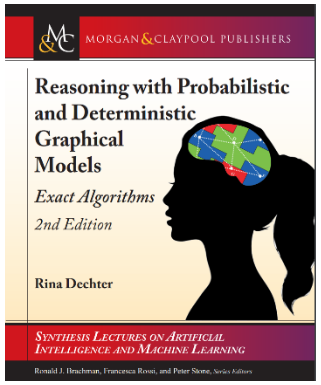
Causal and Probabilistic Reasoning

Slides Set 5:

Exact Inference Algorithms Bucket-elimination

Rina Dechter

(Dechter chapter 4, Darwiche chapter 6)


MORGAN & CLAYPOOL PUBLISHERS

Reasoning with Probabilistic and Deterministic Graphical Models

Exact Algorithms

2nd Edition

Rina Dechter

Synthesis Lectures on Artificial Intelligence and Machine Learning

Ronald J. Brachman, Francesca Rossi, and Peter Stone, *Series Editors*

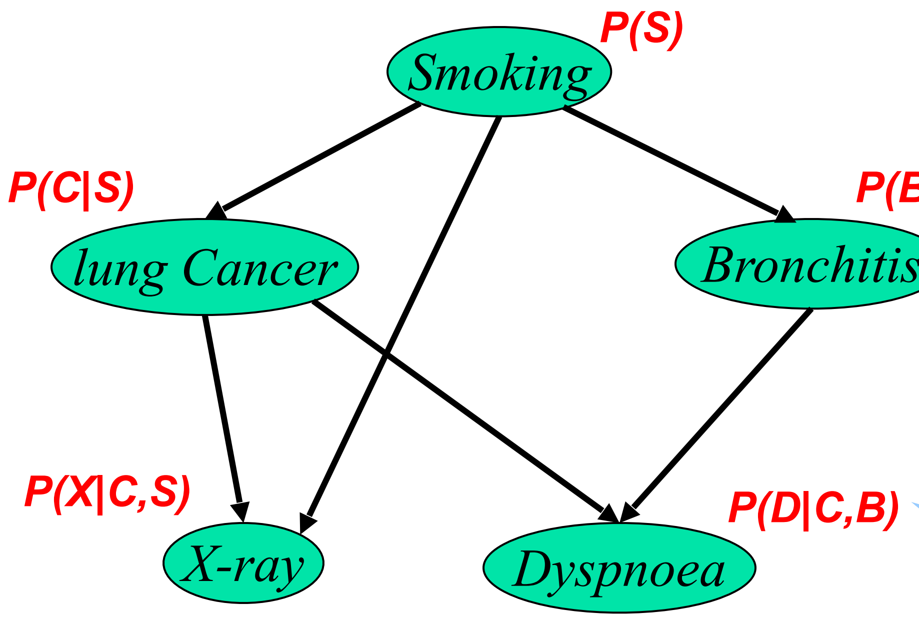
Inference for probabilistic networks

- Bucket elimination (Dechter chapter 4)
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→   Marginal Map)
  - Influence diagrams ?
- Induced-Width (Dechter, Chapter 3.4)

Inference for probabilistic networks

- Bucket elimination
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→ Marginal Map)
- Induced-Width

Bayesian networks: example (Pearl, 1988)



CPD:

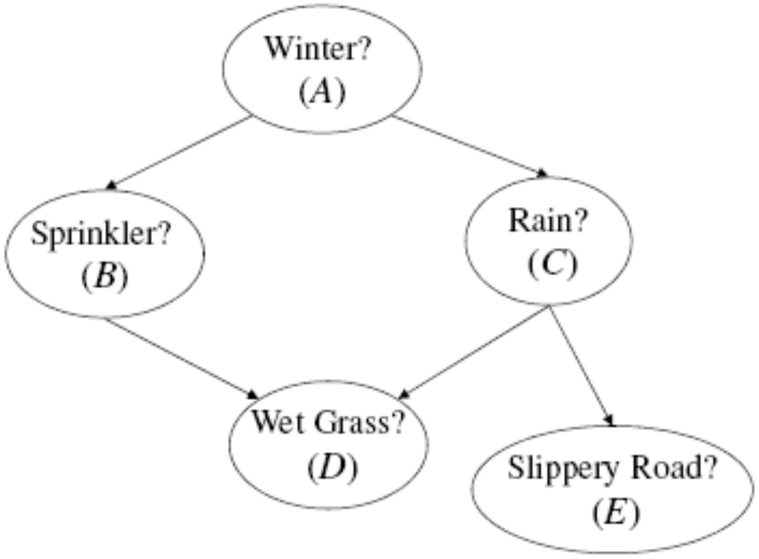
C	B	P(D|C,B)	
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

$$P(S, C, B, X, D) = P(S)\,P(C|S)\,P(B|S)\,P(X|C,S)\,P(D|C,B)$$

Belief Updating:

P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

A Bayesian Network



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

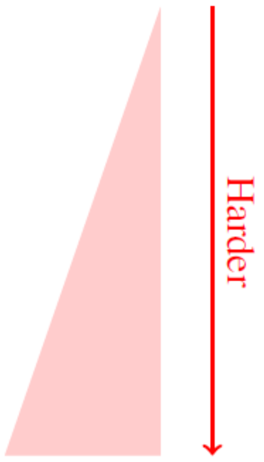
A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Types of queries

Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference: (e.g., causal effects)	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP): (optimal prediction)	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU): (e.g., decisions & planning)	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$



- **NP-hard**: exponentially many terms
- Difficulty (in part) due to restricted elimination orderings
- Focus is on **approximation** and Anytime algorithms
  - **Anytime**: very fast & very approximate ! Slower & more accurate

Belief updating is NP-hard

- Each SAT formula can be mapped into a belief updating query in a Bayesian network

- Example

A simple network

Given:

$$\begin{tikzcd} A \arrow[r, no head] & B \arrow[r] & C \arrow[r] & D \end{tikzcd}$$

- How can we compute P(D)?, P(D|A=0)? P(A|D=0)?
- Brute force O(k^4)
- Maybe O(4k^2)

A simple example

- Suppose we have two factors: $f(X) = f_{12}(X_1, X_2)\, f_{23}(X_2, X_3)$

- To compute the partition function (sum):

$$Z = \sum_{x_1, x_2, x_3} f(x_1, x_2, x_3) = f(0,0,0) + f(0,0,1) + f(0,0,2) + f(0,1,0) + \dots$$
$$+ f(1,0,0) + f(1,0,1) + f(1,0,2) + f(1,1,0) + \dots$$

  - Use the factorization of f(x):

$$Z = f_{12}(0,0)\, f_{23}(0,0) + f_{12}(0,0)\, f_{23}(0,1) + f_{12}(0,0)\, f_{23}(0,2) + f_{12}(0,1)\, f_{23}(1,0) + \dots$$
$$+ f_{12}(1,0)\, f_{23}(0,0) + f_{12}(1,0)\, f_{23}(0,1) + f_{12}(1,0)\, f_{23}(0,2) + f_{12}(1,1)\, f_{23}(1,0) + \dots$$

and apply the distributive rule:

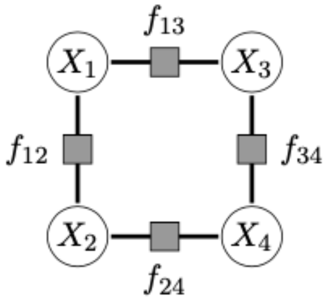
$$= f_{12}(0,0) \Big(f_{23}(0,0) + f_{23}(0,1) + f_{23}(0,2) \Big) + f_{12}(0,1) \Big(f_{23}(1,0) + \dots$$
$$+ f_{12}(1,0) \Big(f_{23}(0,0) + f_{23}(0,1) + f_{23}(0,2) \Big) + f_{12}(1,1) \Big(f_{23}(1,0) + \dots$$

We can pre-compute and re-use these terms in the sum!

$$\lambda(x_2) = \Big(\sum_{x_3} f_{23}(x_2, x_3) \Big)$$

$$Z = \sum_{x_1, x_2} f_{12}(x_1, x_2)\, \lambda(x_2)$$

Variable elimination



Product of factors:

$$p(X_1, X_2, X_3, X_4) = \frac{1}{Z} f_{12}(X_1, X_2) f_{13}(X_1, X_3) f_{24}(X_2, X_4) f_{34}(X_3, X_4).$$

Compute:

$$Z = \sum_{x_4} \sum_{x_3} \sum_{x_2} \sum_{x_1} f_{34}(x_3, x_4) f_{24}(x_2, x_4) f_{12}(x_1, x_2) f_{13}(x_1, x_3),$$

Collect terms involving x_1, then x_2, and so on:

$$Z = \sum_{x_4} \sum_{x_3} f_{34}(x_3, x_4) \sum_{x_2} f_{24}(x_2, x_4) \sum_{x_1} f_{12}(x_1, x_2) f_{13}(x_1, x_3),$$

"Bucket elimination":

$$\lambda_1(x_2, x_3) = \sum_{x_1} f_{12}(x_1, x_2) f_{13}(x_1, x_3),$$

Collect all factors with x_1 in a "bucket

$$\lambda_2(x_3, x_4) = \sum_{x_2} f_{24}(x_2, x_4) \lambda_1(x_2, x_3),$$

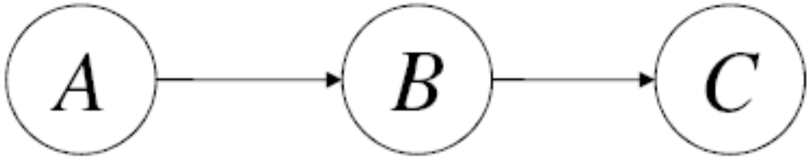
Collect all remaining factors with x_2

$$\lambda_3(x_4) = \sum_{x_3} f_{34}(x_3, x_4) \lambda_2(x_3, x_4),$$

Place intermediate calculations in bucket of their earliest argument

$$Z = \sum_{x_4} \lambda_3(x_4),$$

Elimination as a Basis for Inference



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.9
true	false	.1
false	true	.2
false	false	.8

B	C	$\Theta_{C\|B}$
true	true	.3
true	false	.7
false	true	.5
false	false	.5

To compute the prior marginal on variable C, $\Pr(C)$

we first eliminate variable A and then variable B

Elimination as a Basis for Inference

- There are two factors that mention variable A, Θ_A and $\Theta_{B|A}$
- We multiply these factors first and then sum out variable A from the resulting factor.
- Multiplying Θ_A and $\Theta_{B|A}$:

A	B	$\Theta_A \Theta_{B\|A}$
true	true	.54
true	false	.06
false	true	.08
false	false	.32

- Summing out variable A:

B	$\sum_A \Theta_A \Theta_{B\|A}$
true	$.62 = .54 + .08$
false	$.38 = .06 + .32$

Elimination as a Basis for Inference

- We now have two factors, $\sum_A \Theta_A \Theta_{B|A}$ and $\Theta_{C|B}$, and we want to eliminate variable B
- Since B appears in both factors, we must multiply them first and then sum out B from the result.
- Multiplying:

B	C	$\Theta_{C\|B} \sum_A \Theta_A \Theta_{B\|A}$
true	true	.186
true	false	.434
false	true	.190
false	false	.190

- Summing out:

C	$\sum_B \Theta_{C\|B} \sum_A \Theta_A \Theta_{B\|A}$
true	.376
false	.624

Elimination as a Basis for Inference

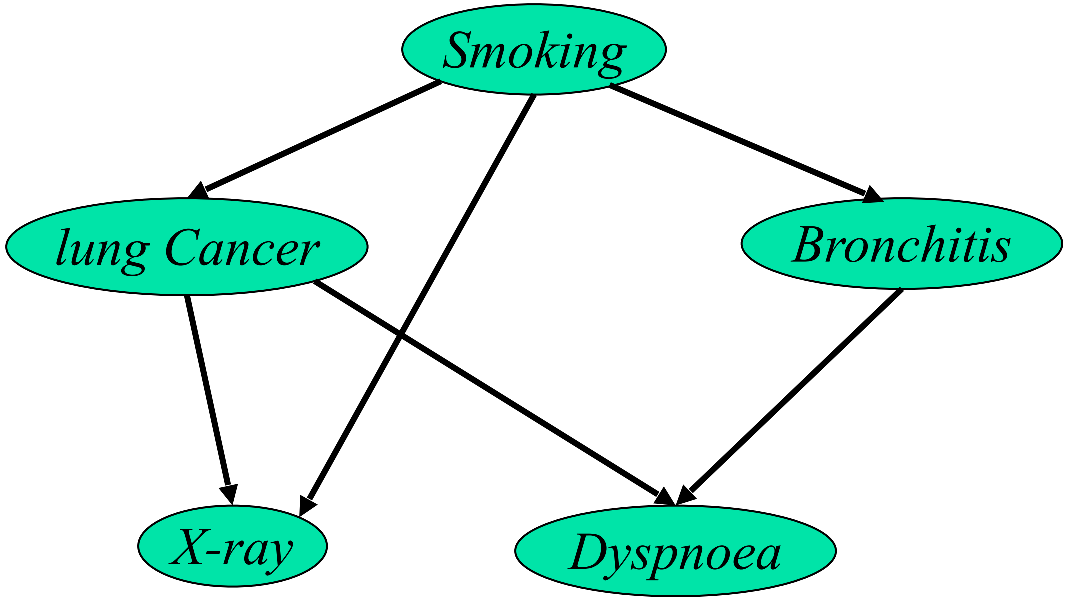
- We now have two factors, $\sum_A \Theta_A \Theta_{B|A}$ and $\Theta_{C|B}$, and we want to eliminate variable B
- Since B appears in both factors, we must multiply them first and then sum out B from the result.
- Multiplying:

B	C	$\Theta_{C\mid B} \sum_A \Theta_A \Theta_{B\mid A}$
true	true	.186
true	false	.434
false	true	.190
false	false	.190

- Summing out:

C	$\sum_B \Theta_{C\mid B} \sum_A \Theta_A \Theta_{B\mid A}$
true	.376
false	.624

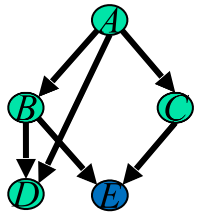
Belief updating



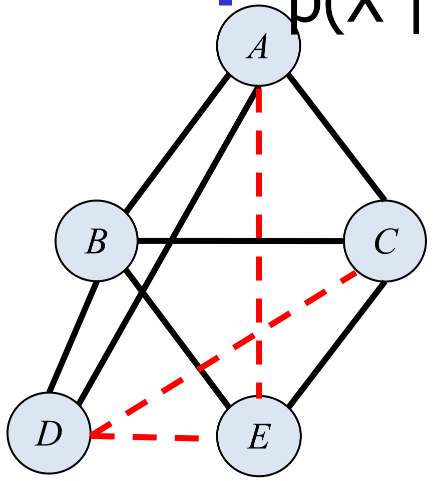
P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

From essai

Belief updating



- p(X | Evidence) = ?



"primal" graph

$$p(A|E=0)$$

$$\propto p(A, E=0)$$

$$= \sum_{e,d,c,b} p(A)\, \underbrace{p(b|A)}\, p(c|A)\, \underbrace{p(d|b,A)\, p(e|b,c)}\, \mathbb{1}[e=0]$$

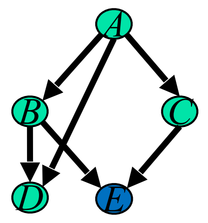
$$p(A) \sum_{e} \sum_{d} \sum_{c} p(c|A)\, \mathbb{1}[e=0]\, \underbrace{\sum_{b} p(b|A)\, p(d|b,A)\, p(e|b,c)}_{\lambda_{B\to C}(a,d,c,e)}$$

$\boxed{\textit{Variable Elimination}}$

From essai

Bucket elimination

Algorithm BE-bel [Dechter 1996]



$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$$\sum_b \prod \longleftarrow$$ *Elimination & combination operators*

bucket B: $\overbrace{p(b|A)\;\; p(d|b,A)\;\; p(e|b,c)}$

bucket C: $p(c|A) \quad \lambda_{B\to C}(\underbrace{A,d,c,e})$

bucket D: $\lambda_{C\to D}(A,d,e)$

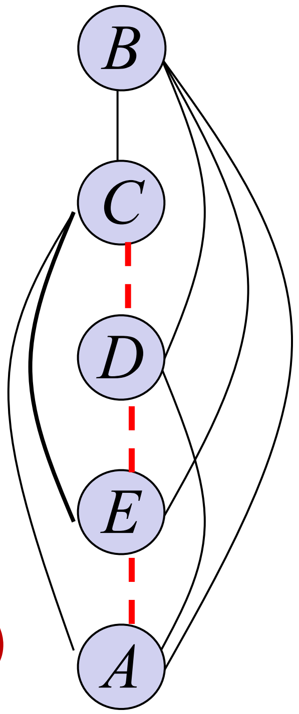
bucket E: $\mathbb{1}[E=0]\ \lambda_{D\to E}(A,e)$

bucket A: $p(A) \quad \lambda_{E\to A}(A)$

$p(E=0)$

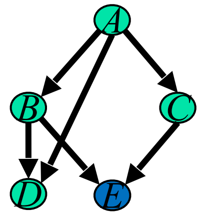
$$p(A|E=0) = p(A,E=0)\,/\,p(E=0)$$

W=4*
"induced width"
(max clique size)



Bucket elimination

Algorithm BE-bel [Dechter 1996]



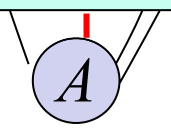
$$p(A|E=0) = \alpha \sum_{e,d,c,b} p(A)\, p(b|A)\, p(c|A)\, p(d|A,b)\, p(e|b,c)\, \mathbb{1}[e=0]$$

$$\sum_b \prod \longleftarrow$$ *Elimination & combination operators*

Time and space exponential in the induced-width / treewidth

bucket A: $p(A)$ $\lambda_{E\to A}(A)$

induced width
(max clique size)

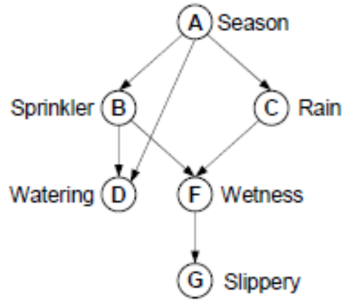


$p(E=0)$

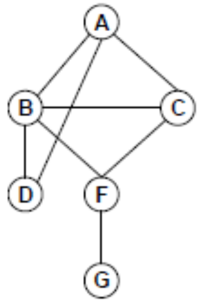
$p(A|E=0) = p(A,E=0)\ /\ p(E=0)$

A Bayesian network

ordering: A,C,B,F,D,G



(a) Directed acyclic graph



(b) Moral graph

$$P(a, g=1) = \sum_{c,b,e,d,g=1} P(a,b,c,d,e,g) = \sum_{c,b,f,d,g=1} P(g|f)P(f|b,c)P(d|a,b)P(c|a)P(b|a)P(a).$$

$$P(a, g=1) = P(a)\sum_{c} P(c|a)\sum_{b} P(b|a)\sum_{f} P(f|b,c)\sum_{d} P(d|b,a)\sum_{g=1} P(g|f). \quad (4.1)$$

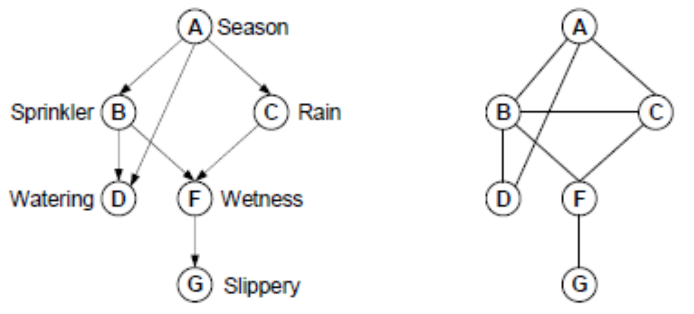
$$P(a, g=1) = P(a)\sum_{c} P(c|a)\sum_{b} P(b|a)\sum_{f} P(f|b,c)\lambda_G(f)\sum_{d} P(d|b,a). \quad (4.2)$$

$$P(a, g=1) = P(a)\sum_{c} P(c|a)\sum_{b} P(b|a)\lambda_D(a,b)\sum_{f} P(f|b,c)\lambda_G(f) \quad (4.3)$$

$$P(a, g=1) = P(a)\sum_{c} P(c|a)\sum_{b} P(b|a)\lambda_D(a,b)\lambda_F(b,c) \quad (4.4)$$

$$P(a, g=1) = P(a)\sum_{c} P(c|a)\lambda_B(a,c) \quad (4.5)$$

A Bayesian network
ordering: A,C,B,F,D,G



(a) Directed acyclic graph (b) Moral graph

$$P(a, g=1) = \sum_{c,b,e,d,g=1} P(a,b,c,d,e,g) = \sum_{c,b,f,d,g=1} P(g|f)P(f|b,c)P(d|a,b)P(c|a)P(b|a)P(a).$$

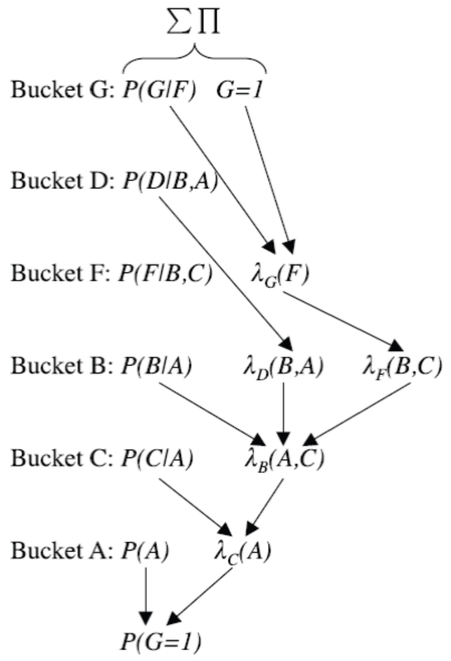
$$P(a, g=1) = P(a)\sum_c P(c|a)\sum_b P(b|a)\sum_f P(f|b,c)\sum_d P(d|b,a)\boxed{\sum_{g=1} P(g|f)} \quad (4.1)$$

$$P(a, g=1) = P(a)\sum_c P(c|a)\sum_b P(b|a)\sum_f P(f|b,c)\lambda_G(f)\boxed{\sum_d P(d|b,a).} \quad (4.2)$$

$$P(a, g=1) = P(a)\sum_c P(c|a)\sum_b P(b|a)\lambda_D(a,b)\boxed{\sum_f P(f|b,c)\lambda_G(f)} \quad (4.3)$$

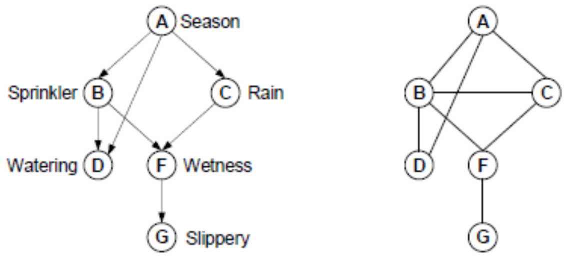
$$P(a, g=1) = P(a)\sum_c P(c|a)\boxed{\sum_b P(b|a)\lambda_D(a,b)\lambda_F(b,c)} \quad (4.4)$$

$$P(a, g=1) = P(a)\sum_c P(c|a)\lambda_B(a,c) \quad (4.5)$$



ΣΠ
Bucket G: P(G|F) G=1
Bucket D: P(D|B,A)
Bucket F: P(F|B,C) λ_G(F)
Bucket B: P(B|A) λ_D(B,A) λ_F(B,C)
Bucket C: P(C|A) λ_B(A,C)
Bucket A: P(A) λ_C(A)
P(G=1)

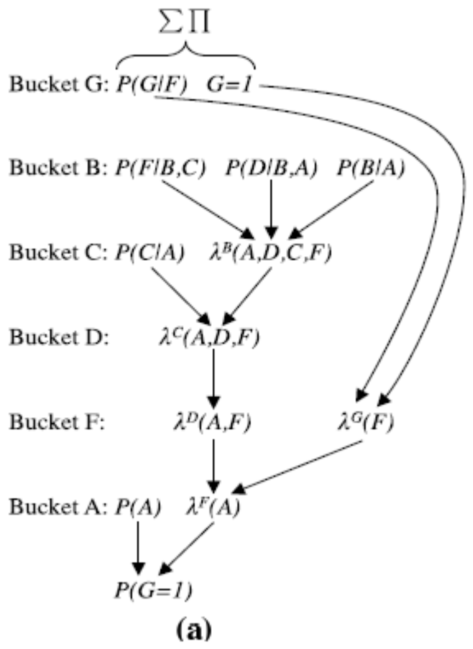
A different ordering



(a) Directed acyclic graph (b) Moral graph

Ordering: A,F,D,C,B,G

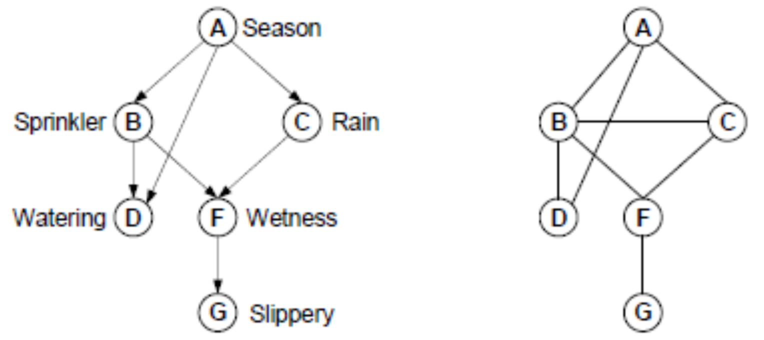
$$P(a, g=1) = P(a)\sum_f \sum_d \sum_c P(c|a) \sum_b P(b|a)\ P(d|a,b)P(f|b,c) \sum_{g=1} P(g|f)$$
$$= P(a)\sum_f \lambda_G(f) \sum_d \sum_c P(c|a) \sum_b P(b|a)\ P(d|a,b)P(f|b,c)$$
$$= P(a)\sum_f \lambda_G(f) \sum_d \sum_c P(c|a)\lambda_B(a,d,c,f)$$
$$= P(a)\sum_f \lambda_g(f) \sum_d \lambda_C(a,d,f)$$
$$= P(a)\sum_f \lambda_G(f)\lambda_D(a,f)$$
$$= P(a)\lambda_F(a)$$



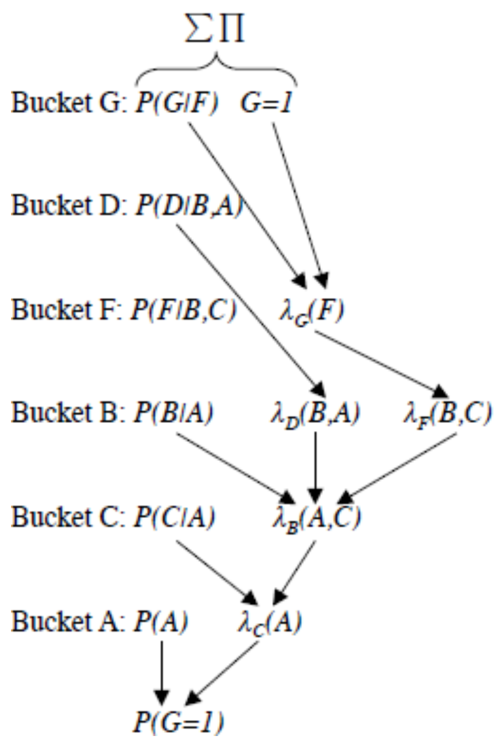
ΣΠ
Bucket G: P(G|F) G=1
Bucket B: P(F|B,C) P(D|B,A) P(B|A)
Bucket C: P(C|A) λ^B(A,D,C,F)
Bucket D: λ^C(A,D,F)
Bucket F: λ^D(A,F) λ^G(F)
Bucket A: P(A) λ^F(A)
P(G=1)
(a)

Figure 4.3: The bucket's output when processing along $d_2 = A, F, D, C, B, G$

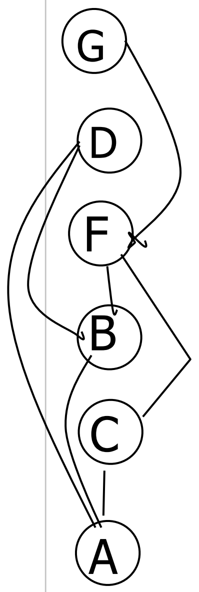
A Bayesian network processed along two orderings

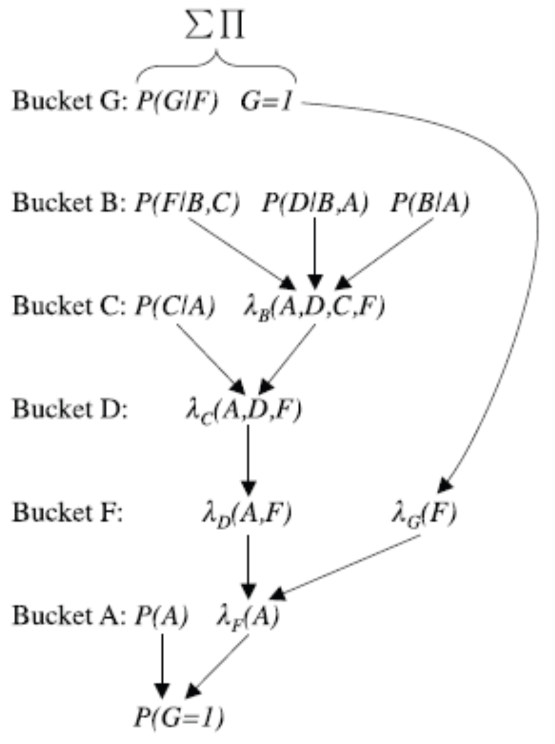


(a) Directed acyclic graph (b) Moral graph



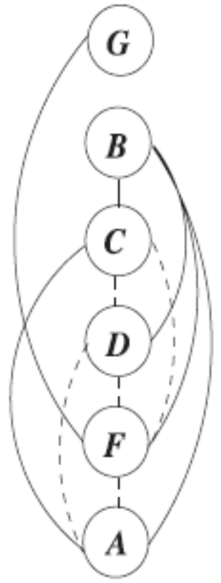
$\Sigma\Pi$
Bucket G: $P(G|F)$ $G=1$
Bucket D: $P(D|B,A)$
Bucket F: $P(F|B,C)$ $\lambda_G(F)$
Bucket B: $P(B|A)$ $\lambda_D(B,A)$ $\lambda_F(B,C)$
Bucket C: $P(C|A)$ $\lambda_B(A,C)$
Bucket A: $P(A)$ $\lambda_C(A)$
$P(G=1)$





$\Sigma\Pi$
Bucket G: $P(G|F)$ $G=1$
Bucket B: $P(F|B,C)$ $P(D|B,A)$ $P(B|A)$
Bucket C: $P(C|A)$ $\lambda_B(A,D,C,F)$
Bucket D: $\lambda_C(A,D,F)$
Bucket F: $\lambda_D(A,F)$ $\lambda_G(F)$
Bucket A: $P(A)$ $\lambda_F(A)$
$P(G=1)$

(a)



(b)

d1=A,C,B,F,D,G

d2: A,F,D,C,B,G

The operation in a bucket

- Multiplying functions
- Marginalizing (summing-out) functions

Combination of Cost Functions

A	B	f(A,B)
b	b	0.4
b	g	0.1
g	b	0
g	g	0.5

●

B	C	f(B,C)
b	b	0.2
b	g	0
g	b	0
g	g	0.8

A	B	C	f(A,B,C)
b	b	b	0.1
b	b	g	0
b	g	b	0
b	g	g	0.08
g	b	b	0
g	b	g	0
g	g	b	0
g	g	g	0.4

= *0.1 x 0.8*

Elimination in a factor

A	B	f(A,B)
b	b	4
b	g	6
b	r	1
g	b	2
g	g	6
g	r	3
r	b	1
r	g	1
r	r	6

Elim = sum

$\xrightarrow{\text{Elim}(\mathbf{f},\mathbf{B})}$

A	g(A)
b	11
g	11
r	8

$\xrightarrow{\text{Elim}(\mathbf{g},\mathbf{A})}$

$h_\emptyset$
30

Factors: Sum-Out Operation

The result of summing out variable X from factor $f(\mathbf{X})$

is another factor over variables $\mathbf{Y} = \mathbf{X} \setminus \{X\}$:

$$\left(\sum_X f\right)(\mathbf{y}) \overset{def}{=} \sum_x f(x, \mathbf{y})$$

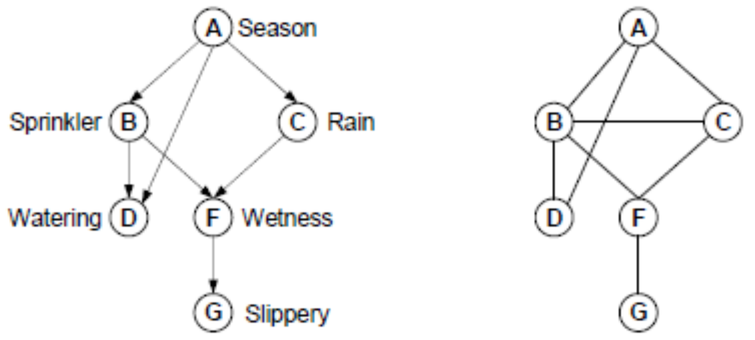
B	C	D	f_1
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

B	C	$\sum_D f_1$
true	true	1
true	false	1
false	true	1
false	false	1

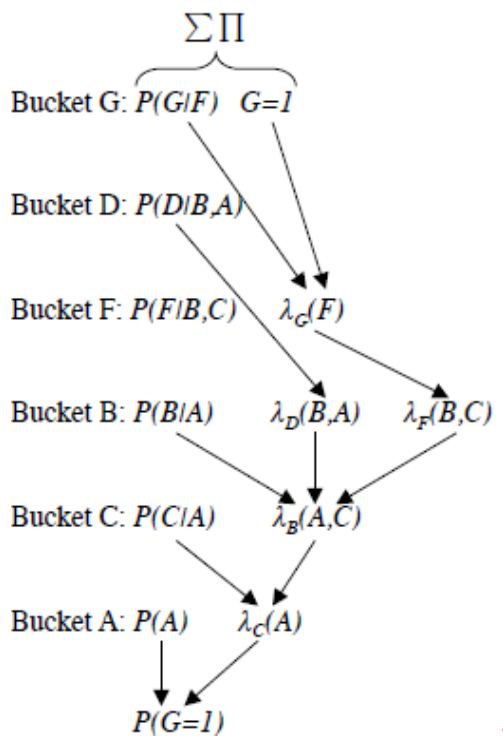
	$\sum_B \sum_C \sum_D f_1$
$\top$	4

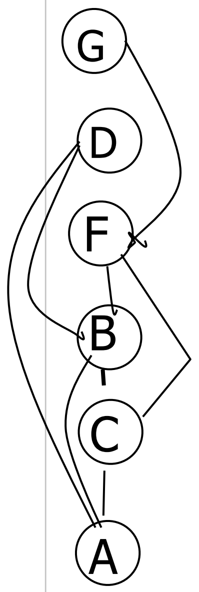
Thanks to Darwiche

Bucket elimination and induced-width



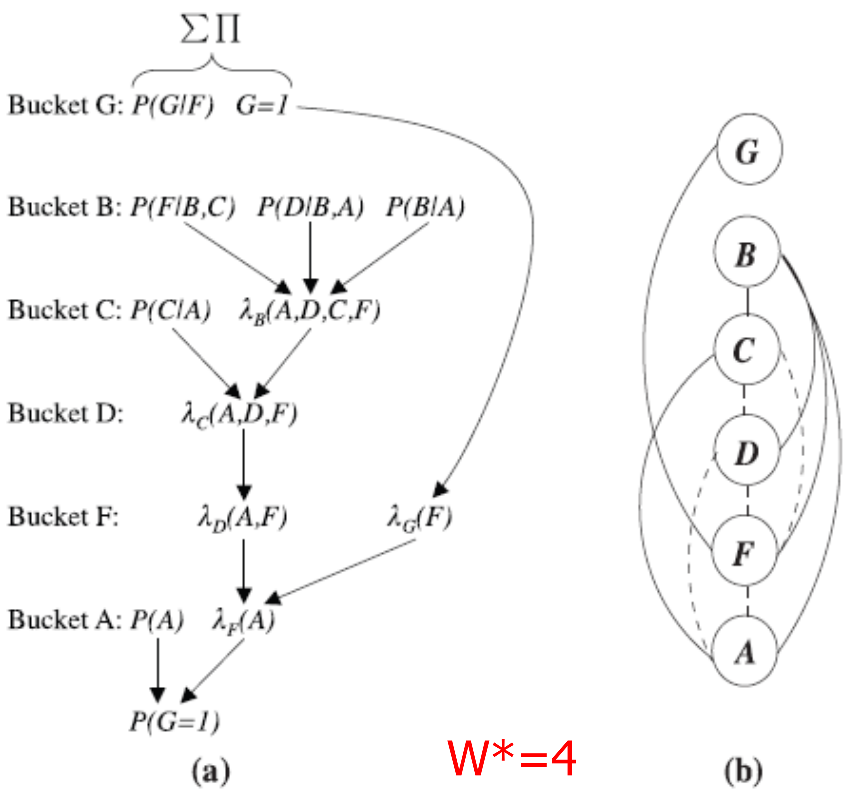
(a) Directed acyclic graph (b) Moral graph


$\Sigma\Pi$
Bucket G: $P(G|F)$ $G=1$
Bucket D: $P(D|B,A)$
Bucket F: $P(F|B,C)$ $\lambda_G(F)$
Bucket B: $P(B|A)$ $\lambda_D(B,A)$ $\lambda_F(B,C)$
Bucket C: $P(C|A)$ $\lambda_B(A,C)$
Bucket A: $P(A)$ $\lambda_C(A)$
$P(G=1)$



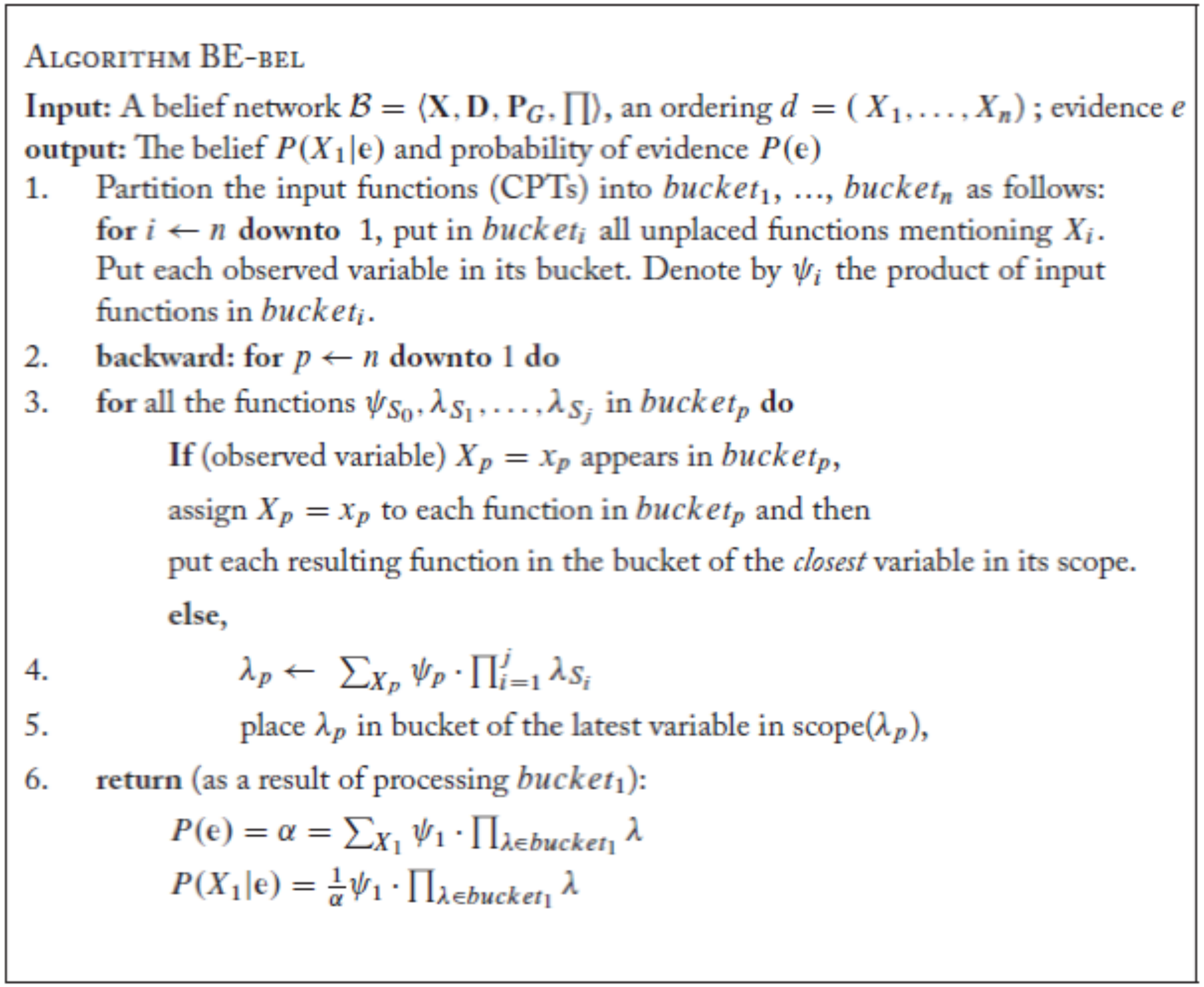
W*=2

d1=A,C,B,F,D,G


$\Sigma\Pi$
Bucket G: $P(G|F)$ $G=1$
Bucket B: $P(F|B,C)$ $P(D|B,A)$ $P(B|A)$
Bucket C: $P(C|A)$ $\lambda_B(A,D,C,F)$
Bucket D: $\lambda_C(A,D,F)$
Bucket F: $\lambda_D(A,F)$ $\lambda_G(F)$
Bucket A: $P(A)$ $\lambda_F(A)$
$P(G=1)$
(a)
G
B
C
D
F
A
(b)

W*=4

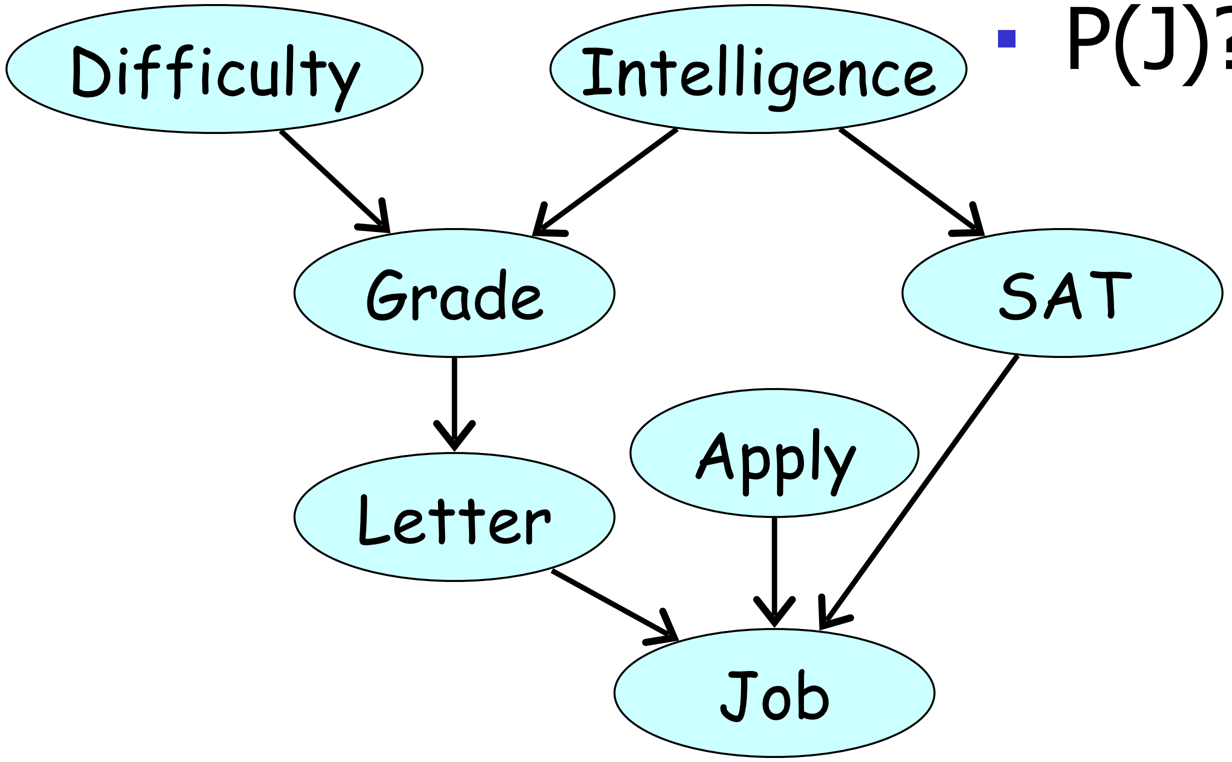
d2: A,F,D,C,B,G



```
ALGORITHM BE-BEL
Input: A belief network $\mathcal{B} = \langle \mathbf{X}, \mathbf{D}, \mathbf{P}_G, \prod \rangle$, an ordering $d = (X_1, \ldots, X_n)$ ; evidence $e$
output: The belief $P(X_1|\mathrm{e})$ and probability of evidence $P(\mathrm{e})$
1.  Partition the input functions (CPTs) into $bucket_1$, …, $bucket_n$ as follows:
    for $i \leftarrow n$ downto 1, put in $bucket_i$ all unplaced functions mentioning $X_i$.
    Put each observed variable in its bucket. Denote by $\psi_i$ the product of input
    functions in $bucket_i$.
2.  backward: for $p \leftarrow n$ downto 1 do
3.  for all the functions $\psi_{S_0}, \lambda_{S_1}, \ldots, \lambda_{S_j}$ in $bucket_p$ do
        If (observed variable) $X_p = x_p$ appears in $bucket_p$,
        assign $X_p = x_p$ to each function in $bucket_p$ and then
        put each resulting function in the bucket of the *closest* variable in its scope.
        else,
4.          $\lambda_p \leftarrow \sum_{X_p} \psi_p \cdot \prod_{i=1}^{j} \lambda_{S_i}$
5.          place $\lambda_p$ in bucket of the latest variable in scope($\lambda_p$),
6.  return (as a result of processing $bucket_1$):
        $P(\mathrm{e}) = \alpha = \sum_{X_1} \psi_1 \cdot \prod_{\lambda \in bucket_1} \lambda$
        $P(X_1|\mathrm{e}) = \frac{1}{\alpha} \psi_1 \cdot \prod_{\lambda \in bucket_1} \lambda$
```

Figure 4.5: BE-bel: a sum-product bucket-elimination algorithm.

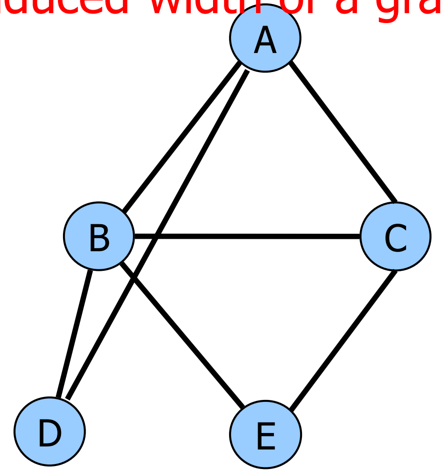
Student network example



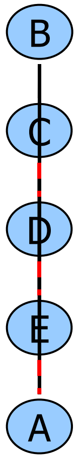
- P(J)?

Induced width

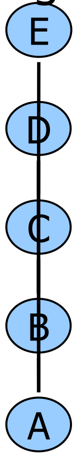
- Width is the max number of parents in the ordered graph
- Induced-width is the width of the induced ordered graph: recursively connecting parents going from last node to first.
- Induced-width w*(d) is the max induced-width over all nodes in ordering d
- Induced-width of a graph, w* is the min w*(d) over all orderings d



primal graph



$w^*(d_1) = 4$



$w^*(d_2) = 2$

Complexity of bucket elimination

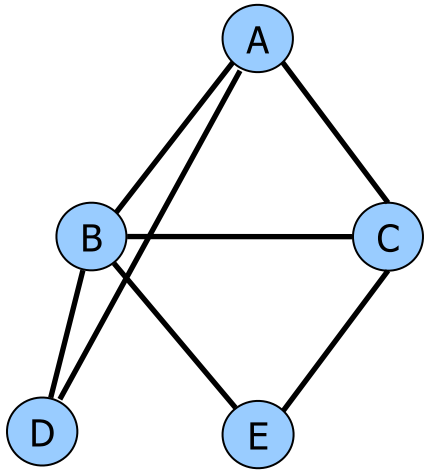
Bucket-Elimination is **time** and **space**

$$O\left(r \exp(w_d^*)\right)$$

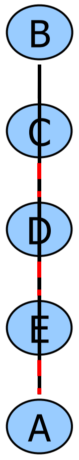
w_d^*: the induced width of the primal graph along ordering d

r = number of functions

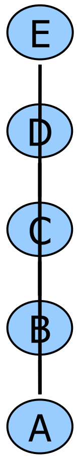
The effect of the ordering:



primal graph



$w^*(d_1) = 4$



$w^*(d_2) = 2$

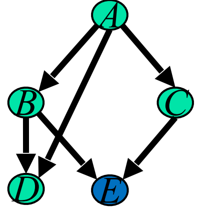
Finding smallest induced-width is hard!

Inference for probabilistic networks

- Bucket elimination
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→   Marginal Map)
- Induced-Width

The impact of evidence?

Algorithm BE-bel



Elimination operator

bucket B: $P(b|a)$ $P(d|b,a)$ $P(e|b,c)$ B=1

bucket C: $P(c|a)$

bucket D:

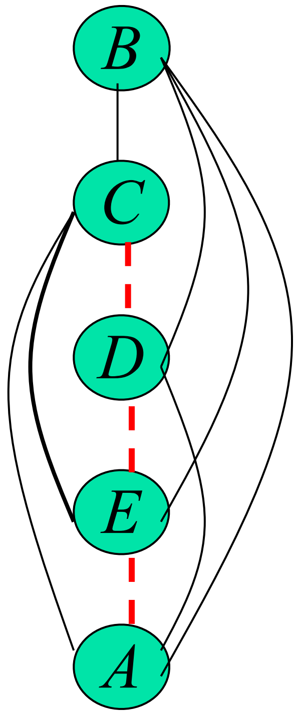
bucket E: $e=0$

bucket A: $P(a)$

$P(e=0)$

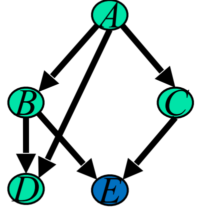
$P(a|e=0)$

W*=4
"induced width"
(max clique size)



The impact of evidence?

Algorithm BE-bel



P(A|E=0,B=1)?

Elimination operator

bucket B: $\underbrace{P(b|a) \quad P(d|b,a) \quad P(e|b,c)}$ B=1

bucket C: $P(c|a)$ P(e|b=1,c)

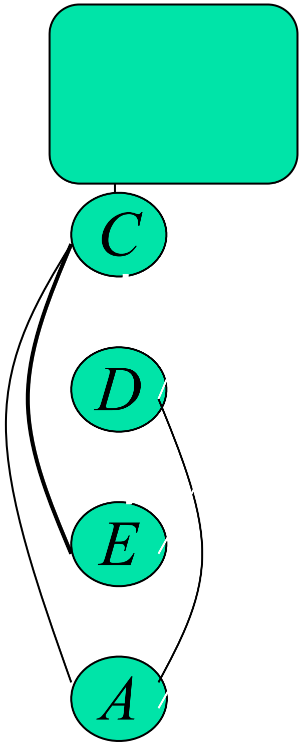
bucket D: P(d|b=1,a)

bucket E: $e=0$

bucket A: $P(a)$ P(b=1|a)

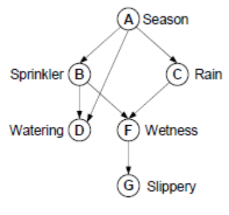
$\mathbf{P(e=0)}$

$\mathbf{P(a|e=0)}$

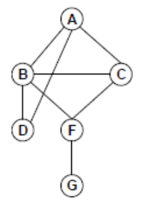


$$P(a|e=0) = \frac{P(a,e=0)}{P(e=0)}$$

The impact of observations



(a) Directed acyclic graph



(b) Moral graph

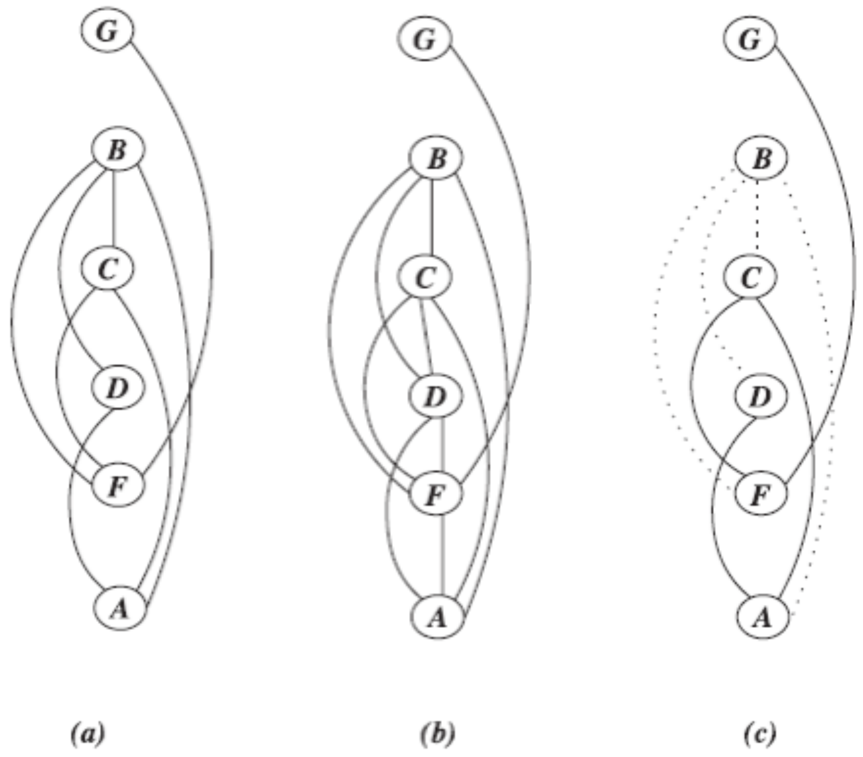
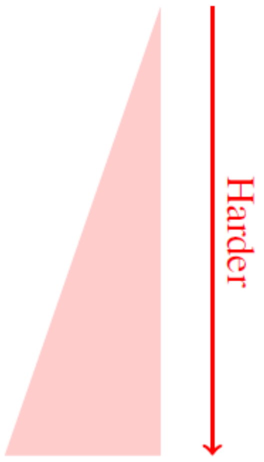


Figure 4.9: Adjusted induced graph relative to observing B.

Ordered graph   Induced graph   Ordered conditioned graph

Types of queries

Max-Inference:	$f(x^*) = \max_x \prod_\alpha f_\alpha(x_\alpha)$
Sum-Inference: (e.g., causal effects)	$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MMAP): (optimal prediction)	$f_M(x_M^*) = \max_{x_M} \sum_{x_S} \prod_\alpha f_\alpha(x_\alpha)$
Mixed-Inference (MEU): (e.g., decisions & planning)	$\text{MEU} = \max_{D_1,\ldots,D_m} \sum_{X_1,\ldots X_n} (\prod_{P_i \in P} P_i) \times (\sum_{r_i \in R} r_i)$



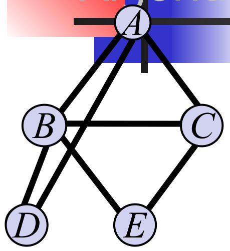
- **NP-hard**: exponentially many terms
- Difficulty (in part) due to restricted elimination orderings
- We will focus on **approximation** algorithms
  - **Anytime**: very fast & very approximate ! Slower & more accurate

Inference for Probabilistic Networks

- Bucket elimination
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→ Marginal Map)
- Induced-Width

Finding MPE/MAP

Algorithm BE-mpe (Dechter 1996, Bertele



$$\text{MPE} = \max_{a,e,d,c,b} p(a)\,p(c|a)\,p(b|a)\,p(\ldots$$

$$= \max_b p(b|a) \cdot p(d|b,a) \cdot p(e|b,c)$$

$$\overbrace{\qquad\qquad\qquad}^{\max_x \prod}$$

bucket B: $p(b|a)\ p(d|b,a)\ p(e|b,c)$

bucket C: $p(c|a)\quad \lambda_{B\to C}\underbrace{(a,d,c,e)}$

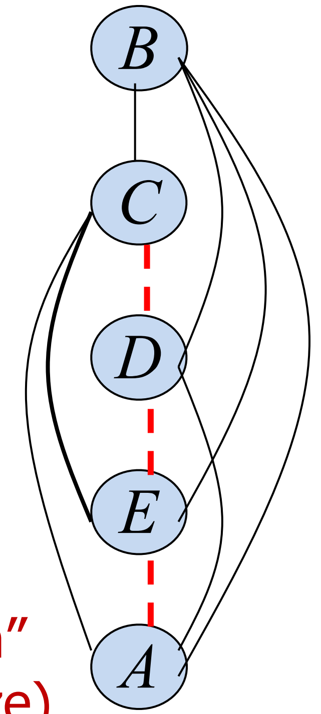
bucket D: $\lambda_{C\to D}(a,d,e)$

bucket E: $\mathbb{1}[e=0]\quad \lambda_{D\to E}(a,e)$

bucket A: $p(a)\quad \lambda_{E\to A}(a)$

OPT

W*=4
"induced width"
(max clique size)



Generating the optimal assignment

- Given BE messages, select optimum config in reverse order

$$\mathbf{b}^* = \arg\max_b \, p(b|a^*)\, p(d^*|b,a^*)\, p(e^*|b,c^*)$$

$$\mathbf{c}^* = \arg\max_c \, p(c|a^*)\, \lambda_{B\to C}(a^*, c, d^*, e^*)$$

$$\mathbf{d}^* = \arg\max_d \, \lambda_{C\to D}(a^*, d, e^*)$$

$$\mathbf{e}^* = \arg\max_e \, \mathbb{1}[e=0]\, \lambda_{D\to E}(a^*, e)$$

$$\mathbf{a}^* = \arg\max_a \, p(a) \cdot \lambda_{E\to A}(a)$$

$$\begin{tikzcd}
\textbf{B:} & \underbrace{p(b|a)\;\; p(d|b,a)\;\; p(e|b,c)} \arrow[d] \\
\textbf{C:} & \underbrace{p(c|a) \qquad \lambda_{B\to C}(a,c,d,e)} \arrow[d] \\
\textbf{D:} & \underbrace{\lambda_{C\to D}(a,d,e)} \arrow[d] \\
\textbf{E:} & \underbrace{\mathbb{1}[e=0] \quad \lambda_{D\to E}(a,e)} \arrow[d, dash] \\
\textbf{A:} & \underbrace{p(a)\;\; \lambda_{E\to A}(a)} \arrow[d] \\
 & \textbf{OPT = optimal value}
\end{tikzcd}$$

Return optimal configuration (a*,b*,c*,d*,e*)

Complexity of bucket elimination

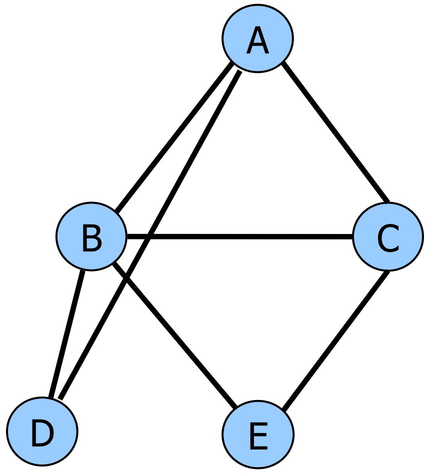
Bucket-Elimination is **time** and **space**

$$O\left(r \exp(w_d^*)\right)$$

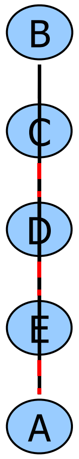
w_d^* : the induced width of the primal graph along ordering d

r = number of functions

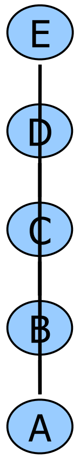
The effect of the ordering:



primal graph



$w^*(d_1) = 4$

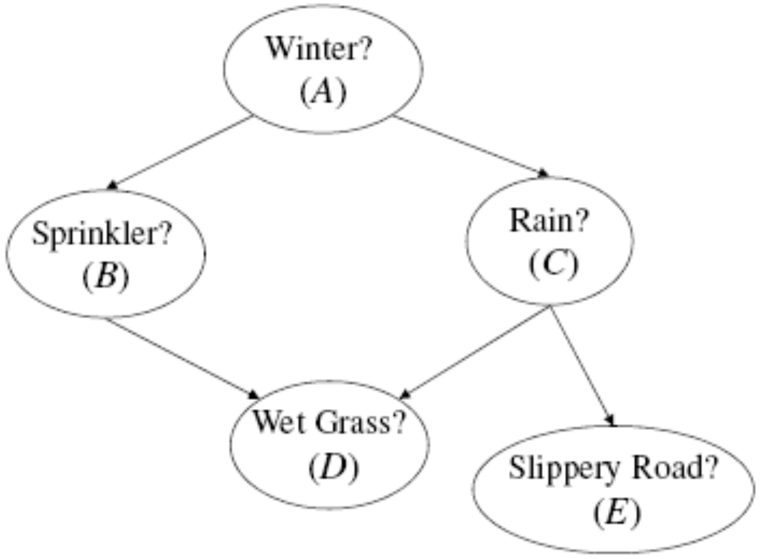


$w^*(d_2) = 2$

Finding smallest induced-width is hard!

A Bayesian Network

Example with mpe?



A	Θ_A
true	.6
false	.4

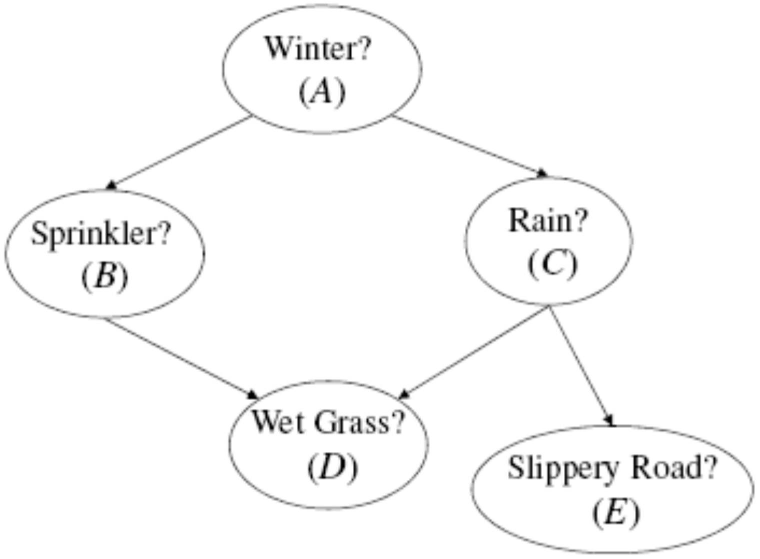
A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Try to compute MPE when E=0



A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

Complexity of bucket-elimination

- **Theorem:**

BE is O(n exp(w*+1)) time and O(n exp(w*)) space, when w* is the induced-width of the moral graph along d when evidence nodes are processed (edges from evidence nodes to earlier variables are removed.)

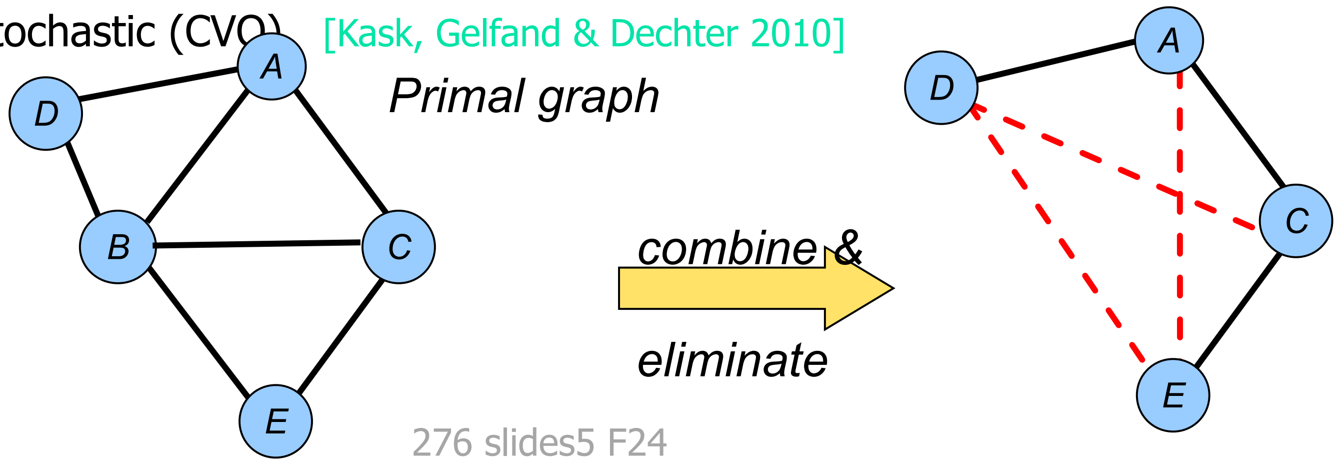
More accurately: O(r exp(w*(d)) where r is the number of CPTs.
For Bayesian networks r=n. For Markov networks?

Inference for probabilistic networks

- Bucket elimination
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→   Marginal Map)
- Induced-Width (Dechter 3.4,3.5)

Variable ordering heuristics

- What makes a good order?
  - Low induced width
  - Elimination creates a function over neighbors
- Finding the best order is hard (NP-complete!)
  - But we can do well with simple heuristics
    - Min-induced-width, Min-Fill, …
  - Anytime algorithms
    - Search-based [Gogate & Dechter 2003]
    - Stochastic (CVO) [Kask, Gelfand & Dechter 2010]



Min-width ordering

```
MIN-WIDTH (MW)
input: a graph $G = (V, E)$, $V = \{v_1, ..., v_n\}$
output: A min-width ordering of the nodes $d = (v_1, ..., v_n)$.
1.  for $j = n$ to 1 by -1 do
2.          $r \leftarrow$ a node in $G$ with smallest degree.
3.          put $r$ in position $j$ and $G \leftarrow G - r$.
            (Delete from $V$ node $r$ and from $E$ all its adjacent edges)
4.  endfor
```

Proposition: algorithm min-width finds a min-width ordering of a graph

What is the Complexity of MW?

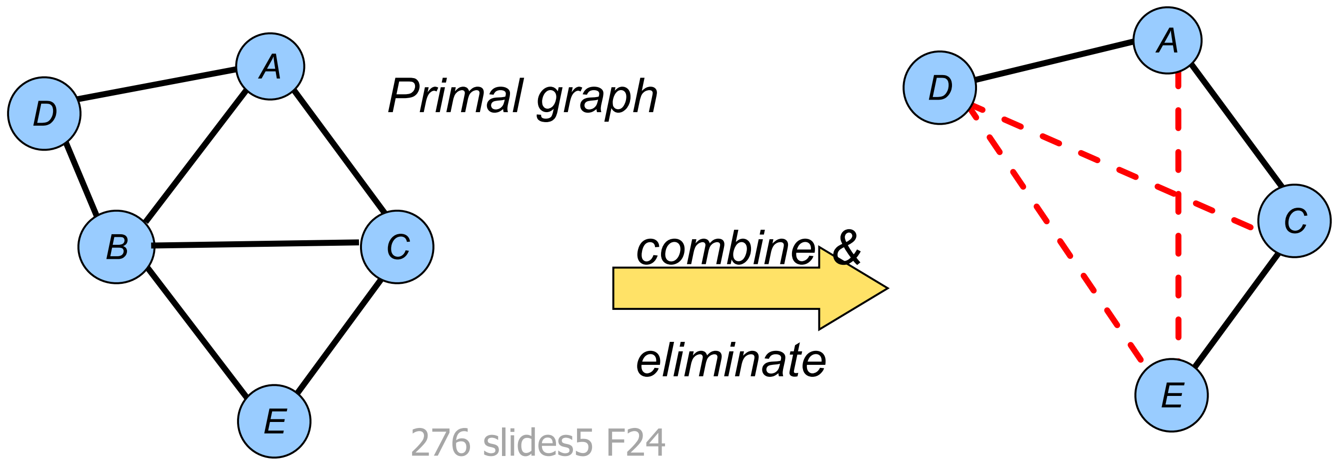
O(e)

Variable ordering heuristics

- Min (induced) width heuristic

```
1. for i=1 to n   ( # of variables)
2.     Select a node $X_i$ with smallest degree as next eliminated
3.     Connect Xi's neighbors:
4.             E = E + { $(X_j, X_k)$ : $(X_i, X_j)$ and $(X_i, X_k)$ in E }
5.     Remove $X_i$ from the graph:  V = V – $\{X_i\}$
6. end
```

("Weighted" version: weight edges by domain size

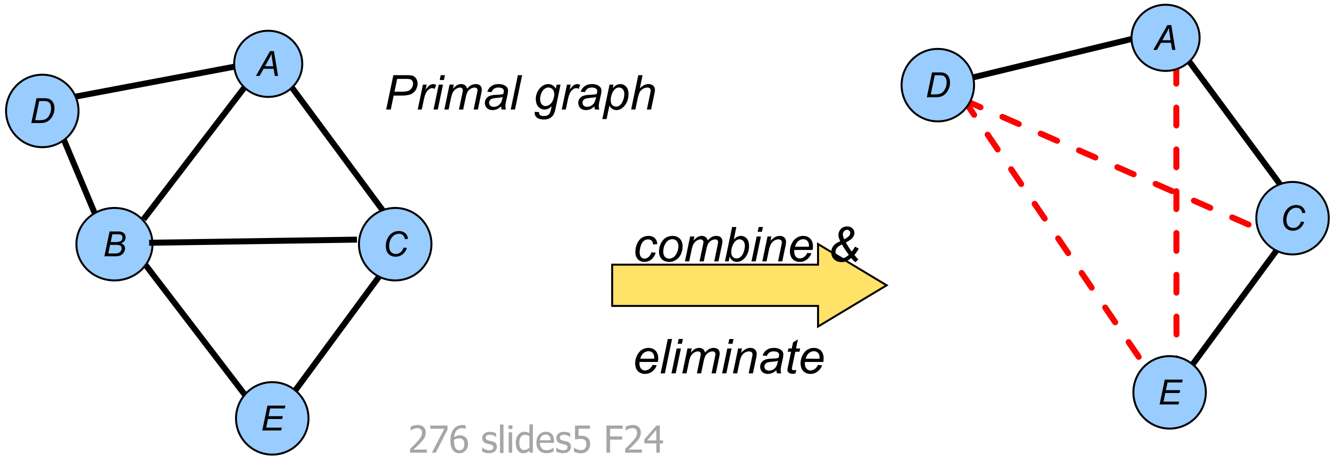


Variable ordering heuristics

- Min fill heuristic

```
1.  for i=1 to n   ( # of variables)
2.      Select a node $X_i$ with smallest "fill edges" as next eliminated
3.      Connect Xi's neighbors:
4.          E = E + { $(X_j, X_k)$ : $(X_i, X_j)$ and $(X_i, X_k)$ in E }
5.      Remove $X_i$ from the graph:  V = V – $\{X_i\}$
6.  end
```

("Weighted" version: weight edges by domain size

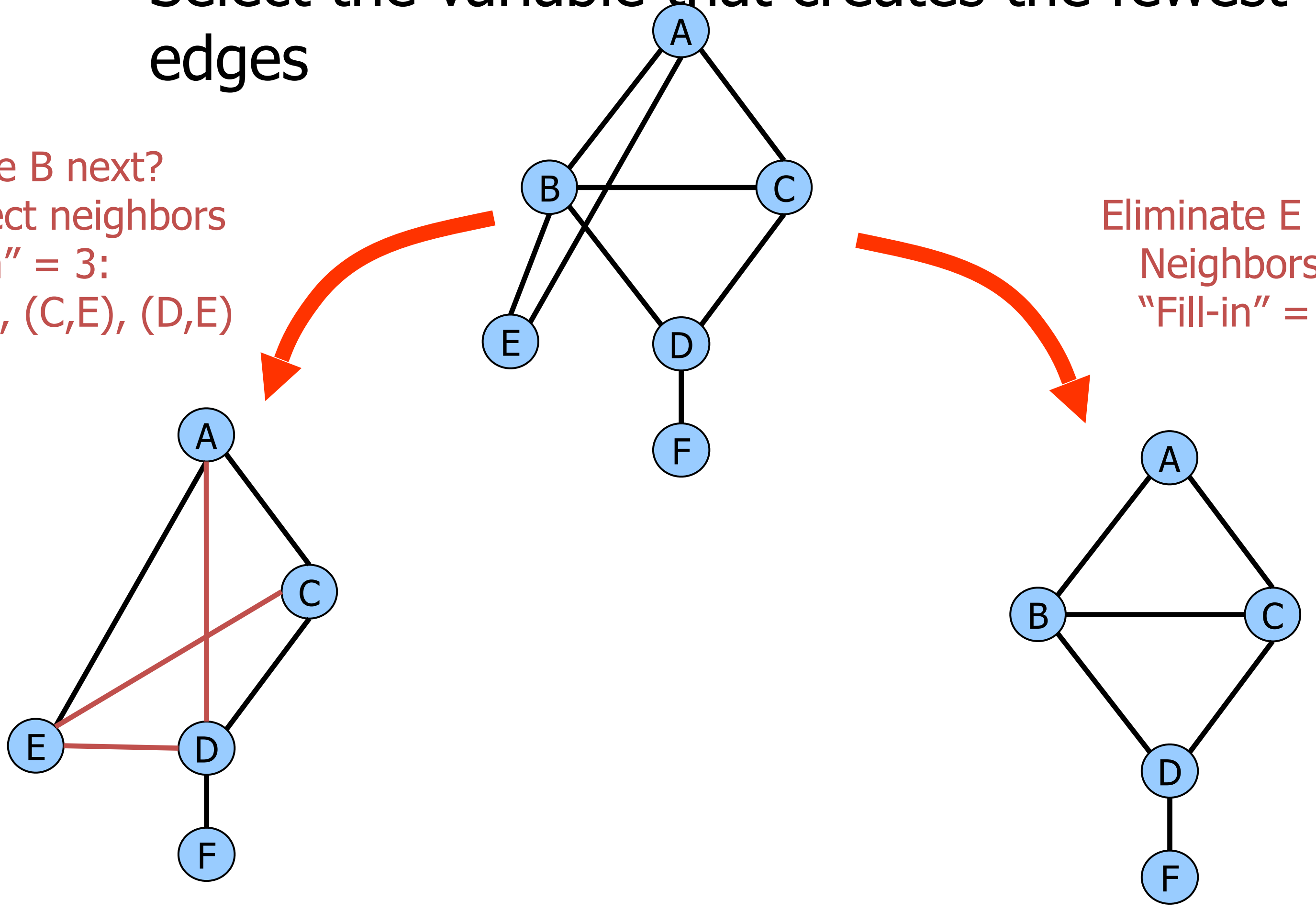


Min-Fill heuristic

- Select the variable that creates the fewest "fill-in" edges

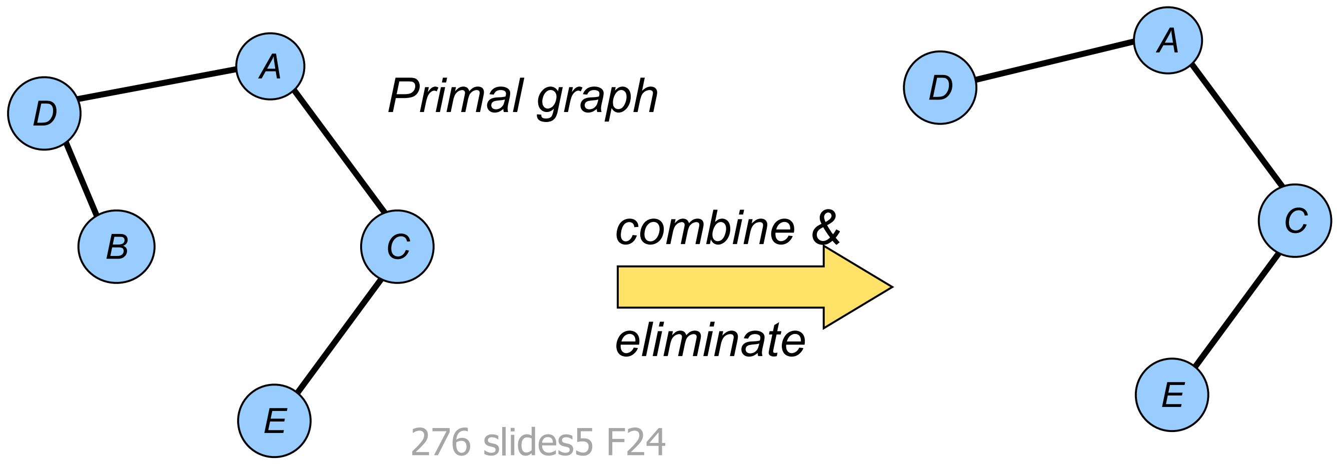
Eliminate B next?
Connect neighbors
"Fill-in" = 3:
(A,D), (C,E), (D,E)

Eliminate E next?
Neighbors already connected
"Fill-in" = 0



Tree-structured graphs

- If the graph is a tree, the best ordering is easy:
  - B, E have only one neighbor; no "fill"
  - Select one to eliminate; remove it
  - Now D or E have only one neighbor; no "fill"…
- Order
  - leaves to root
  - never increases the size of the factors

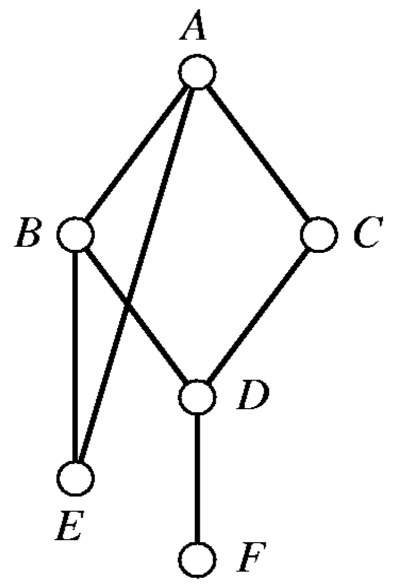


Greedy orderings heuristics

- Min-induced-width
  - From last to first, pick a node with smallest width, then connect parent and remove
- Min-Fill
  - From last to first, pick a node with smallest fill-edges

Complexity? O(n^3)

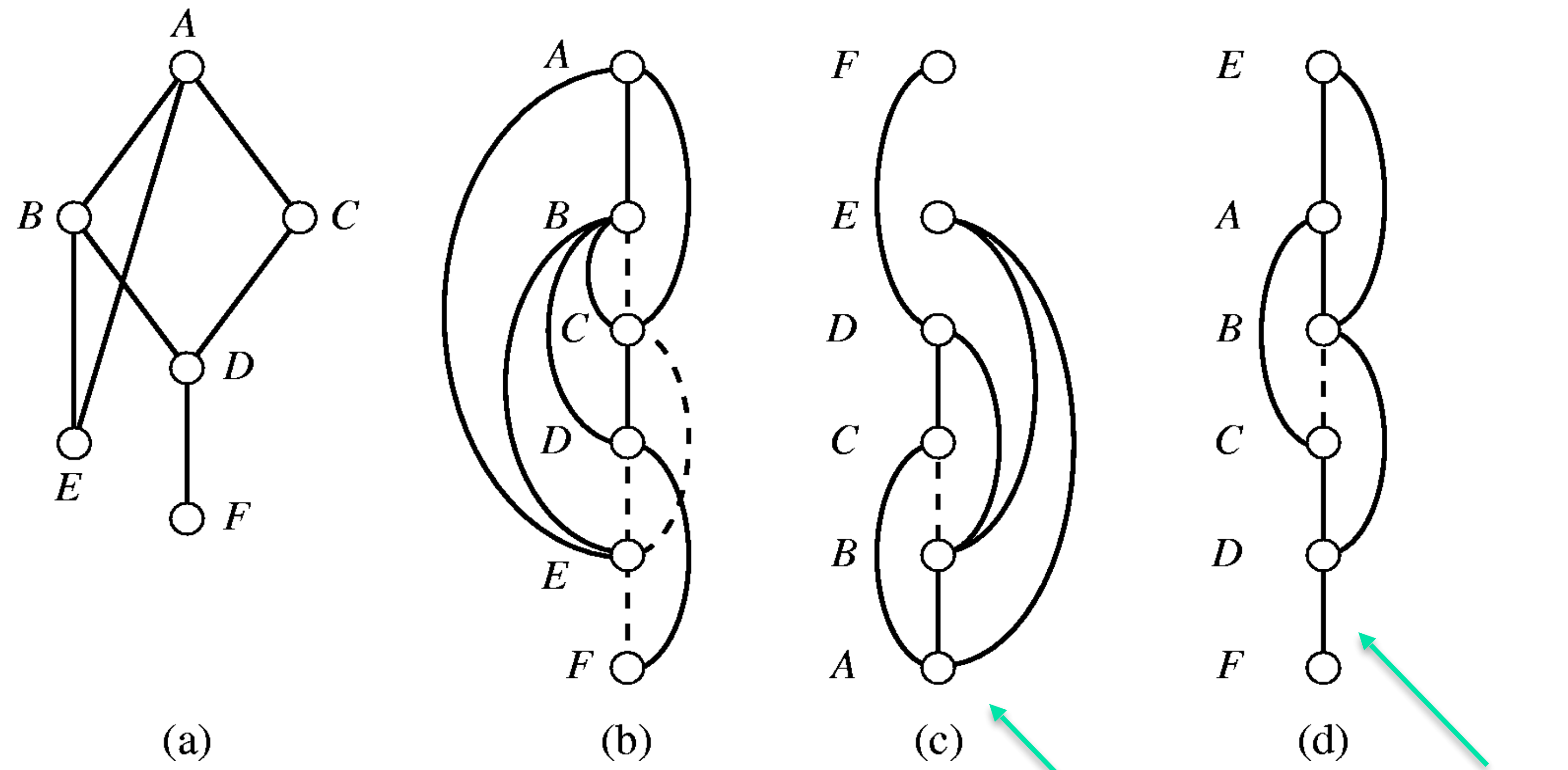
Different induced-graphs



(a)

Let’s find a miw ordering and a min-fill ordering

Different induced-graphs



A Miw ordering

A Min-fill ordering

Which greedy algorithm is best?

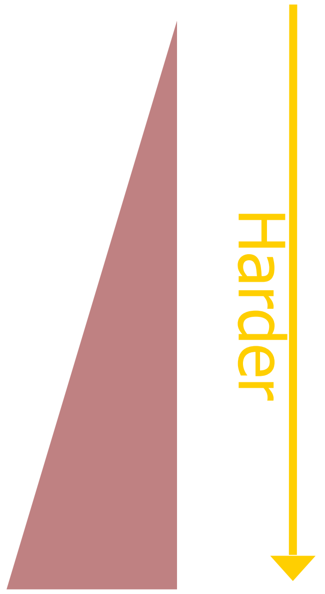
- Min-Fill, prefers a node who adds the least number of fill-in arcs.
- Empirically, fill-in is the best among the greedy algorithms (MW,MIW,MF,MC)
- Complexity of greedy orderings?
- MW is O(e), MIW: O(n^3) MF O(n^3) MC is O(e+n)

Inference for probabilistic networks

- Bucket elimination (Dechter chapter 4)
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→ Marginal Map)
  - Influence diagrams ?
- Induced-Width (Dechter, Chapter 3.4)

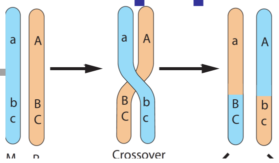
Marginal Map

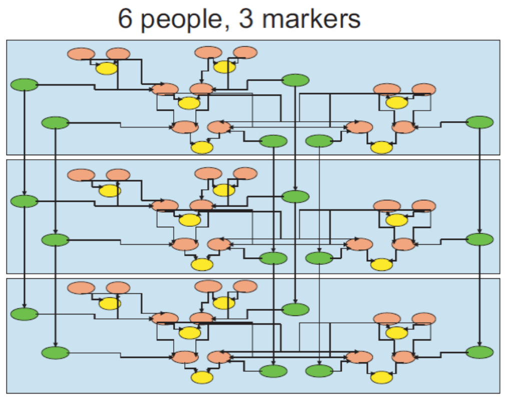
Max-Inference	$f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Sum-Inference	$Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$
Mixed-Inference	$f(\mathbf{x}_M^*) = \max_{\mathbf{x}_M} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$



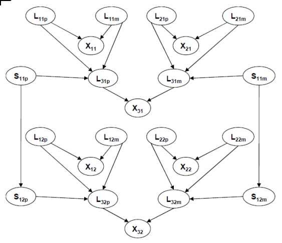
- **NP-hard**: exponentially many terms

Example for MMAP Applications





- Haplotype in Family pedigrees



- Coding networks

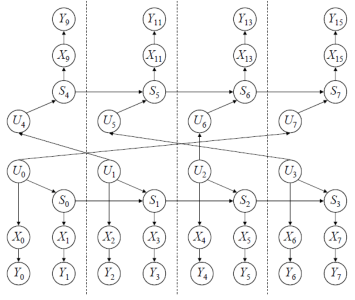
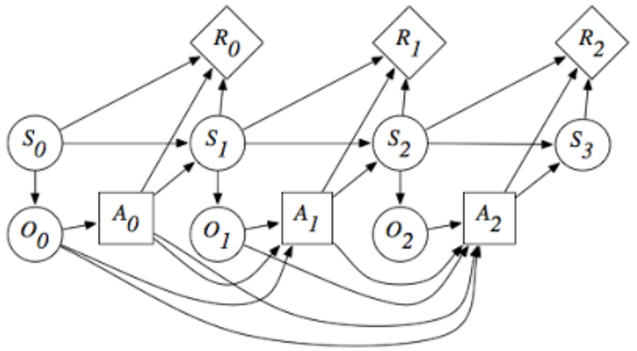
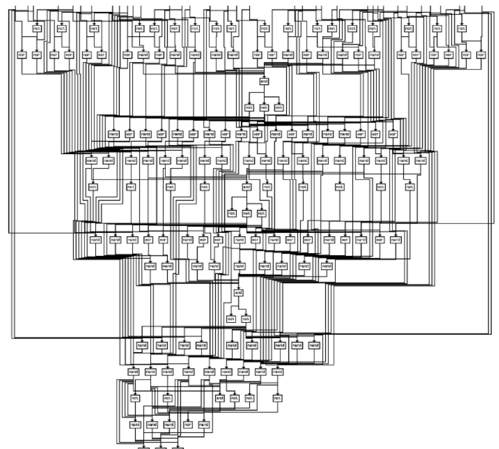


Figure 5.24: A Bayesian network for a turbo code.

- Probabilistic planning

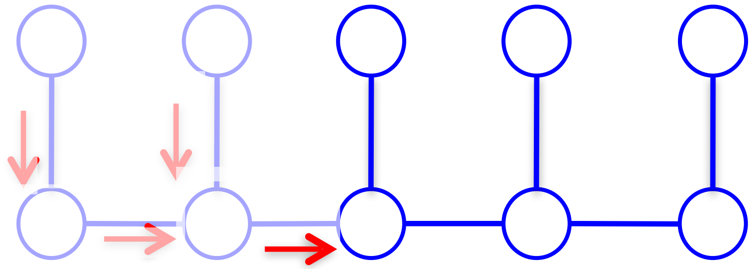


- Diagnosis



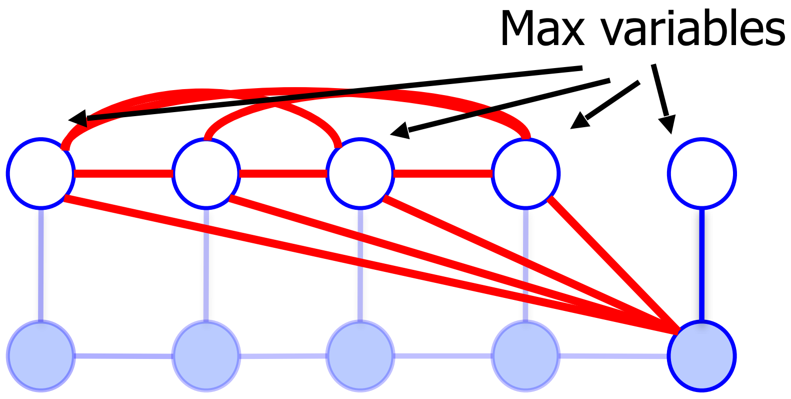
Marginal MAP is not easy on trees

- Pure MAP or summation tasks
  - Dynamic programming
  - Ex: efficient on trees



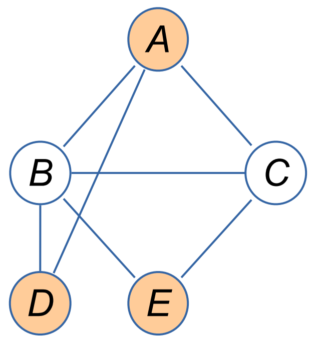
- Marginal MAP
  - Operations do not commute:
  - Sum must be done first!

$$\sum \max \neq \max \sum$$



Bucket elimination for MMAP

Bucket Elimination



$\mathbf{X}_M = \{A, D, E\}$

$\mathbf{X}_S = \{B, C\}$

$$\max_{\mathbf{X}_M} \sum_{\mathbf{X}_S} P(\mathbf{X})$$

constrained elimination order

SUM

B: $\underbrace{f(A,B)\, f(B,C)\, f(B,D) f(B,E)}$

Σ_B

C: $\underbrace{\lambda^B(A,C,D,E) f(A,C)\; f(C,E)}$

Σ_C

MAX

D: $\underbrace{\lambda^C(A,D,E)\; f(A,D)}$

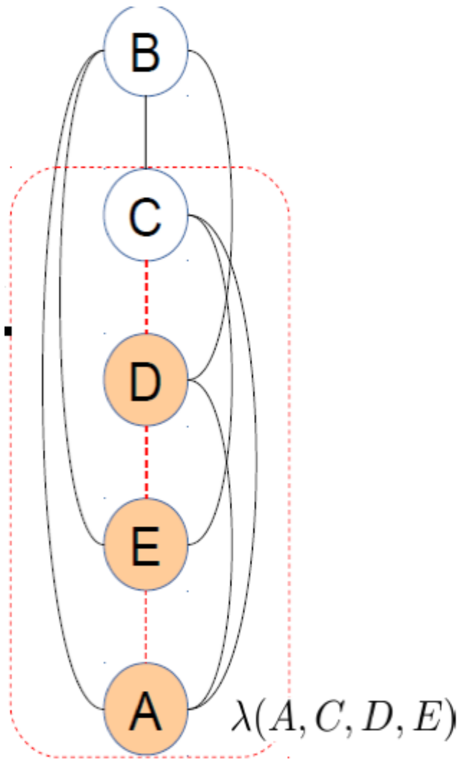
max_D

E: $\underbrace{\lambda^D(A,E)}$

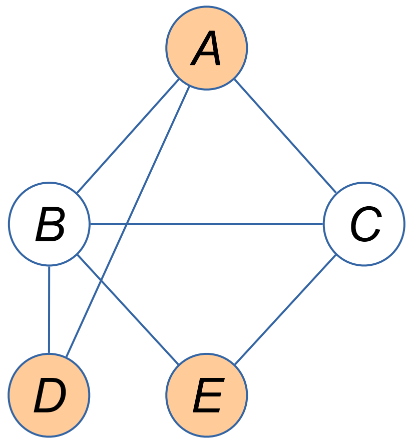
max_E

A: $\underbrace{\lambda^E(A)}$

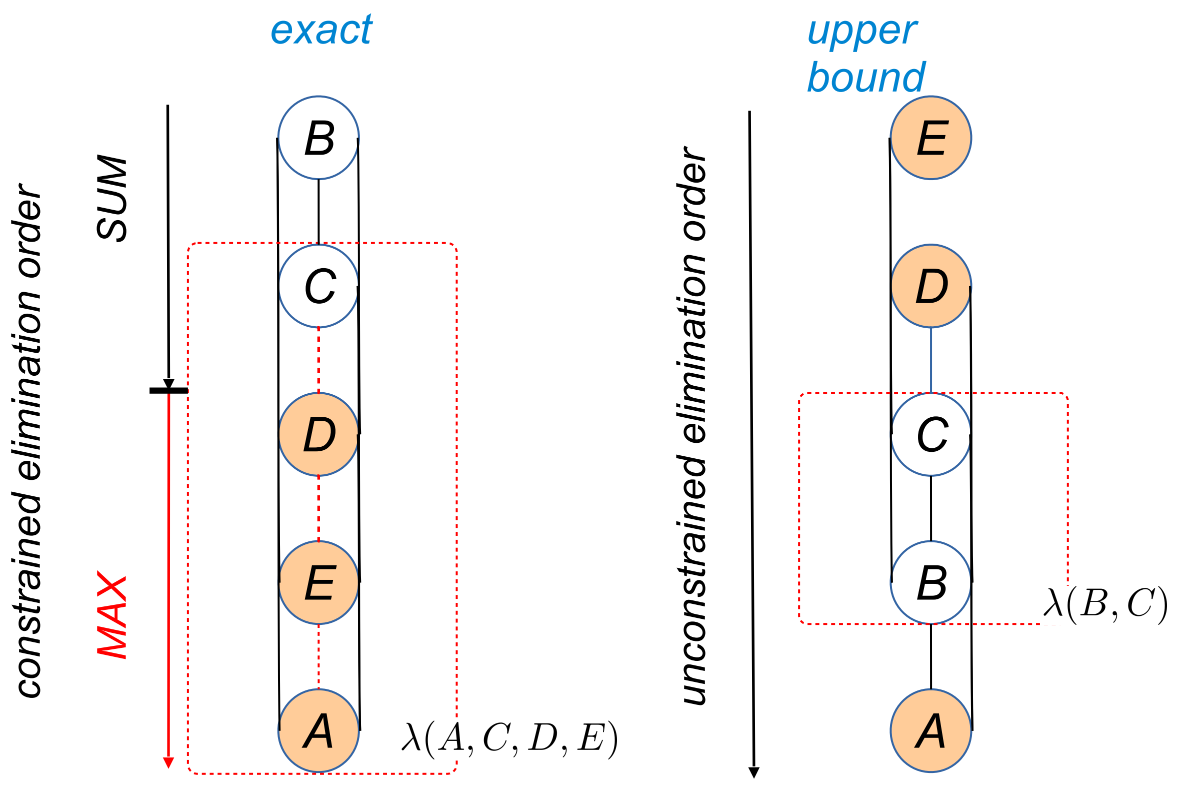
MAP is the marginal MAP value*



Why is MMAP harder?



$\mathbf{X}_M = \{A, D, E\}$
$\mathbf{X}_S = \{B, C\}$



$w^* = 4$ $w^* = 2$

In practice, constrained induced is much larger!

$$\max_X \sum_Y \phi \le \sum_Y \max_X \phi$$

(Park & Darwiche, 2003)
(Yuan & Hansen, 2009)

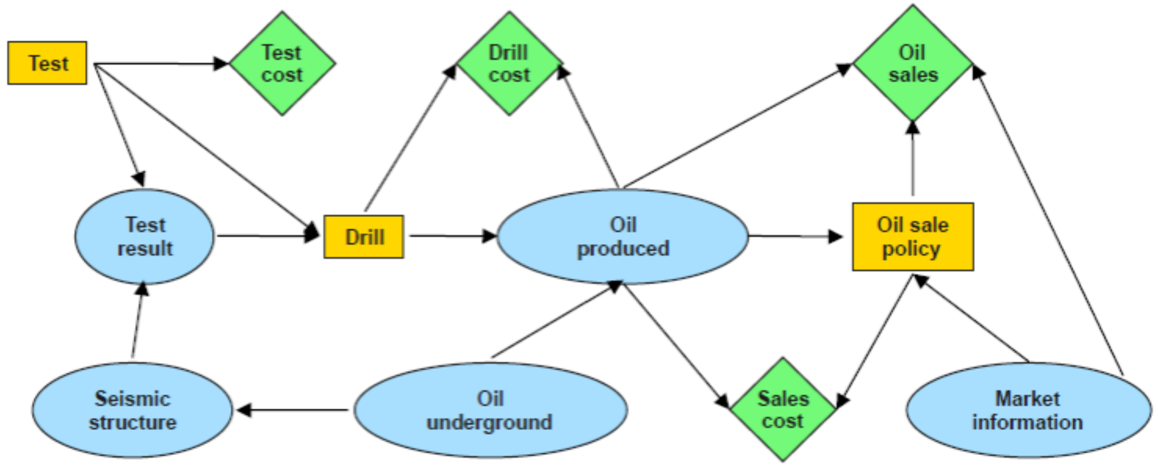
Inference for probabilistic networks

- Bucket elimination (Dechter chapter 4)
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→ Marginal Map)
- Induced-Width (Dechter, Chapter 3.4)
- Mixed networks
- Influence diagrams ?

Ex: “oil wildcatter”

e.g., [Raiffa 1968; Shachter 1986]

- Influence diagram:



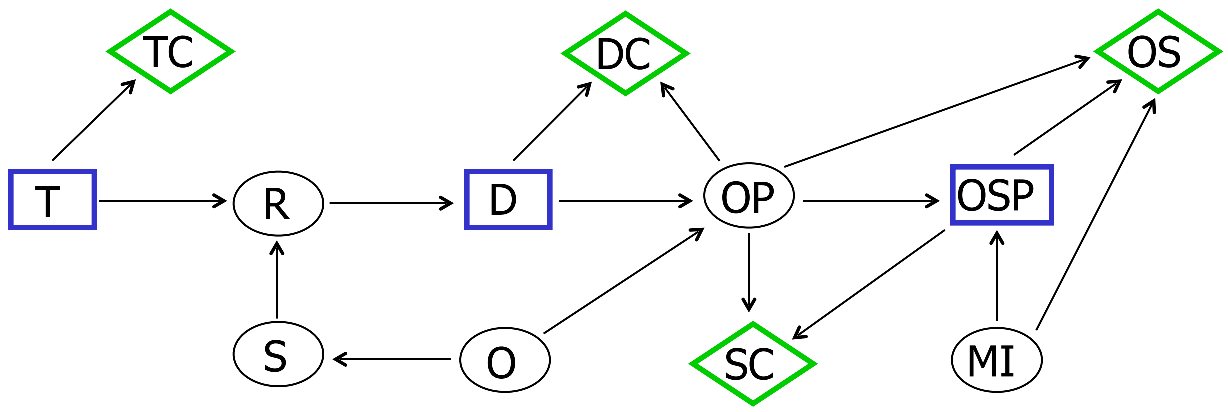
- Three actions: test, drill, sales policy
- Chance variables:

  P(oil) P(seismic|oil) P(result | seismic, test) P(produced | oil, drill) P(market)
- Utilities capture costs of actions, rewards of sale

  Oil sales - Test cost - Drill cost - Sales cost

Influence Diagrams

***Influence diagram** ID = (X,D,P,R).*



Chance variables ***over domains.***

Decision variables

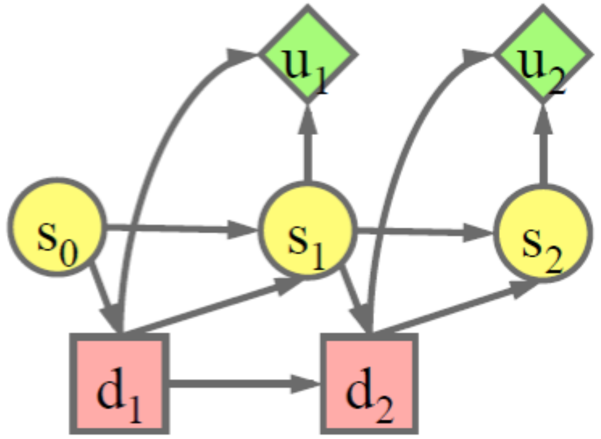
CPT's for chance variables

Reward components

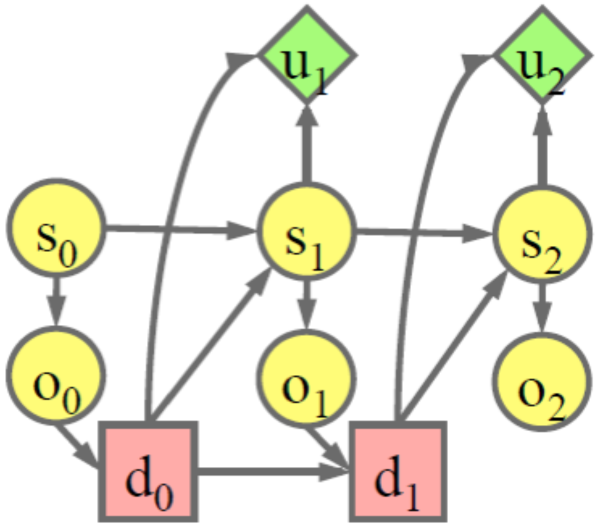
Utility function

Common examples

- Markov decision process
  - Markov chain state sequence
  - Actions “di” influence state transition
  - Rewards based on action, new state
  - Temporally homogeneous



- Partially observable MDP
  - Hidden Markov chain state sequence
  - Generate observations
  - Actions based on observations



Influence Diagrams (continue)

A decision rule *for is a mapping:*

where is the cross product of domains in S.

A policy *is a list of decision rules*

Task: *Find an optimal policy that maximizes the expected utility.*

The Car Example (Howard 1976)

A car buyer needs to buy one of two used cars. The buyer can carry out tests with various costs, and then, decide which car to buy.

T: Test variable (t_0, t_1, t_2) (t_1 test car 1, t_2 test car 2)

D: the decision of which car to buy, $D \in \{buy1, buy2\}$

C_i: the quality of car i, $C_i \in \{q_1, q_2\}$

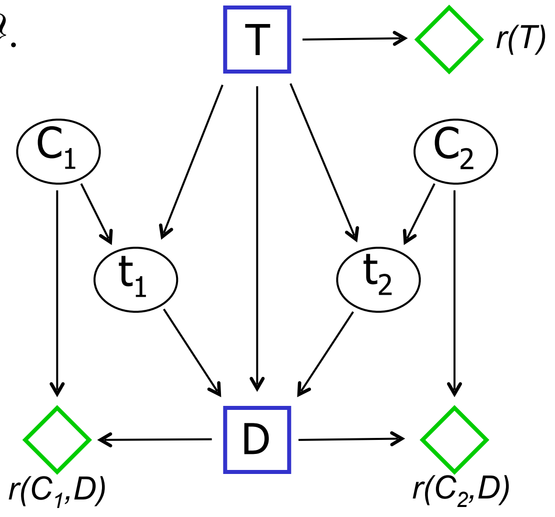
t_i: the outcome of the test on car i, $t_i \in \{pass, fail, null\}$.

$r(T)$: The cost of testing,

$r(C_1,D)$, $r(C_2,D)$: the reward in buying cars 1 and 2.

The utility is: $r(T) + r(C_1,D) + r(C_2,D)$.

Task: determine decision rules T and D such that:



Bucket Elimination for meu (Algorithm Elim-meu-id)

Input: *An Influence diagram* $ID = \{P_1, \ldots, P_n, r_1, \ldots, r_j\}$

Output: *Meu and optimizing policies.*

1. ***Order the variables and partition into buckets.***
2. ***Process buckets from last to first:***

$o = T, t_2, t_2, D, C_2, C_1$

$bucket(C_1): \underbrace{P(C_1), P(t_1|C_1,T), r(C_1,D)}$

$bucket(C_2): \underbrace{P(C_2), P(t_2|C_2,T), r(C_2,D)}$

$bucket(D): \underbrace{\theta_{C_1}(t_1,T,D), \theta_{C_2}(t_2,T,D)}$

$bucket(t_1): \underbrace{\lambda_{C_1}(t_1,T) \quad \theta_D(t_1,t_2,T),} \delta(t_1,t_2,T)$

$bucket(t_2): \underbrace{\lambda_{C_2}(t_2,T) \quad \theta_{t_1}(t_2,T)}$

$bucket(T): r(T) \quad \underbrace{\lambda_{t1}(T) \quad \lambda_{t2}(T) \quad \theta_{t1}(T)}$

$\theta_{T'}, \delta_T$

3. ***Forward:*** *Assign values in ordering* d

The Bucket Description

Final buckets: *(λs or Ps) utility components (θ's or r's).*

$bucket(C_1)$: $P(C_1), P(t_1|C_1,T), r(C_1,D)$

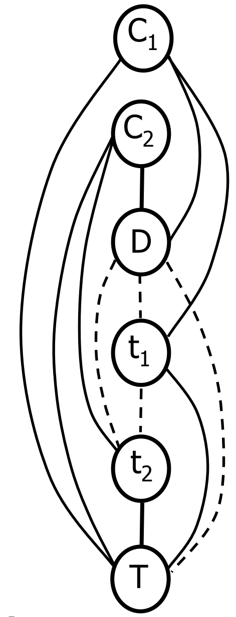
$bucket(C_2)$: $P(C_2), P(t_2|C_2,T), r(C_2,D)$

$bucket(D)$:

$bucket(t_1)$:

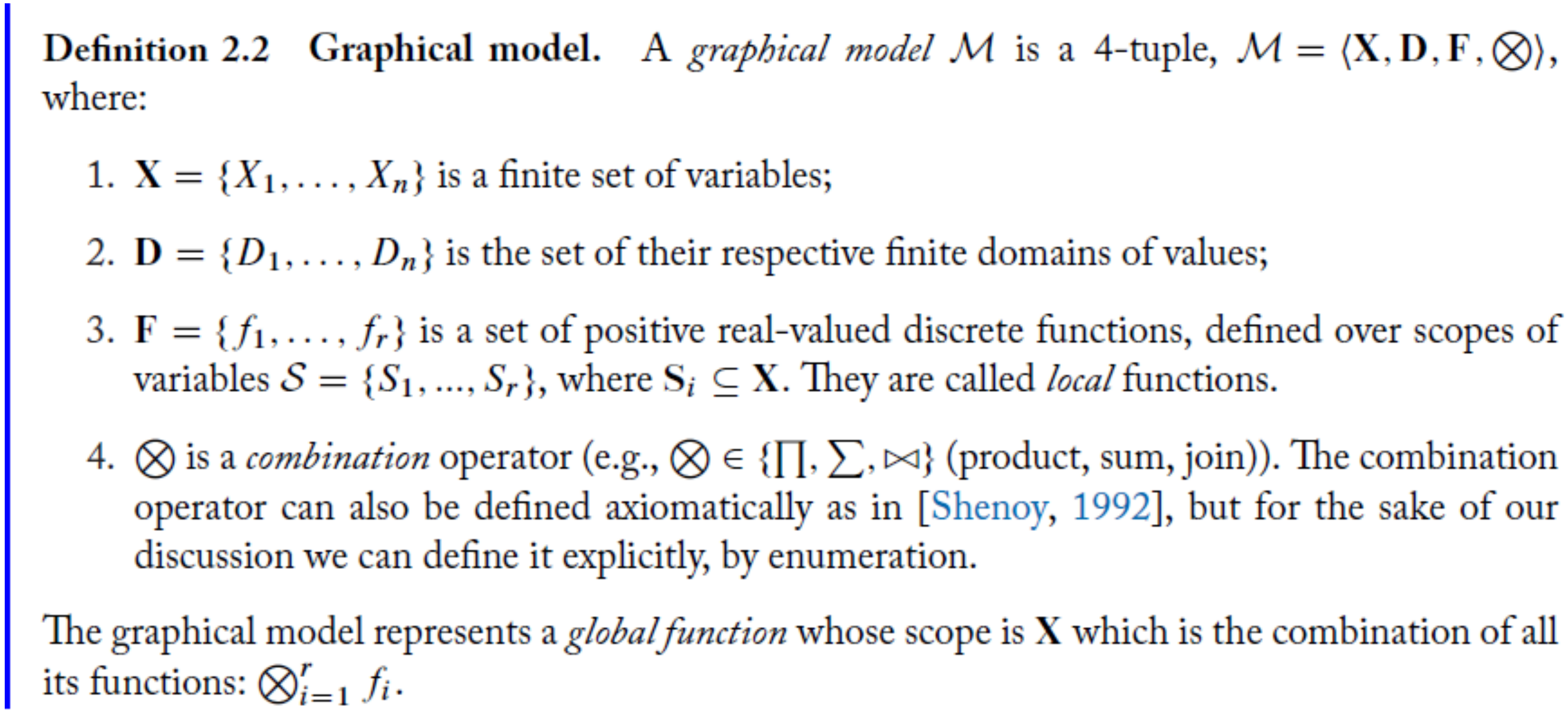
$bucket(t_2)$:

$bucket(T)$: $r(T)$



Optimizing policies: *is argmax of* *computed in* $bucket(T)$, *and* *in* $bucket(t_1)$.

General Graphical Models

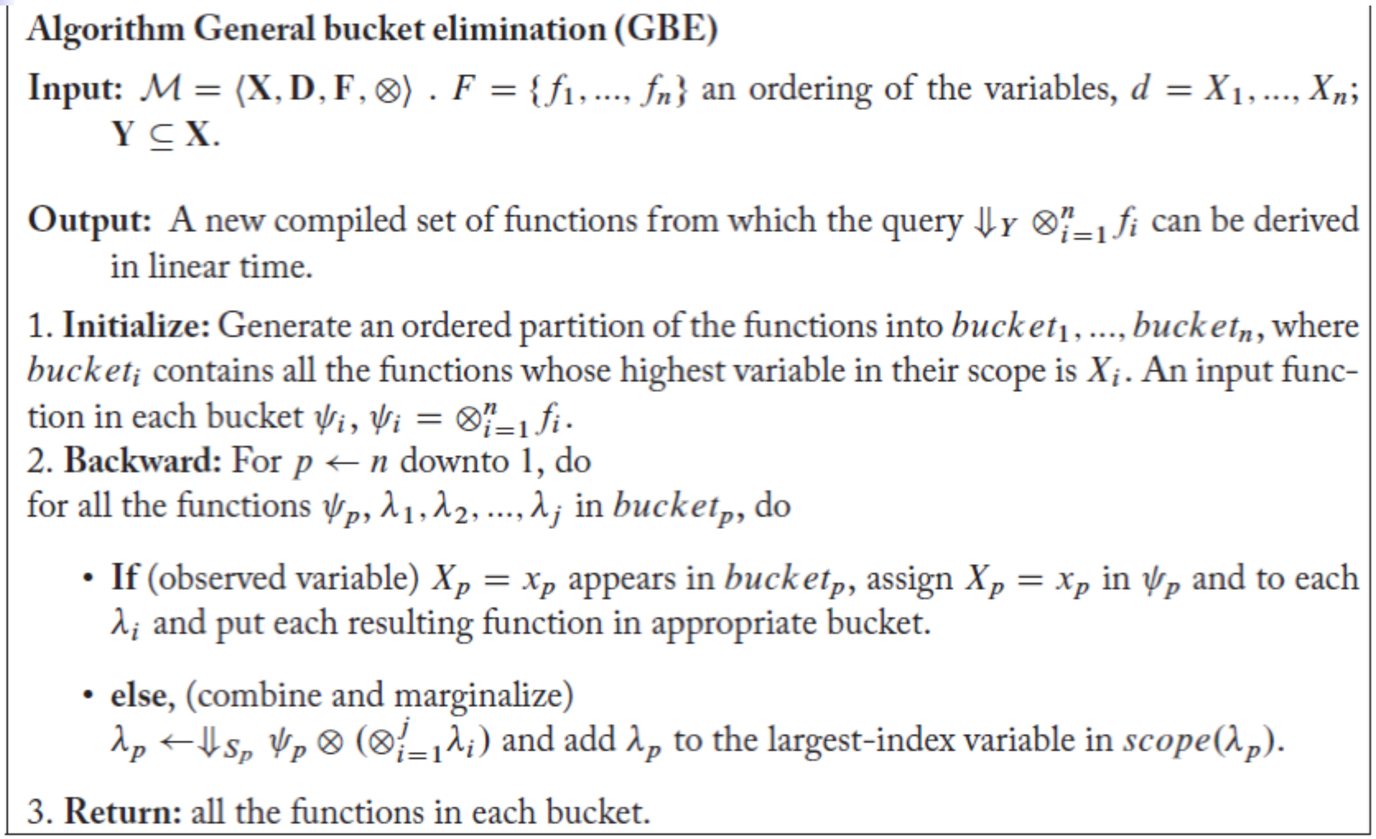


Definition 2.2 Graphical model. A *graphical model* $\mathcal{M}$ is a 4-tuple, $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \bigotimes \rangle$, where:

1. $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a finite set of variables;
2. $\mathbf{D} = \{D_1, \ldots, D_n\}$ is the set of their respective finite domains of values;
3. $\mathbf{F} = \{f_1, \ldots, f_r\}$ is a set of positive real-valued discrete functions, defined over scopes of variables $\mathcal{S} = \{S_1, ..., S_r\}$, where $\mathbf{S}_i \subseteq \mathbf{X}$. They are called *local* functions.
4. $\bigotimes$ is a *combination* operator (e.g., $\bigotimes \in \{\prod, \sum, \bowtie\}$ (product, sum, join)). The combination operator can also be defined axiomatically as in [Shenoy, 1992], but for the sake of our discussion we can define it explicitly, by enumeration.

The graphical model represents a *global function* whose scope is $\mathbf{X}$ which is the combination of all its functions: $\bigotimes_{i=1}^{r} f_i$.

General bucket elimination

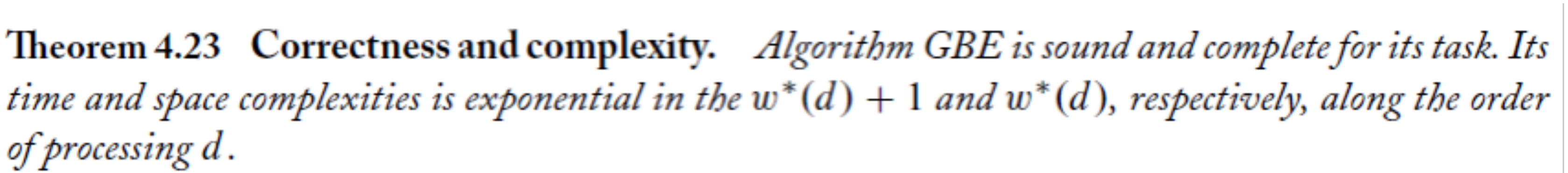


Algorithm General bucket elimination (GBE)

Input: $\mathcal{M} = \langle \mathbf{X}, \mathbf{D}, \mathbf{F}, \otimes \rangle$. $F = \{f_1, ..., f_n\}$ an ordering of the variables, $d = X_1, ..., X_n$; $\mathbf{Y} \subseteq \mathbf{X}$.

Output: A new compiled set of functions from which the query $\Downarrow_Y \otimes_{i=1}^n f_i$ can be derived in linear time.

```
1. Initialize: Generate an ordered partition of the functions into $bucket_1, ..., bucket_n$, where
   $bucket_i$ contains all the functions whose highest variable in their scope is $X_i$. An input func-
   tion in each bucket $\psi_i$, $\psi_i = \otimes_{i=1}^n f_i$.
2. Backward: For $p \leftarrow n$ downto 1, do
   for all the functions $\psi_p, \lambda_1, \lambda_2, ..., \lambda_j$ in $bucket_p$, do
      • If (observed variable) $X_p = x_p$ appears in $bucket_p$, assign $X_p = x_p$ in $\psi_p$ and to each
        $\lambda_i$ and put each resulting function in appropriate bucket.
      • else, (combine and marginalize)
        $\lambda_p \leftarrow \Downarrow_{S_p} \psi_p \otimes (\otimes_{i=1}^j \lambda_i)$ and add $\lambda_p$ to the largest-index variable in $scope(\lambda_p)$.
3. Return: all the functions in each bucket.
```



Theorem 4.23 Correctness and complexity. *Algorithm GBE is sound and complete for its task. Its time and space complexities is exponential in the* $w^*(d) + 1$ *and* $w^*(d)$, *respectively, along the order of processing* d.