

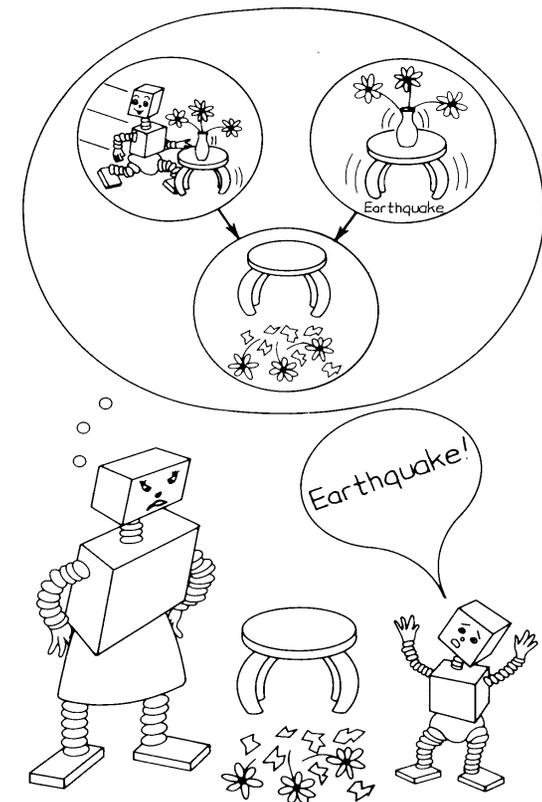
Modelos Gráficos Probabilistas

L. Enrique Sucar

INAOE

Sesión 10: Redes Bayesianas – Inferencia

1era parte



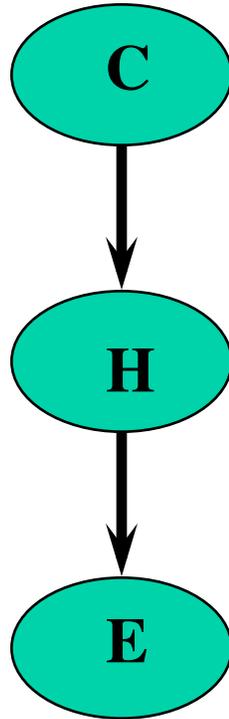
Probabilistic Reasoning in a Causal Network

Inferencia en Redes Bayesianas

1era Parte

- Introducción
- Clases de algoritmos
- Propagación en árboles
- Propagación en poli-árboles
- Extensión del algoritmo de árboles a redes multi-conectadas (Loopy)
- Abducción

Inferencia bayesiana



Causal:

$$C \rightarrow H$$

Evidencial:

$$E \rightarrow H$$

Mixta:

$$C, E \rightarrow H$$

Tipos de Técnicas

- Calcular probabilidades posteriores:
 - Una variable, cualquier estructura: algoritmo de eliminación (*variable elimination*)
 - Todas las variable, estructuras sencillamente conectadas (árboles, poliárboles): propagación
 - Todas las variables, cualquier estructura:
 - Agrupamiento (*junction tree*)
 - Condicionamiento
 - Técnicas aproximadas (*Loopy BP*, simulación estocástica)

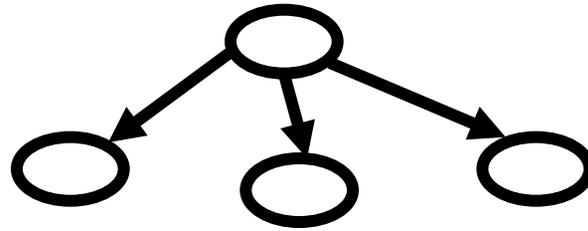
Tipos de Técnicas

- Obtener variable(s) de mayor probabilidad dada cierta evidencia – abducción:
 - Abducción total
 - Abducción parcial

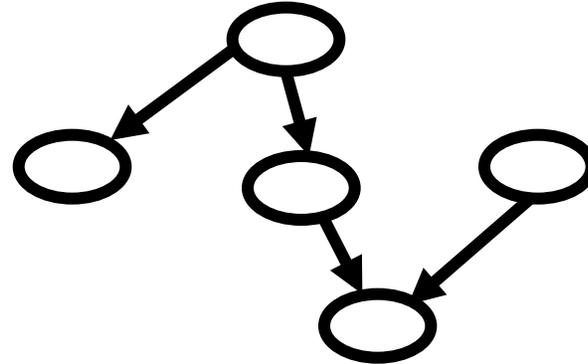
Tipos de estructuras

- Sencillamente conectadas

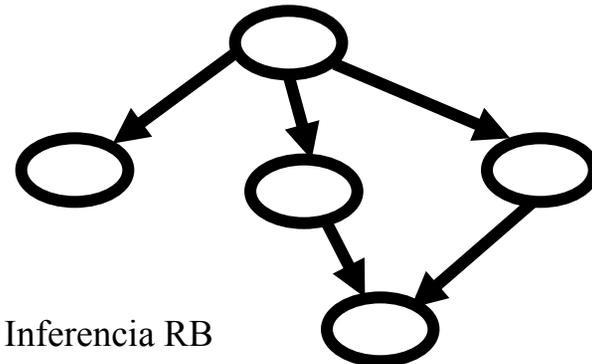
- Árboles



- Poliárboles



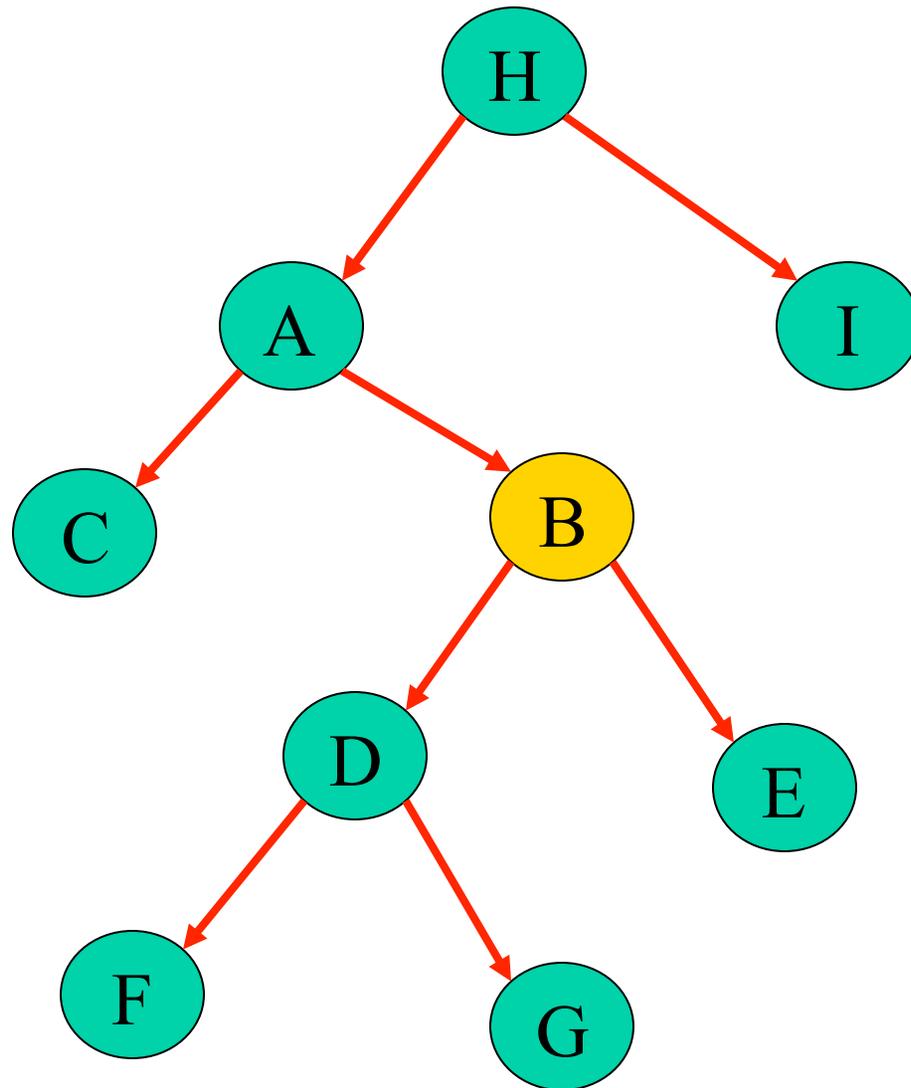
- Multiconectadas



Propagación en Árboles

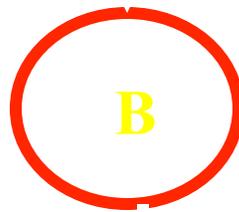
Cada nodo corresponde a una variable discreta, B (B_1, B_2, \dots, B_m) con su respectiva matriz de probabilidad condicional, $P(B|A) = P(B_j | A_i)$

Propagación en Árboles

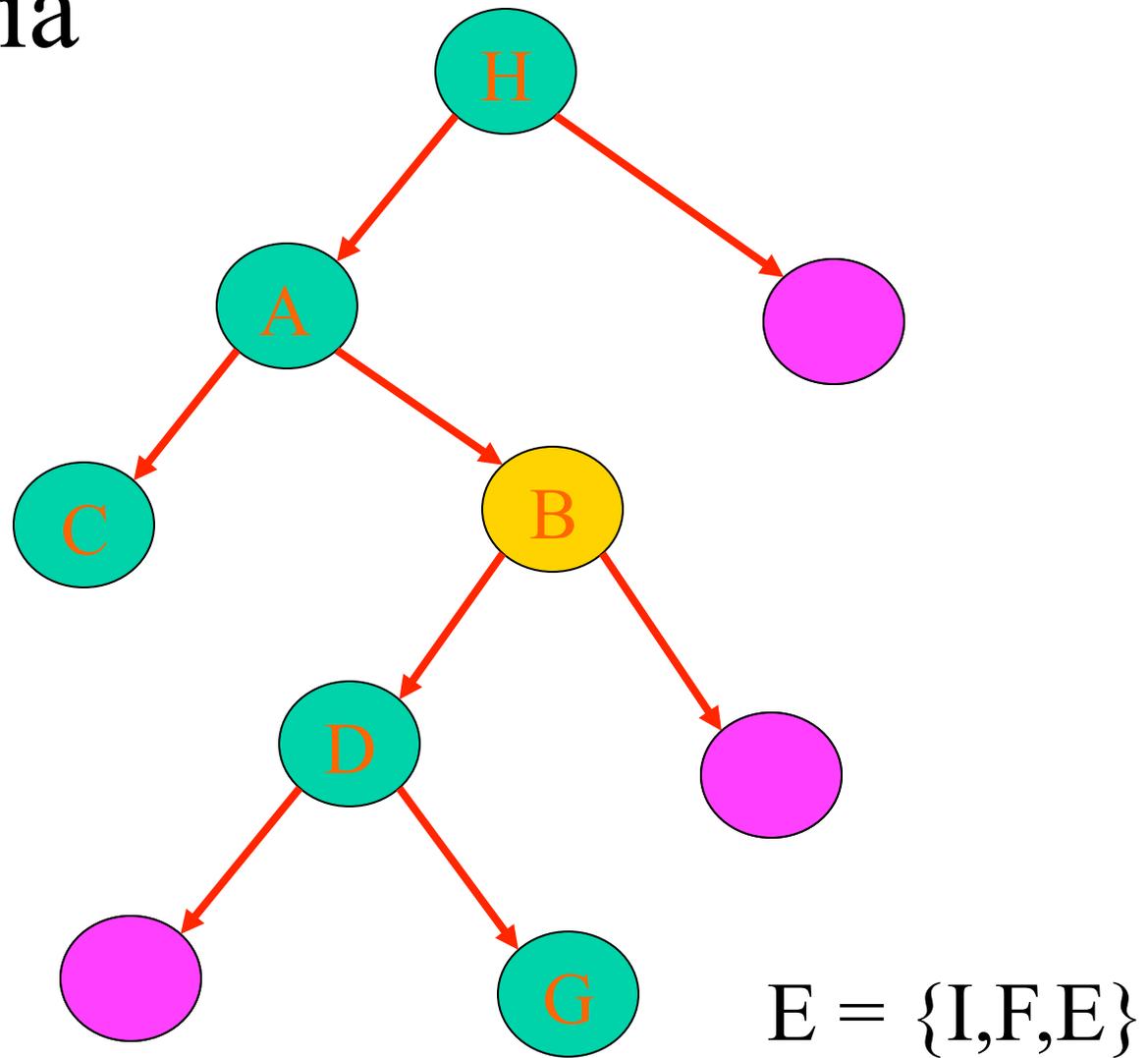


Dada cierta evidencia E -representada por la instanciación de ciertas variables- la probabilidad posterior de cualquier variable B , por el teorema de Bayes:

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



Evidencia



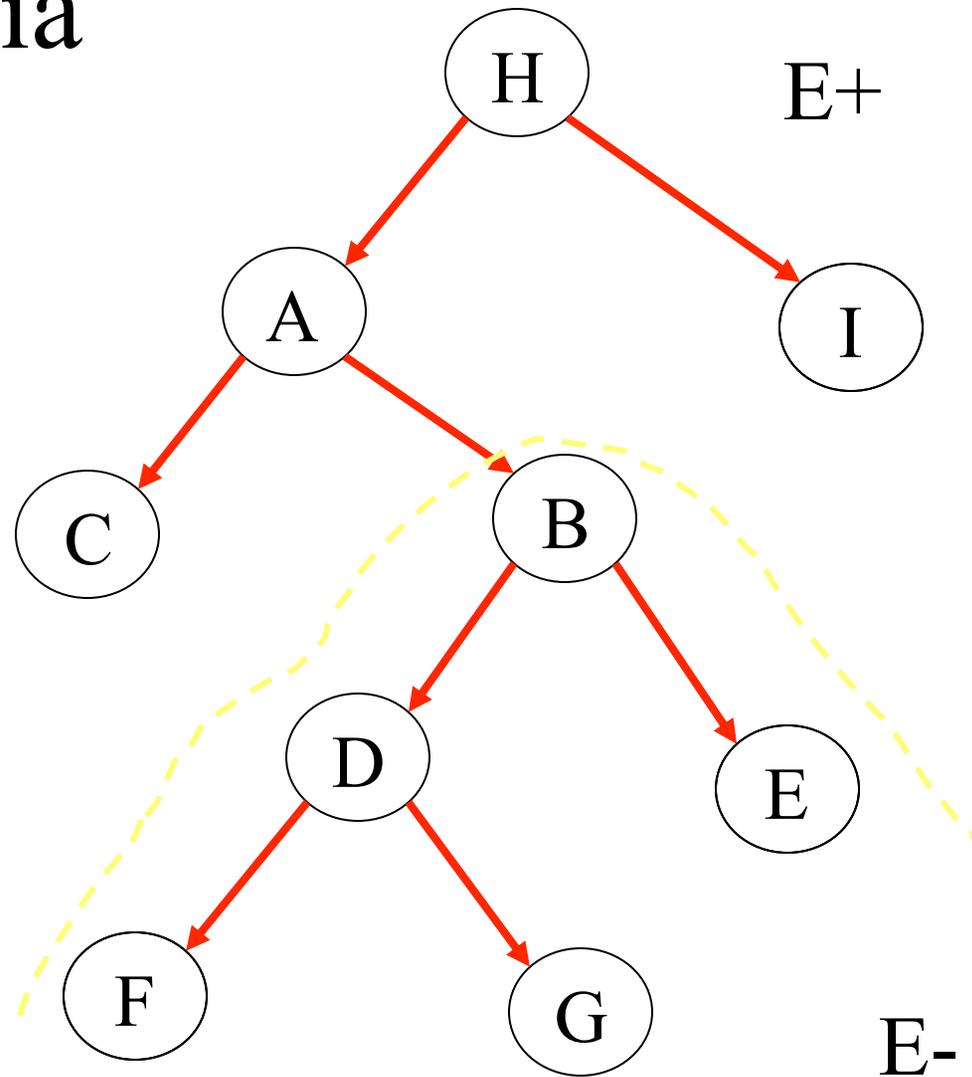
Evidencia

Ya que la estructura de la red es un árbol, el Nodo B la separa en dos subárboles, por lo que podemos dividir la evidencia en dos grupos:

E^- : Datos en el árbol que cuya raíz es B

E^+ : Datos en el resto del árbol

Evidencia



Entonces:

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

Pero dado que ambos son independientes y aplicando nuevamente Bayes:

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

Donde α es una constante de normalización

Definiciones:

Si definimos los siguientes términos:

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

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

Entonces:

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

Desarrollo

- En base a la ecuación anterior, se puede integrar un algoritmo distribuido para obtener la probabilidad de un nodo dada cierta evidencia
- Para ello se descompone el cálculo de cada parte:
 - Evidencia de los hijos (λ)
 - Evidencia de los demás nodos (π)

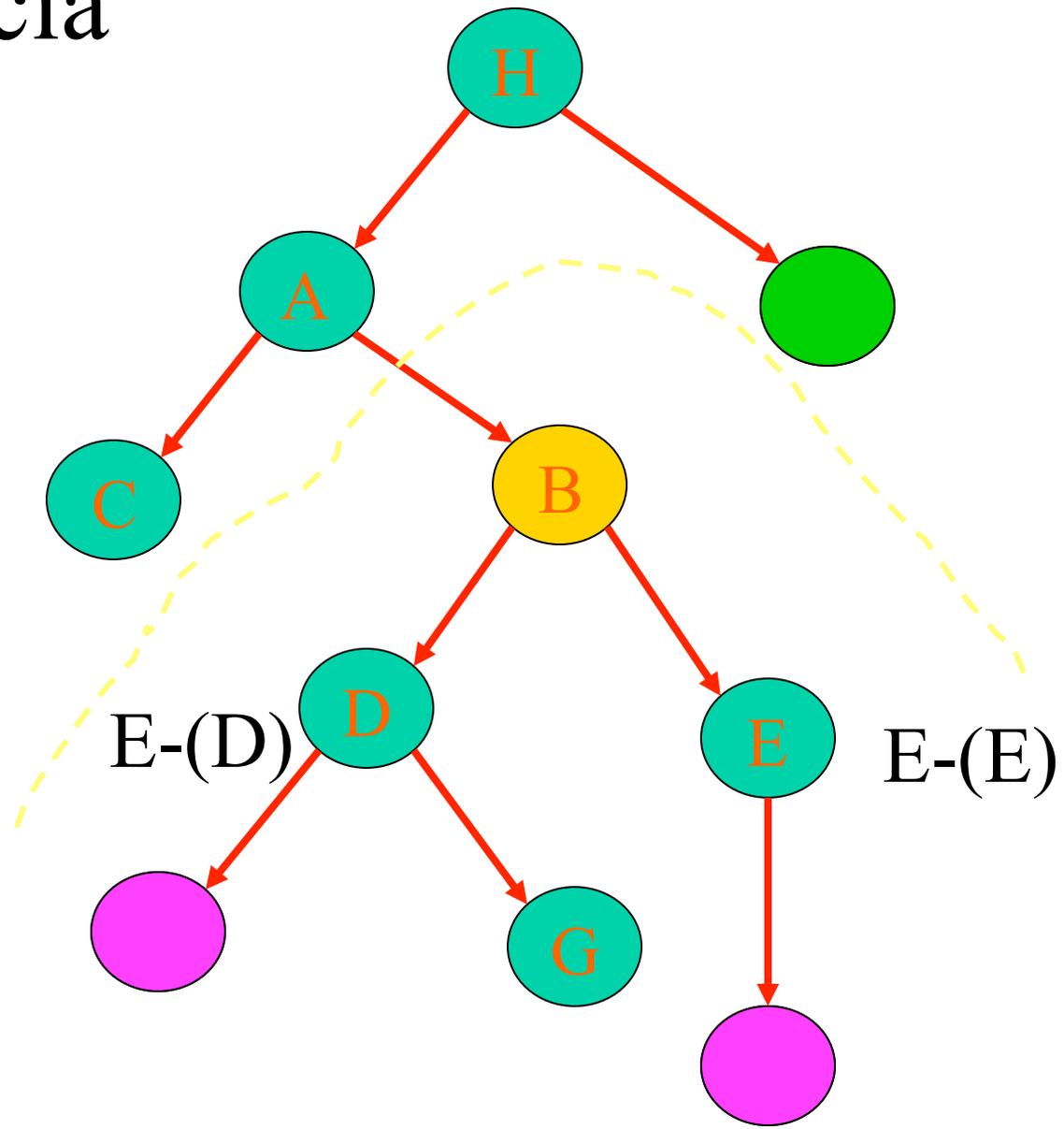
Evidencia de los hijos (λ)

- Dado que los hijos son condicionalmente independientes dado el padre:

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

- Donde E_k^- corresponde a la evidencia del subárbol del hijo k

Evidencia hijos



Evidencia de los hijos (λ)

- Condicionando respecto a los posibles valores de los hijos de B:

$$\lambda (B_i) = \prod_k \left[\sum_j P (E_k^- | B_i, S_j^k) P(S_j^k | B_i) \right]$$

- Donde S^k es el hijo k de B, y la sumatoria es sobre los valores de dicho nodo (teorema de probabilidad total)

Evidencia de los hijos (λ)

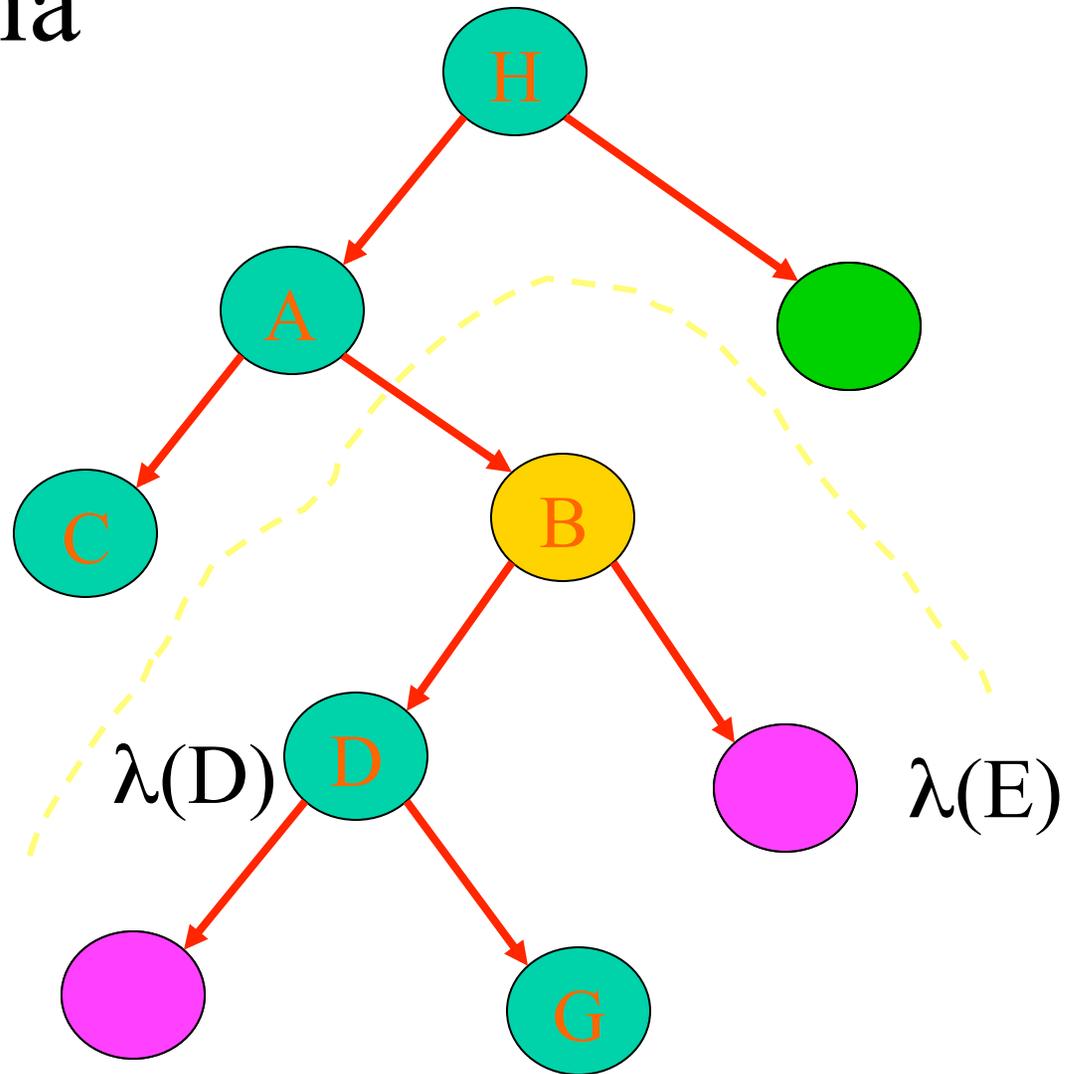
- Dado que B es condicionalmente independiente de la evidencia dados sus hijos:

$$\lambda (B_i) = \prod_k [\sum_j P (E_k \cdot | S_j^k) P(S_j^k | B_i)]$$

- Substituyendo la definición de λ :

$$\lambda (B_i) = \prod_k [\sum_j P(S_j^k | B_i) \lambda (S_j^k)]$$

Evidencia hijos



Evidencia de los hijos (λ)

- Recordando que λ es un vector (un valor por cada posible valor de B), lo podemos ver en forma matricial:

$$\lambda = \lambda P(S | B)$$

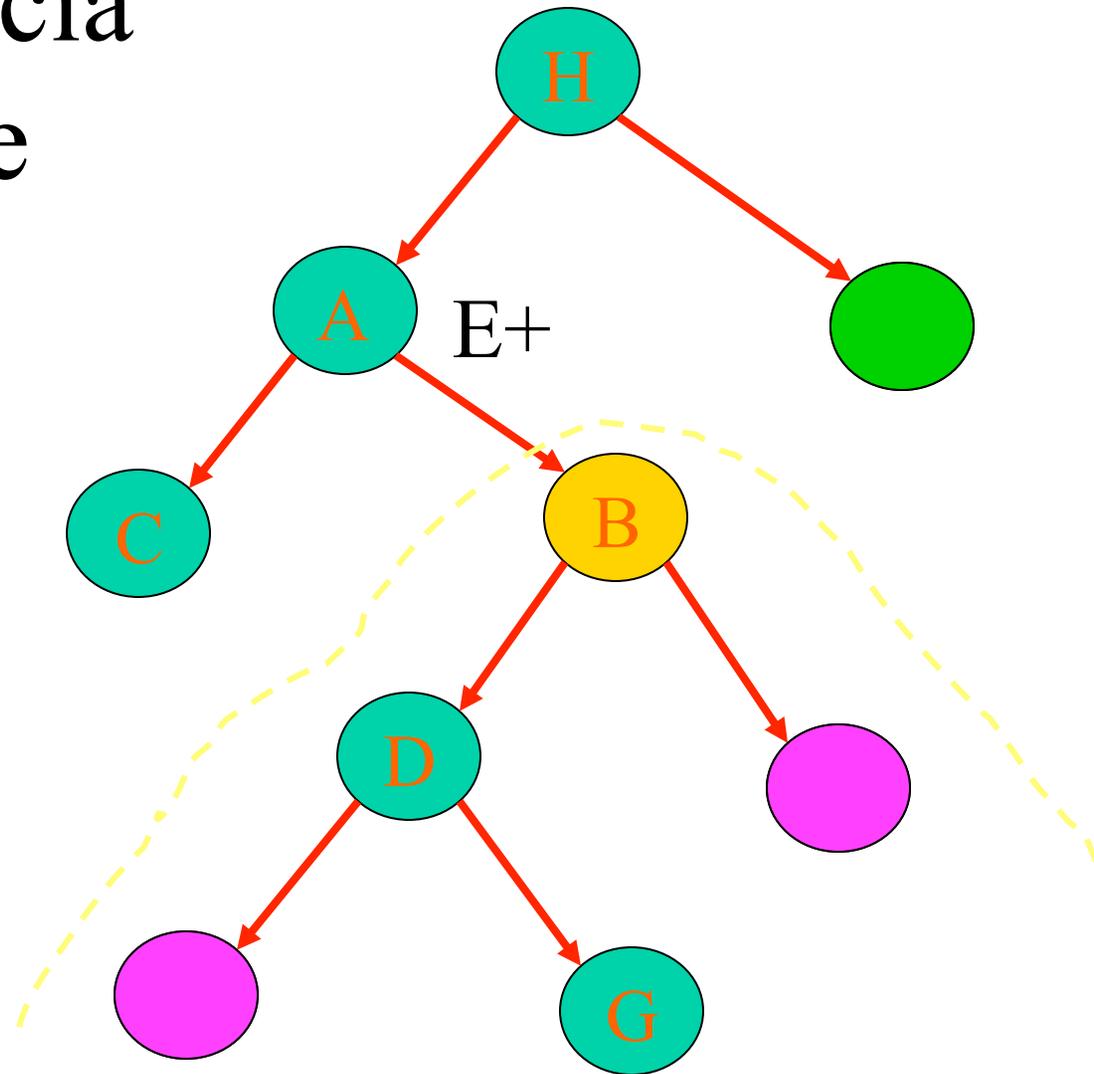
Evidencia de los demás nodos (π)

- Condicionando sobre los diferentes valores del nodo padre (A):

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

- Donde A_j corresponde a los diferentes valores del nodo padre de B

Evidencia padre



Evidencia de los demás nodos (π)

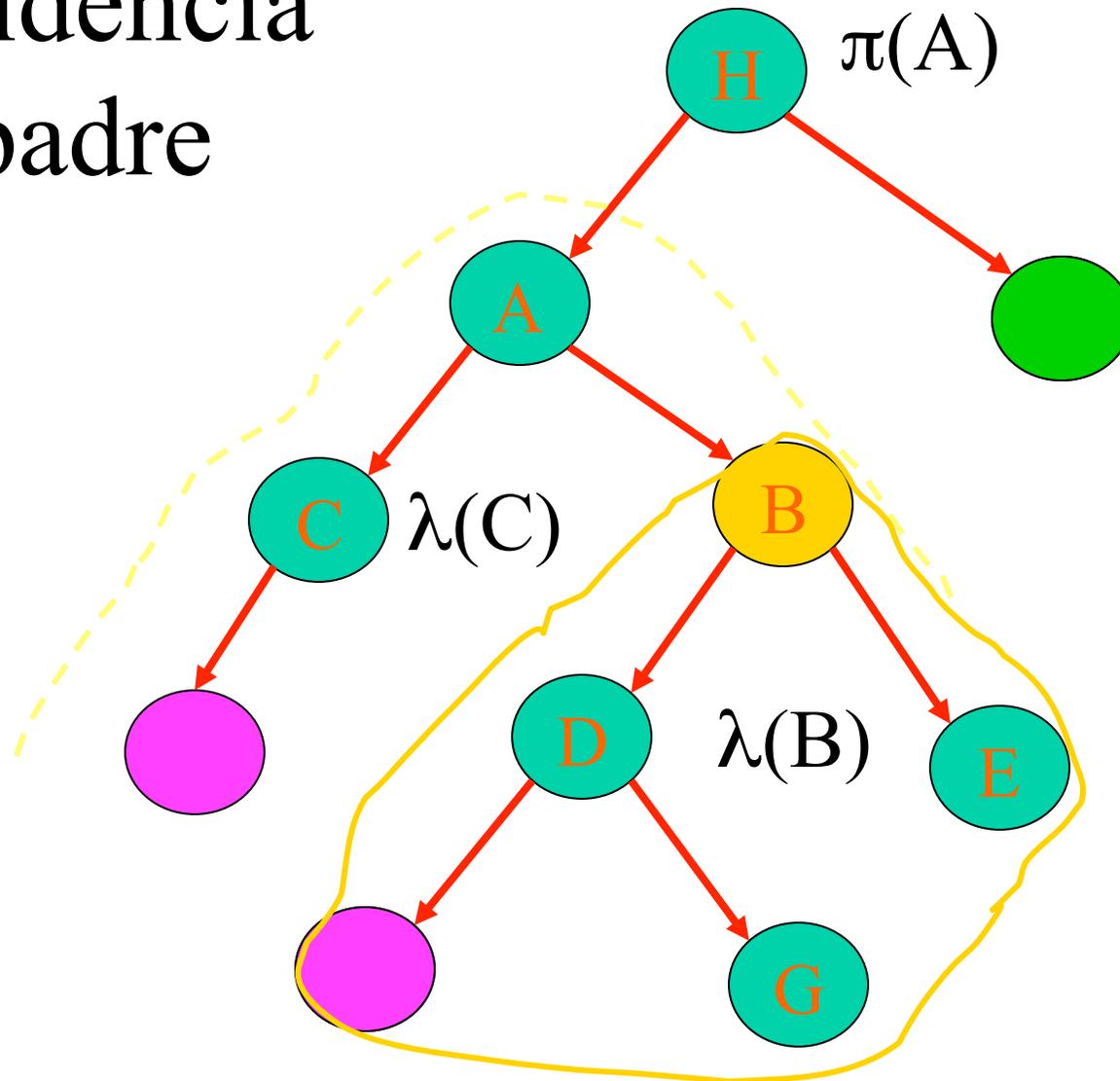
- Dado que B es independiente de la evidencia “arriba” de A, dado A:

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

- La $P(A_j | E^+)$ corresponde a la P posterior de A dada toda la evidencia excepto B y sus hijos, por lo que se puede escribir como:

$$P(A_j | E^+) = \alpha \pi (A_i) \prod_{k \in B} \lambda_k (A_i)$$

Evidencia padre



Evidencia de los demás nodos (π)

- Substituyendo $\mathbf{P}(\mathbf{A}_j | \mathbf{E}^+)$ en la ecuación de π :

$$\pi (\mathbf{B}_i) = \sum_j \mathbf{P} (\mathbf{B}_i | \mathbf{A}_j) \left[\alpha \pi (\mathbf{A}_i) \prod_{k \in \mathbf{B}} \lambda_k (\mathbf{A}_i) \right]$$

- De forma que se obtiene combinando la π de del nodo padre con la λ de los demás hijos

Evidencia de los demás nodos (π)

- Dado que también π es un vector, lo podemos ver en forma matricial (donde P_A es el producto de la evidencia de padre y otros hijos):

$$\pi = P(B | A) P_A$$

Algoritmo

- Mediante estas ecuaciones se integra un algoritmo de propagación de probabilidades en árboles.
- Cada nodo guarda los valores de los vectores π y λ , así como su matriz de probabilidad condicional (CPT), P .
- La propagación se hace por un mecanismo de paso de mensajes, en donde cada nodo envía los mensajes correspondientes a su padre e hijos

Mensaje al padre (hacia arriba) –
nodo B a su padre A:

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

Mensaje a los hijos (hacia abajo) -
nodo B a su hijo S_k :

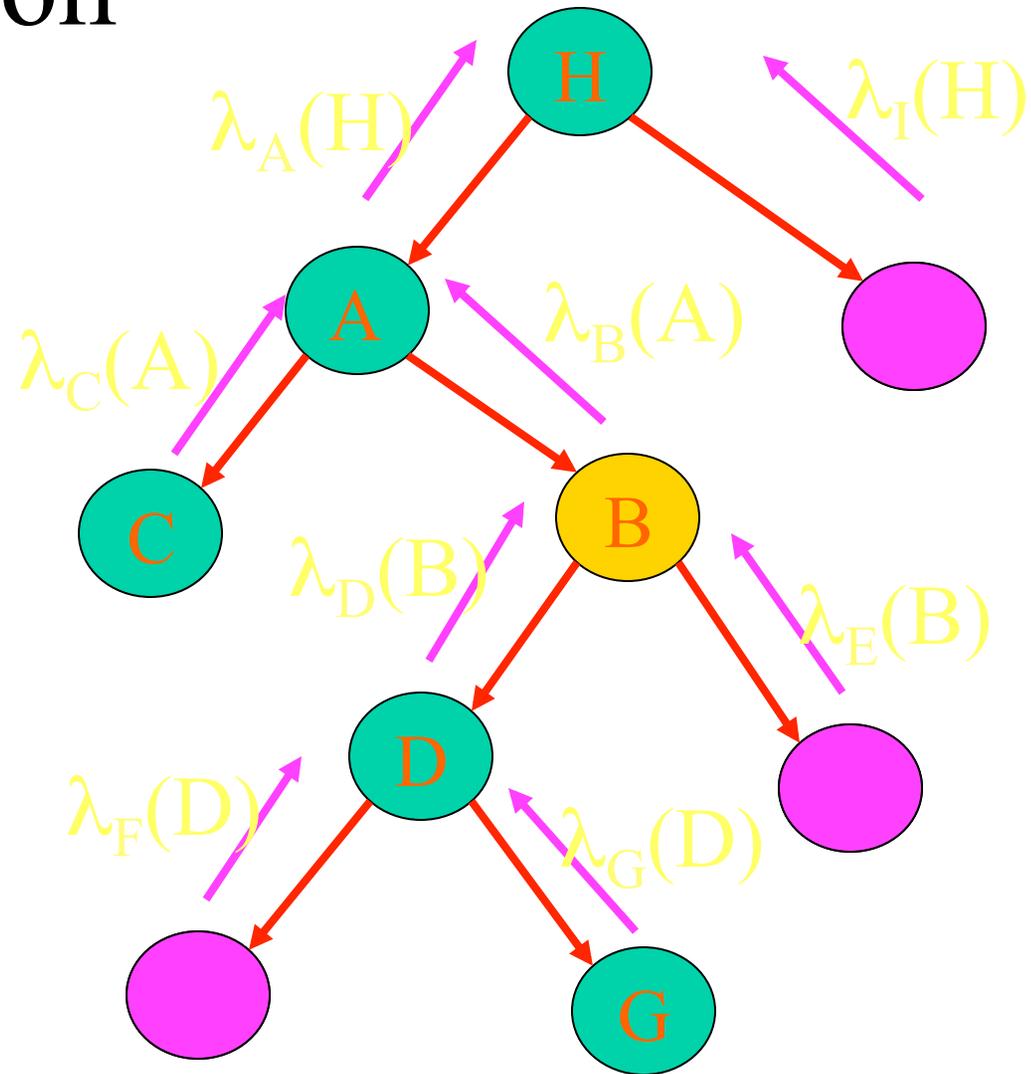
$$\pi_k(B_i) = \alpha \pi(B_j) \prod_{I \neq k} \lambda_I(B_j)$$

Algoritmo

- Al instanciarse ciertos nodos, éstos envían mensajes a sus padres e hijos, y se propagan hasta a llegar a la raíz u hojas, o hasta encontrar un nodo instanciado.
- Así que la propagación se hace en un solo paso, en un tiempo proporcional al diámetro de la red.

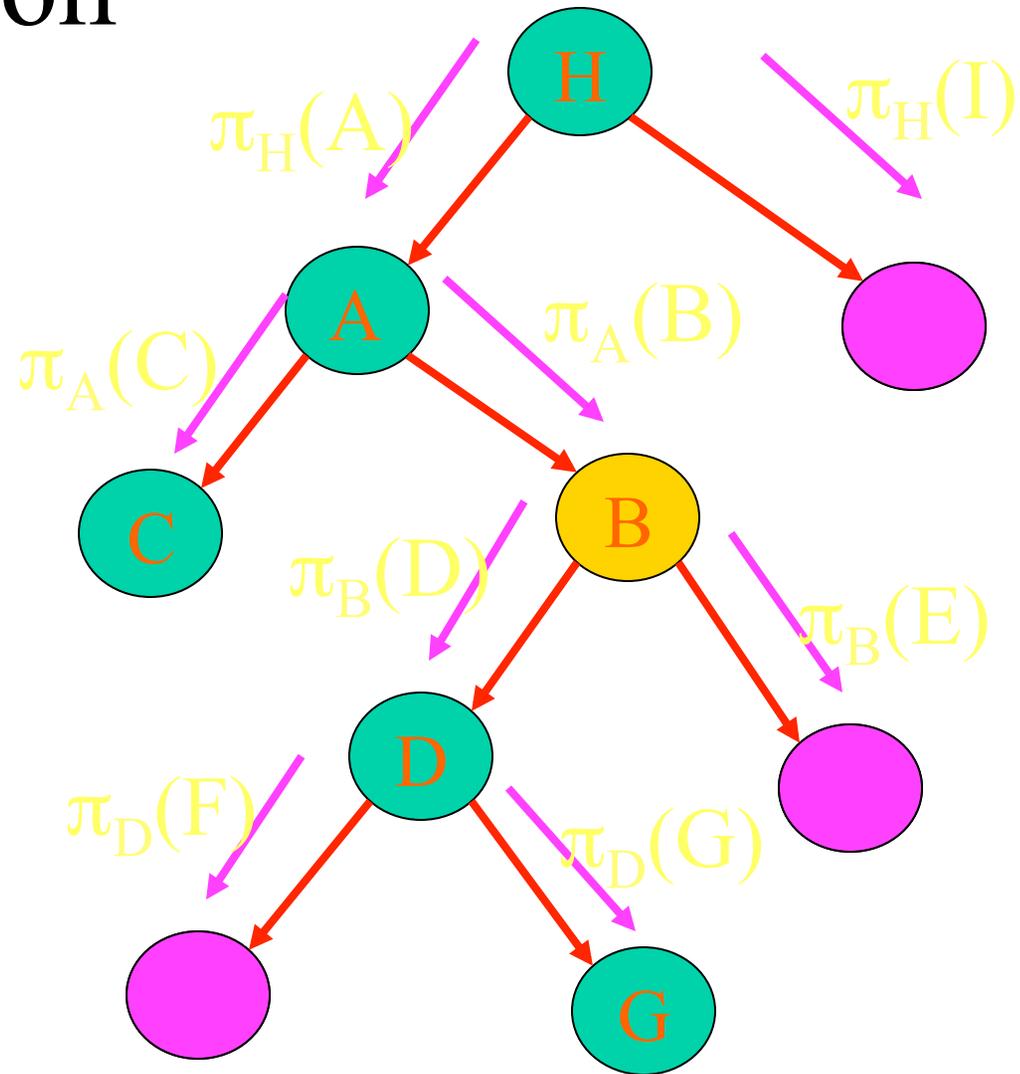
Propagación

λ



Propagación

π



Condiciones Iniciales

- Nodos hoja no conocidos:

$$\lambda (B_i) = [1, 1, \dots]$$

- Nodos asignados (conocidos):

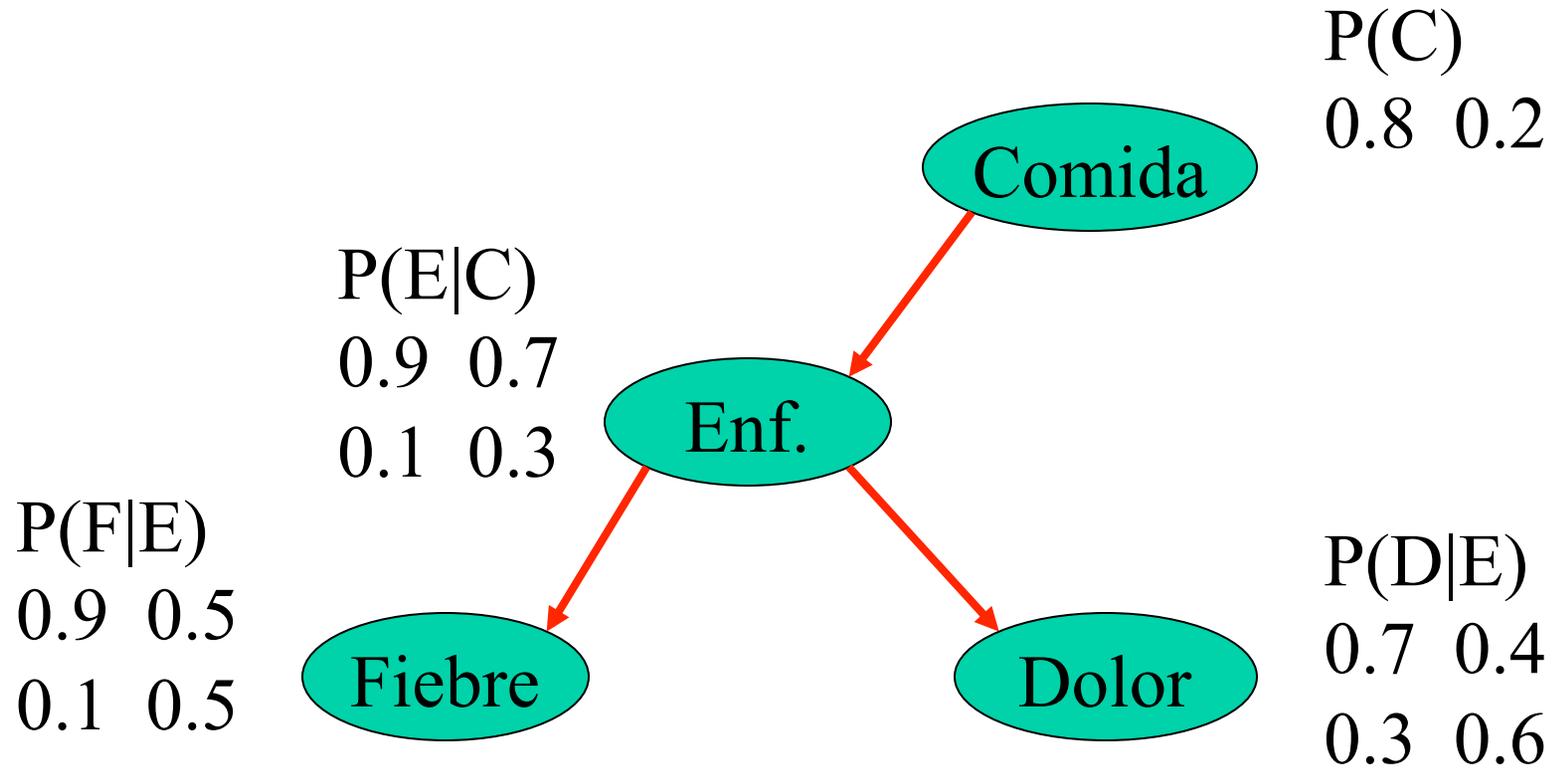
$$\lambda (B_i) = [0, 0, \dots, 1, 0, \dots, 0] \text{ (1 para valor asignado)}$$

$$\pi (B_i) = [0, 0, \dots, 1, 0, \dots, 0] \text{ (1 para valor asignado)}$$

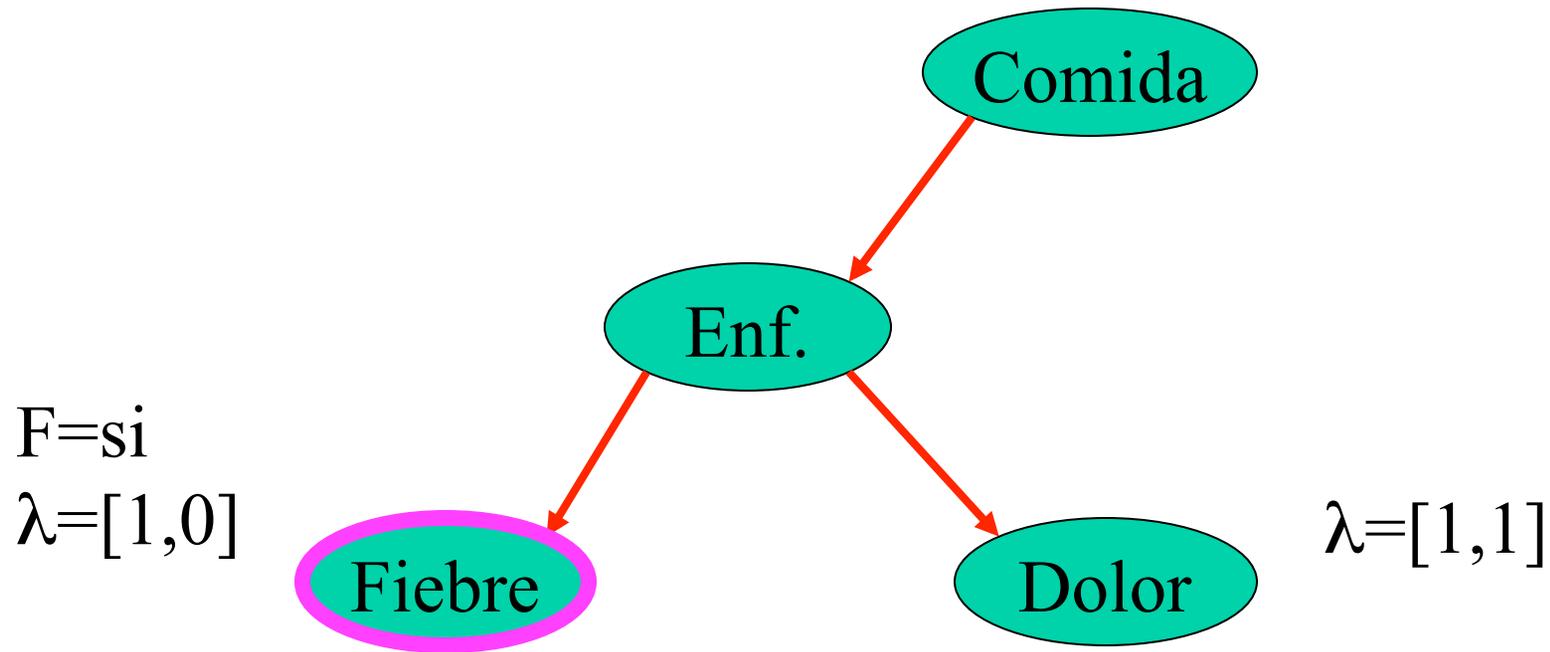
- Nodo raíz no conocido:

$$\pi (A) = P(A), \text{ (probabilidad marginal inicial)}$$

Ejemplo

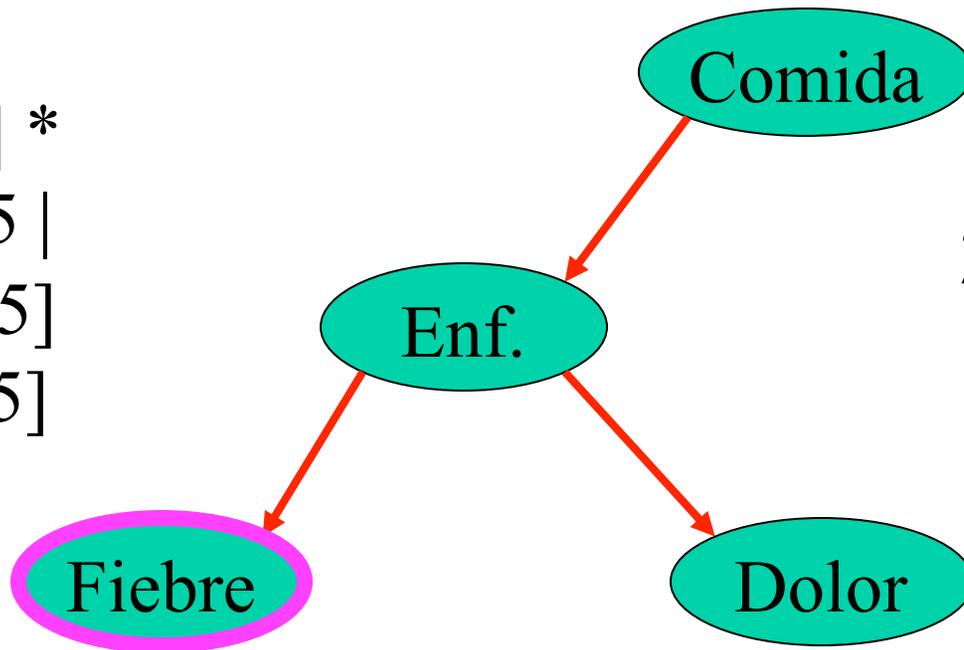


Ejemplo



Ejemplo

$$\lambda_F = [1, 0] * \begin{bmatrix} .9 & .5 \\ .1 & .5 \end{bmatrix} = [.9 & .5]$$



$$\lambda_D = [1, 1] * \begin{bmatrix} .7 & .4 \\ .3 & .6 \end{bmatrix} = [1 & 1]$$

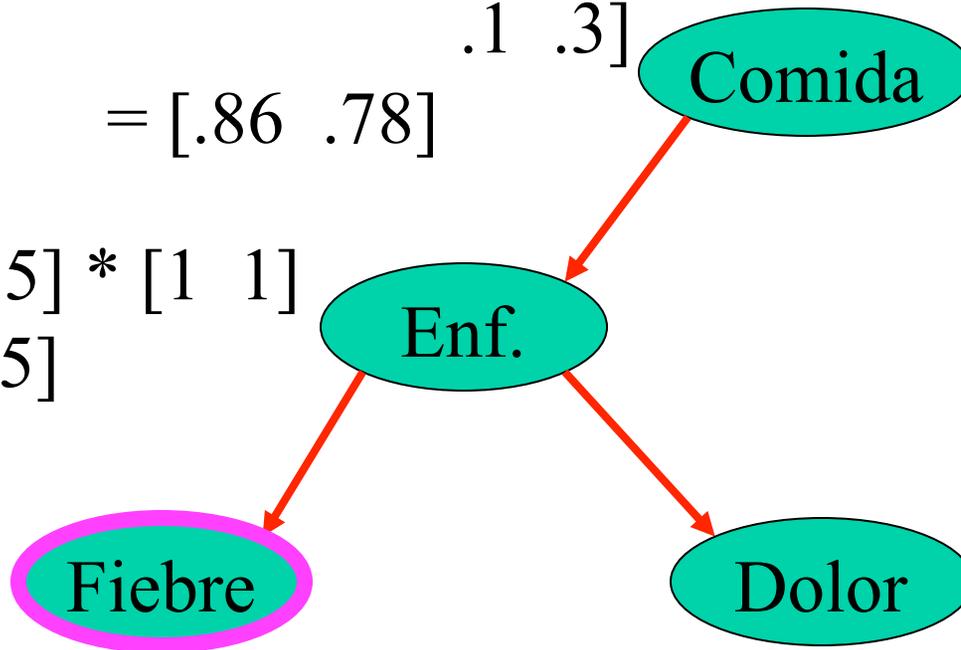
$$P(F|E) \begin{bmatrix} 0.9 & 0.5 \\ 0.1 & 0.5 \end{bmatrix}$$

$$P(D|E) \begin{bmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{bmatrix}$$

Ejemplo

$$\begin{aligned}\lambda(C) &= [.9 \ .5] * [.9 \ .7 | \\ &\quad \quad \quad .1 \ .3] \\ &= [.86 \ .78]\end{aligned}$$

$$\begin{aligned}\lambda(E) &= [.9 \ .5] * [1 \ 1] \\ &= [.9 \ .5]\end{aligned}$$



$$\begin{aligned}P(E|C) \\ 0.9 \ 0.7 \\ 0.1 \ 0.3\end{aligned}$$

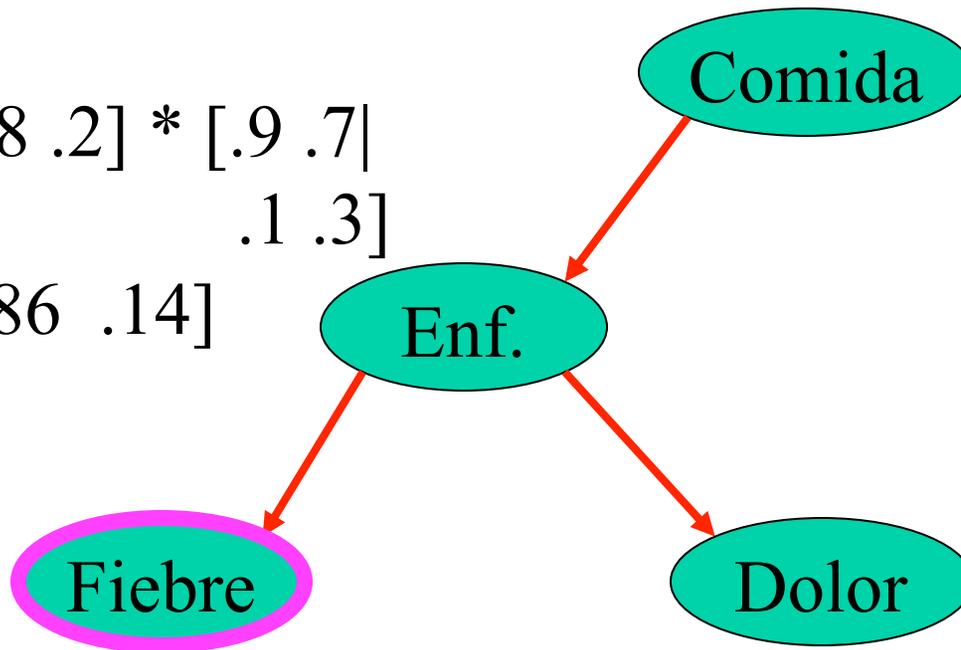
$$\begin{aligned}P(F|E) \\ 0.9 \ 0.5 \\ 0.1 \ 0.5\end{aligned}$$

$$\begin{aligned}P(D|E) \\ 0.7 \ 0.4 \\ 0.3 \ 0.6\end{aligned}$$

Ejemplo

$$\pi(C) = [.8 \ .2]$$

$$\begin{aligned}\pi(E) &= [.8 \ .2] * [.9 \ .7 | \\ &\quad .1 \ .3] \\ &= [.86 \ .14]\end{aligned}$$



$$P(E|C)$$

0.9	0.7
0.1	0.3

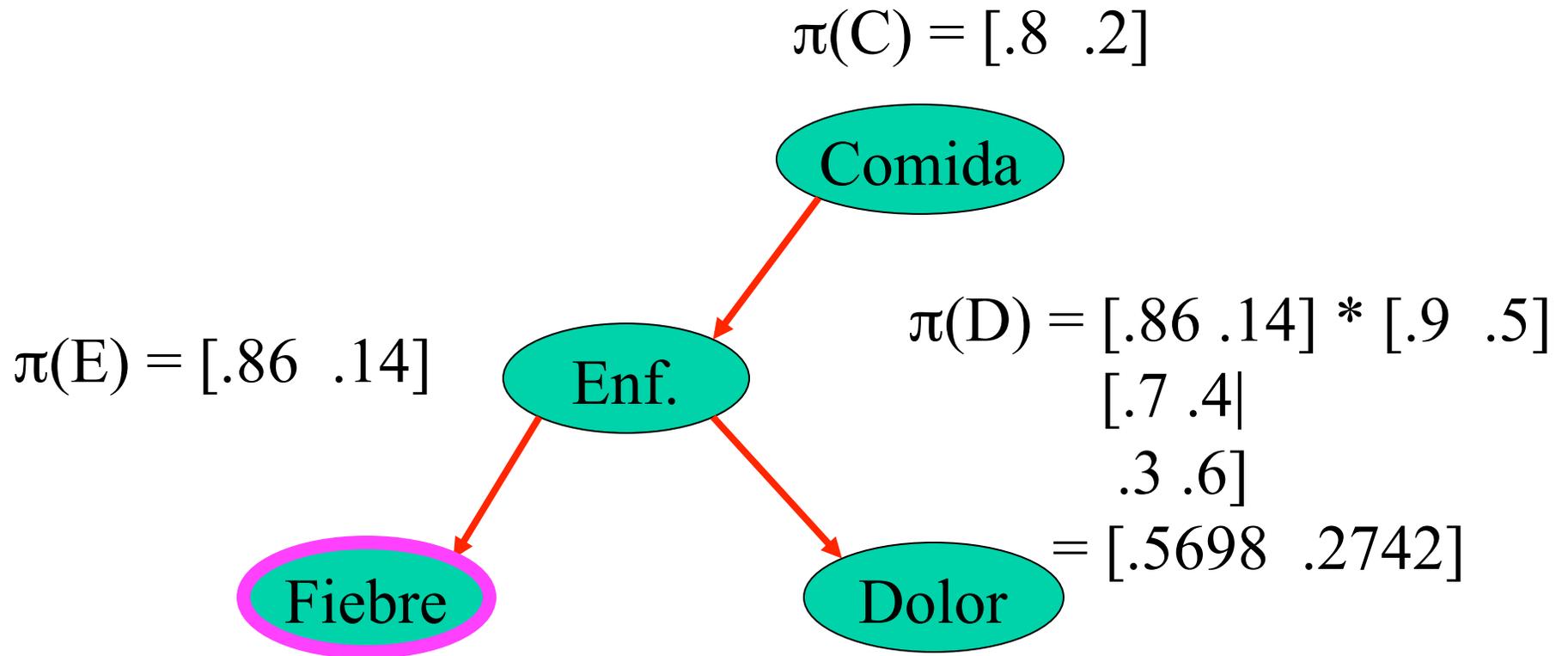
$$P(F|E)$$

0.9	0.5
0.1	0.5

$$P(D|E)$$

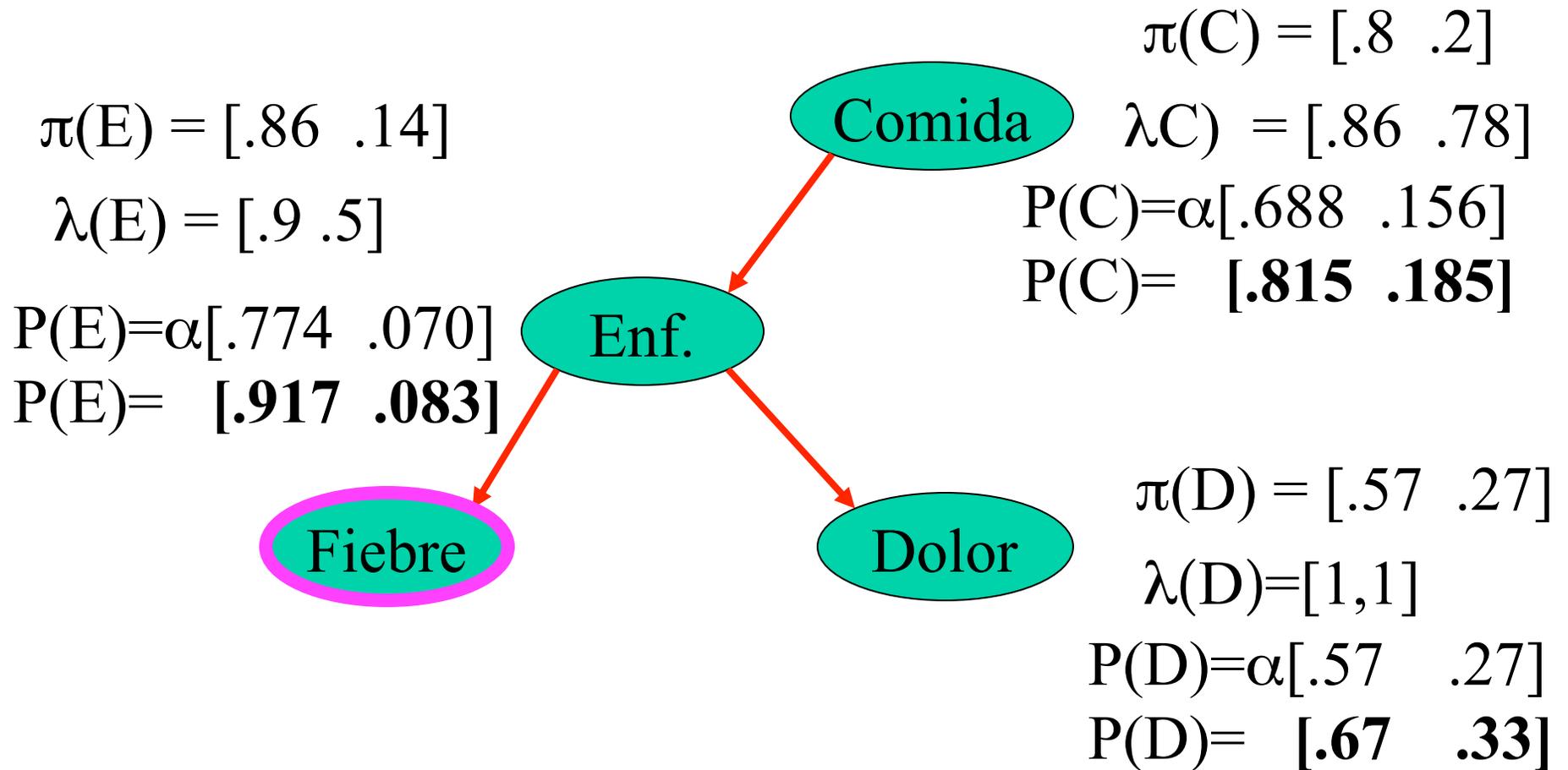
0.7	0.4
0.3	0.6

Ejemplo



P(D E)	
0.7	0.4
0.3	0.6

Ejemplo



Demo 1

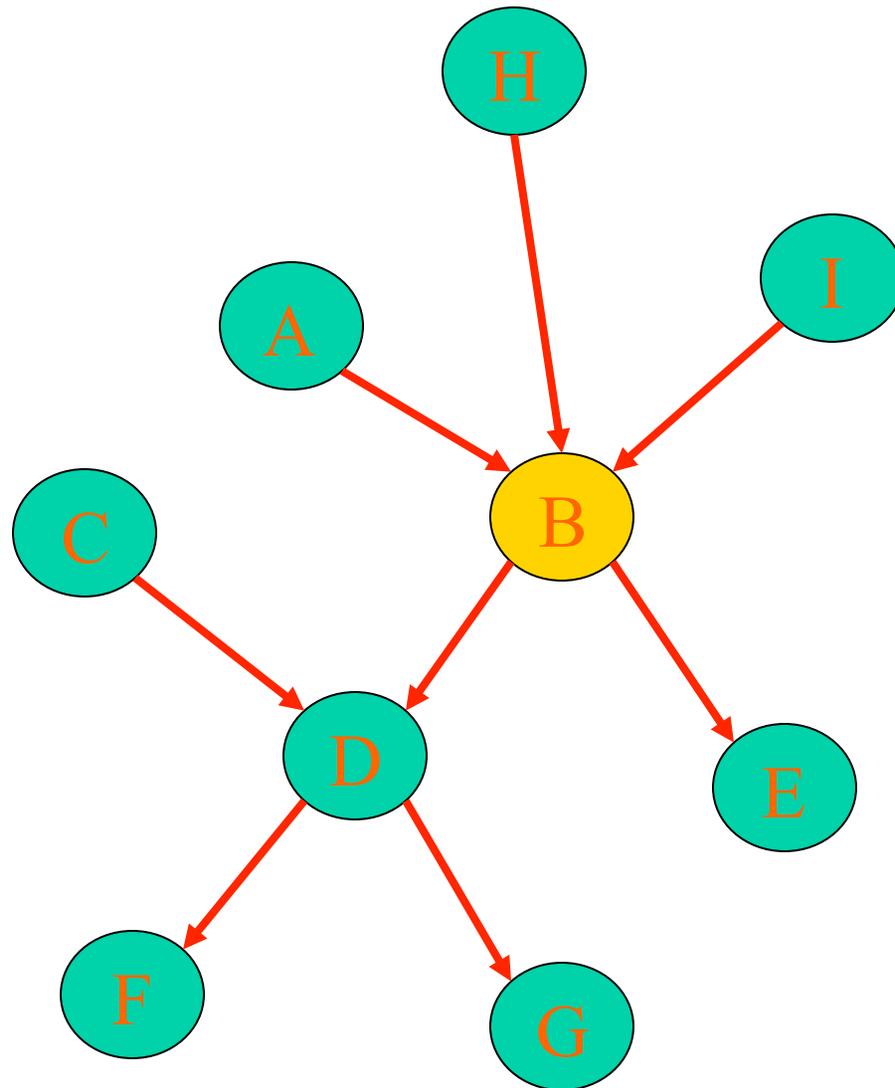
- Ejemplo en HUGIN

Propagación en poliárboles

- Un poliárbol es una red conectada en forma sencilla, pero en la que un nodo puede tener varios padres:

$$P(B \mid A_1, A_2, \dots, A_n)$$

Propagación en Poliárboles



Algoritmo

- El método es muy similar al de árboles, con algunas consideraciones adicionales:
 - Considerar la probabilidad condicional del nodo dados todos sus padres para el cálculo de π y λ
 - Enviar los mensajes λ a cada uno de los padres de un nodo

Propagación en redes multiconectadas

- Una red multiconectada es un grafo no conectado en forma sencilla, es decir, en el que hay múltiples trayectorias entre nodos.
- Para este tipo de redes existen varios tipos de técnicas de inferencia:
 - **Propagación “Loopy”**
 - **Condicionamiento**
 - **Simulación estocástica**
 - **Agrupamiento**

Propagación “Loopy”

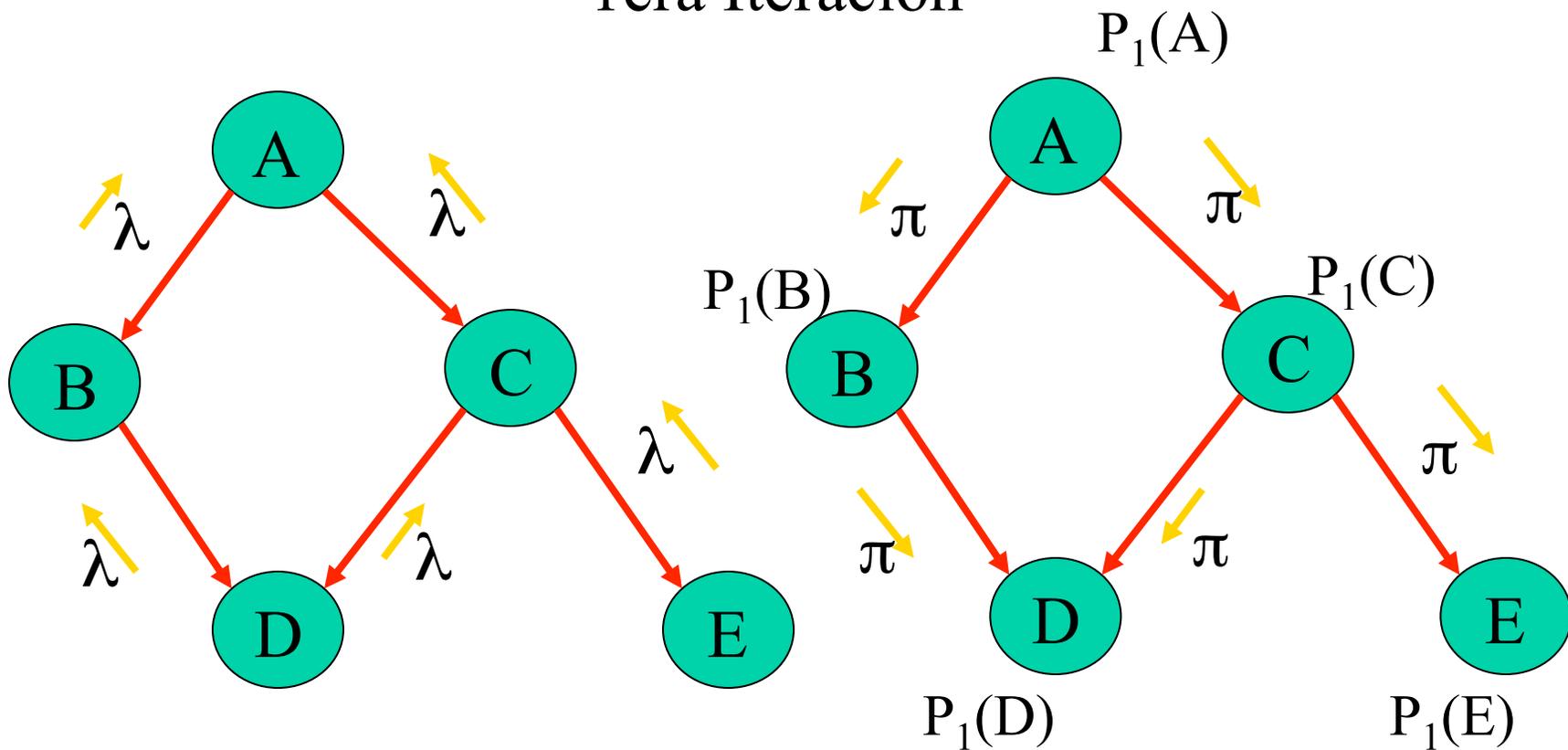
- Este tipo de inferencia es simplemente la aplicación de la técnica de propagación en árboles a redes multiconectadas (con ciclos)
- En este caso ya no hay garantía de obtener una solución exacta, pero en muchos casos se obtiene una “buena” solución aproximada

Propagación “Loopy”

- Algoritmo:
 1. Inicializar π y λ en todos los nodos en forma aleatoria
 2. Repetir hasta que converja o un máximo número de iteraciones:
 - a. Propagar de acuerdo al algoritmo de propagación en árboles
 - b. Obtener la probabilidad marginal en cada nodo y comparar con la anterior

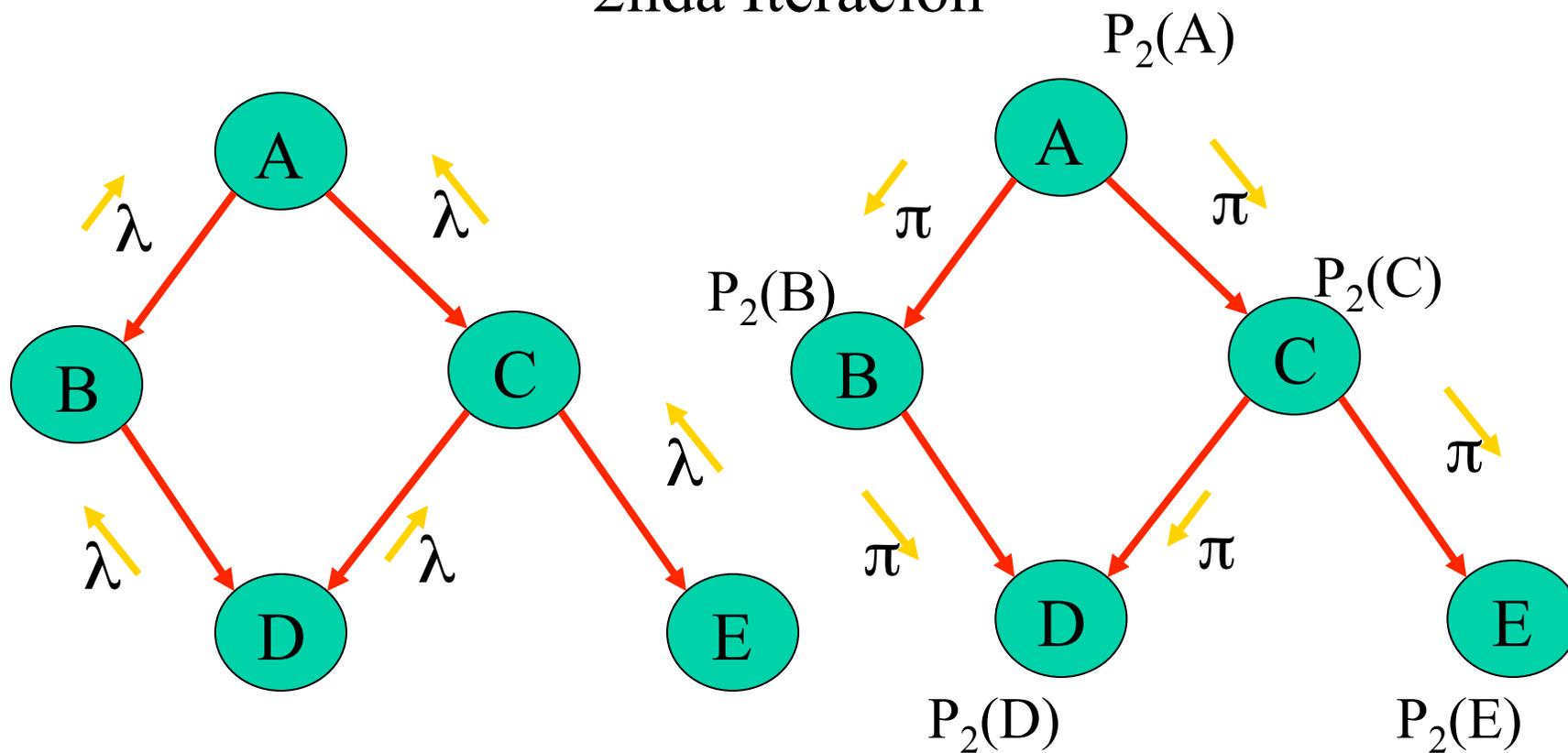
Propagación “Loopy”

– 1era Iteración



Propagación “Loopy”

– 2nda Iteración



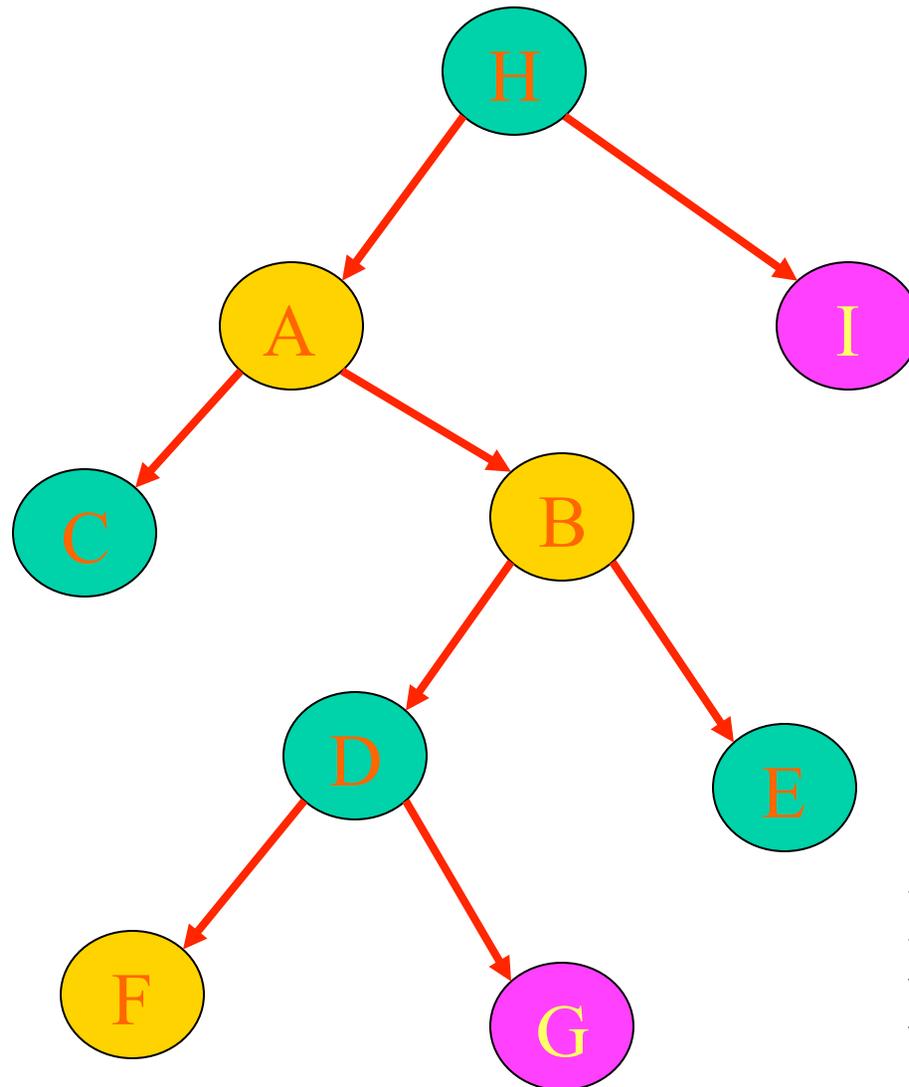
Propagación “Loopy”

- Para ciertas estructuras el algoritmo converge a una buena aproximación de la solución exacta
- Pero en algunas estructuras no converge!
- La aplicación más impactante de este esquema es en códigos de corrección de errores (“Turbo Codes”)

Abducción

- La “abducción” se define como encontrar la mejor “explicación” (valores de un cierto conjunto de variables) dada cierta evidencia
- Normalmente se buscan los valores del conjunto “explicación” que tiene mayor probabilidad
- En general, el conjunto de mayor probabilidad NO es igual a los valores individuales de mayor probabilidad

Abducción



Ejemplo:
 $\text{Max } P(A, B, F | G, I)$

Referencias

- Pearl 88 – Cap. 4,5
- Neapolitan 90 – Cap. 6,7,8
- Sucar, Morales, Hoey - Cap. 2
- Koller & Friedman – Cap. 9, 10
- K. Murphy, Y. Weiss, M. Jordan, “Loopy Belief Propagation for Approximate Inference: An Empirical Study”, UAI’99

Actividades

- Leer sobre inferencia en redes bayesianas (capítulo en la página, referencias)
- Ejercicio Inferencia en RB