# Principles of Reasoning with Graphical Models

**Rina Dechter**

Bren school of ICS, University of California, Irvine

**IAAI 2012**

# Outline

- Graphical models and reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments and competitions

# Outline

- Graphical models and reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments

# Sample Applications for Graphical Models

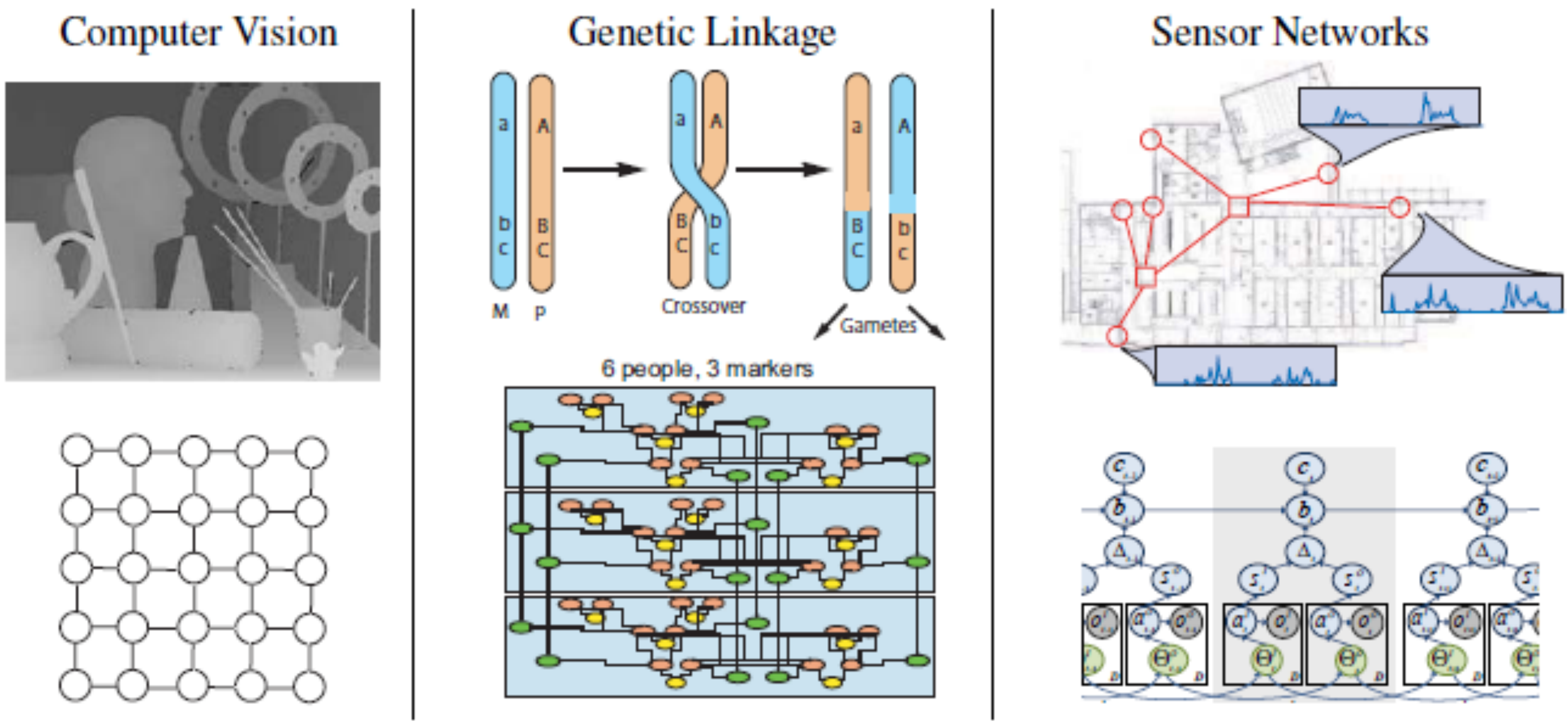



Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal models.

# Constraint Networks

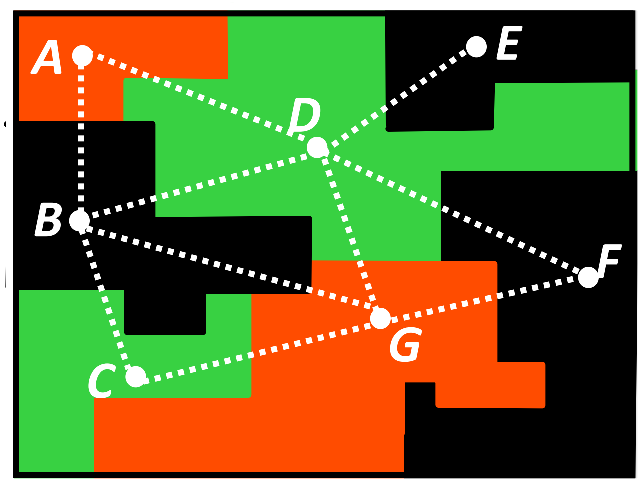
## *Map coloring*

*Variables:* *countries (A B C etc.)*

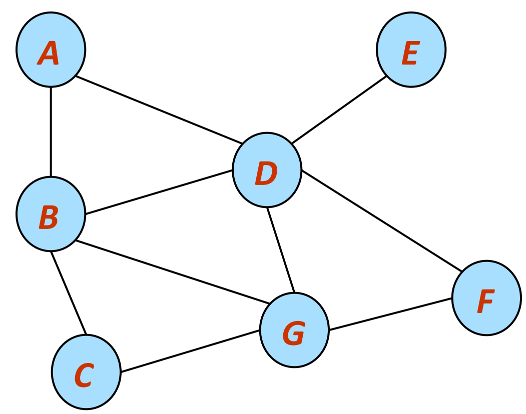
*Values:* *colors (red green blue)*

*Constraints:* $A \neq B, \; A \neq D, \; D \neq E, \ldots$

| *A* | *B* |
|---|---|
| *red* | *green* |
| *red* | *yellow* |
| *green* | *red* |
| *green* | *yellow* |
| *yellow* | *green* |
| *yellow* | *red* |



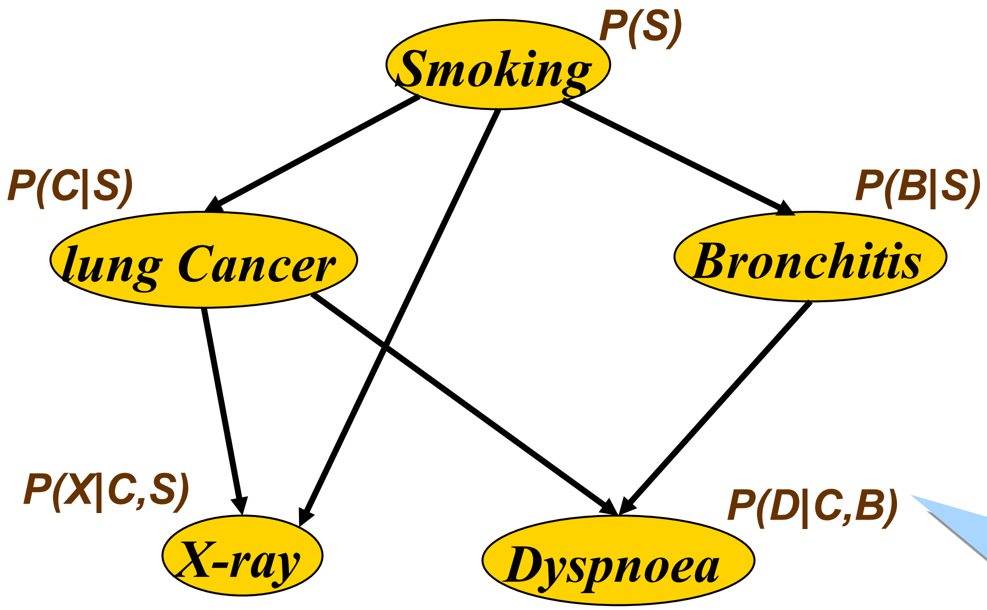

***Constraint graph***






**Queries: Find one solution, all solutions, counting**

# Bayesian Networks (Pearl 1988)




**BN = (G, Θ)**

**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 |

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

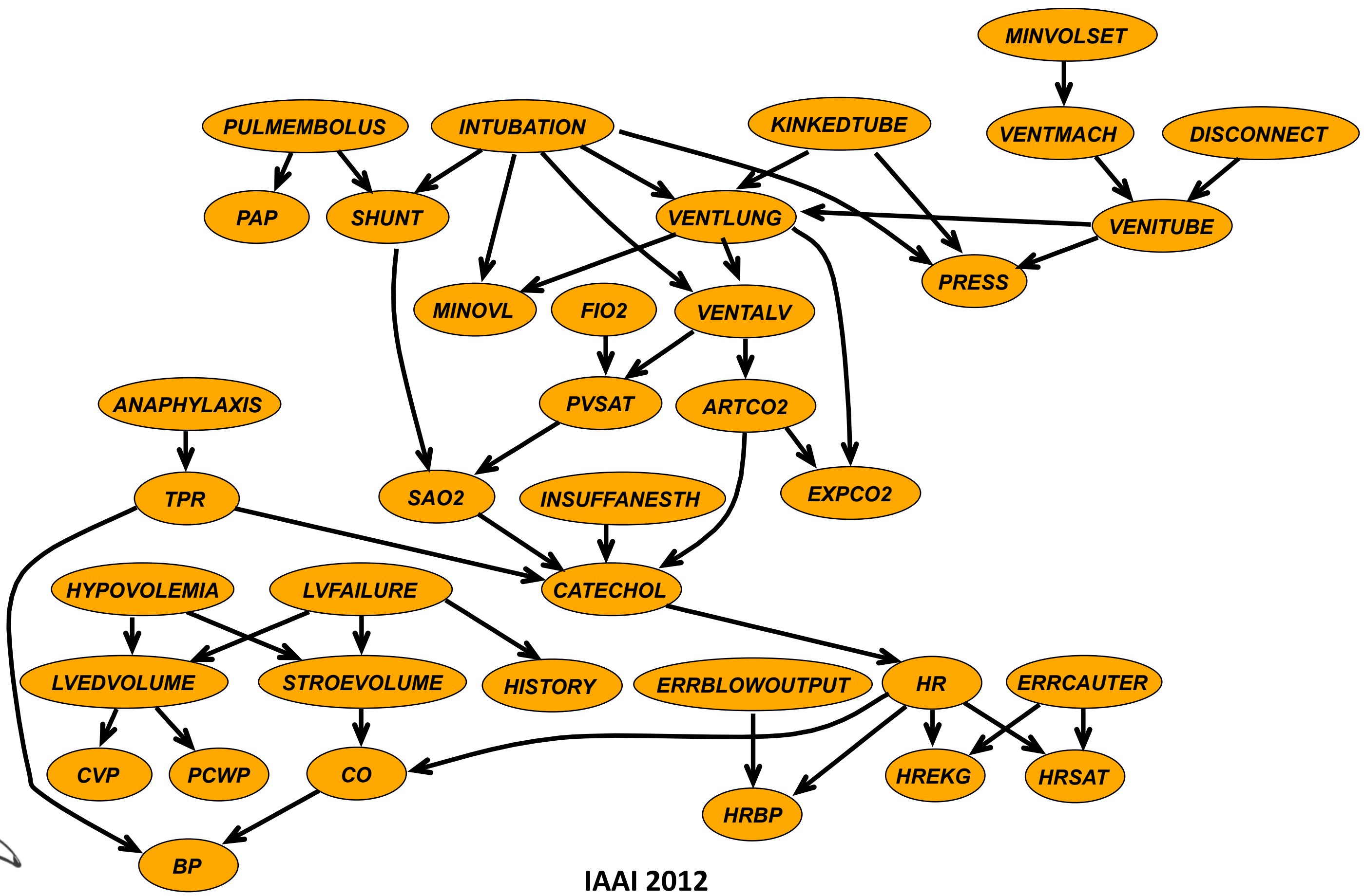
*Combination: Product*
*Marginalization: sum/max*

- *Posterior marginals, probability of evidence, MPE*
- $P(D=0) = \sum_{S,L,B,X} P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B$

$MAP(P) = \max_{S,L,B,X} P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B)$

# Monitoring Intensive-Care Patients

The “alarm” network - 37 variables, 509 parameters (instead of $2^{37}$)

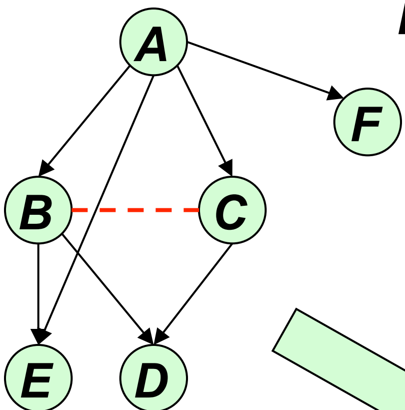



# Mixed Networks

(Mateescu and Dechter, 2004)

***Examples: NLP, Linkgage, Software verification, probabilistic languages***

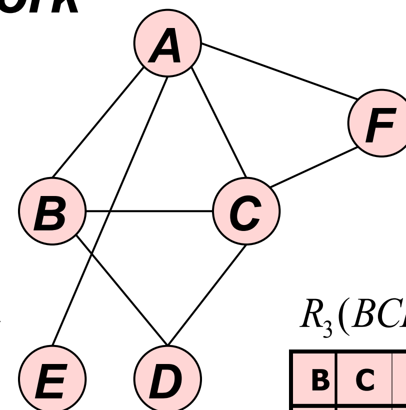
***Belief Network***




$P(D \mid B, C)$

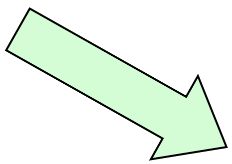
| B | C | D=0 | D=1 |
|---|---|---|---|
| 0 | 0 | .2 | .8 |
| 0 | 1 | .1 | .9 |
| 1 | 0 | .3 | .7 |
| 1 | 1 | .5 | .5 |

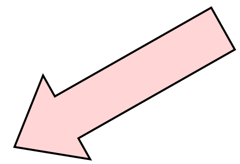
***Constraint Network***



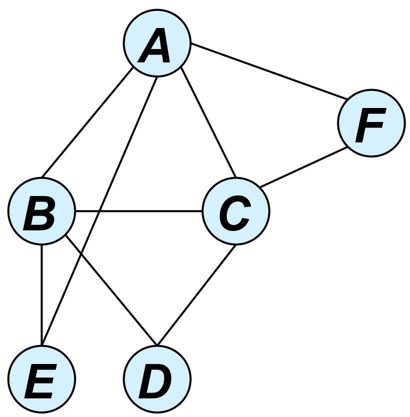

$R_3(BCD)$

| B | C | D |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 1 | 0 |





***Moral mixed graph***



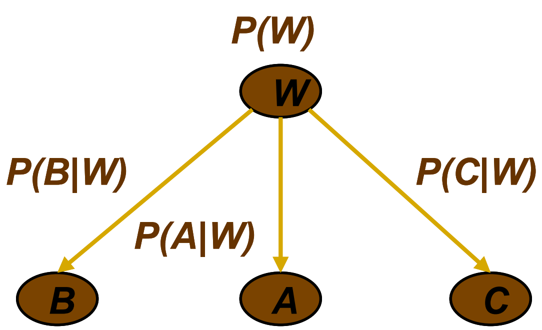

***Complex cnf queries: P((A or B) and (~CVD))***

$$P_M(\bar{x}) = \begin{cases} P_B(\bar{x} \mid \bar{x} \in \rho) = \dfrac{P_B(\bar{x})}{P_B(\bar{x} \in \rho)}, \text{if } \bar{x} \in \rho \\ 0, \text{otherwise} \end{cases}$$

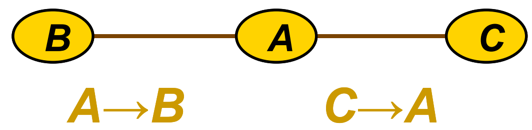
# Mixed Probabilistic and Deterministic networks

***Alex is-likely-to-go in bad weather***
***Chris rarely-goes in bad weather***
***Becky is indifferent but unpredictable***

**PN**




**CN**




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

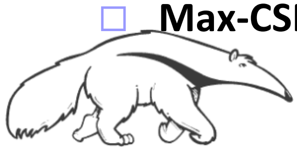
$$P(C, \neg B \mid w = bad, A \rightarrow B, C \rightarrow A)$$

# Graphical Models

- A graphical model (**X**,**D**,**F**):
  - **X** = $\{X_1, \dots X_n\}$ variables
  - **D** = $\{D_1, \dots D_n\}$ domains
  - **F** = $\{f_1, \dots, f_r\}$ functions
    (constraints, CPTS, CNFs …)

- Operators:
  - combination
  - elimination (projection)

- Tasks:
  - **Belief updating**: $\Sigma_{X-y} \prod_j P_i$
  - **MPE**: $\max_X \prod_j P_j$
  - **CSP**: $\prod_X \times_j C_j$
  - **Max-CSP**: $\min_X \Sigma_j F_j$



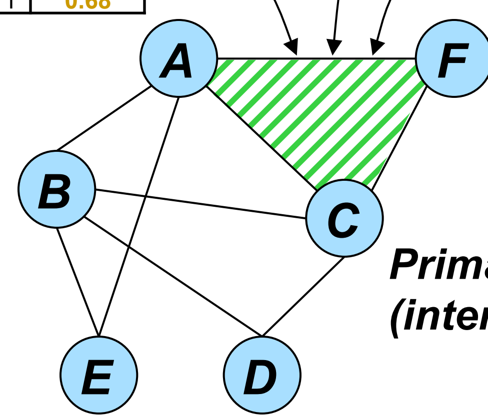
***Conditional Probability Table (CPT)***

| 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 |

$f_i := (F = A + C)$




***Primal graph (interaction graph)***

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

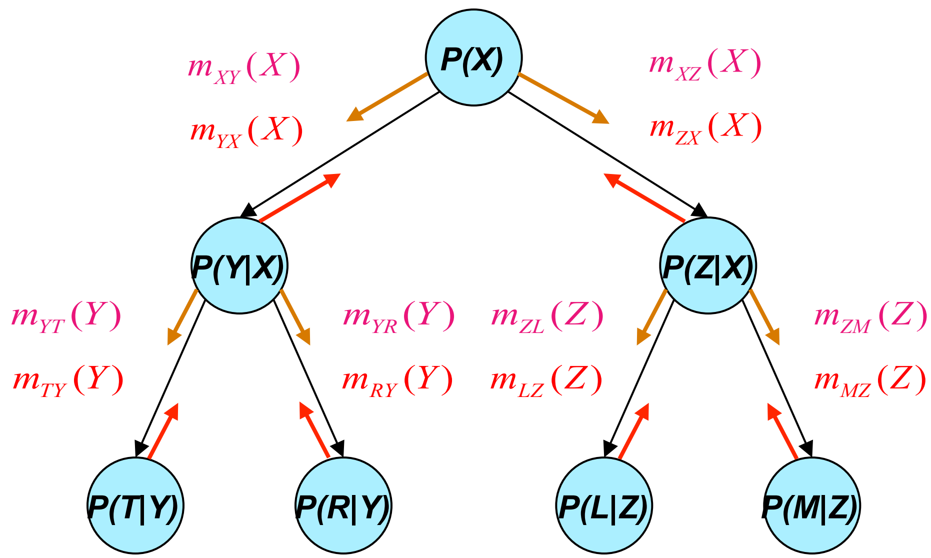
# Outline

- Graphical models: reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments

# Tree-solving is Easy

***Belief updating (sum-prod)***

***CSP – consistency (projection-join)***



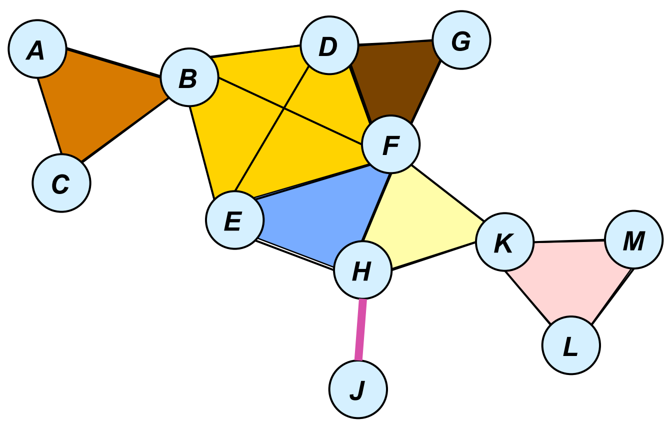

***Dynamic Programming, Inference***

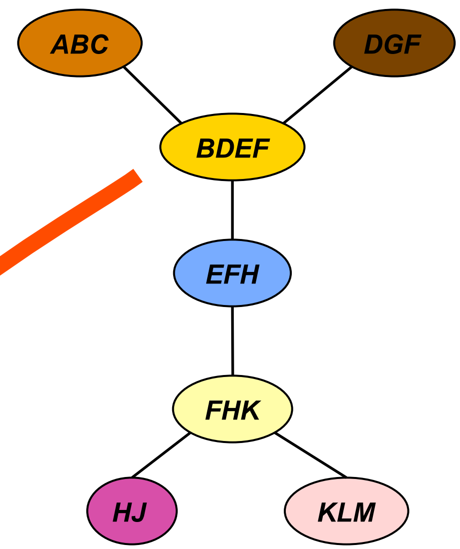
***MPE (max-prod)***

***#CSP (sum-prod)***

***Trees are processed in linear time and memory***
***Message-passing***

# Clustering and Treewidth

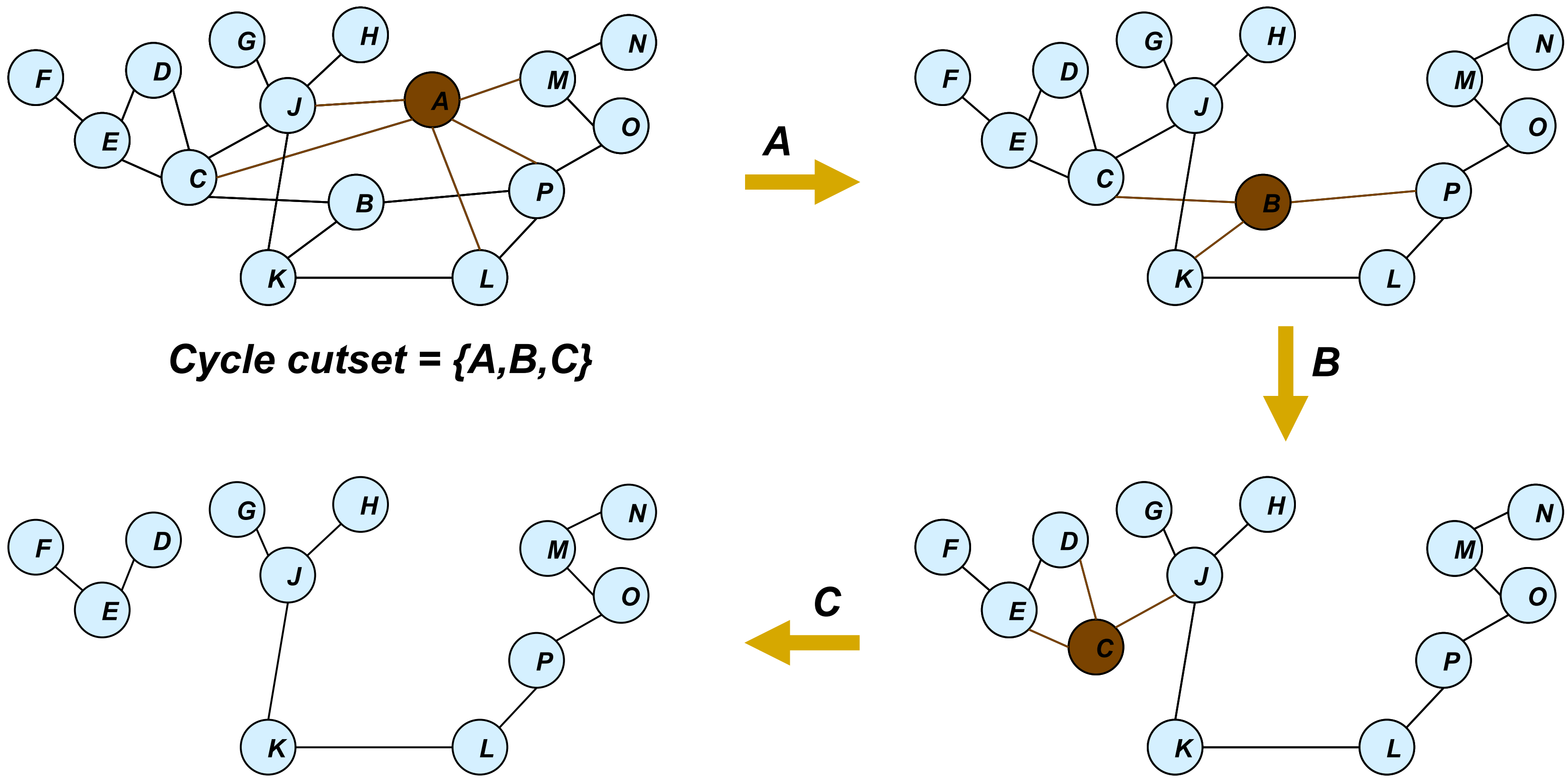



***Inference algorithm:***
***Time:*** *exp(tree-width+1)*
***Space:*** *exp(separator-width)*

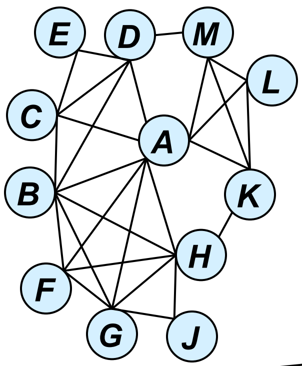
***treewidth = 4 - 1 = 3***
***treewidth = (maximum cluster size) – 1***
***Separator-width=2***

# Conditioning and Cycle cutset

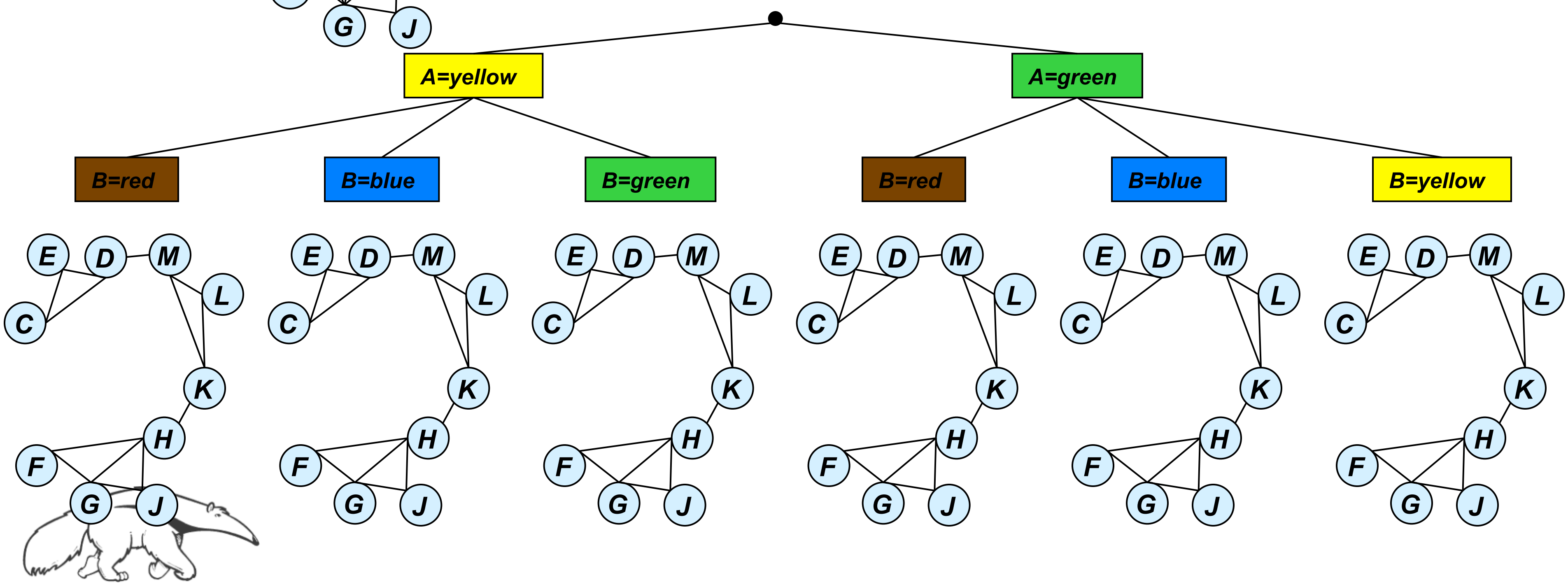


# Search over the Cutset (cont)

*Graph Coloring problem*

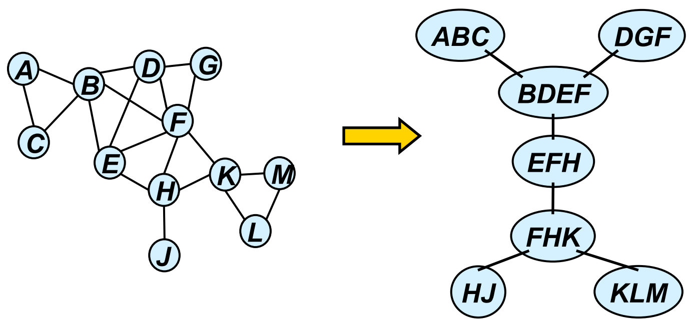


- *Inference may require too much memory*
- *Condition on some of the variables*



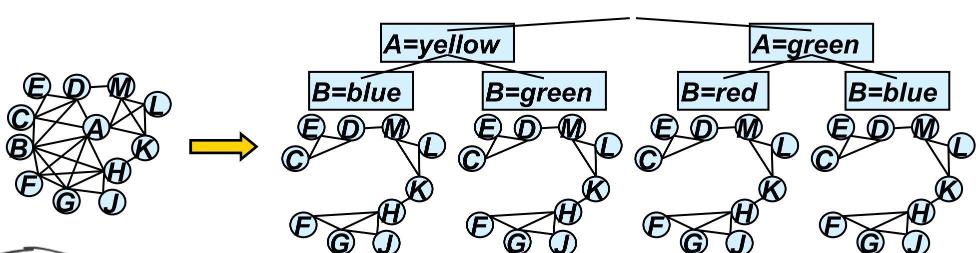
# Inference vs. Conditioning

- By Inference (thinking)




*Exponential in treewidth*
*Time and memory*

- ***By Conditioning (guessing)***




*Exponential in cycle-cutset*
*Time-wise, linear memory*

# Solution Techniques, State of the art

**AND/OR search**

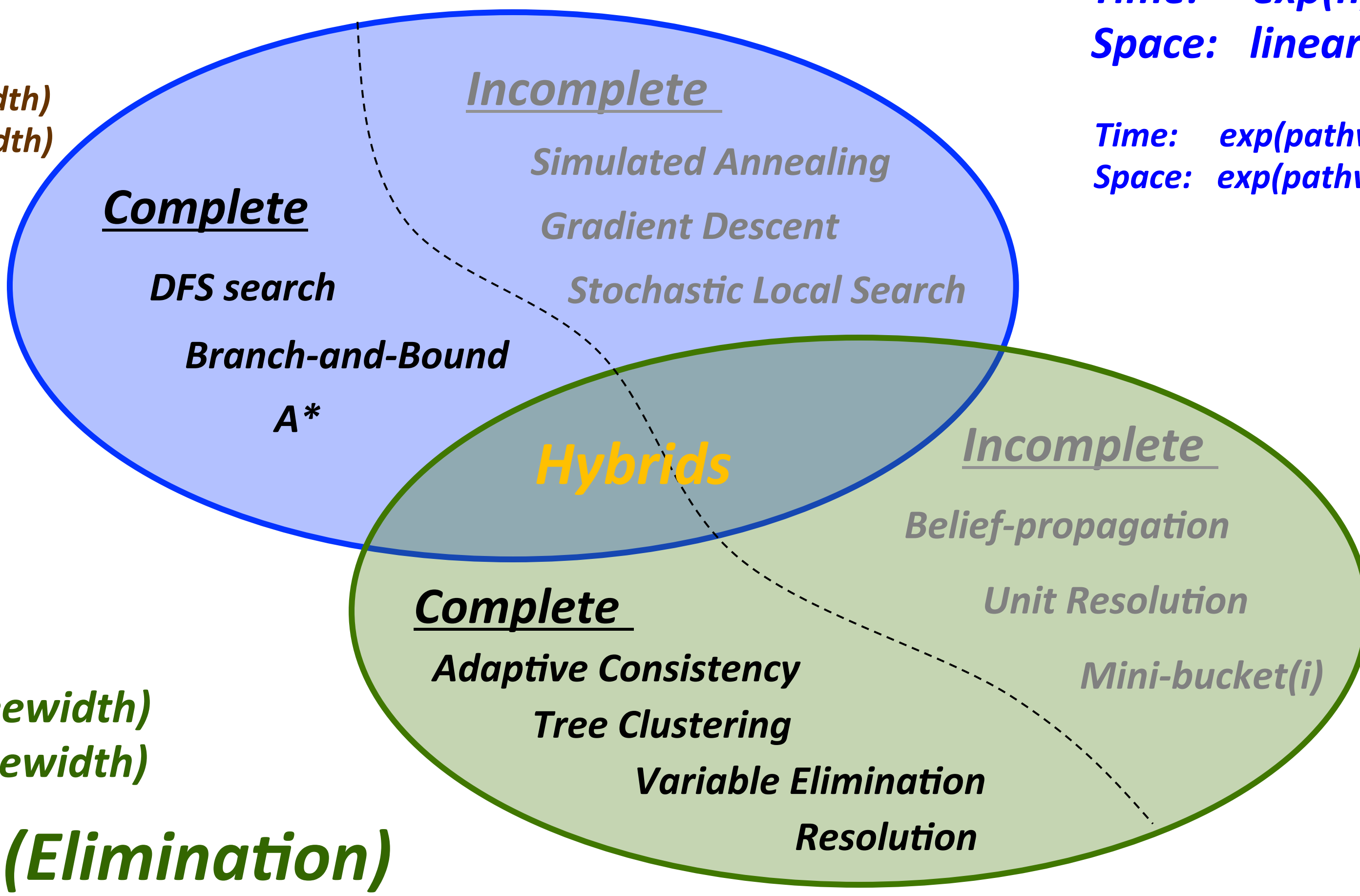
*Time: exp(treewidth * log n)*
*Space: linear*

*Space: exp(treewidth)*
*Time: exp(treewidth)*

**Search (Conditioning)**

*Time: exp(n)*
*Space: linear*

*Time: exp(pathwidth)*
*Space: exp(pathwidth)*



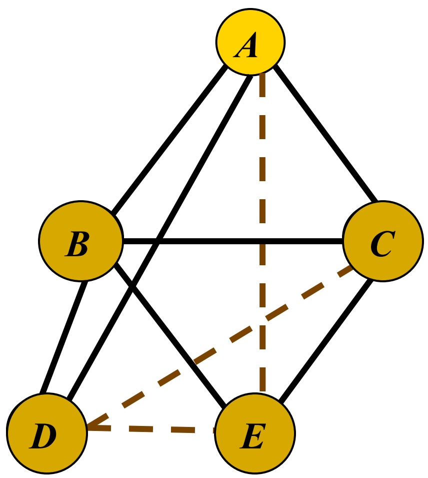

*Time: exp(treewidth)*
*Space: exp(treewidth)*

**Inference (Elimination)**

# Outline

- Graphical models: reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments

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



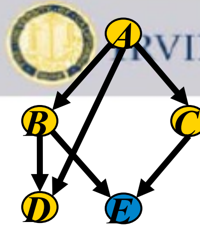

$$P(a|e=0) \propto P(a,e=0)=$$

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

$$P(a)\sum_{e=0}\sum_{d}\sum_{c} P(c|a)\sum_{b}\underbrace{P(b|a)P(d|b,a)P(e|b,c)}_{h^{B}(a,d,c,e)}$$

*Variable Elimination*

# 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:*** $\overbrace{P(b|a) \quad P(d|b,a) \quad P(e|b,c)}$

***bucket C:*** $P(c|a)$ $\quad \lambda^B(\underbrace{a,d,c,e})$

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

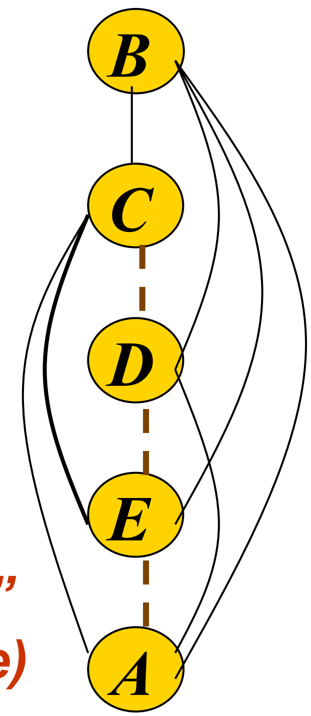
***bucket E:*** $e=0$ $\quad \lambda^D(a,e)$

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

$P(e=0)$

$P(a|e=0)$

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




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

# Inference for Optimization: Bucket Elimination

*Algorithm BE-mpe* *(Dechter 1996, Bertele and Briochi, 1977)*

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

$$\underbrace{\max_X \prod}$$

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

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

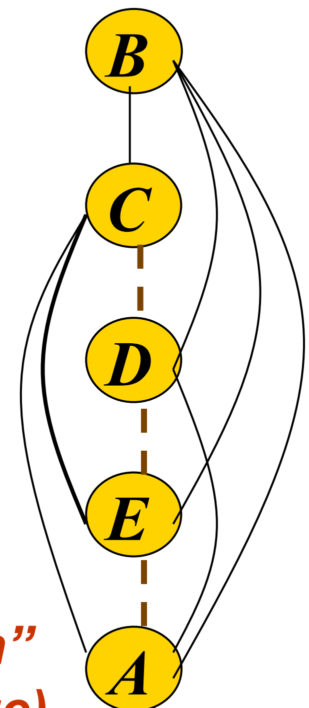
***bucket D:*** $h^C(a,d,e)$

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

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

***OPT***

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




# Generating the MPE-tuple

| | | |
|---|---|---|
| $B$: | $P(b\|a)$ $P(d\|b,a)$ $P(e\|b,c)$ | |
| $C$: | $P(c\|a)$ | $h^B(a,d,c,e)$ |
| $D$: | | $h^C(a,d,e)$ |
| $E$: | $e=0$ | $h^D(a,e)$ |
| $A$: | $P(a)$ | $h^E(a)$ |

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

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

# Combination of Probability 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 |

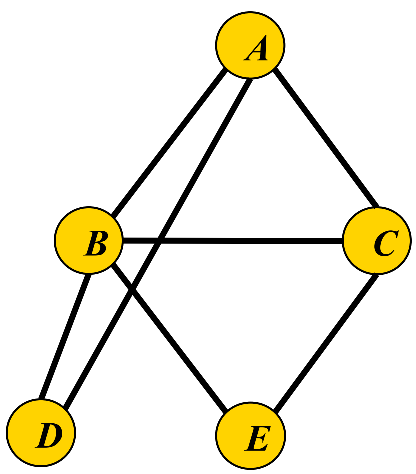
***= 0.1 x 0.8***

# Complexity of Elimination

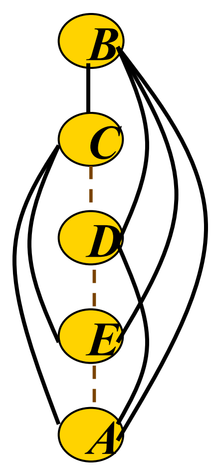
$$O(n \exp(w^*(d)))$$

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

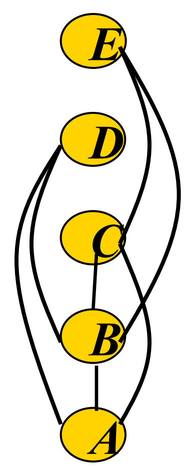
**The effect of the ordering:**




**"Moral" graph**



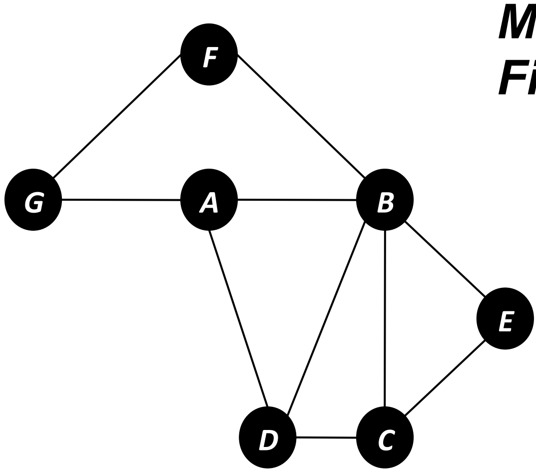

$w^*(d_1) = 4$

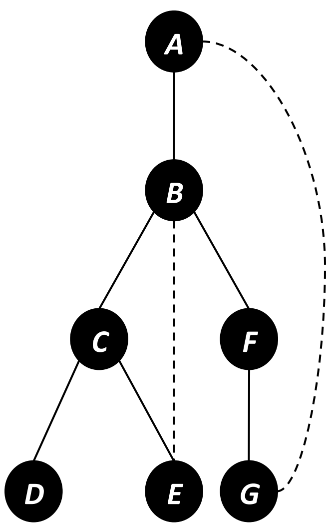


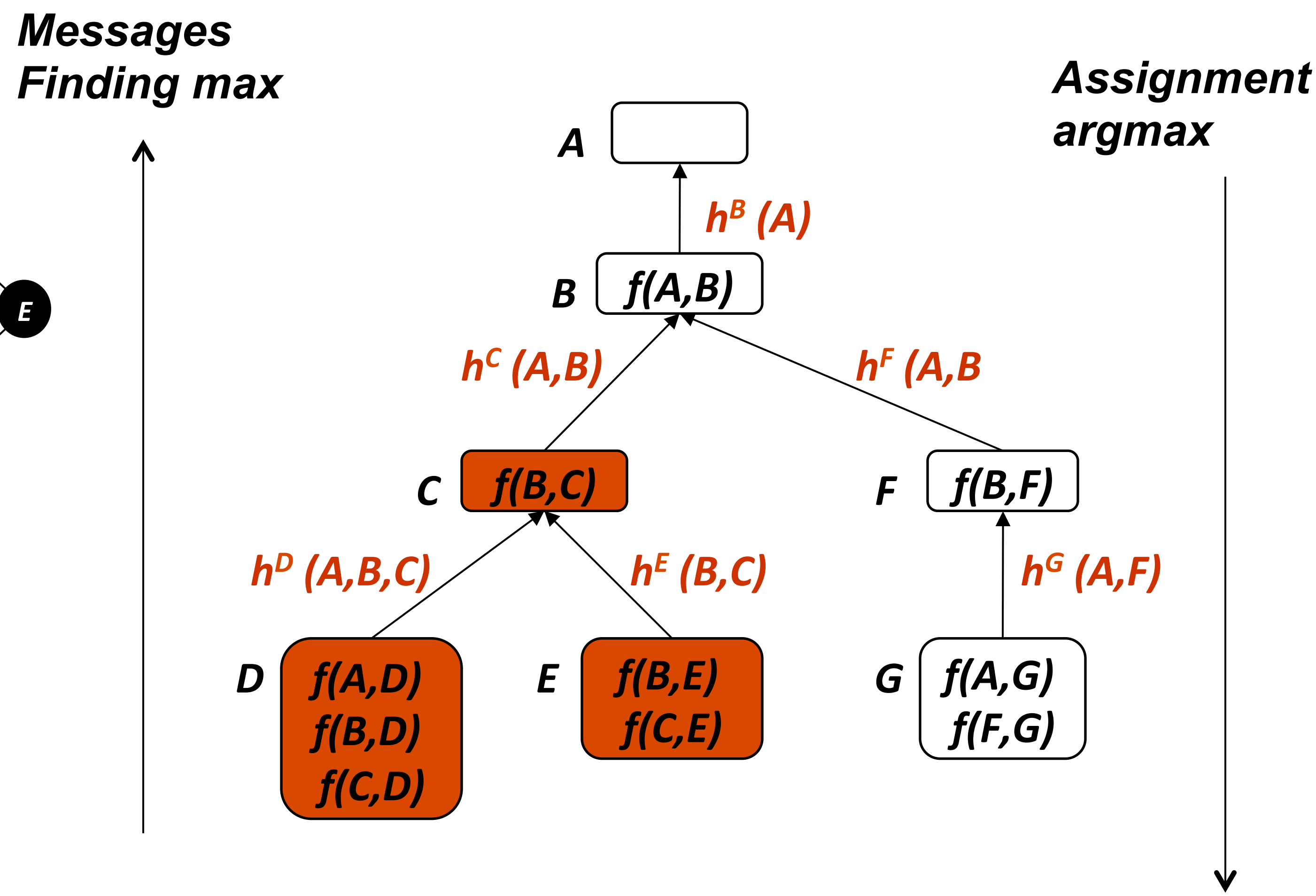

$w^*(d_2) = 2$

# Bucket Elimination

$$\min_{a,b,c,d,e,f,g} f(a,b)+f(a,d)+f(b,c)+f(a,d)+f(b,d)+ f(c,d)+f(b,e)+f(c,e)+f(b,f)+f(a,g)+f(f,g)=$$



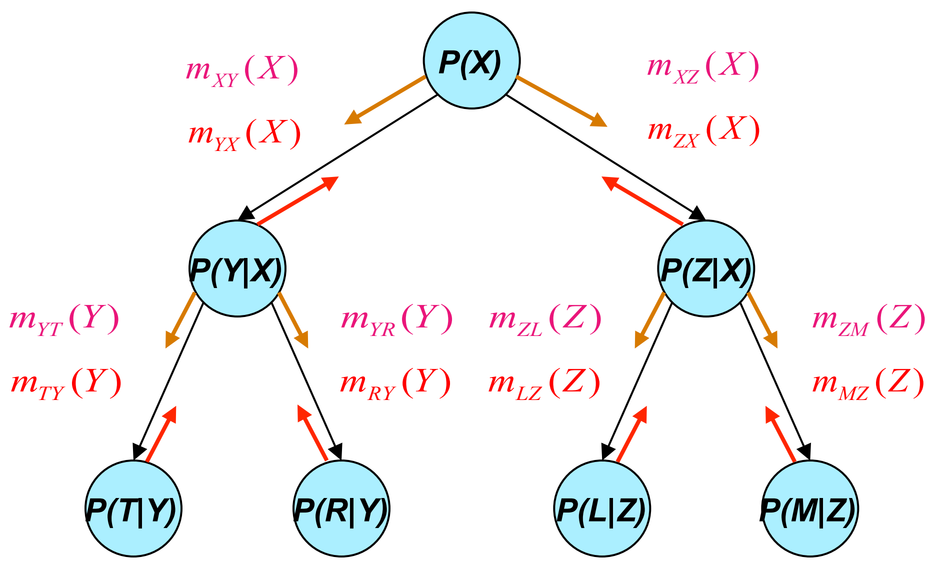




*Ordering: (A, B, C, D, E, F, G)*

# Tree-solving is Easy

***Belief updating (sum-prod)***

***CSP – consistency (projection-join)***




***Dynamic Programming, Inference***

***MPE (max-prod)***

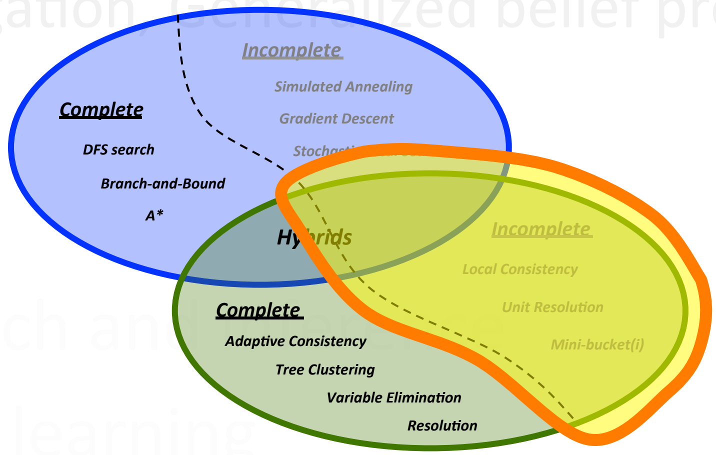
***#CSP (sum-prod)***

***Trees are processed in linear time and memory***
***Message-passing***

BREN:ICS INFORMATION AND COMPUTER SCIENCES — UNIVERSITY of CALIFORNIA IRVINE

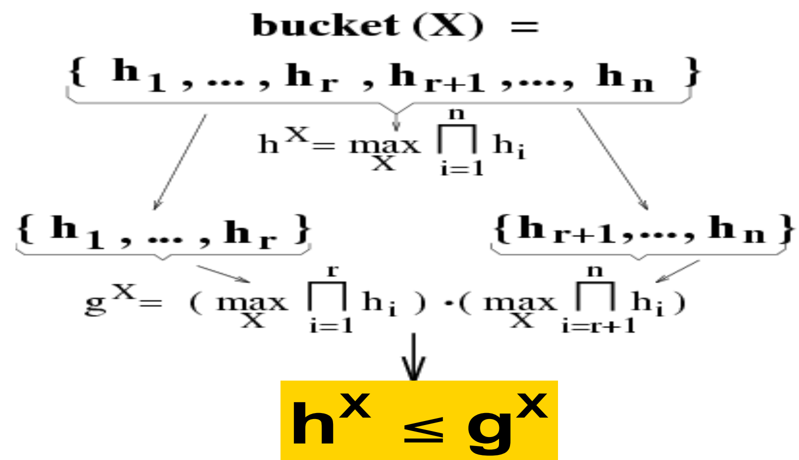
# Bounded Inference

- Overview: Bayesian networks and algorithms
- Exact Inference
- Bounded-inference
  - Mini-buckets, mini-clusters
  - Belief propagation, Generalized belief propagation
- Search
- Sampling
- Hybrid of search and
- Modeling and learning
- Software

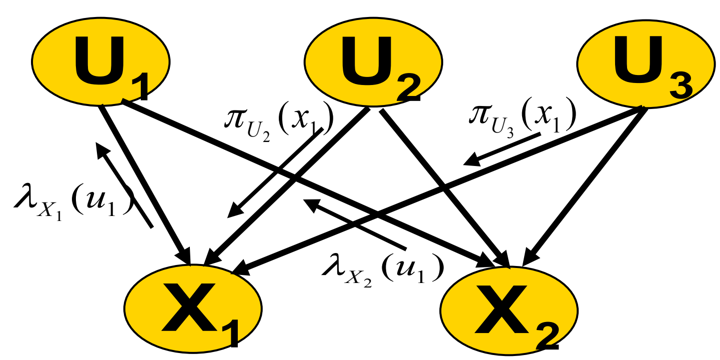


# Two Principles for Bounded Inference

- **Bounded-Partitioning**
  - **mini-bucket(i), MC(i)**
  - **Computes a bound**
  - **Exp(i) time, space**



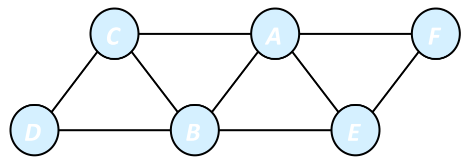

- **Belief propagation**
  - **Loopy BP**
  - **No guarantees, unless constraint propagation**
- **Generalized BP**
  - **IJGP(i)**
  - **Each iteration is exp(i)**



# Outline

- Graphical models and reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments

# The Search Space




| A | B | $f_1$ |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 4 |

| A | C | $f_2$ |
|---|---|---|
| 0 | 0 | 3 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | E | $f_3$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 3 |
| 1 | 0 | 2 |
| 1 | 1 | 0 |

| A | F | $f_4$ |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 2 |

| B | C | $f_5$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 4 |

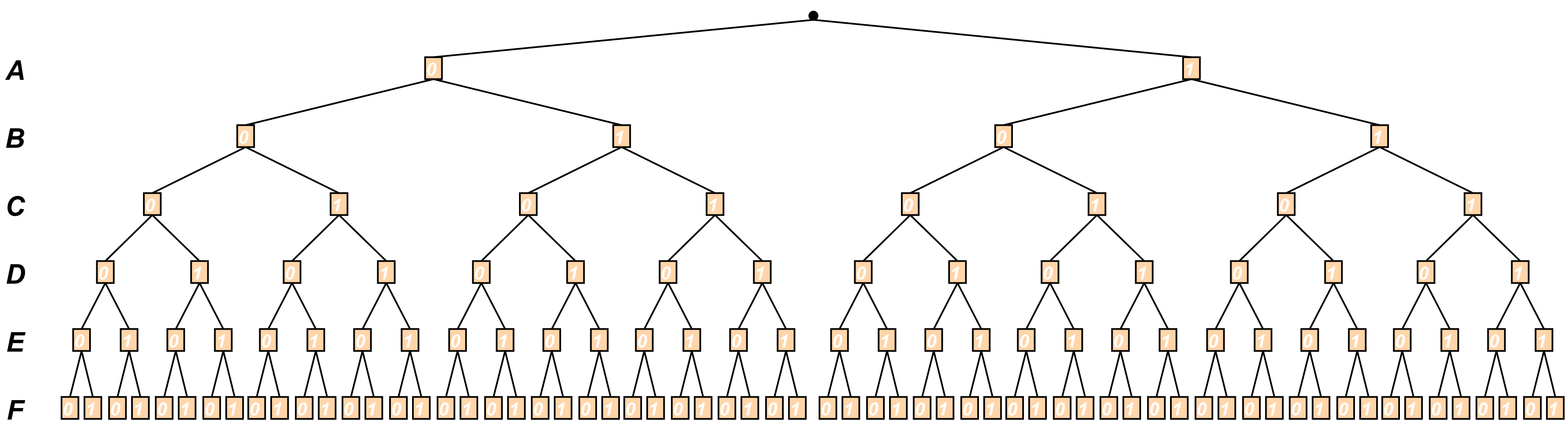
| B | D | $f_6$ |
|---|---|---|
| 0 | 0 | 4 |
| 0 | 1 | 2 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| B | E | $f_7$ |
|---|---|---|
| 0 | 0 | 3 |
| 0 | 1 | 2 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

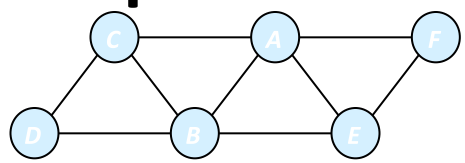
| C | D | $f_8$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 4 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

| E | F | $f_9$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 2 |

$$f(X) = \sum_{i=1}^{9} f_i(X)$$




# An Optimal Solution




| A | B | $f_1$ |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 4 |

| A | C | $f_2$ |
|---|---|---|
| 0 | 0 | 3 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | E | $f_3$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 3 |
| 1 | 0 | 2 |
| 1 | 1 | 0 |

| A | F | $f_4$ |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 2 |

| B | C | $f_5$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 4 |

| B | D | $f_6$ |
|---|---|---|
| 0 | 0 | 4 |
| 0 | 1 | 2 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

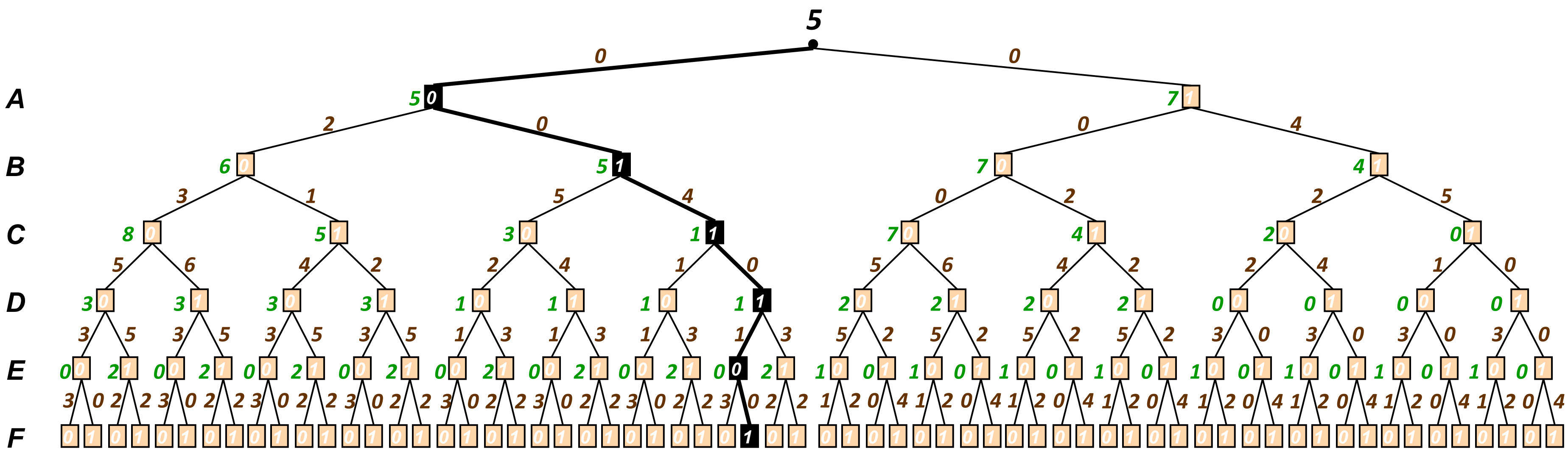
| B | E | $f_7$ |
|---|---|---|
| 0 | 0 | 3 |
| 0 | 1 | 2 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| C | D | $f_8$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 4 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

| E | F | $f_9$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 2 |

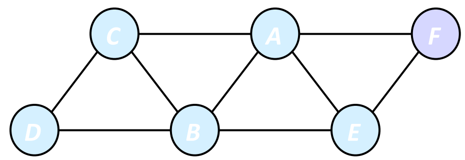
$$f(\mathrm{X}) = \sum_{i=1}^{9} f_i(\mathrm{X})$$

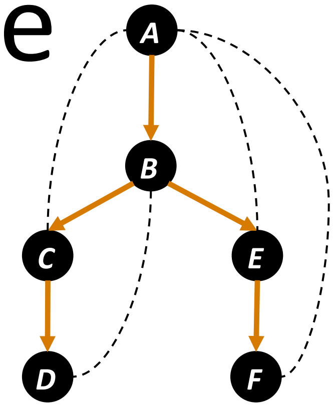
$$\min_{a,b,c,d,e,f} f_1(a,b) + f_2(a,c) + f_3(a,f) + f_4(b,c) + f_5(b,d) + f_6(b,e) + f_7(c,d) + f_8(e,f)$$



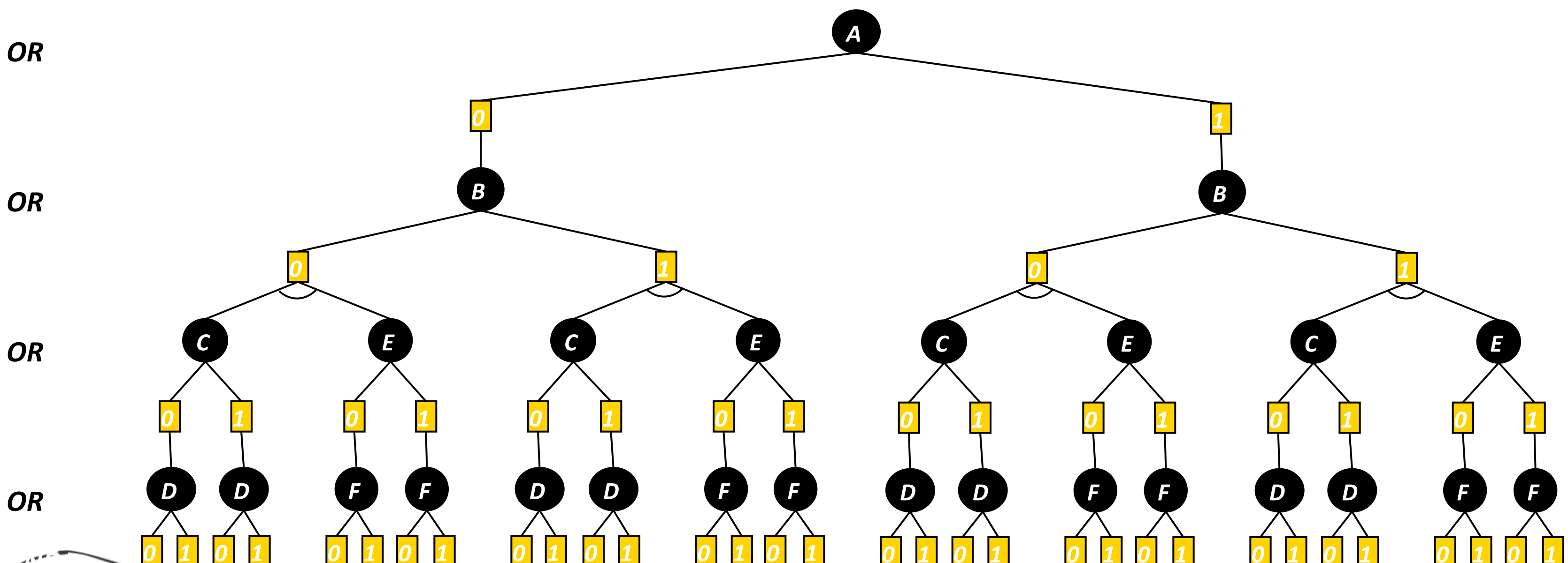

***An optimal assignment is A=0, B=1, C=1, D=1, E=0, F=1 with cost 5***

# The AND/OR Search Tree

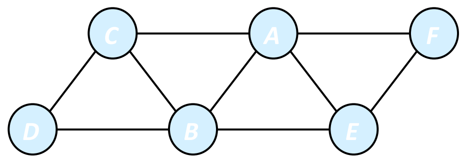


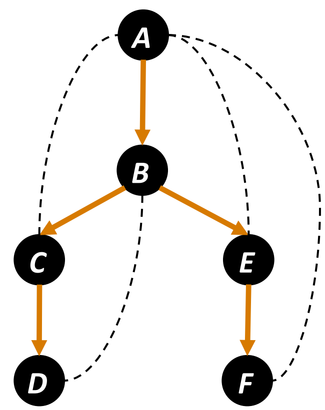




***Pseudo tree** (Freuder & Quinn85)*



# Weighted AND/OR Search Tree




| A | B | $f_1$ |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 4 |

| A | C | $f_2$ |
|---|---|---|
| 0 | 0 | 3 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | E | $f_3$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 3 |
| 1 | 0 | 2 |
| 1 | 1 | 0 |

| A | F | $f_4$ |
|---|---|---|
| 0 | 0 | 2 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 2 |

| B | C | $f_5$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 4 |

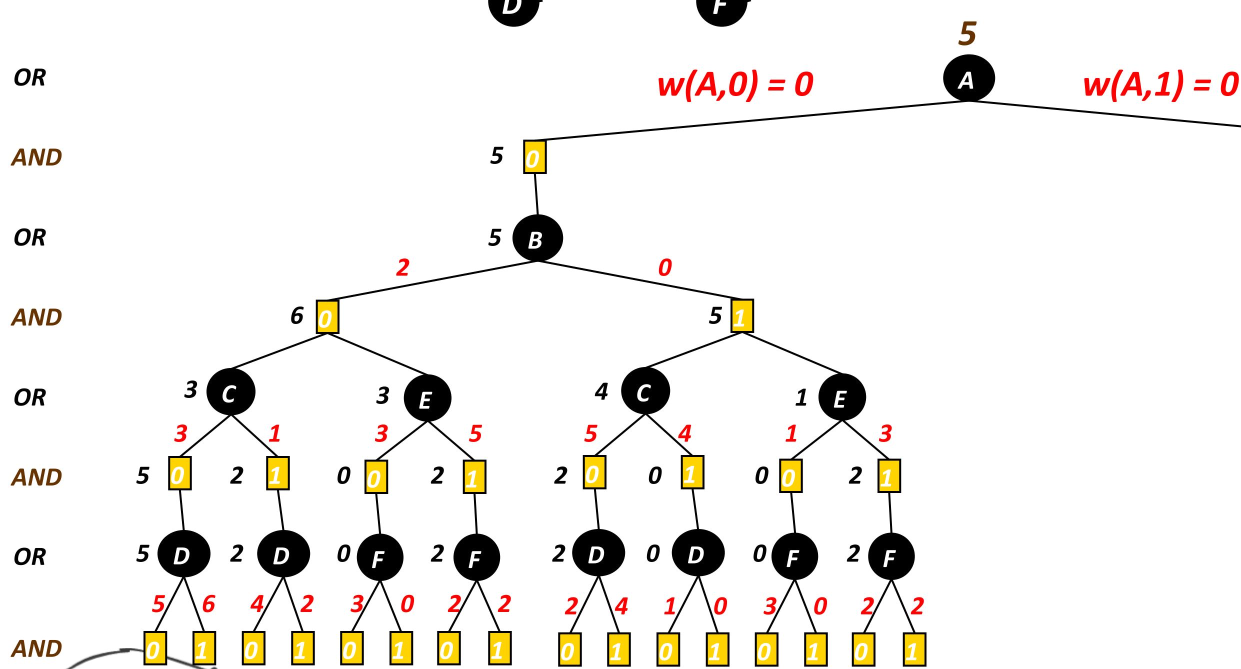
| B | D | $f_6$ |
|---|---|---|
| 0 | 0 | 4 |
| 0 | 1 | 2 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| B | E | $f_7$ |
|---|---|---|
| 0 | 0 | 3 |
| 0 | 1 | 2 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

| C | D | $f_8$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 4 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

| E | F | $f_9$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 2 |

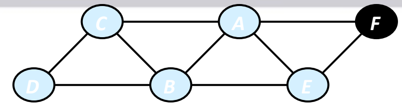
$$f(\mathrm{X}) = \sum_{i=1}^{9} f_i(\mathrm{X})$$

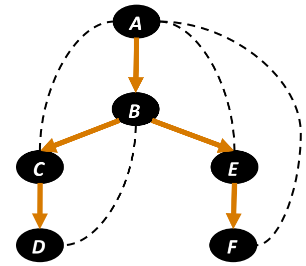


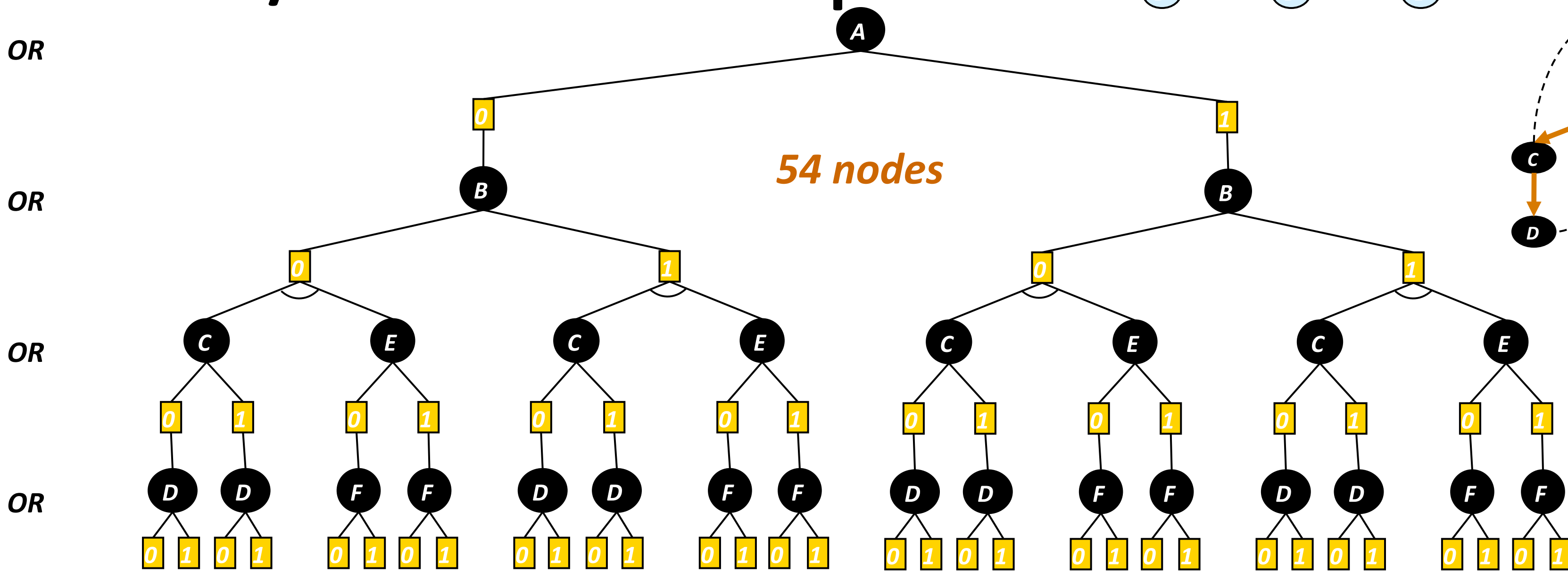

***Node Value***
***(bottom-up evaluation)***

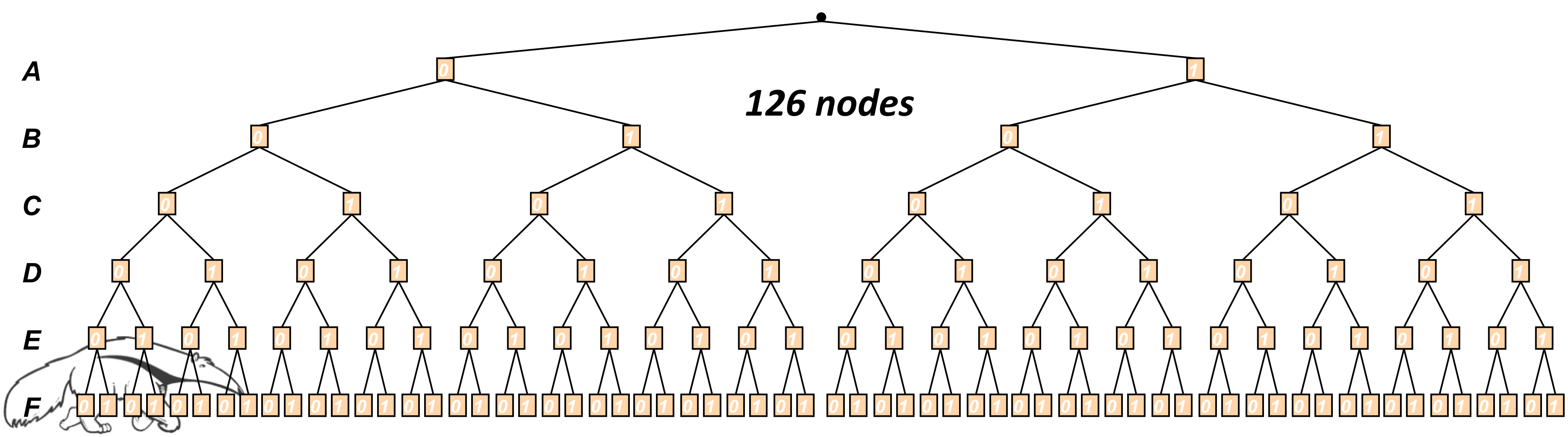
***OR – minimization***
***AND – summation***

# AND/OR vs. OR Spaces

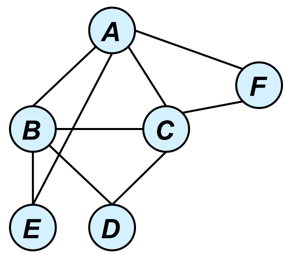


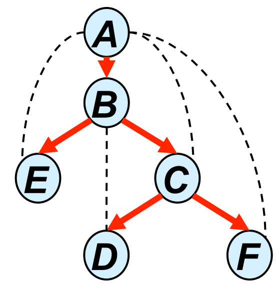



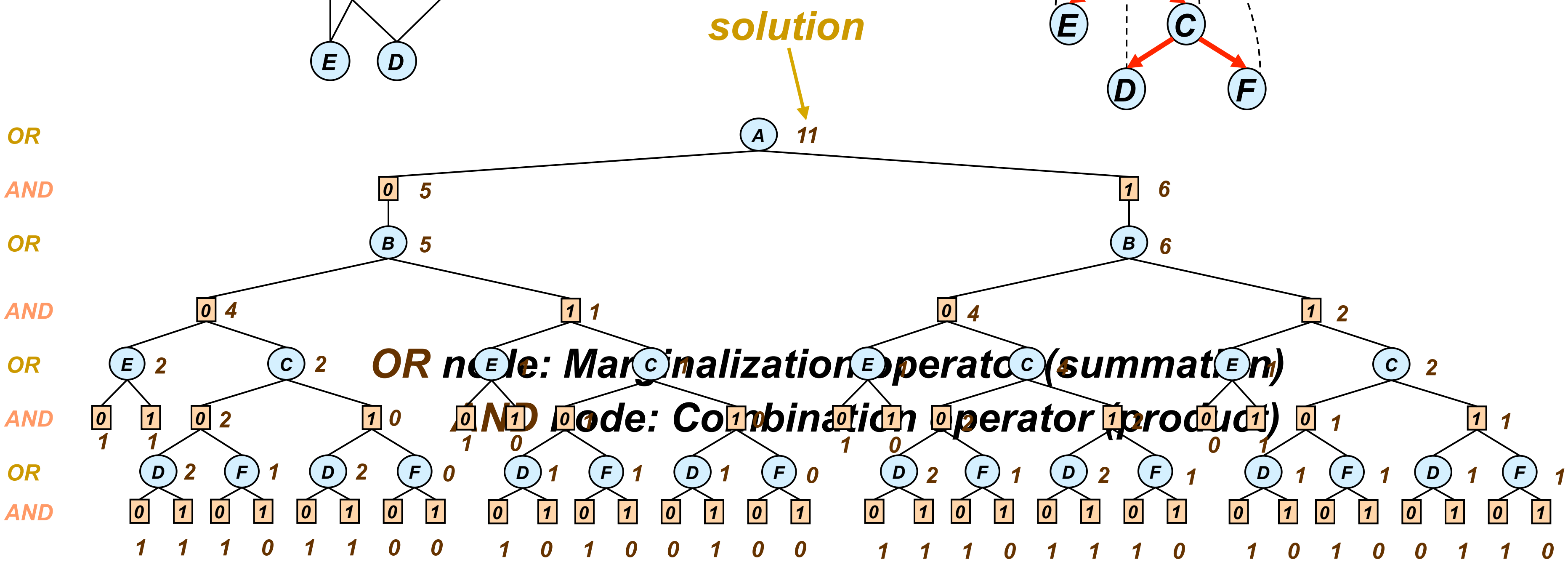



# DFS algorithm (#CSP example)








**OR node: Marginalization operator (summation)**

**AND node: Combination operator (product)**

***Value of node = number of solutions below it***

# AND/OR Tree DFS Algorithm (Belief Updating)

$P(E \mid A,B)$

| A | B | E=0 | E=1 |
|---|---|---|---|
| 0 | 0 | .4 | .6 |
| 0 | 1 | .5 | .5 |
| 1 | 0 | .7 | .3 |
| 1 | 1 | .2 | .8 |

Evidence: E=0

$P(B \mid A)$

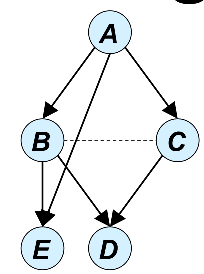
| A | B=0 | B=1 |
|---|---|---|
| 0 | .4 | .6 |
| 1 | .1 | .9 |

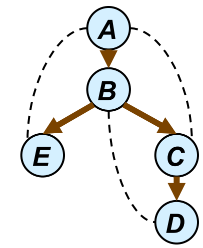
$P(C \mid A)$

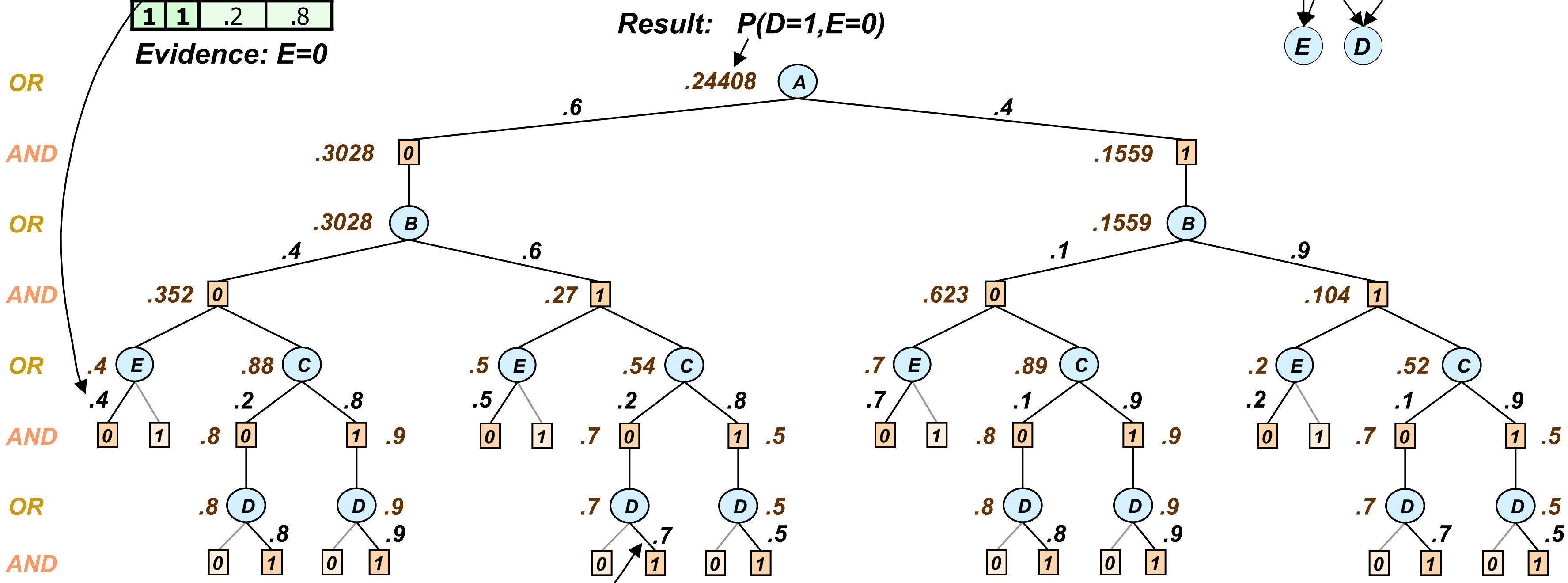
| A | C=0 | C=1 |
|---|---|---|
| 0 | .2 | .8 |
| 1 | .7 | .3 |

$P(A)$

| A | P(A) |
|---|---|
| 0 | .6 |
| 1 | .4 |




$P(D \mid B,C)$

| B | C | D=0 | D=1 |
|---|---|---|---|
| 0 | 0 | .2 | .8 |
| 0 | 1 | .1 | .9 |
| 1 | 0 | .3 | .7 |
| 1 | 1 | .5 | .5 |

Evidence: D=1

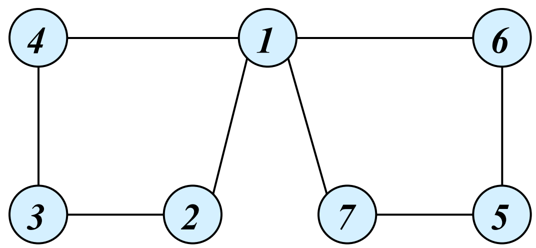
**OR node: Marginalization operator (summation)**

**AND node: Combination operator (product)**

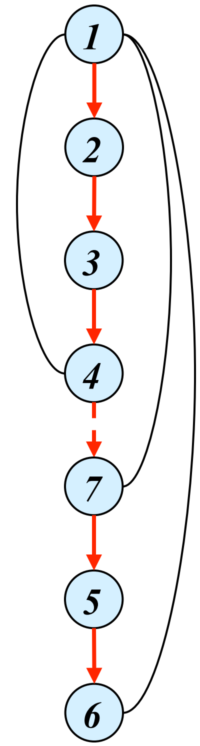
**Value of node = updated belief for sub-problem below**

# Pseudo-Trees

(Freuder 85, Bayardo 95, Bodlaender and Gilbert, 91)

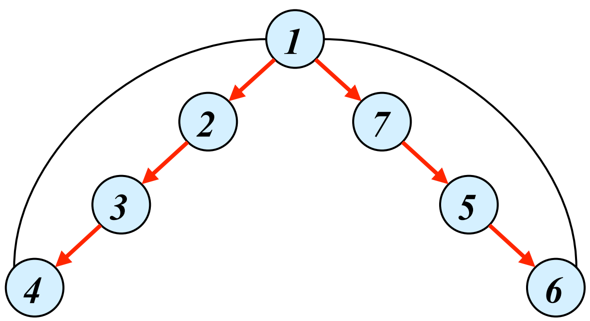



*(a) Graph*

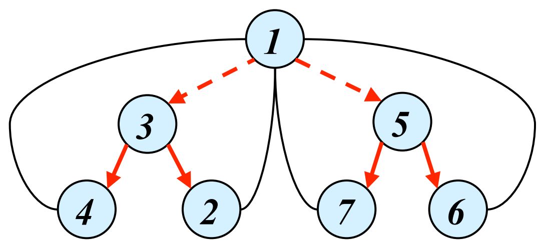



*(d) Chain*
*depth=6*

$$m <= w^* \log n$$




*(b) DFS tree*
*depth=3*




*(c) pseudo- tree*
*depth=2*

# Complexity of AND/OR Tree Search

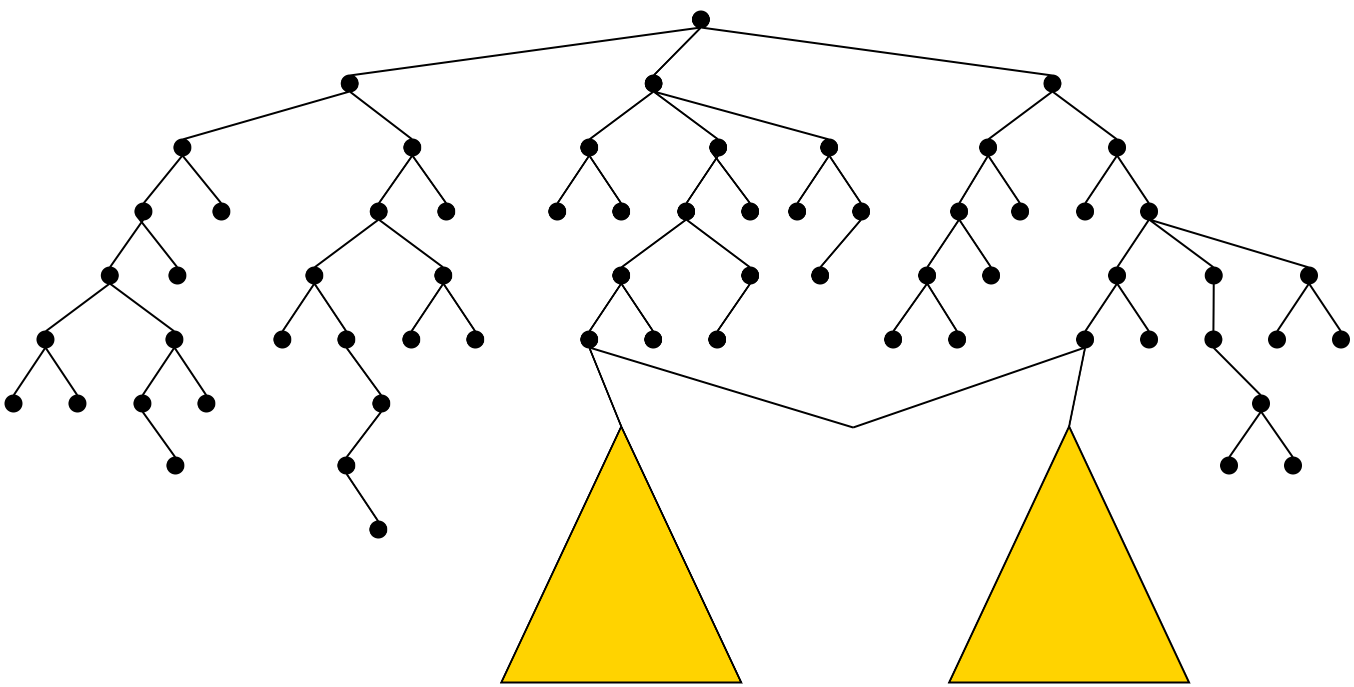
| | AND/OR tree | OR tree |
|---|---|---|
| Space | $O(n)$ | $O(n)$ |
| Time | $O(n\ k^m)$<br>$O(n\ k^{w^* \log n})$<br>[Freuder & Quinn85], [Collin, Dechter & Katz91], [Bayardo & Miranker95], [Darwiche01] | $O(k^n)$ |

***k*** **= *domain size***
***m*** **= *depth of pseudo-tree***
***n*** **= *number of variables***
***w******= *treewidth***

***Tasks:* *Consistency, Counting, Optimization, Belief updating Max-expected utility, partition function***

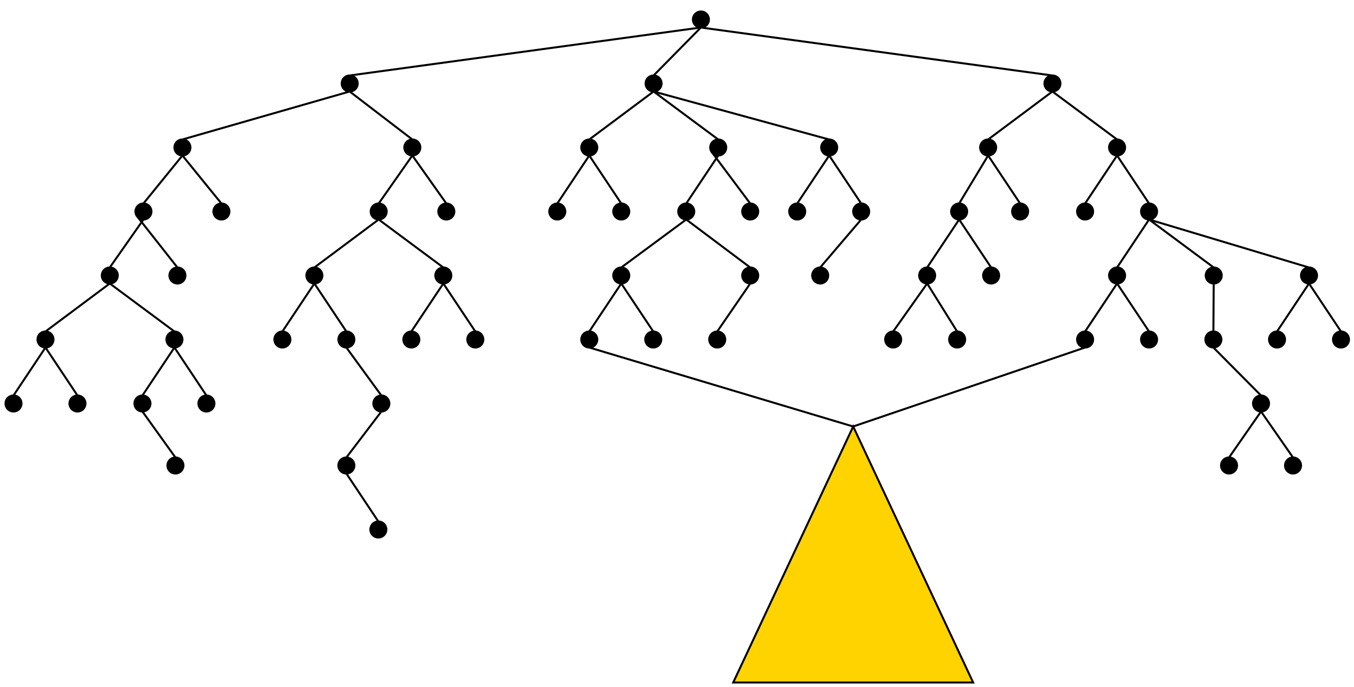
# From Search Trees to Search Graphs

- Any two nodes that root identical sub-trees or sub-graphs can be merged

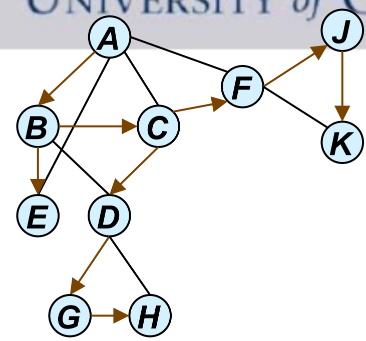


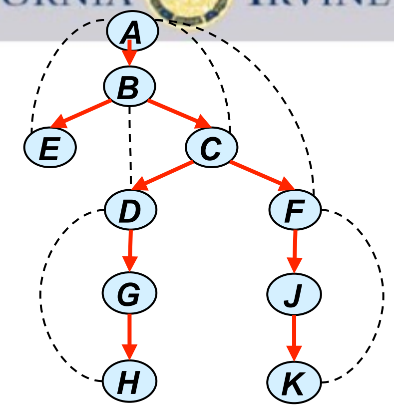
# From Search Trees to Search Graphs

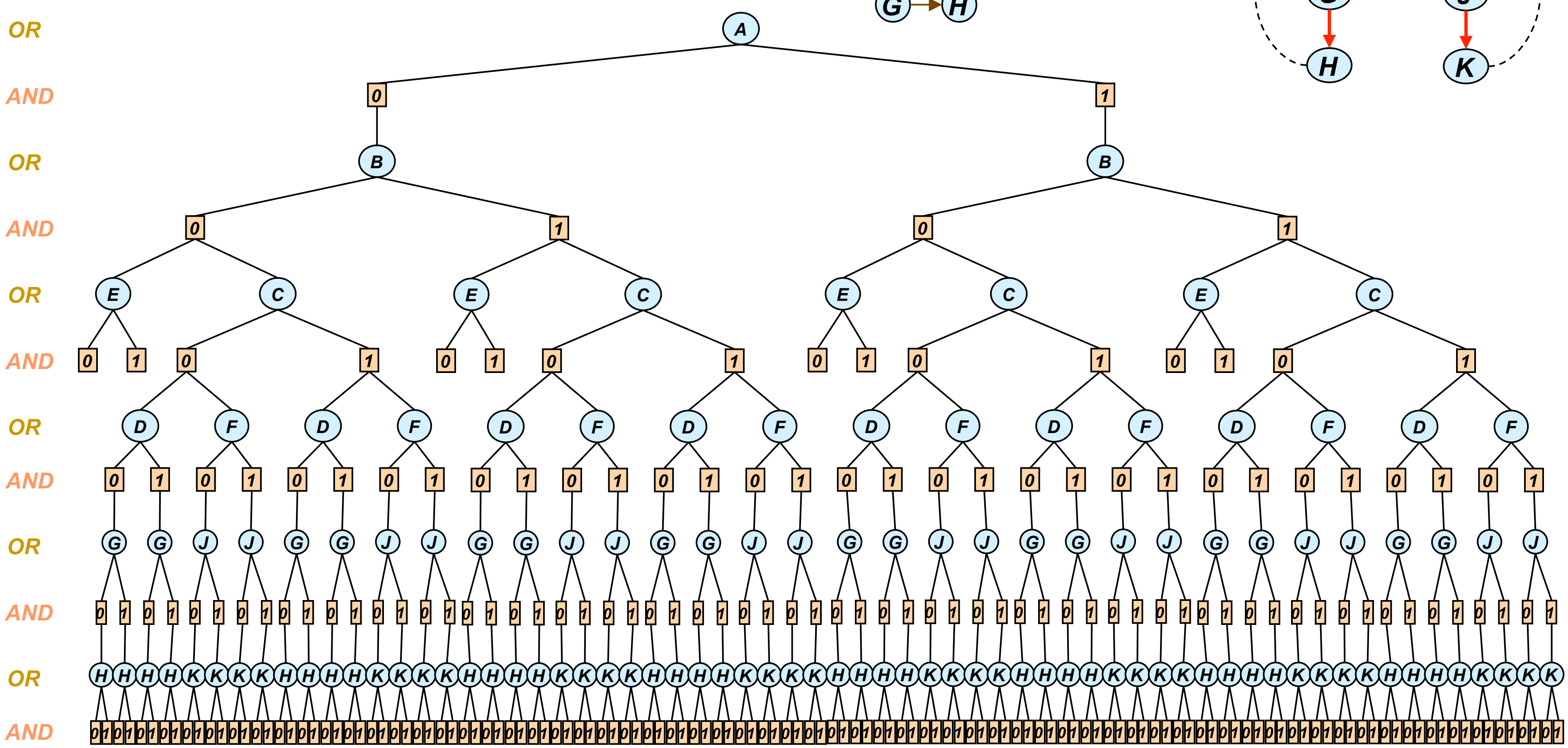
- Any two nodes that root identical sub-trees or sub-graphs can be merged



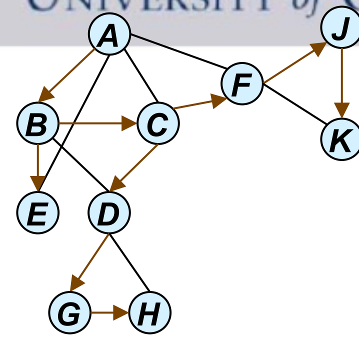
# From AND/OR Tree

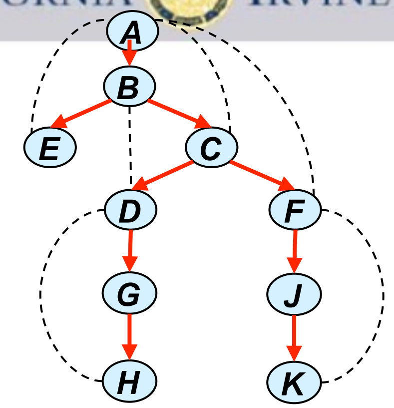


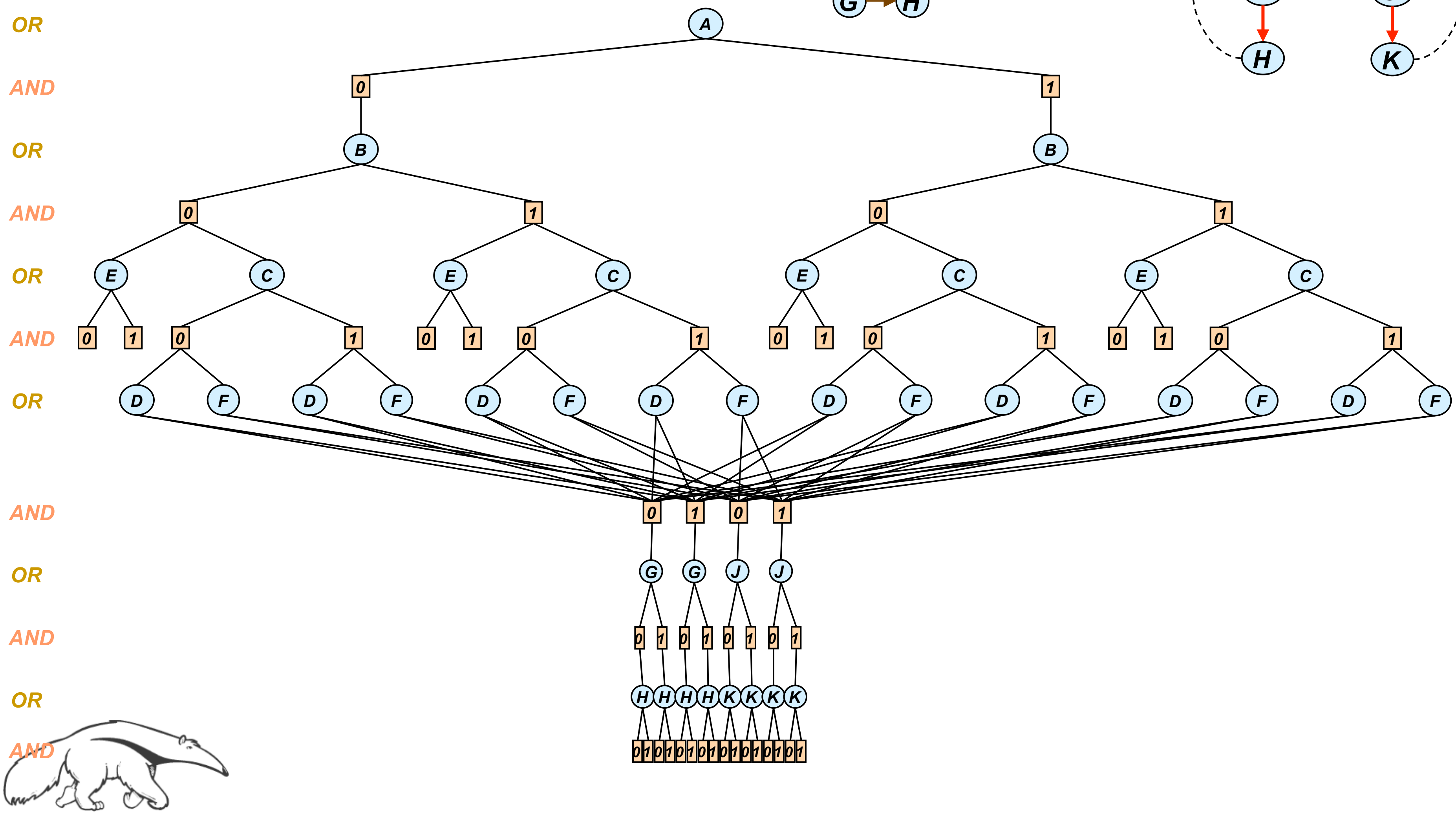




# An AND/OR Graph



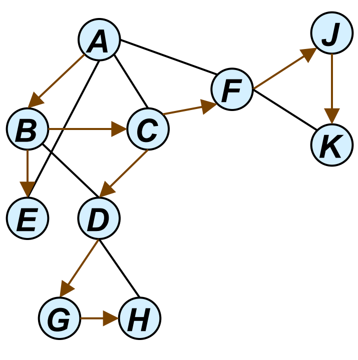



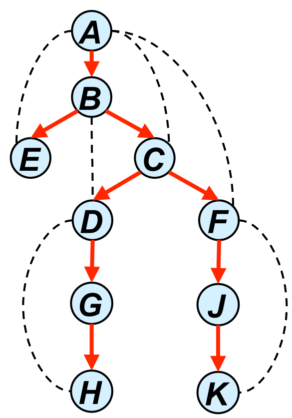



# Context-based Caching

- Caching is possible when context is the same

- context = parent-separator set in induced pseudo-graph
  = current variable +
  parents connected to subtree below




***context(B) = {A, B}***

***context(c) = {A,B,C}***

***context(D) = {D}***

***context(F) = {F}***

# AND/OR Graph DFS Algorithm (Belief Updating)

$P(E \mid A,B)$

| A | B | E=0 | E=1 |
|---|---|---|---|
| 0 | 0 | .4 | .6 |
| 0 | 1 | .5 | .5 |
| 1 | 0 | .7 | .3 |
| 1 | 1 | .2 | .8 |

**Evidence: E=0**

$P(B \mid A)$

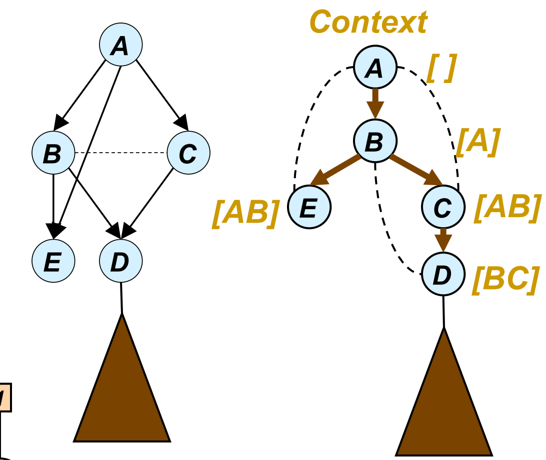
| A | B=0 | B=1 |
|---|---|---|
| 0 | .4 | .6 |
| 1 | .1 | .9 |

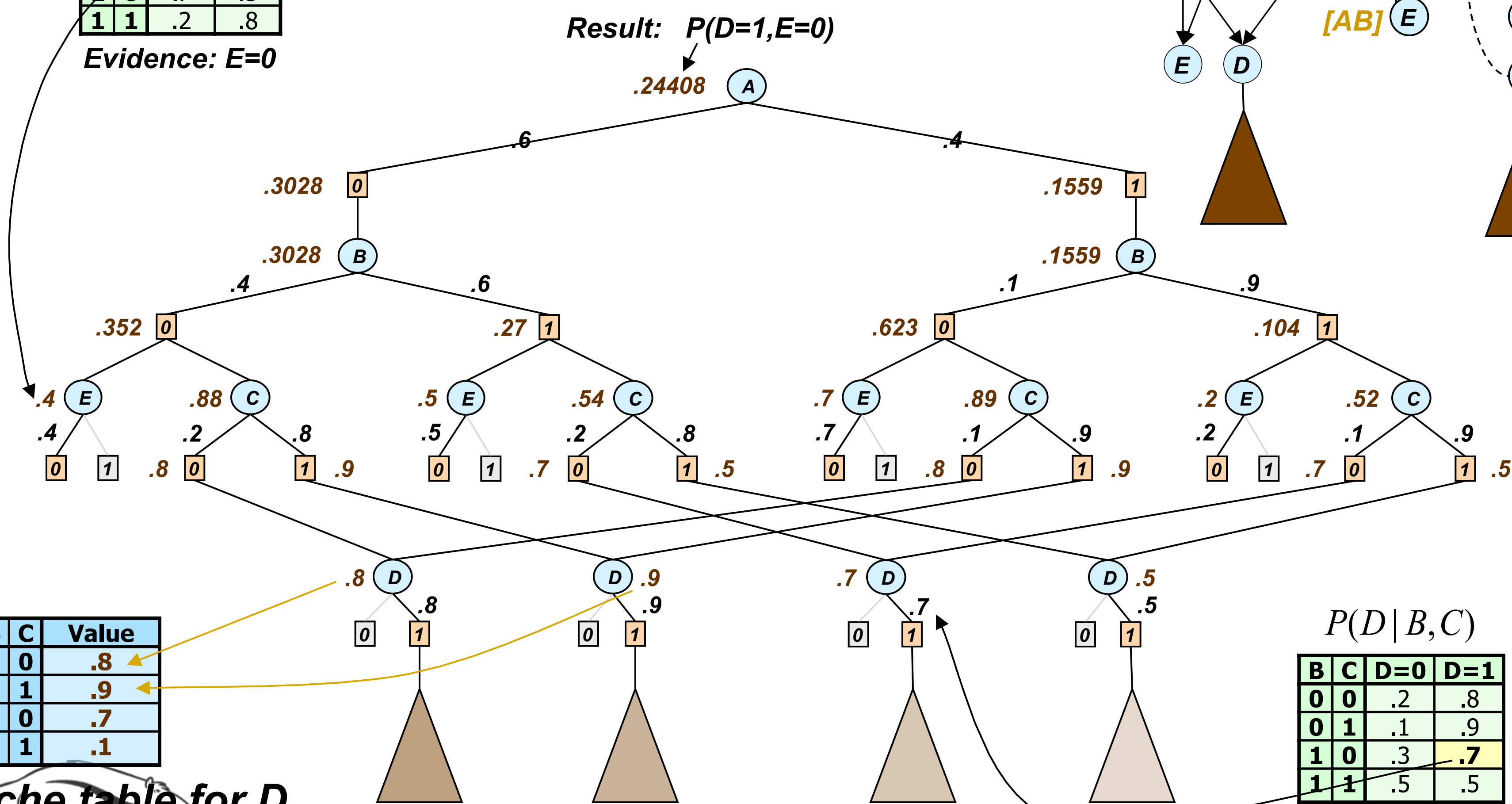
$P(C \mid A)$

| A | C=0 | C=1 |
|---|---|---|
| 0 | .2 | .8 |
| 1 | .7 | .3 |

$P(A)$

| A | P(A) |
|---|---|
| 0 | .6 |
| 1 | .4 |




| B | C | Value |
|---|---|---|
| 0 | 0 | .8 |
| 0 | 1 | .9 |
| 1 | 0 | .7 |
| 1 | 1 | .1 |

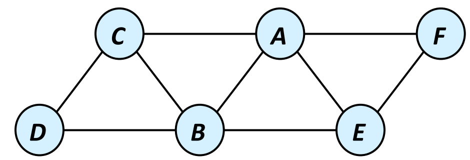
**Cache table for D**

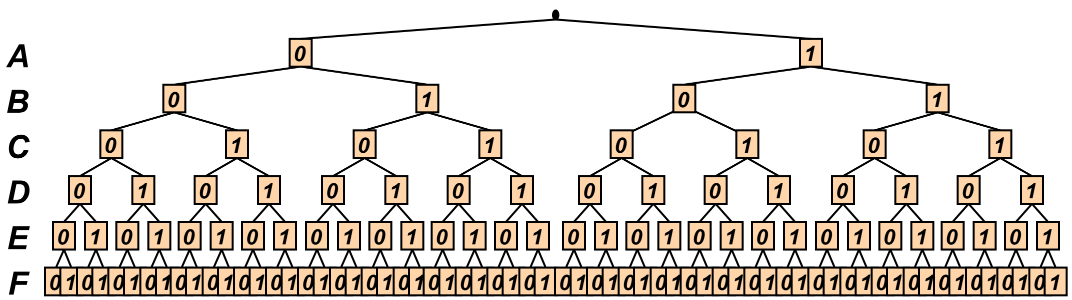
$P(D \mid B,C)$

| B | C | D=0 | D=1 |
|---|---|---|---|
| 0 | 0 | .2 | .8 |
| 0 | 1 | .1 | .9 |
| 1 | 0 | .3 | .7 |
| 1 | 1 | .5 | .5 |

**Evidence: D=1**

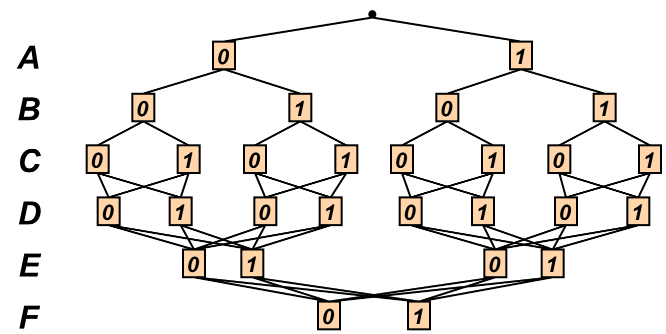
# All Four Search Spaces



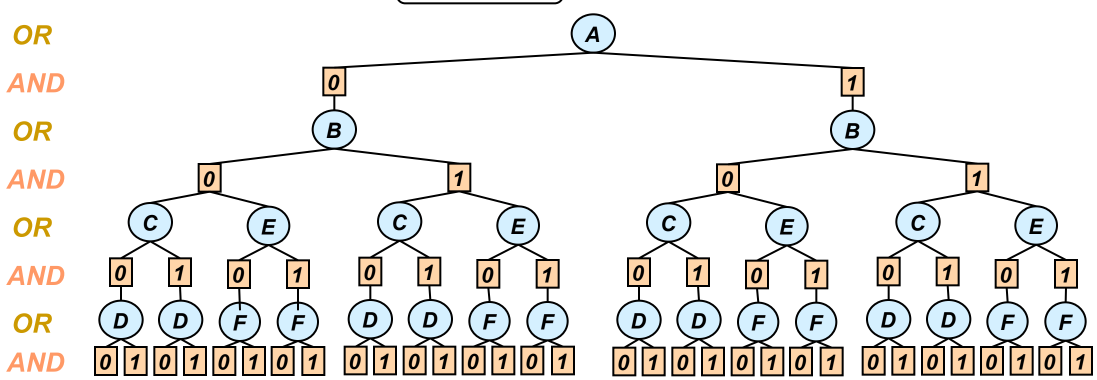

Full OR search tree

126 nodes



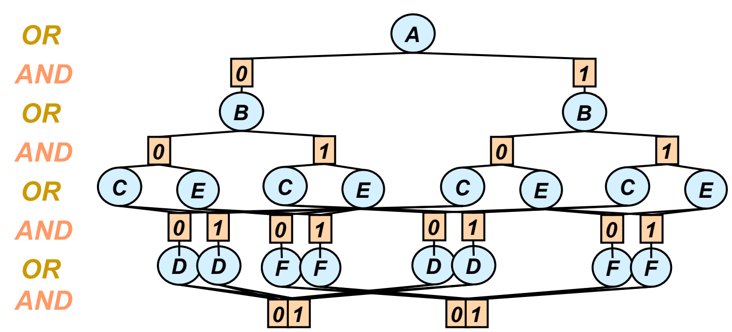

Context minimal OR search graph

28 nodes




Full AND/OR search tree

54 AND nodes




Context minimal AND/OR search graph

18 AND nodes

*Any query is best computed*
*Over the c-minimal AO space*

# Complexity of AND/OR Graph Search

| | AND/OR graph | OR graph |
|---|---|---|
| **Space** | $O(n\ k^{w^*})$ | $O(n\ k^{pw^*})$ |
| **Time** | $O(n\ k^{w^*})$ | $O(n\ k^{pw^*})$ |

*k = domain size*
*n = number of variables*
*w*= treewidth*
*pw*= pathwidth*

*Tasks: Consistency, Counting, Optimization, Belief updating Max-expected utility, partition function*

# AOC – Adaptive Caching

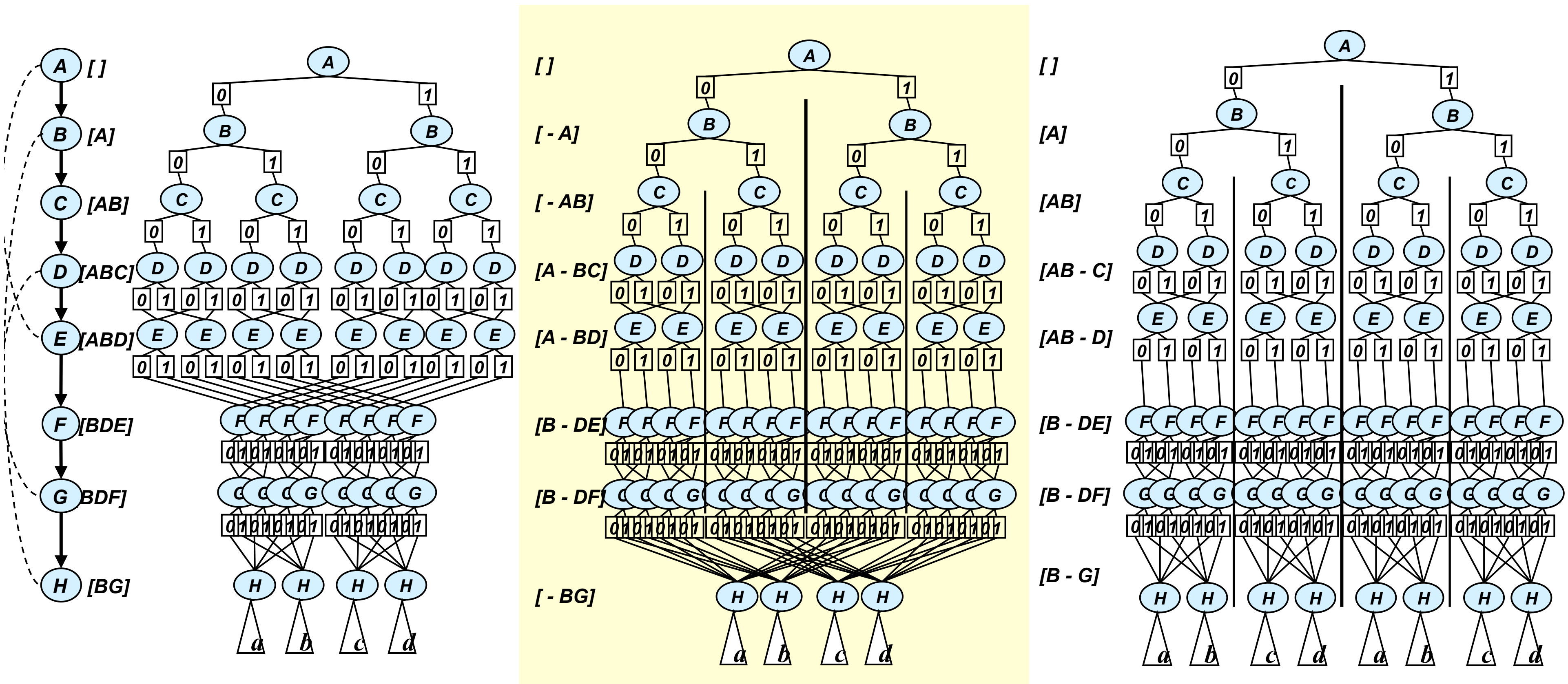
- i-bound – limit for cache size
- If context(X) = $[X_1 \ldots X_k]$ and k>i then
  i-context(X) = $[X_{k-i+1} \ldots X_k]$

$$[X_1 \ldots X_{k-i}\, X_{k-i+1} \ldots X_k]$$

*cutset* *i-context*

- i-cache is purged when search retracts to $X_{k-i}$

# Adaptive Caching Search



**Context minimal graph (full caching)**

**AOC(2) graph (Adaptive Caching)**

**AOCutset(2) graph (AND/OR Cutset)**

# Searching AND/OR Graphs

- AO(j): searches depth-first, cache i-context
  - j = the max size of a cache table (i.e. number of variables in a context)

*i=0* — *j* — *i=w**

*Space: O(n)*
*Time: O(exp(w* log n))*

*Space: O(exp(j) )*
*Time: O(exp(m_j+j )*

*Space: O(exp w*)*
*Time: O(exp w*)*

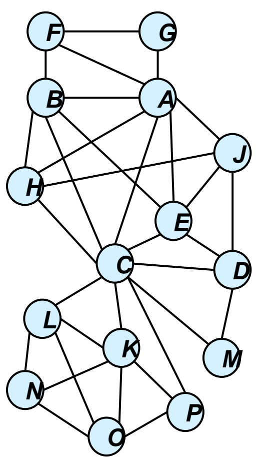
# Sampling: approximating search

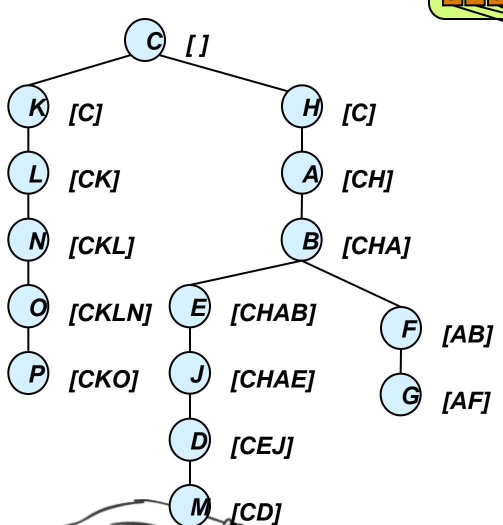
- Gibbs Sampling: An MCMC approach
- Likelihood weighting: An importance sampling approach
- Exploit structure
  - Cutset-sampling (likelihood and Gibbs)
  - SamplingSearch (avoid inconsistency)
  - AND/OR sampling

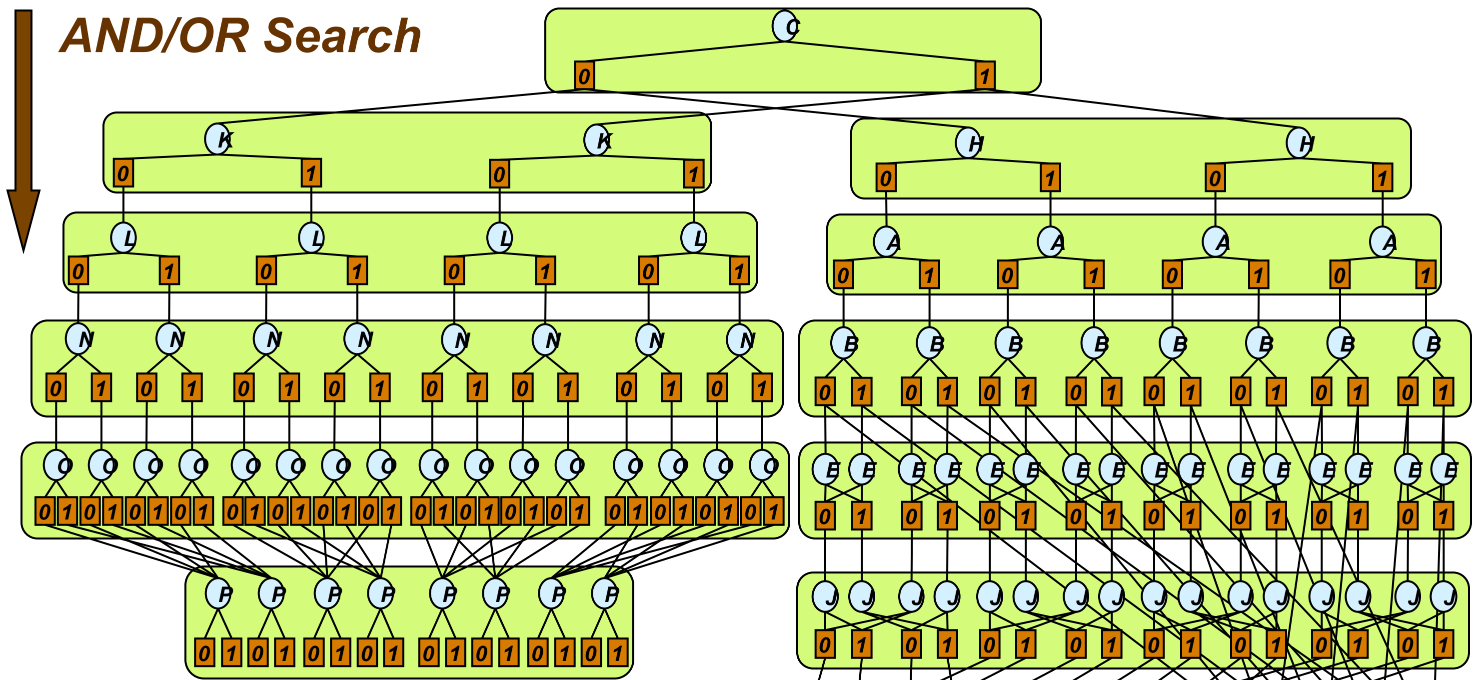
# Outline

- Graphical models and reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments

# AND/OR Context Minimal Graph

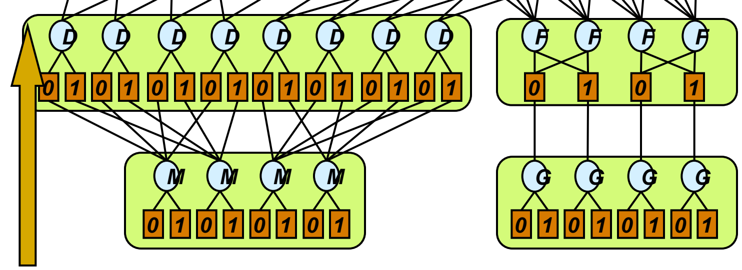


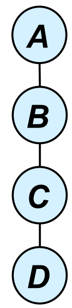

***THEOREM*** *(Mateescu and Dechter, 2005)* **:**

***For positive models and given a pseudo tree, AND/OR search and Bucket-elimination have the performance***



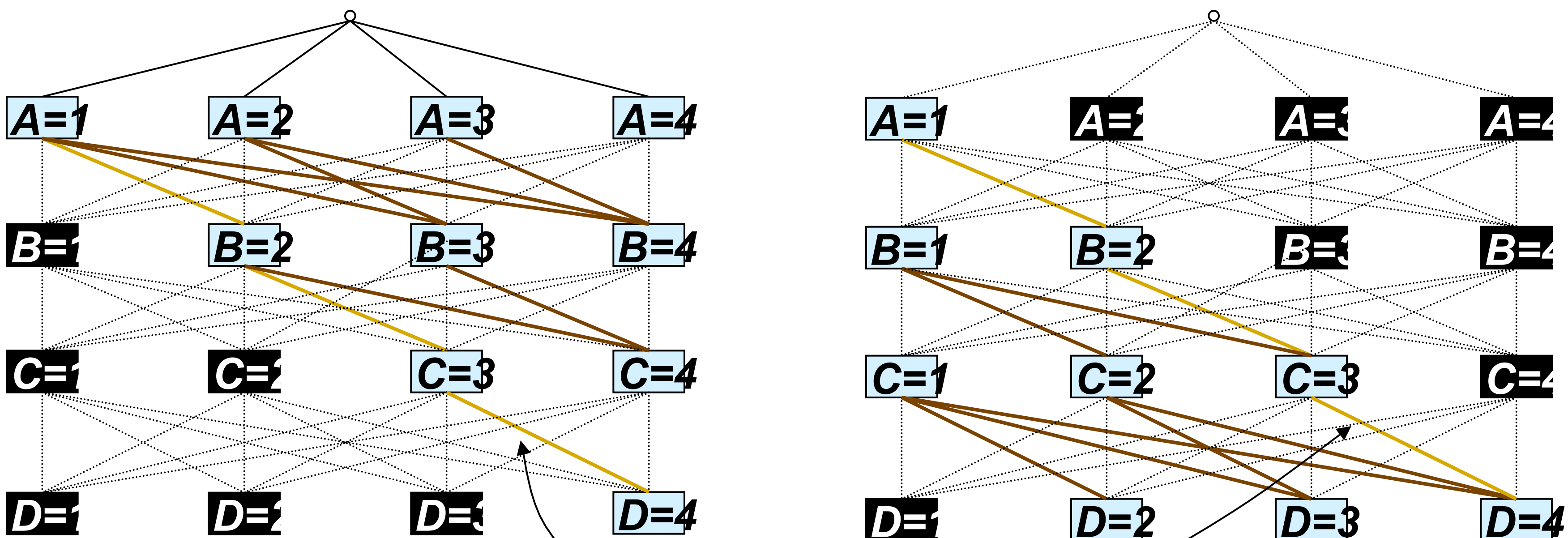

***(C K H A B E J L N O D P M F G)***

# AO vs. VE with Determinism




*Domains: {1, 2, 3, 4}*

*A < B   B < C   C < D*



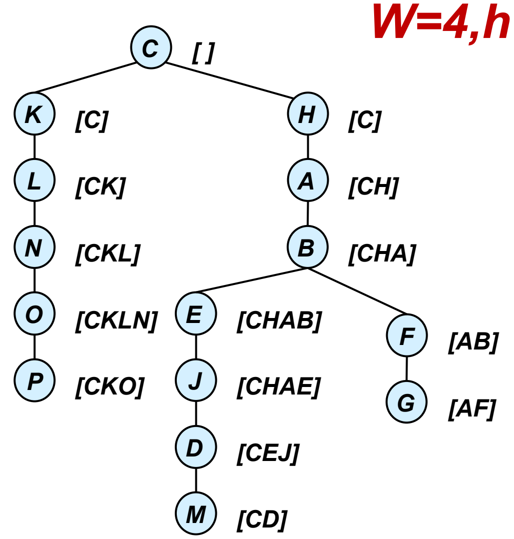

*AND/OR Search*

*Variable Elimination*

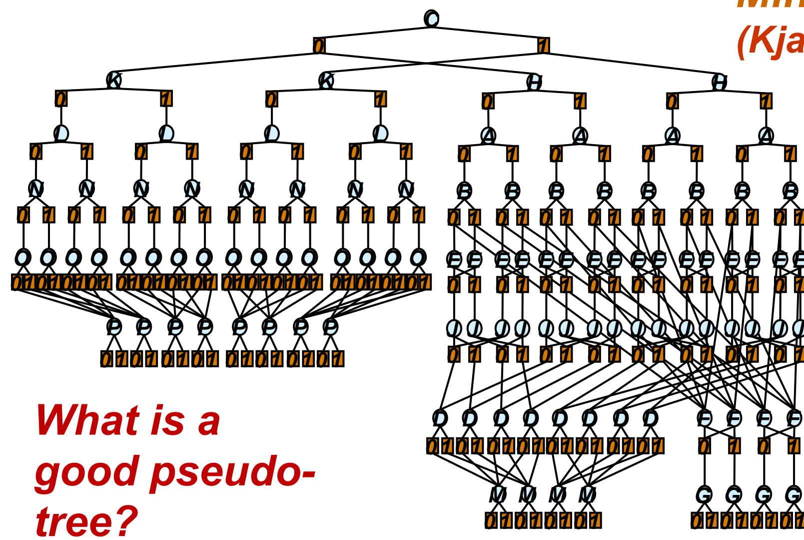
*Backtrack-free graph*

# The impact of the pseudo-tree

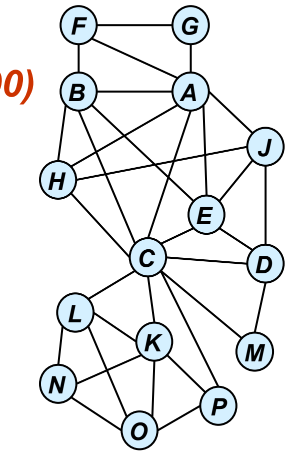
**W=4,h=8**




(C K H A B E J L N O D P M F G)

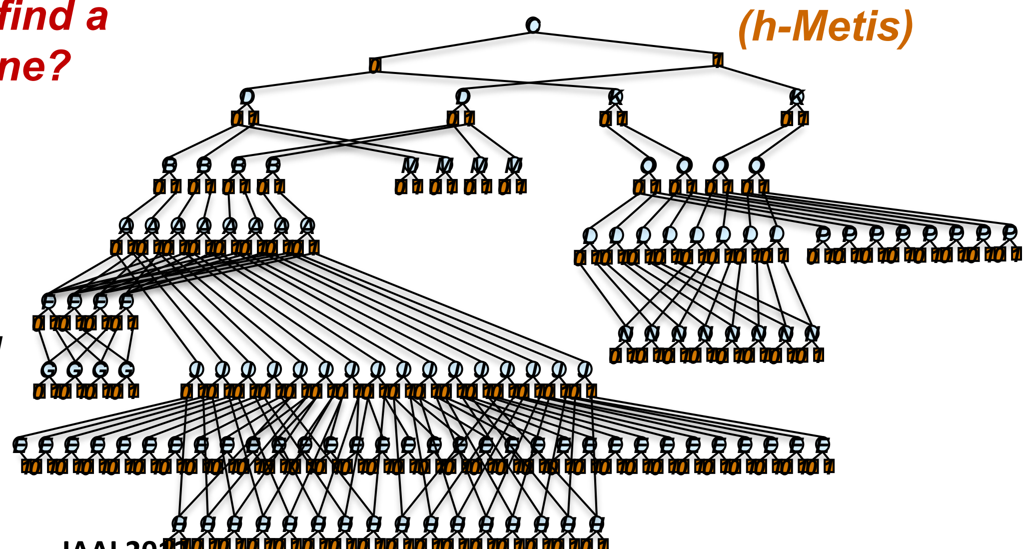


**Min-Fill (Kjaerulff90)**

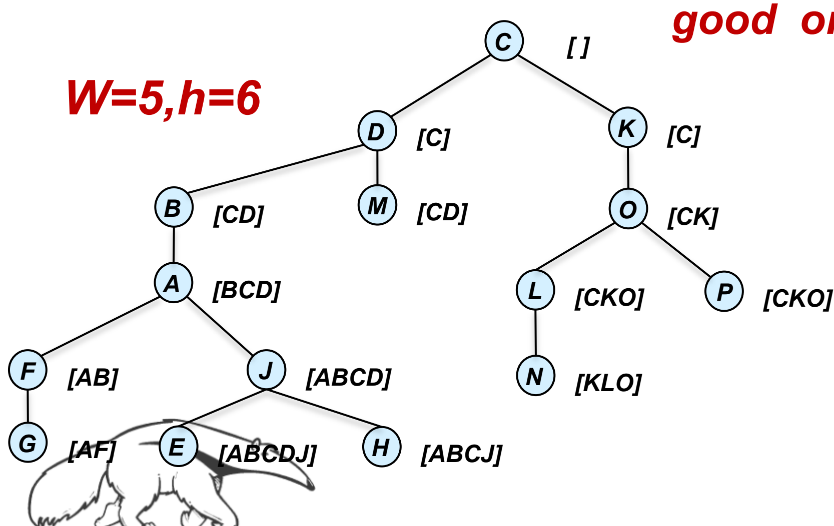



**What is a good pseudo-tree?**
**How to find a good one?**

**Hypergraph Partitioning (h-Metis)**



**W=5,h=6**




(C D K B A O M L N P J H E F G)

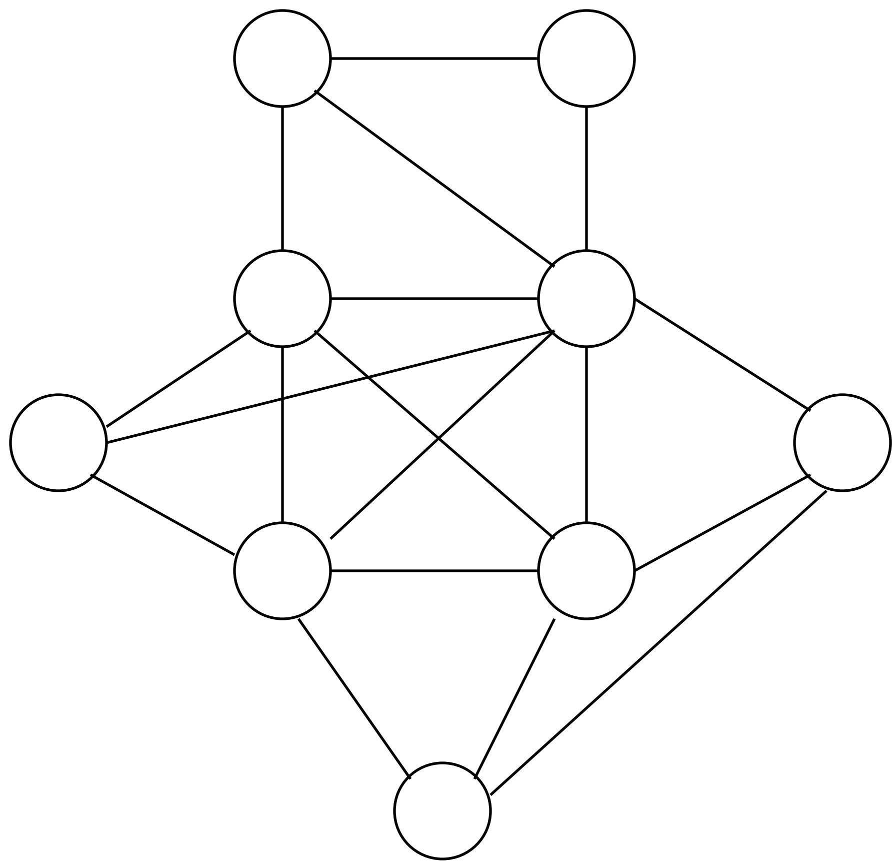
# Outline

- Graphical models and reasoning principles
- Inference
- AND/OR Search
- Inference vs Search
- Hybrid of search and Inference
- Experiments

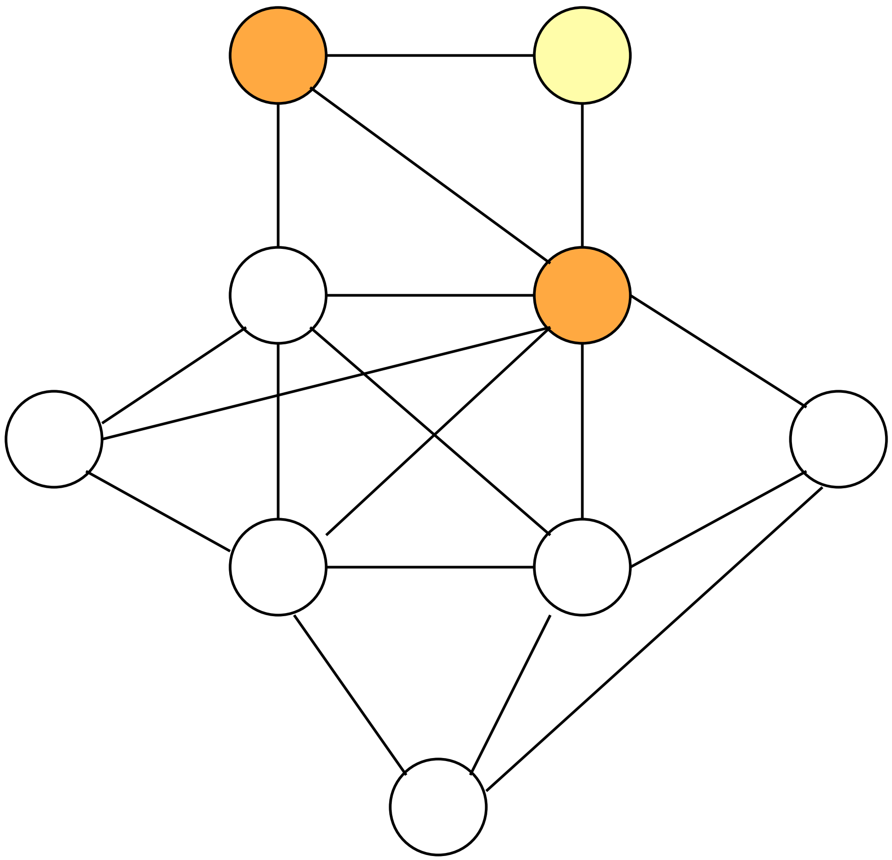
# Outline

- Graphical models: reasoning principles
- Inference
- Search;  via AND/OR Search
- Hybrid of search and Inference
  - Adaptive AND/OR search
  - Cycle-cutset + elimination
  - Interleaving elimination and conditioning
- Experiments

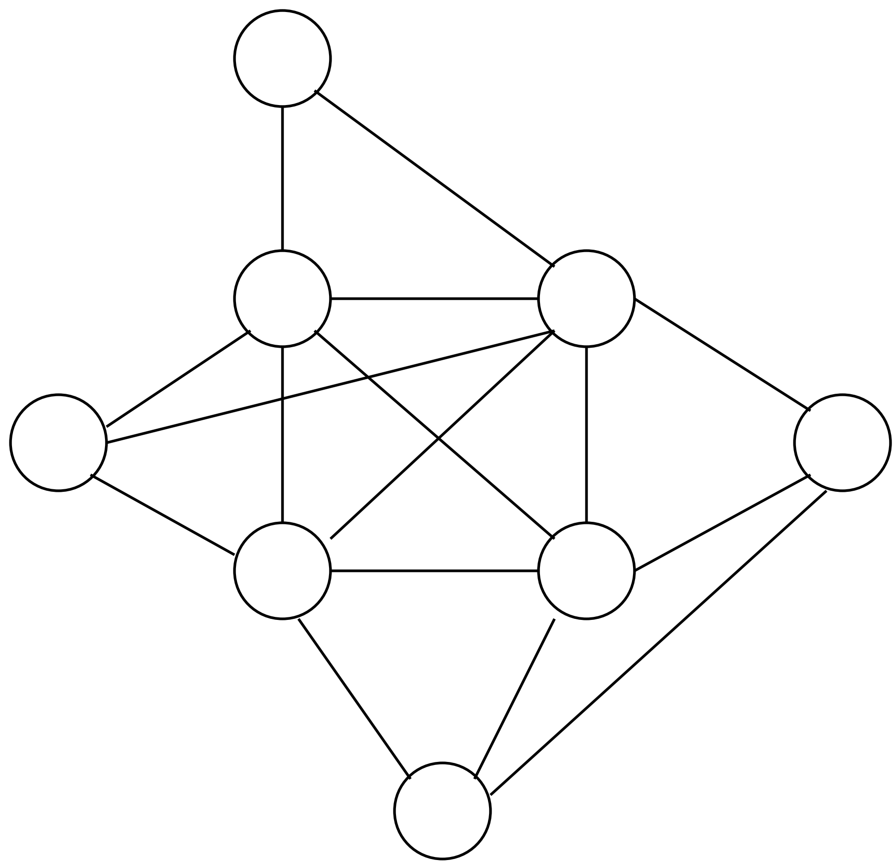
# Interleaving Cond and Elim



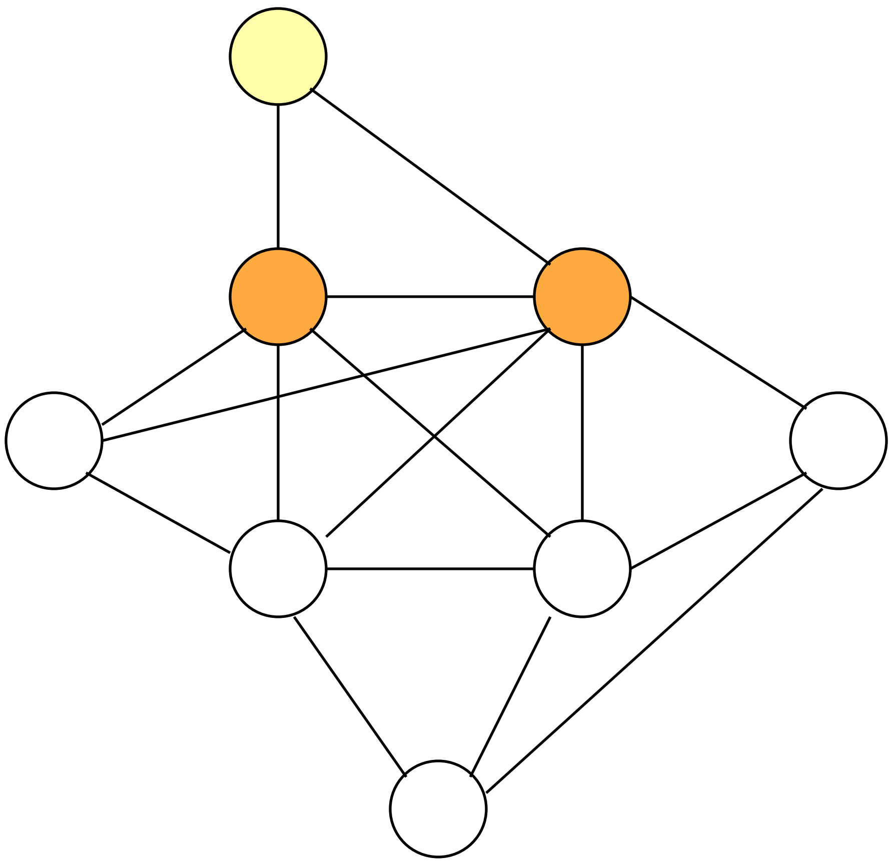
# Interleaving Cond and Elim



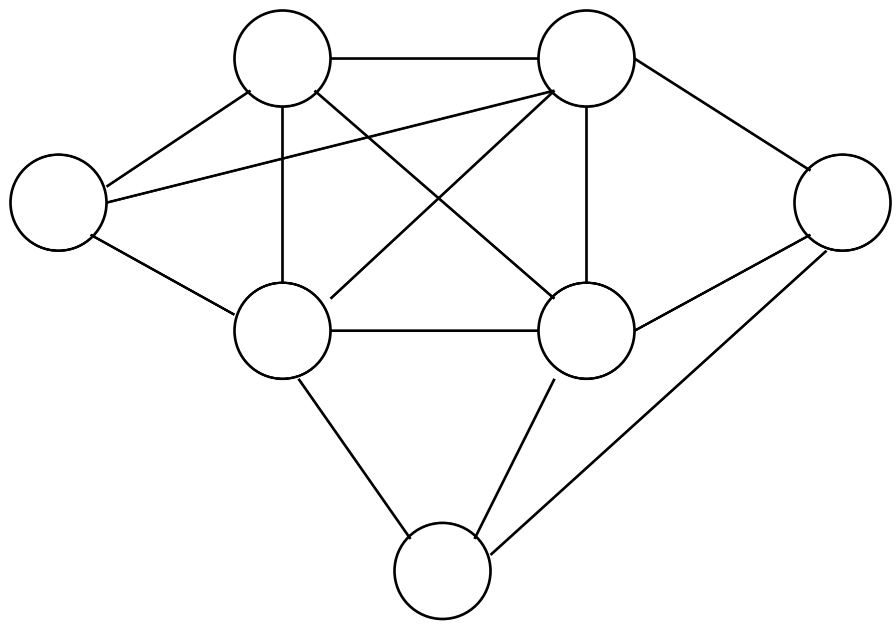
# Interleaving Cond and Elim



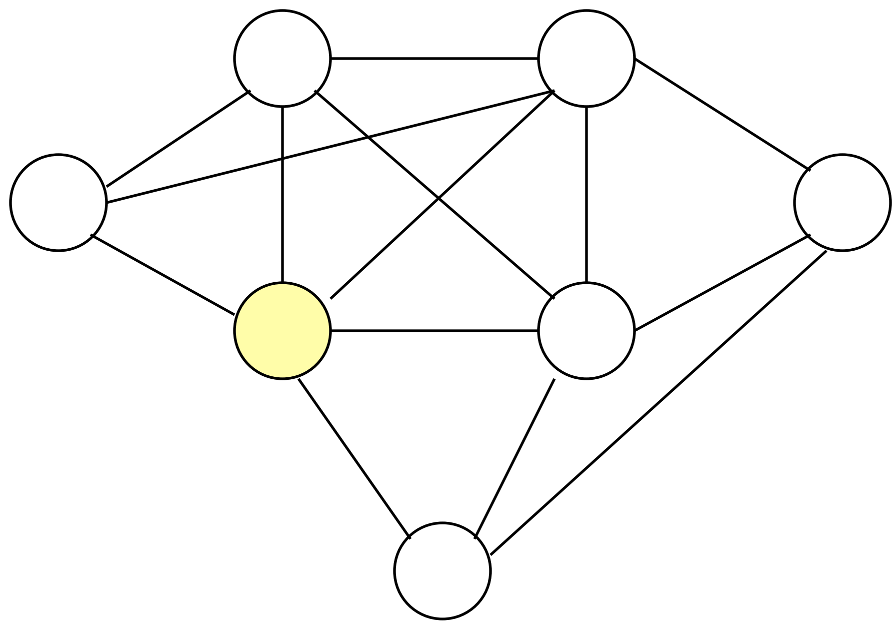
# Interleaving Cond and Elim



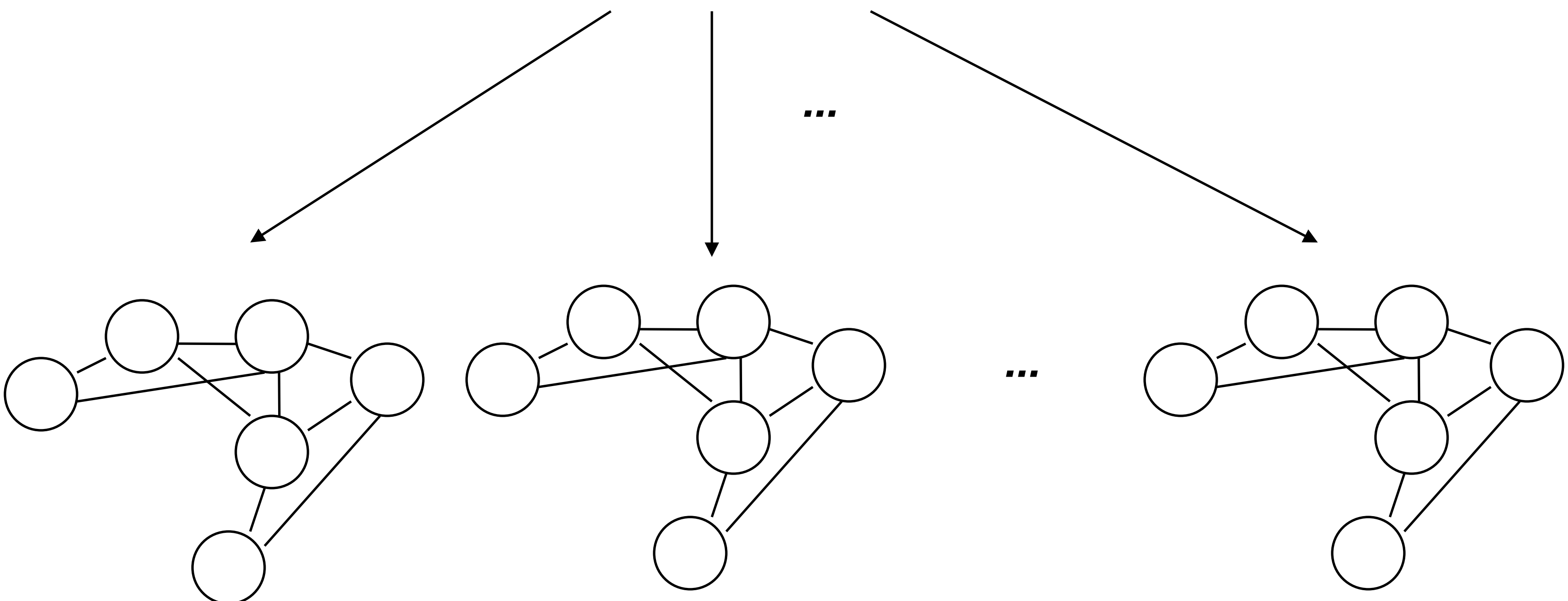
# Interleaving Cond and Elim



# Interleaving Cond and Elim

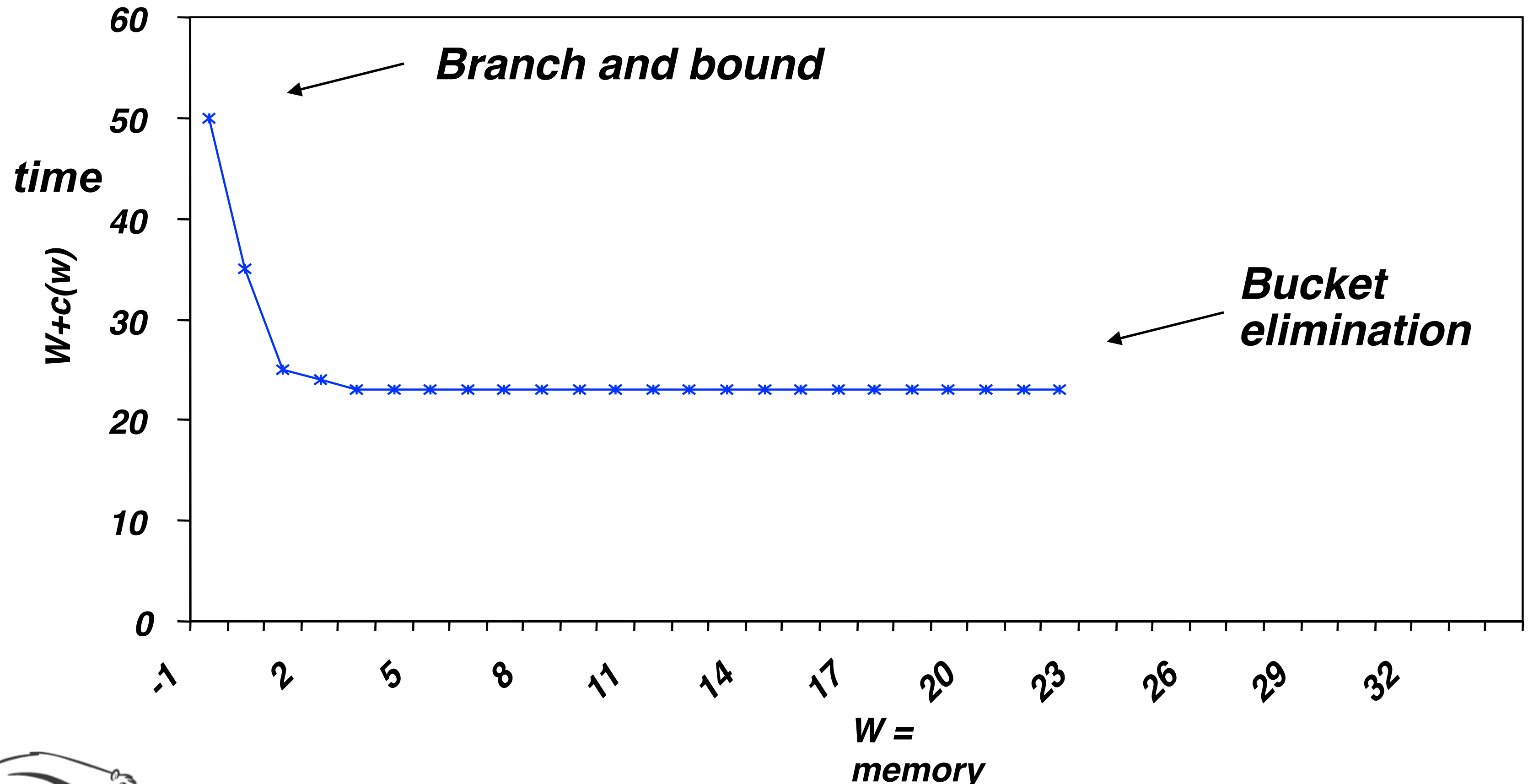


# Interleaving Cond and Elim



# Time vs space

- *Random Graphs (50 nodes, 200 edges, average degree 8, w*≈23)*



# Searching AND/OR Graphs

- AO(i): searches depth-first, cache i-context
  - i = the max size of a cache table (i.e. number of variables in a context)

***i=0*** ——— ***i*** ——— ***i=w****

***Space: O(n)***
***Time: O(exp(w* log n))***

***Space: O(exp(i) )***
***Time: O(exp(mi+i )***
***mi - AO depth of w-cutset***

***Space: O(exp w*)***
***Time: O(exp w*)***

# Outline

- Graphical models: the primary reasoning principles
- Inference
- AND/OR Search Trees and Graphs
- Lower Bounding heuristics for search
- AND/OR Branch-and-Bound Search
- Experiments and competitions

# UAI 2010 evaluation, 2008, 2006

***We are first in Pascal 2012, so far…***
***Please join***

- UAI-2010 (first on pr and MAR, 3rd on mpe)
- UAI-2006, 2008, Competitions
  - PE, MAR, MPE tasks
- CP-2006 Competition
  - WCSP task
- Daoopt: UCI Irvine

  **Summary**: "daoopt" and "daoopt.anytime" are based on AND/OR branch and bound graph search, with mini bucket heuristics and LDS (Limited Discrepancy Search) initialization.

  **Web-site**: http://graphmod.ics.uci.edu

# Software

- AND/OR search algorithms
- Bucket-tree elimination
- Generalized belief propagation
- Samplesearch sampling

are available at:

- http://graphmod.ics.uci.edu/group/Software

BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

# Thank you!


***We are first in Pascal challenge 2012 (Globerson, Elidan),***
***so far…Please join***

**For publication see:**

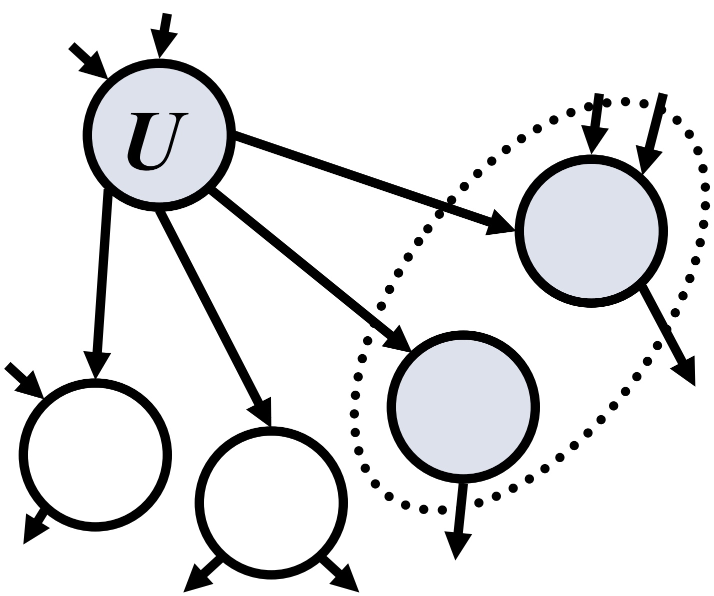
**http://www.ics.uci.edu/~dechter/publications.html**

***Kalev Kask***
***Irina Rish***
***Bozhena Bidyuk***
***Robert Mateescu***
***Radu Marinescu***
***Vibhav Gogate***
***Emma Rollon***
***Natalia Flerova***

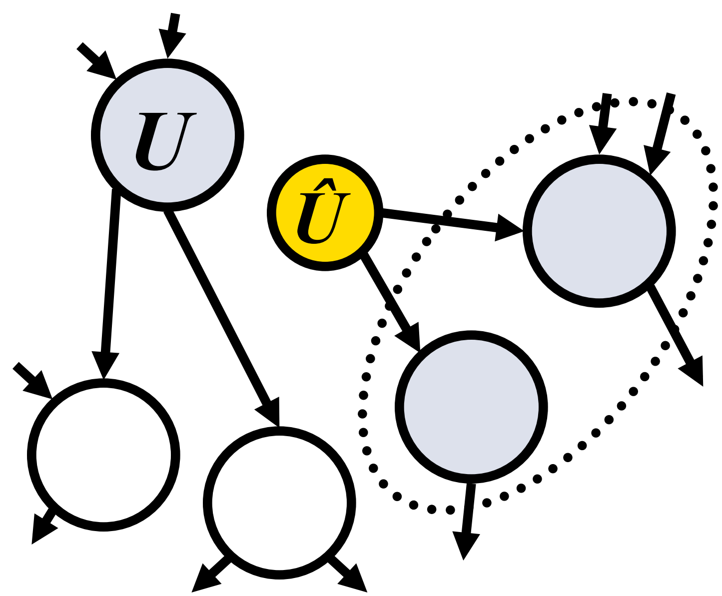
# Semantics of Mini-Bucket: Splitting a Node

***Variables in different buckets are renamed and duplicated (Kask et. al., 2001), (Geffner et. al., 2007), (Choi, Chavira, Darwiche , 2007)***

*Before Splitting:*
*Network N*




*After Splitting:*
*Network N'*



# UAI 2010 evaluation, 2008, 2006

***We are first in Pascal 2012, so far… Please join***

- Toulbar2: INRA
  **Summary**: Toulbar2 is an open source exact anytime Weighted CSP solver using Branch and Bound and soft local consistency
  **Team members**: S. de Givry, D. Allouche, A. Favier, T. Schiex
  **Additional contributors**: M. Sanchez, S. Bouveret, H. Fargier, F. Heras, P. Jegou, J. Larrosa, K. L. Leung, S. N'diaye, E. Rollon, C. Terrioux, G. Verfaillie, M. Zytnicki
  **Contact person**: Thomas Schiex, Thomas.Schiex@toulouse.inra.fr
  Detailed description

- Daoopt: UCI Irvine
  **Summary**: "daoopt" and "daoopt.anytime" are based on AND/OR branch and bound graph search, with mini bucket heuristics and LDS (Limited Discrepancy Search) initialization.
  **Team members**: Lars Otten, Rina Dechter
  **Additional Contributor:** Radu Marinescu
  **Contact person**: Lars Otten, lotten@ics.uci.edu
  Detailed description
  **Web-site**: http://graphmod.ics.uci.edu

***3rd in all 3 categories After Toolbar, Joris***