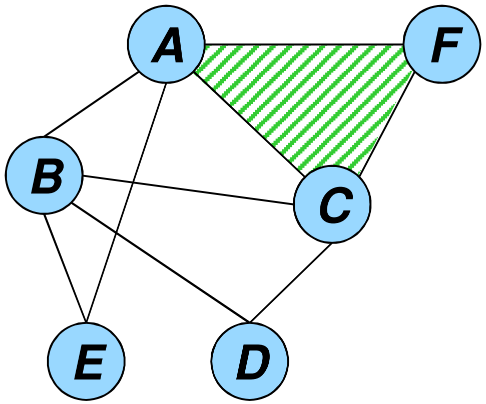
Inference and Search for Graphical Models

Rina Dechter

Bren school of ICS, University of California, Irvine

Joint work with Radu Marinescu, Robert Mateescu, Kalev Kask, Irina Rish, Alex Ihler



Los Alamos

Sample Applications for Graphical Models

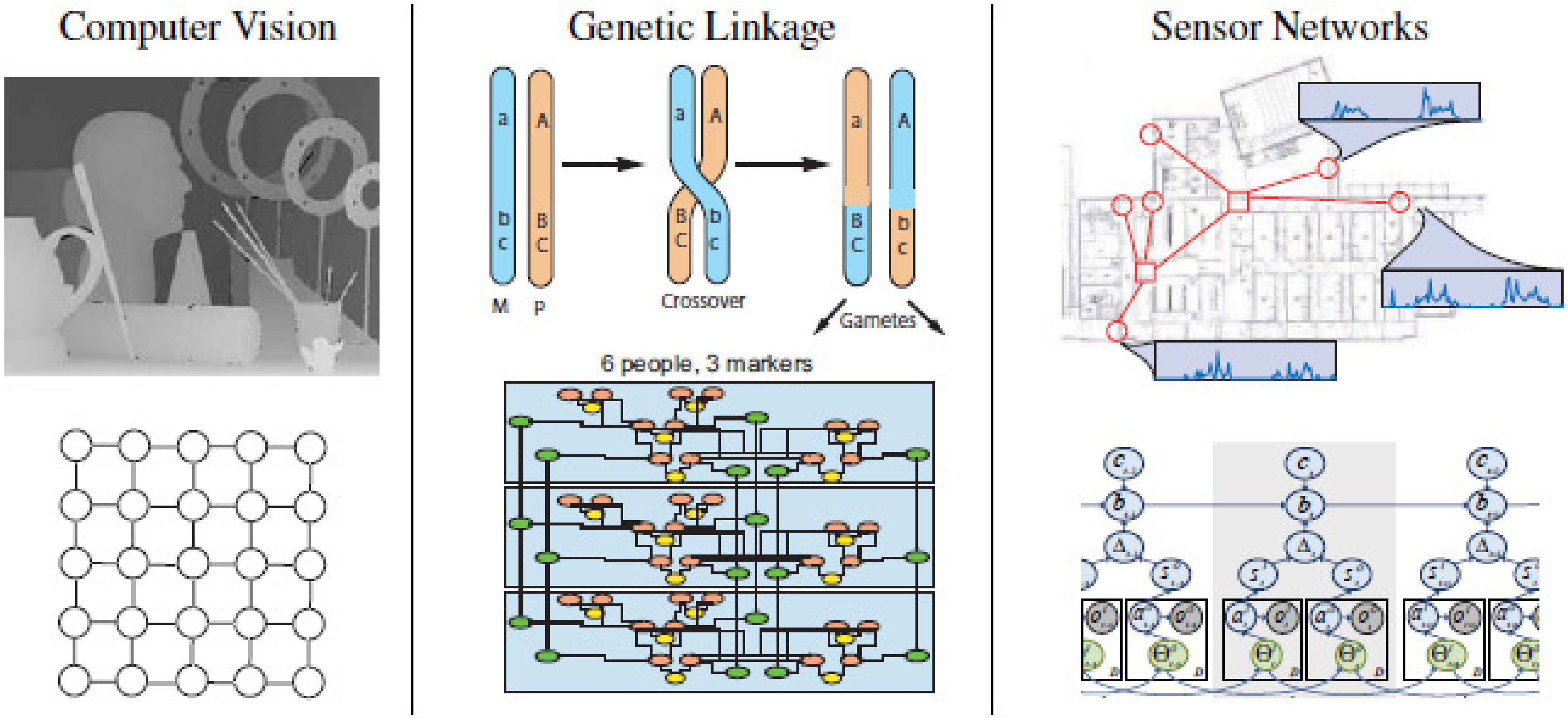


Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal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

Sample Applications for Graphical Models

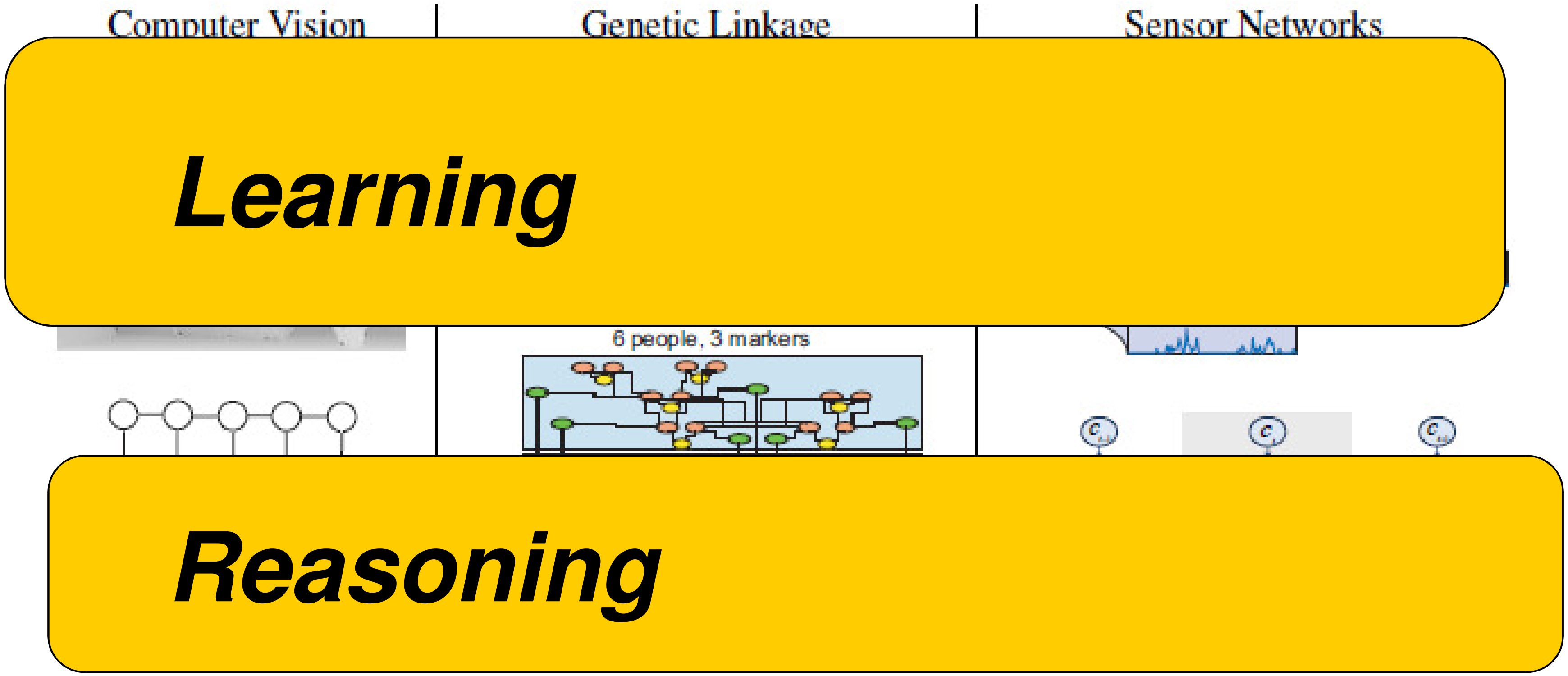


Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal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

Outline

- What are graphical models?
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference
- Anytime AND/OR branch and bound
- Experiments, Competitions

and conclusions

Exact algorithms

Approximations

Back to exact: Anytime schemes

Constraint Networks

Map coloring

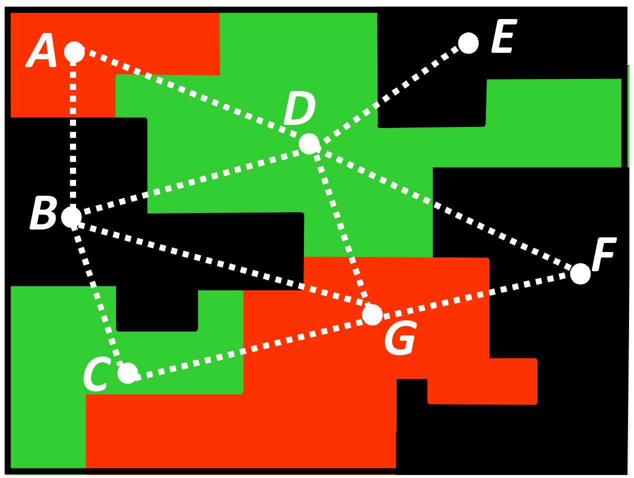
Combination = join
Marginalization = projection

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

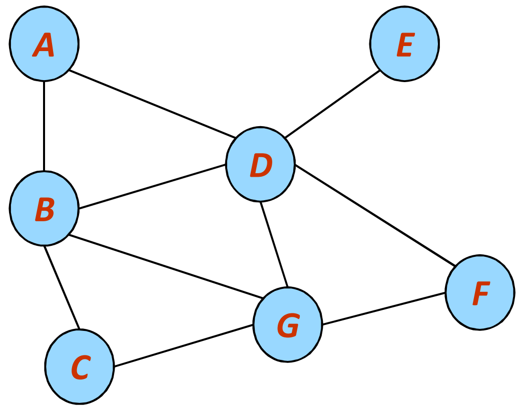
Values: *colors (red green blue)*

Constraints: $A \neq B$, $A \neq D$, $D \neq E$, …

A	*B*
red	*green*
red	*yellow*
green	*red*
green	*yellow*
yellow	*green*
yellow	*red*

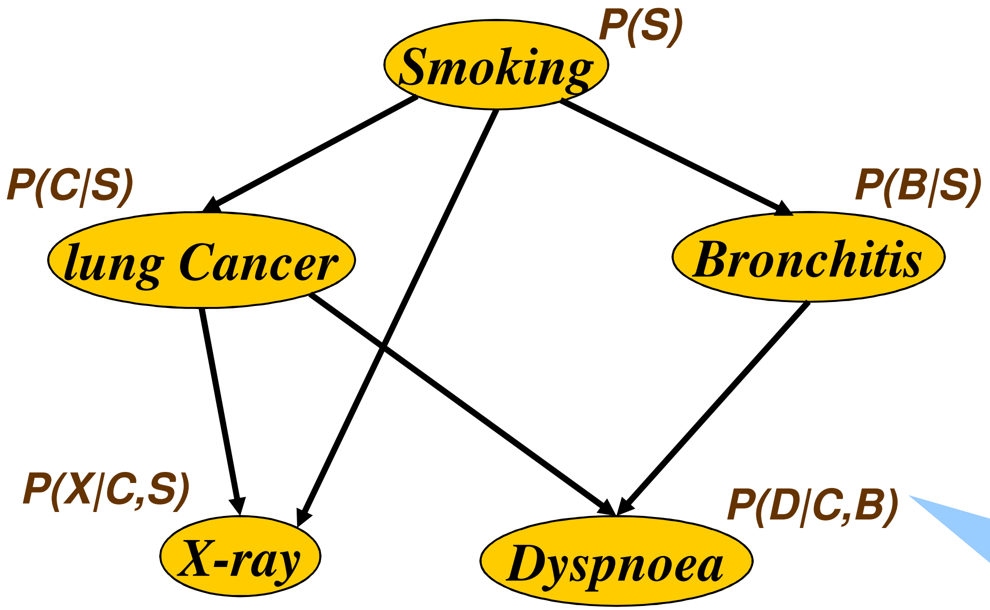


Constraint graph



Queries: Find one solution, all solutions, counting

Bayesian Networks (Pearl 1988)



$$BN = (G, \Theta)$$

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

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

$$P(x_1 \ldots x_n) = \prod_i p(x_i \mid pa(x_i))$$

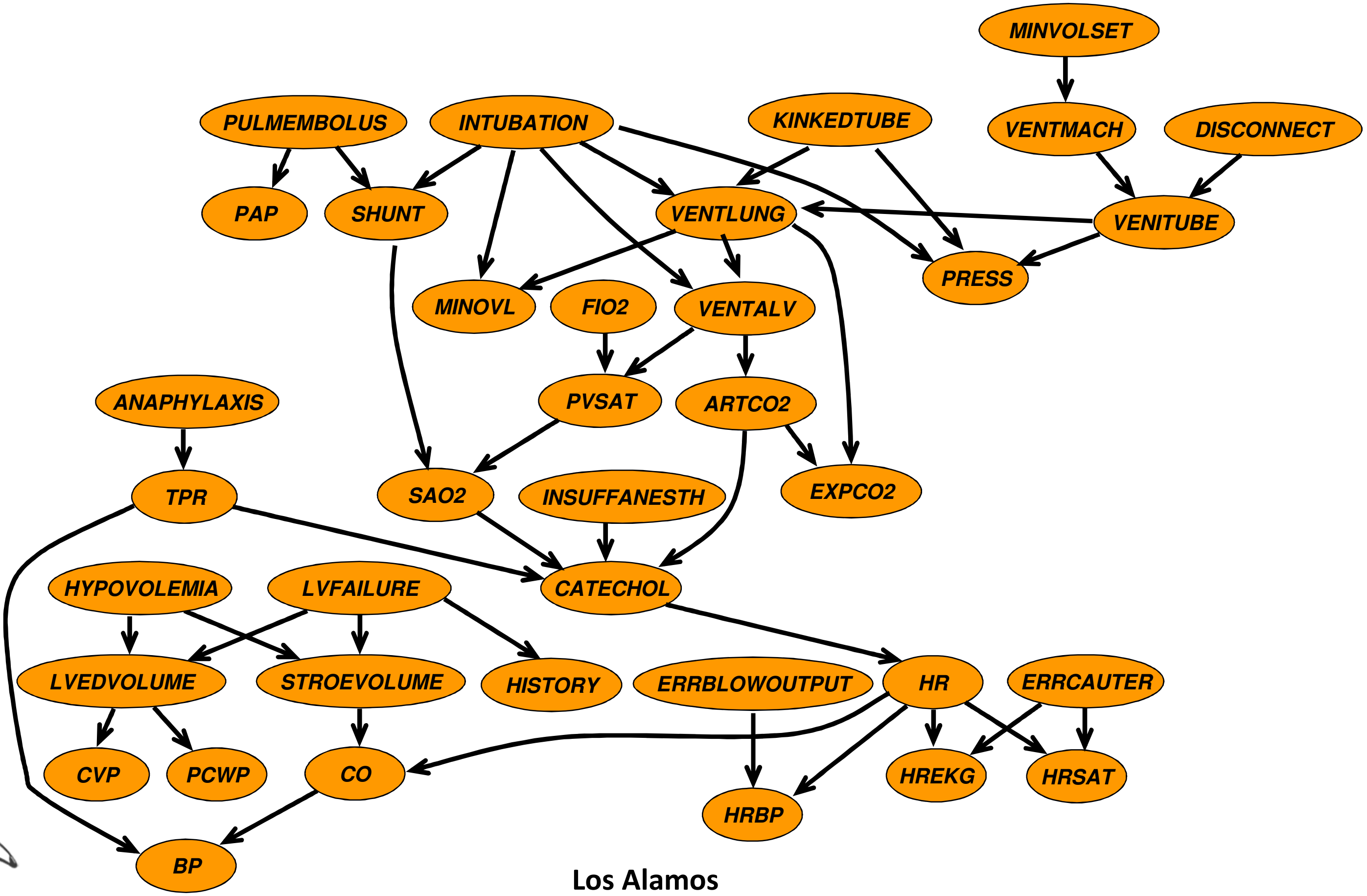
$$P(e) = \sum_{X-E} \prod_i p(x_i \mid pa(x_i))$$

$$mpe = \max_x \prod_i p(x_i \mid pa(x_i))$$

Combination: Product
Marginalization: sum/max

Monitoring Intensive-Care Patients

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



Los Alamos

Constraint Optimization Problems
for Graphical Models

A *finite* COP is a triple $R = \langle X, D, F \rangle$ where:

$X = \{X_1, ..., X_n\}$ - variables

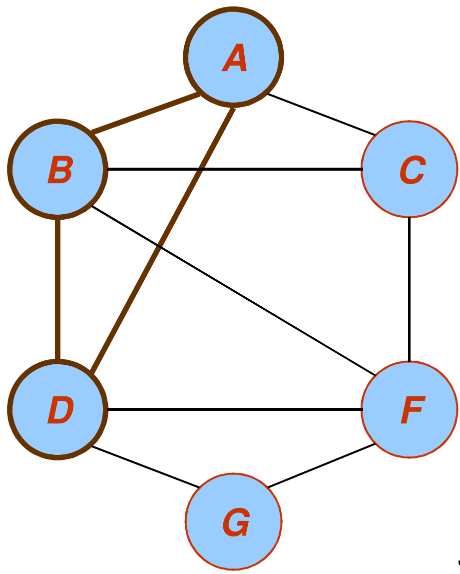
$D = \{D_1, ..., D_n\}$ - domains

$F = \{f_1, ..., f_m\}$ - cost functions

f(A,B,D) has scope {A,B,D}

A	B	D	Cost
1	2	3	3
1	3	2	2
2	1	3	∞
2	3	1	0
3	1	2	5
3	2	1	0

Primal graph =
Variables --> nodes
Functions, Constraints -→ arcs



F(a,b,c,d,f,g)= f_1(a,b,d)+f_2(d,f,g)+f_3(b,c,f)

Global Cost Function

$$F(X) = \sum_{i=1}^{m} f_i(X)$$

Graphical Models

- A graphical model M = (X,D,F,⊗)
  - X = $\{X_1,\ldots,X_n\}$ variables
  - D = $\{D_1,\ldots,D_n\}$ domains
  - F = $\{f_1,\ldots,f_t\}$ functions over $\{S_1,\ldots,S_t\}$
  - Global function ⊗ Combine

$$F(x) = \otimes_f f(x_f)$$

$$G(y) = \Downarrow_{x-y} F(x)$$

$$G(y) = \Downarrow_{x-y} \otimes_f f(x_f)$$

f(x)

- Queries:
  - **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**: $\min_X \Sigma_j F_j$

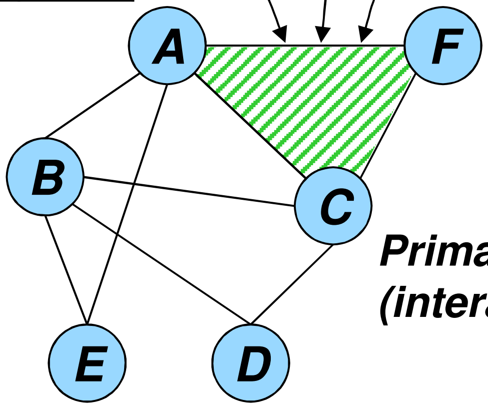
Conditional Probability Table (CPT)

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)

When combine and marginalize obey Some properties they can be solved By the same algorithms

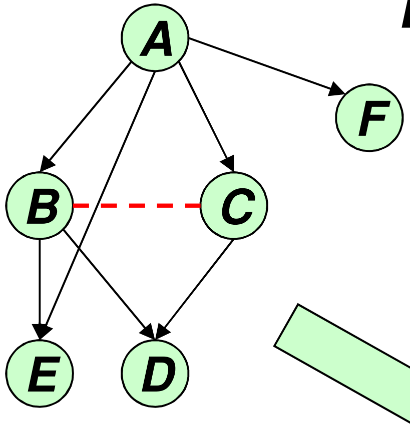
(Bistareli , Rossi and Montanari, 1995, Shenoy , Shafer, 1990, Kasket. Al., 2005 et. Al.)

Mixed Networks

(Mateescu and Dechter, 2004)

Examples: NLP, Linkgage, Software verification, probabilistic languages

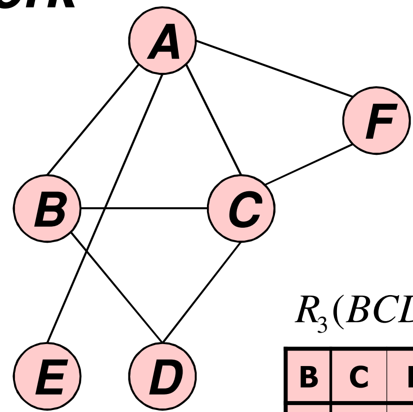
Belief Network



$P(D|B,C)$

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

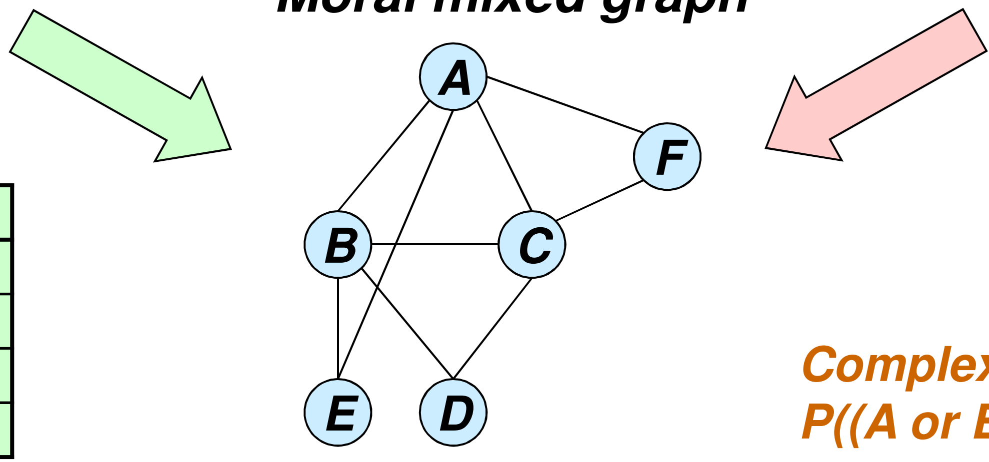
Constraint Network



$R_3(BCD)$

B	C	D
0	0	1
0	1	0
1	1	0

Moral mixed graph



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

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

Graphical Models

- A graphical model M = (X,D,F,$\otimes$)

$$F(x) = \otimes_f f(x_f)$$

$$G(y) = \Downarrow_{x-y} F(x)$$

$$G(y) = \Downarrow_{x-y} \otimes_f f(x_f)$$

Reasoning Queries:
Satisfaction
Optimization
Counting
Hybrids: optimization-counting
Hybrid: Minimize expected utility

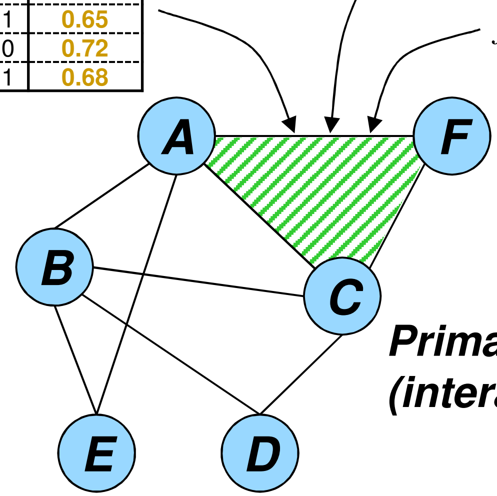
Conditional Probability Table (CPT)

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***

Reasoning Algorithms

- **By Inference (thinking)**
  - Transform into a single, equivalent (tree) of sub-problems
  - Atomic operation: inference
    - $f_1(X) \otimes f_2\,(Y) \longrightarrow f_3(X,Y)$
- **By Conditioning (guessing)**
  - Transform into many (tree-like) sub-problems.
  - Atomic operation: assign a value $f_1(X=5)$

Outline

- What are graphical models?
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference
- Anytime AND/OR branch and bound
- Experiments, Competitions

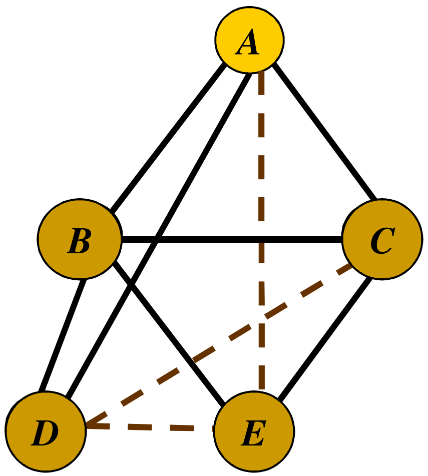
and conclusions

Exact algorithms

Approximations

Back to exact:
Anytime schemes

Query 1: Belief updating: P(X|evidence)=?



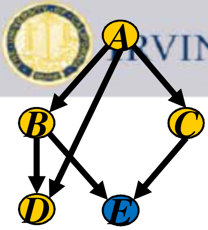
"Moral" graph

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

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

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

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

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

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

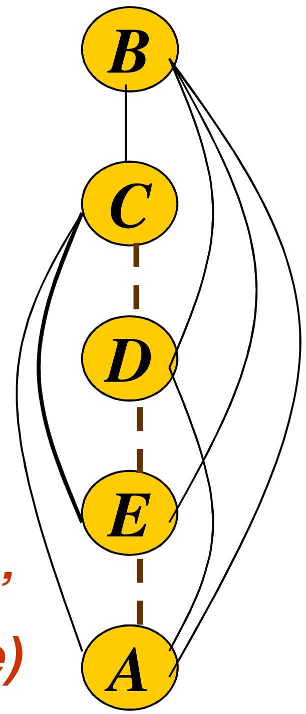
bucket E: $e=0$ $\lambda^D(a, e)$

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

$P(e=0)$

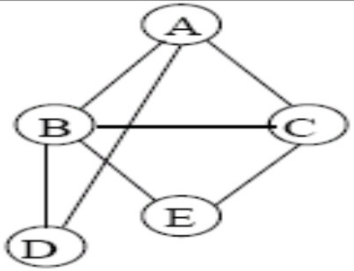
$P(a|e=0)$

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



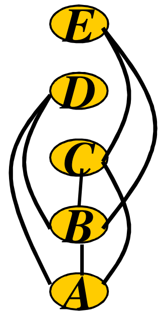
$$P(a|e=0) = \frac{P(a,e=0)}{P(e=0)}$$

Bucket Elimination and Induced Width



Ordering: a, b, c, d, e

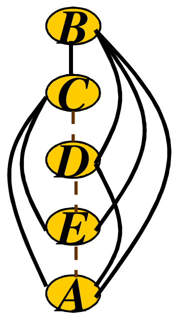
$bucket(E) = \quad P(e|b,c),\ e = 0$
$bucket(D) = \quad P(d|a,b)$
$bucket(C) = \quad P(c|a) \quad || \ P(e = 0|b,c)$
$bucket(B) = \quad P(b|a) \quad || \quad \lambda_D(a,b), \lambda_C(b,c)$
$bucket(A) = \quad P(a) \quad || \quad \lambda_B(a)$



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

Query 2: Finding MPE by Bucket Elimination

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

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

$$\max_X \prod$$

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

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

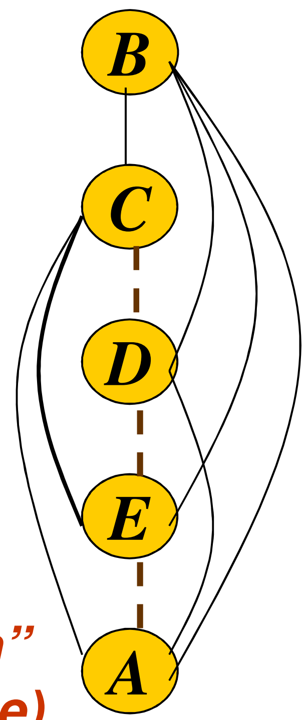
bucket D: $h^C(a,d,e)$

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

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

OPT

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')

Generating the MPE-tuple

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

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

Time and space exponential in the induced-width / treewidth

$$O(nk^{w^*+1})$$

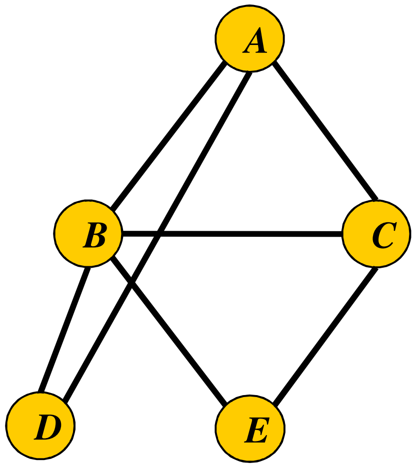
1. $a' = \arg\max_a P(a) \cdot h^E(a)$

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

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

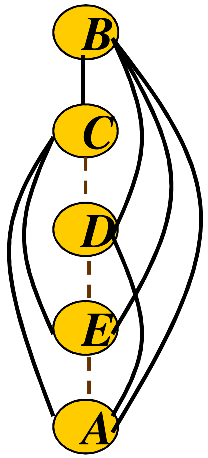
The Induced-width/treewidth

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

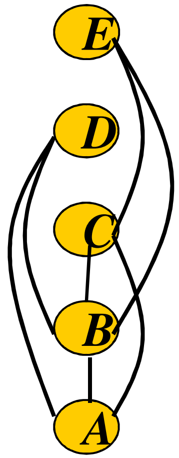


"Moral" graph

The effect of the ordering:



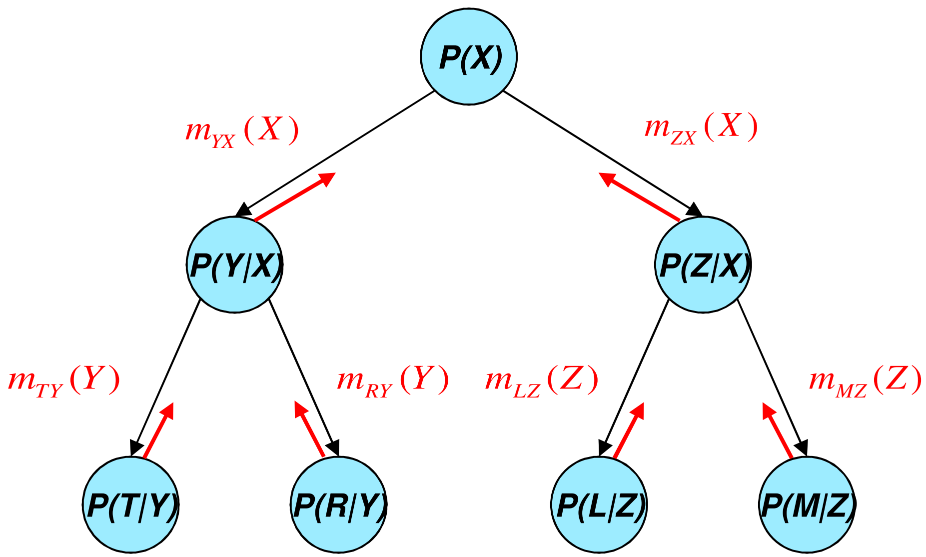
$w^*(d_1) = 4$



$w^*(d_2) = 2$

Complexity of Bucket-elimination

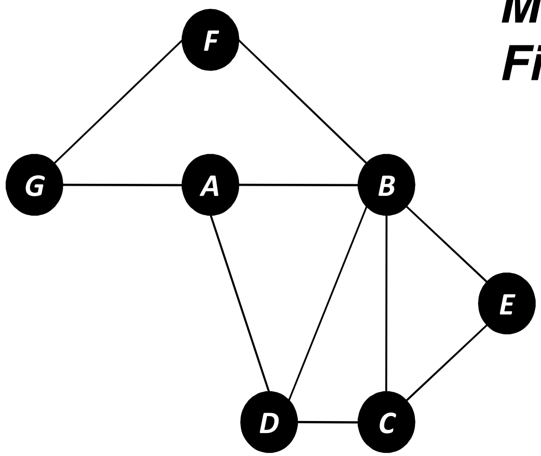
- Theorem: Bucket-elimination is O(r•k^{w*+1}) time and O(nk^{w*}) space.
- When w=1 then w*=1 → trees
- When we have a tree of functions w=w* .

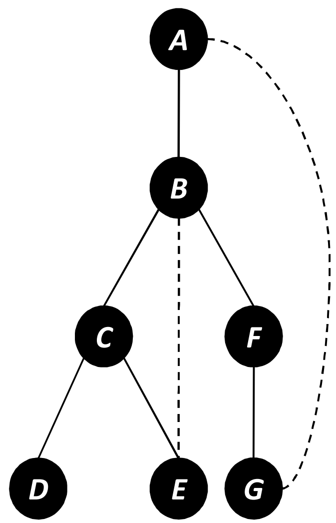


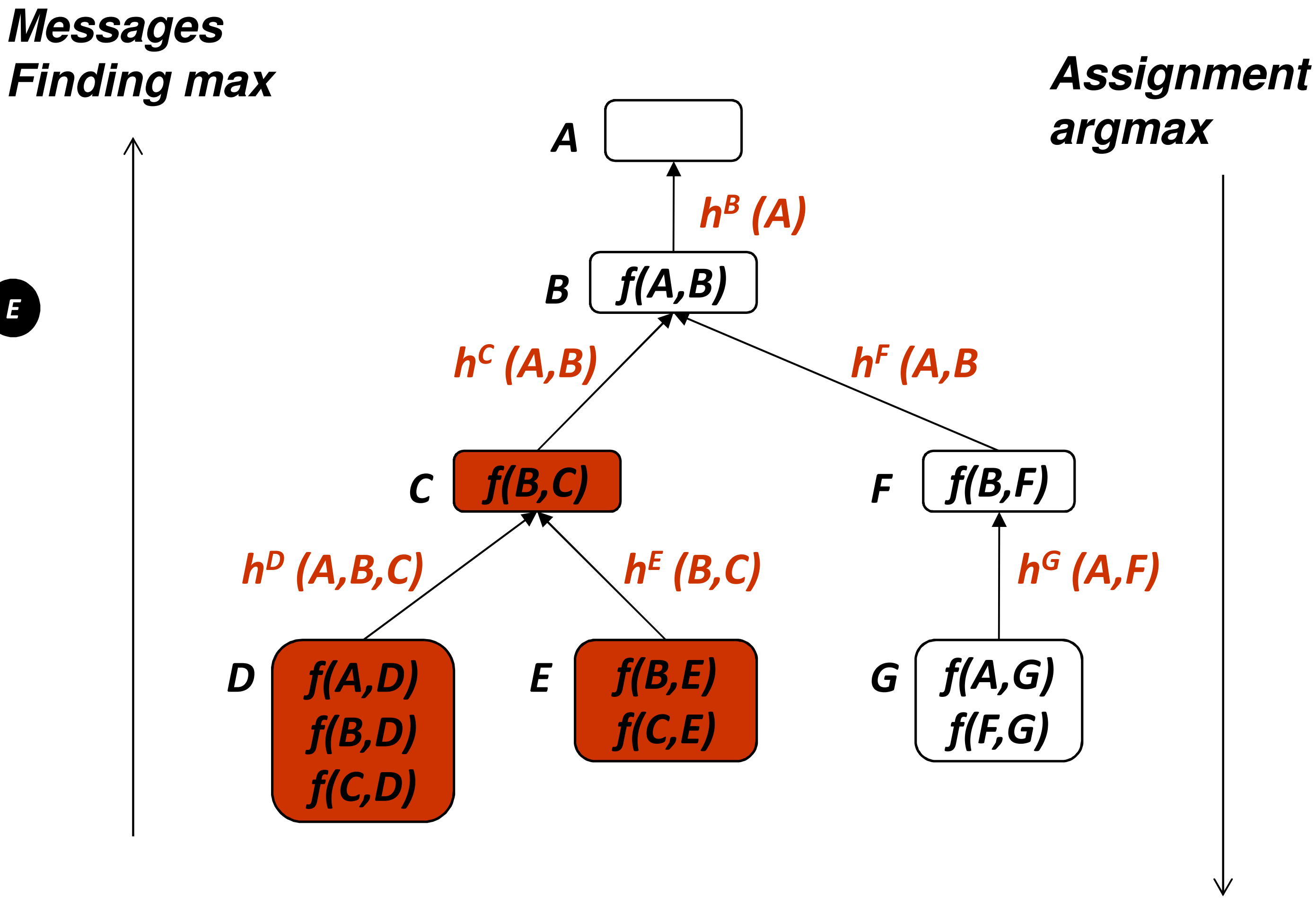
bucket-elimination Sends messages From leaves to root

Bucket-Tree Elimination

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

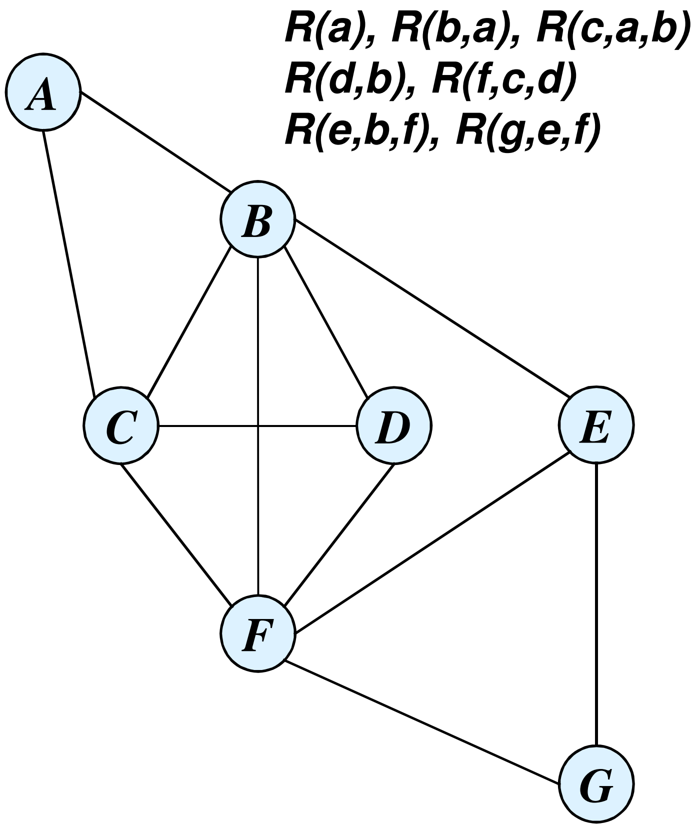


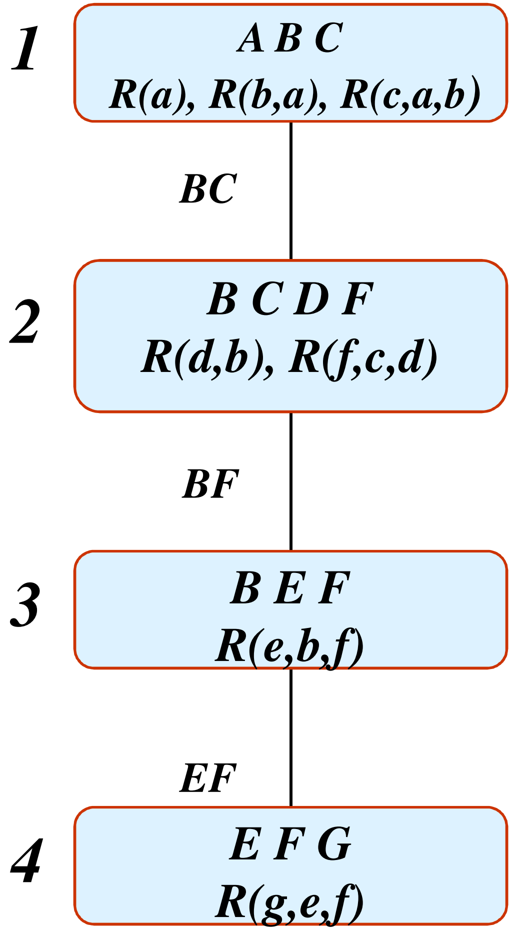




Ordering: (A, B, C, D, E, F, G)

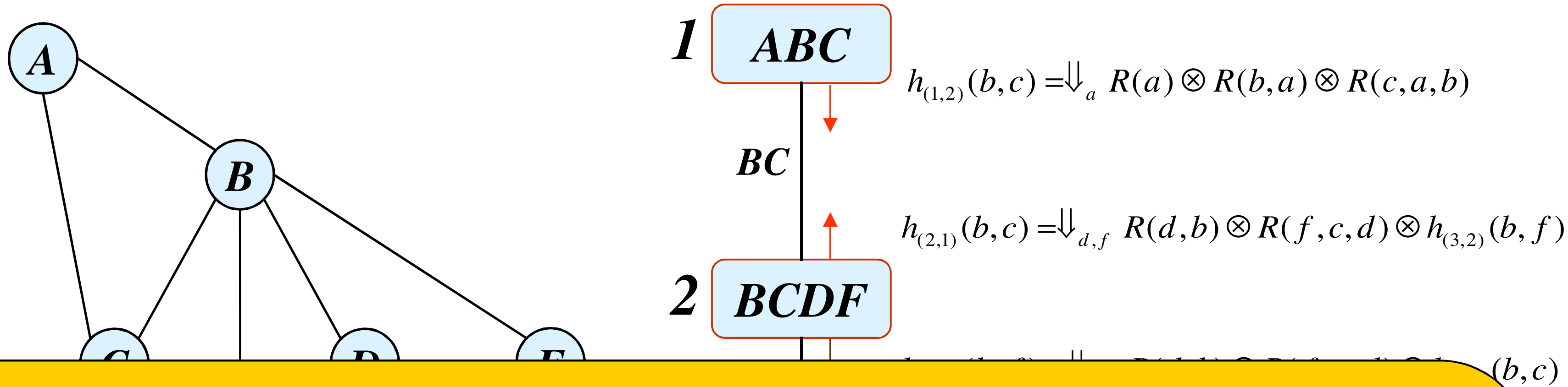
Tree Decomposition





- Each function in a cluster
- Satisfy running intersection property
- Then infer a function in a cluster and send to neighbors

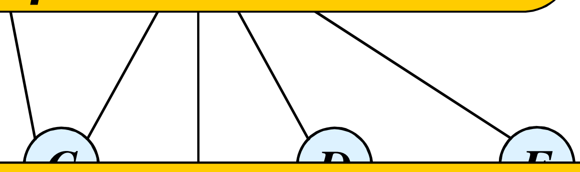
Junction-tree Clustering

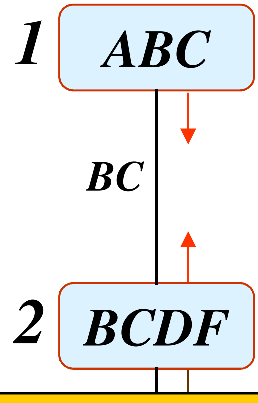


Time exponential in the induced-width / treewidth O(nk^{w*+1}), space exponential in the separator width along ordering d

Junction-tree Clustering

Each cluster is perfectly explicit.
Inference re-parameterizes the graphical model



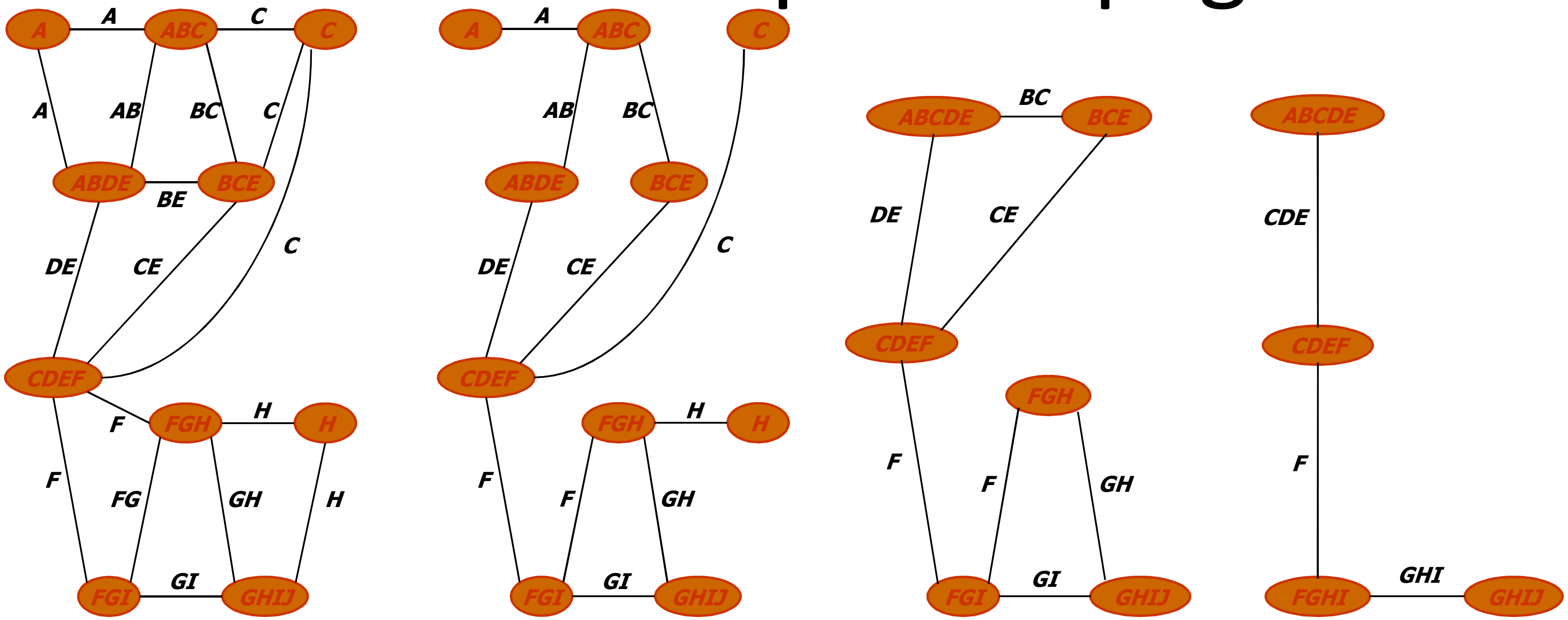


$$h_{(1,2)}(b,c) = \Downarrow_a R(a) \otimes R(b,a) \otimes R(c,a,b)$$

$$h_{(2,1)}(b,c) = \Downarrow_{d,f} R(d,b) \otimes R(f,c,d) \otimes h_{(3,2)}(b,f)$$

Time exponential in the induced-width / treewidth O(nk^{w*+1}), space exponential in the separator width along ordering d

Iterative Join-Graphs Propagation



more accuracy

less complexity

(Kask, Mateescu,Gogate,Dechter, 2002,2009)

Summary of Inference

- Inference is time and space exponential in w*(d)
- Finding the best ordering is NP-complete but good orderings can be achieved
- Inference is a **compilation** scheme; a re-parameterization scheme: each cluster possesses local information to answer whatever query without looking outside.
- Main limitation: memory complexity

Outline

- What are graphical models?
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference
- Anytime AND/OR branch and bound
- Experiments, Competitions

and conclusions

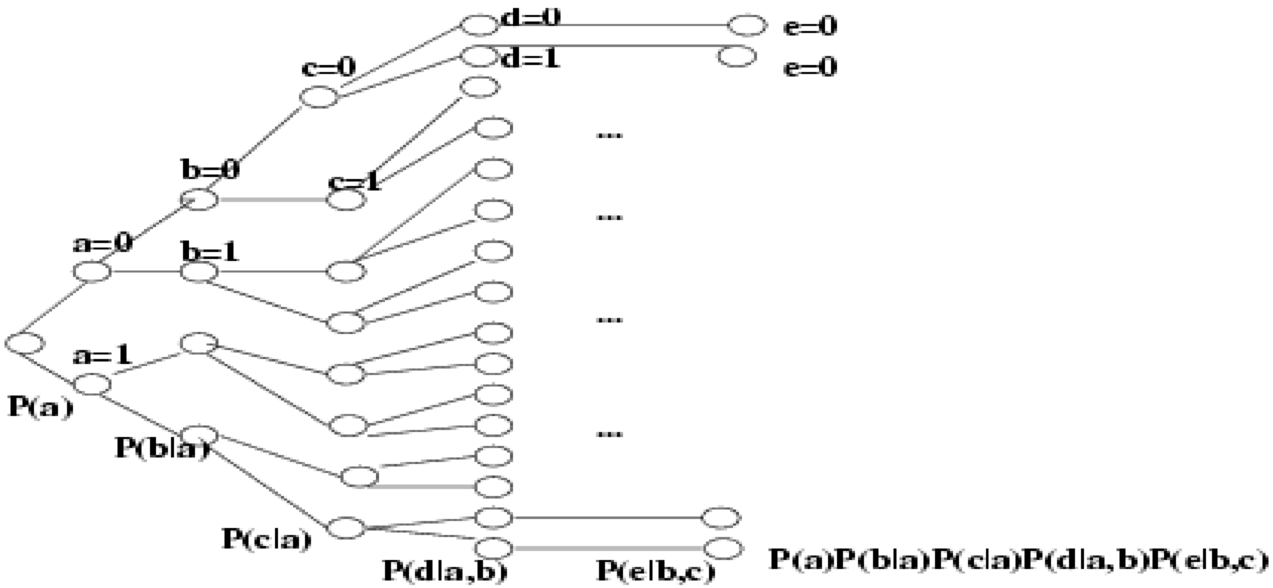
Exact algorithms

Approximations

Back to exact: Anytime schemes

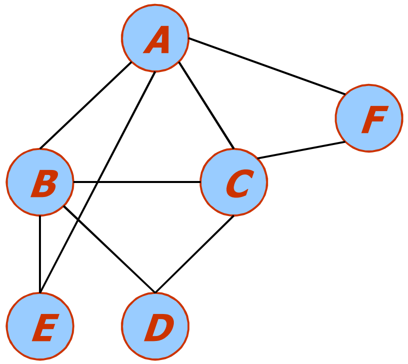
Enumeration of the the probability tree

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

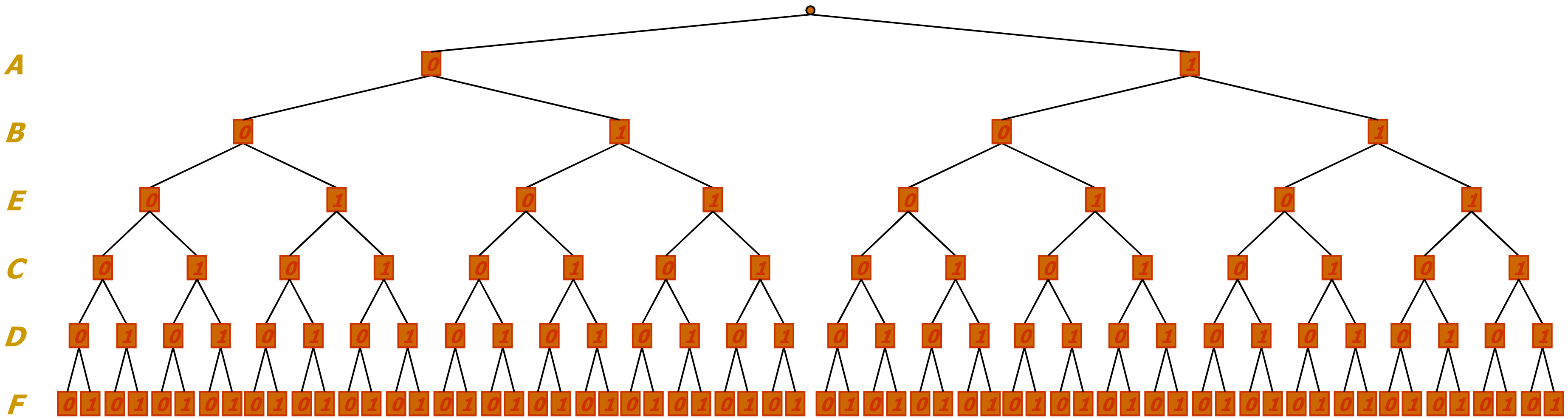


Complexity of searching the probability tree: exponential time. But can use linear space

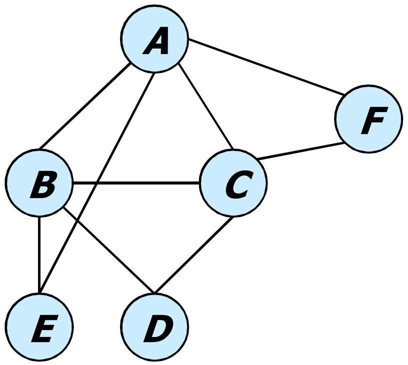
Classic OR Search Space



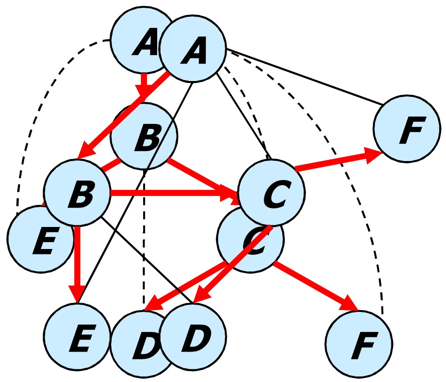
Ordering: A B E C D F



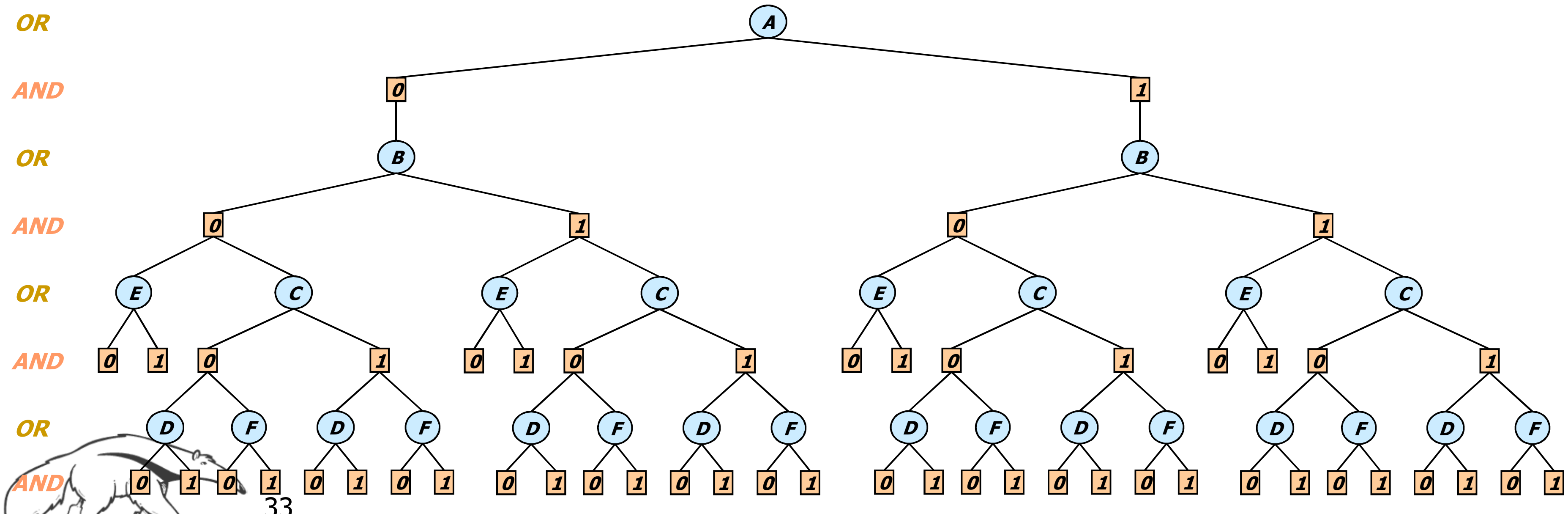
AND/OR Search Space



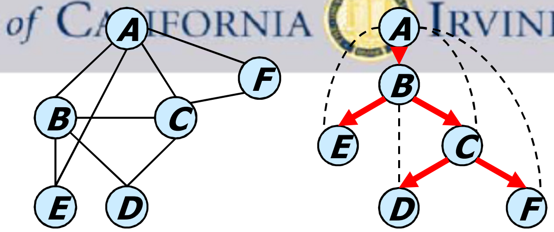
Primal graph

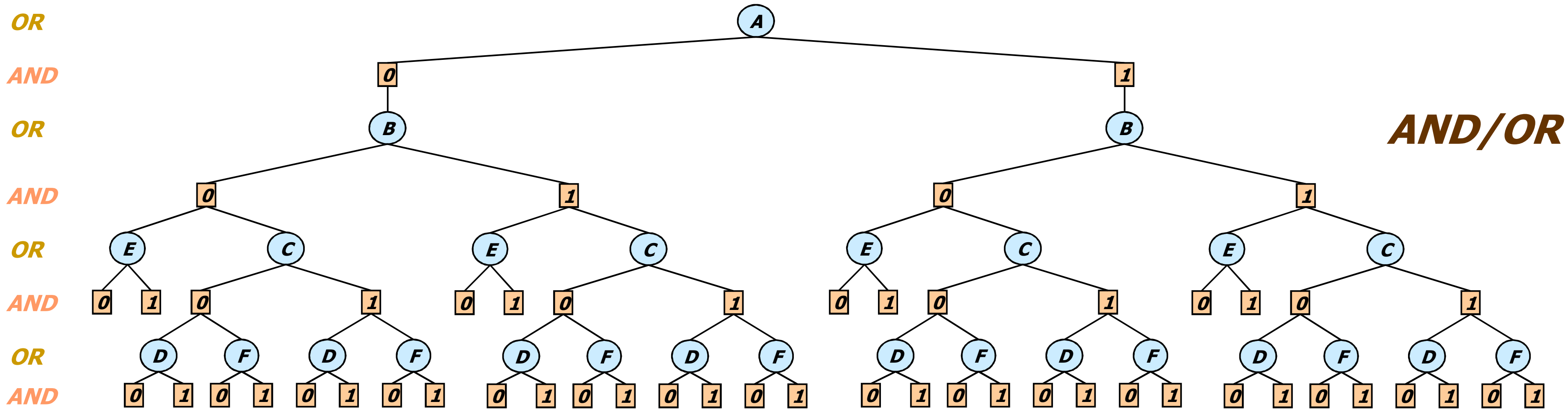


DFS tree

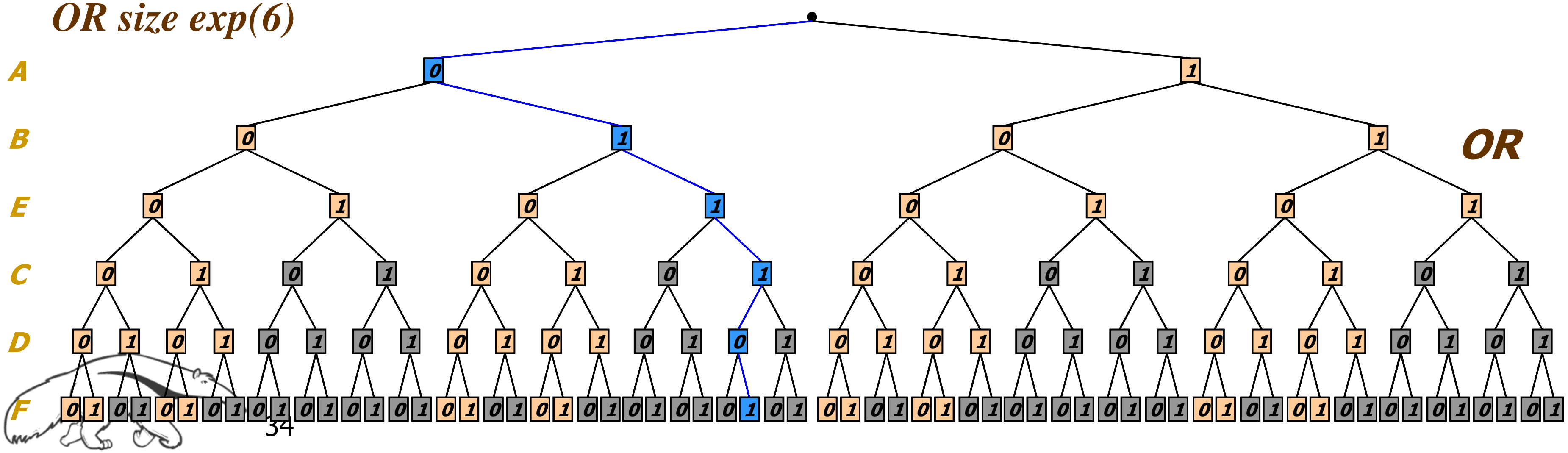


AND/OR vs. OR





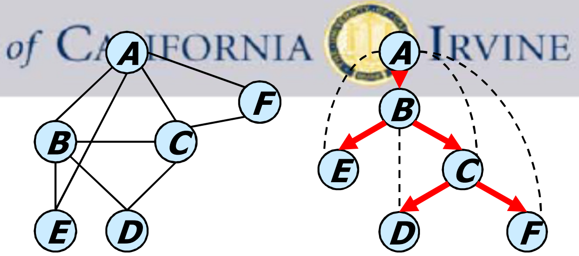
AND/OR size: exp(4),
OR size exp(6)

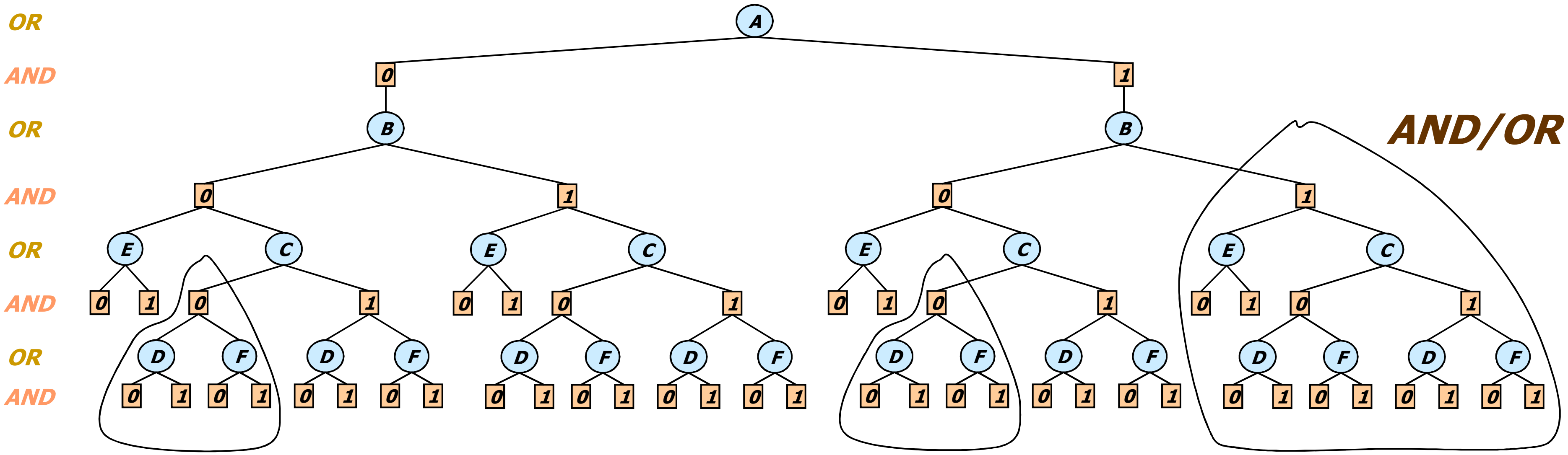


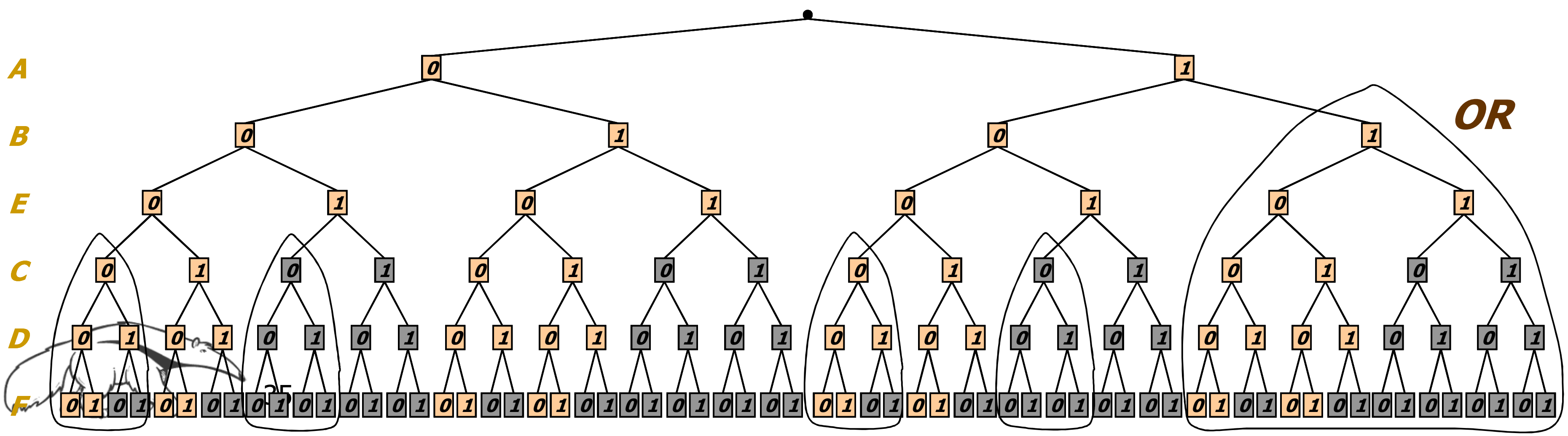
AND/OR vs. OR

with Constraints

No-goods
(A=1,B=1)
(B=0,C=0)



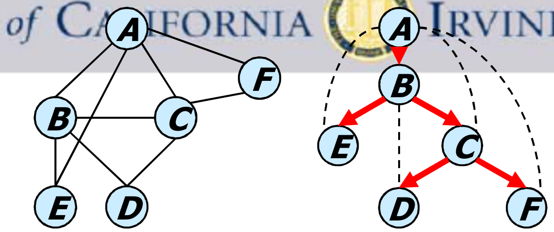


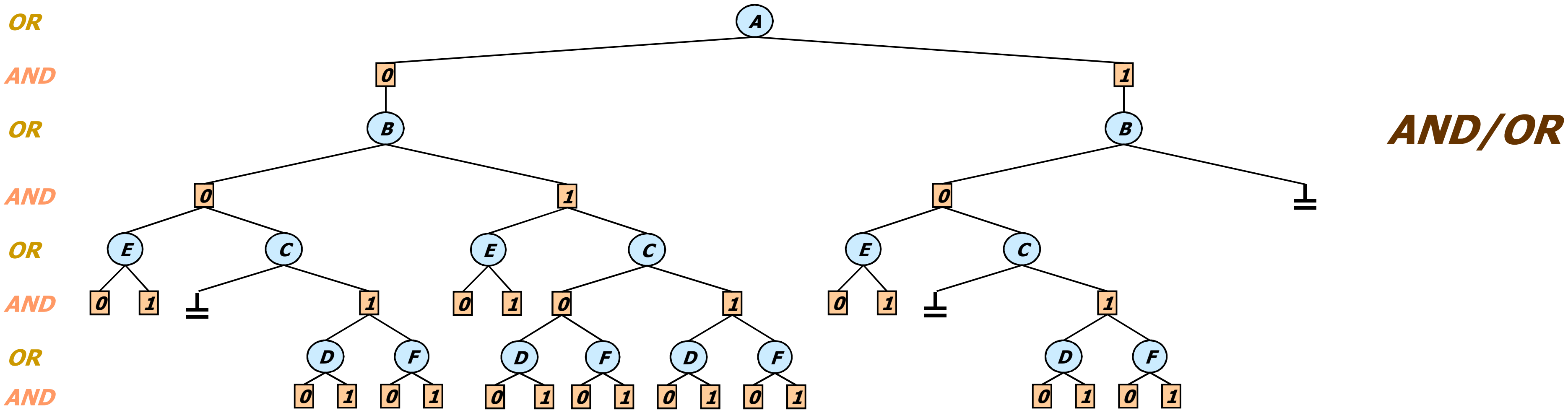


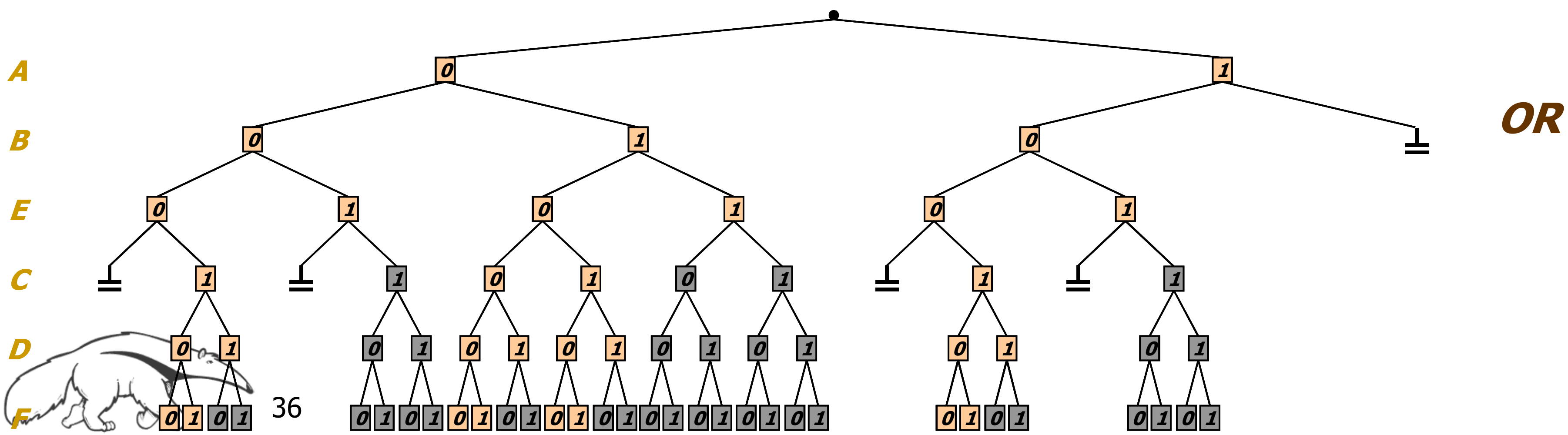
AND/OR vs. OR

with Constraints

No-goods
(A=1,B=1)
(B=0,C=0)

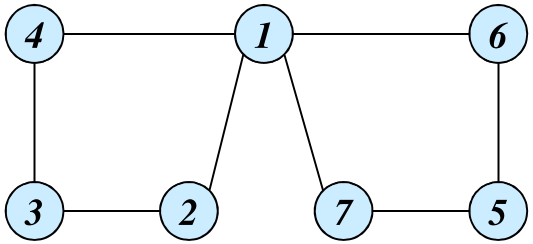






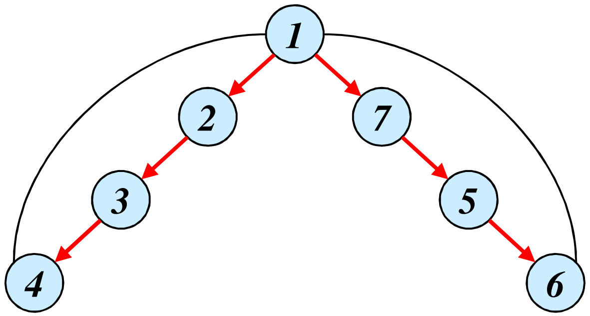
Pseudo-Trees

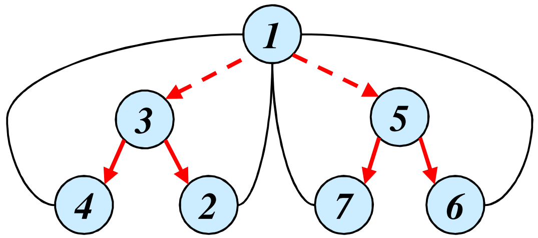
(Freuder 85, Bayardo 95, Bodlaender and Gilbert, 91)

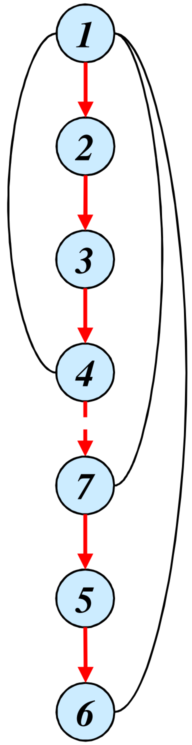


(a) Graph

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





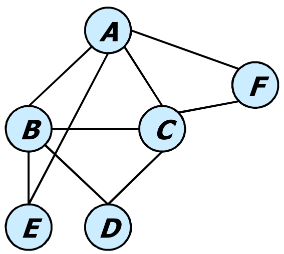


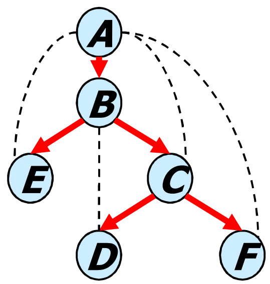
(b) DFS tree
depth=3

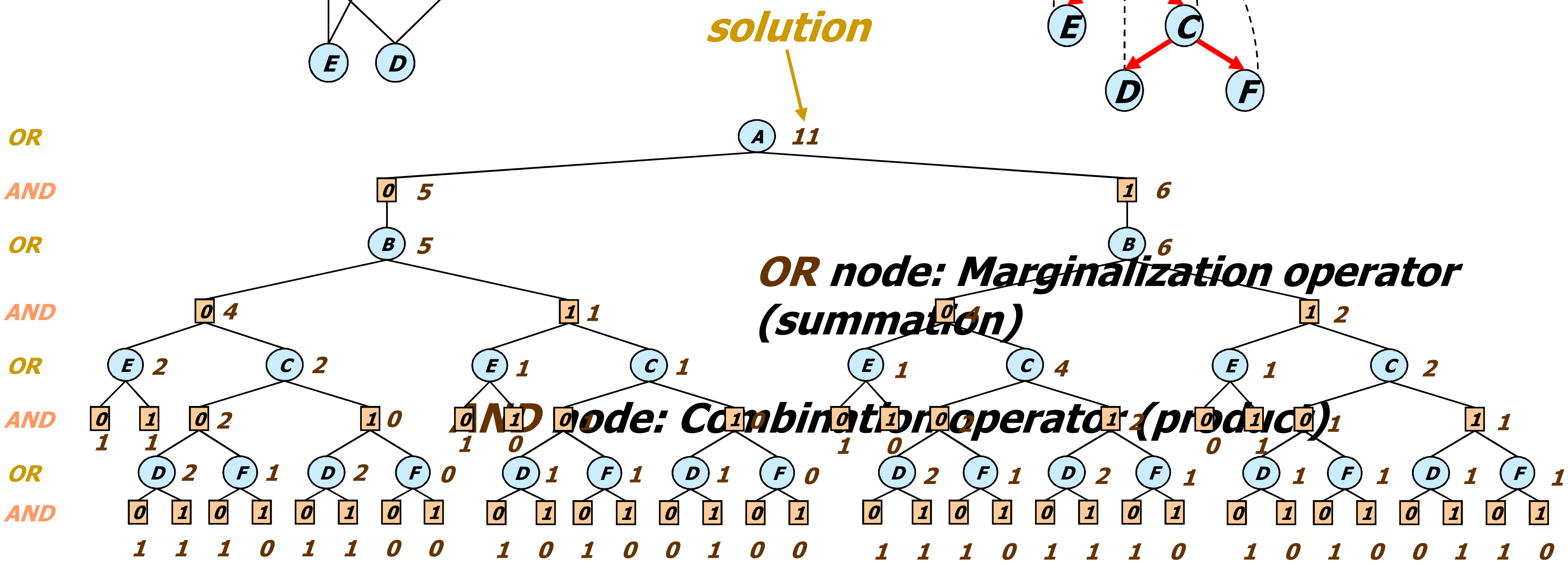
(c) pseudo- tree
depth=2

(d) Chain
depth=6

DFS algorithm (#CSP example)







Value of node = number of solutions below it

The recursive value rule

$$\mathbf{G}(\boldsymbol{Z}) = \Downarrow_{\boldsymbol{X}-\boldsymbol{Z}} \otimes_{\boldsymbol{f}\in\boldsymbol{F}} \mathrm{f(x)}$$

$$\mathrm{v(n)} = \mathrm{G}|_{n=(x_1,\ldots,x_i})$$

$$v(n) = \bigotimes_{n'\in children(n)} v(n'), \qquad \textit{if } n = \langle X, x\rangle \textit{ is an AND node,}$$

$$v(n) = \Downarrow_{n'\in children(n)} (w_{(n,n')} \bigotimes v(n')), \qquad \textit{if } n = X \textit{ is an OR node.}$$

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

$P(B \mid A)$

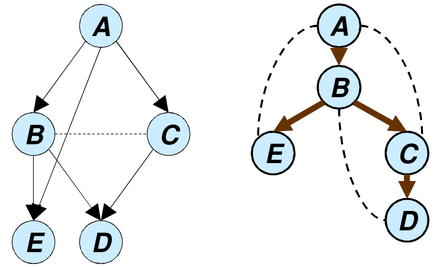
A	B=0	B=1
0	.4	.6
1	.1	.9

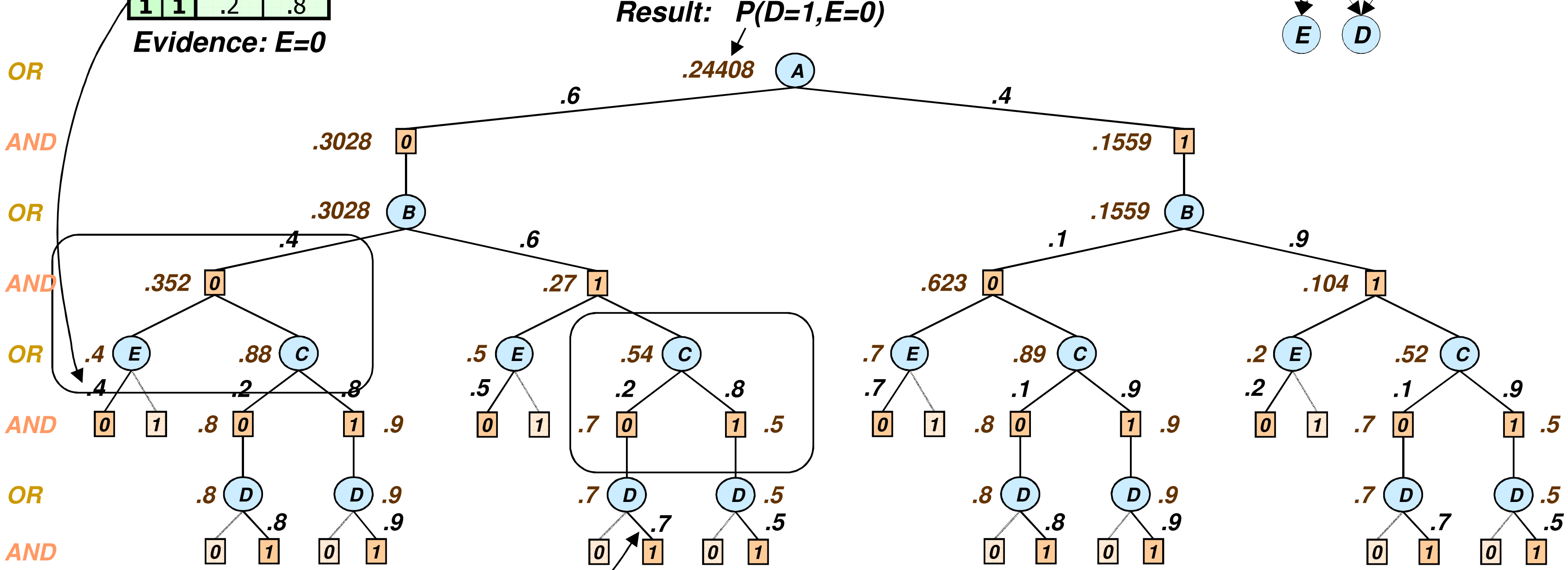
$P(C \mid A)$

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

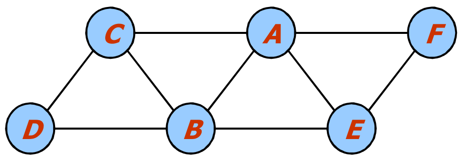
Evidence: D=1

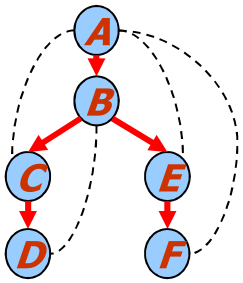
***OR* node: Marginalization operator (summation)**

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

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

AND/OR Tree Search for COP





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**

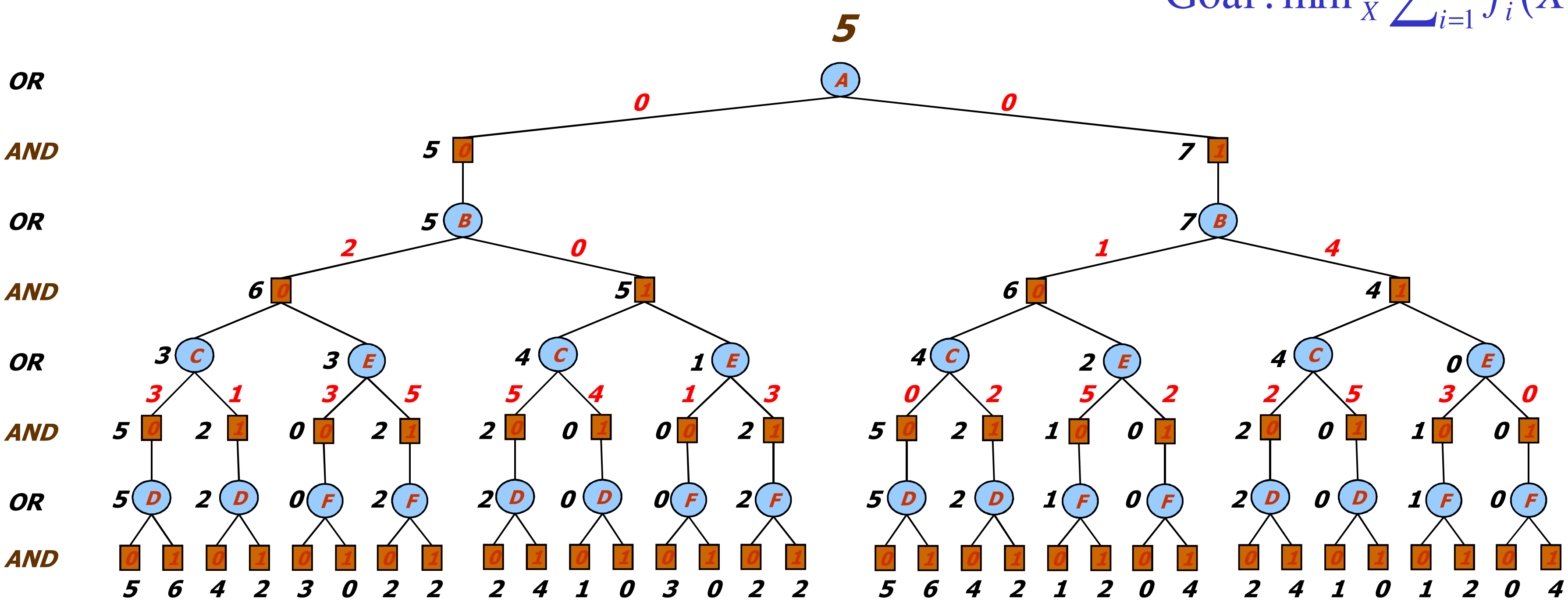
B	D	f_6
0	0	**4**
0	1	**2**
1	0	**1**
1	1	**0**

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**

$$\text{Goal} : \min_X \sum_{i=1}^{9} f_i(X)$$



AND node = Combination operator (summation)

OR node = Marginalization operator (minimization)

Properties of AND/OR Tree Search

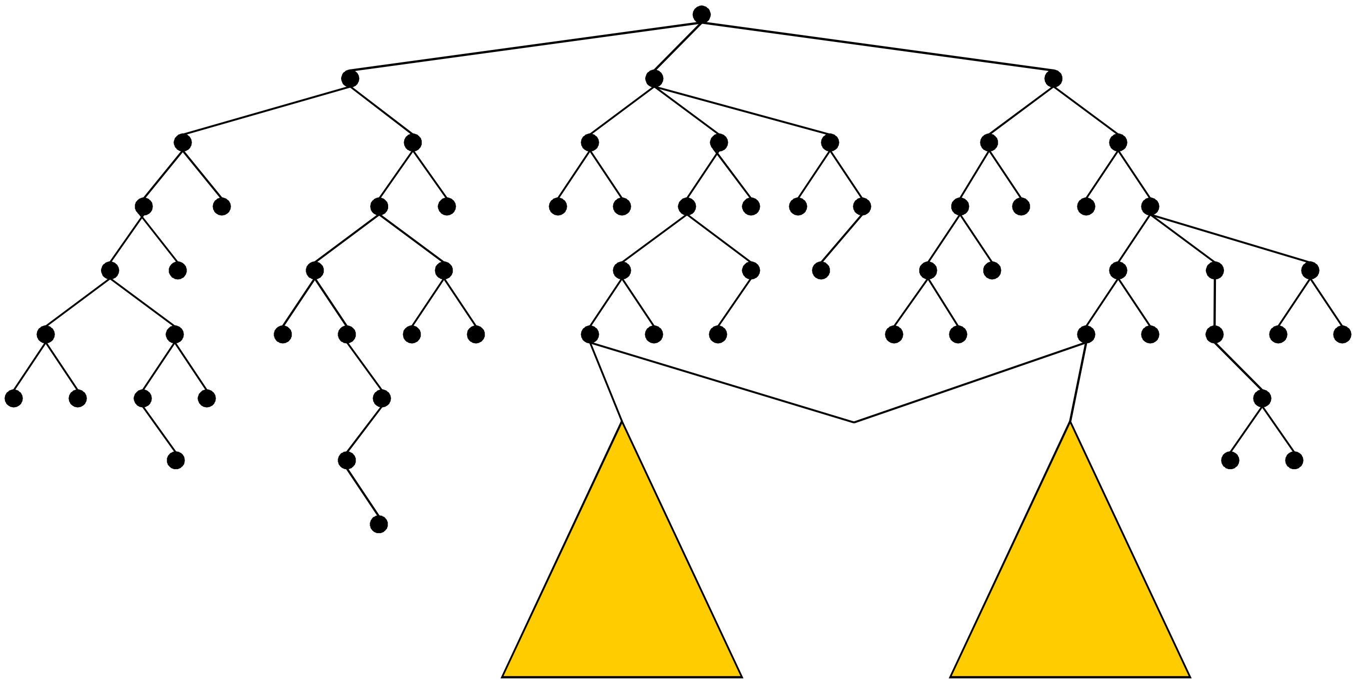
	AND/OR tree	**OR tree**
Space	O(n)	O(n)
Time	$O(n\ k^{h})$ $O(n\ k^{w^* \log n})$ (Freuder & Quinn85), (Collin, Dechter & Katz91), (Bayardo & Miranker95), (Darwiche01)	$O(k^{n})$

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

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

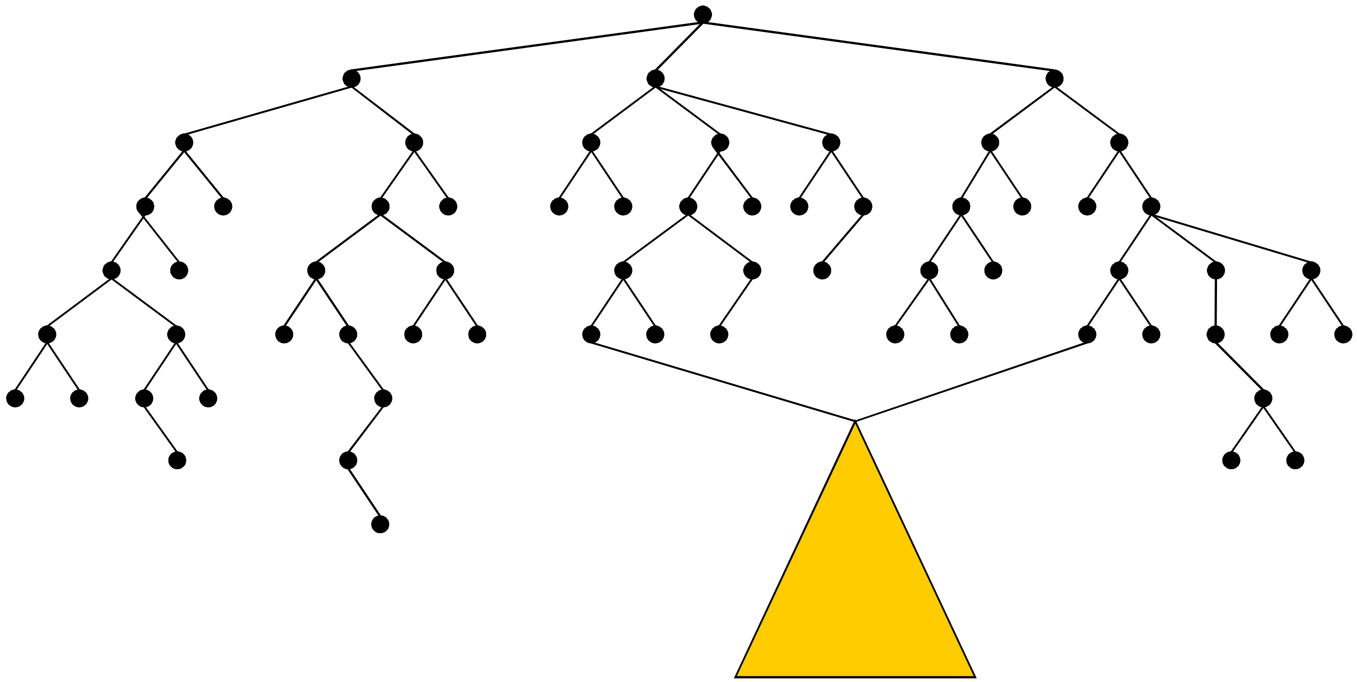
From Search Trees to Search Graphs

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

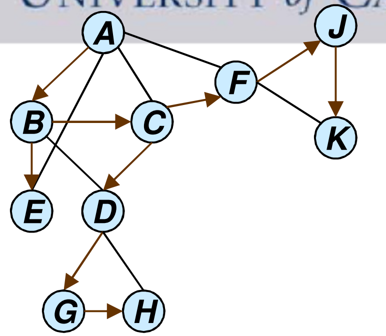


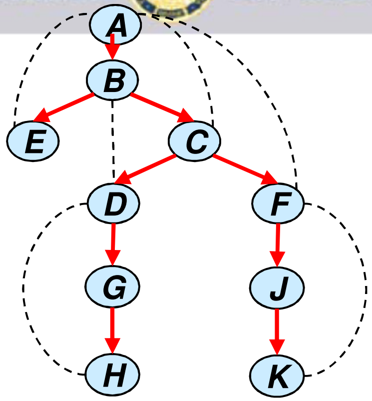
From Search Trees to Search Graphs

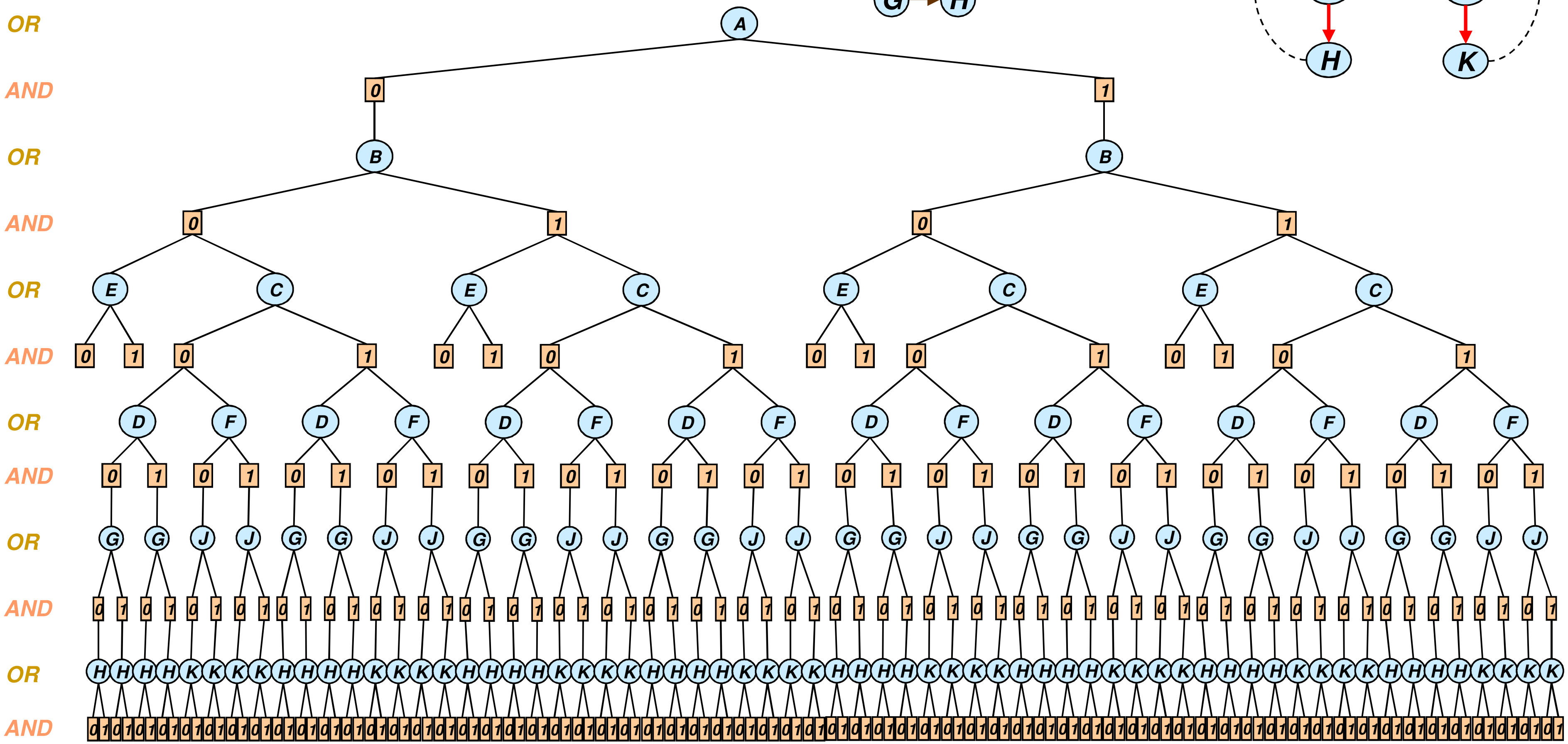
- Any two nodes that root identical sub-trees or sub-graphs can be merged



From AND/OR Tree



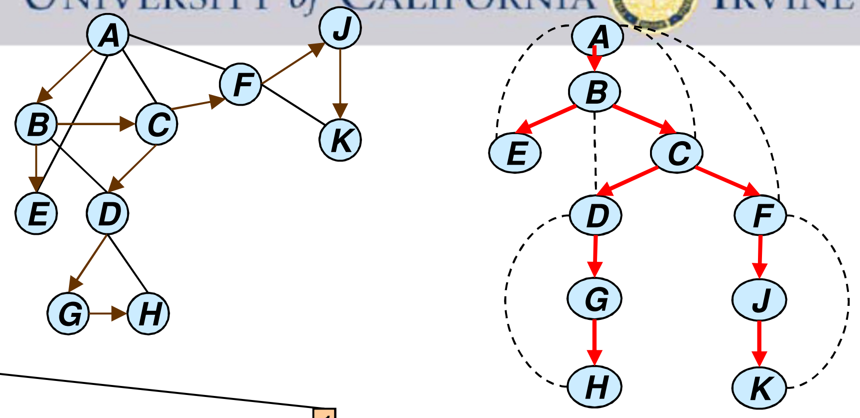


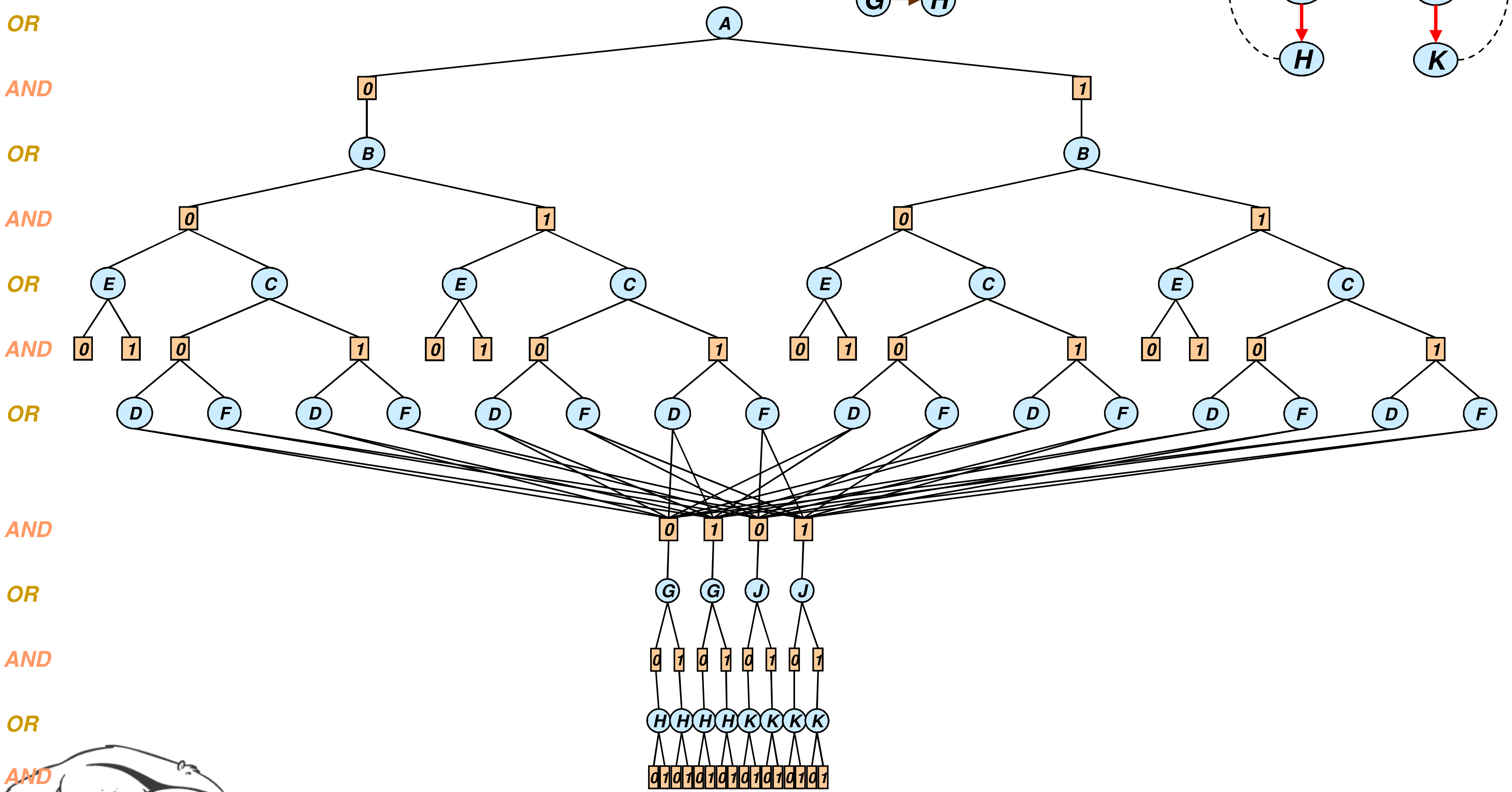


BREN:ICS
INFORMATION AND COMPUTER SCIENCES

UNIVERSITY of CALIFORNIA IRVINE

An AND/OR Graph

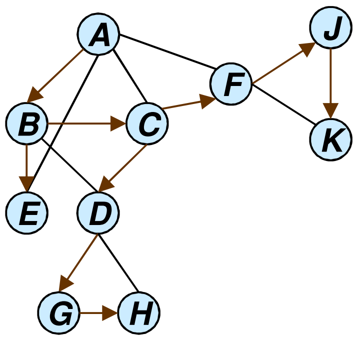


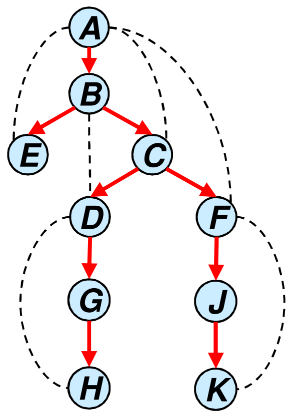


Context-based Caching

- 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 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

$P(B \mid A)$

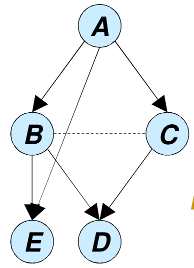
A	B=0	B=1
0	.4	.6
1	.1	.9

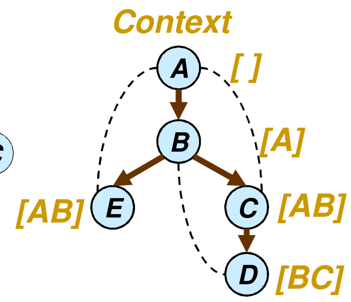
$P(C \mid A)$

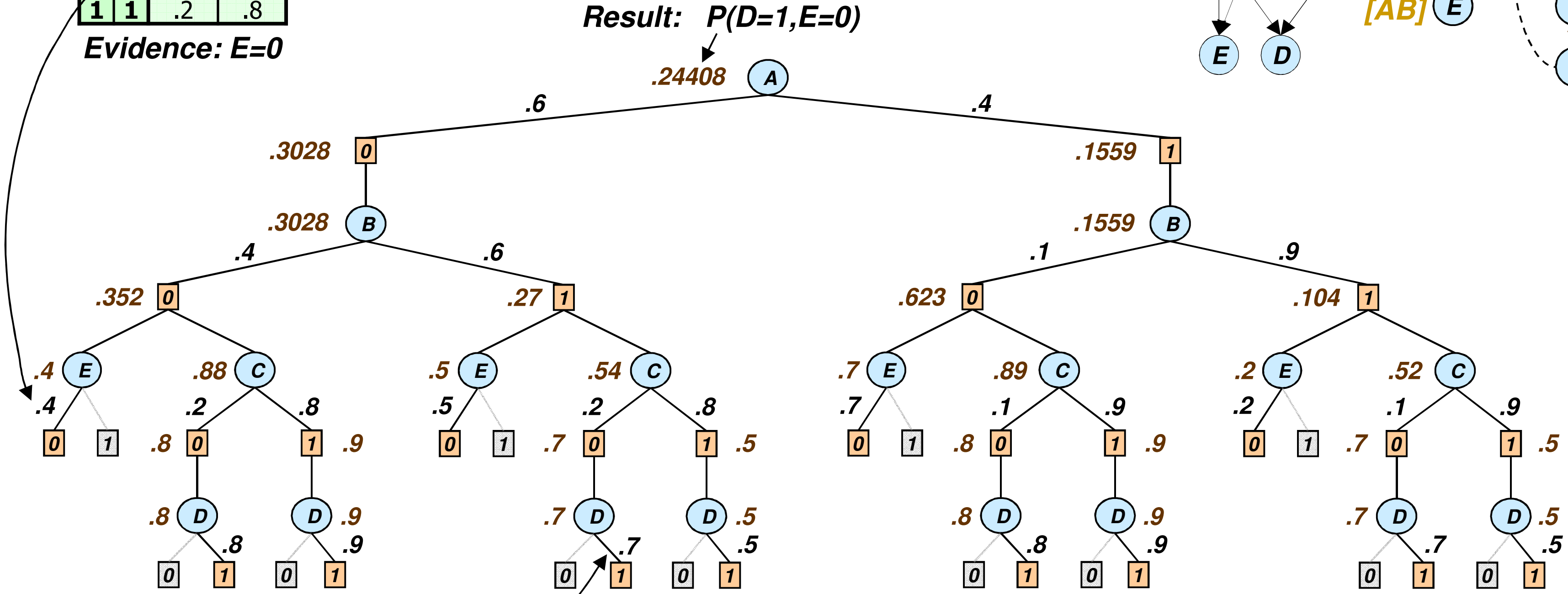
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

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

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

$P(B \mid A)$

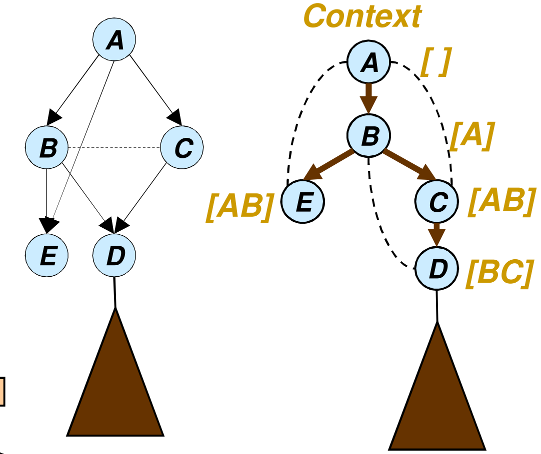
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

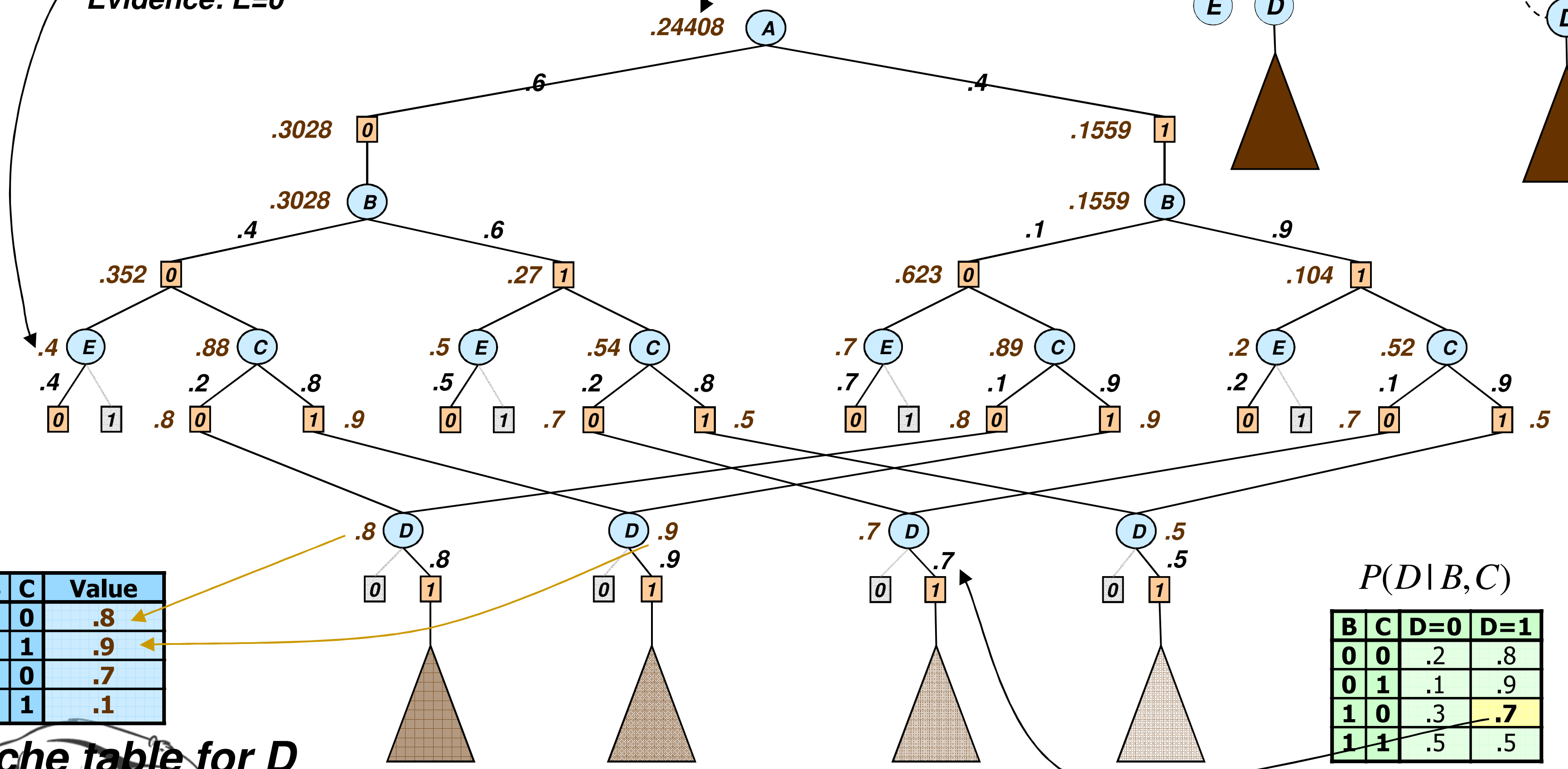
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



Result: P(D=1,E=0)



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

Cache table for D

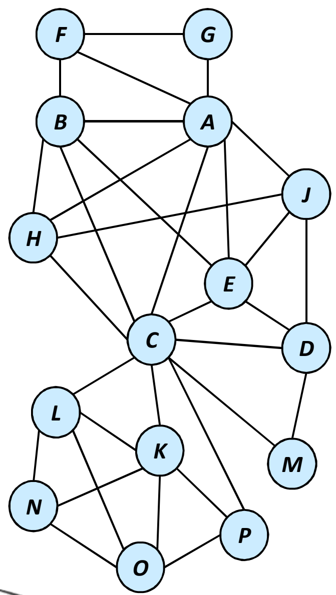
$P(D \mid B, C)$

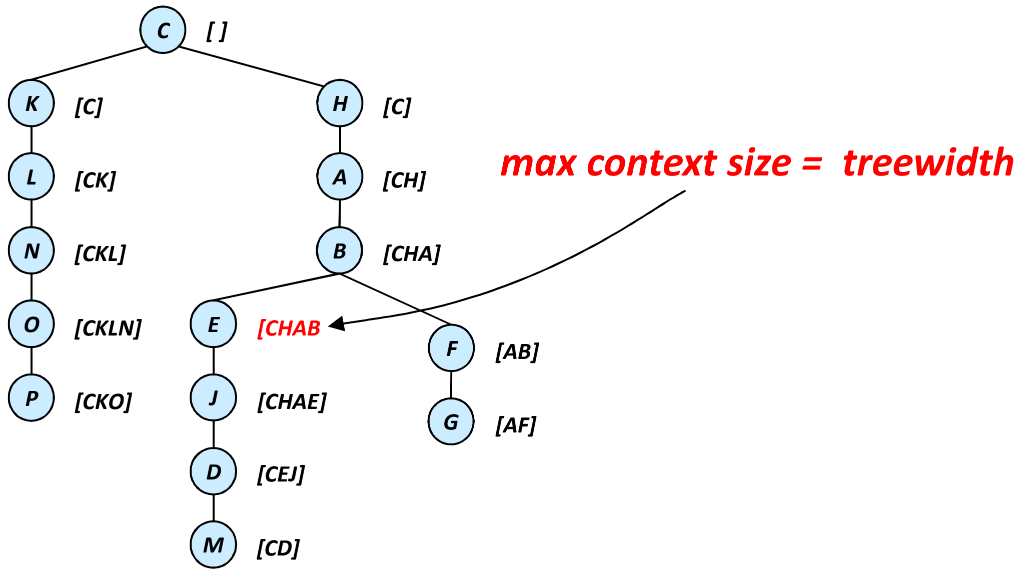
B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

How Big Is The Context?

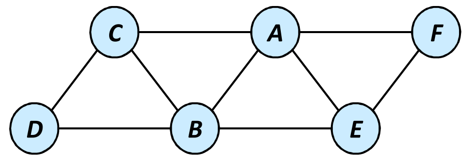
Theorem: The maximum context size for a pseudo tree is equal to the treewidth of the graph along the pseudo tree.



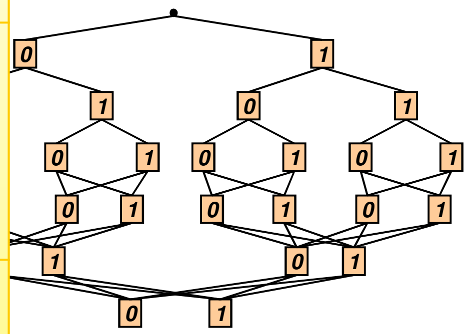


(C K H A B E J L N O D P M F G)

All Four Search Spaces



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

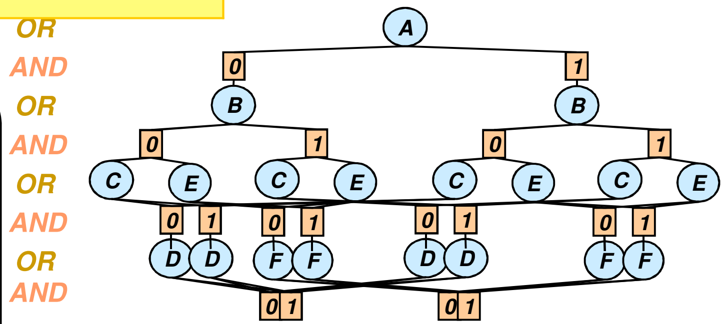


...xt minimal OR search graph

28 nodes

Computes any query:

- *Constraint satisfaction*
- *Optimization*
- *Weighted counting*

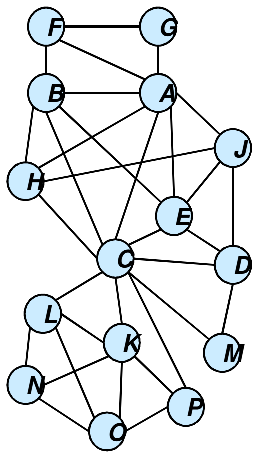


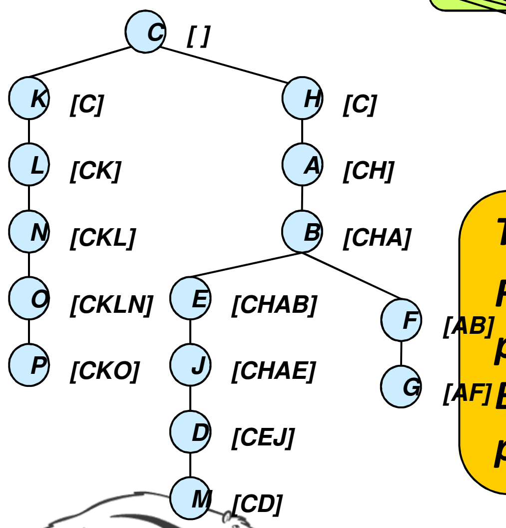
Context minimal AND/OR search graph

18 AND nodes

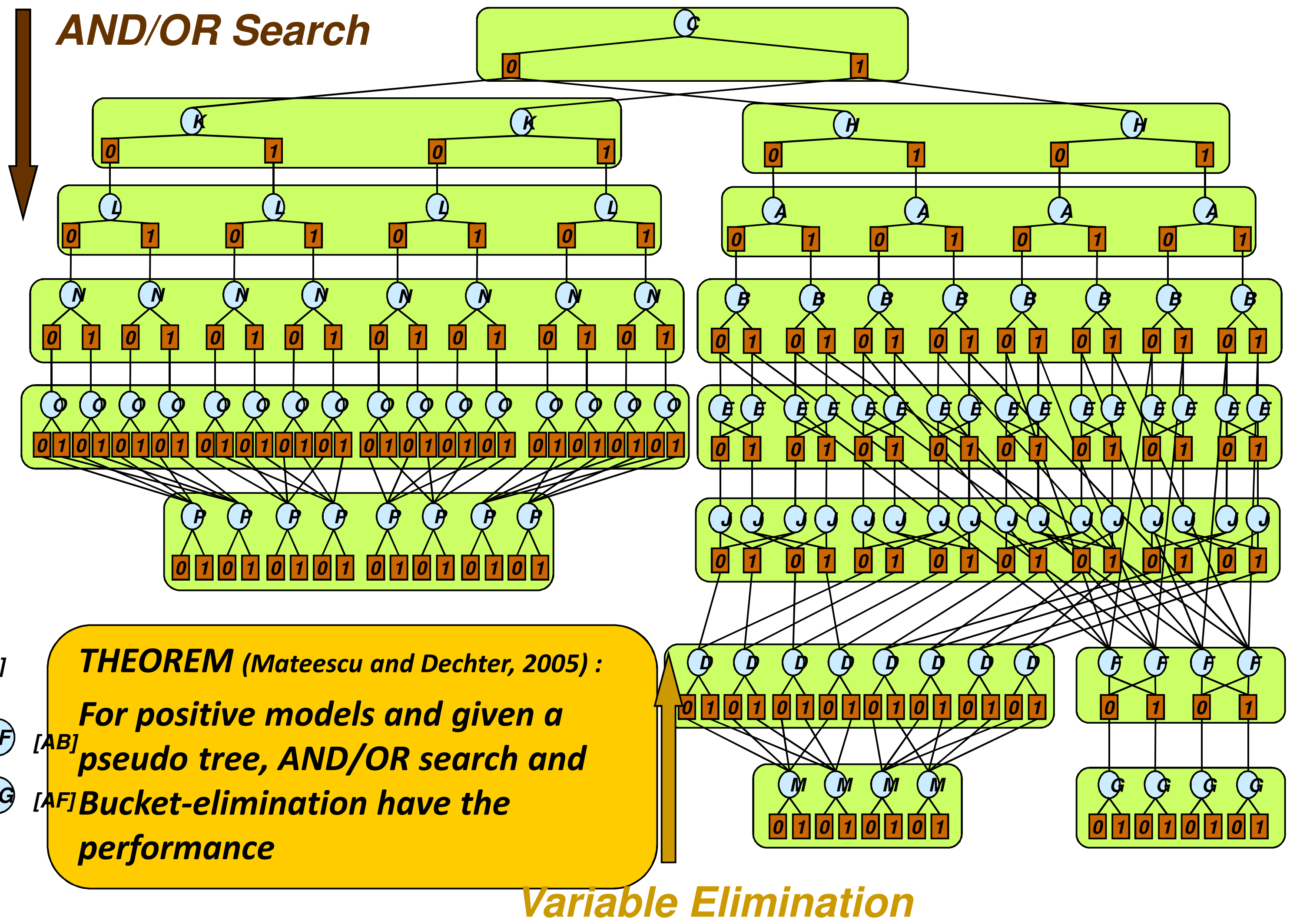
Any query is best computed Over the c-minimal AO space

What is better? Inference or Search





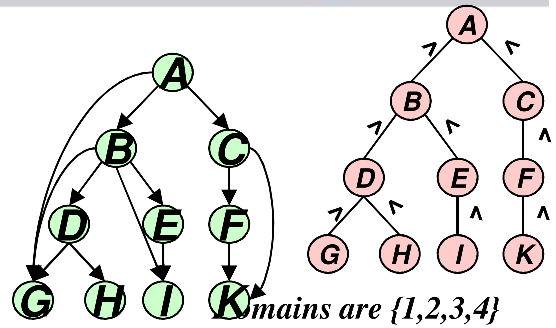
(C K H A B E J L N O D P M F G)

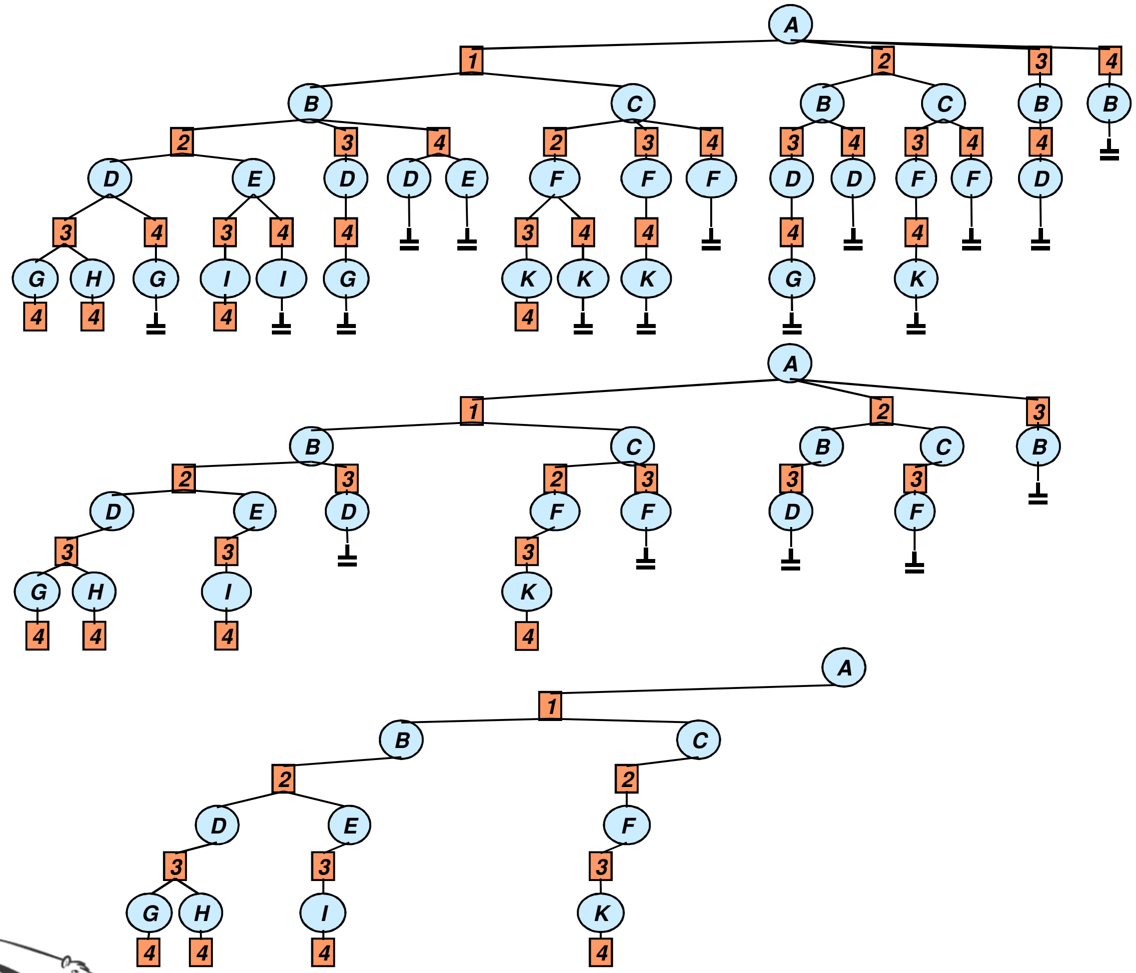


***THEOREM** (Mateescu and Dechter, 2005) :*

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

The Effect of Constraint Propagation





CONSTRAINTS ONLY

FORWARD CHECKING

MAINTAINING ARC CONSISTENCY

Outline

- What are graphical models?
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference
- Anytime AND/OR branch and bound
- Experiments, Competitions

and conclusions

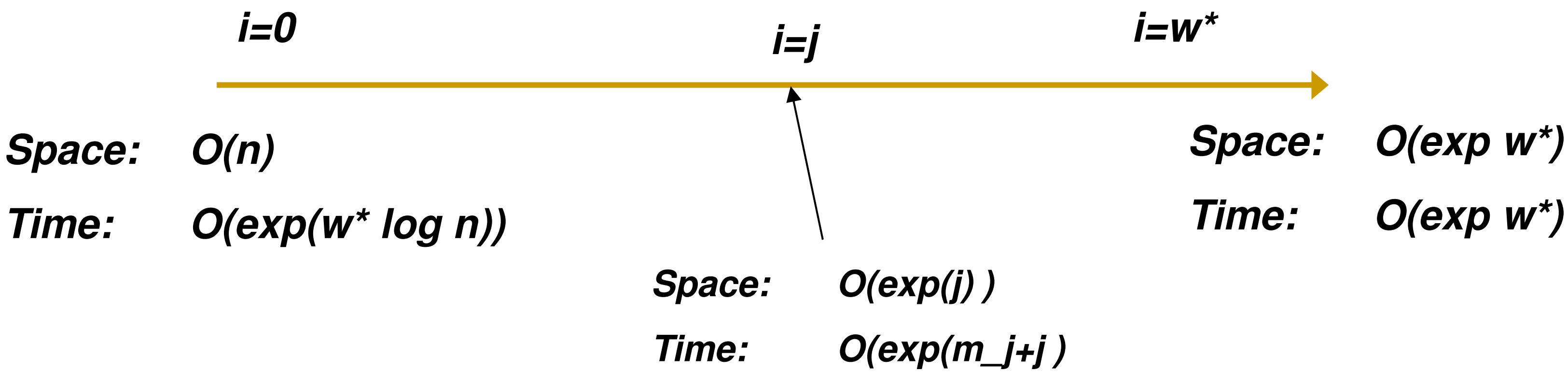
Exact algorithms

Approximations

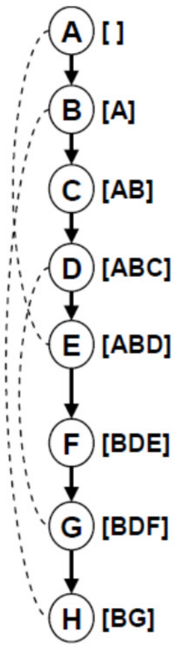
Back to exact: Anytime schemes

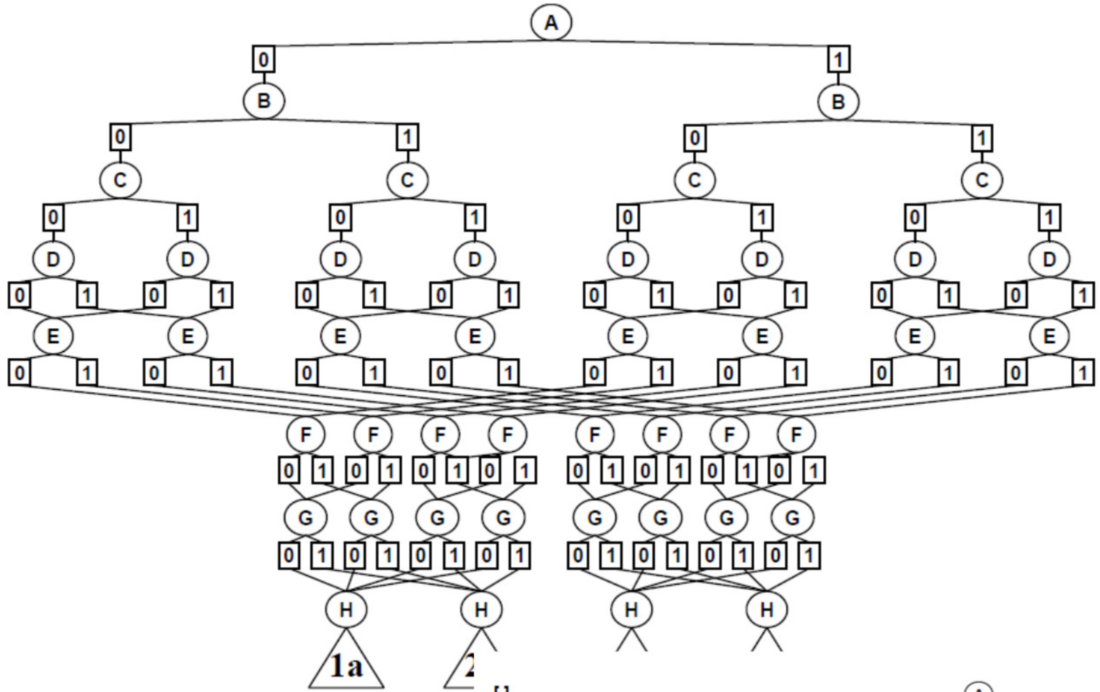
Searching AND/OR Graphs trading space for time

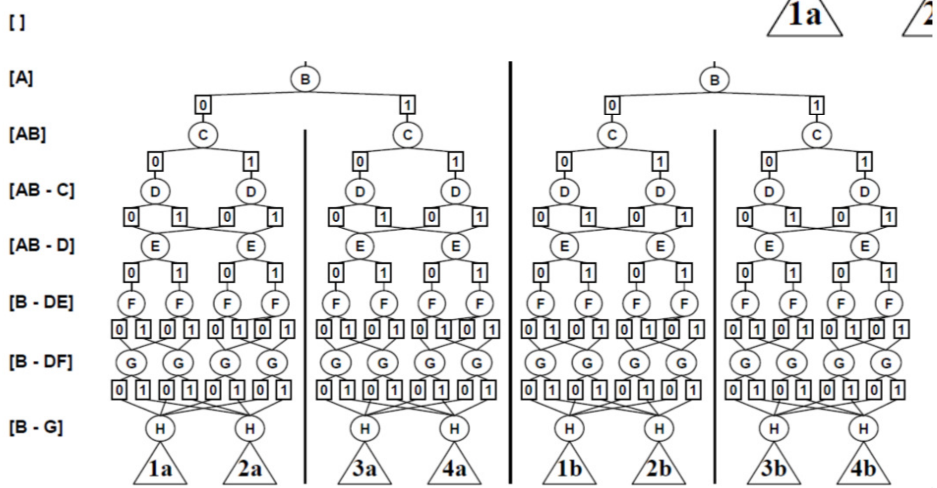
- AO(j): searches depth-first, cache i-context
  - j = the max size of a cache table (i.e. number of variables in a context)

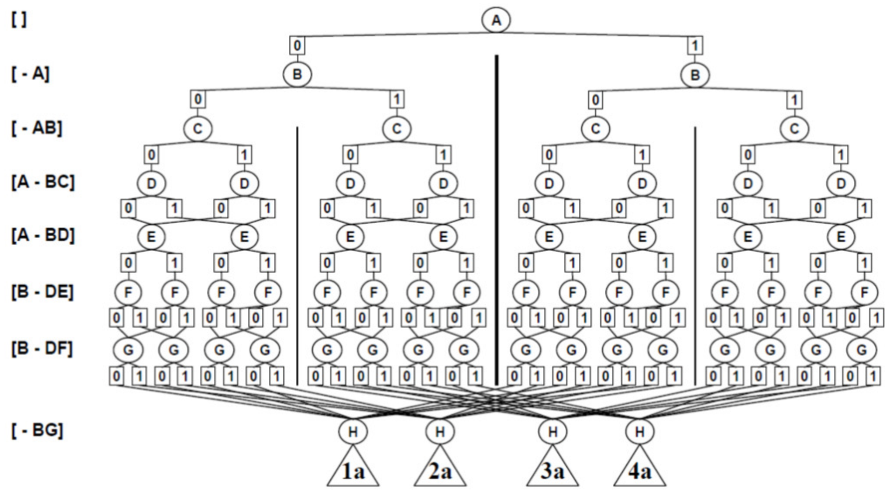


AOC(j) Adaptive Caching









Outline

- What are graphical models?
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference (mini-bucket, cost-shifting, belief propagation)
- Anytime AND/OR branch and bound
- Experiments and conclusions

Exact algorithms

Approximations

Back to exact: Anytime schemes

Mini-bucket approximation:MPE task
(Dechter and Rish, 1997, 2003)

Split a bucket into mini-buckets =>bound complexity

$$\textbf{bucket}(X) = \{ h_1, \ldots, h_r, h_{r+1}, \ldots, h_n \}$$

$$h^X = \max_X \prod_{i=1}^{n} h_i$$

$$\{ h_1, \ldots, h_r \} \qquad \{ h_{r+1}, \ldots, h_n \}$$

$$g^X = (\max_X \prod_{i=1}^{r} h_i) \cdot (\max_X \prod_{i=r+1}^{n} h_i)$$

$$h^X \leq g^X$$

Exponential complexity decrease : $O(e^n) \rightarrow O(e^r) + O(e^{n-r})$

Mini-Bucket Element

$max_B \Pi$ $max_B \Pi$

Bucket B P(E|B,C) P(B|A) P(D|A,B)

Bucket C P(C|A) h^B (C,E)

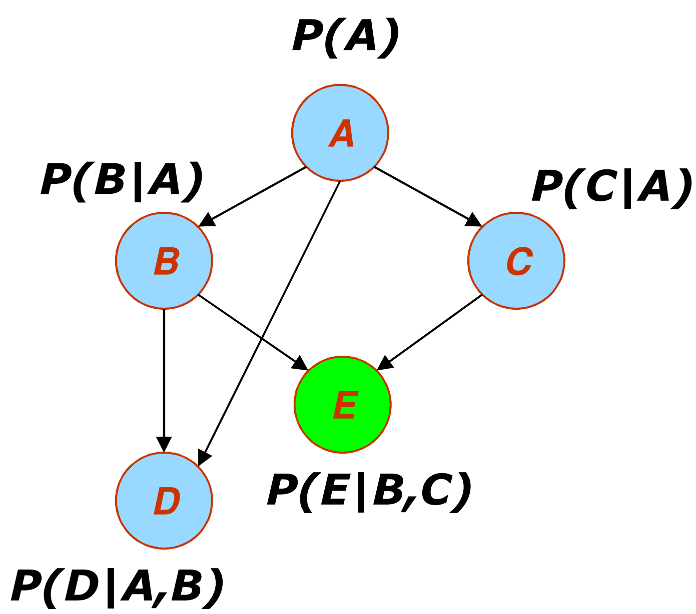
Bucket D h^B (A,D)

Bucket E E = 0 h^C (A,E)

Bucket A P(A) h^E (A) h^D (A)

W=2

Node duplication, renaming



MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

Mini-Clustering (for Sum-Product)

Split a cluster into mini-clusters => bound complexity

$$\{h_1, \dots, h_r, h_{r+1}, \dots, h_n\}$$

APPROXIMATE algorithm

$$\{h_1, \dots, h_r\} \qquad \{h_{r+1}, \dots, h_n\}$$

$$\sum_{elim} \prod_{i=1}^{n} h_i \leq \left(\sum_{elim} \prod_{i=1}^{r} h_i \right) \cdot \left(\sum_{elim} \prod_{i=r+1}^{n} h_i \right)$$

$$O(e^{n}) \rightarrow O(e^{\text{var}(r)}) + O(e^{\text{var}(n-r)})$$

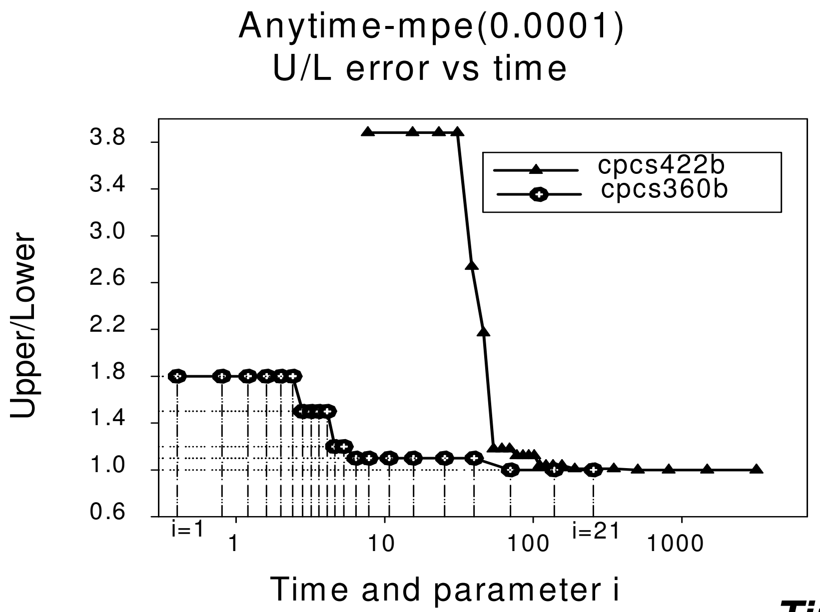
Exponential complexity decrease

Properties of MBE(i)

- **Complexity:** *O(r exp(i))* time and *O(exp(i)) space*.
- Yields an upper-bound and a lower-bound.
- **Accuracy:** determined by upper/lower (U/L) bound.
- As ***i*** increases, both accuracy and complexity increase.
- Possible use of mini-bucket approximations:
  - As anytime algorithms
  - As heuristics in search
- Other tasks: similar mini-bucket approximations for: belief updating, MAP and MEU (Dechter and Rish, 1997)

CPCS networks – medical diagnosis
(noisy-OR CPD’s)

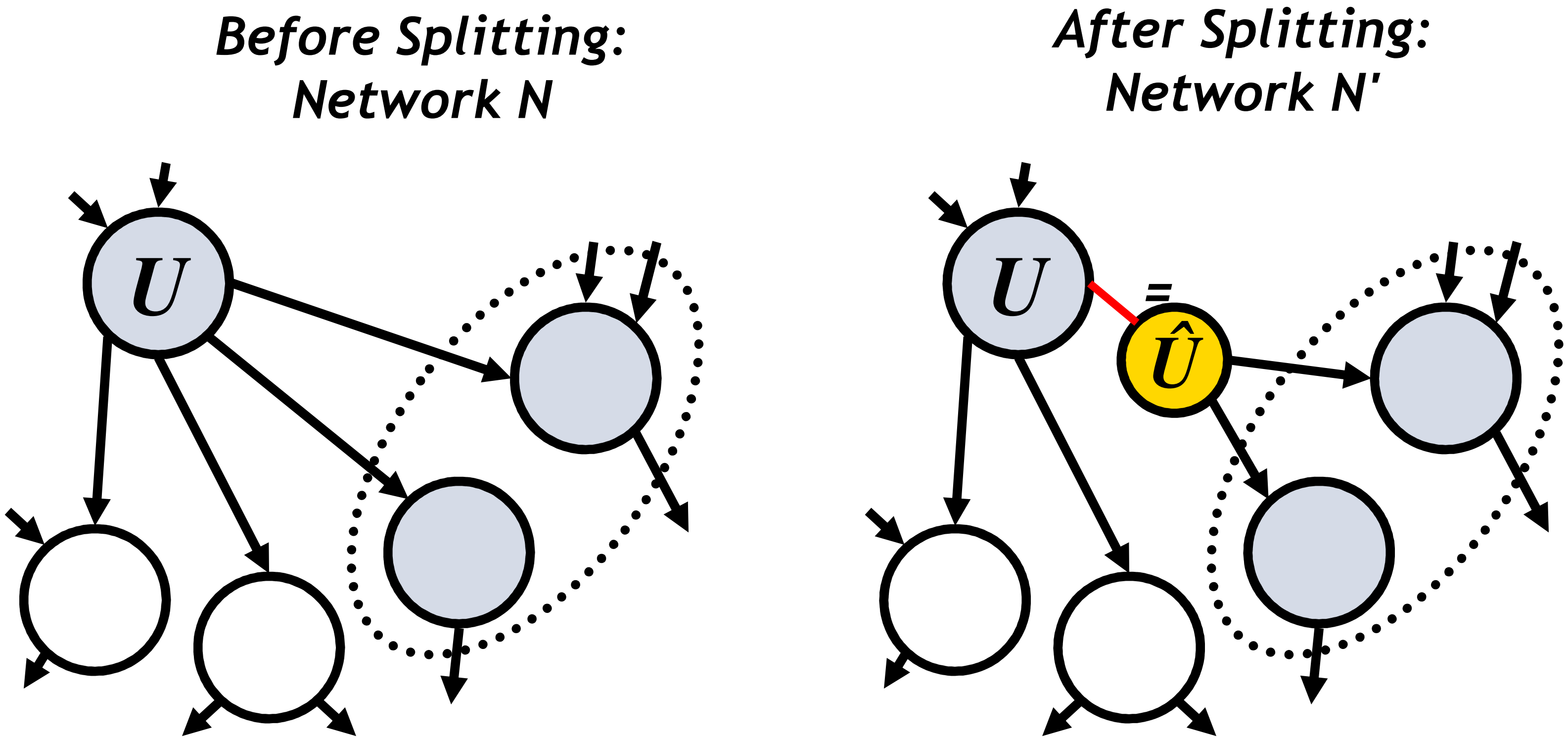
Test case: no evidence



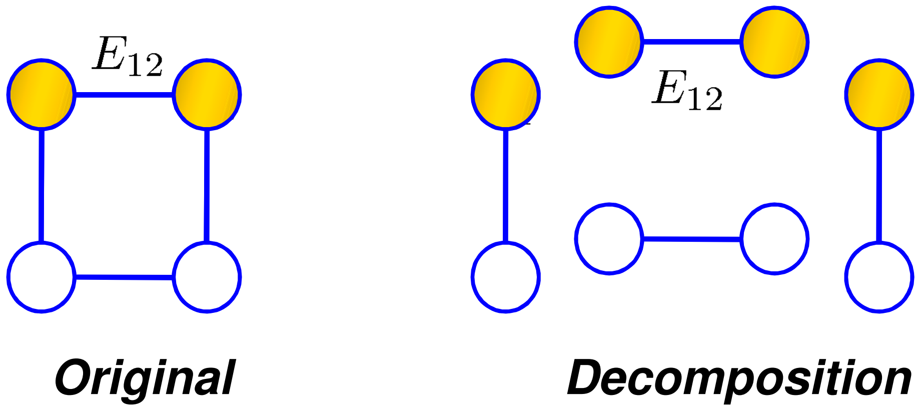
Algorithm	Time (sec) cpcs360	cpcs422
elim-mpe	115.8	1697.6
anytime-mpe(ε), $\varepsilon = 10^{-4}$	70.3	505.2
anytime-mpe(ε), $\varepsilon = 10^{-1}$	70.3	110.5

Mini-Bucket: duplicating/Splitting a Node

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



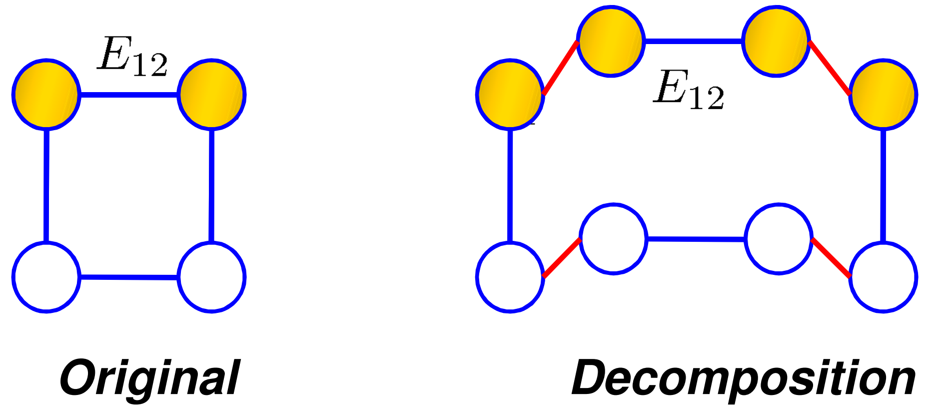
Tightening Bounds via cost-shifting



$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \leq \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j)$$

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree

Decomposition view

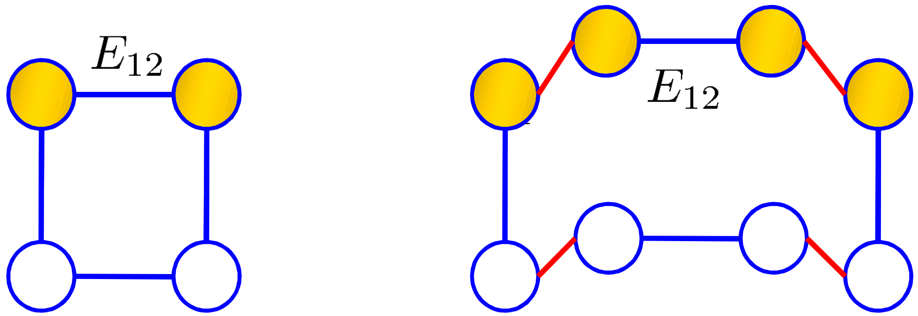


$$\forall i \sum_j \lambda_{ij}(x_i) = 0$$

$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{ij}(x_i) + \lambda_{ji}(x_j)$$

- Decompose graph into smaller subproblems
- Solve each independently; optimistic bound
- Exact if all copies agree
- Enforce lost equality constraints via Lagrange multipliers

Decomposition view



Original ***Decomposition***

$$\forall i \sum_j \lambda_{ij}(x_i) = 0$$

$$\max_{\underline{x}} \sum_{ij} E_{ij}(x_i, x_j) \le \min_{\lambda} \sum_{ij} \max_{\underline{x}} E_{ij}(x_i, x_j) + \lambda_{ij}(x_i) + \lambda_{ji}(x_j)$$

Same bound by different names

- Dual decomposition (Komodakis et al. 2007)
- TRW, MPLP (Wainwright et al. 2005; Globerson & Jaakkola 2007)
- Soft arc consistency (Cooper & Schiex 2004)

Join-graph based cost-shifting

m_{11}, m_{12}- moment-matching messages

Bucket B: $\max_B \prod$ P(E|B,C) ⟷ (m_{11} →, ← m_{12}) $\max_B \prod$ P(B|A) P(D|A,B)

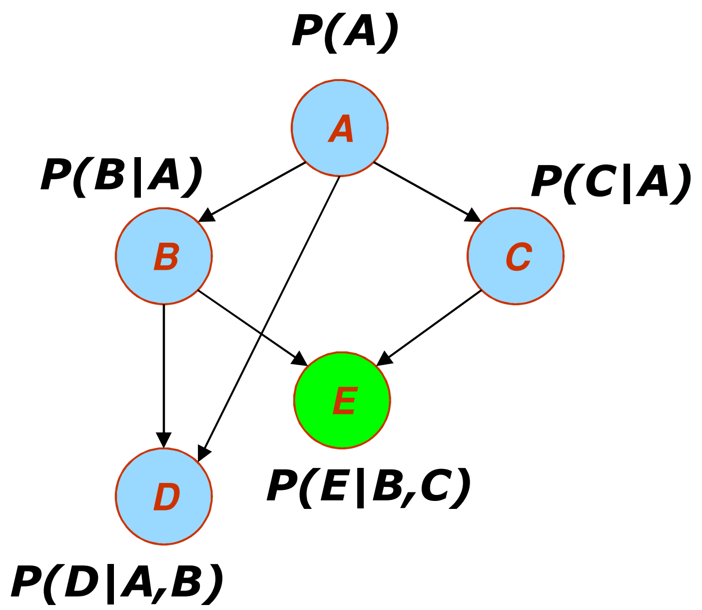
Bucket C: P(C|A) h^B (C,E)

Bucket D: h^B (A,D)

Bucket E: E = 0 h^C (A,E)

Bucket A: P(A) h^E (A) h^D (A)

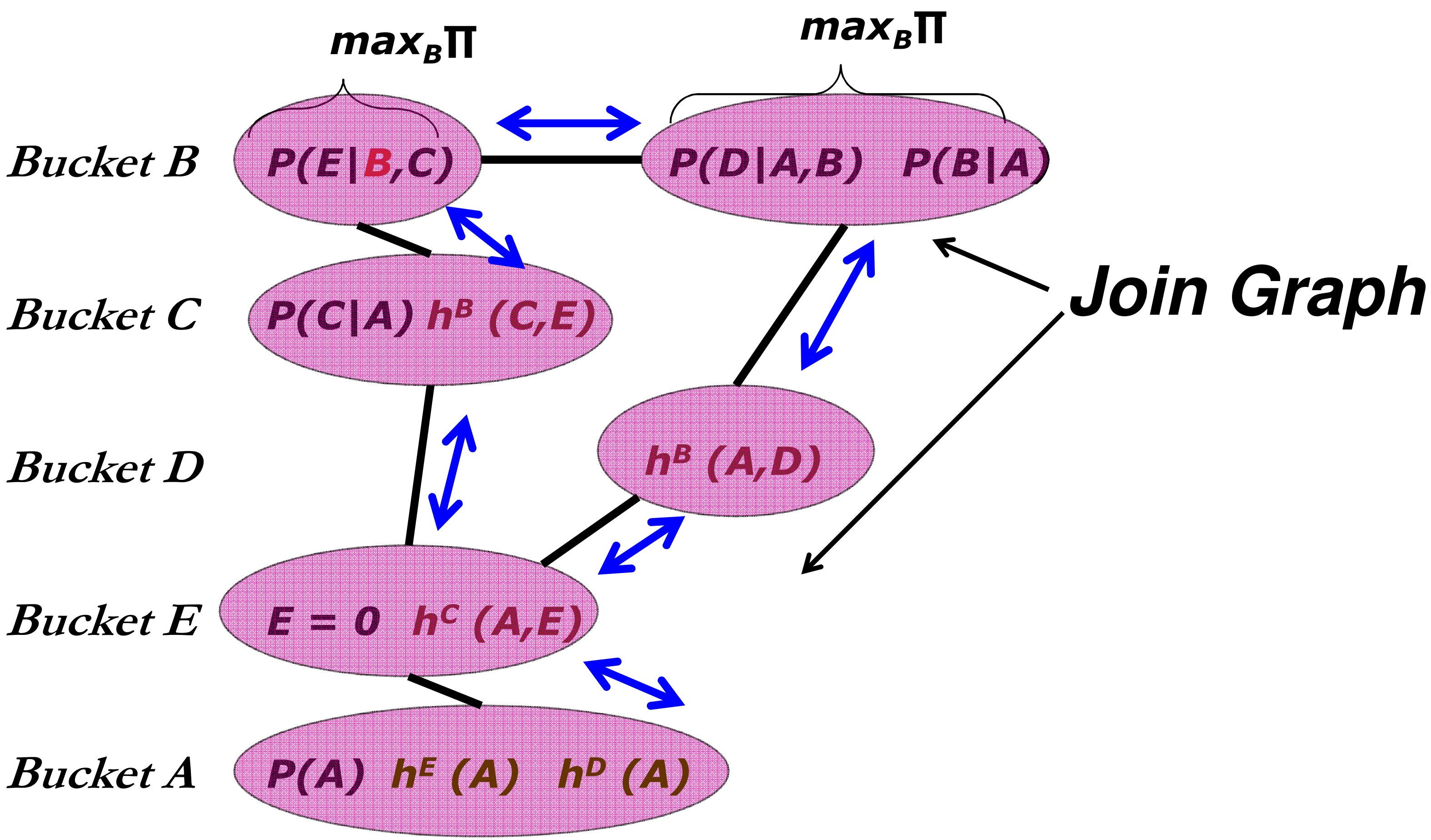
W=2



MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

Join-graph based cost-shifting

(Ihler, Flerova, Dechter, Otten, UAI 2012)



AOBB + MBE+ Join-graph based cost-shifting

Alexander Ihler, Natalia Flerova, Rina Dechter and Lars Otten (UAI '12)

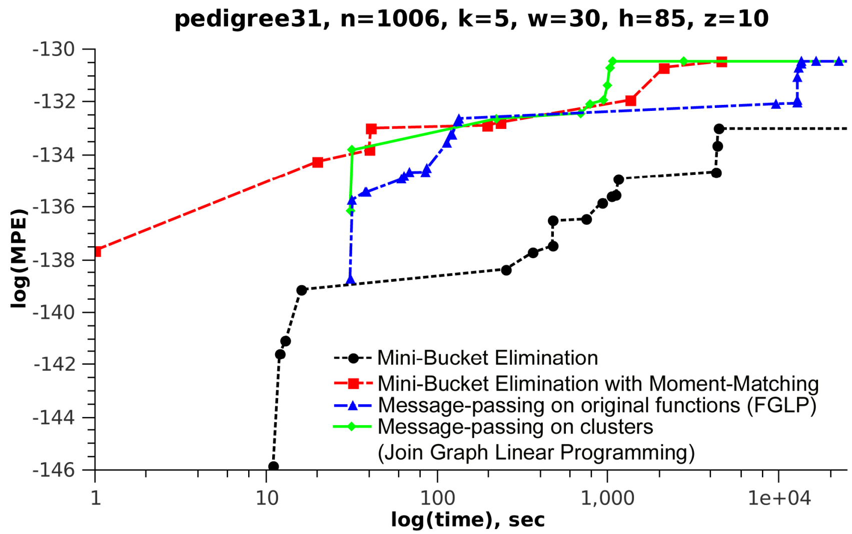
Mini-Bucket Elimination(i):
- *single-pass*
- *operates on clusters of size i*

\+

Linear Programming:
- *iterative*
- *operates on original factors*

Join Graph Linear Programming

MBE with Moment-Matching



Outline

- What are graphical models?
- Inference
- Search; via AND/OR search
- Time vs space, search vs inference
- Bounding inference
- Anytime AND/OR branch and bound
- Experiments, Competitions

and conclusions

Exact algorithms

Approximations

Back to exact: Anytime schemes

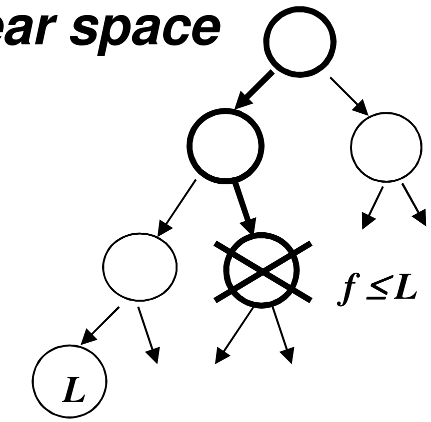
Basic Heuristic Search Schemes

Heuristic function $f(x^p)$ computes a lower bound on the best extension of x^p and can be used to guide a heuristic search algorithm. We focus on:

1. Branch-and-Bound

Use heuristic function $f(x^p)$ to prune the depth-first search tree

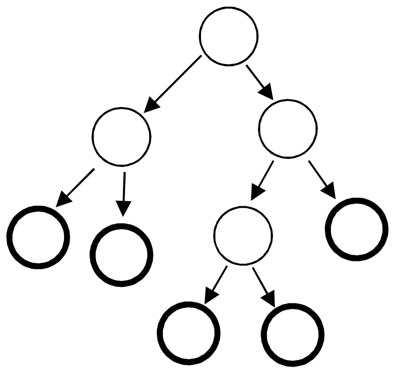
Linear space



2. Best-First Search

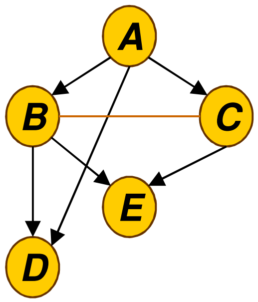
Always expand the node with the highest heuristic value

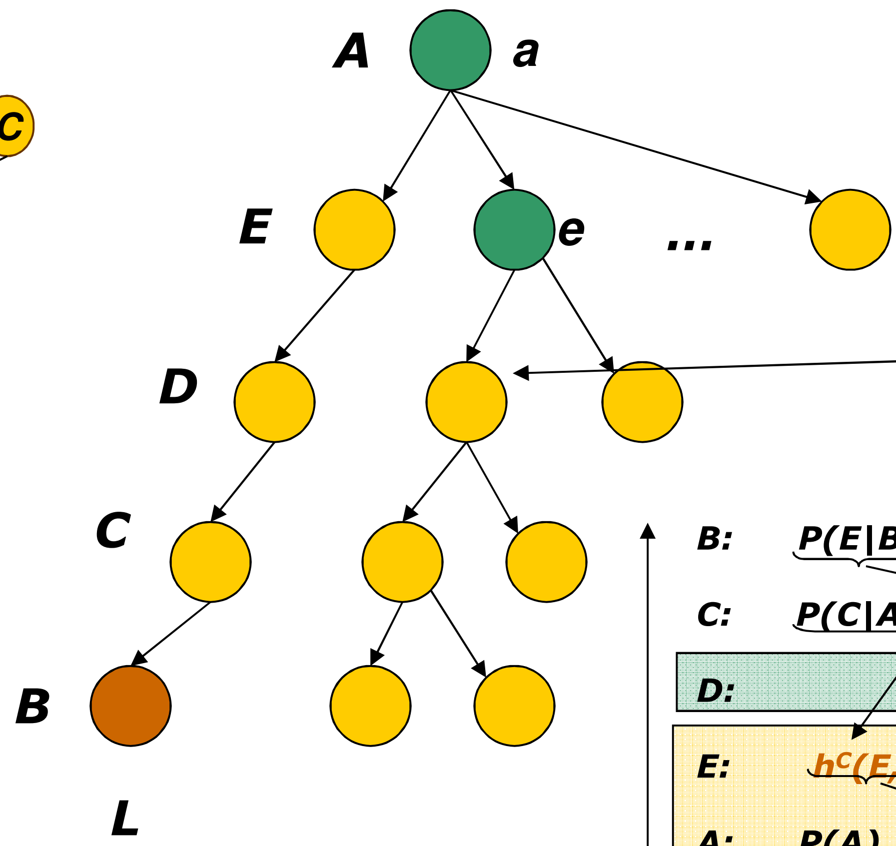
$f(x^p)$ needs lots of memory



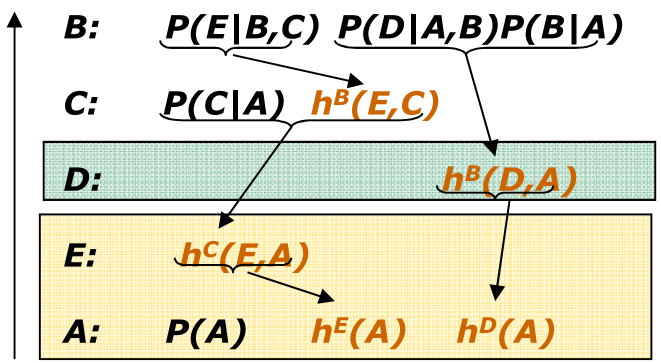
Mini-bucket Heuristics for BB search

(Kask and dechterAIJ, 2001, Kask, Dechter and Marinescu 2004, 2005, 2009, Otten 2012)



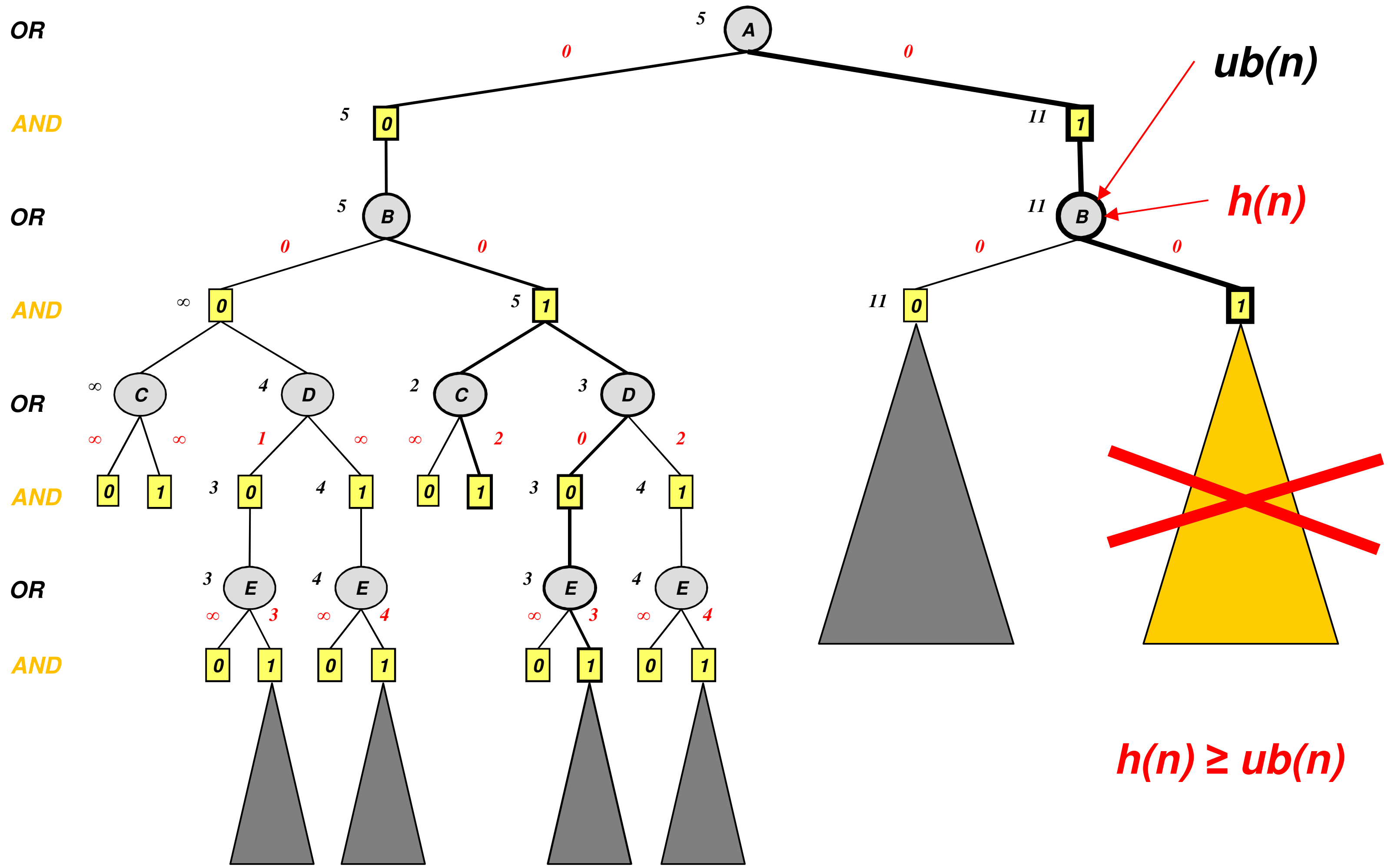


h(x) computed by MB(i) before or during search



$$f(a,e,D) = P(a) \cdot h^B(D,a) \cdot h^C(e,a)$$

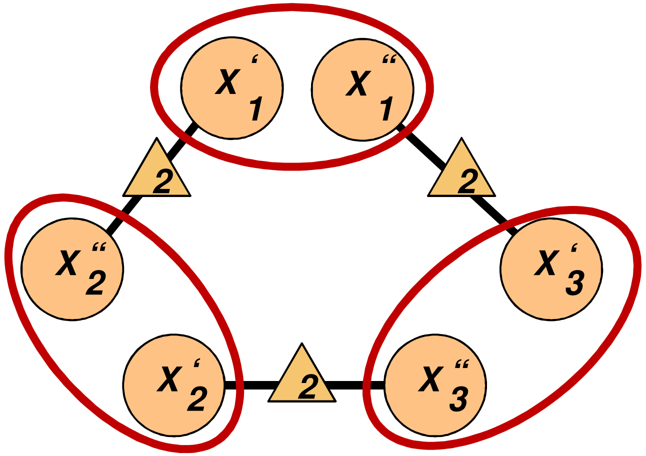
AND/OR Branch-and-Bound



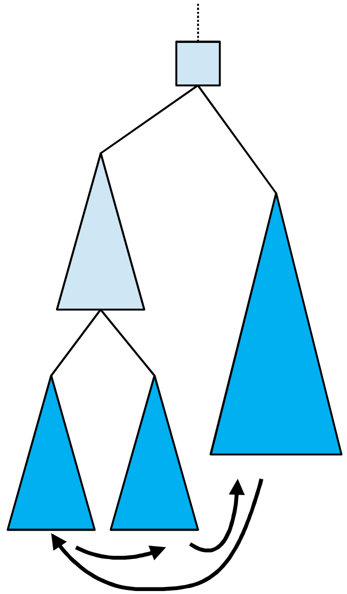
PASCAL 2011 Probabilistic Inference Challenge

- http://www.cs.huji.ac.il/project/PASCAL/
- Evaluates solvers in three categories:
  - PR: Probability of evidence / partition function
  - MAR: Posterior node marginals
  - MPE: Most probable explanation (our entry)
- Three tracks each: 20 sec, 20 min, 1 hour.
- Variety of benchmark domains:
  - CSPs, Deep Belief Nets, Image Alignment and Segmentation, Object Detection, Protein Folding, …

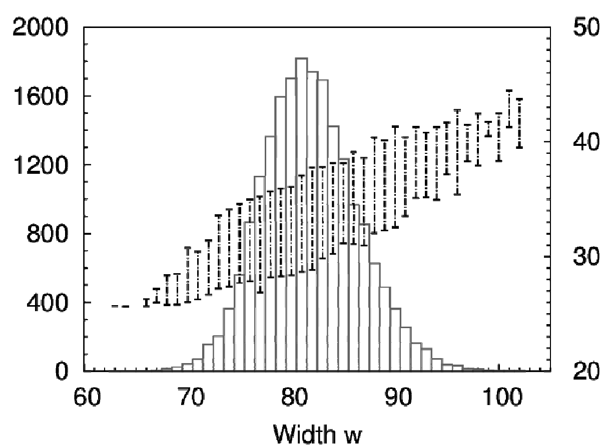
AOBB + Central Enhancements



$$\min_{\lambda} \sum_{(ij)} \max_{X}(f_{ij}(X_i, X_j) + \lambda_{ij}(X_i), \lambda_{ji}(X_j))$$



+ SLS *(Hutter et. Al 2005)*



Cost-shifting (MPLP) Re-parametrization

Tighter bounds by iteratively solving linear programming relaxations and message passing on join graph.

Breadth-First Subproblem Rotation

Improved anytime performance through interleaved processing of independent subproblems.

Enhanced Variable Ordering Schemes

Highly efficient, stochastic minfill / mindegree implementations for lower-width orderings.

(Ihler ,Flerova, Dechter, Otten, 2012, Otten and Dechter 2011, Kask, Gelfand, Otten, Dechter 2010)

Competition:
Gurobi vs AOBB on Pascal2 instances (Junkyu and Lam, ongoing)

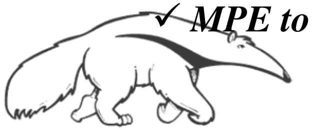
Benchmark

Problem	PEDIGREE	WCSP	Protein Folding
Total #. Instances	22 From UAI'08 Competition	61 From PASCAL2Competition	10 From PASCAL2Competition
Compared Instances	22	15 Exclude 46 Indeterminate Cases*	7 Exclude 3 Indeterminate Cases*
BRAOBB (G.M of Time)	51.07 sec G.M of 20 instances terminated for both	15.96 sec G. M of 13 instances terminated for both	1004.25 sec G. M of 7 instances, terminated for AOBB
GUROBI (G.M of Time)	7.15 sec G.M of 20 instances terminated for both	72.65 sec G. M of 13 instances terminated for both	NA 9/10 Memory Out
BRAOBB VS .GUROBI	0 VS. 22	12 VS. 3	6 Vs. 1 **AOBB won by memory out of Gurobi**

Indeterminate Cases :

- ✓ ***Both Time out, or BRAOBB Time out and Gurobi Memory out (4GB),***
- ✓ ***One of the solver time out earlier than the other's running time***
- ✓ ***MPE to 0/1 ILP Conversion was not available for experiments***

Link to more results



WCSP

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
1	SPOT5 1502 209,412,4,3,6,6,15	1.72	24.57%	7
2	SPOT5 29 82,463,4,2,14,14,24	25.31	76.70%	33
3	SPOT5 404 100,711,4,3,19,19,45	310.58	2218.43%	14
4	SPOT5 503 143,636,4,3,9,9,44	622.49	8892.71%	7
5	SPOT5 54 67,272,4,3,11,11,19	28.15	402.14%	7
6	SPOT5 42 190,1395,4,3,13,26,87	3600 (TO)	NA	289
7	bwt3ac 45,686,11,2,12,16,27	164.4	342.50%	48
8	bwt4ac 179,7110,18,2,7,42,90	(MO)	NA	697

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
9	driverlog01ac 71,619,4,2,9,9,30	7.74	96.75%	8
10	GEOM30a_3 30,82,3,2,6,6,15	42.42	707.00%	6
11	GEOM30a_4 30,82,4,2,6,6,15	6.76	112.67%	6
12	GEOM30a_5 30,82,5,2,6,6,15	6.69	111.50%	6
13	myciel5g_3 47,237,3,2,14,21,24	1012.8	1875.56%	54
14	queen5_5_3 25,161,3,2,15,18,20	2744.57	8072.26%	34
15	queen5_5_4 25,161,4,2,12,18,20	977.24	534.01%	183

BRAOBB : Tout 60 min (MBE-MM, MPLP 2ec, JGLP 2 sec, SLS 2x2 sec, Ordering from Kalev's Code 3sec)
GUROBI : Tout 60 min(Default, Single Processor, Dual Simplex at the Root)
Memory Limit : Both 4 GB

Indeterminate Cases Total 46 (Not presented Above)

- ✓***Both Time Out : 5/46***
- ✓***AOBB Time Out, Gurobi Memory out 15/46***
- ✓***ILP Conversion NA : 26/46***

PROTEIN FOLDING

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
1	pdb1b25 1972,8817,81,2,3,51,178	(MO)	NA	21600 (TO)
2	pdb1d2e 1328,5220,81,2,3,22,136	(MO)	NA	557
3	pdb1fmj 614,2760,81,2,3,35,118	(MO)	NA	21600 (TO)
4	pdb1i24 337,1360,81,2,3,33,58	78.54	21.76%	361
5	pdb1iqc 1040,4042,81,2,3,26,107	(MO)	NA	1801

	name (n,f,k,s,i,w,h)	Gurobi Time	Tg/Ta	AOBB Time
6	pdb1jmx 739,2943,81,2,3,37,80	(MO)	NA	21600 (TO)
7	pdb1kgn 1060,4715,81,2,3,38,164	(MO)	NA	536
8	pdb1kwh 424,1881,81,2,3,27,93	(MO)	NA	16638
9	pdb1m3y 1364,5037,81,2,3,29,93	(MO)	NA	647
10	pdb1qks 926,3712,81,2,3,36,124	(MO)	NA	493

BRAOBB : Tout 6Hr (MBE-MM, MPLP 60 sec, JGLP 60 sec, SLS 20x10 sec, Ordering from Kalev's Code 3min)
GUROBI : Tout 6Hr (Default, Single Processor, Dual Simplex at the Root)
Memory Limit : Both 4 GB

Indeterminate Cases Total 3

- ✓***AOBB Time Out, Gurobi Memory Out 3***

Pedigree

name (n,f,k,s,i,w,h)		Gurobi Time	Tg/Ta	AOBB Time
1	Pedigree1 (298,335,4,5,15,15,60)	0.7	10.00%	7
2	Pedigree13 (888,1078,3,4,20,32,163)	15.8	2.33%	679
3	Pedigree18 (931,1185,5,5,19,19,98)	6.45	53.75%	12
4	Pedigree19 (693,794,5,5,14,24,99)	173.66	NA	1800 (TO)
5	Pedigree20 (387,438,5,4,19,22,73)	11.81	18.75%	63
6	Pedigree23 (309,403,5,4,18,25,60)	2.72	4.77%	57
7	Pedigree25 (993,1290,5,5,20,24,70)	3.1	16.32%	19
8	Pedigree30 (1015,1290,5,5,20,20,121)	7.25	55.77%	13
9	Pedigree31 (1006,1184,5,5,18,30,116)	20.32	18.81%	108
10	Pedigree33 (581,799,4,5,20,27,139)	4.12	9.36%	44
11	Pedigree34 (922,1161,5,4,17,30,130)	40.06	41.30%	97

name (n,f,k,s,i,w,h)		Gurobi Time	Tg/Ta	AOBB Time
12	Pedigree37 (726,1033,5,4,13,21,59)	3.09	11.44%	27
13	Pedigree38 (581,725,5,4,12,16,62)	5.99	13.31%	45
14	Pedigree39 (953,1273,5,4,20,20,78)	6.01	30.05%	20
15	Pedigree40 (842,1031,7,5,14,28,160)	221.15	NA	1800 (TO)
16	Pedigree41 (885,1063,5,5,18,32,113)	24.5	8.75%	280
17	Pedigree42 (390,449,5,4,15,23,60)	2.36	3.58%	66
18	Pedigree44 (644,812,4,5,20,26,72)	9.83	18.90%	52
19	Pedigree50 (478,515,6,4,10,17,53)	12.05	48.20%	25
20	Pedigree51 (871,1153,5,4,19,38,114)	14.64	3.30%	443
21	Pedigree7 (867,1069,4,4,19,31,119)	8.01	10.27%	78
22	Pedigree9 (935,1119,7,4,20,26,132)	7.85	25.32%	31

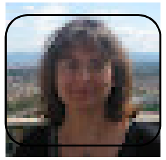
BRAOBB : Tout 30min (MBE-MM, MPLP 2ec, JGLP 2 sec, SLS 2x2 sec, Ordering from Kalev's Code 3sec)
GUROBI : Tout 30min (Default, Single Processor, Dual Simplex at the Root)
Memory Limit : Both 4 GB

Conclusions and Software

- **Conclusion:** Exact Search+Inference algorithms lead to effective anytime schemes and effective approximation and should be exploited.
- Search and inference are not so different; you may do effective inference by search
- Software
  - AND/OR search algorithms
  - Bucket-tree elimination
  - Generalized belief propagation
  - Samplesearch sampling
  - AOBB source code is freely available, under an open-source license.
  - http://graphmod.ics.uci.edu/group/Software

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
Lars Otten
Natalia Flerova
Andrew Gelfand
William Lam
Junkyu Lee