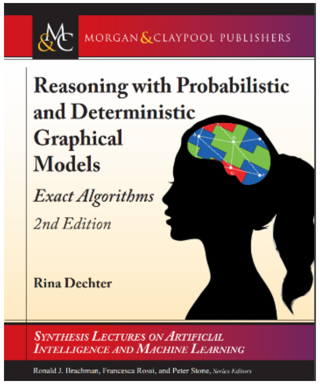
*Algorithms for Reasoning with graphical models*

# Slides Set 7:

# Exact Inference Algorithms Bucket-elimination

*Rina Dechter*

(Dechter1 chapter 4, Darwiche chapter 6)



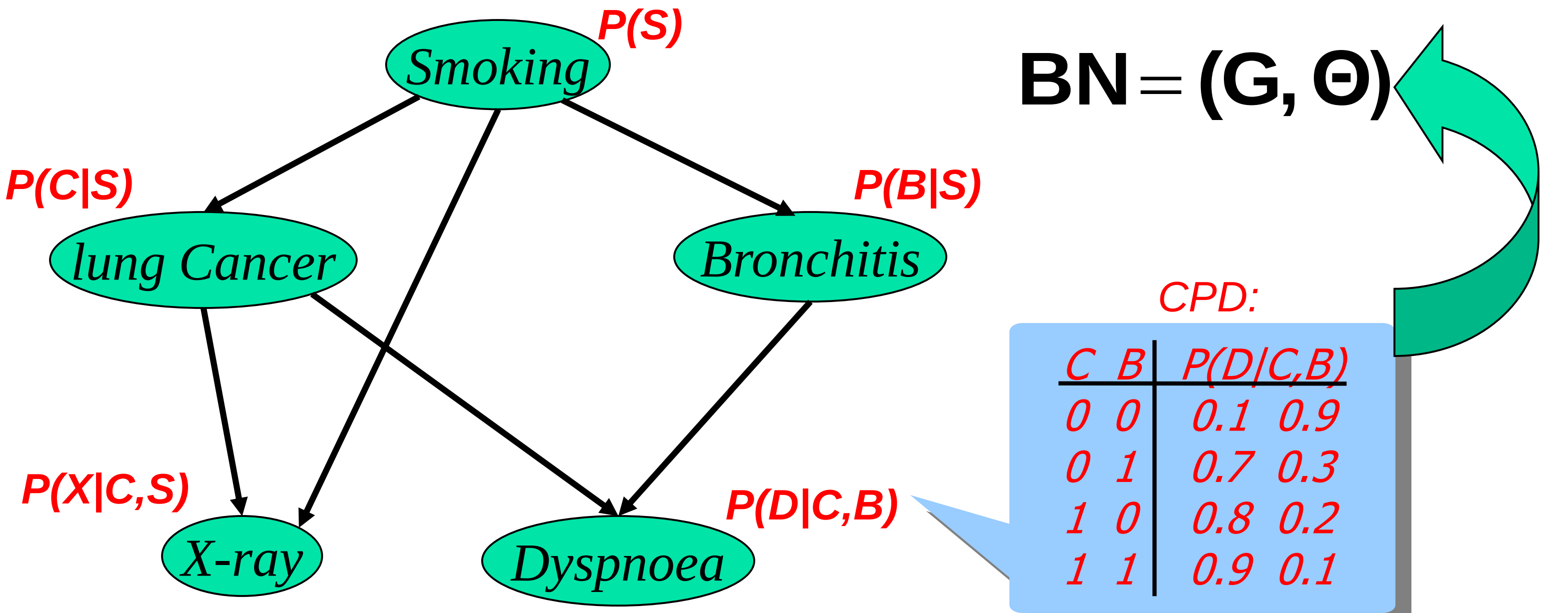
# Inference for probabilistic networks

- Bucket elimination
  - Belief-updating, P(e), partition function
  - Marginals, probability of evidence
  - The impact of evidence
  - for MPE (→MAP)
  - for MAP (→ Marginal Map)
  - Mixed netwroks
- Tree-decomposition schemes
  - Bucket tree elimination
  - Cluster tree elimination

# 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)
  - Mixed netwroks
- Tree-decomposition schemes
  - Bucket tree elimination
  - Cluster tree elimination

# Bayesian Networks: Example (Pearl, 1988)



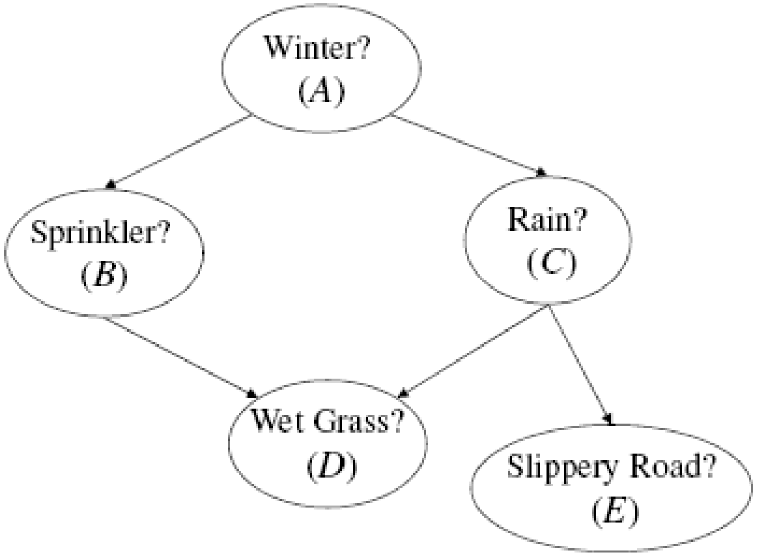

| C | B | P(D\|C,B) | |
|---|---|---|---|
| 0 | 0 | 0.1 | 0.9 |
| 0 | 1 | 0.7 | 0.3 |
| 1 | 0 | 0.8 | 0.2 |
| 1 | 1 | 0.9 | 0.1 |

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

***Belief Updating:***

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

# A Bayesian Network




| $A$ | $\Theta_A$ |
|---|---|
| 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 |

# Graphical Models

A ***graphical model*** *consists of:*

$X = \{X_1, \ldots, X_n\}$ *-- variables*

$D = \{D_1, \ldots, D_n\}$ *-- domains*

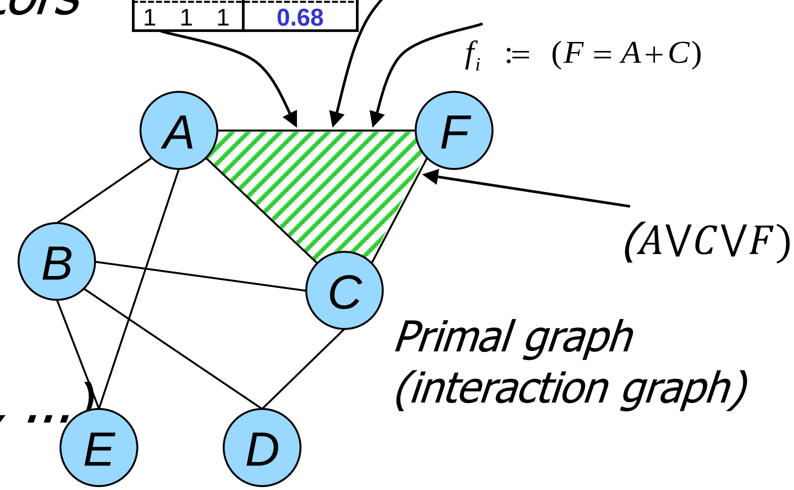
$F = \{f_{\alpha_1}, \ldots, f_{\alpha_m}\}$ *-- functions or "factors"*

*Conditional Probability Table (CPT)*

| A | C | F | P(F\|A,C) |
|---|---|---|---|
| 0 | 0 | 0 | 0.14 |
| 0 | 0 | 1 | 0.96 |
| 0 | 1 | 0 | 0.40 |
| 0 | 1 | 1 | 0.60 |
| 1 | 0 | 0 | 0.35 |
| 1 | 0 | 1 | 0.65 |
| 1 | 1 | 0 | 0.72 |
| 1 | 1 | 1 | 0.68 |

*Relation*

| A | C | F |
|---|---|---|
| red | green | blue |
| blue | red | red |
| blue | blue | green |
| green | red | blue |




***Operators:***

*combination operator*
*(sum, product, join, ...)*

*elimination operator*
*(projection, sum, max, min, ...)*

***Types of queries:***

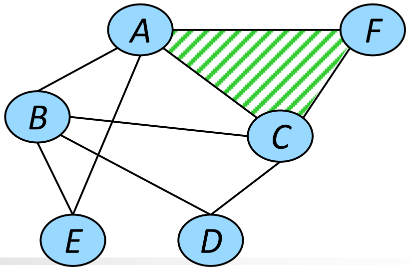
*Marginal:* $Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

*MPE / MAP:* $f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

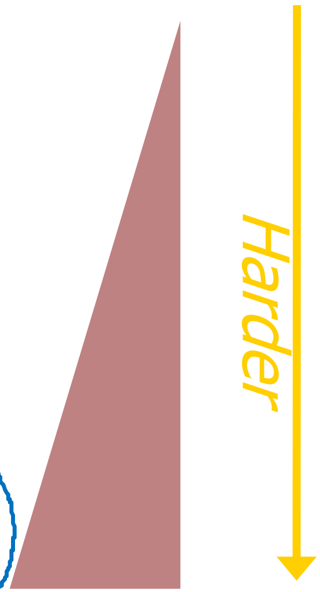
*Marginal MAP:* $f(\mathbf{x}_M^*) = \max_{\mathbf{x_M}} \sum_{\mathbf{x}_S} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$

- *All these tasks are NP-hard*
  - *exploit problem structure*
  - *identify special cases*
  - *approximate*

# Types of queries




| | |
|---|---|
| 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
- We will focus on exact and then on **approximation** algorithms
  - **Anytime**: very fast & very approximate ! Slower & more accurate

# Belief Updating is NP-hard

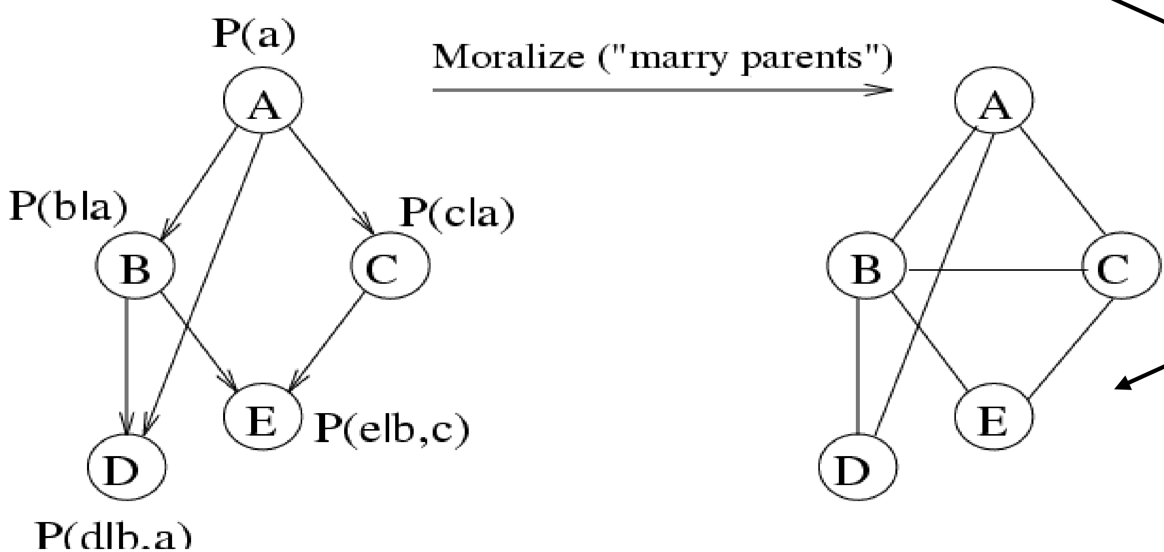
- Each SAT formula can be mapped into a belief updating query in a Bayesian network

- Example

$$(\neg u \vee \neg w \vee y) \wedge (u \vee \neg v \vee w)$$

# “Moral” Graph

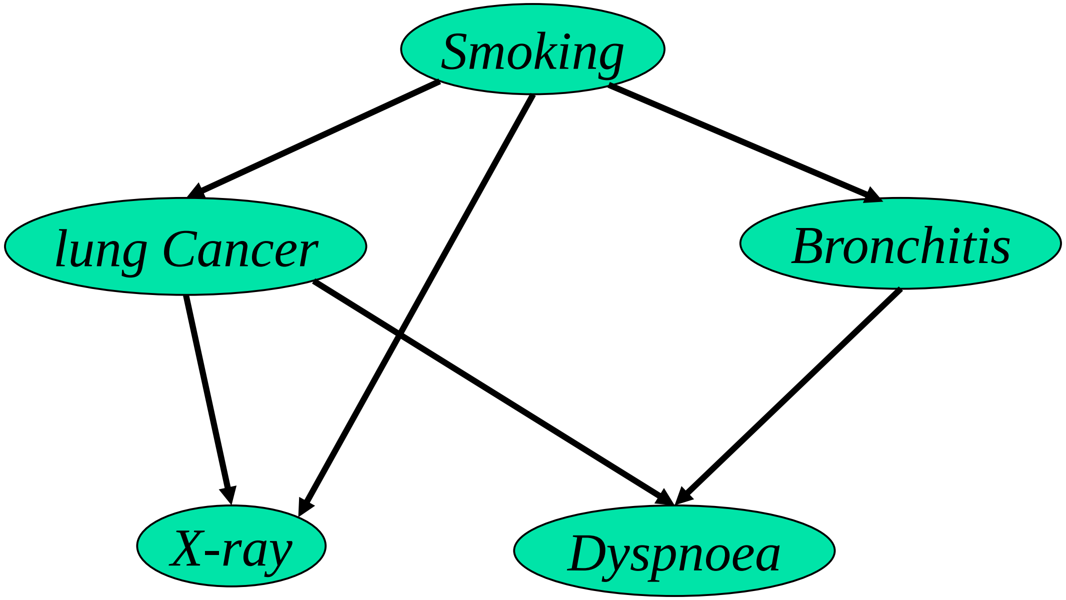
$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} \underbrace{P(X_i \mid parents(X_i))}$$




*Conditional Probability Distribution (CPT)*

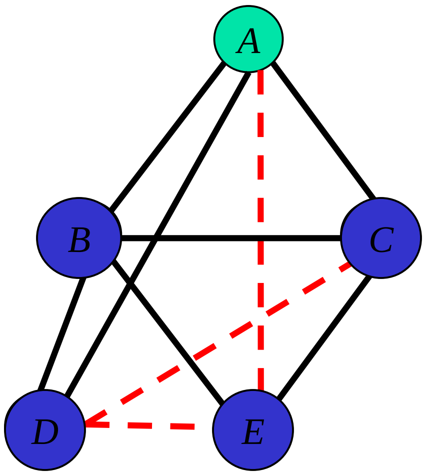
*Clique in moral graph (“family”)*

# Belief Updating




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

# Belief updating: P(X|evidence)=?




"Moral" graph

$$P(a|e=0) \propto P(a,e=0) =$$

$$\sum_{e=0,d,c,b} P(a)P(b|a)P(c|a)P(d|b,a)P(e|b,c) =$$

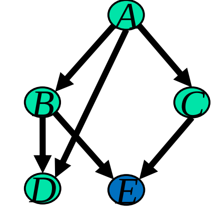
$$P(a)\sum_{e=0}\sum_{d}\sum_{c}P(c|a)\sum_{b}P(b|a)P(d|b,a)P(e|b,c)$$

Variable Elimination

$$h^{B}(a,d,c,e)$$

# Bucket elimination

## Algorithm *BE-bel* (Dechter 1996)




$$P(A \mid E=0) = \alpha \sum_{E=0,D,C,B} P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid A,B) \cdot P(E \mid B,C)$$

$\sum_b \prod$ ← *Elimination operator*

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

bucket C: $P(c|a)$ $\lambda^B(a,d,c,e)$

bucket D: $\lambda^C(a,d,e)$

bucket E: $e=0$ $\lambda^D(a,e)$

bucket A: $P(a)$ $\lambda^E(a)$

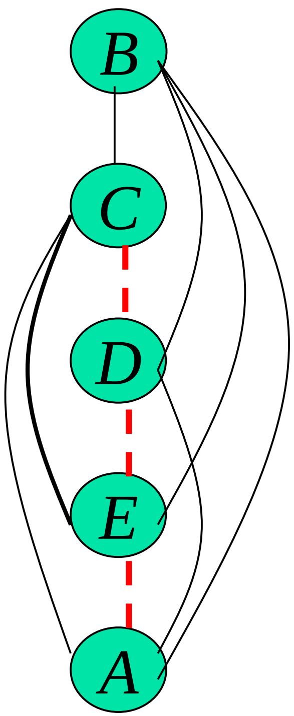
$P(e=0)$

$P(a,e=0)$

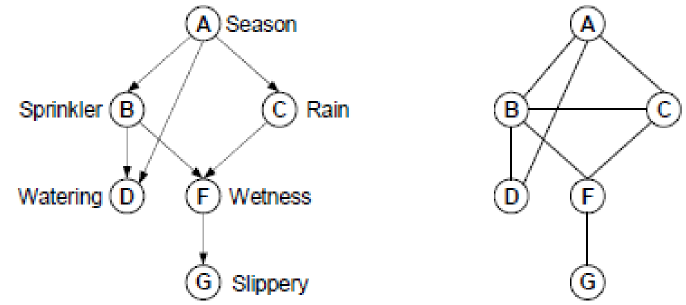
*W*=4*

*"induced width" (max clique size)*

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



# A Bayesian Network
## Ordering: A,C,B,E,D,G




(a) Directed acyclic graph (b) Moral graph

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

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

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

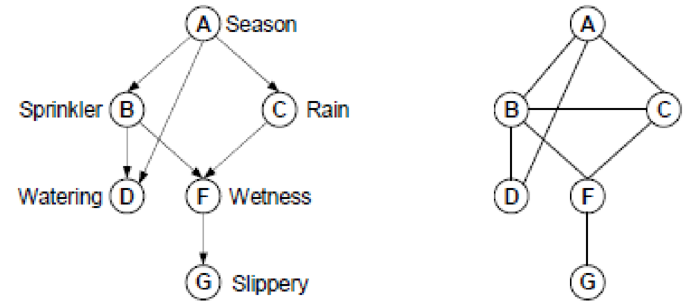
$$P(a, g = 1) = P(a)\sum_{c} P(c|a)\sum_{b} P(b|a)\lambda_D(a, b)\sum_{f} P(f|b, c)\lambda_G(f) \quad (4.3)$$

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

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

# A Bayesian Network

## Ordering: A,C,B,E,D,G




(a) Directed acyclic graph (b) Moral graph

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

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

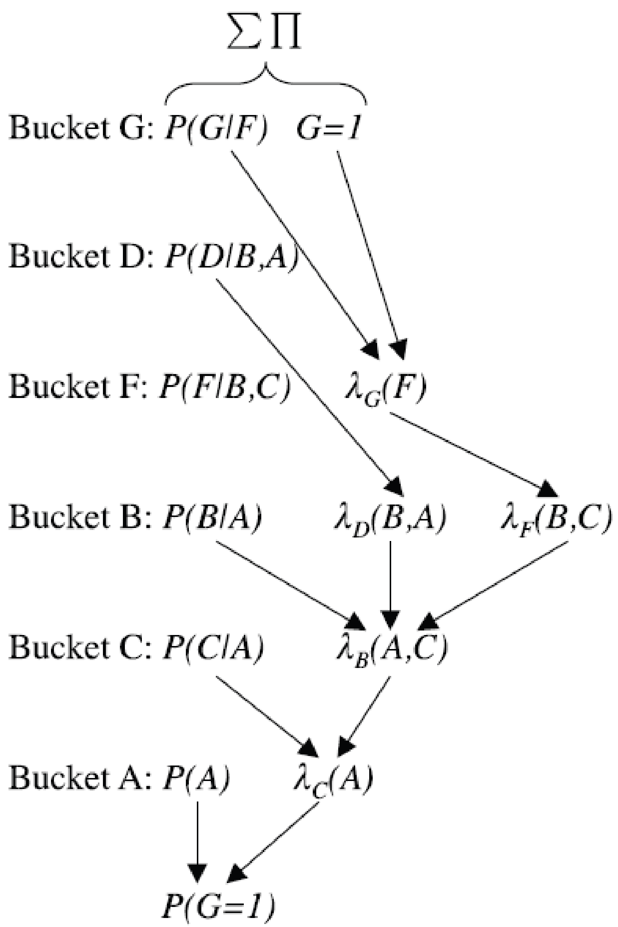
$$P(a, g=1) = P(a)\sum_c P(c|a)\sum_b P(b|a)\sum_f P(f|b,c)\lambda_G(f)\sum_d P(d|b,a). \quad (4.2)$$

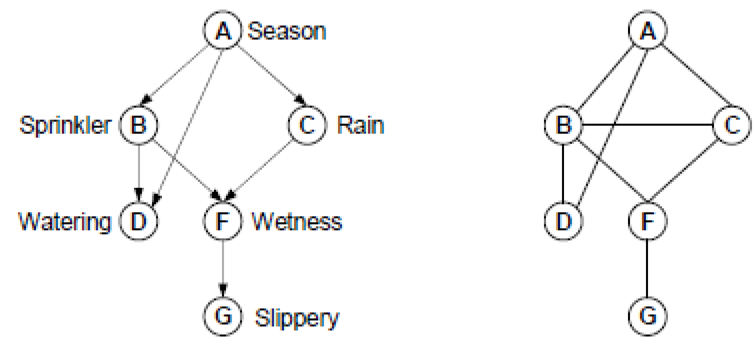
$$P(a, g=1) = P(a)\sum_c P(c|a)\sum_b P(b|a)\lambda_D(a,b)\sum_f P(f|b,c)\lambda_G(f) \quad (4.3)$$

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

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

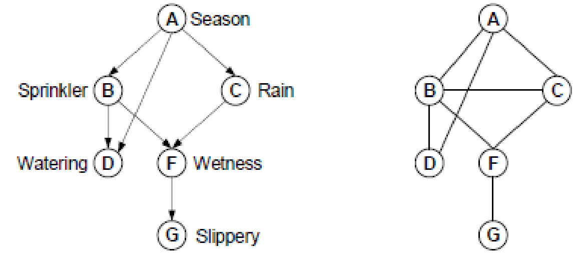
# A Bayesian Network
## Ordering: A,C,B,F,D,G




(a) Directed acyclic graph (b) Moral graph

# A Different Ordering




(a) Directed acyclic graph (b) Moral graph

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

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

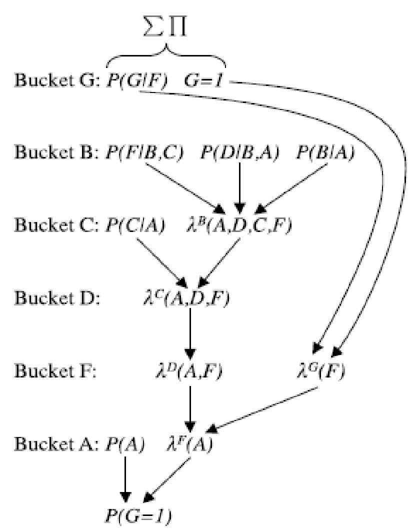
$$= P(a)\sum_f \lambda_G(f) \sum_d \sum_c P(c|a) \sum_b P(b|a)\ P(d|a,b)P(f|b,c)$$

$$= P(a)\sum_f \lambda_G(f) \sum_d \sum_c P(c|a)\lambda_B(a,d,c,f)$$

$$= P(a)\sum_f \lambda_g(f) \sum_d \lambda_C(a,d,f)$$

$$= P(a)\sum_f \lambda_G(f)\lambda_D(a,f)$$

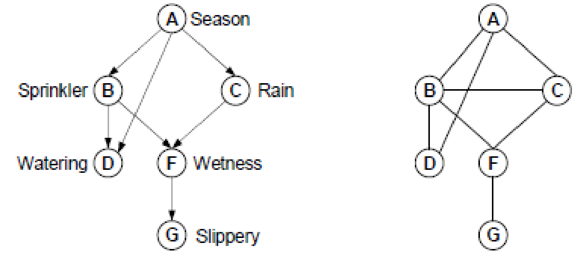
$$= P(a)\lambda_F(a)$$




**(a)**

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

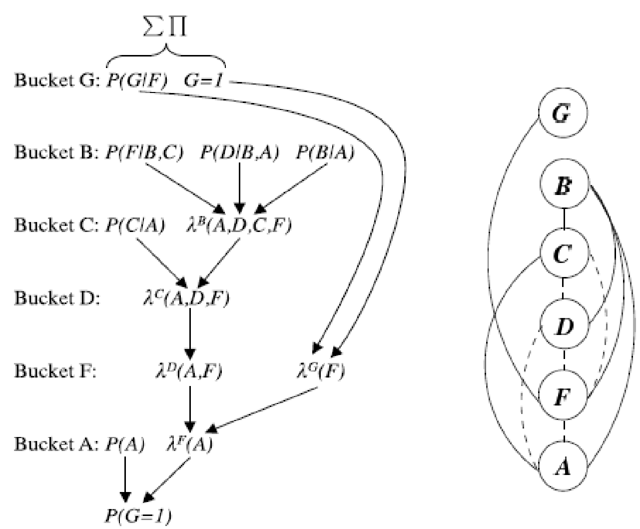
# A Different Ordering




(a) Directed acyclic graph (b) Moral graph

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

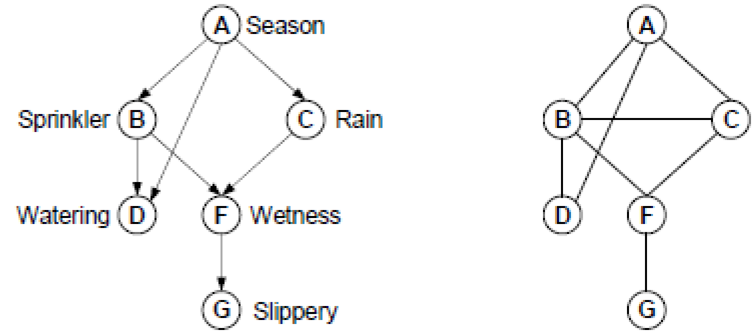
$$P(a, g=1) = P(a)\sum_f \sum_d \sum_c P(c|a)\sum_b P(b|a)\ P(d|a,b)P(f|b,c)\sum_{g=1} P(g|f)$$
$$= P(a)\sum_f \lambda_G(f)\sum_d \sum_c P(c|a)\sum_b P(b|a)\ P(d|a,b)P(f|b,c)$$
$$= P(a)\sum_f \lambda_G(f)\sum_d \sum_c P(c|a)\lambda_B(a,d,c,f)$$
$$= P(a)\sum_f \lambda_g(f)\sum_d \lambda_C(a,d,f)$$
$$=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



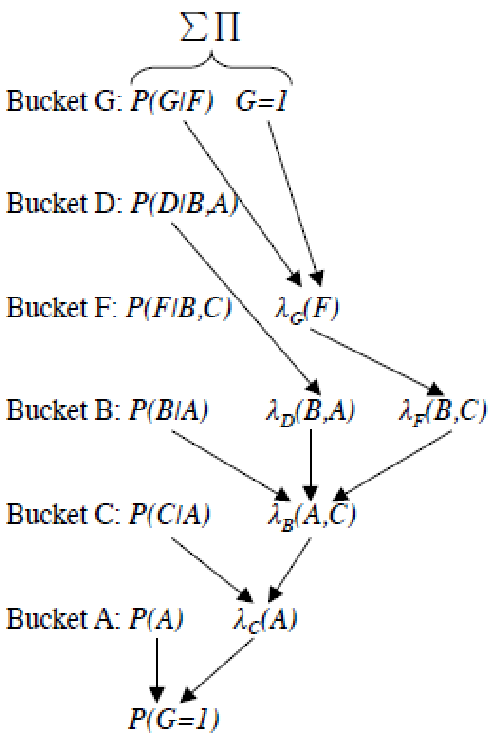

(a) (b)

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

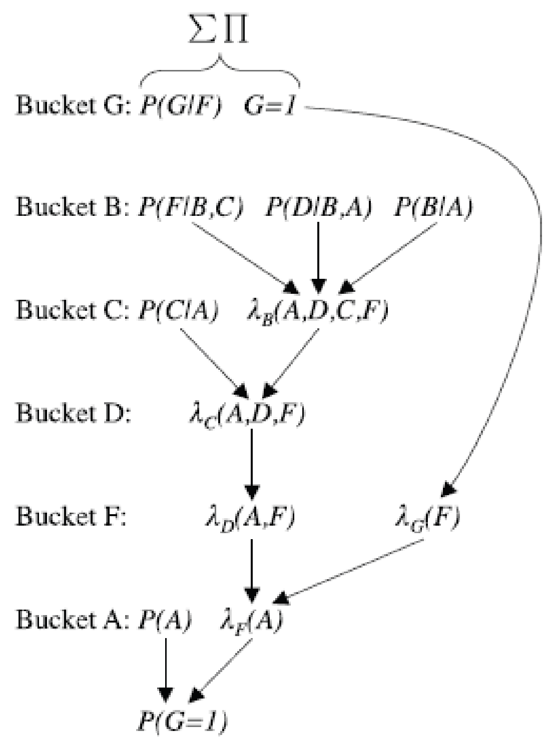
# A Bayesian Network Processed Along 2 Orderings



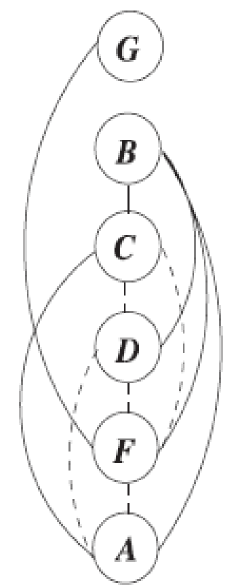

(a) Directed acyclic graph (b) Moral graph




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




**(a)**




**(b)**

**Figure 4.4:** The bucket's output when processing along $d_2 = A, F, D, C, B, G$.

# 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*

# Factors: Sum-Out Operation

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

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

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

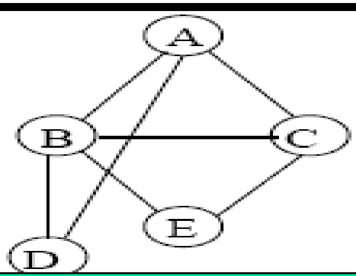
| $B$ | $C$ | $D$ | $f_1$ |
|---|---|---|---|
| true | true | true | .95 |
| true | true | false | .05 |
| true | false | true | .9 |
| true | false | false | .1 |
| false | true | true | .8 |
| false | true | false | .2 |
| false | false | true | 0 |
| false | false | false | 1 |

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

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

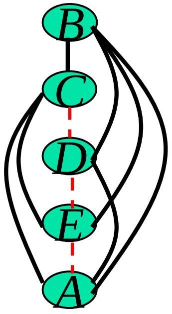
*Thanks to Darwiche*

# Bucket Elimination and Induced Width



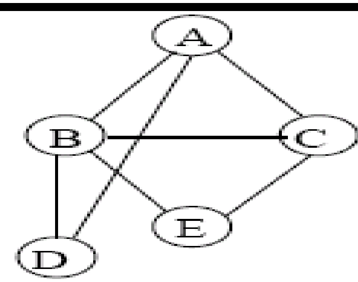

**Ordering: a, e, d, c, b**

$bucket(B) = \quad P(e|b,c), P(d|a,b), P(b|a)$
$bucket(C) = \quad P(c|a) \quad || \quad \lambda_B(a,c,d,e)$
$bucket(D) = \quad || \ \lambda_C(a,d,e)$
$bucket(E) = \quad e = 0 \quad || \ \lambda_D(a,c)$
$bucket(A) = \quad P(a) \quad || \quad \lambda_E(a)$



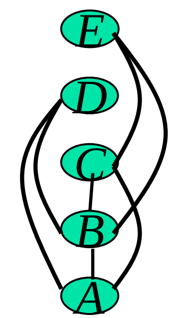

*W*=4*

# Bucket Elimination and Induced Width




**Ordering: a, b, c, d, e**

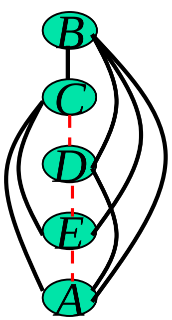
$bucket(E) = \quad P(e|b,c),\ \ e = 0$
$bucket(D) = \quad P(d|a,b)$
$bucket(C) = \quad P(c|a) \quad || \ P(e=0|b,c)$
$bucket(B) = \quad P(b|a) \quad || \quad \lambda_D(a,b), \lambda_C(b,c)$
$bucket(A) = \quad P(a) \quad || \quad \lambda_B(a)$




W*=2

**Ordering: a, e, d, c, b**

$bucket(B) = \quad P(e|b,c), P(d|a,b), P(b|a)$
$bucket(C) = \quad P(c|a) \quad || \quad \lambda_B(a,c,d,e)$
$bucket(D) = \quad || \ \lambda_C(a,d,e)$
$bucket(E) = \quad e = 0 \quad || \ \lambda_D(a,c)$
$bucket(A) = \quad P(a) \quad || \quad \lambda_E(a)$




W*=4

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.

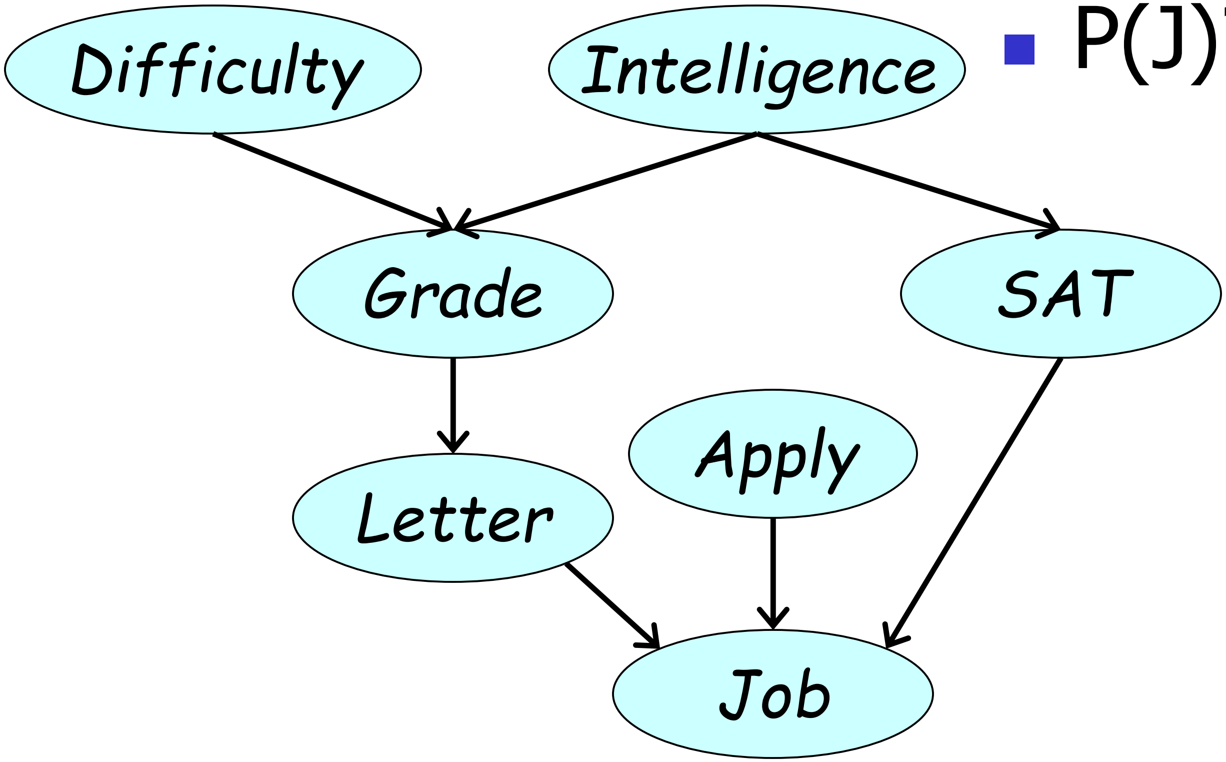
   **else,**
4. $\lambda_p \leftarrow \sum_{X_p} \psi_p \cdot \prod_{i=1}^{j} \lambda_{S_i}$
5. place $\lambda_p$ in bucket of the latest variable in scope($\lambda_p$),
6. **return** (as a result of processing $bucket_1$):

   $P(\mathrm{e}) = \alpha = \sum_{X_1} \psi_1 \cdot \prod_{\lambda \in bucket_1} \lambda$

   $P(X_1|\mathrm{e}) = \frac{1}{\alpha} \psi_1 \cdot \prod_{\lambda \in bucket_1} \lambda$

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

# Student Network Example

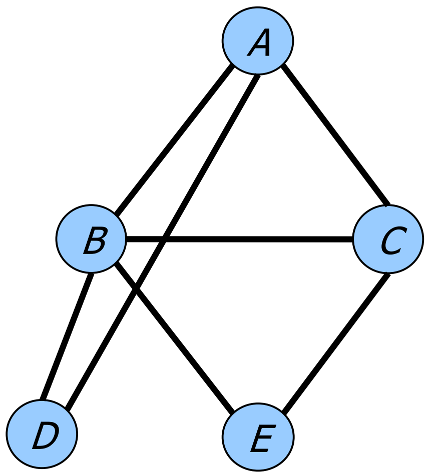



- P(J)?

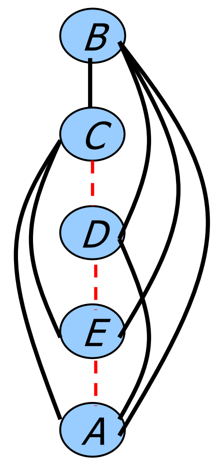
# Induced Width (continued)

$w^*(d)$ – the induced width of the primal graph along ordering $d$

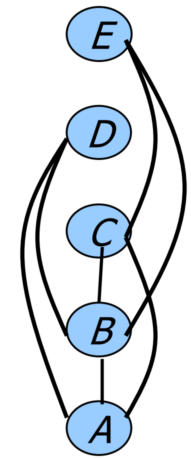
*The effect of the ordering:*




*Primal (moraal) graph*




$$w^*(d_1) = 4$$

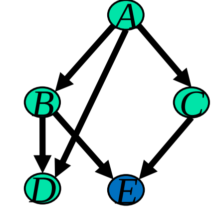



$$w^*(d_2) = 2$$

## *The impact of evidence*

# The impact of evidence?

## Algorithm *BE-bel*




$$P(A \mid E=0) = \alpha \sum_{E=0,D,C,B} P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid A,B) \cdot P(E \mid B,C)$$

$\sum_b \prod$ ← *Elimination operator*

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

*B=1*

bucket C: $P(c|a)$ $\lambda^B(a,d,c,e)$

bucket D: $\lambda^C(a,d,e)$

bucket E: $e=0$ $\lambda^D(a,e)$

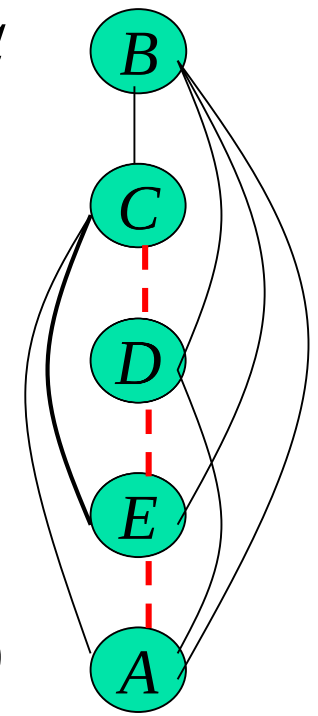
bucket A: $P(a)$ $\lambda^E(a)$

$P(e=0)$

$P(a|e=0)$

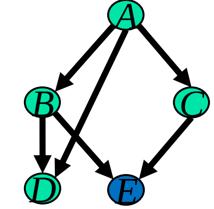
*W*=4*

*"induced width" (max clique size)*



# The impact of evidence?

## Algorithm *BE-bel*




$$P(A \mid E=0) = \alpha \sum_{E=0,D,C,B} P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid A,B) \cdot P(E \mid B,C)$$

$\sum_b \prod$ ← *Elimination operator*

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

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

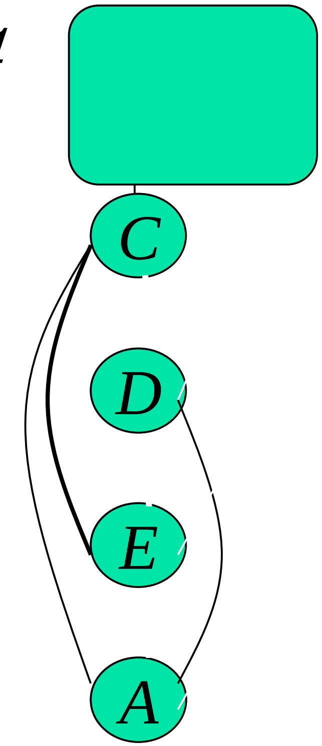
bucket D: *P(d/b=1,a)*

bucket E: $e=0$

bucket A: $P(a)$ *P(b=1/a)*

**$P(e=0)$**

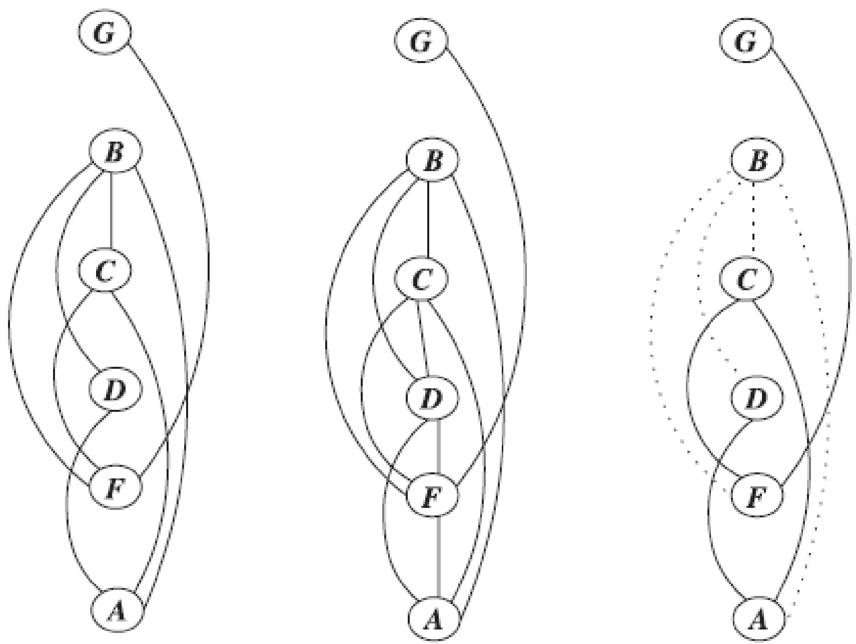
**$P(a|e=0)$**




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

# The impact of observations


(a) Directed acyclic graph (b) Moral graph



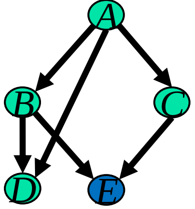

(a) (b) (c)

Figure 4.9: Adjusted induced graph relative to observing $B$.

*Ordered graph* *Induced graph* *Ordered conditioned graph*

# *Bucket-elimination for MPE (MAP)*

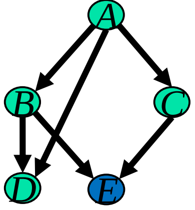
# $\text{MPE} = \max_{\bar{x}} P(\bar{x})$




$\sum$ is replaced by ***max*** *:*

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

# $\text{MPE} = \max_{\bar{x}} P(\bar{x})$




$\sum$ is replaced by ***max*** :

$$MPE = \max_{a,e,d,c,b} P(a)P(c \mid a)P(b \mid a)P(d \mid a,b)P(e \mid b,c)$$

$\max_b \prod$ ← *Elimination operator*

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

*bucket C:* $P(c|a)$ $h^B(a,d,c,e)$

*bucket D:* $h^C(a,d,e)$

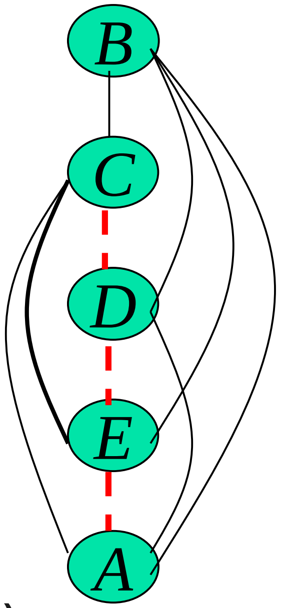
*bucket E:* $e=0$ $h^D(a,e)$

*bucket A:* $P(a)$ $h^E(a)$

***MPE***

*W*=4*

*"induced width" (max clique size)*



# Generating the MPE-tuple

5. $b' = \arg\max_b P(b \mid a') \times P(d' \mid b, a') \times P(e' \mid b, c')$

4. $c' = \arg\max_c P(c \mid a') \times h^B(a', d', c, e')$

3. $d' = \arg\max_d h^C(a', d, e')$

2. $e' = 0$

1. $a' = \arg\max_a P(a) \cdot h^E(a)$

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

C: $P(c|a)$ $h^B(a,d,c,e)$

D: $h^C(a,d,e)$

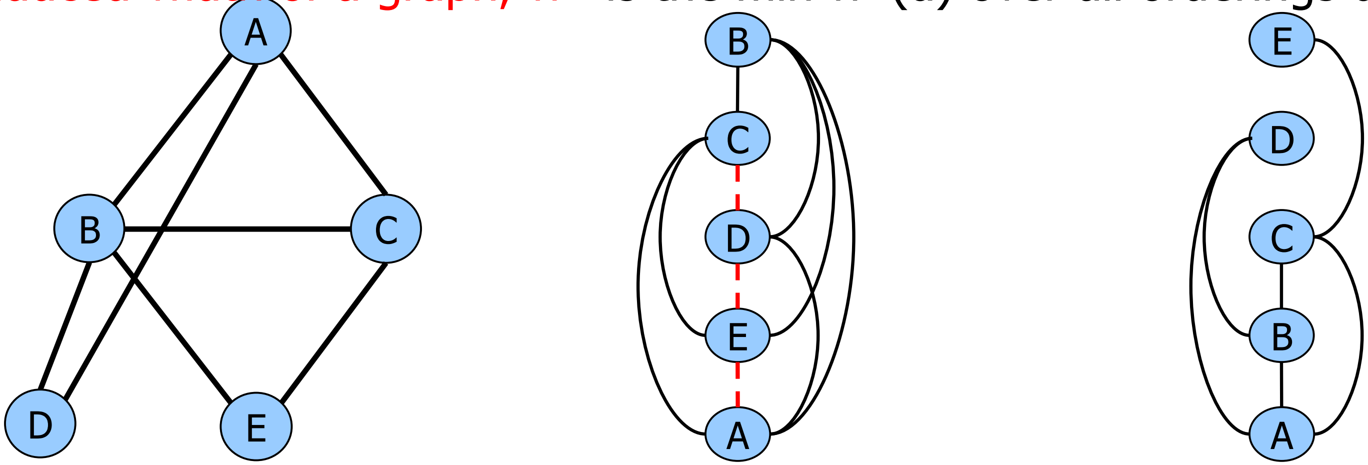
E: $e=0$ $h^D(a,e)$

A: $P(a)$ $h^E(a)$

**Return** $(a', b', c', d', e')$

# 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

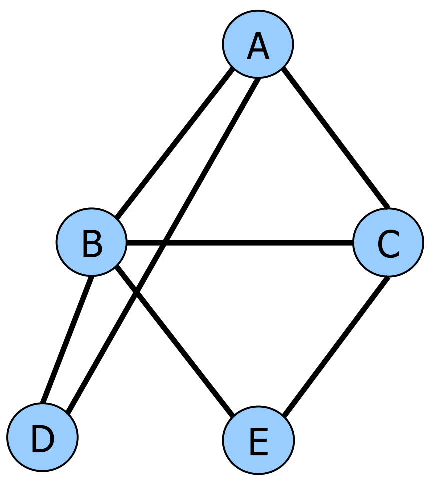
*Bucket-Elimination is **time** and **space***

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

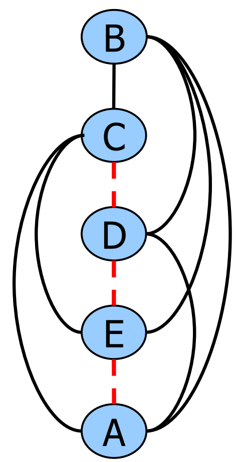
$w_d^*$ *: the induced width of the primal graph along ordering d*

*r = number of functions*

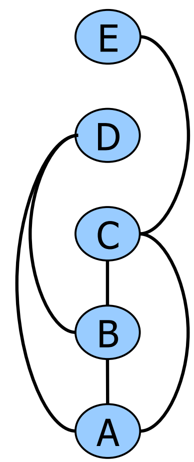
*The effect of the ordering:*




primal graph




$w^*(d_1) = 4$

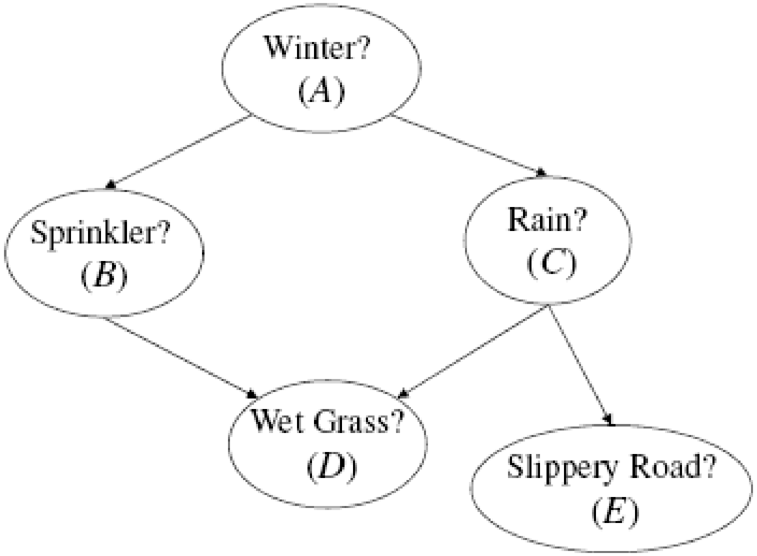



$w^*(d_2) = 2$

***Finding smallest induced-width is hard!***

# A Bayesian Network

*Example with mpe?*




| A | $\Theta_A$ |
|---|---|
| true | .6 |
| false | .4 |

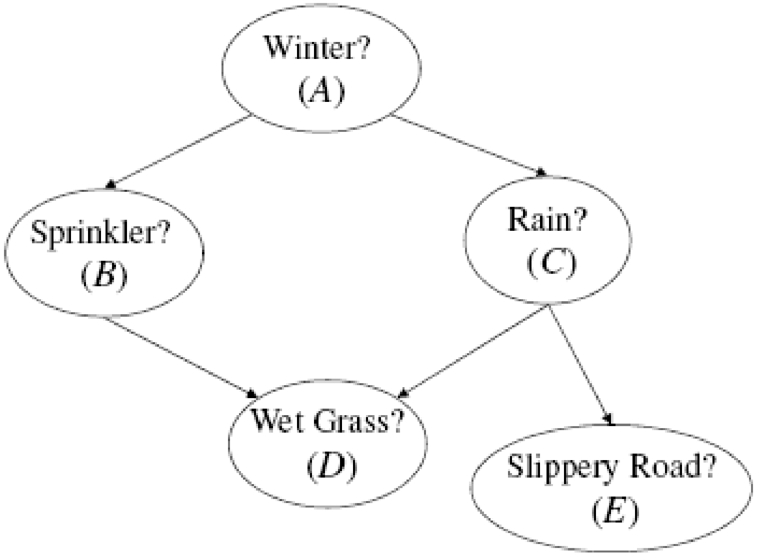
| A | B | $\Theta_{B\|A}$ |
|---|---|---|
| true | true | .2 |
| true | false | .8 |
| false | true | .75 |
| false | false | .25 |

| 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




| $A$ | $\Theta_A$ |
|---|---|
| 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 |

# Cost Networks

$$P(a,b,c,d,f,g) = P(a)P(b|a)P(c|a)P(f|b,c)P(d|a,b)P(g|f)$$

becomes

$$C(a,b,c,d,e) = -logP = C(a) + C(b,a) + C(c,a) + C(f,b,c) + C(d,a,b) + C(g,f)$$

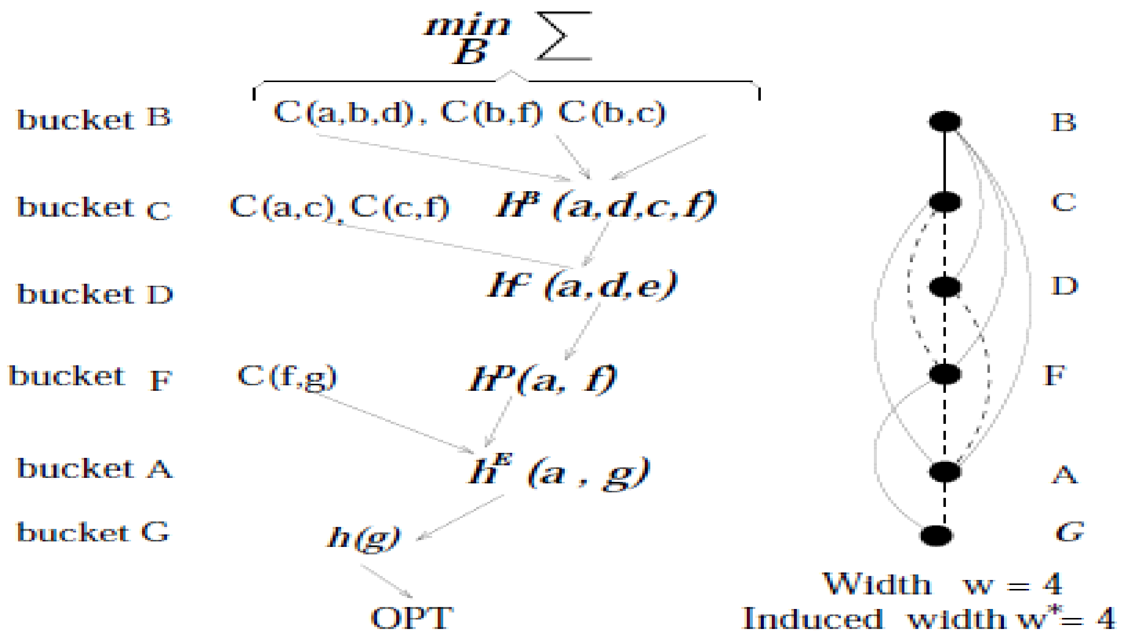
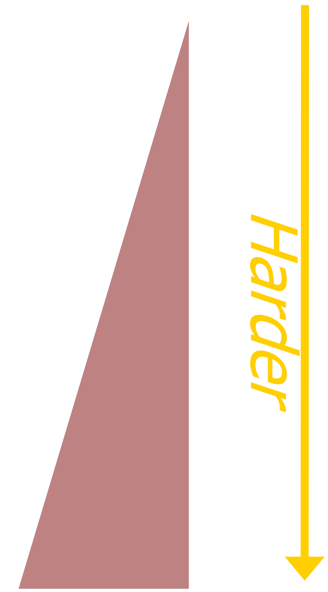



Figure 5.12: Schematic execution of BE-Opt

# Marginal Map

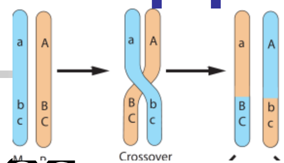
| | |
|---|---|
| Max-Inference | $f(\mathbf{x}^*) = \max_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$ |
| Sum-Inference | $Z = \sum_{\mathbf{x}} \prod_{\alpha} f_\alpha(\mathbf{x}_\alpha)$ |
| Mixed-Inference | $f(\mathbf{x}_M^*) = \max_{\mathbf{x}_\mathrm{M}} \sum_{\mathbf{x}_S} \prod_{\alpha}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



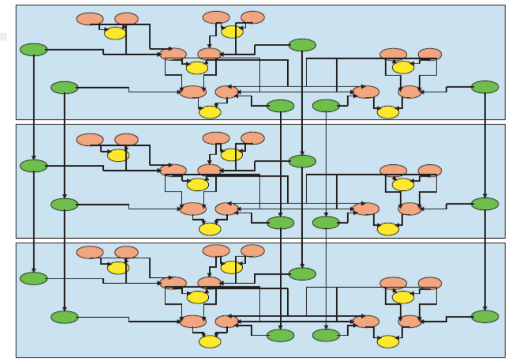

- **NP-hard**: exponentially many terms

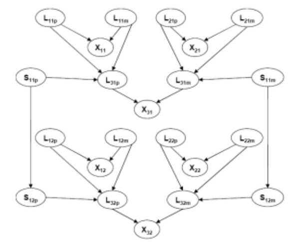
# Example for MMAP Applications

- Haplotype in Family pedigrees




6 people, 3 markers





- Coding networks

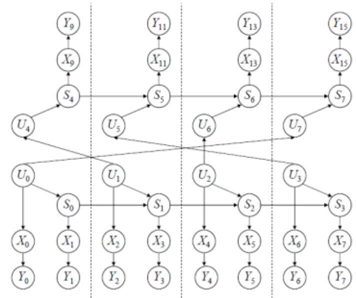
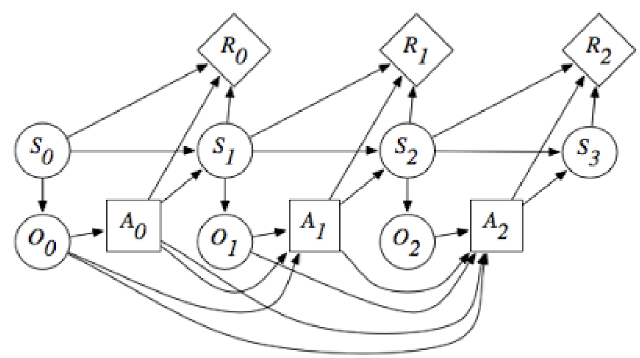
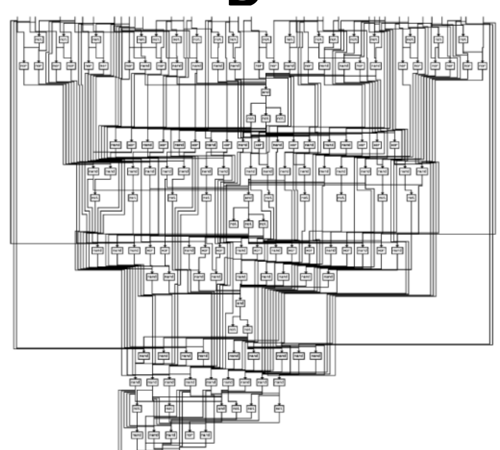


Figure 5.24: A Bayesian network for a turbo code.

- Probabilistic planning

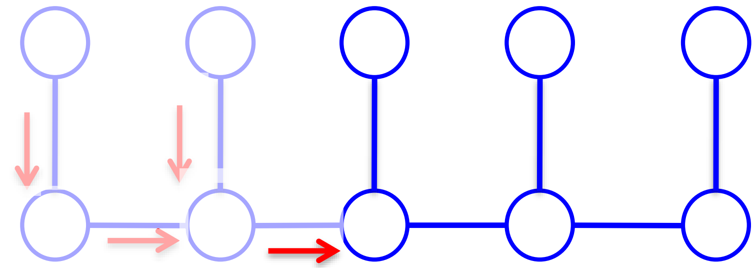


- Diagnosis



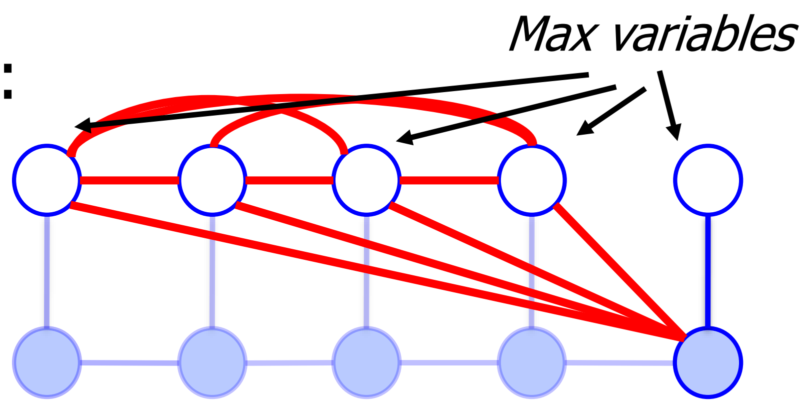
# Marginal MAP is Not Easy on Trees

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



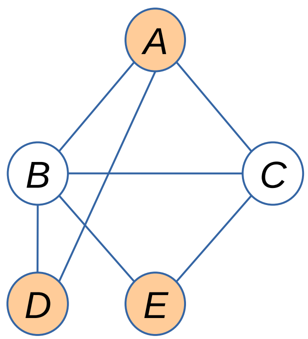
- Marginal MAP
  - Operations do not commute:
  - Sum must be done first!

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



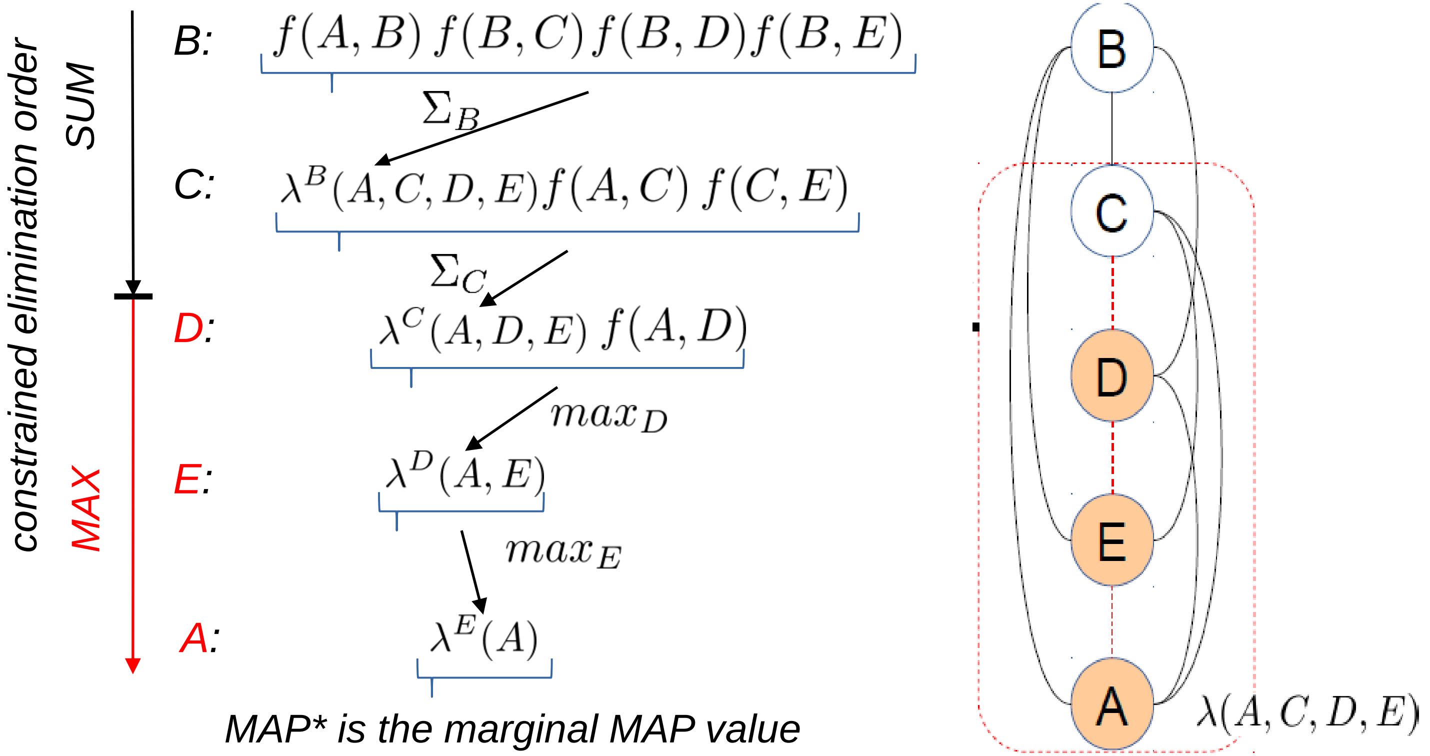
# Bucket Elimination for MMAP

*Bucket Elimination*



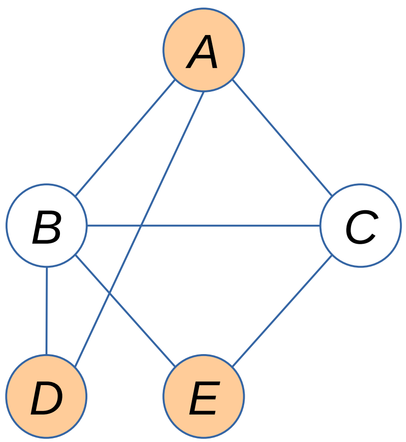

$$\mathbf{X}_M = \{A, D, E\}$$
$$\mathbf{X}_S = \{B, C\}$$

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



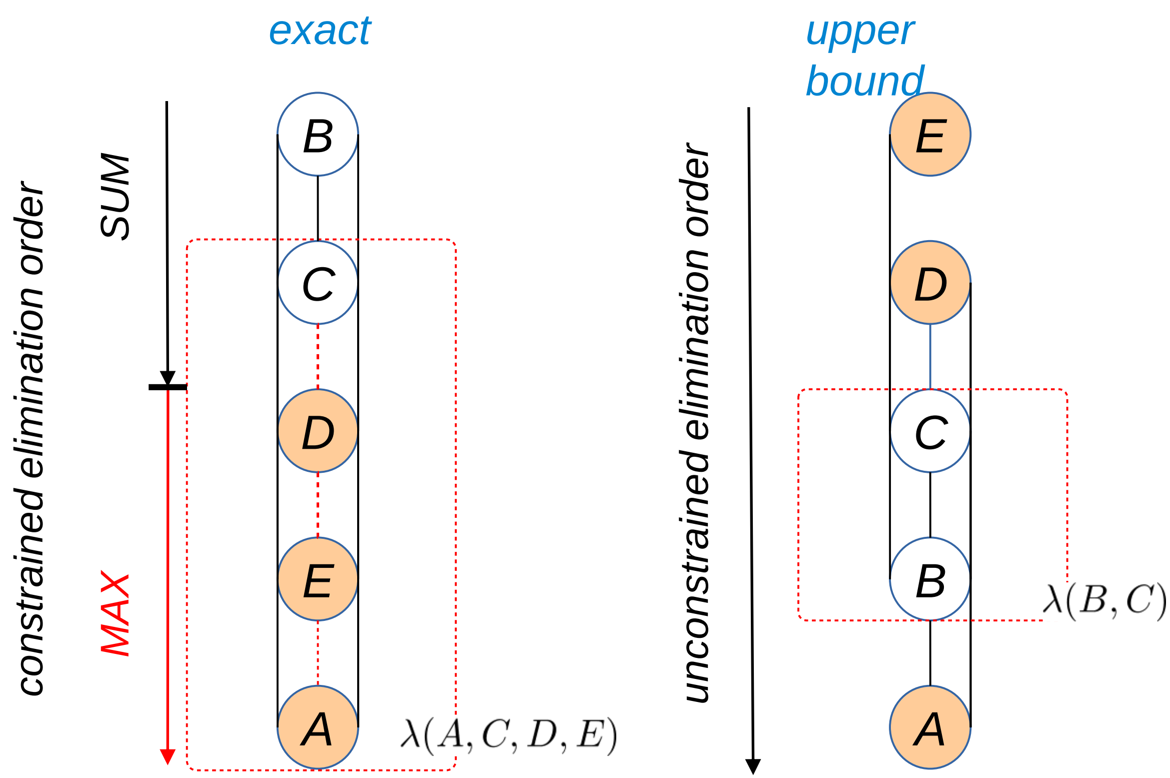

*MAP* is the marginal MAP value*

# 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)*

# 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 with Markov Networks

- Undirected graphs with potentials on cliques
- Query: find *partition function. Same as* probability of the evidence in a Bayesian network.
- The joint probability distribution of a Markov network is defined by:

$$P(x) = \frac{1}{Z} \sum_{x \in D} \Pi_{C \in \mathcal{C}} \Psi_C(x_C)$$

*BE is equally applicable*

$$Z = \Sigma_x \Pi_{C \in \mathcal{C}} \Psi_C(x_C) \tag{2.2}$$

For example. A markov network over the moral graph in Figure 2.4(b) is defined by:

$$P(a, b, c, d, f, g) = \frac{\Psi(a, b, c) \cdot \Psi(b, c, f) \cdot \Psi(a, b, d) \cdot \Psi(f, g)}{Z} \tag{2.3}$$

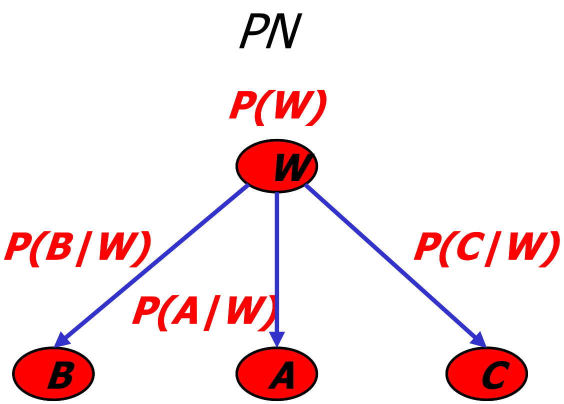
where,

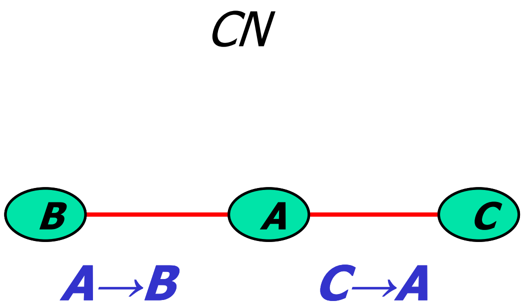
$$Z = \sum_{a,b,c,d,e,f,g} \Psi(a, b, c) \cdot \Psi(b, c, f) \cdot \Psi(a, b, d) \cdot \Psi(f, g) \tag{2.4}$$

# 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)
  - Mixed networks
- Tree-decomposition schemes
  - Bucket tree elimination
  - Cluster tree elimination

# Party Example




***Semantics?***

***Algorithms?***

***Query:***
*Is it likely that Chris goes to the party if Becky does not but the weather is bad?*

$$P(C, \neg B \mid w = bad, A \rightarrow B, C \rightarrow A)$$

# Bucket Elimination for Mixed Networks

*The CPE query*

$$P_{\mathcal{B}}(\varphi) = \sum_{\mathbf{x}_\varphi \in Mod(\varphi)} P(\mathbf{x}_\varphi)$$

Using the belief network product form we get:

$$P_{\mathcal{B}}(\varphi) = \sum_{\{\mathbf{x} | \mathbf{x}_\varphi \in Mod(\varphi)\}} \prod_{i=1}^{n} P(x_i | \mathbf{x}_{pa_i}).$$

*P((C → B) and P(A → C))*

# Bucket-Elimination example for a Mixed Network $\varphi = (B \vee C), (G \vee D), (\sim D \vee \sim B)$

In $Bucket G$ : $\lambda_G(f,d) = \sum_{\{g | g \vee d = true\}} P(g|f)$

In $Bucket_F$ : $\lambda_F(b,c,d) = \sum_f P(f|b,c)\lambda_G(f,d)$

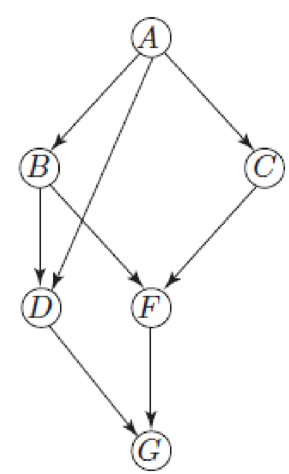
In $Bucket_D$ : $\lambda_D(a,b,c) = \sum_{\{d | \neg d \vee \neg b = true\}} P(d|a,b)\lambda_F(b,c,d)$

In $Bucket_B$ : $\lambda_B(a,c) = \sum_{\{b | b \vee c = true\}} P(b|a)\lambda_D(a,b,c)\lambda_F(b,c)$

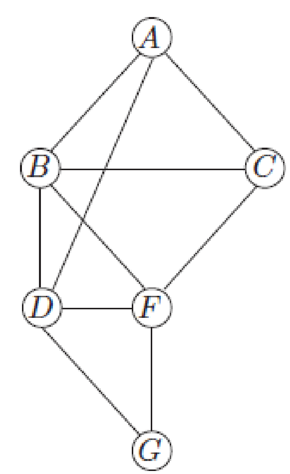
In $Bucket_C$ : $\lambda_C(a) = \sum_c P(c|a)\lambda_B(a,c)$

In $Bucket_A$ : $\lambda_A = \sum_a P(a)\lambda_C(a)$

$P(\varphi) = \lambda_A$.



(a) Directed Acyclic Graph



(b) Moral Graph

# Bucket-Elimination example for a Mixed Network

$\varphi = (B \vee C), (G \vee D), (\sim D \vee \sim B)$

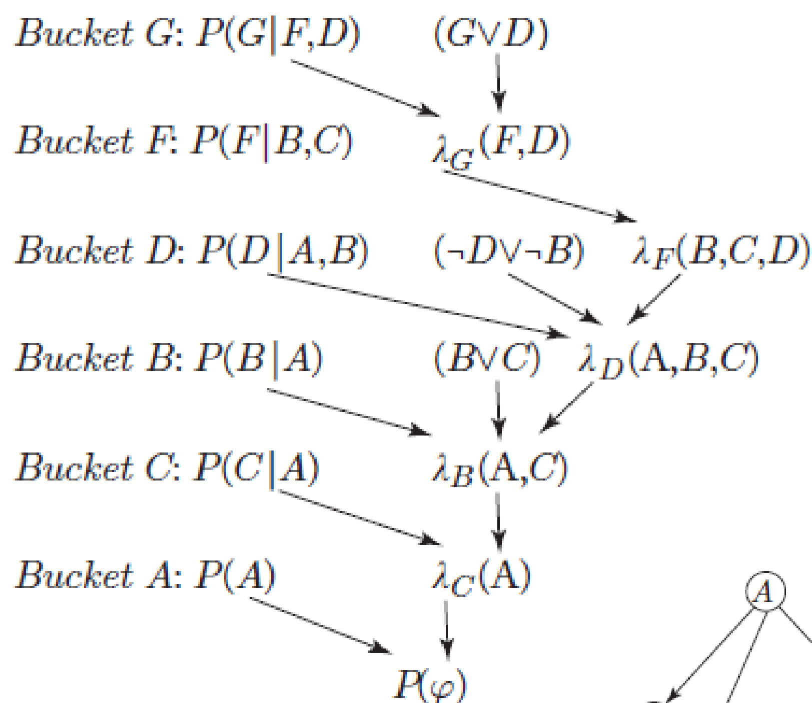
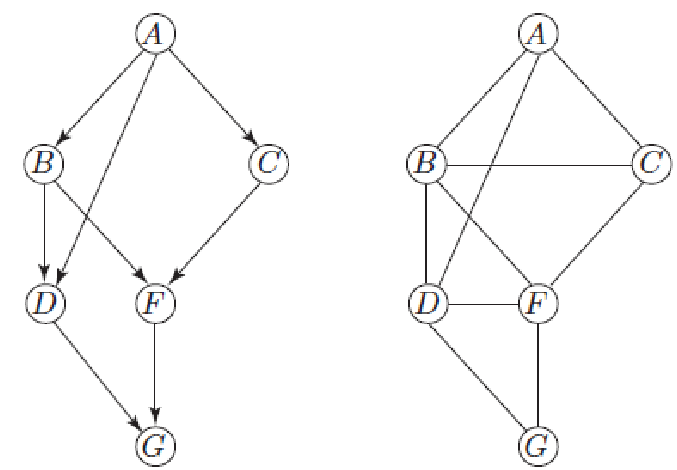
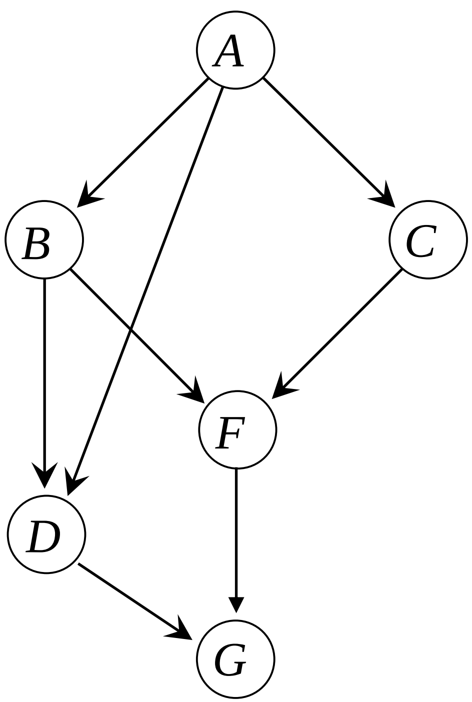


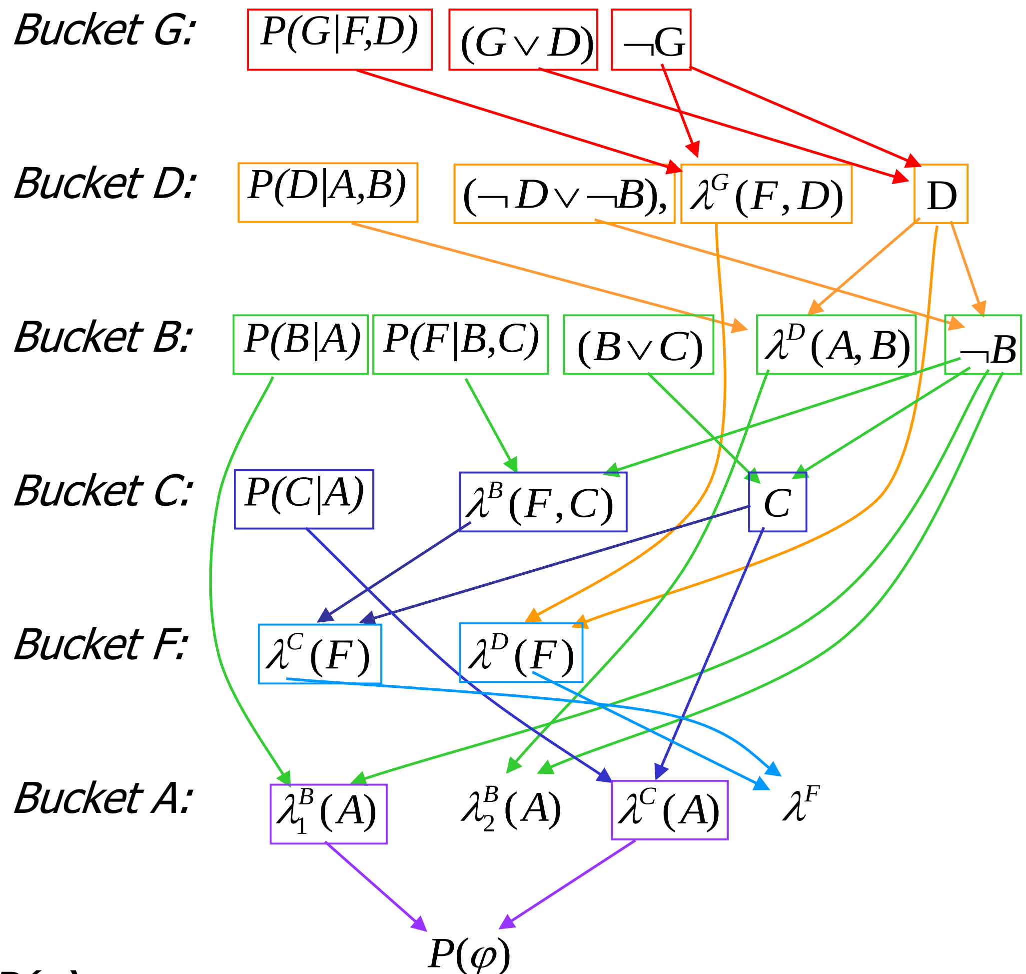

Figure 4.18: Execution of BE-CPE.




(a) Directed Acyclic Graph (b) Moral Graph

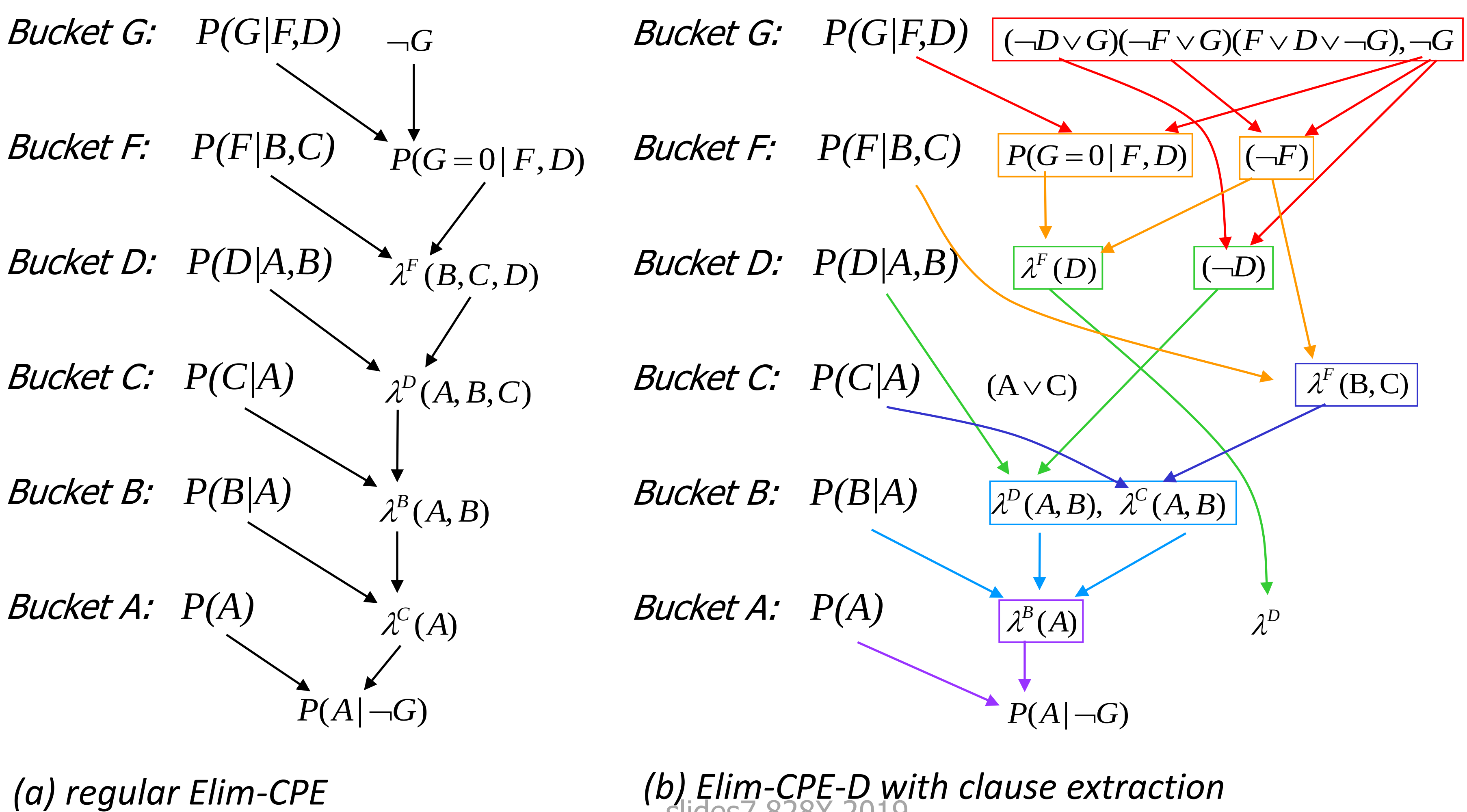
# Trace of Elim-CPE



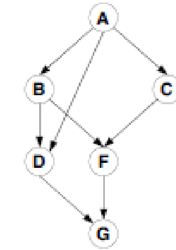

*Belief network P(g,f,d,c,b,a)*
*=P(g|f,d)P(f|c,b)P(d|b,a)P(b|a)P(c|a)P(a)*

# Bucket-Elimination for CPE



# Bucket-Elimination example for a Mixed Network

$\varphi = (B \vee C), (G \vee D), (\sim D \vee \sim B)$



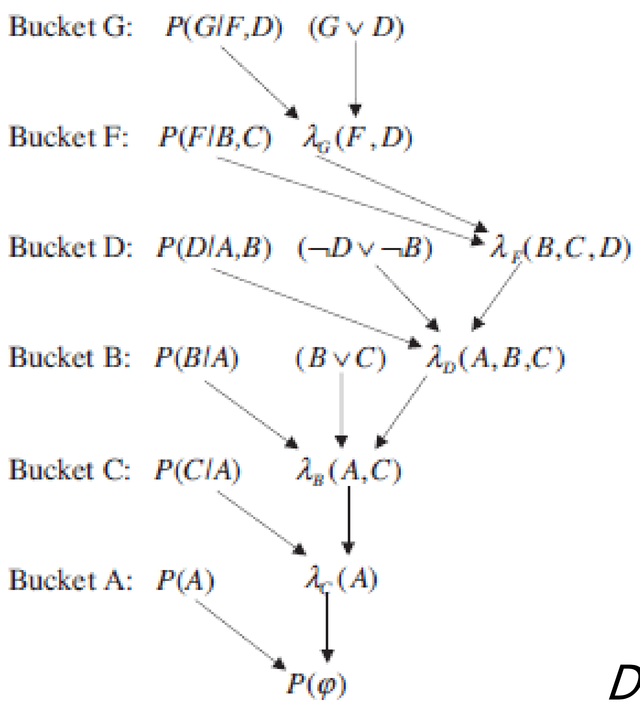




Figure 4.15: Execution of BE-CPE.

*Different orderings*

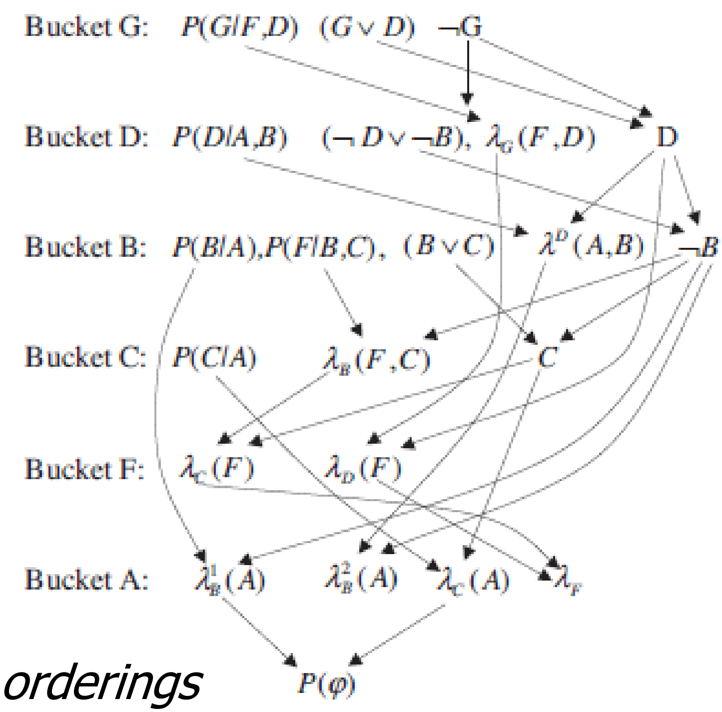
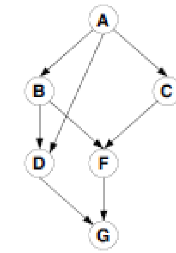



Figure 4.16: Execution of BE-CPE (evidence $\neg G$).

# Bucket-Elimination example for a Mixed Network

$\varphi = (B \vee C), (G \vee D), (\sim D \vee \sim B)$



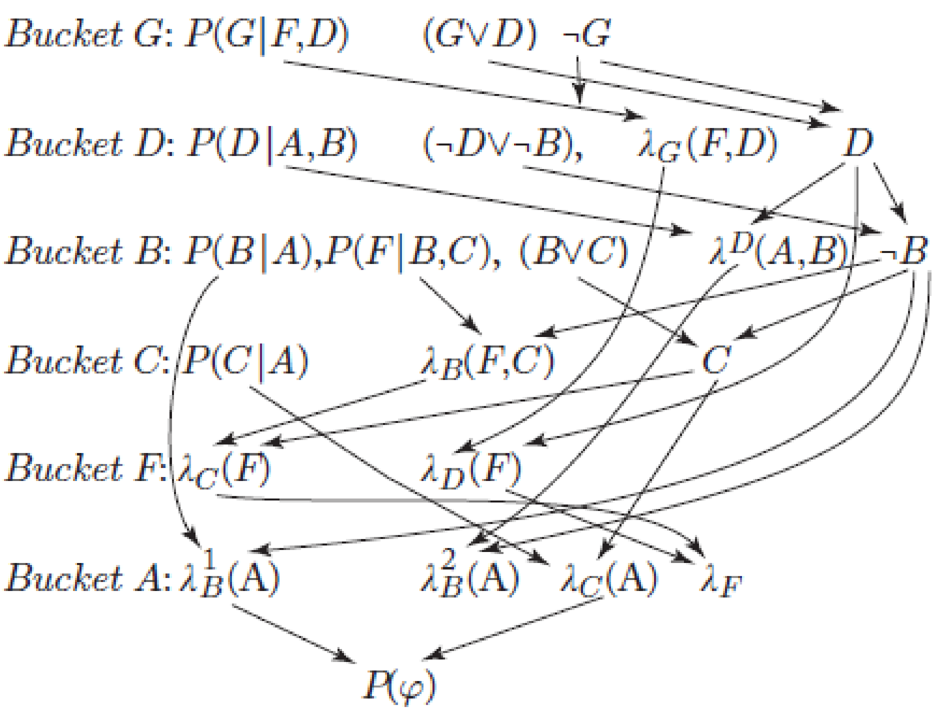




Figure 4.19: Execution of BE-CPE (evidence $\neg G$).