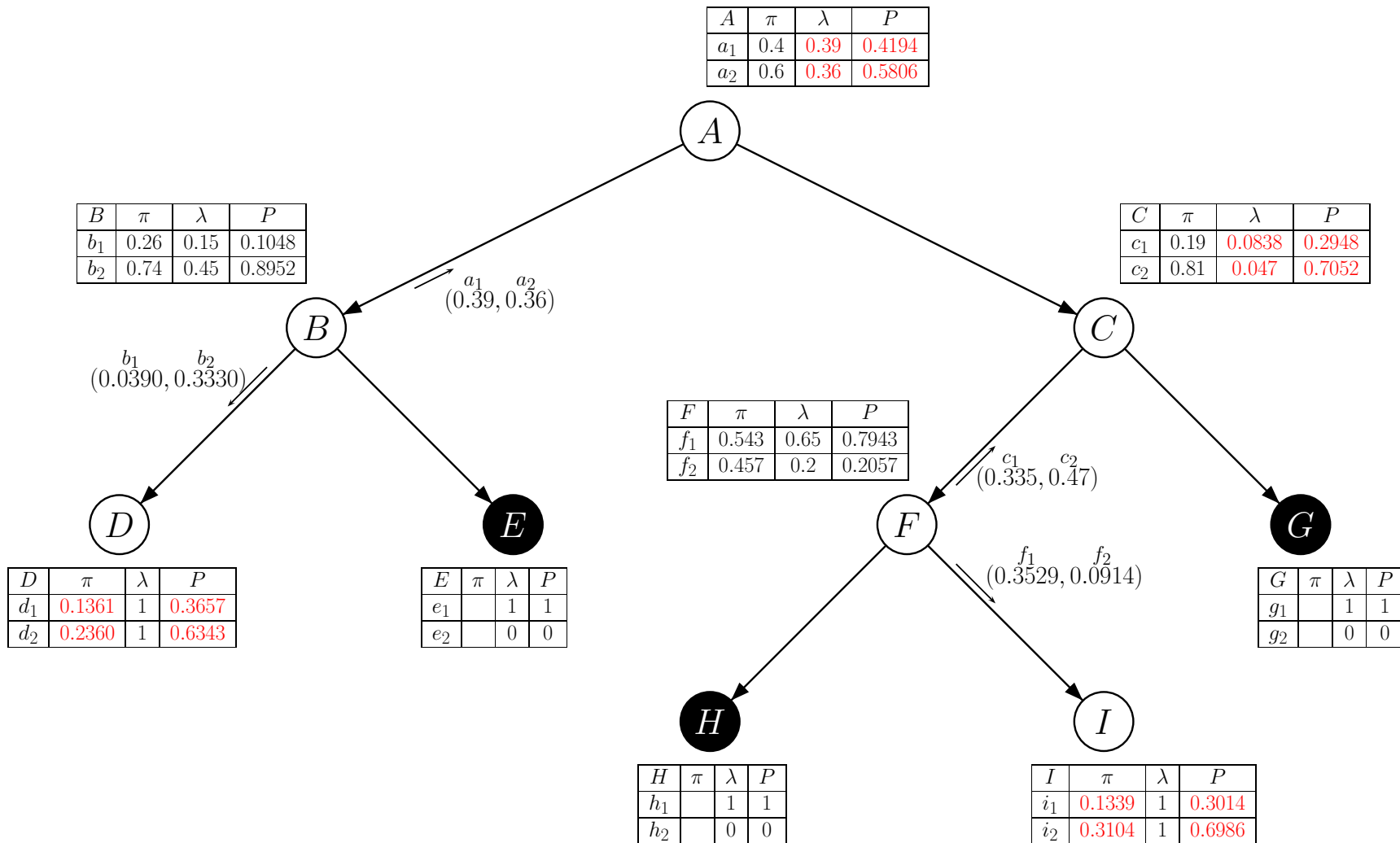
Larger Network (5): Propagate Evidence, cont.



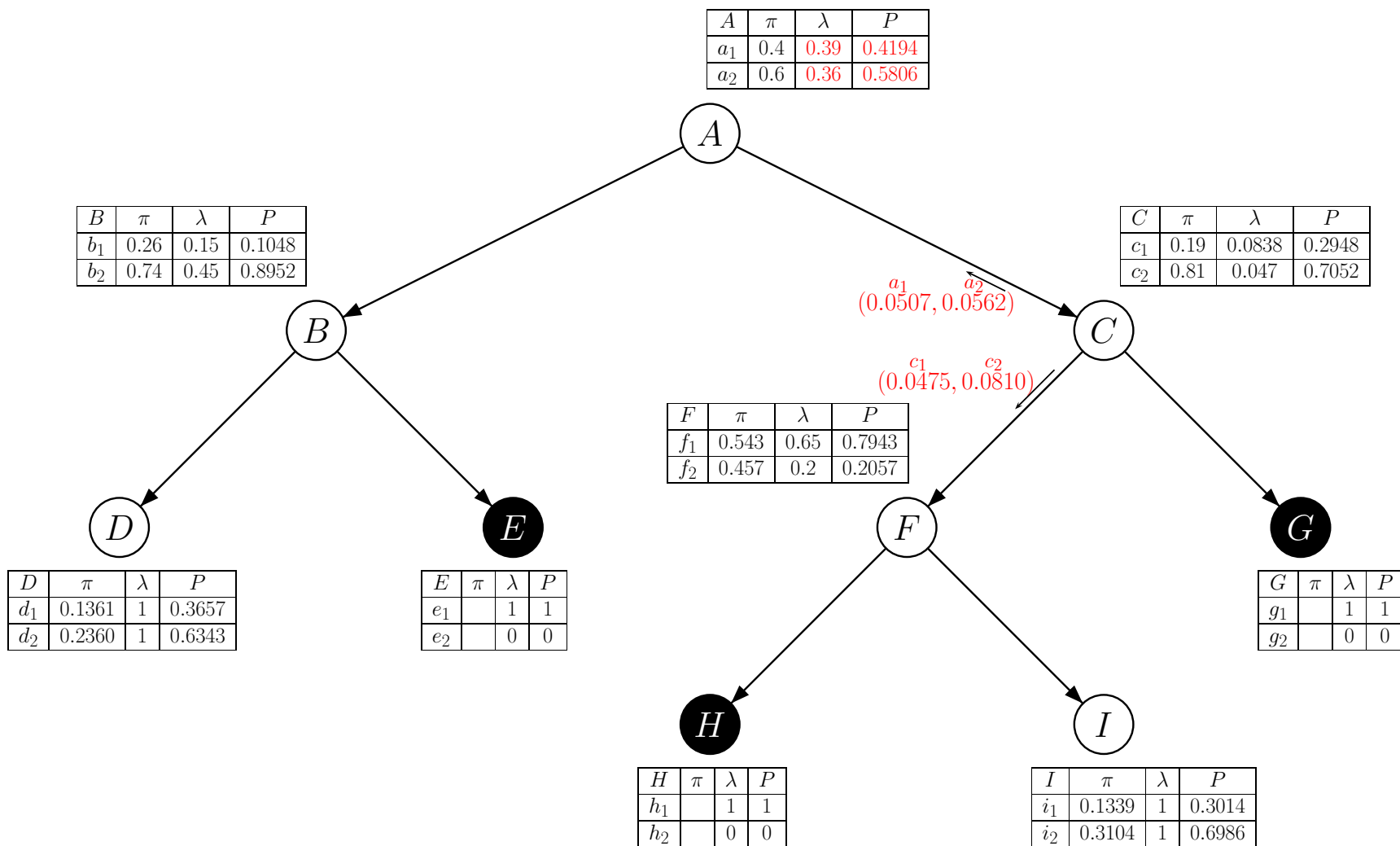
Larger Network (6): Propagate Evidence, cont.



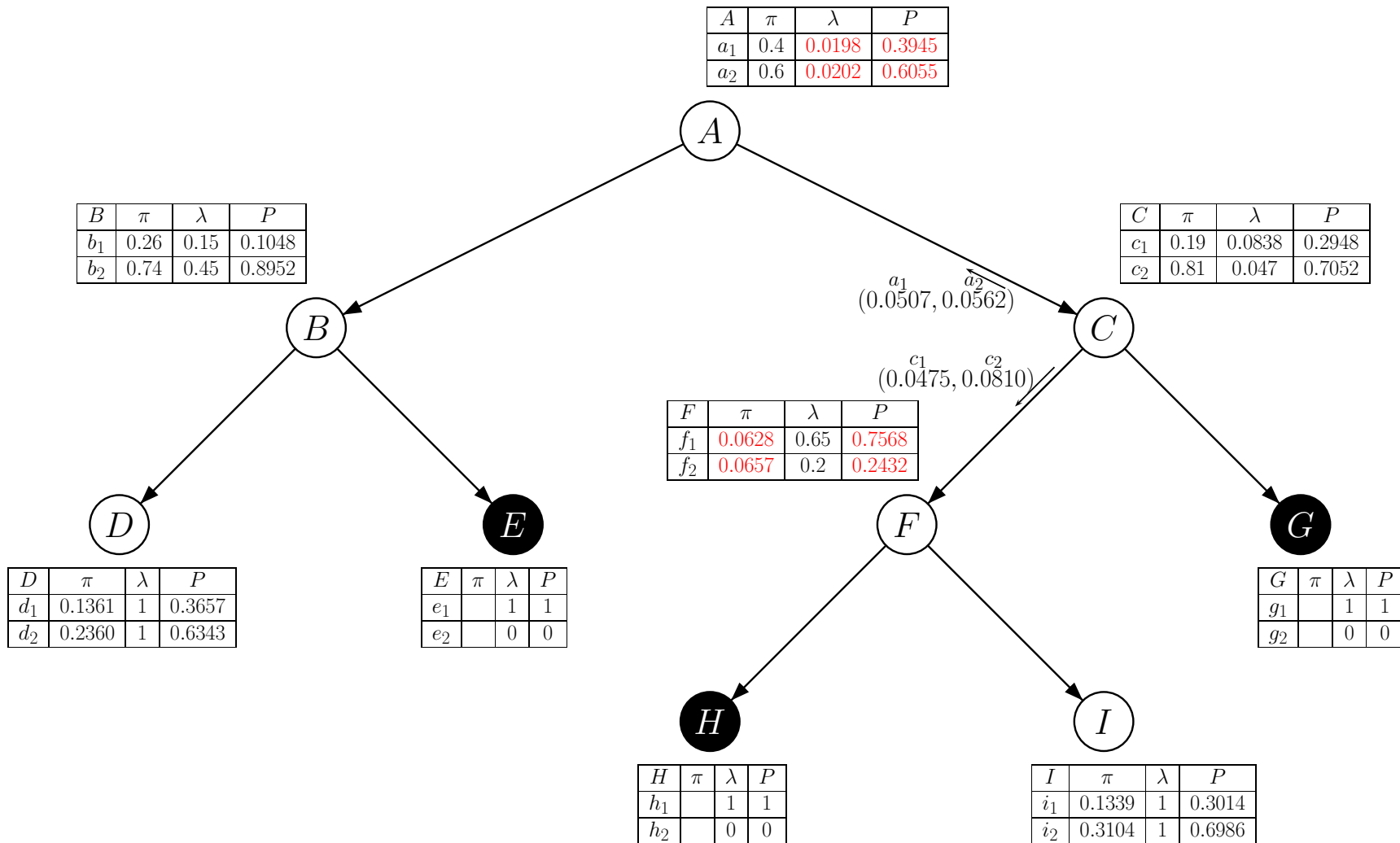
Larger Network (7): Propagate Evidence, cont.



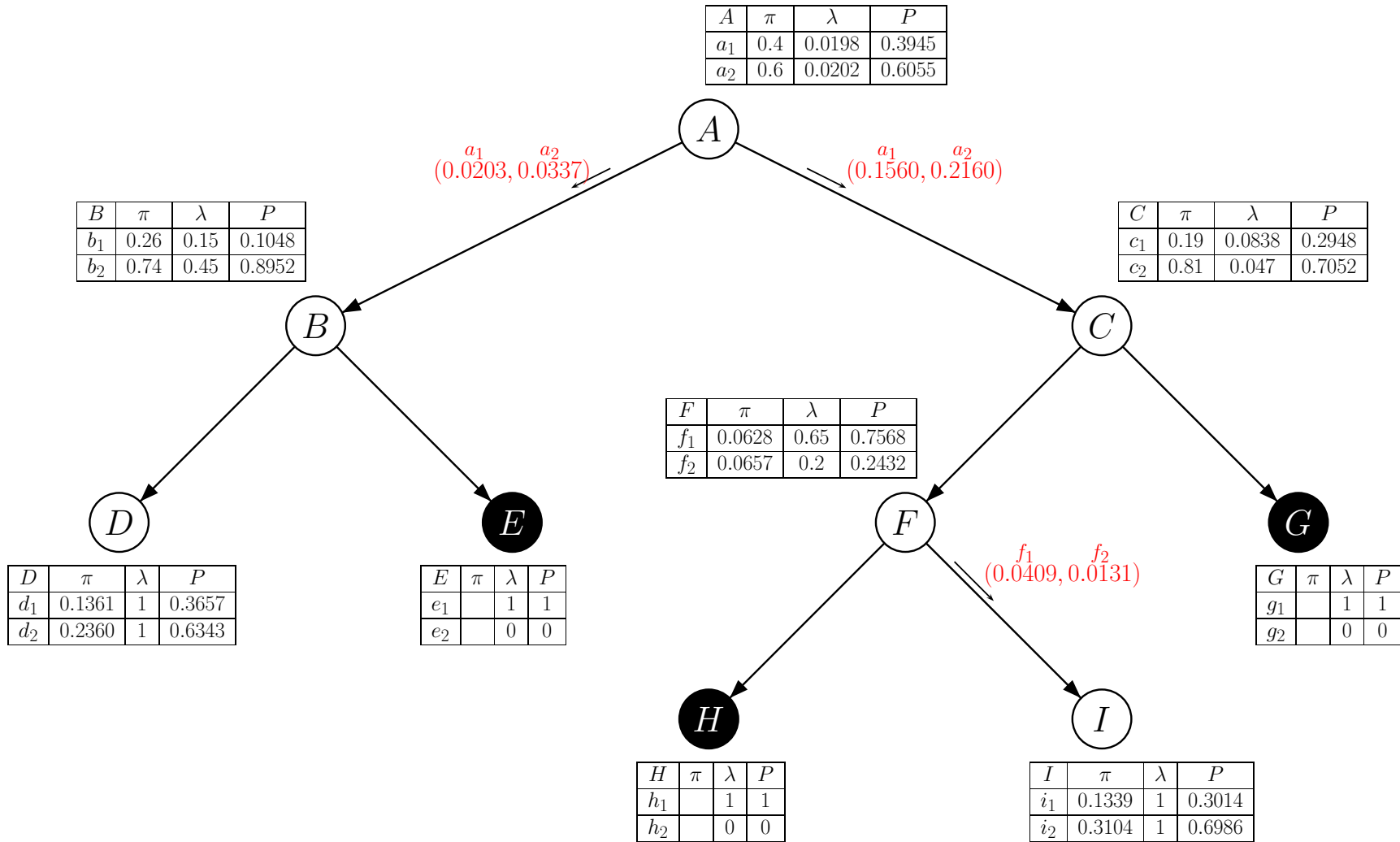
Larger Network (8): Propagate Evidence, cont.



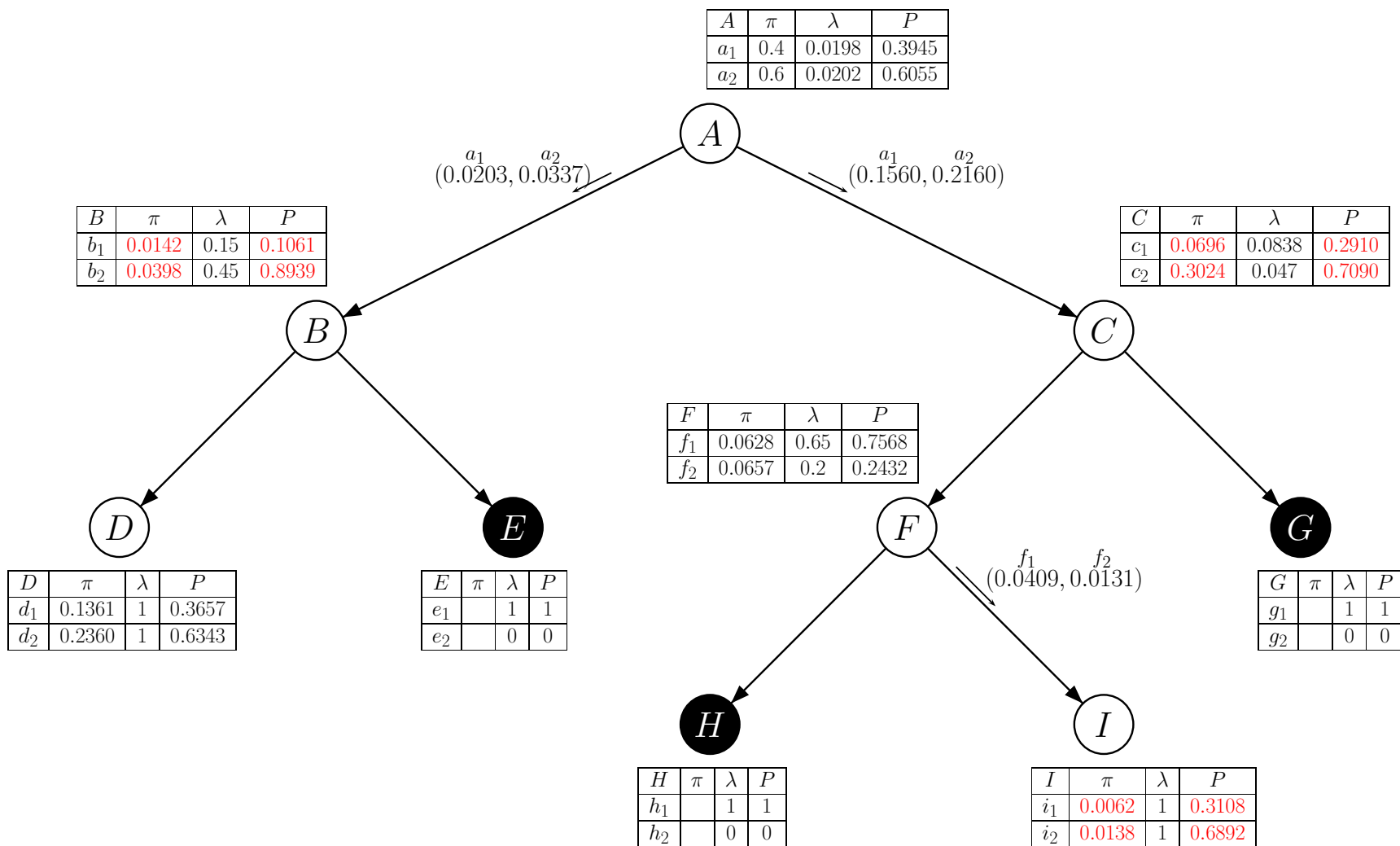
Larger Network (9): Propagate Evidence, cont.



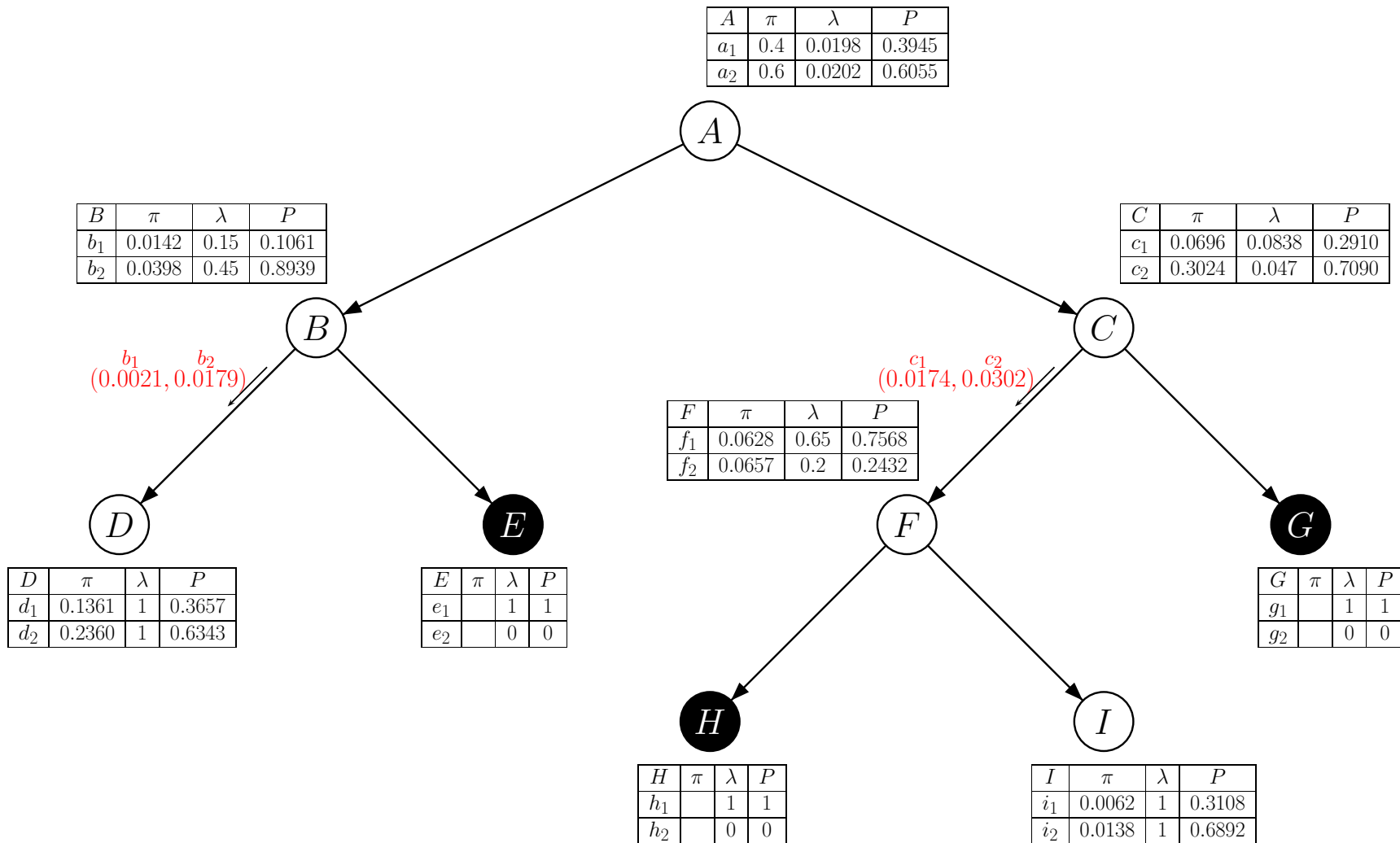
Larger Network (10): Propagate Evidence, cont.



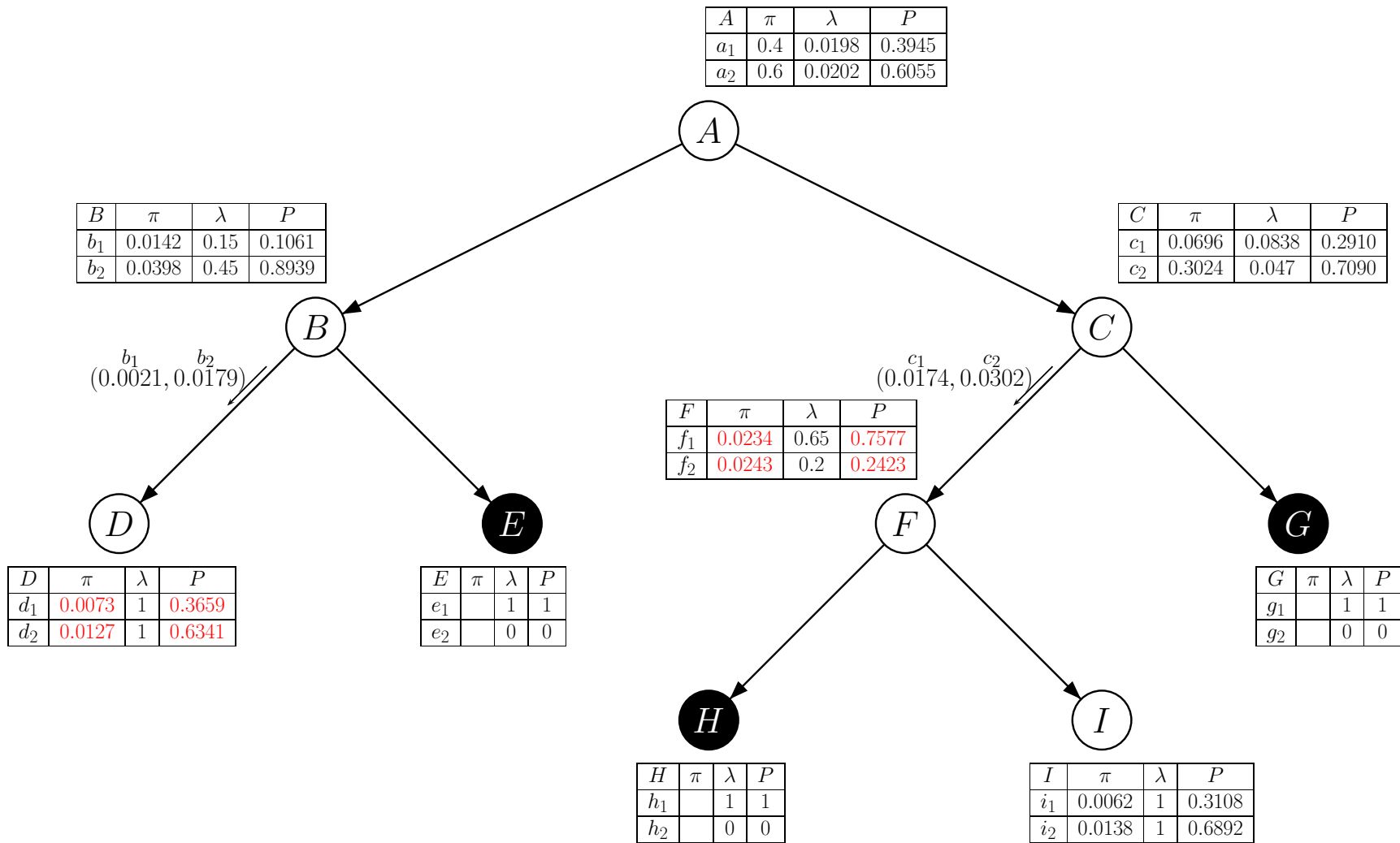
Larger Network (11): Propagate Evidence, cont.



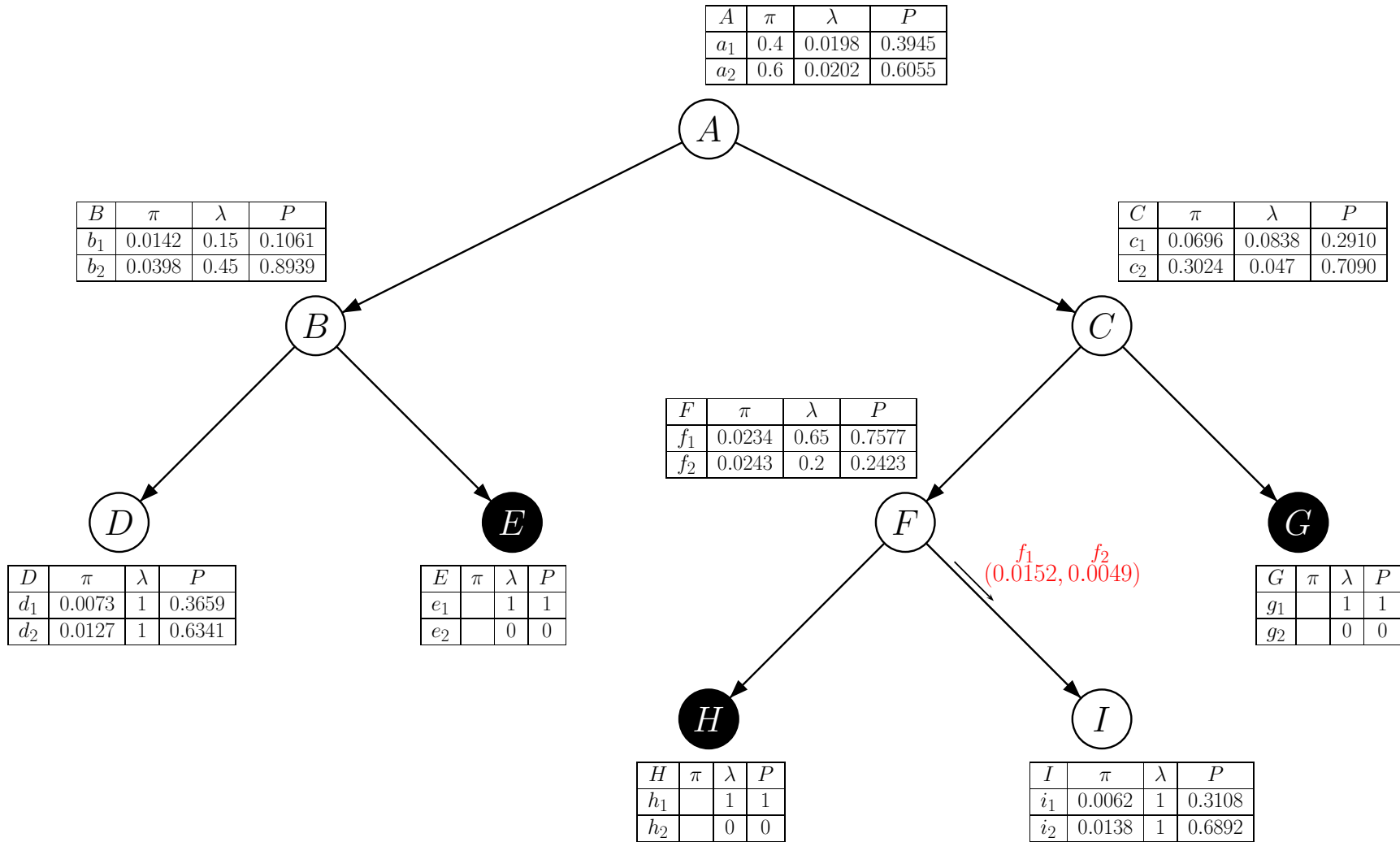
Larger Network (12): Propagate Evidence, cont.



Larger Network (13): Propagate Evidence, cont.



Larger Network (14): Propagate Evidence, cont.



Larger Network (15): Finished

