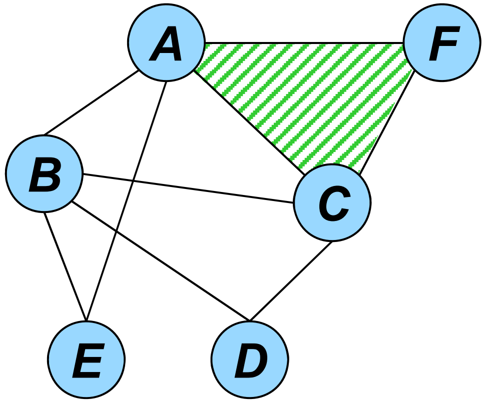
BREN:ICS INFORMATION AND COMPUTER SCIENCES UNIVERSITY of CALIFORNIA IRVINE

Modern Exact and Approximate Combinatorial Optimization Algorithms: Max-Product and Max-Sum-product

In the pursuit of universal solver

Rina Dechter

Donald Bren school of ICS, University of California, Irvine



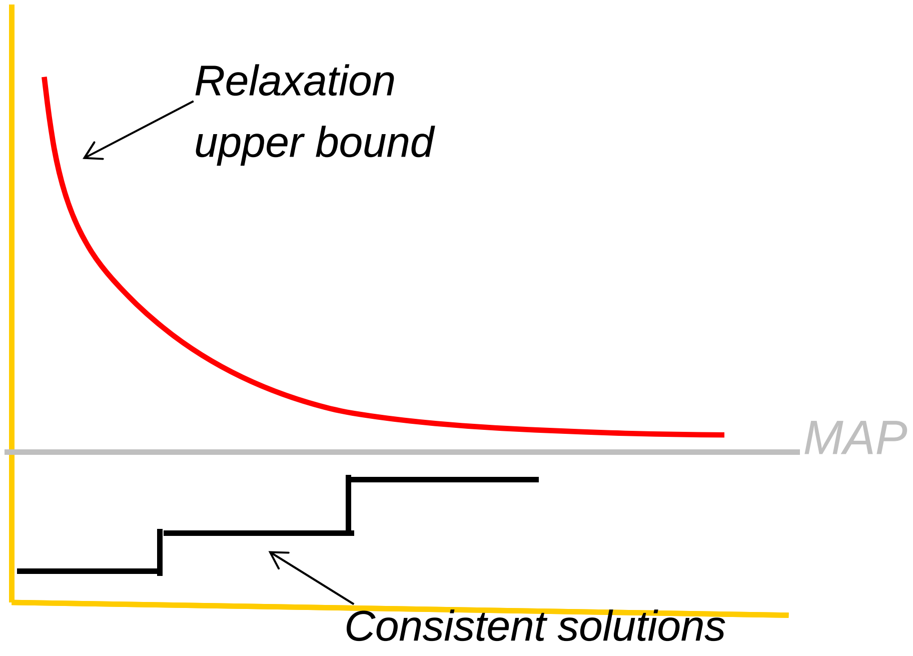
Main collaborators:
Radu Marinescu
Lars Otten
Alex Ihler
Kalev Kask
Robert Mateescu

ISAIM 2016

How to Design a Good Solver

- Heuristic Search
- The core of a good search algorithm
  - A compact search space
  - A good heuristic evaluation function
  - A good traversal strategy
- Anytime search yields a good approximation.

Bounding from Above and Below



For a maximization probelm

Relaxation provides upper bound
Any configuration: lower bound

Outline

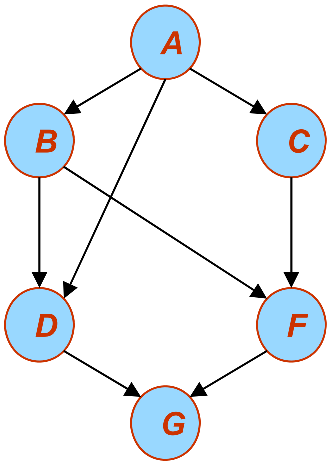
- Graphical models, Queries, Algorithms
- Inference Algorithms: bucket-elimination
- Bounded Inference: a) mini-bucket, b) cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Summary and future work

Outline

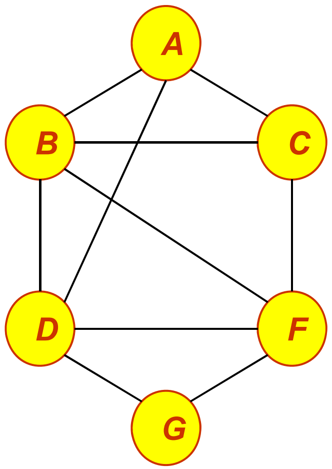
- Graphical models, Queries, Algorithms
- Inference Algorithms: bucket-elimination
- Bounded Inference: a) mini-bucket, b) cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Conclusions

Graphical Models

P(A)
P(B|A)
P(C|A)
P(D|A,B)
P(F|B,C)
P(G|D,F)

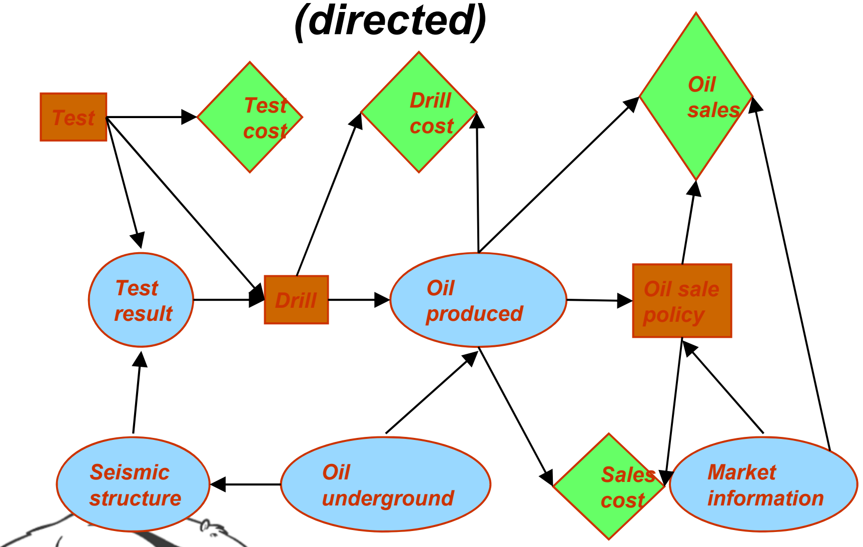


a) Belief network (directed)

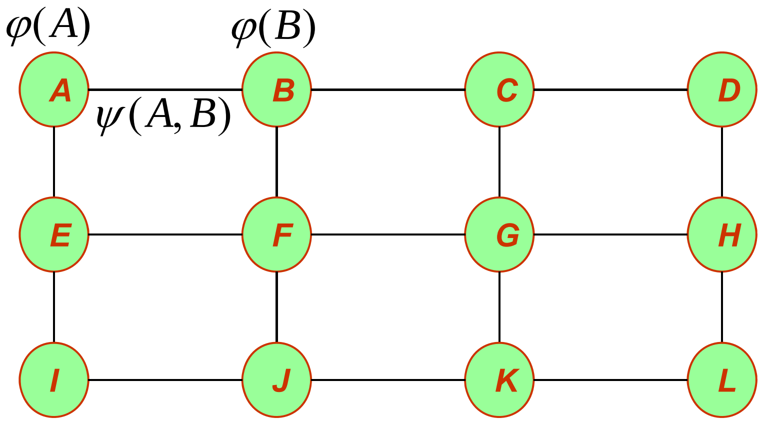


R(A)
R(A,B)
R(A,C)
R(A,B,D)
R(B,C,F)
R(D,F,G)

b) Constraint network (undirected)



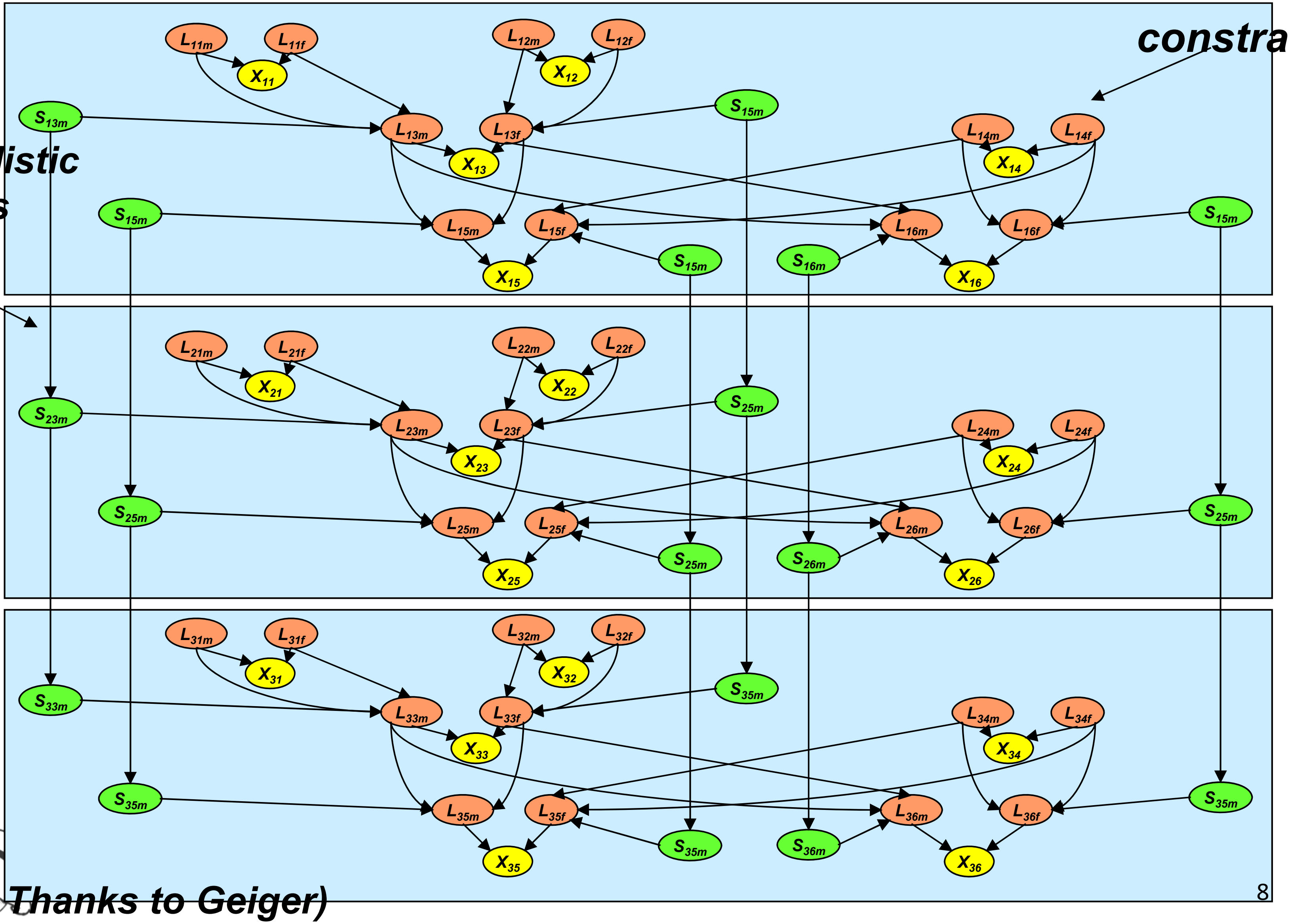
c) Influence diagram



d) Markov network

Many Many applications

Linkage Analysis: Pedigree: 6 People, 3 Markers



constraints

Probabilistic relations

(Slide, Thanks to Geiger)

Graphical Models

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

- Operators:
  - □ combination : Sum, product, join
  - □ Elimination: projection, sum, max/min

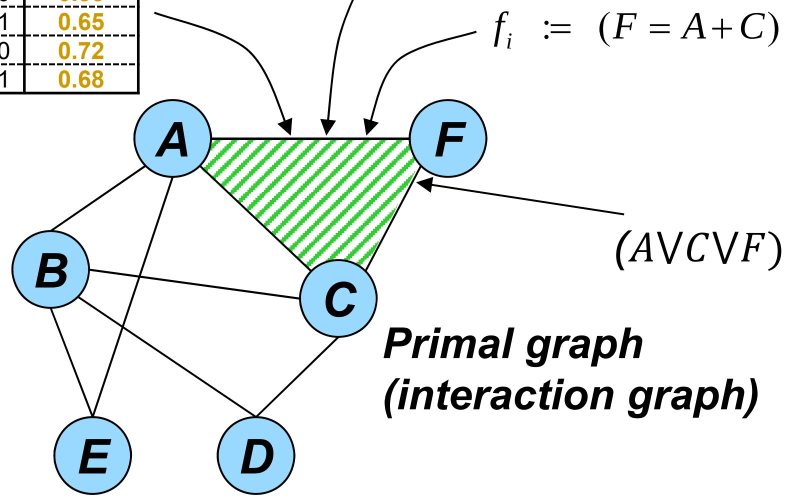
- Tasks:
  - □ **Belief updating**: $\Sigma_{X\backslash Y}\prod_j P_i$
  - □ **MPE\MAP**: $\max_X \prod_j P_j$
  - □ **Marginal MAP**: $\max_Y \sum_{X\backslash Y}\prod_j P_j$
  - □ **CSP**: $\prod_X \times_j C_j$
  - □ **Max-CSP**: $\min_X \Sigma_j F_j$

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

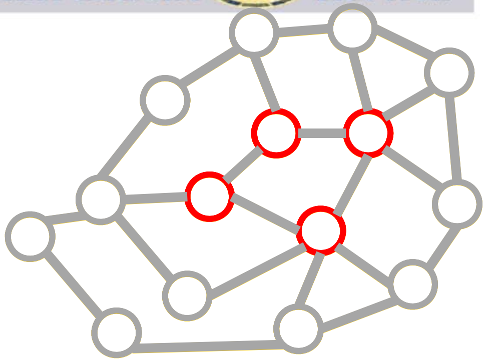
A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue



Primal graph (interaction graph)

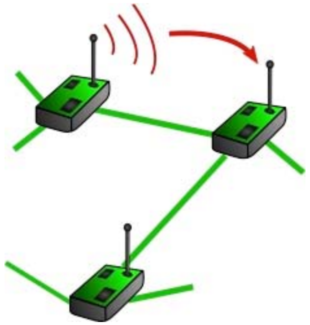
- ***All these tasks are NP-hard***
  - ***exploit problem structure***
  - ***identify special cases***
  - ***approximate***

Why Marginal MAP?

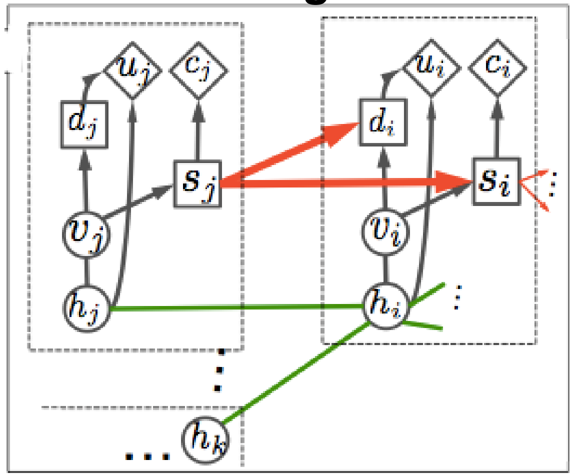


- Often, Marginal MAP is the "right" task:
  - We have a model describing a large system
  - We care about predicting the state of some part
- Example: decision making
  - Sum over random variables (random effects, etc.)
  - Max over decision variables (specify action policies)

Sensor network



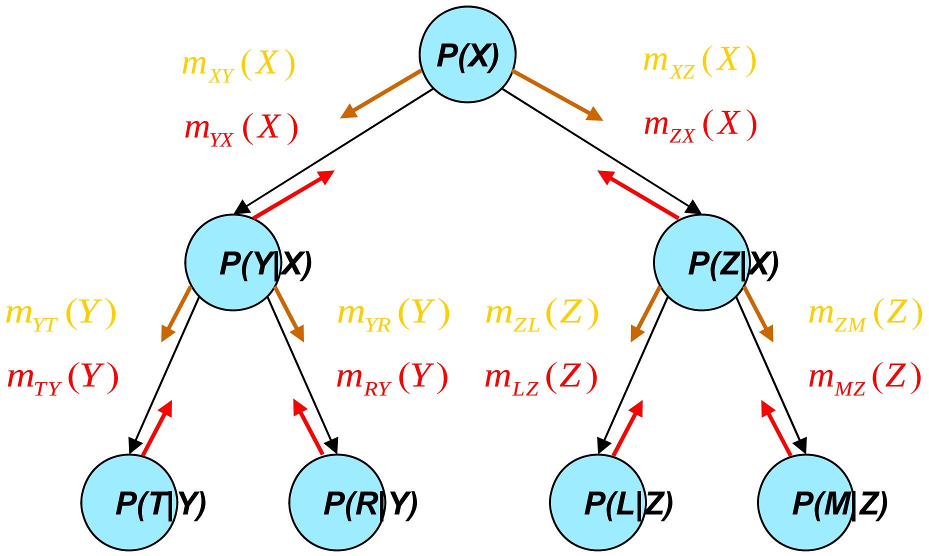
Influence diagram:



Tree-solving is easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



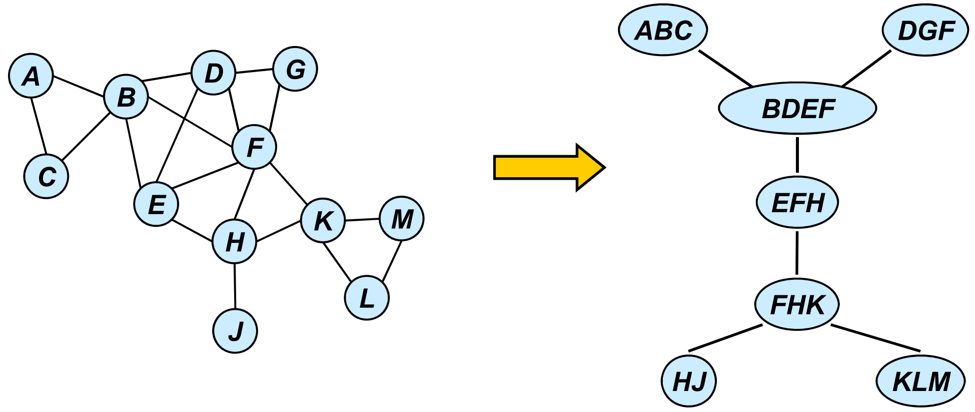
Marginal Map is not Easy even for trees

MPE (max-prod)

#CSP (sum-prod)

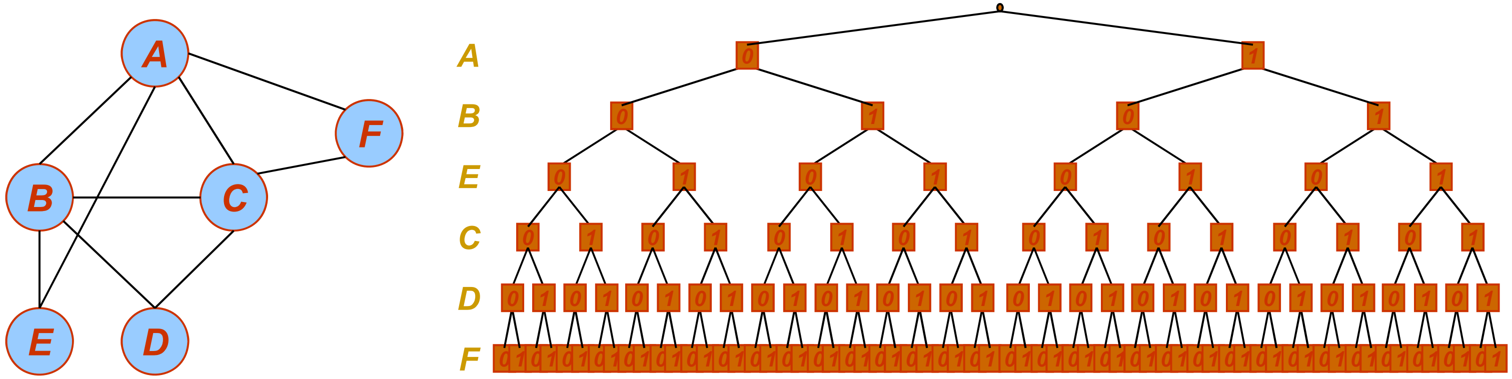
Trees are processed in linear time and memory

Inference vs Conditioning-Search



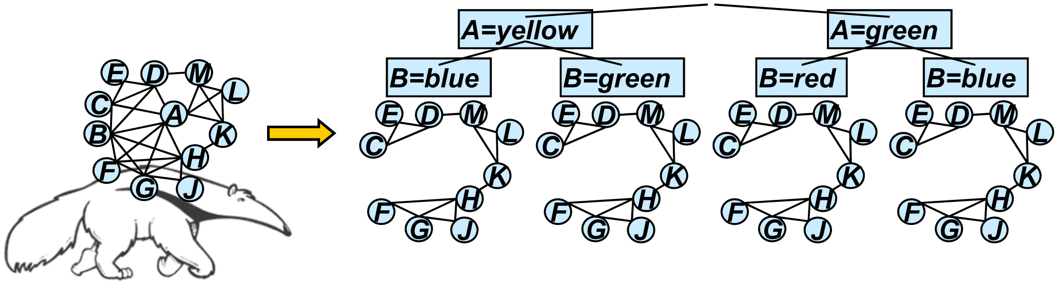
Inference

exp(w) time/space*



Search

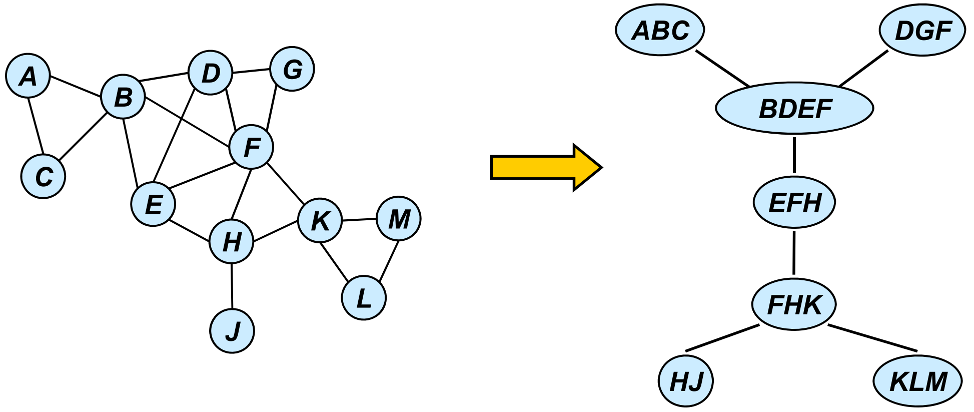
Exp(n) time
O(n) space



Search+inference:
Space: exp(w)
Time: exp(w+c(w))

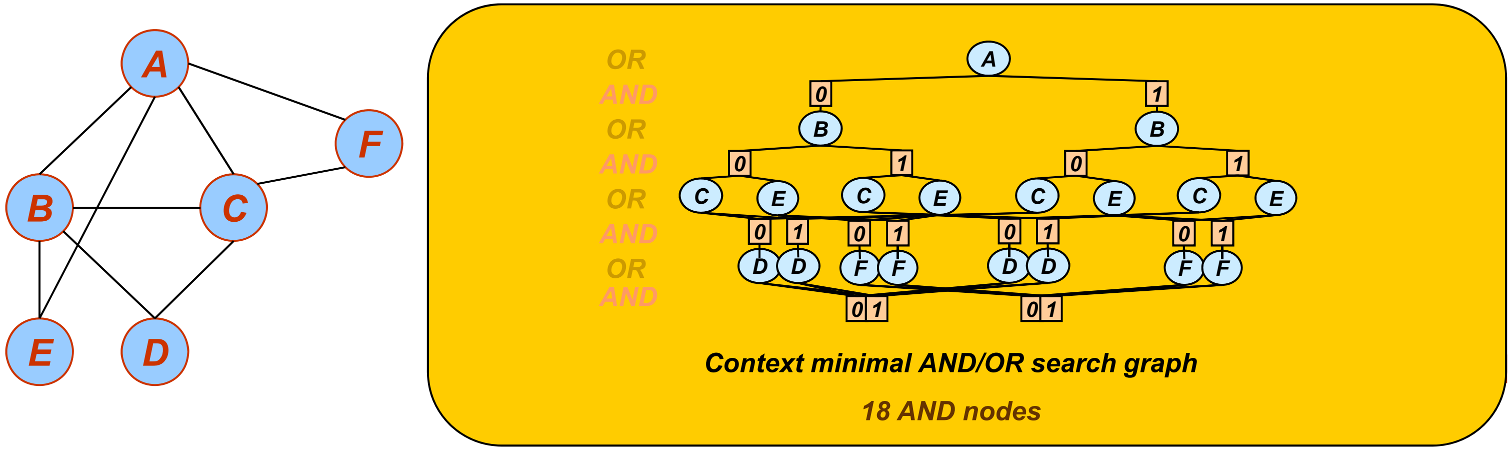
w: user controlled

Inference vs conditioning-search



Inference

exp(w*) time/space

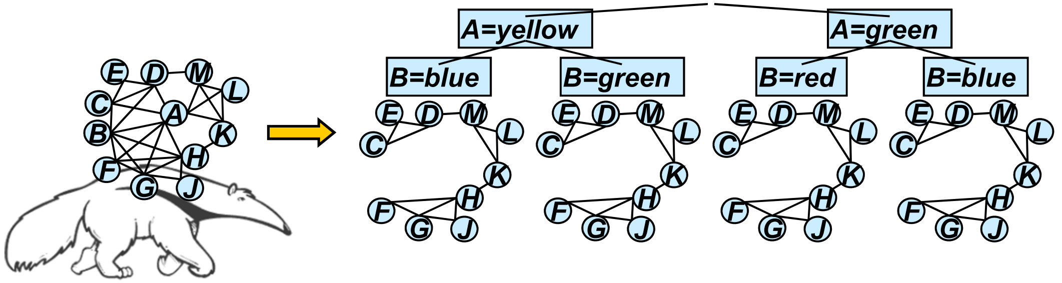


Context minimal AND/OR search graph

18 AND nodes

Search

Exp(w*) time
O(w*) space



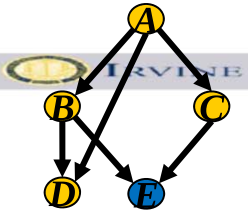
Search+inference:
Space: exp(q)
Time: exp(q+c(q))

q: user controlled

Outline

- Graphical models, Queries, Algorithms
- Inference Algorithms
- Bounded Inference: a) mini-bucket, b) cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Conclusions

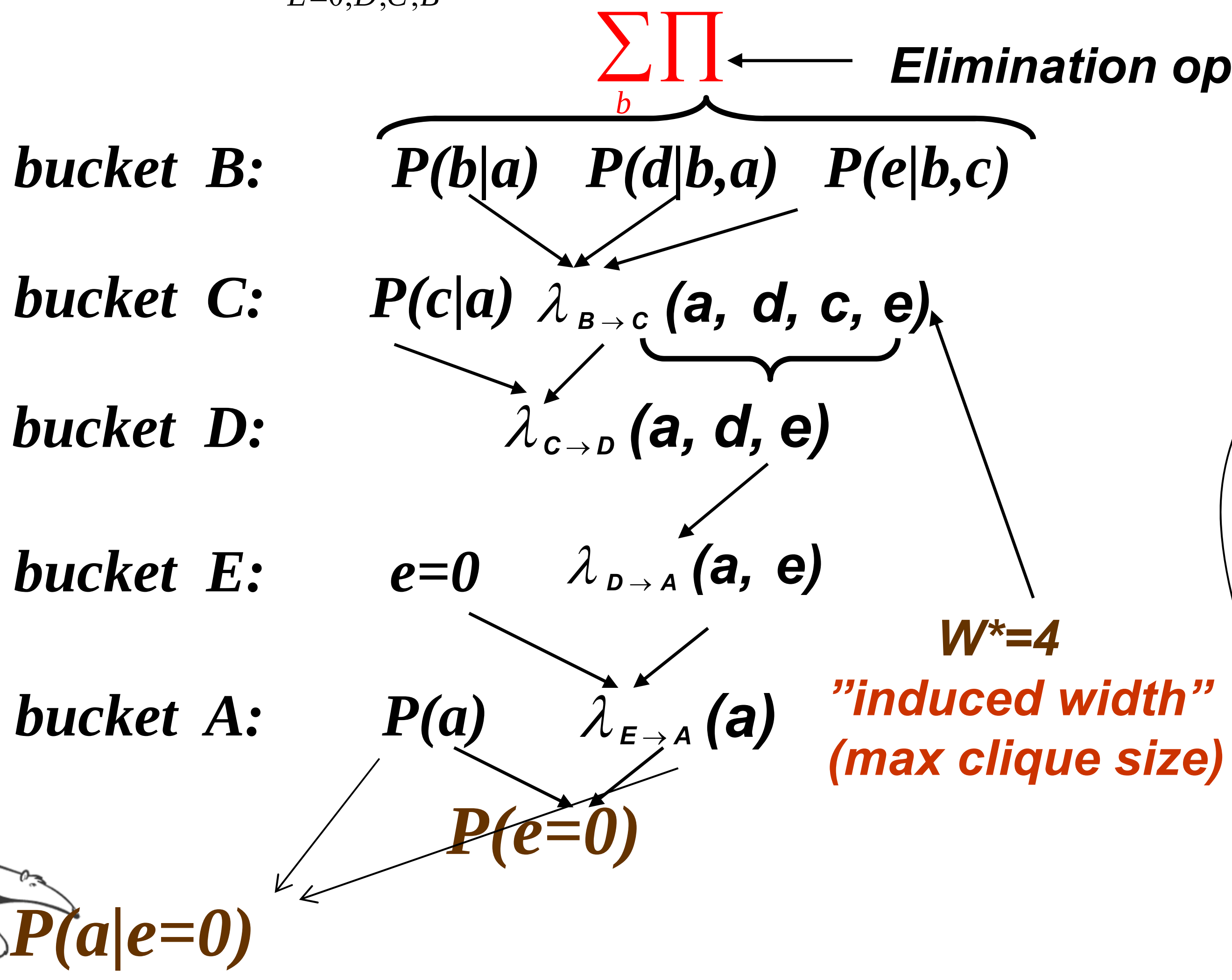
Likelihood queries: Inference (sum-product)



Finding Marginals by Bucket elimination

Algorithm *BE-bel* (Dechter 1996)

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



$\sum_b \prod$ Elimination operator

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

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

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

bucket E: $e=0\quad \lambda_{D\rightarrow A}(a,\ e)$

bucket A: $P(a)\quad \lambda_{E\rightarrow A}(a)$

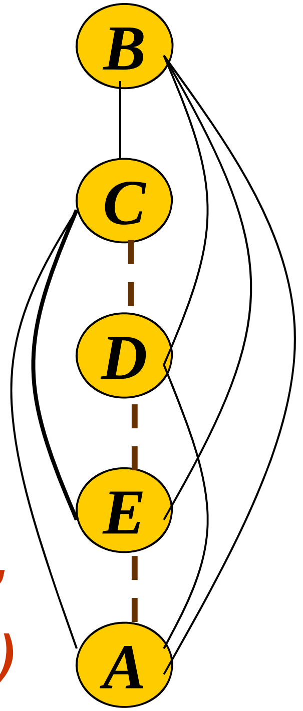
$P(e=0)$

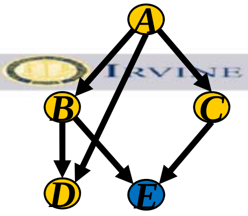
$P(a|e=0)$

W=4*

"induced width"

(max clique size)





Finding Marginals by Bucket elimination

Algorithm *BE-bel* (Dechter 1996)

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

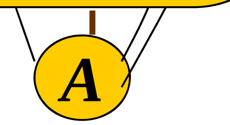
$\sum_b \prod$ ← ***Elimination operator***

Time and space exponential in the induced-width / treewidth

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

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

induced width (max clique size)

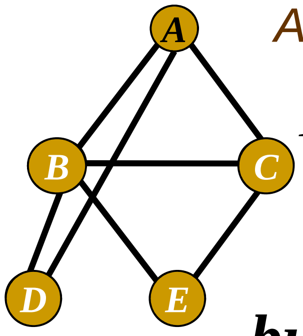


$P(e=0)$

$P(a|e=0)$

Optimization queries: Inference

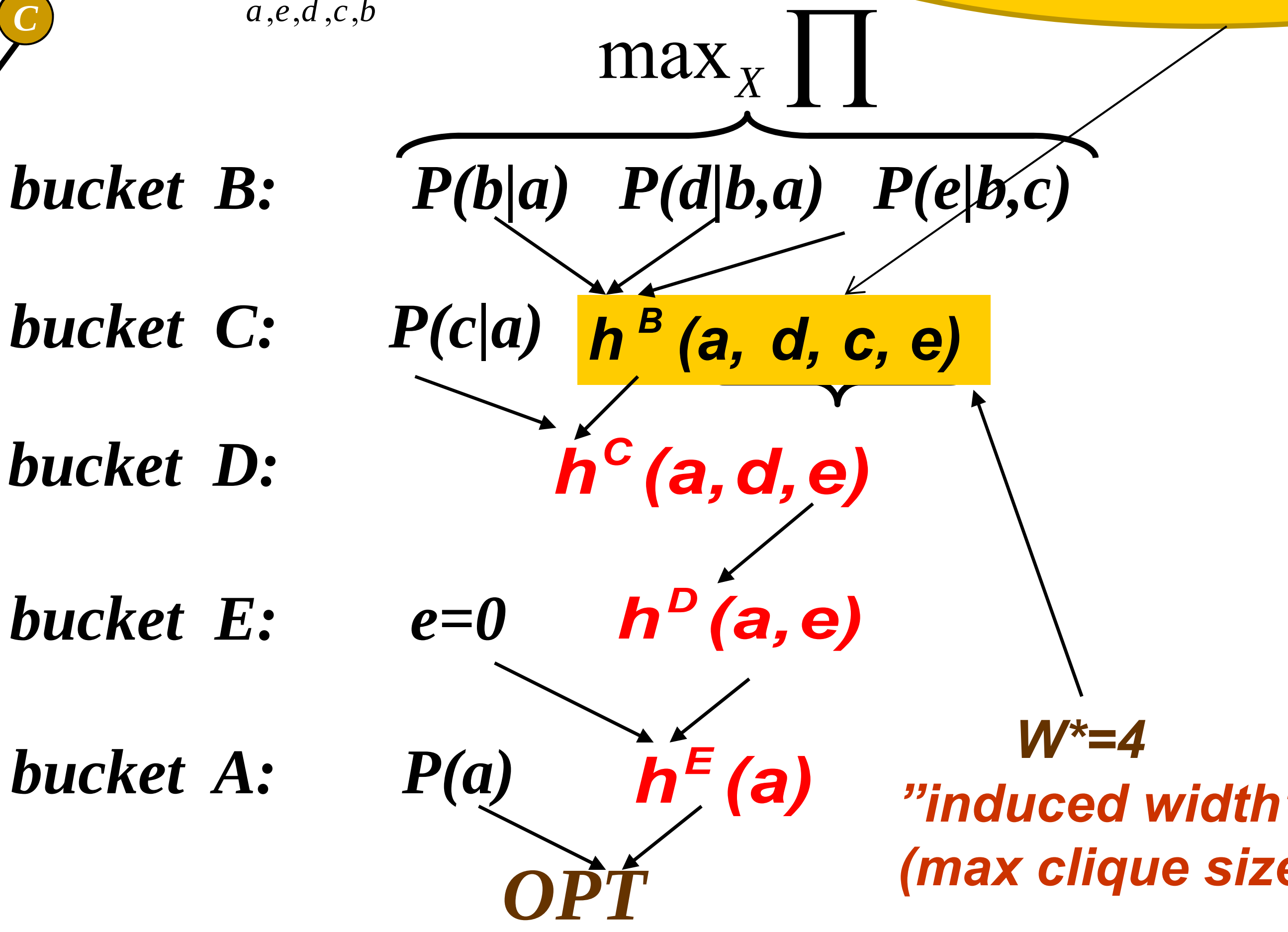
Finding MAP by Bucket Elimination



Algorithm BE-mpe (Dechter 199

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

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



$$\overbrace{\max_X \prod}$$

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

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

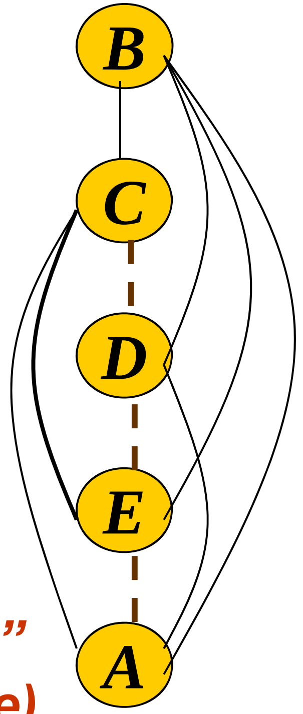
bucket D: $h^C(a,d,e)$

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. $b' = \arg\max_{b} P(b \mid a') \times \times P(d' \mid b, a') \times P(e' \mid b, c')$

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

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: $h^{C}(a, d, e)$

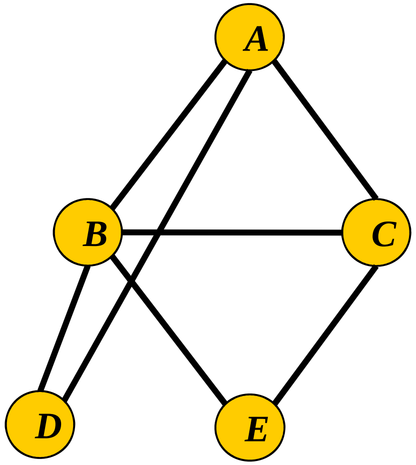
E: $e=0$ $h^{D}(a, e)$

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

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

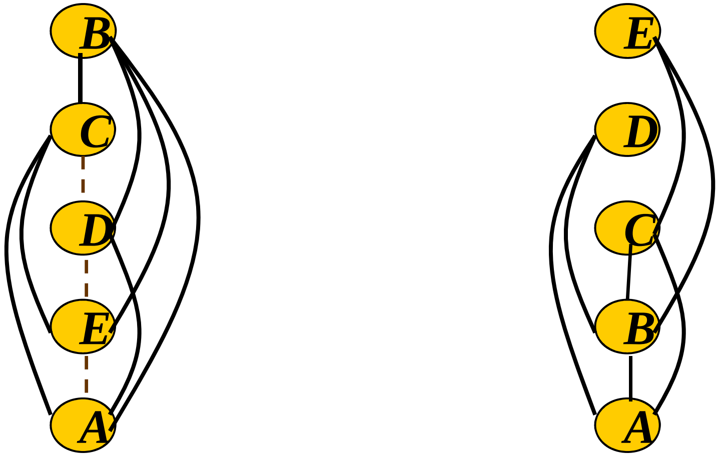
The Induced-Width/Treewidth

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



"Moral" graph

The effect of the ordering:

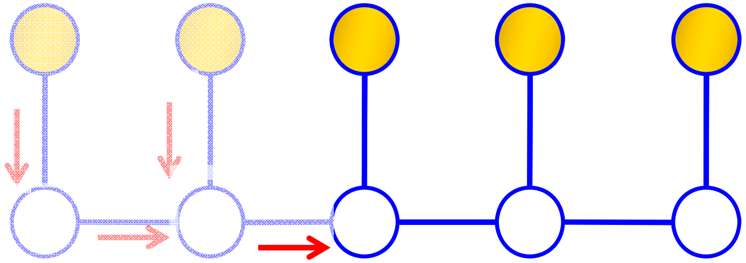


$w^*(d_1) = 4$ $w^*(d_2) = 2$

Marginal Map: Inference

Variable Elimination (max-sum-product)

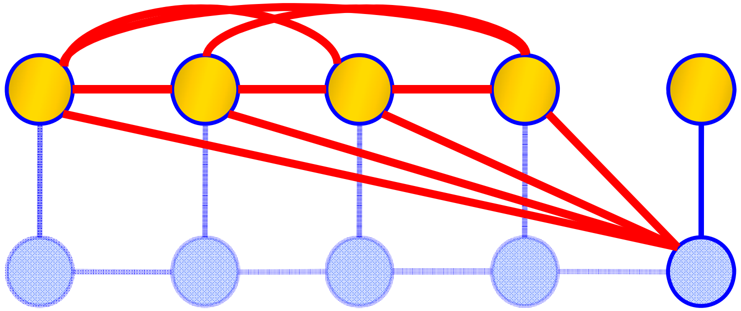
- Pure MAP or summation tasks
  - □ Dynamic programming
  - □ Ex: efficient on trees



- Marginal MAP
  - □ Operations do not commute:

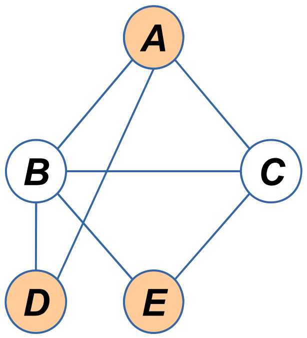
$$\sum \max \neq \max \sum$$

  - □ Sum must be done first!



Bucket Elimination for MMAP

Bucket Elimination

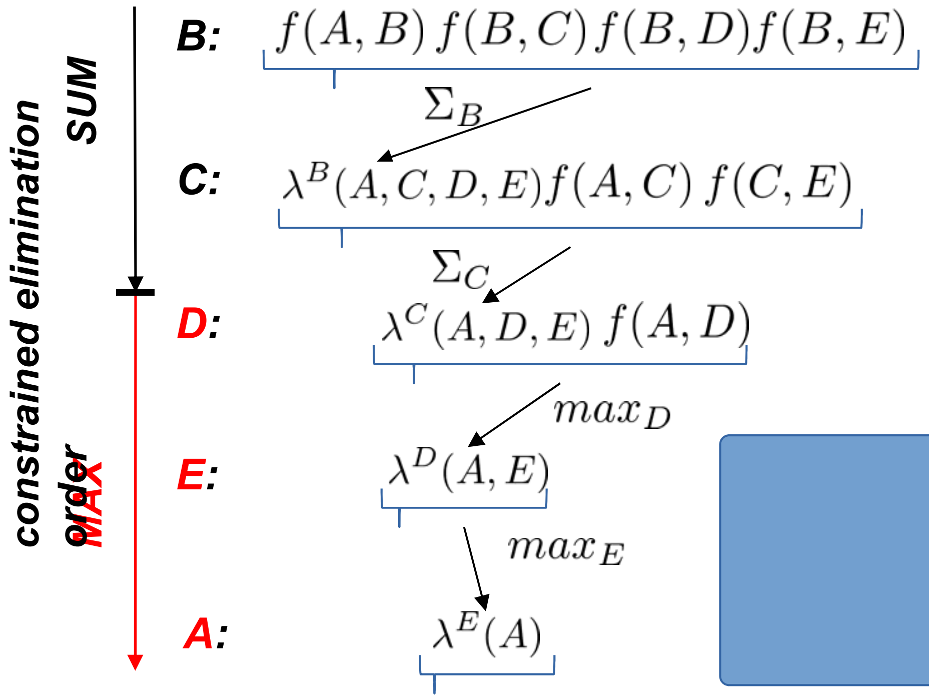


$$\mathbf{X}_M = \{A, D, E\}$$

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

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

[Dechter, 1999]



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

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

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

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

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

MAP* is the marginal MAP value

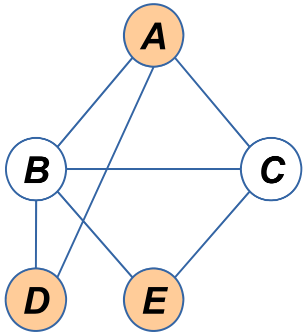
Bucket Elimination for MMAP

Bucket Elimination

$$\max_X \sum_Y \phi \leq \sum_Y \max_X \phi$$

exact

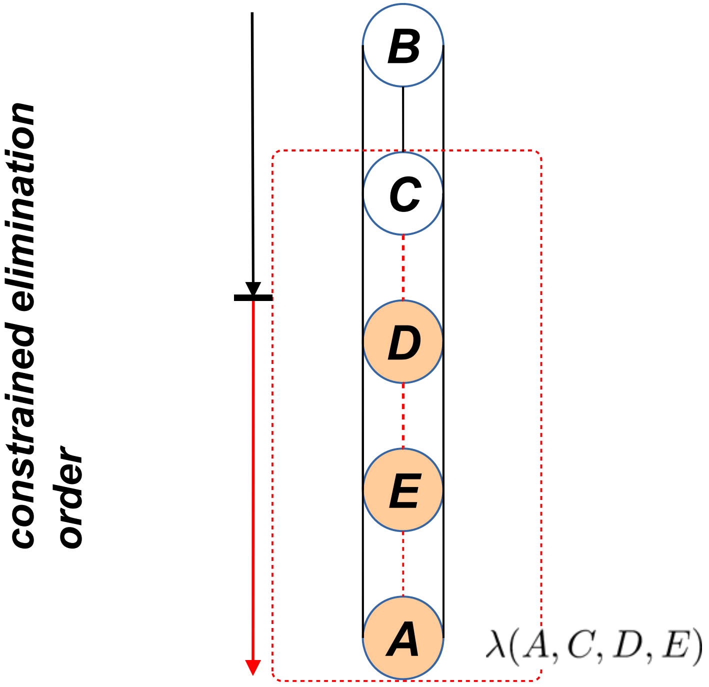
upper bound



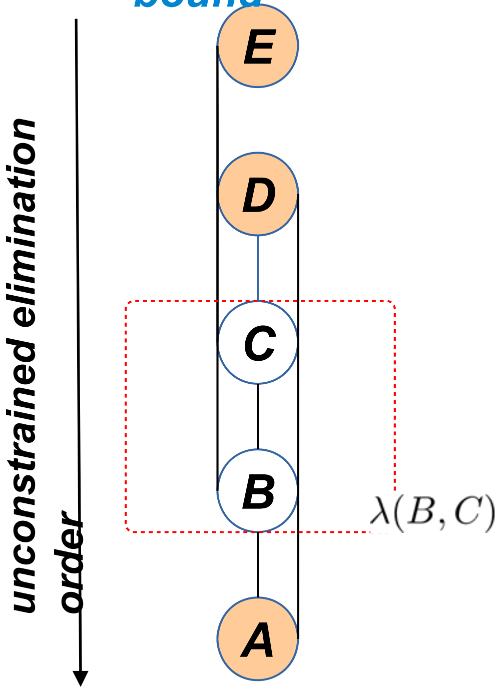
$$\mathbf{X}_M = \{A, D, E\}$$
$$\mathbf{X}_S = \{B, C\}$$

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

[Dechter, 1999]



$w^* = 4$



$w^* = 2$

In practice, constrained induced is much larger!

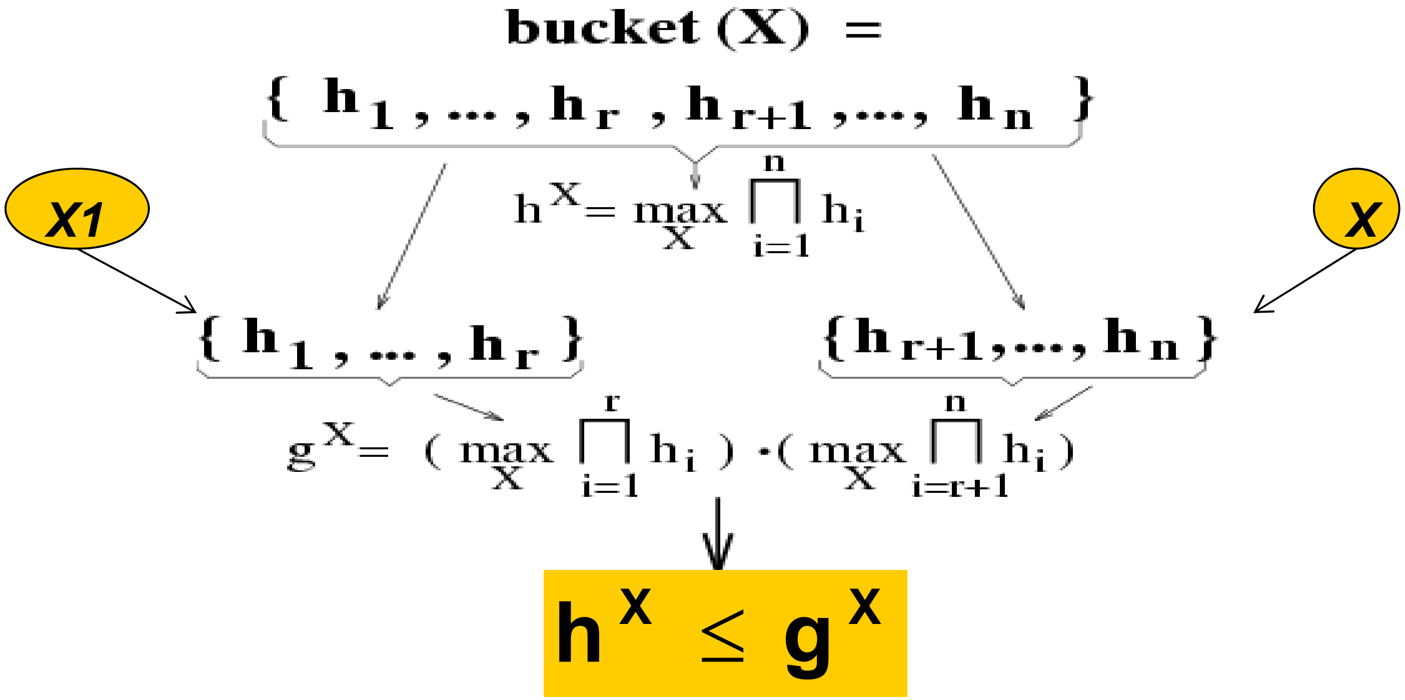
Outline

- Graphical models, Queries, Algorithms
- Inference Algorithms: bucket-elimination
- Bounded Inference: a) mini-bucket, b) cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Conclusions

Mini-bucket Approximation: Relaxation

(Dechter and Rish, 1997, 2003)

Split a bucket into mini-buckets =>bound complexity



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

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

X1

X

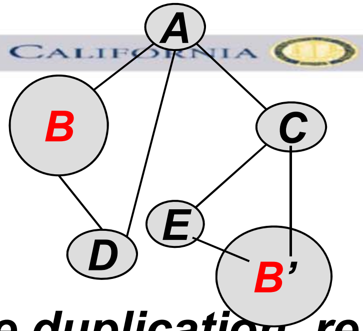
$$\underbrace{\{ h_1, \ldots, h_r \}} \qquad \underbrace{\{ h_{r+1}, \ldots, h_n \}}$$

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

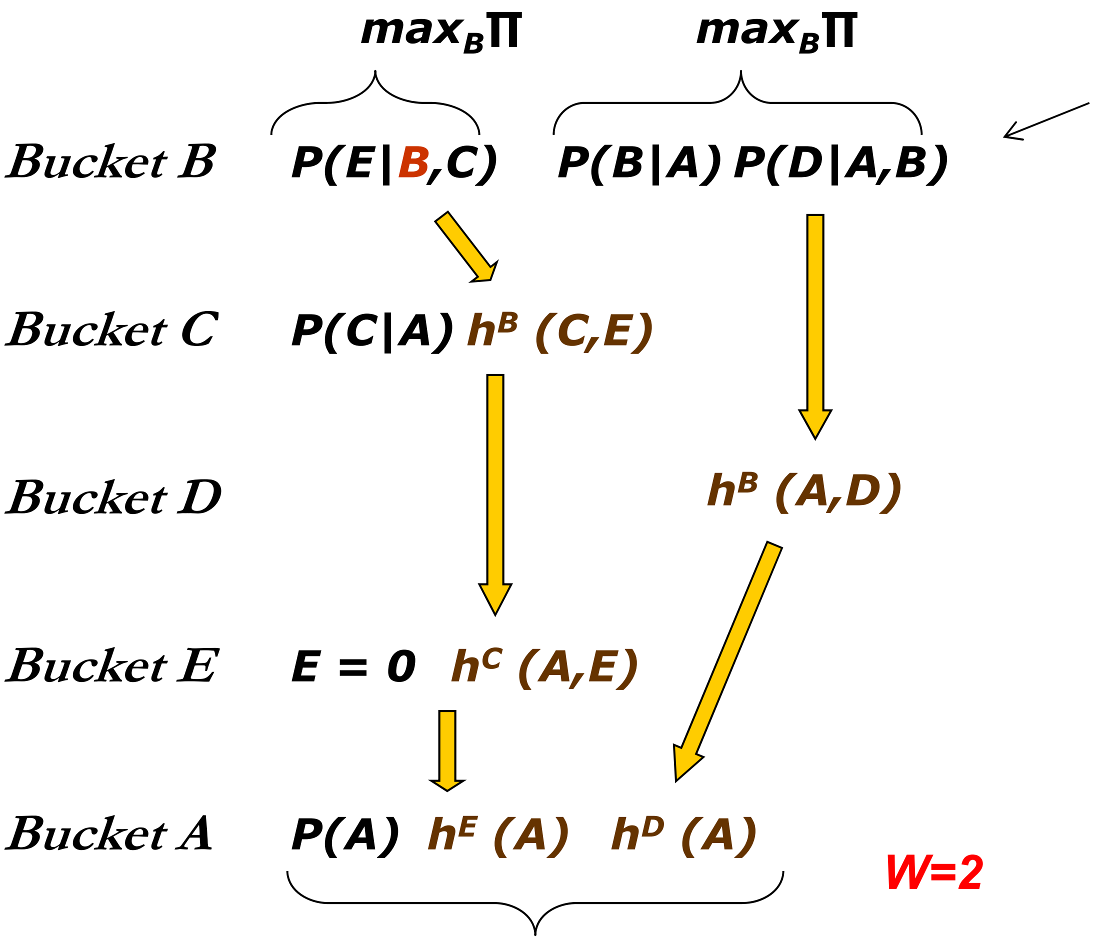
$$h^X \le g^X$$

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

Mini-Bucket Elimination



Node duplication, renaming



$\max_B \Pi$ $\max_B \Pi$

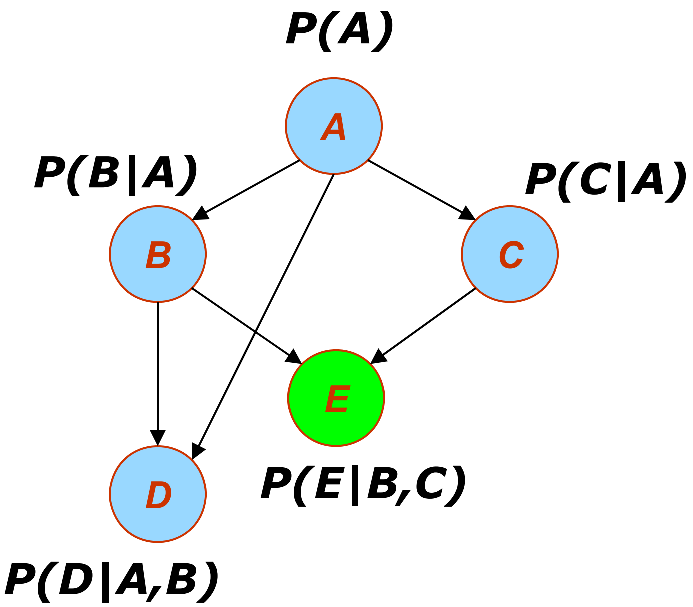
Bucket B **P(E|B,C) P(B|A) P(D|A,B)**

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

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

Mini-Bucket Decoding

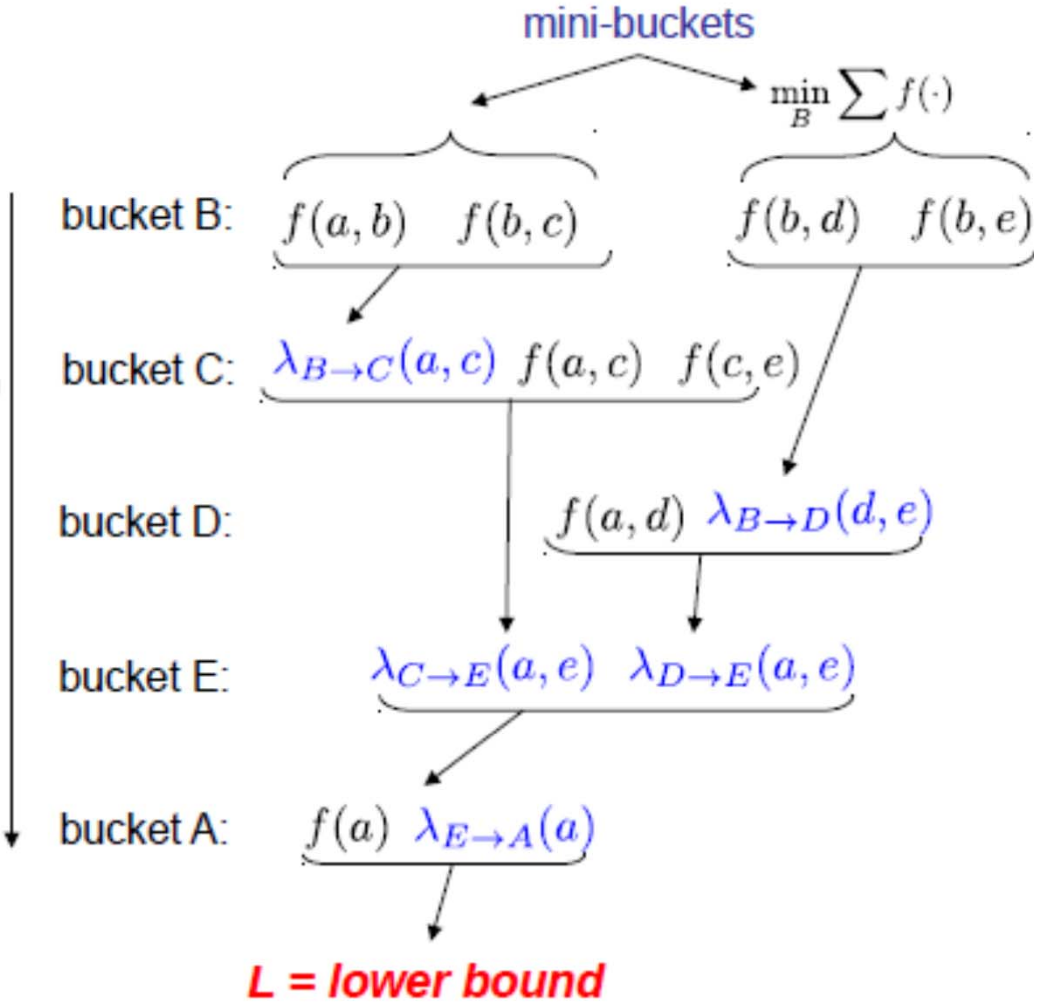
$$\hat{\mathbf{b}} = \arg\min_{\mathbf{b}} f(\hat{a},b) + f(b,\hat{c}) + f(b,\hat{d}) + f(b,\hat{e})$$

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}} \lambda_{B\to C}(\hat{a},c) + f(c,\hat{a}) + f(c,\hat{e})$$

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{d}} f(\hat{a},d) + \lambda_{B\to D}(d,\hat{e})$$

$$\hat{\mathbf{e}} = \arg\min_{\mathbf{e}} \lambda_{C\to E}(\hat{a},e) + \lambda_{D\to E}(\hat{a},e)$$

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} f(a) + \lambda_{E\to A}(a)$$



mini-buckets

$\min_B \sum f(\cdot)$

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

bucket C: $\lambda_{B\to C}(a,c)$ $f(a,c)$ $f(c,e)$

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

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

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

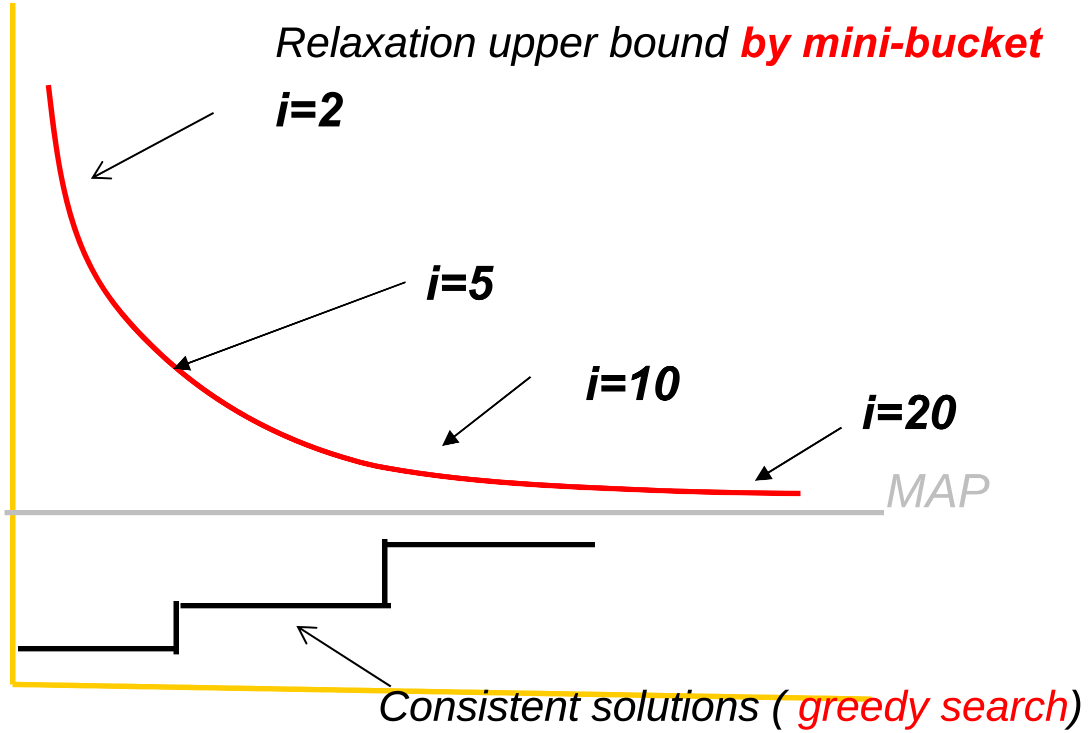
L = lower bound

Greedy configuration = upper bound

Properties of Mini-Bucket Eliminaton

- **Complexity:** *O(r exp(i))* time and *O(exp(i)) space*.
- **Accuracy:** determined by Upper/Lower bound.
- As ***i*** increases, both accuracy and complexity increase.
- Possible use of mini-bucket approximations:
  - As anytime algorithms
  - As heuristics in search

Bounding from above and below


Relaxation upper bound by mini-bucket
i=2
i=5
i=10
i=20
MAP
Consistent solutions (greedy search)

Relaxation provides upper bound
Any configuration: lower bound

Outline

- Graphical models, Queries, Algorithms
- Inference Algorithms: bucket-elimination
- Bounded Inference: a) mini-bucket, b) cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Conclusions

Cost-Shifting

(Reparameterization)

$+\lambda(B)$

A	B	f(A,B)
b	b	6 + 3
b	g	0 - 1
g	b	0 + 3
g	g	6 - 1

+

$-\lambda(B)$

B	C	f(B,C)
b	b	6 - 3
b	g	0 - 3
g	b	0 + 1
g	g	6 + 1

A	B	C	f(A,B,C)
b	b	b	12
b	b	g	6
b	g	b	0
b	g	g	6
g	b	b	6
g	b	g	0
g	g	b	6
g	g	g	12

= 0 + 6

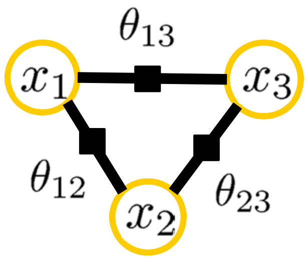
Modify the individual functions

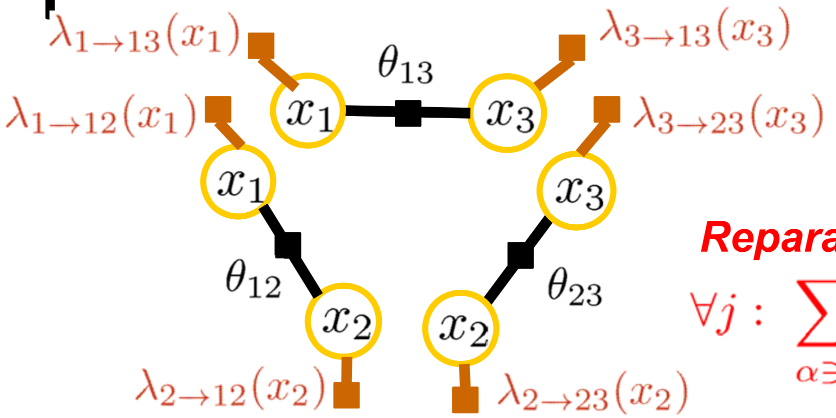
But

keep the sum of functions unchanged

B	$\lambda(B)$
b	3
g	-1

Ex: Dual Decomposition





Reparameterization:

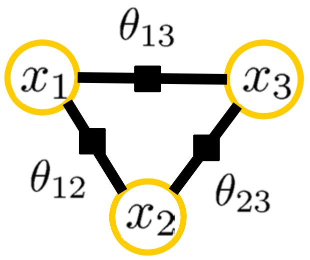
$$\forall j : \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0$$

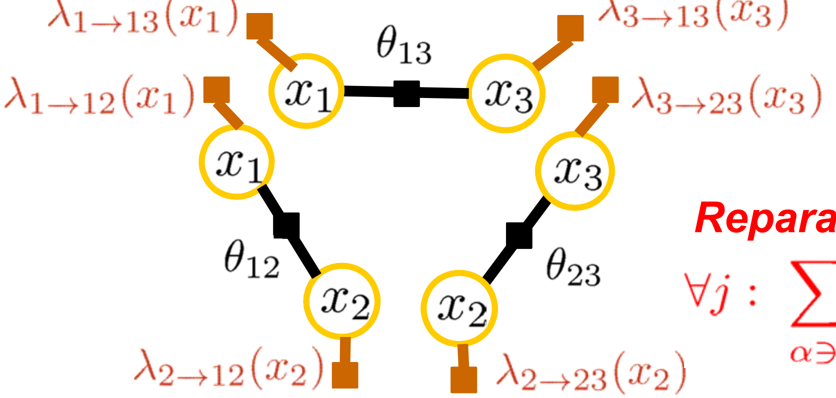
$$\log f(\mathbf{x}^*) = \max_{\mathbf{x}} \sum_{\alpha} \theta_\alpha(\mathbf{x}_\alpha) \quad \le \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \left[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \right]$$

- Bound solution using decomposed optimization
- Solve independently: optimistic bound
- Exact if all copies agree
- Tighten the bound by reparameterization
  - Enforces lost equality constraints using Lagrange multipliers

Add factors that "adjust" each local term, but cancel out in total

Ex: Dual Decomposition





Reparameterization:

$$\forall j : \sum_{\alpha \ni j} \lambda_{j\to\alpha}(x_j) = 0$$

$$\log f(\mathbf{x}^*) = \max_{\mathbf{x}} \sum_{\alpha} \theta_\alpha(\mathbf{x}_\alpha) \quad \leq \min_{\{\lambda_{i\to\alpha}\}} \sum_{\alpha} \max_{\mathbf{x}_\alpha} \left[\theta_\alpha(\mathbf{x}_\alpha) + \sum_{i\in\alpha} \lambda_{i\to\alpha}(x_i) \right]$$

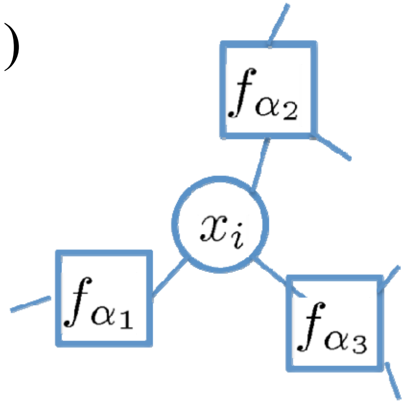
Many names for the same class of bounds:

- Dual decomposition [Komodakis et al. 2007]
- TRW, MPLP [Wainwright et al. 2005, Globerson & Jaakkola 2007]
- Soft arc consistency [Cooper & Schiex 2004]
- Max-sum diffusion [Warner 2007]

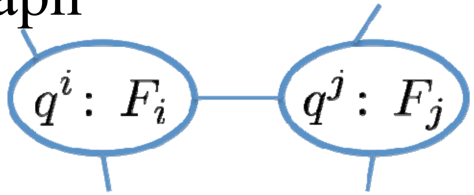
Various Update Schemes

- Can use any decomposition updates
  - (message passing, subgradient, augmented, etc.)

- FGLP: Update the original factors



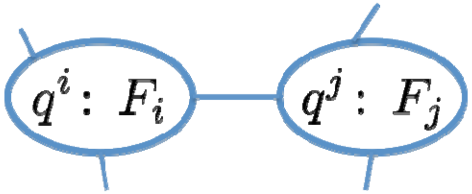
- JGLP: Update clique function of the join graph



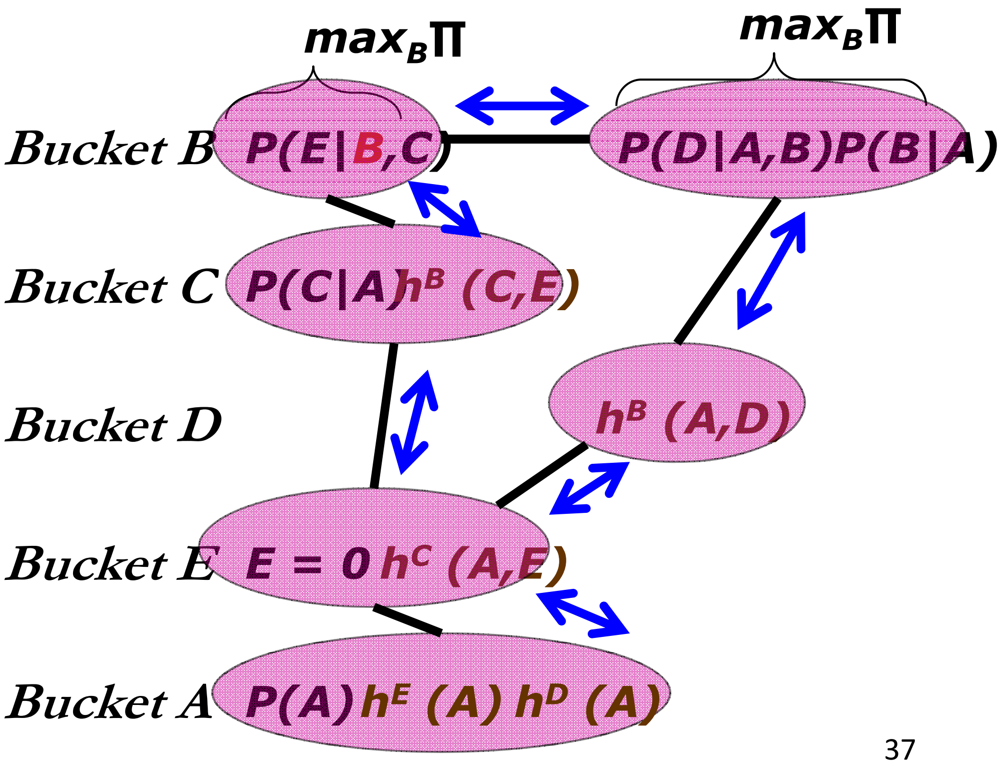
- MBE-MM Update within each bucket only

JGLP: Fixed-Point Updates

- **JGLP:** Update clique function of the join graph
  - □ Use MBE to generate the join graph
  - □ Define function F_i for each clique (mini-bucket) q^i
  - □ Update each edge over separator set

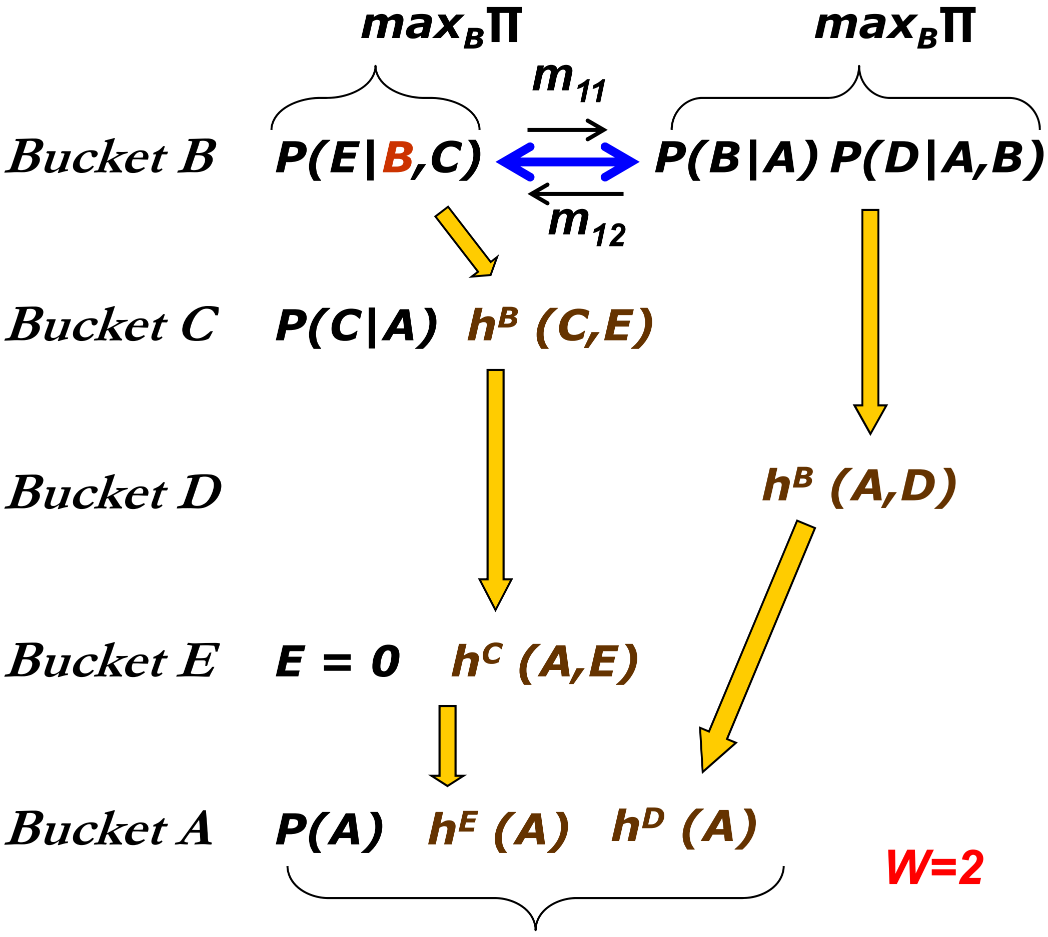


$$F_i \leftarrow F_i + \frac{1}{2}(\gamma_j(x_s) - \gamma_i(x_s))$$



MBE-MM: MBE with Moment Matching

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



$\max_B \prod$ $\max_B \prod$

m_{11} m_{12}

Bucket B $\underbrace{P(E|B,C)}$ $\longleftrightarrow$ $\underbrace{P(B|A)\ P(D|A,B)}$

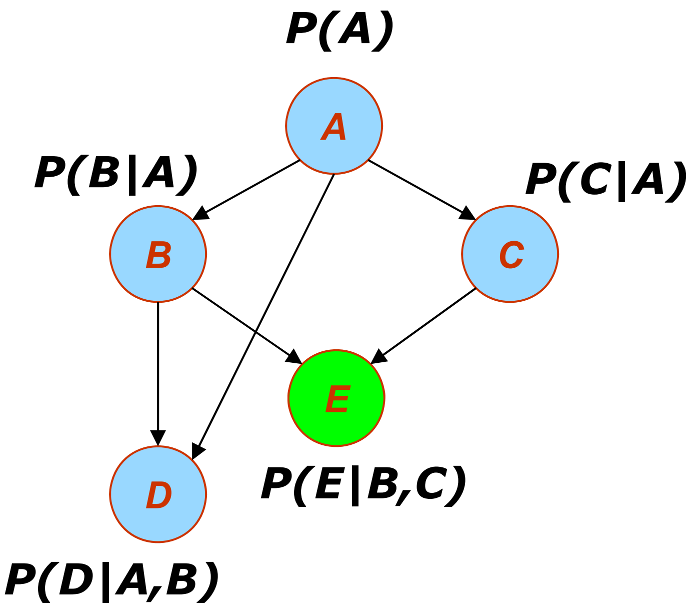
Bucket C $P(C|A)$ $h^B(C,E)$

Bucket D $h^B(A,D)$

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

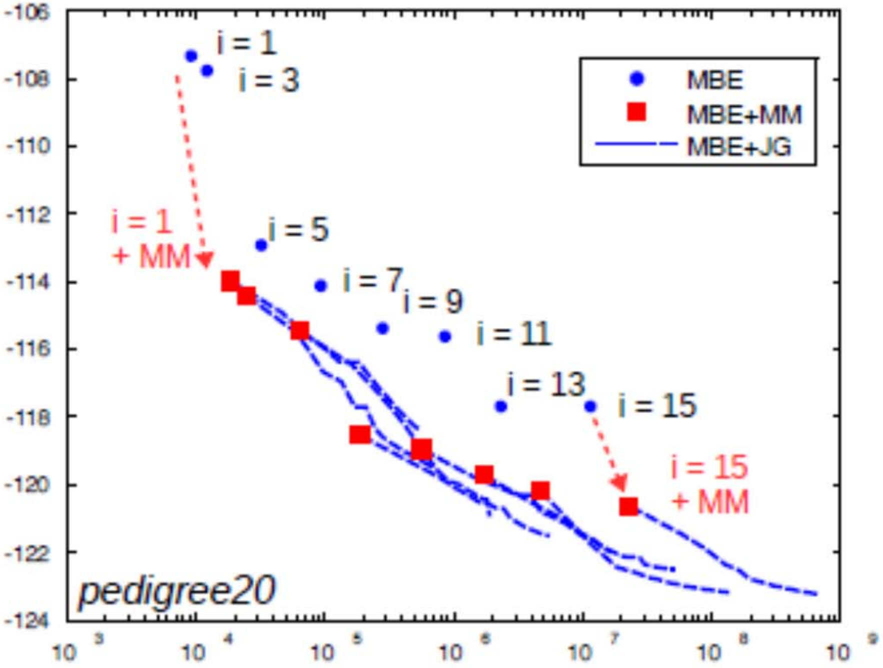
Bucket A $\underbrace{P(A)\ h^E(A)\ h^D(A)}$

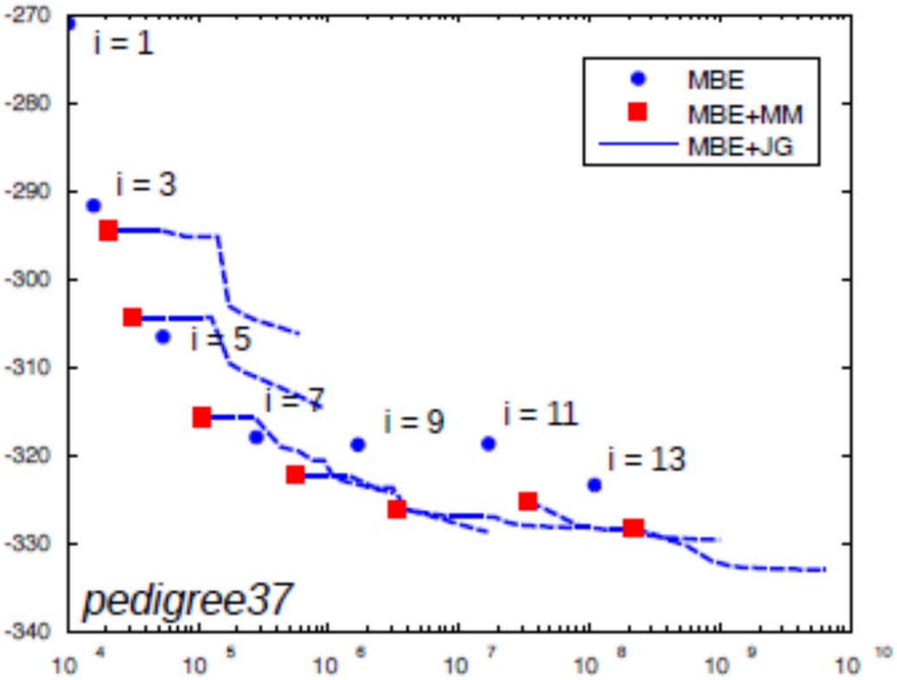
W=2



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

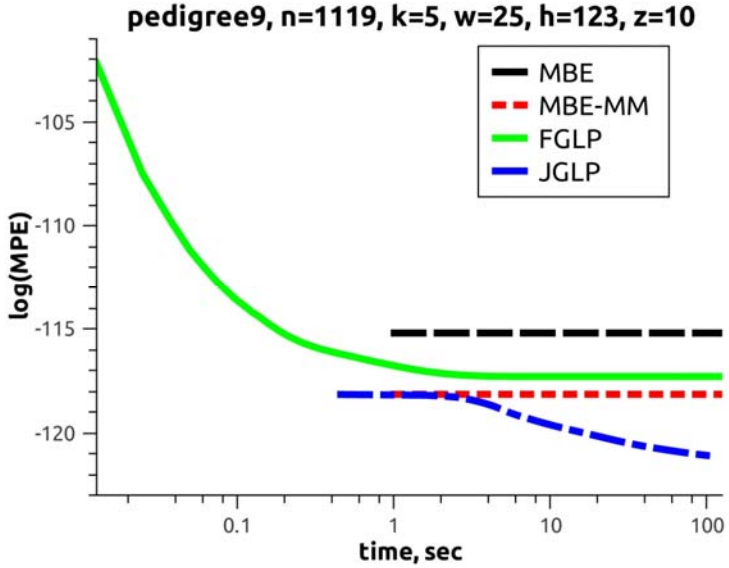
Anytime Approximation

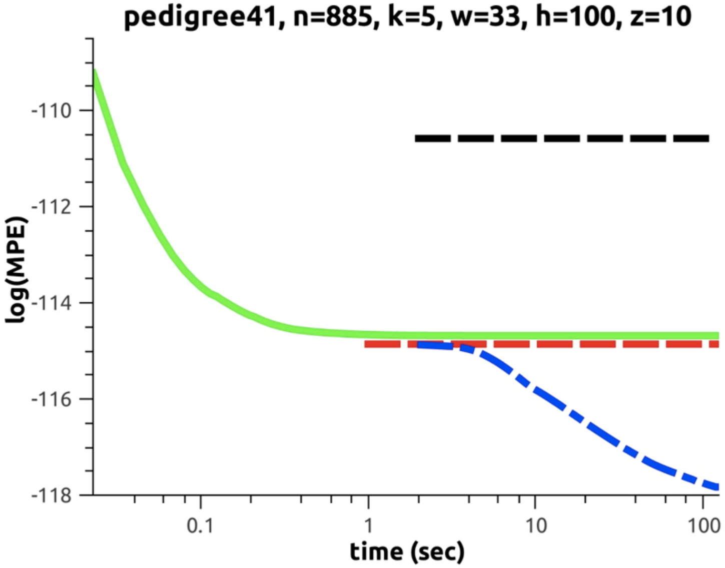


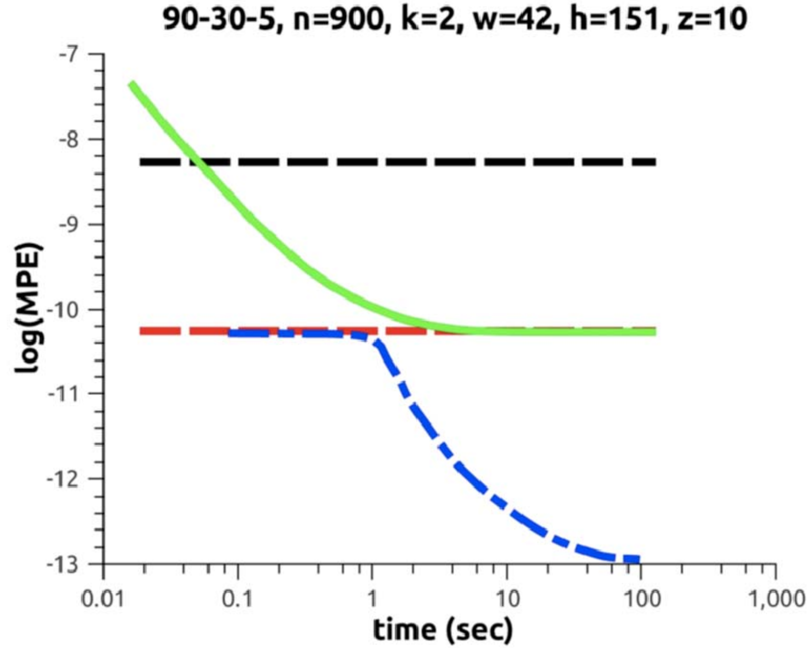


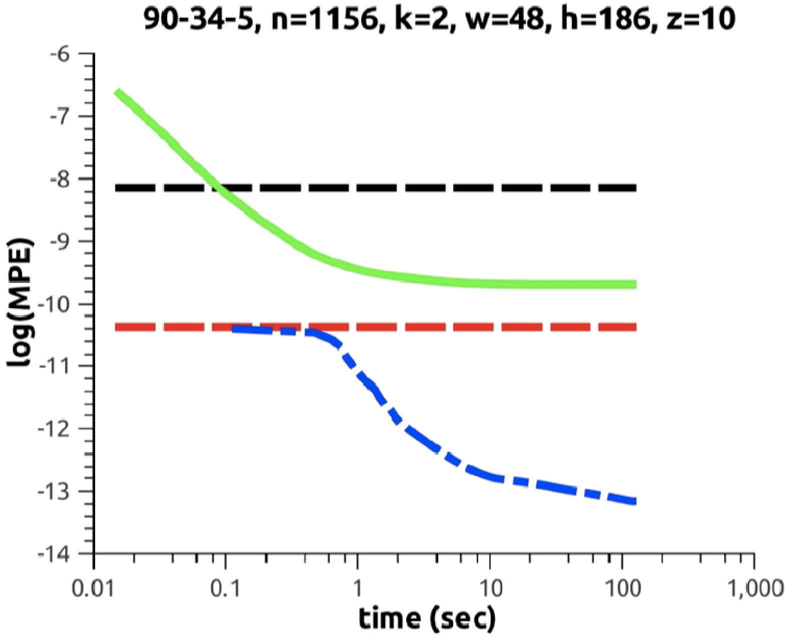
- Can tighten the bound in various ways
  - Cost-shifting (improve consistency between cliques)
  - Increase i-bound (higher order consistency)
- Simple moment-matching step improves bound significantly

Iterative Tightening as Bounding Schemes


pedigree9, n=1119, k=5, w=25, h=123, z=10


pedigree41, n=885, k=5, w=33, h=100, z=10


90-30-5, n=900, k=2, w=42, h=151, z=10

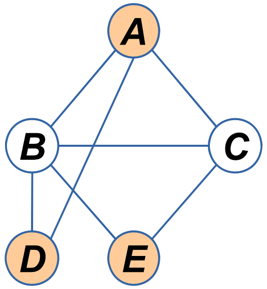

90-34-5, n=1156, k=2, w=48, h=186, z=10

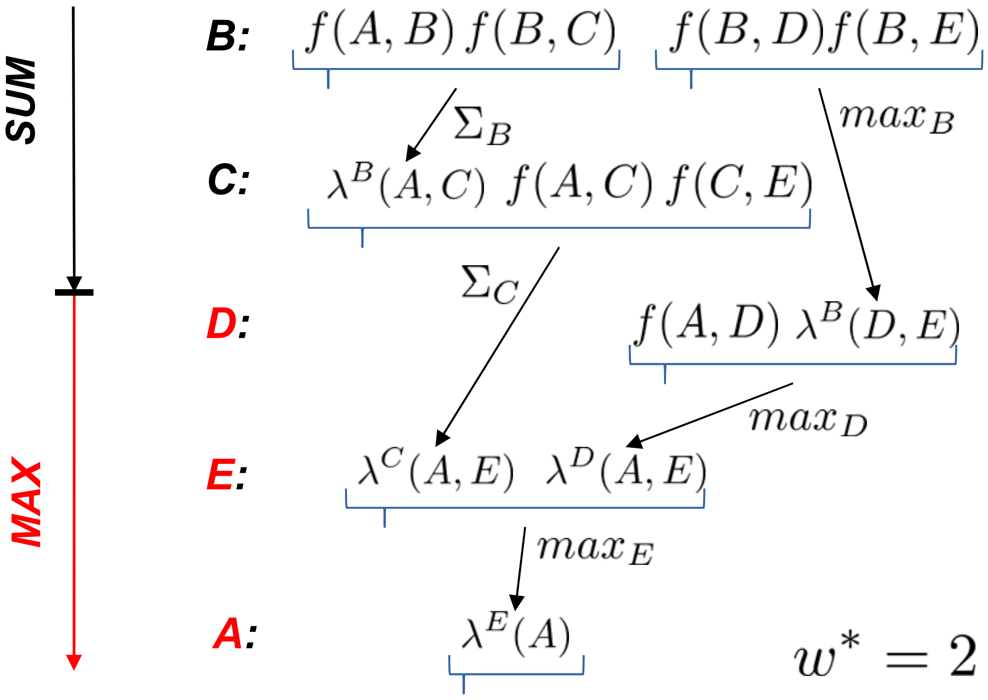
Bounded Inference for sum-product and max-sum-oproduct Inference:

Mini-bucket and weighted mini-bucket

Marginal (sum) and Marginal MAP (max-sum-product)

Partition a bucket into mini-buckets with i variables





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

Σ_B max_B

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

Σ_C

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

max_D

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

max_E

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

$$w^* = 2$$

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

[Dechter and Rish, 2001]

MAP* is an upper bound on the marginal MAP value

Weighted Mini-Bucket

(for summation bounds)

Exact bucket elimination:

$$\lambda_B(a,c,d,e) = \sum_b \left[\, f(a,b)\cdot f(b,c)\cdot f(b,d)\cdot f(b,e) \,\right]$$

$$\leq \Big[\sum_b^{w_1} f(a,b)f(b,c)\Big] \cdot \Big[\sum_b^{w_2} f(b,d)f(b,e)\Big]$$

$$= \lambda_{B\to C}(a,c) \qquad \cdot\, \lambda_{B\to D}(d,e)$$

(mini-buckets)

where $\sum_x^{w} f(x) = \left[\sum_x f(x)^{1/w}\right]^w$

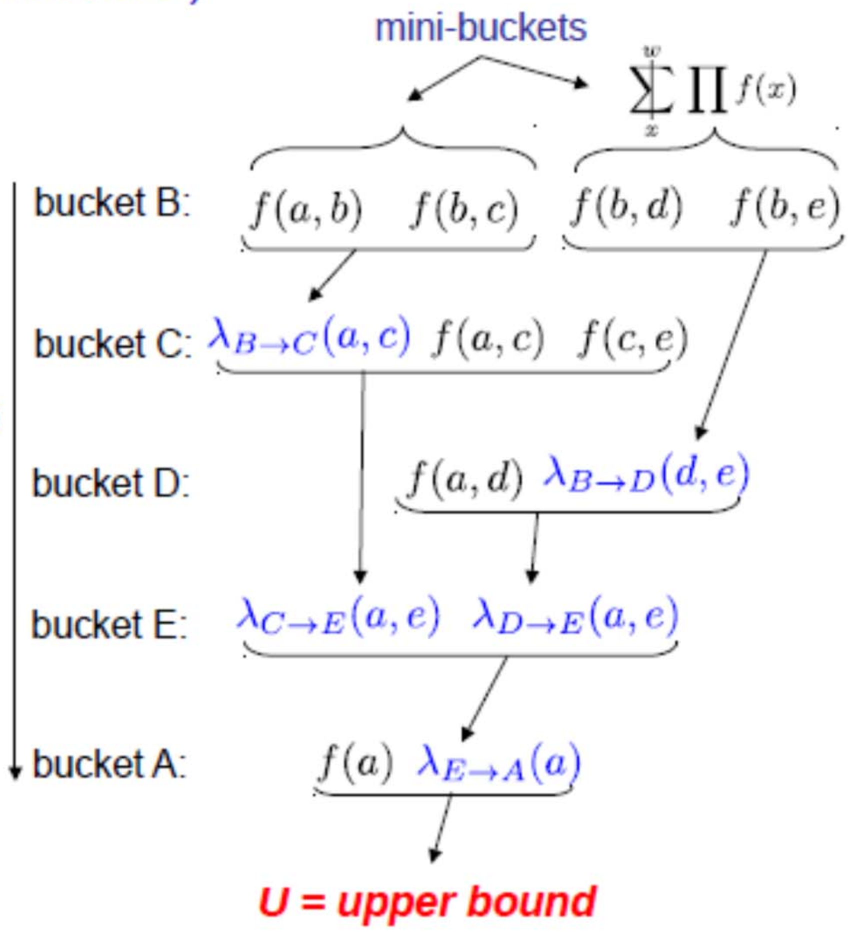
is the weighted or "power" sum operator

By Holder's inequality,

$$\sum_x^{w} f_1(x)f_2(x) \leq \left[\sum_x^{w_1} f_1(x)\right]\left[\sum_x^{w_2} f_2(x)\right]$$

where $w_1 + w_2 = w$ and $w_1 > 0,\ w_2 > 0$

(lower bound if $w_1 > 0,\ w_2 < 0$)



mini-buckets

$\sum_x^{w} \prod f(x)$

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

bucket C: $\lambda_{B\to C}(a,c)$ $f(a,c)$ $f(c,e)$

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

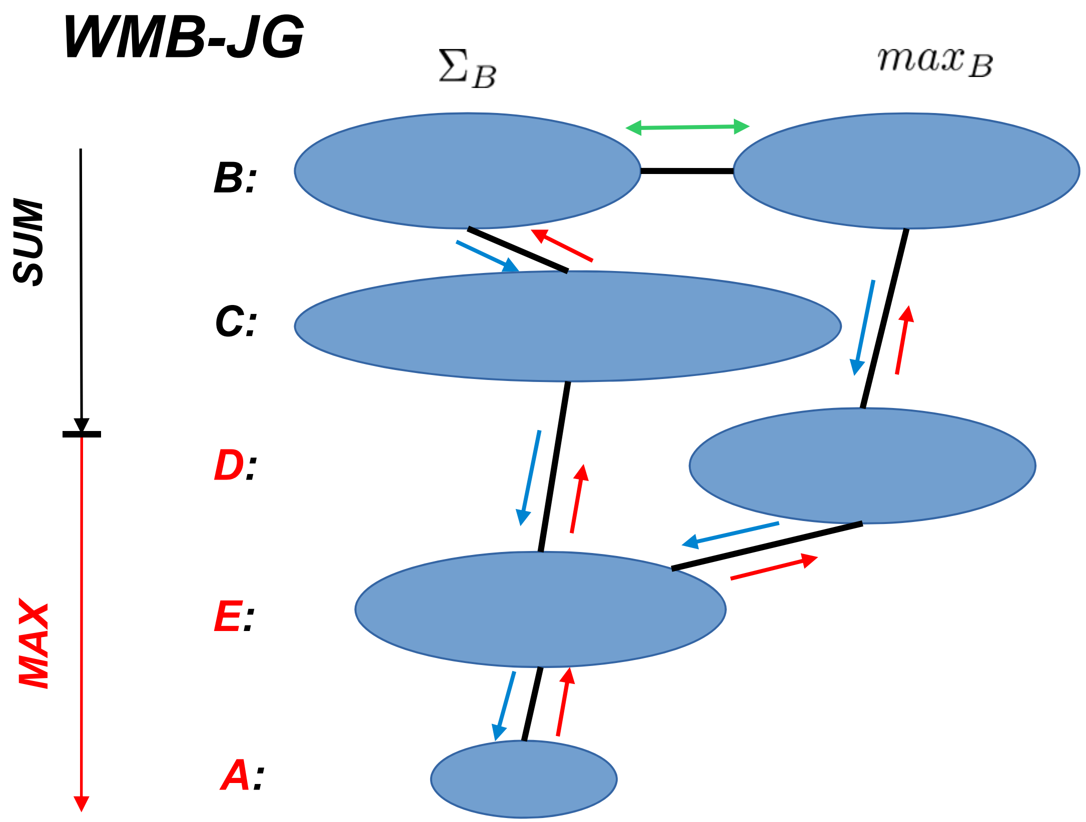
bucket E: $\lambda_{C\to E}(a,e)$ $\lambda_{D\to E}(a,e)$

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

U = upper bound

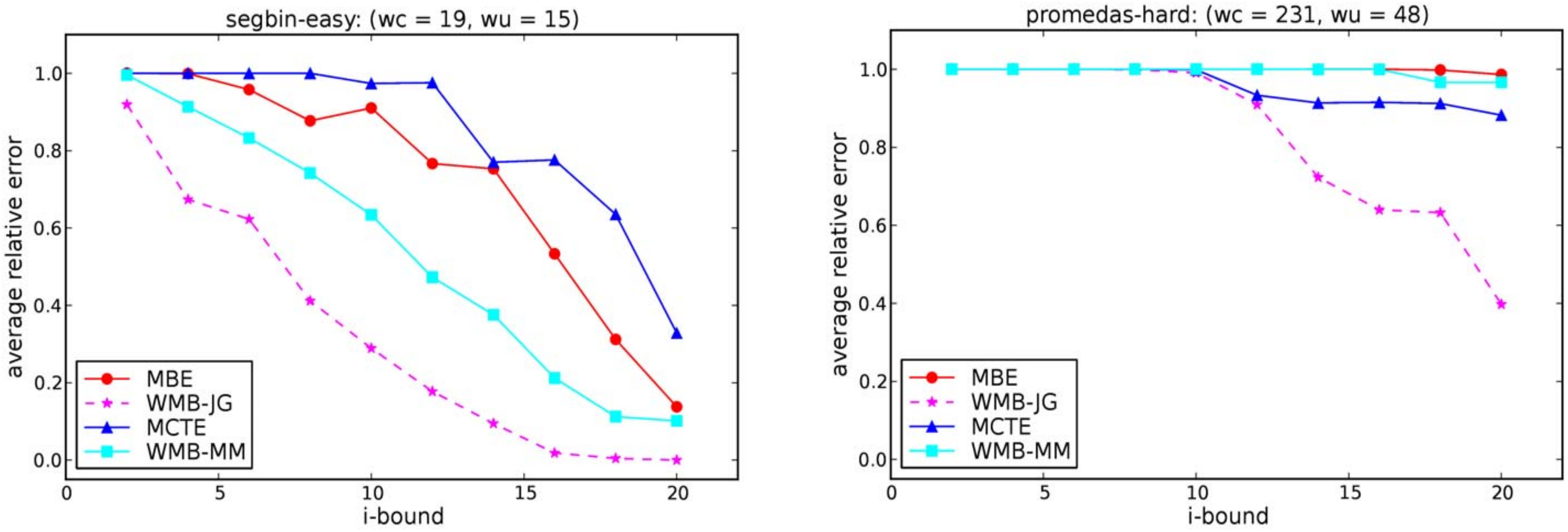
[Liu & Ihler 2011]

WMB on Join-Graphs



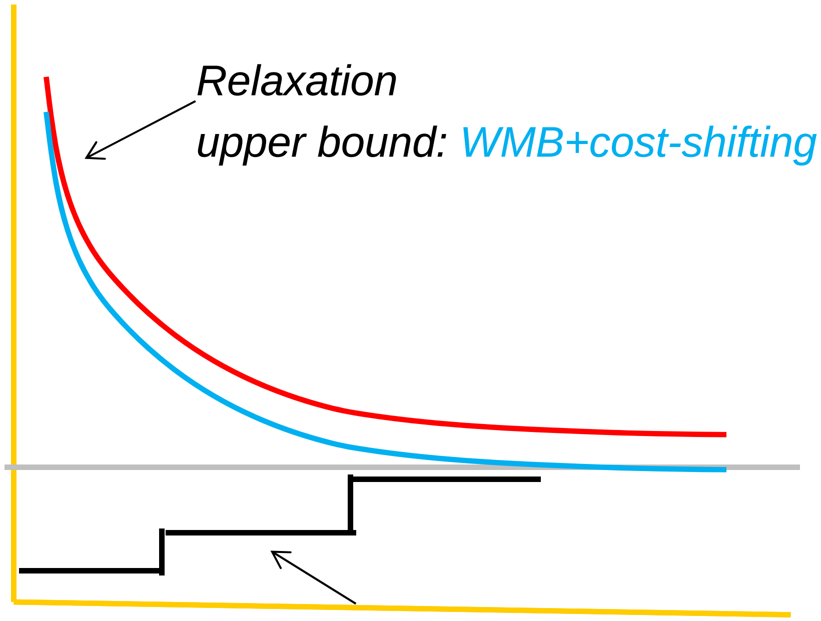
- ***WMB defines a join-graph***
- ***Propagate downward / upward messages until convergence***
- ***Downward messages***
  - ***e.g. from B to D***
- ***Upward messages***
  - ***e.g. from D to B***
  - ***used during cost-shifting***
- ***Cost-shifting within a bucket***
- ***In practice, yields a much tighter upper bound than WMB***

MMAP: Quality of the Upper Bounds



Average relative error wrt tightest upper bound. 10 iterations for WMB-JG(i).

Bounding from Above and Below



Relaxation
upper bound: WMB+cost-shifting

Marginal MAP

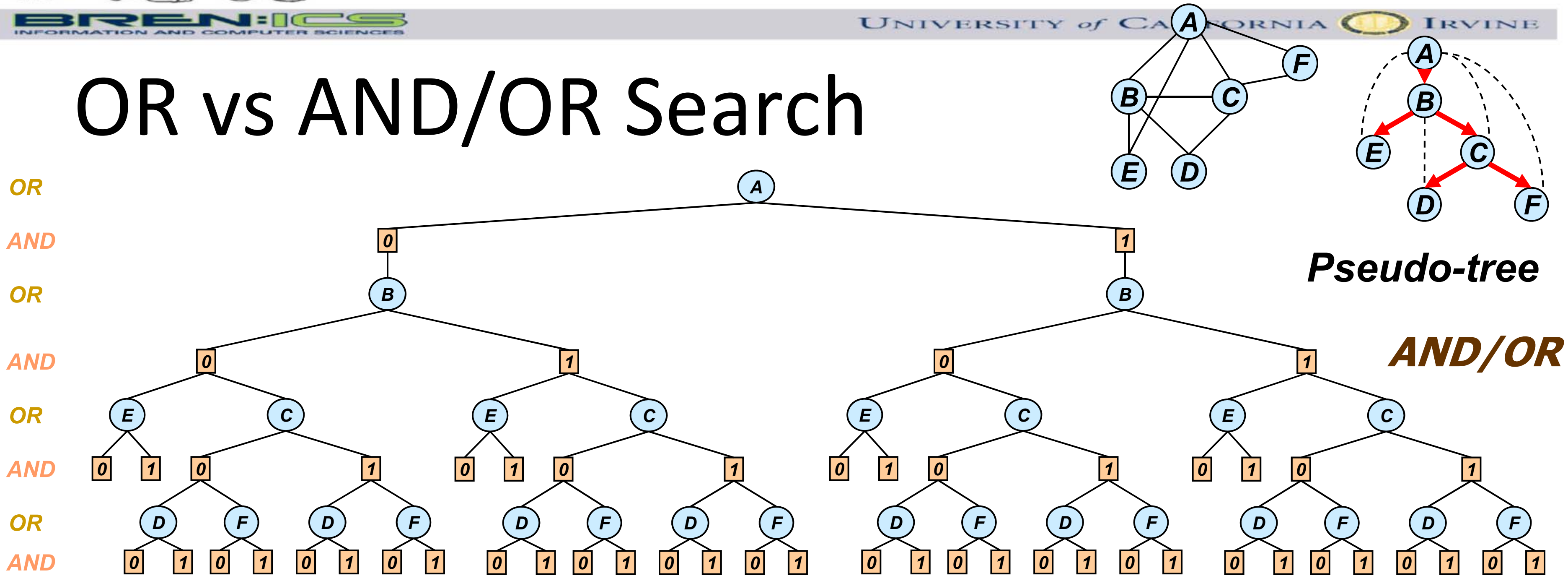
Consistent solutions (greedy search)

Relaxation provides upper bound
Any configuration: lower bound but NP-hard

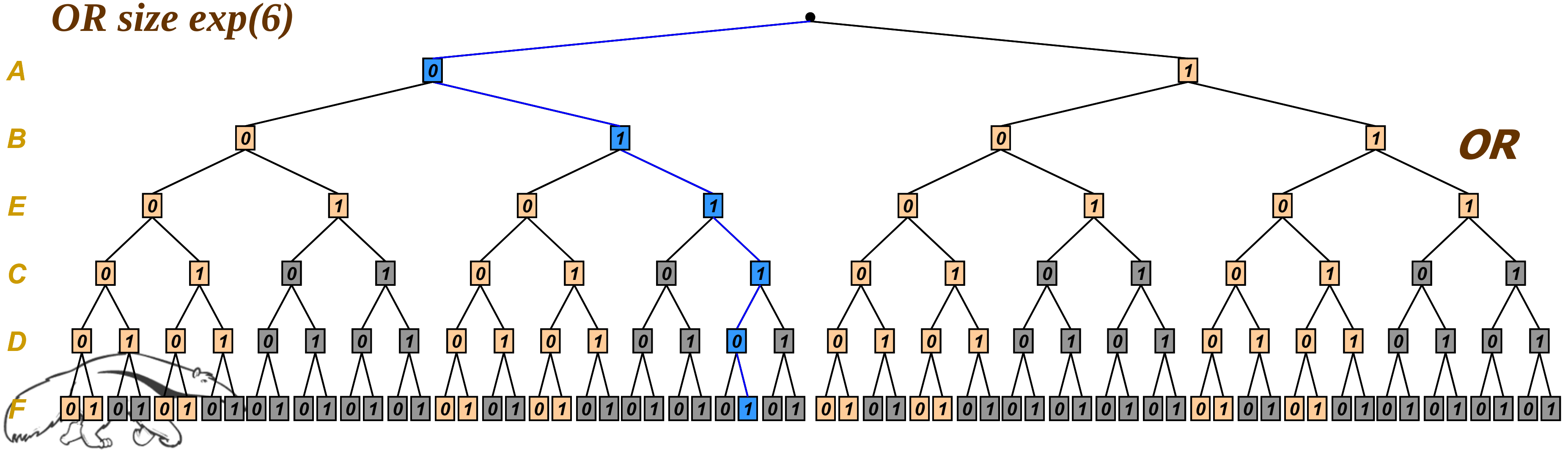
Outline

- Graphical models, Queries , Algorithms
- Inference Algorithms
- Bounded Inference: mini-bucket, cost-shifting
- AND/OR search spaces and AND/OR BnB
- Evaluation, Software
- Conclusions

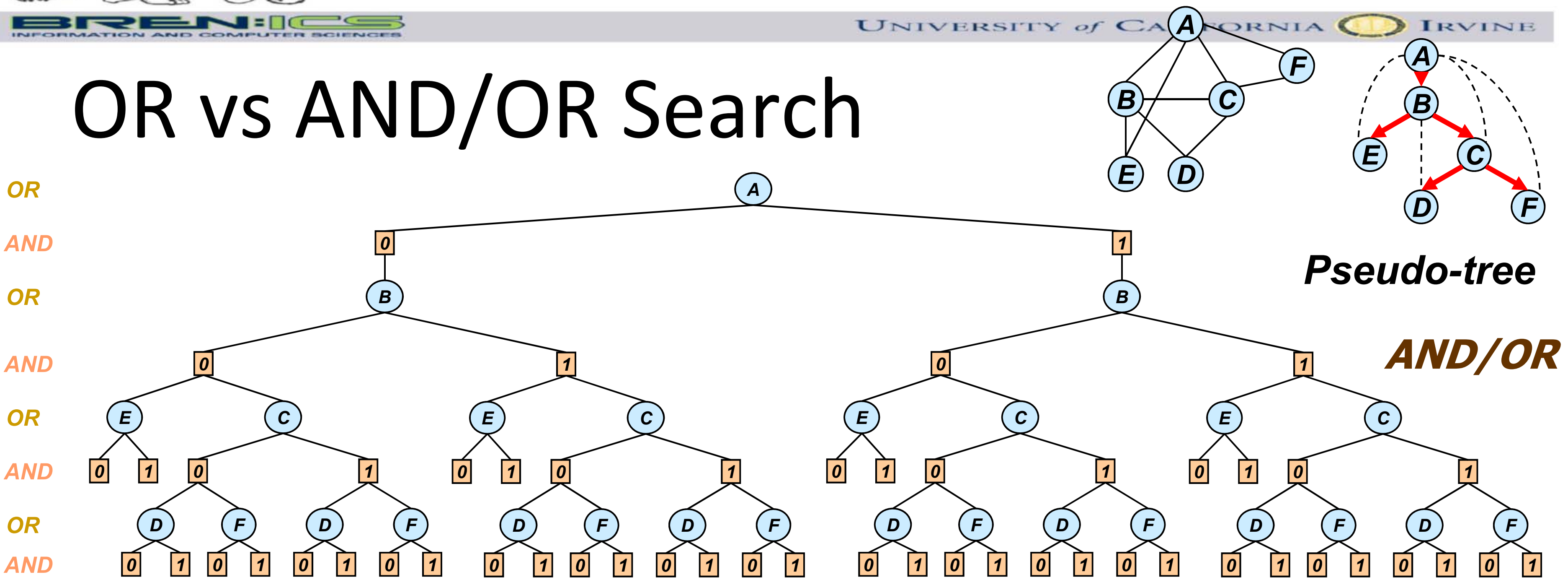
OR vs AND/OR Search



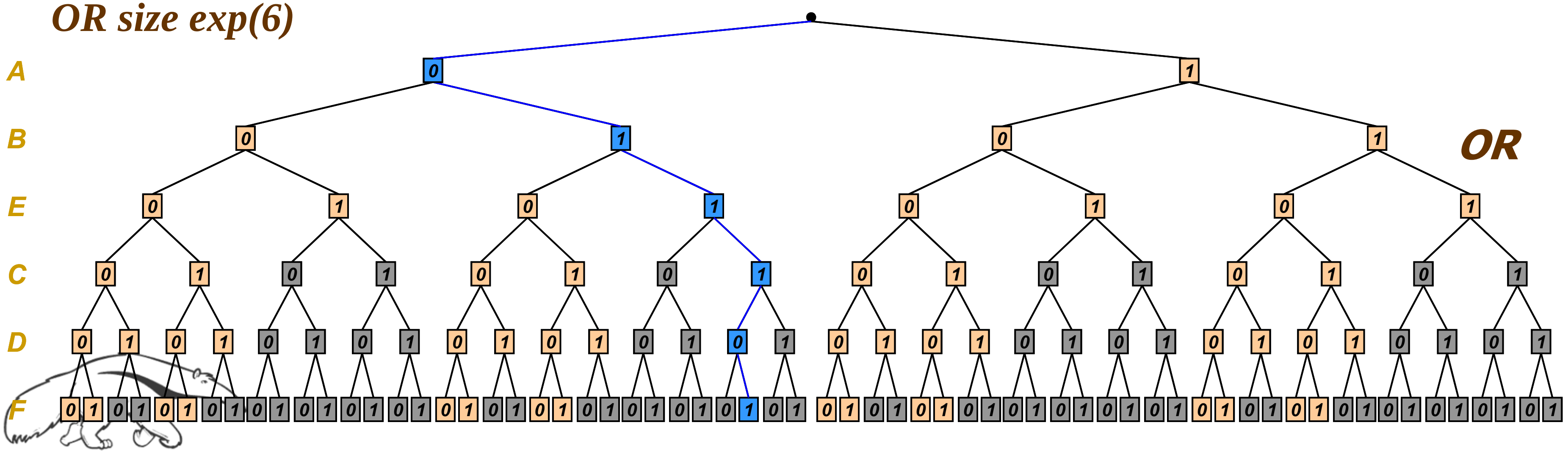
AND/OR size: exp(4),
OR size exp(6)



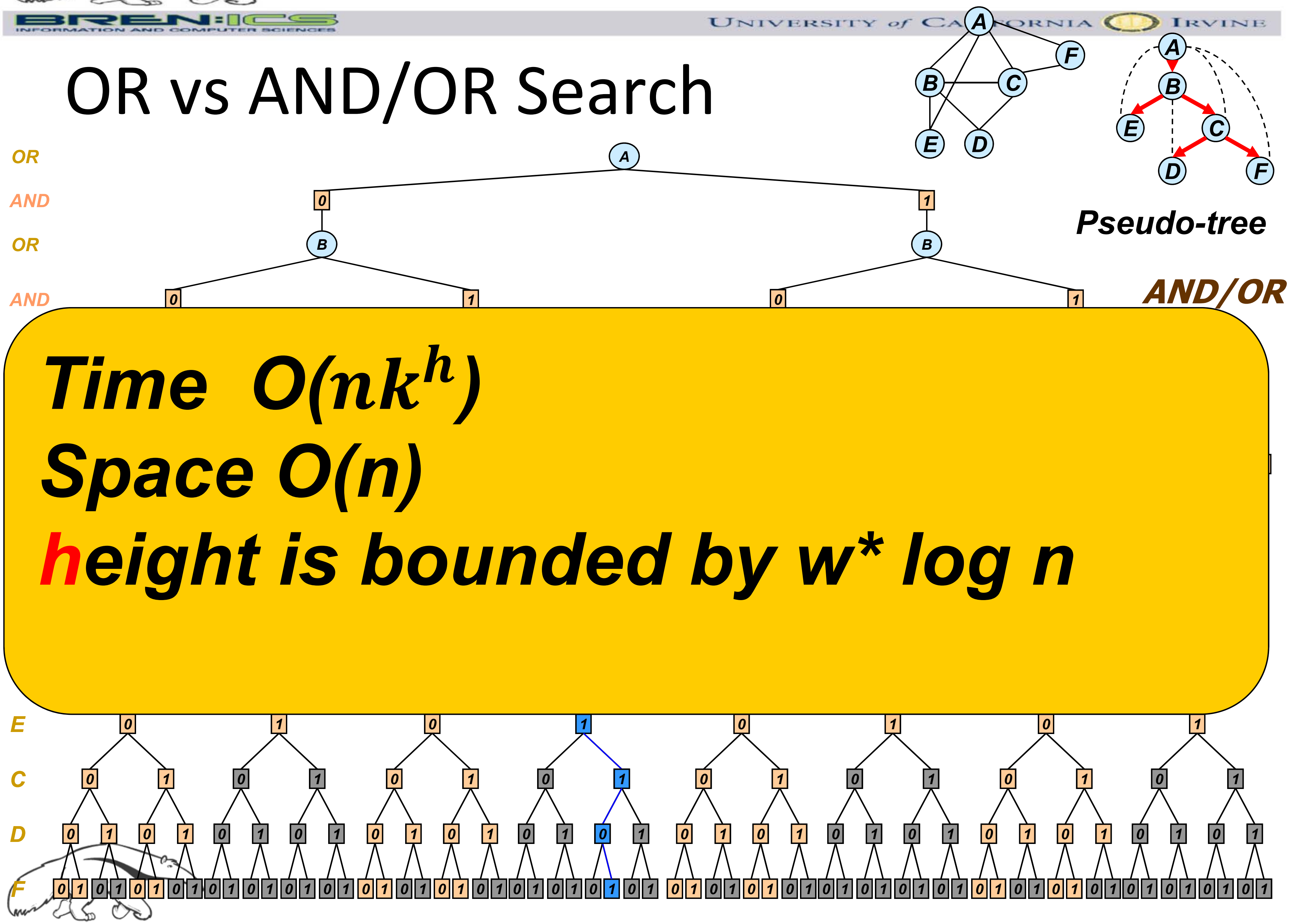
OR vs AND/OR Search



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



OR vs AND/OR Search

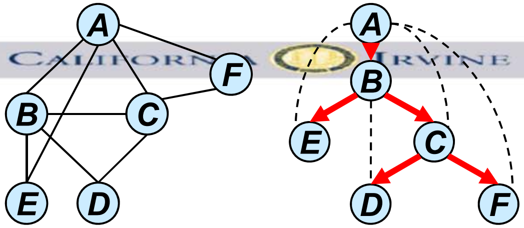


Time $O(nk^h)$

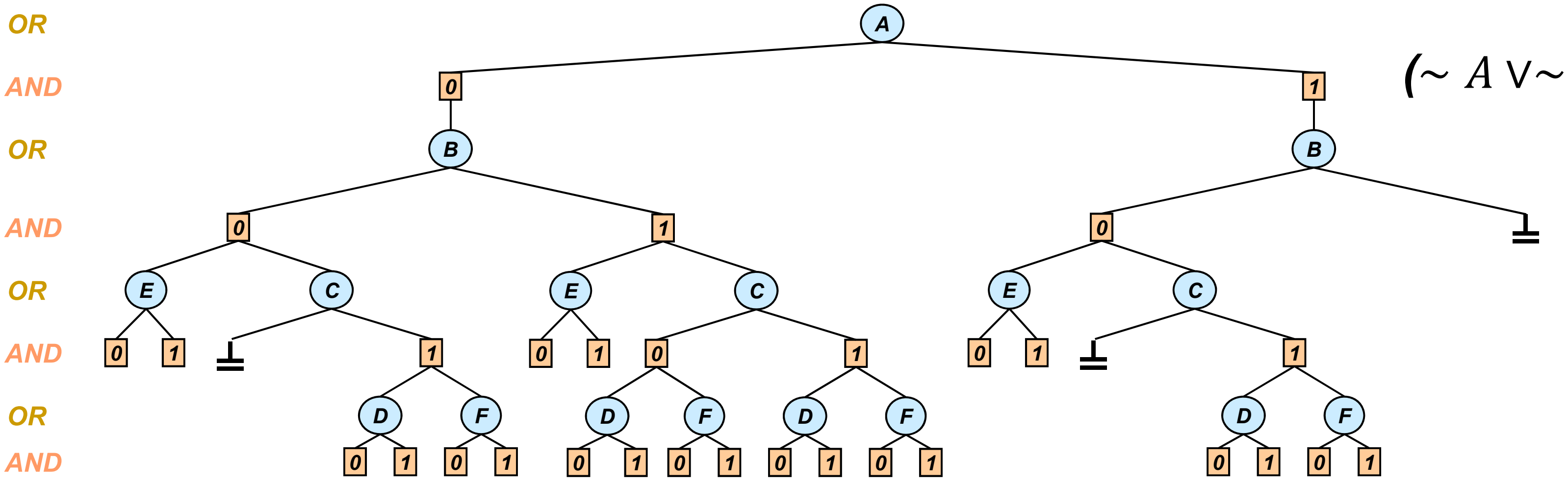
Space O(n)

height is bounded by w* log n

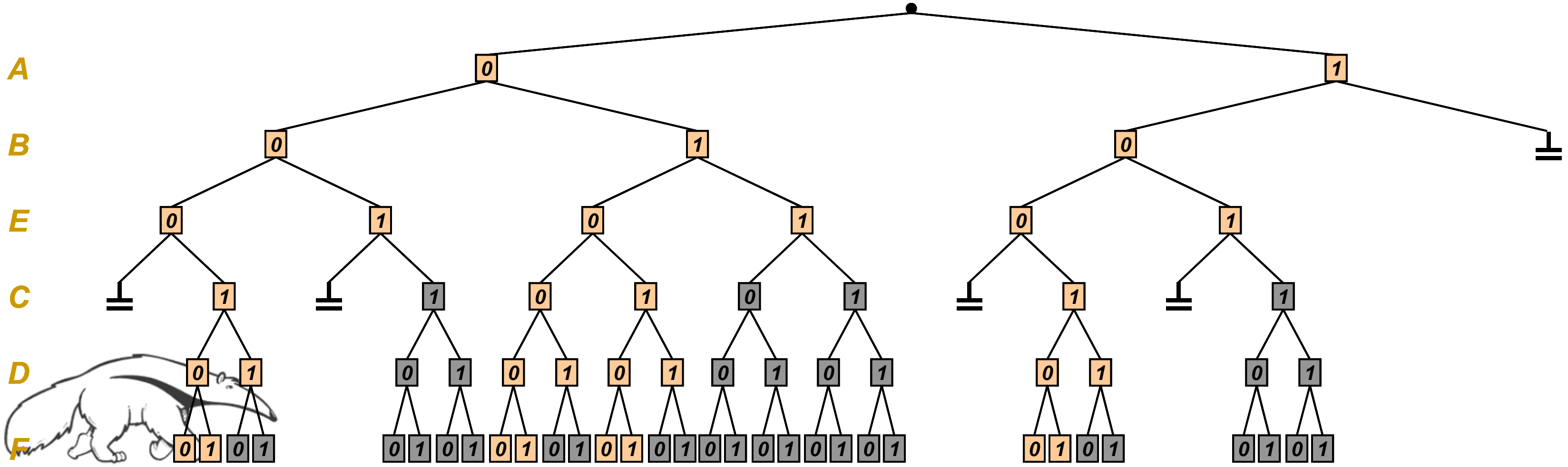
AND/OR vs. OR with Constraints



$(\sim A \vee \sim B) \wedge (B \vee C)$



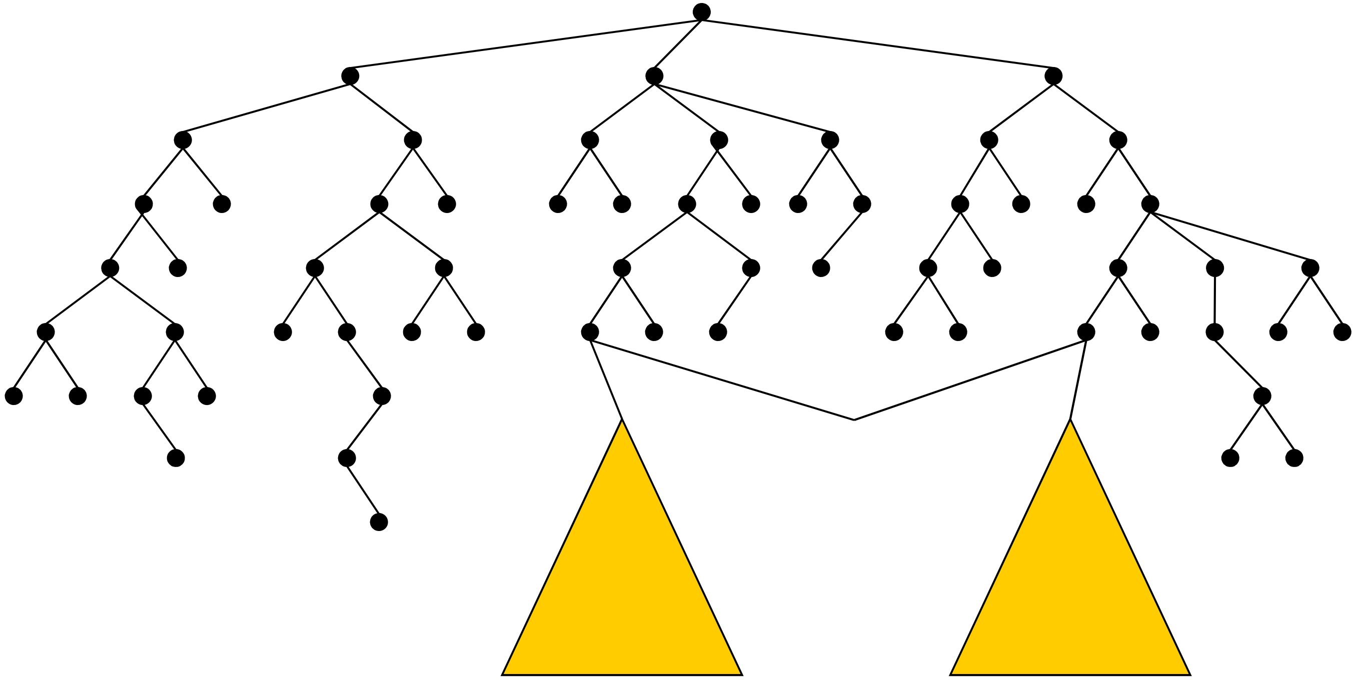
AND/OR



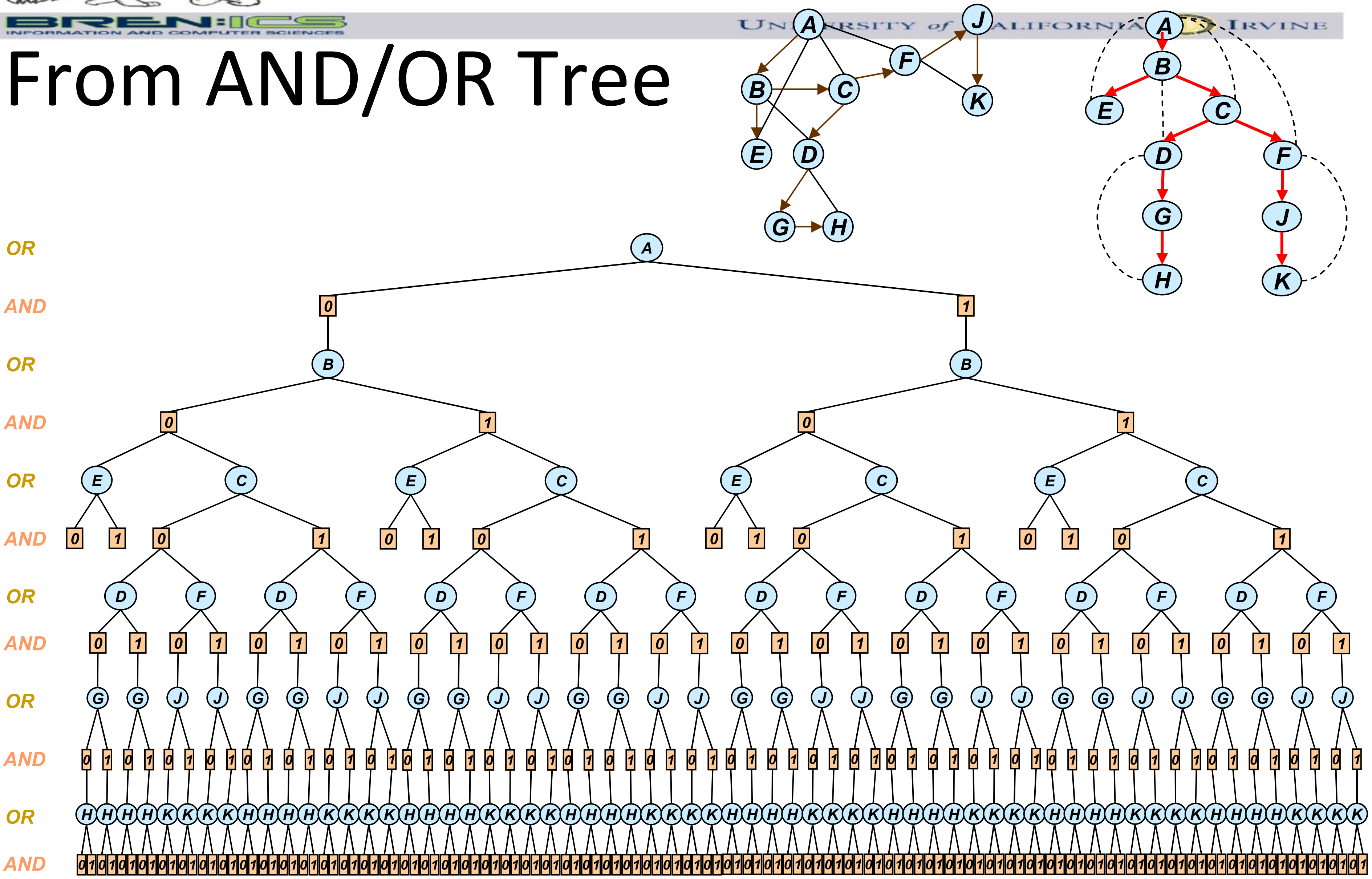
OR

From Search Trees to Search Graphs

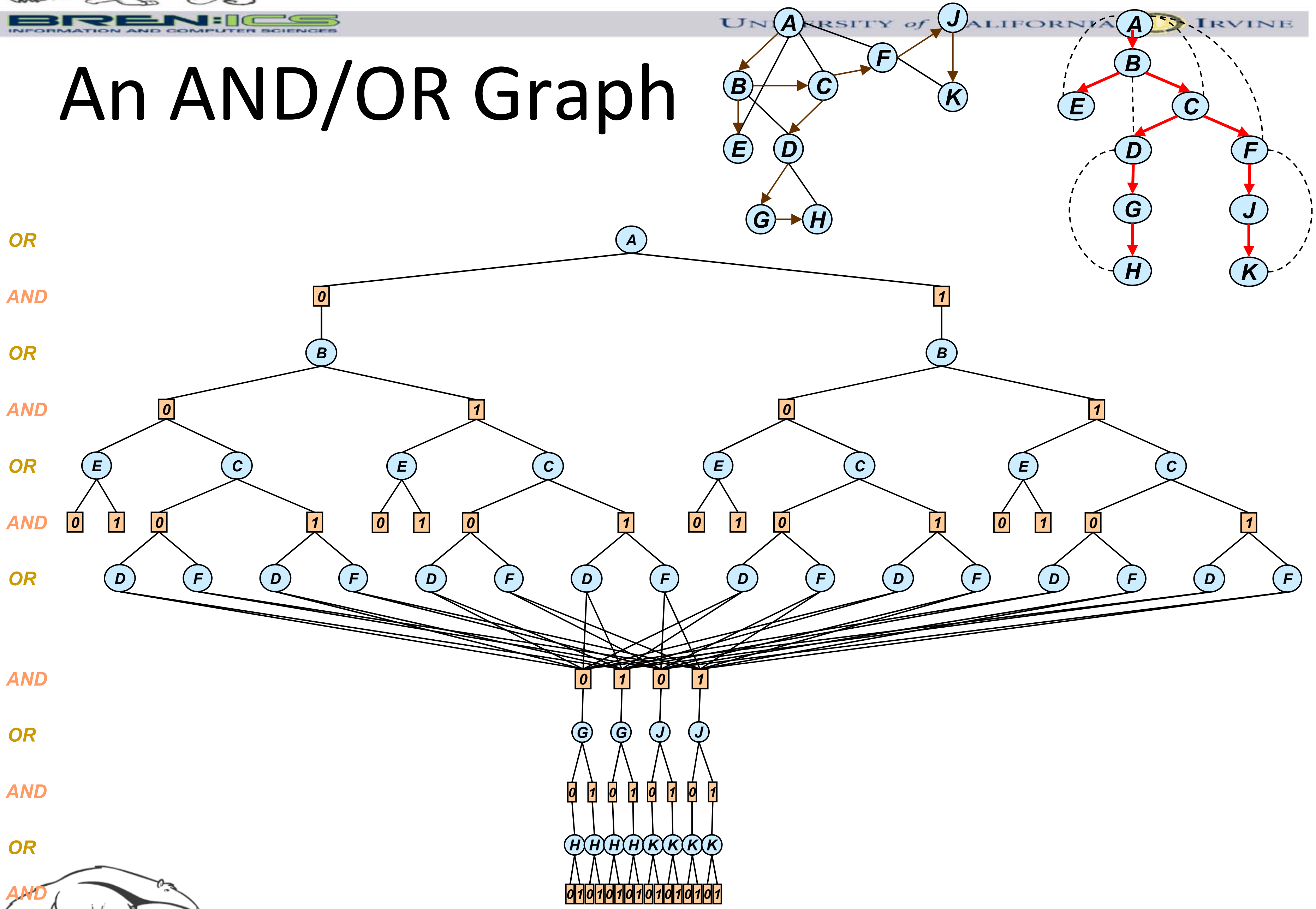
- Any two nodes that root identical sub-trees or sub-graphs can be merged



From AND/OR Tree



An AND/OR Graph

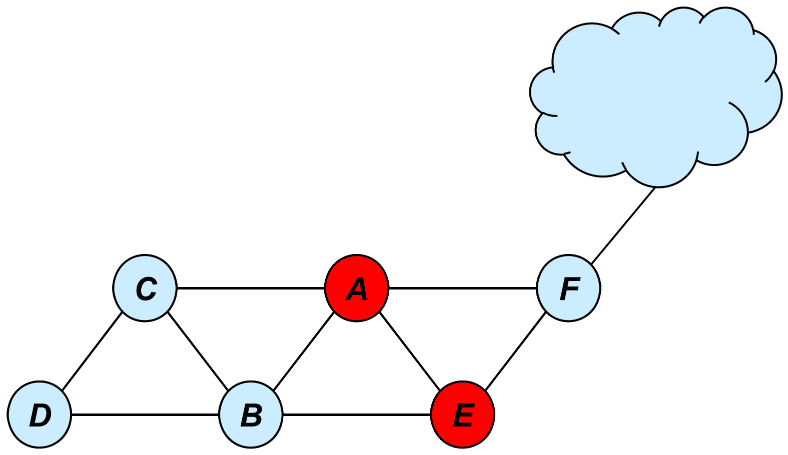


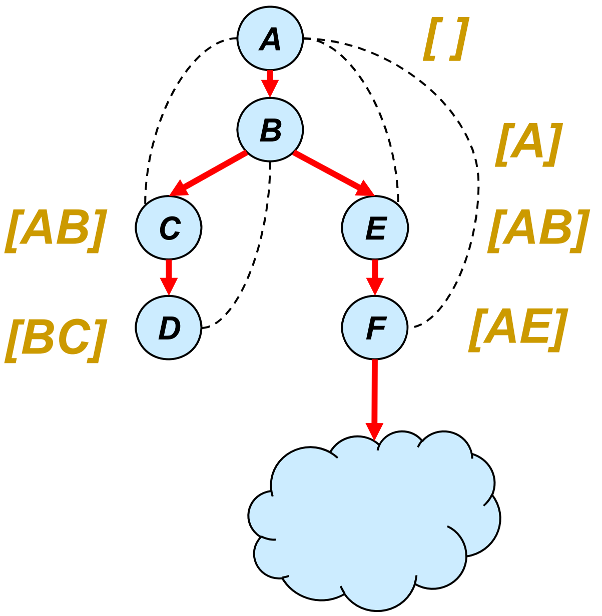
Merging based on context

context (X) = ancestors of X connected to

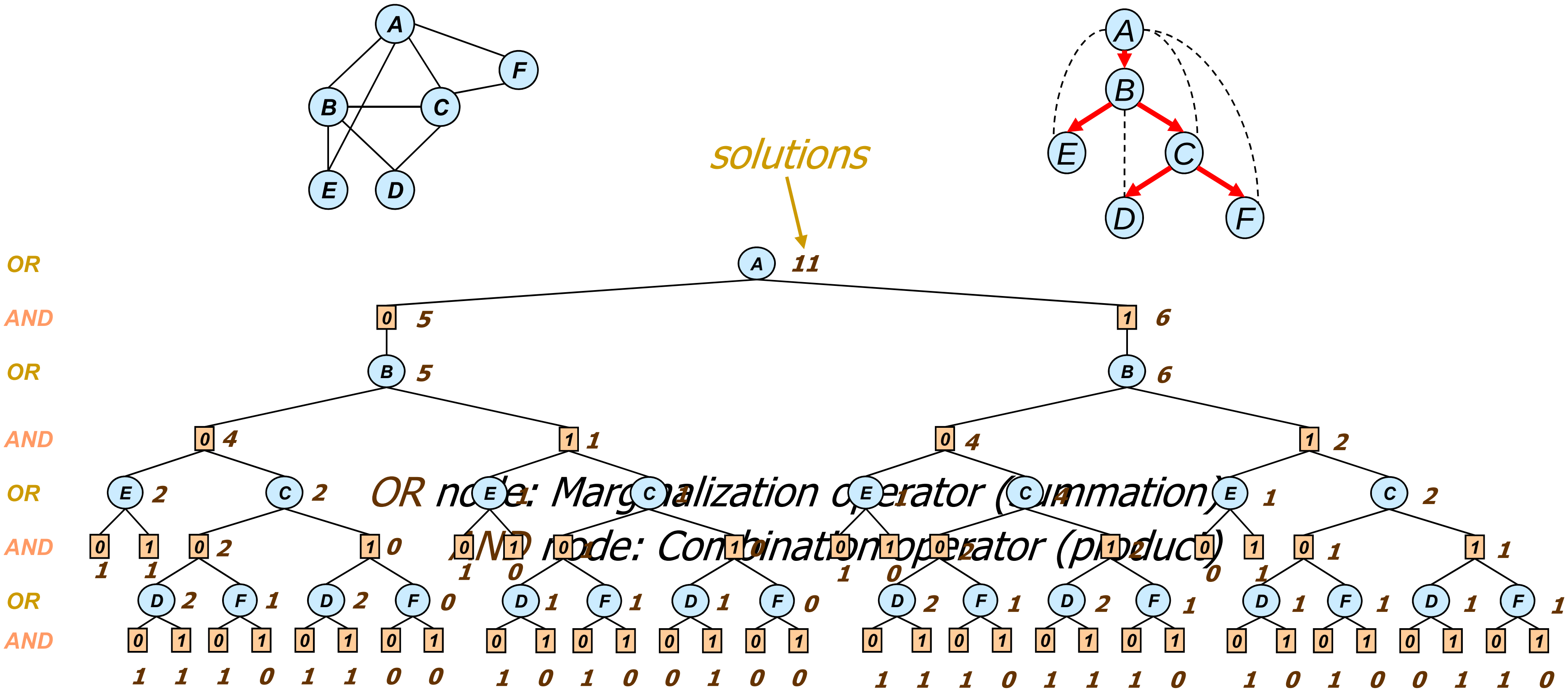
X

descendants of X





Counting Solutions: Sum-Product Networks

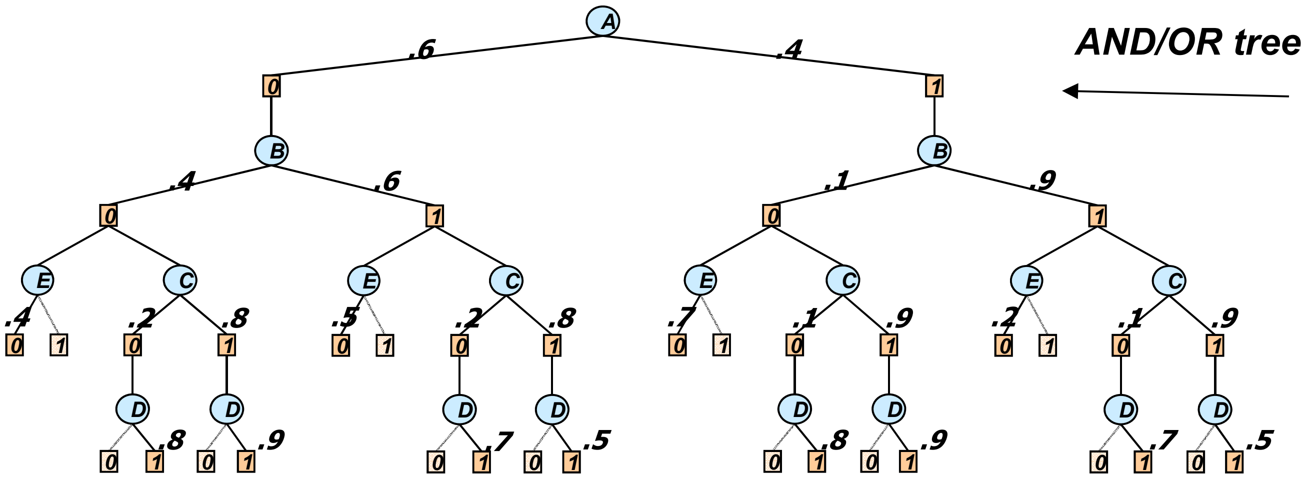


OR node: Marginalization operator (summation)

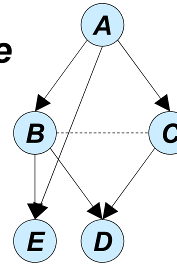
AND node: Combination operator (product)

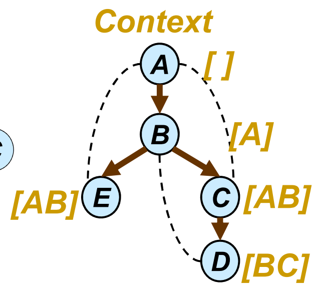
Value of node = number of solutions below it

Weighted AND/OR Search Space



AND/OR tree





$P(B|A)$

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

$P(E|A,B)$

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

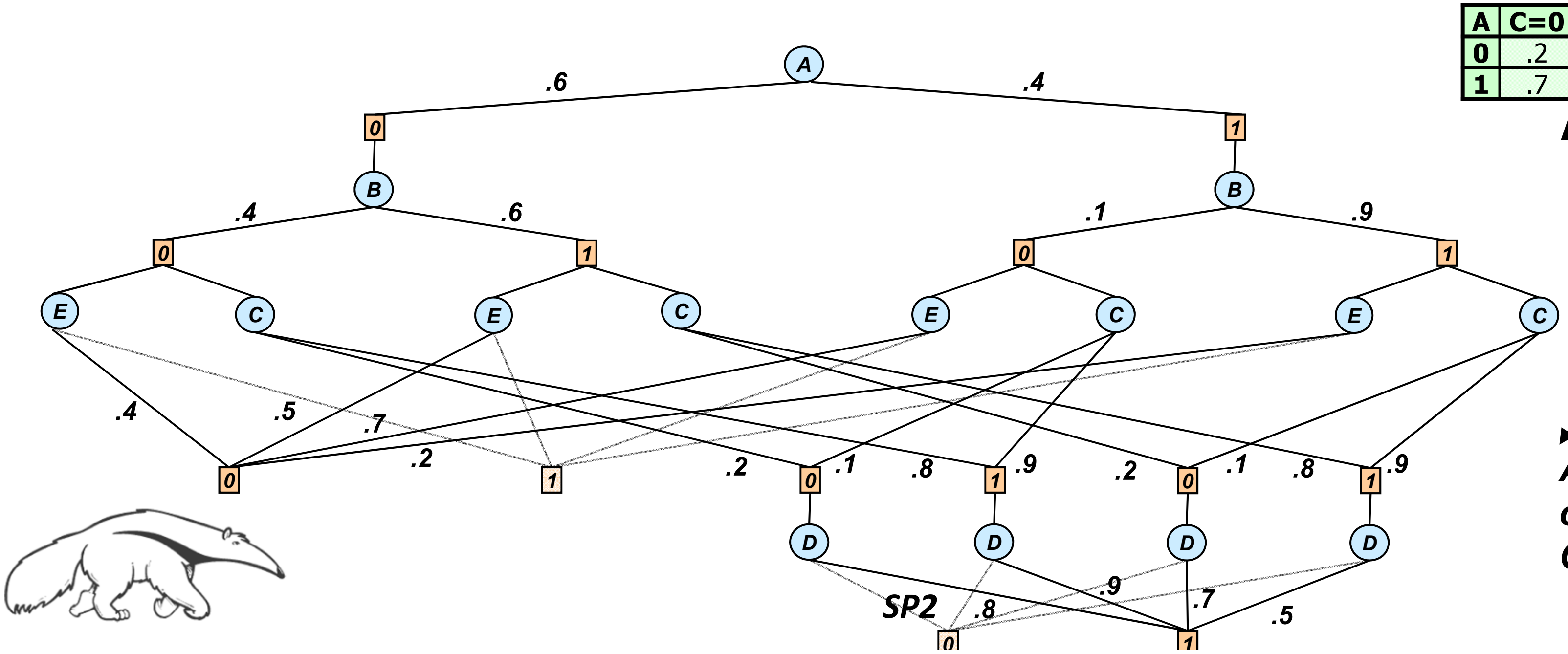
$P(C|A)$

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

$P(A)$

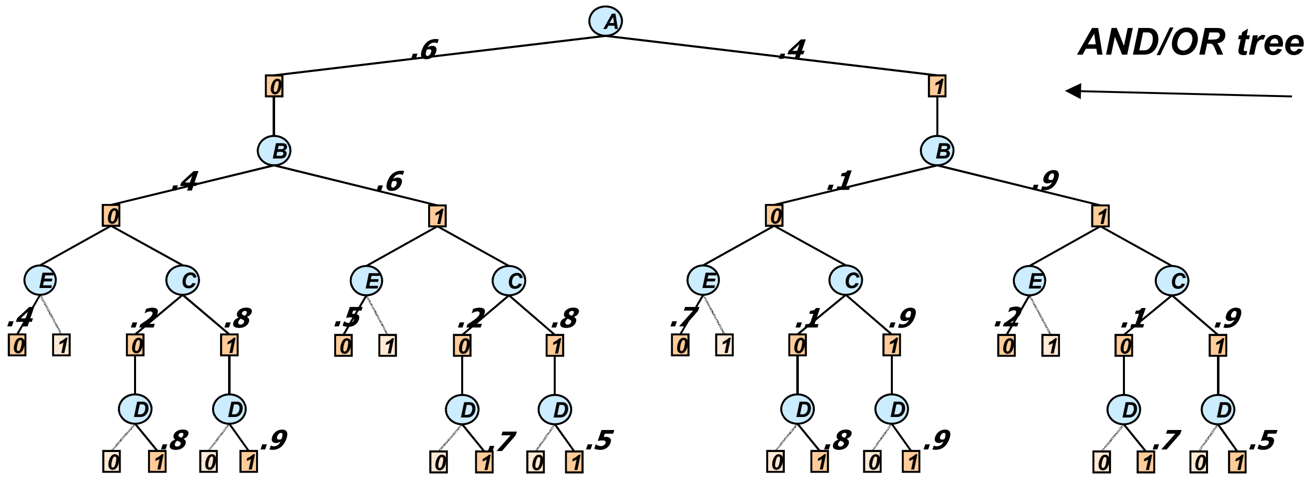
A	P(A)
0	.6
1	.4

Evidence: E=0

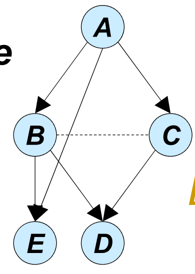


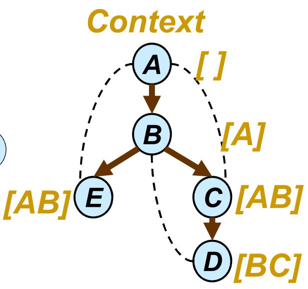
AND/OR context-minimal Graph

Weighted AND/OR Search Space



AND/OR tree





$P(B|A)$

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

$P(E|A,B)$

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

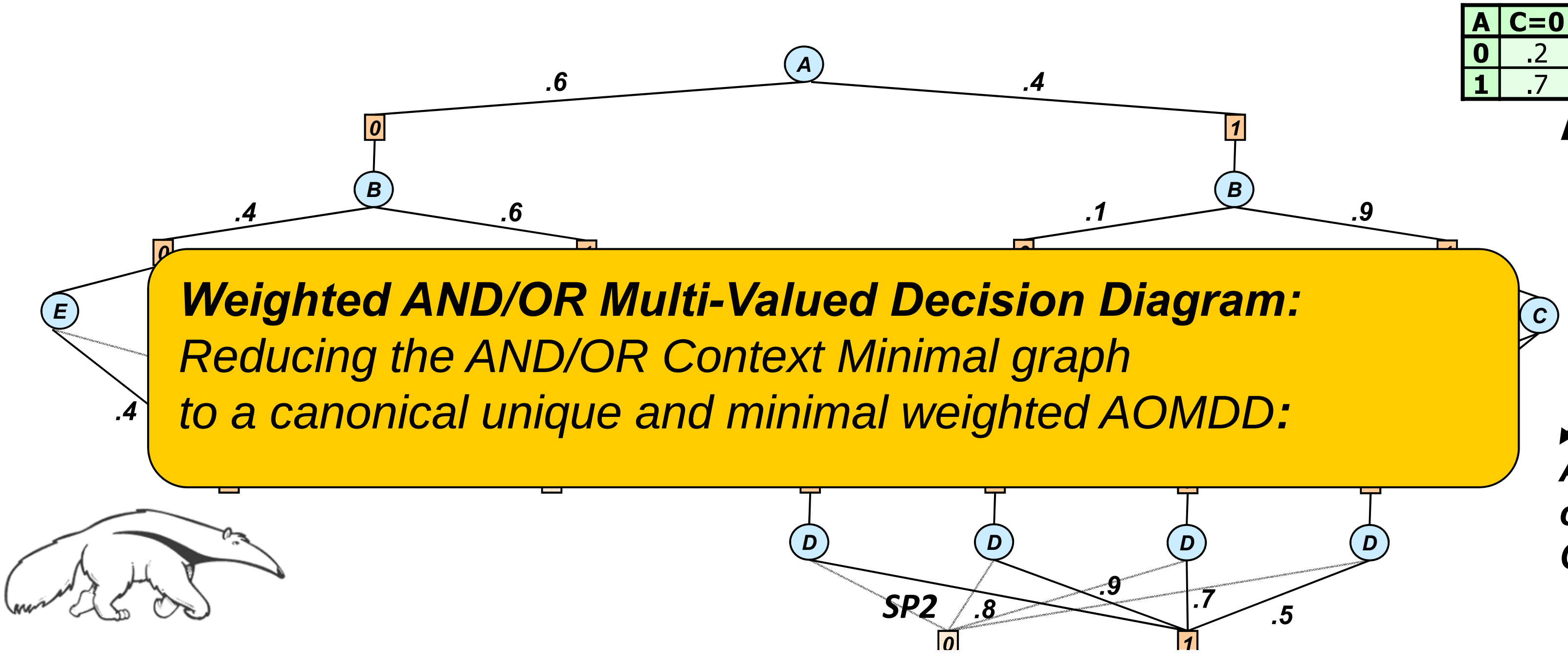
$P(C|A)$

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

$P(A)$

A	P(A)
0	.6
1	.4

Evidence: E=0



Weighted AND/OR Multi-Valued Decision Diagram:

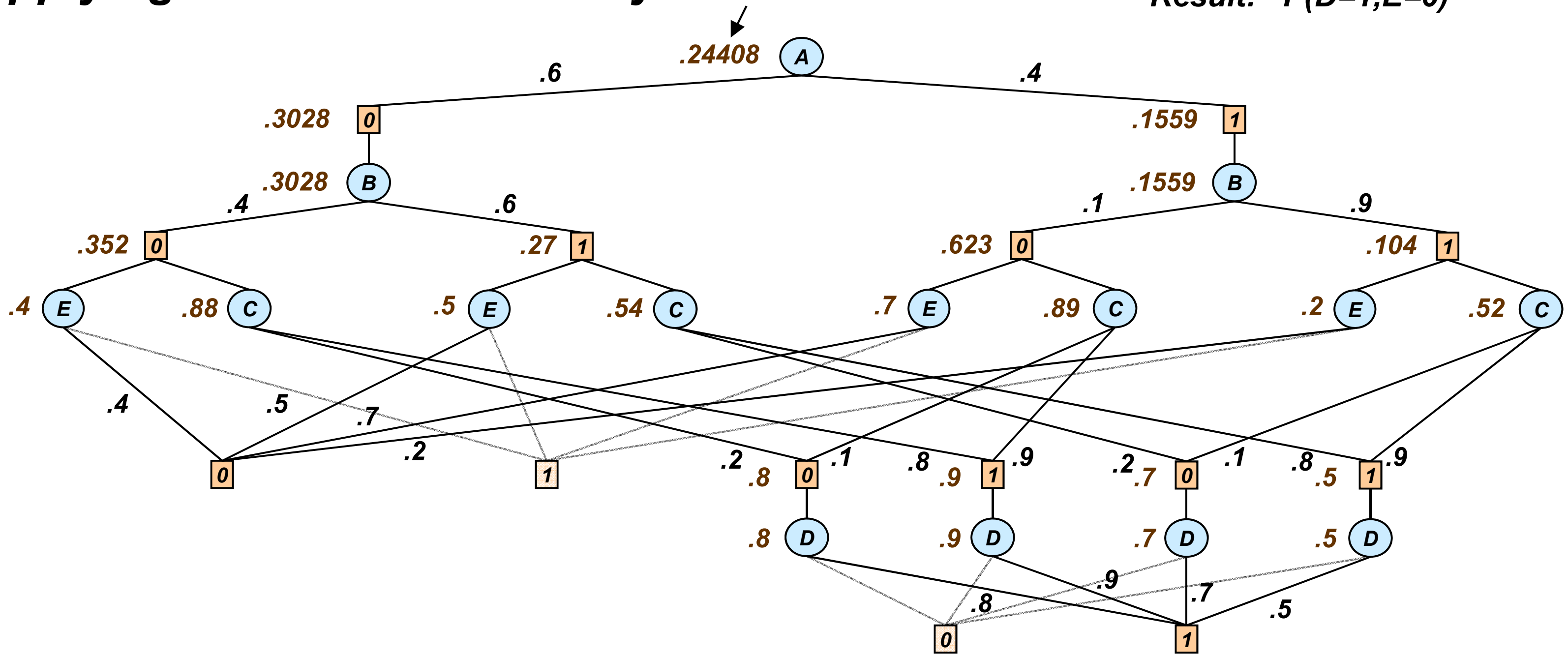
Reducing the AND/OR Context Minimal graph
to a canonical unique and minimal weighted AOMDD:

AND/OR context-minimal Graph

Answering Queries: Sum-Product

Applying Sum-Product is easy

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



$P(B|A)$

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

$P(E|A,B)$

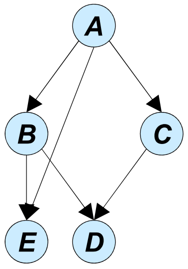
A	B	E=0	E=1
0	0	.4	.6
0	1	.5	.5
1	0	.7	.3
1	1	.2	.8

$P(C|A)$

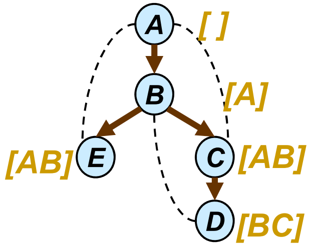
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



Context



Answering Queries: Sum-Product(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

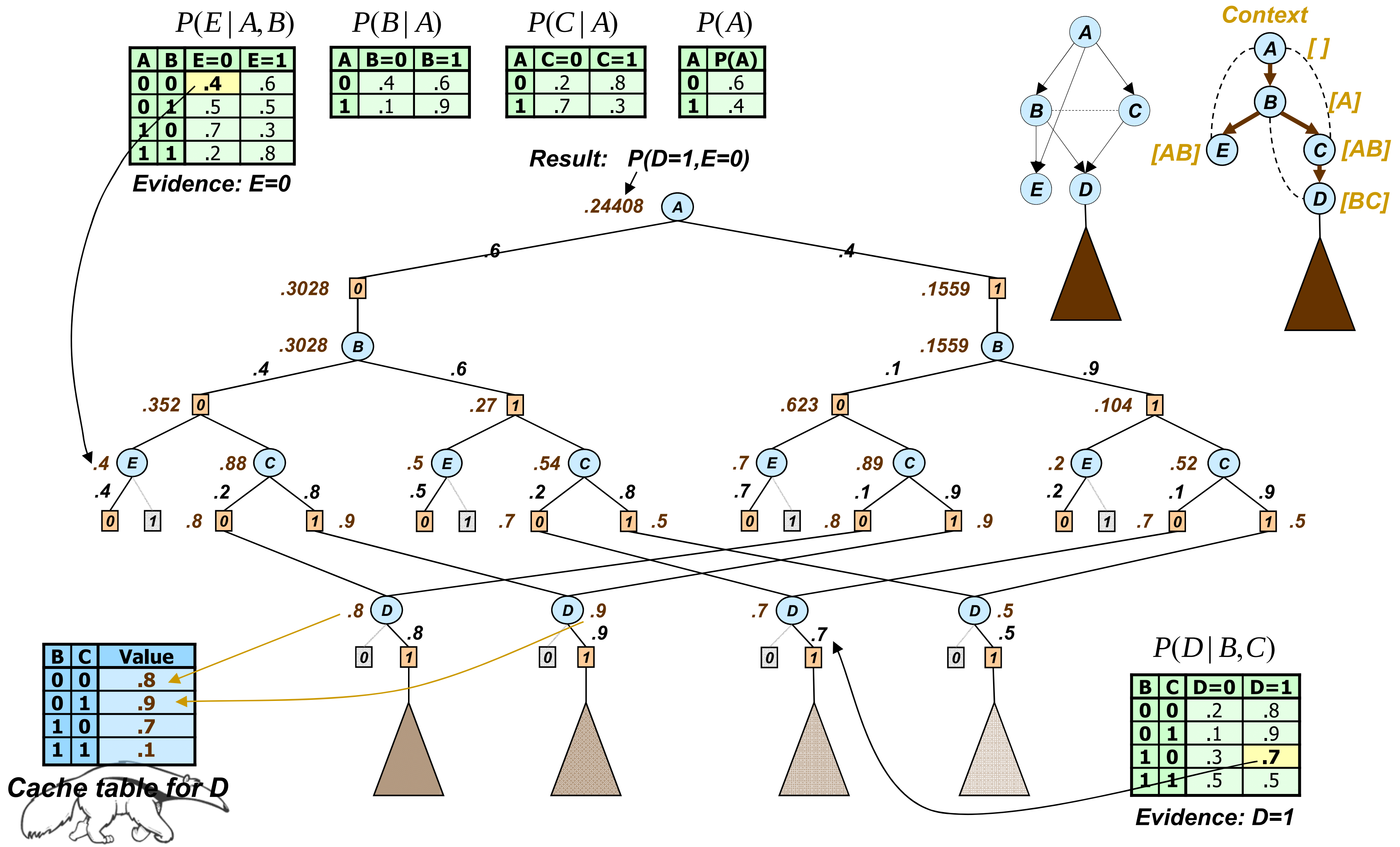
$P(C \mid A)$

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

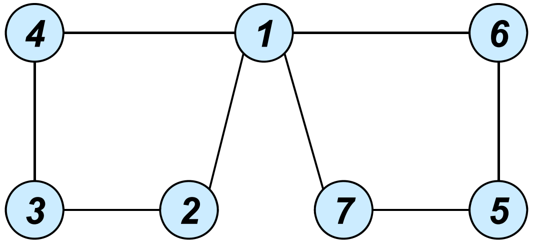
$P(D \mid B,C)$

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

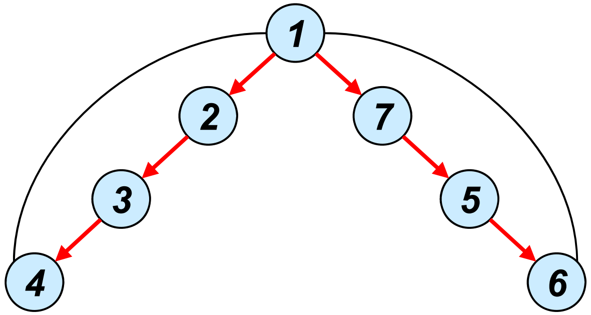
Pseudo-Trees

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

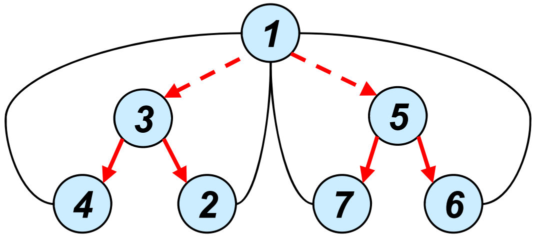


(a) Graph

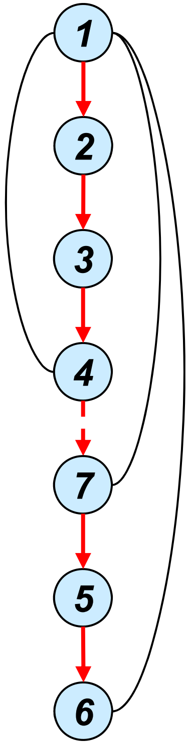
h <= w* log n



(b) DFS tree
depth=3

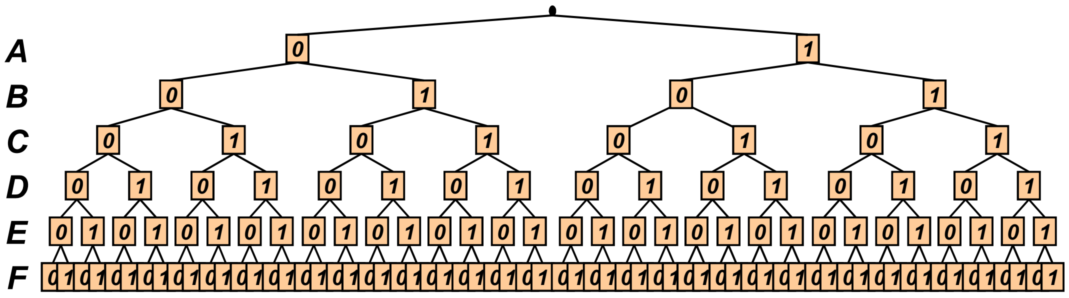


(c) pseudo- tree
depth=2



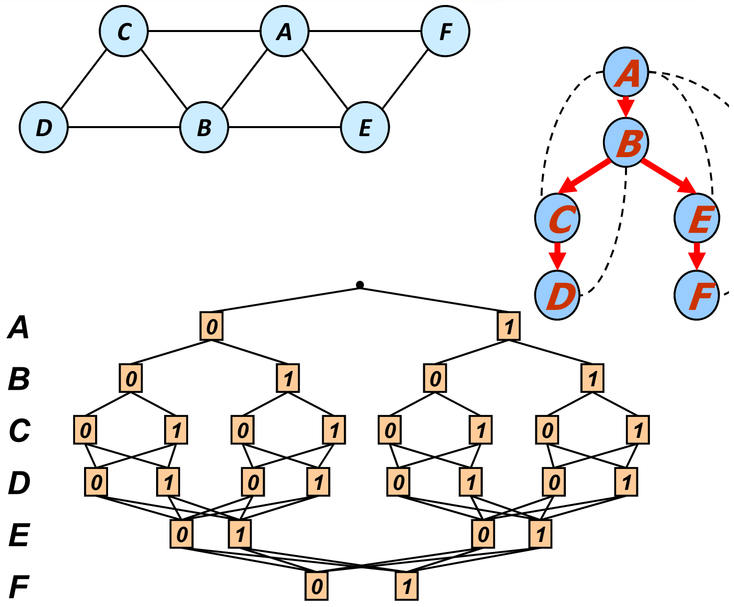
(d) Chain
depth=6

All Four Search Spaces



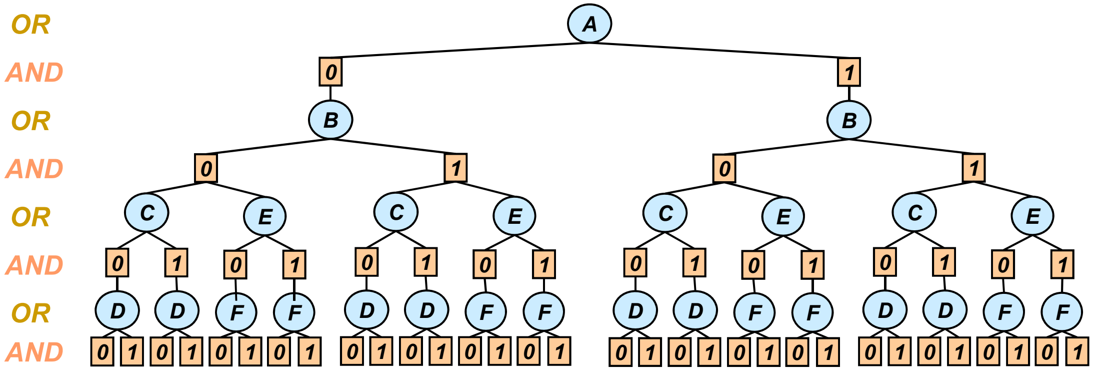
Full OR search tree

126 nodes



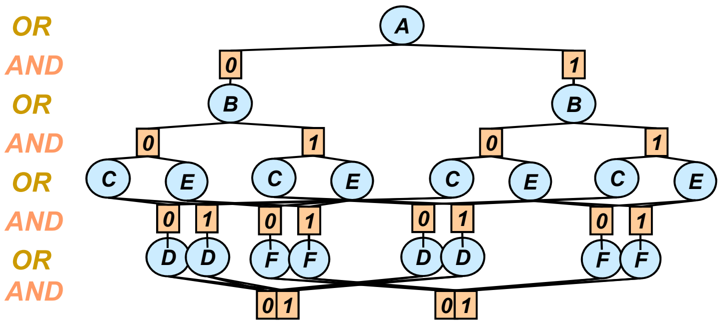
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes

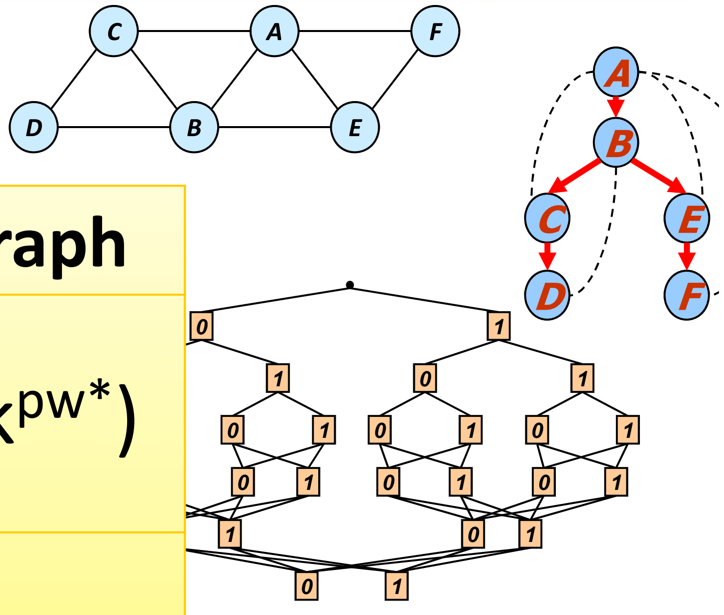


Context minimal AND/OR search graph

18 AND nodes

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

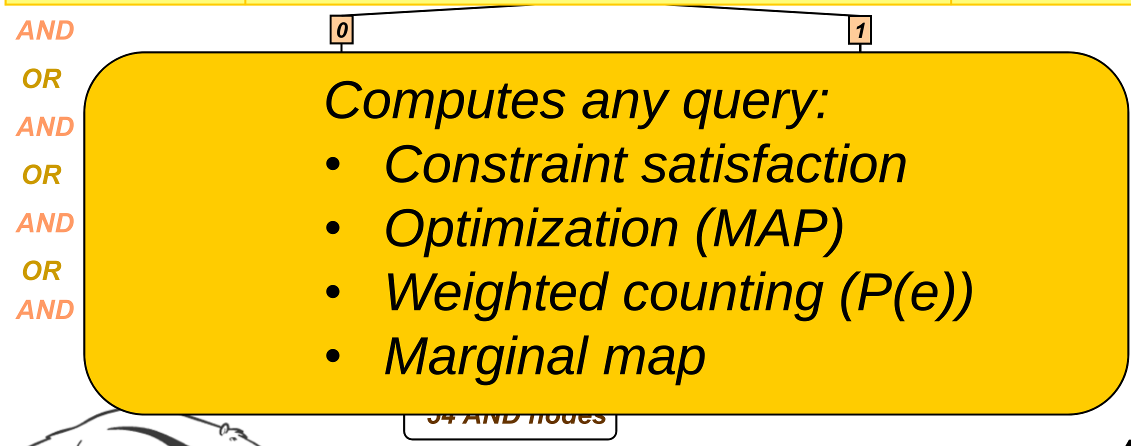
All Four Search Spaces



xt minimal OR search graph

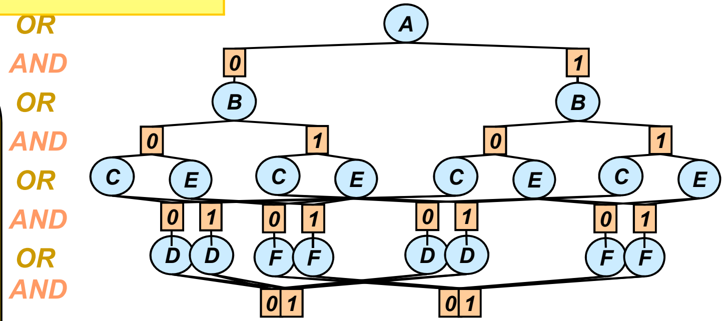
28 nodes

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



Computes any query:

- *Constraint satisfaction*
- *Optimization (MAP)*
- *Weighted counting (P(e))*
- *Marginal map*



Context minimal AND/OR search graph

18 AND nodes

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

MAP: AND/OR BnB search

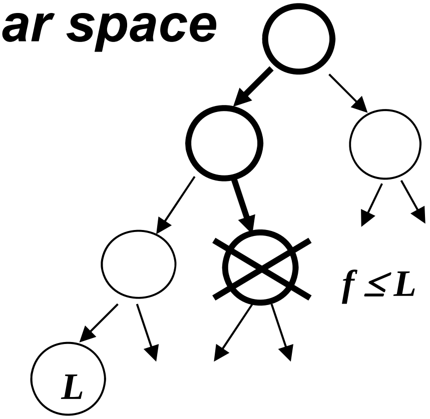
Basic Heuristic Search Schemes

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

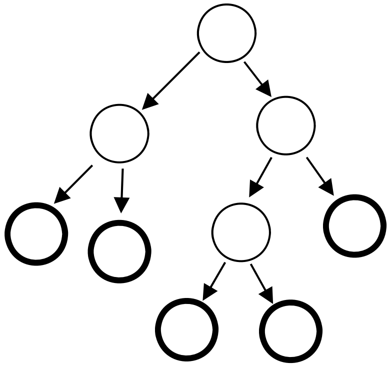
Linear space



2. Best-First Search

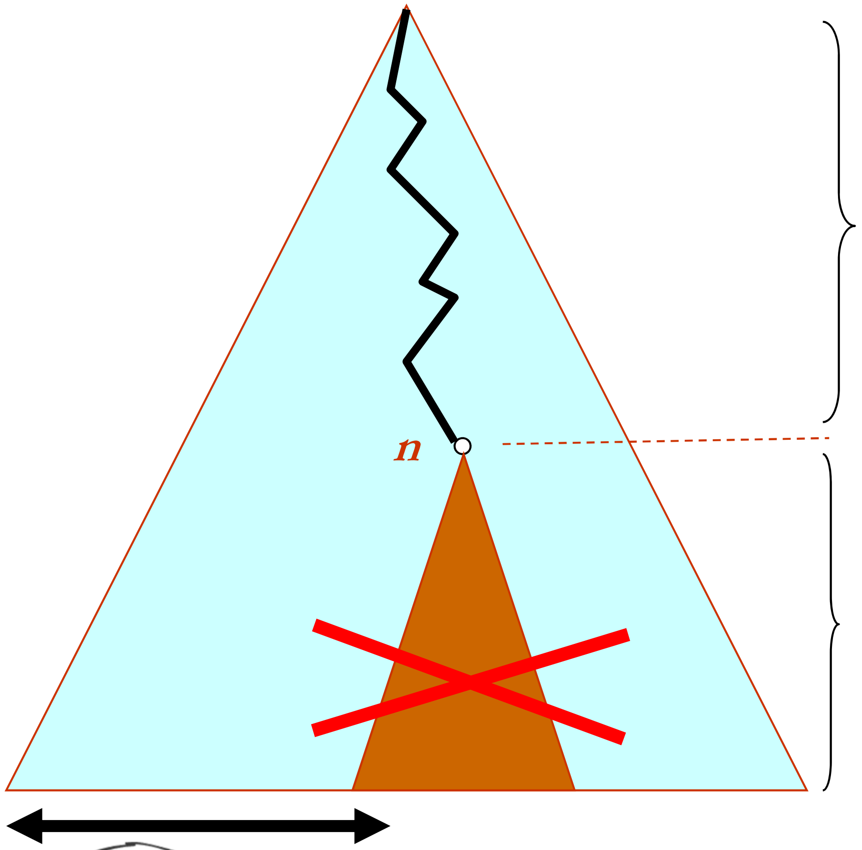
Always expand the node with the highest heuristic value

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



Classic Branch-and-Bound

Each node is a COP subproblem (defined by current conditioning)



g(n)

f(n) = g(n) + h(n)
f(n) = lower bound

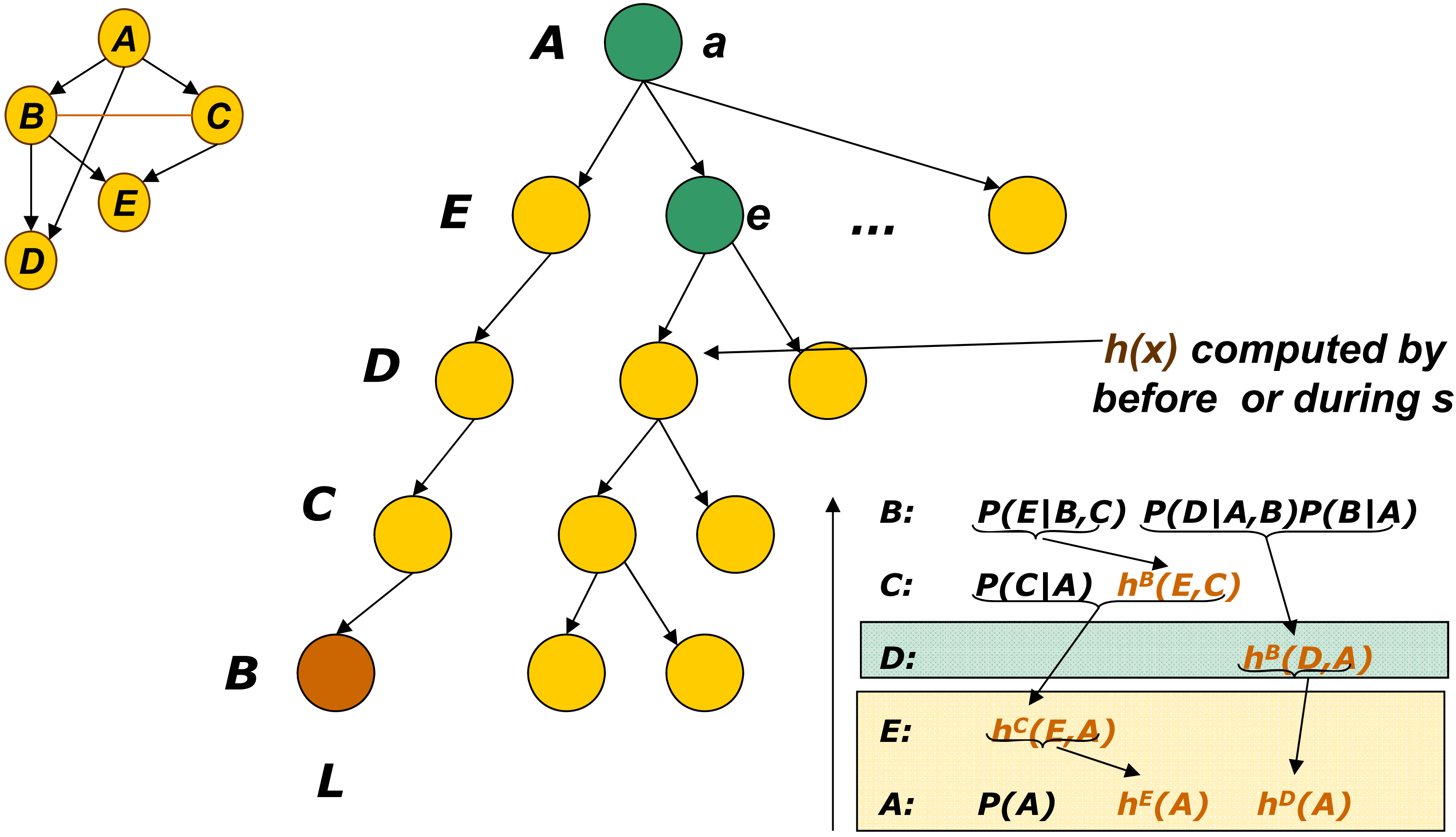
Prune if f(n) ≥ UB

h(n) - under-estimates Optimal cost below n

(UB) Upper Bound = best solution so far

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

B: $P(E|B,C)$ $P(D|A,B)P(B|A)$

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

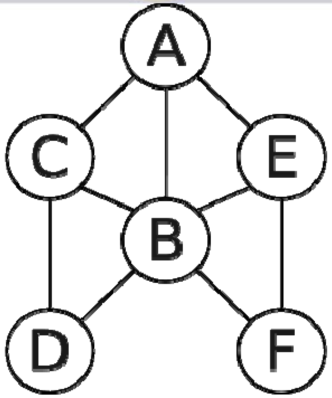
D: $h^B(D,A)$

E: $h^C(E,A)$

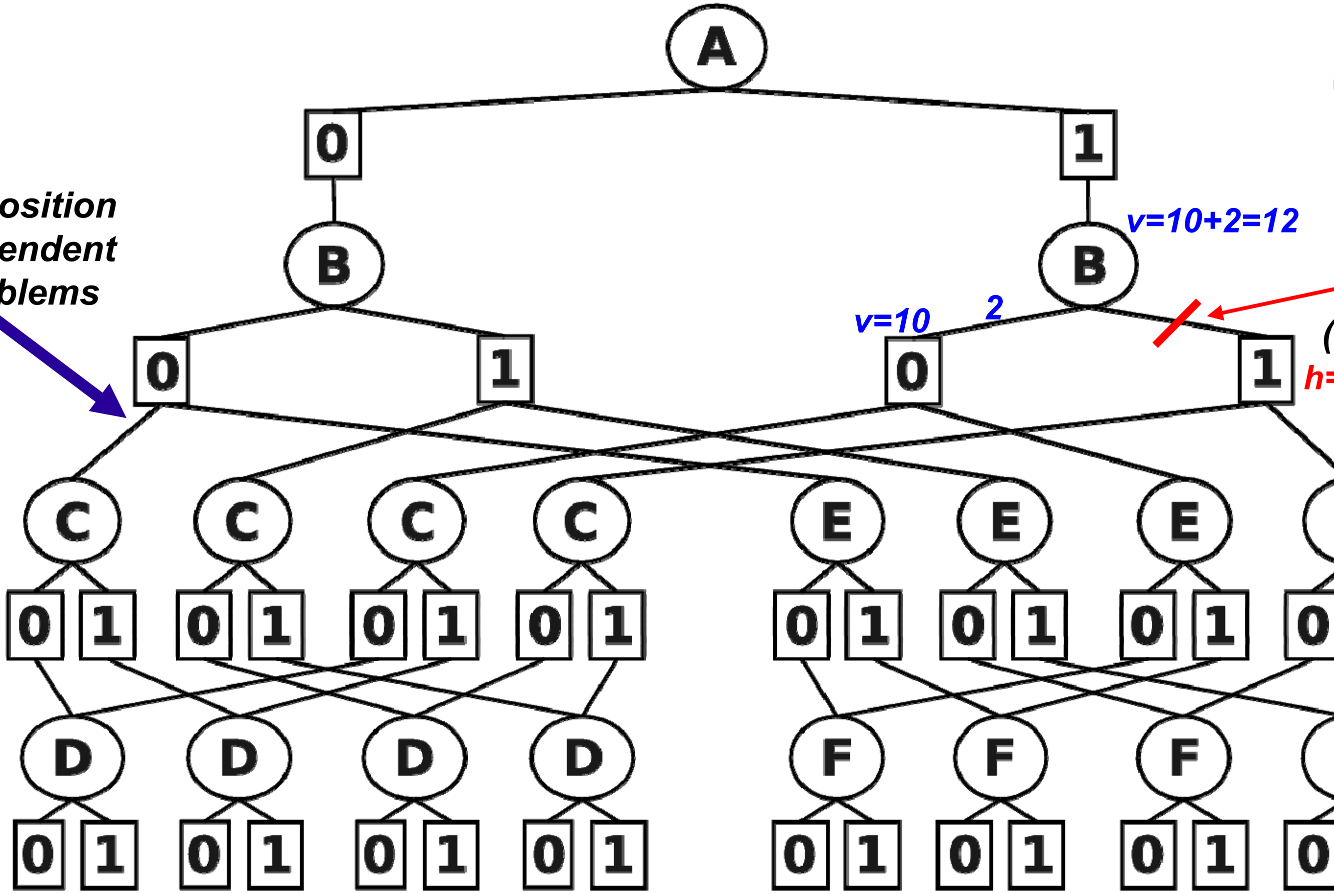
A: $P(A)$ $h^E(A)$ $h^D(A)$

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

AND/OR Branch-and-Bound



Decomposition of independent subproblems



Prune based on current best solution and heuristic estimate (mini-bucket heuristic).

Cache table for F (independent of A)

B E	cost
0 0	10
0 1	6
1 0	...
1 1	...

[Marinescu & Dechter, AIJ'09]

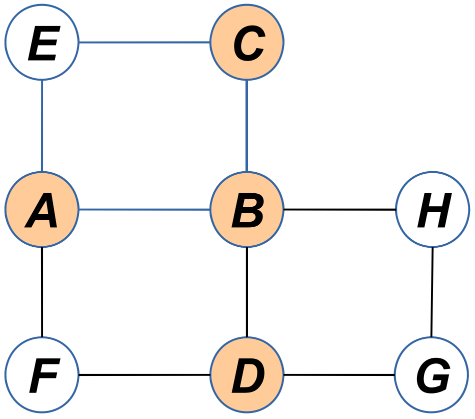
MAP: Anytime, BnB

- Best-First, Recursive Best-First
- Anytime:
  - Breadth-Rotate AND/OR BnB
  - Weighted heuristic AND/OR search
- Finding m-best solutions

Marginal Map: AND/OR BnB Search

AND/OR BnB over the appropriate search space
Guided by weighted mini-bucket +cost-shiifting

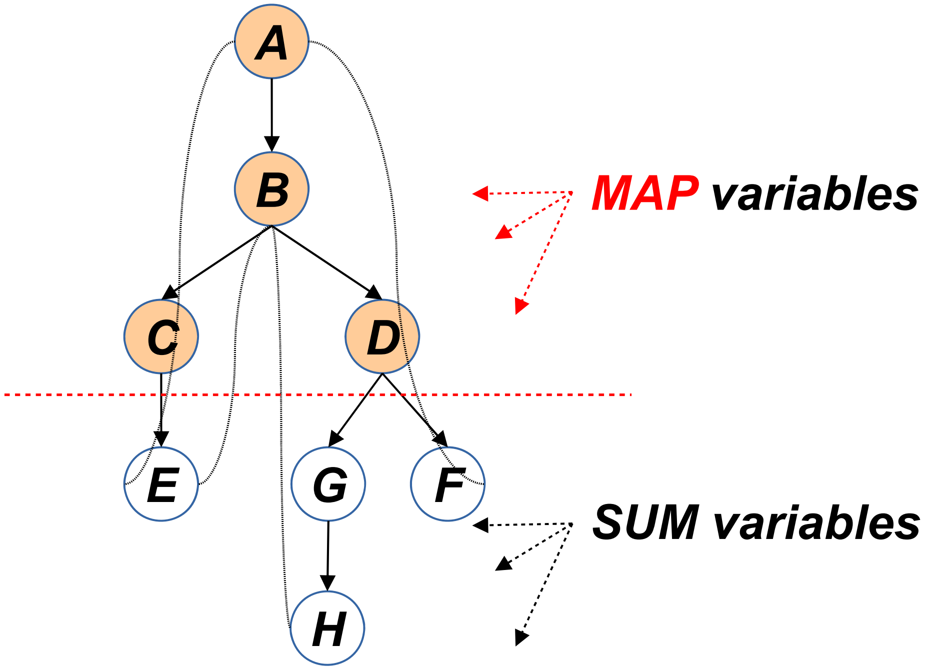
AND/OR Search Spaces for MMAP



primal graph

$$X_M = \{A, B, C, D\}$$

$$X_S = \{E, F, G, H\}$$



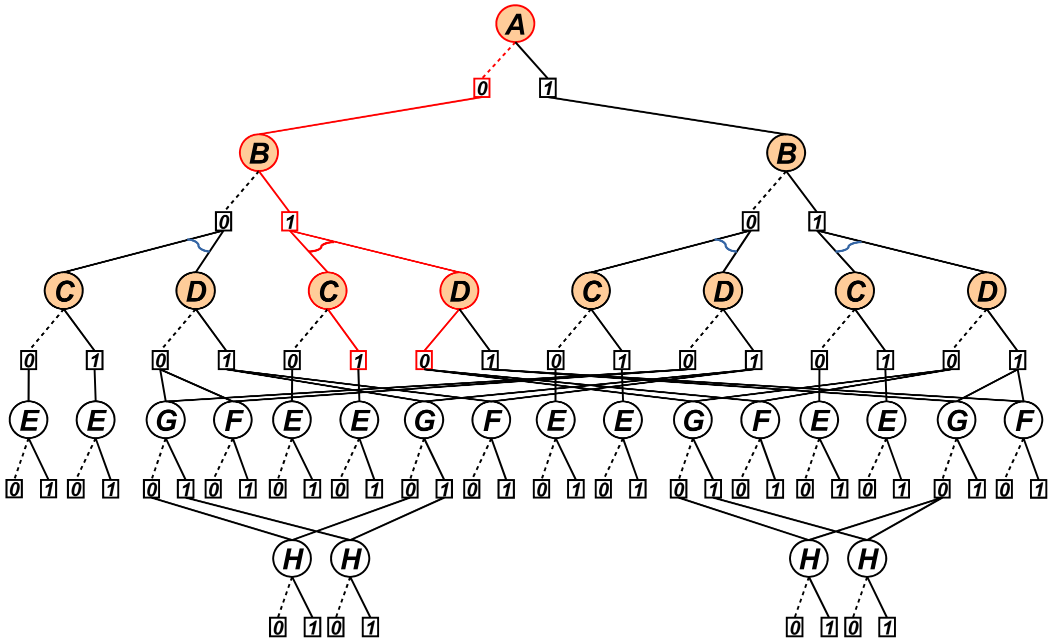
***MAP* variables**

SUM variables

constrained pseudo tree

[Marinescu, Dechter and Ihler, 2014]

AND/OR Search Spaces for MMAP



Node types

- OR (MAP): max
- OR (SUM): sum
- AND: multiplication

Arc weights

- Derived from input F

Problem decomposition over MAP variables

AND/OR Search for Marginal MAP

AOBB: Depth-First AND/OR Branch and Bound

- ***Depth-first traversal of the AND/OR search graph***
- ***Prune only at OR nodes that correspond to MAP variables***
- ***Cost of MAP assignment obtained by searching the SUM sub-problem***

AOBF: Best First AND/OR Search

- ***Best-first (AO*) traversal of the AND/OR space corresponding to the MAP variables***
- ***SUM subproblem solved exactly***

RBFAOO: Recursive Best-First AND/OR Search

- ***Recursive best-first traversal of the AND/OR graph***
- ***For SUM subproblems, the threshold is set to ∞ (equivalent to depth-first search)***

AND/OR Search for Marginal MAP

AOBB: Depth-First AND/OR Branch and Bound

Depth-first traversal of the AND/OR search graph

Prune only at OR nodes that correspond to MAP

Also Anytime Marginal Map solvers

corresponding to the MAP variables

SUM subproblem solved exactly

RBFAOO: Recursive Best-First AND/OR Search

Recursive best-first traversal of the AND/OR graph

For SUM subproblems, the threshold is set to ∞ (equivalent to depth-first search)

Outline

- Graphical models, Queries
- Inference Algorithms
- Bounding Inference schemes (mini-bucket, re-parameterization)
- AND/OR search
- **Evaluation, Software**
- Conclusions and recent work: Parallelism, m-best, weighted best-first, marginal map, tree-SLS)

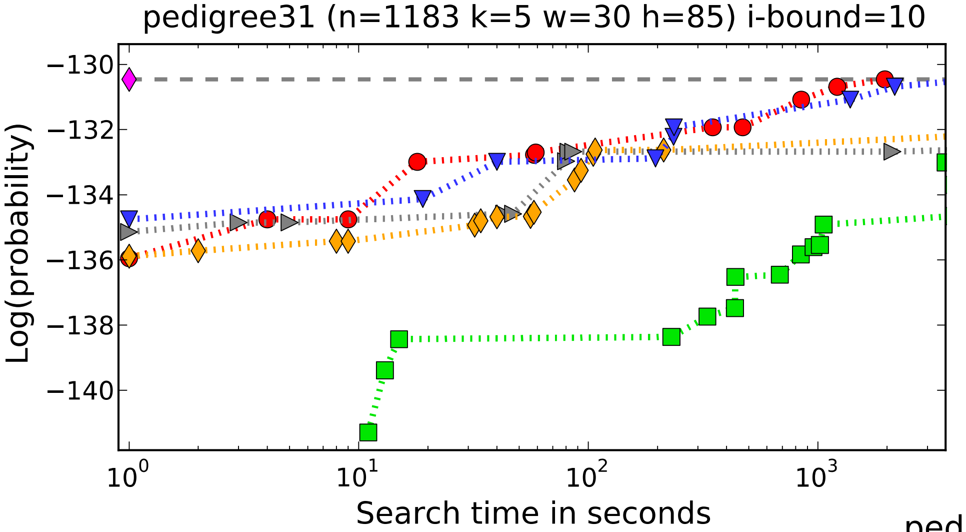
Software

- **aolib**
  - http://graphmod.ics.uci.edu/group/Software (standalone AOBB, AOBF solvers)
- **daoopt**
  - https://github.com/lotten/daoopt (distributed and standalone AOBB solver)

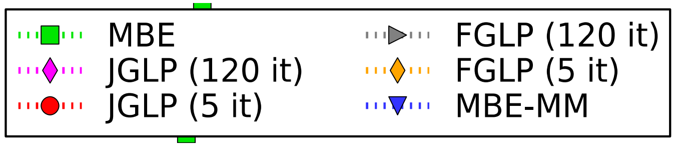
UAI Probabilistic Inference Competitions

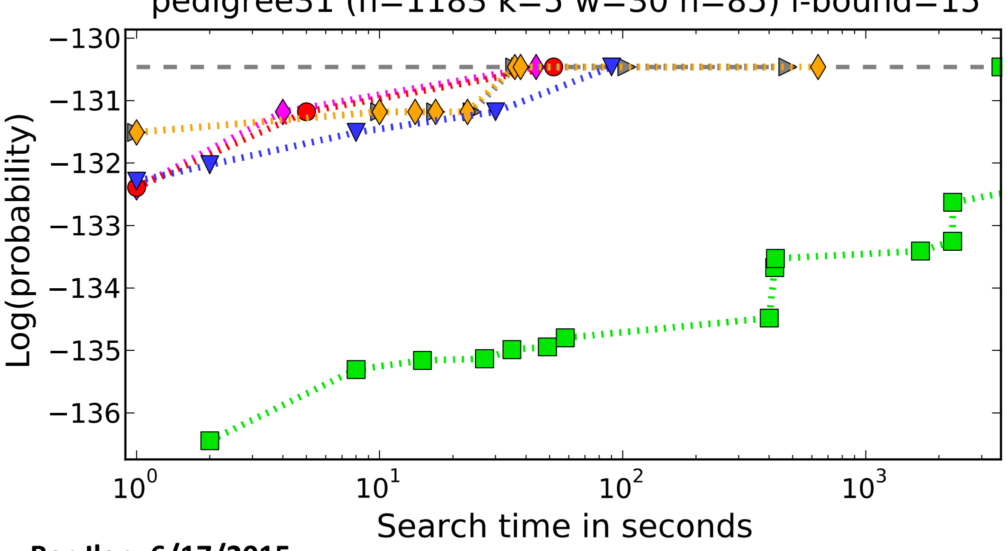
2006	🥇 (aolib)	
2008	🥈 (aolib)	
2011	🥇 (daoopt)	
2014	🥈 (daoopt)	🥇 (daoopt) 🥈 (merlin)
	MPE/MAP	**MMAP**

Empirical Evaluation: Haplotype problems

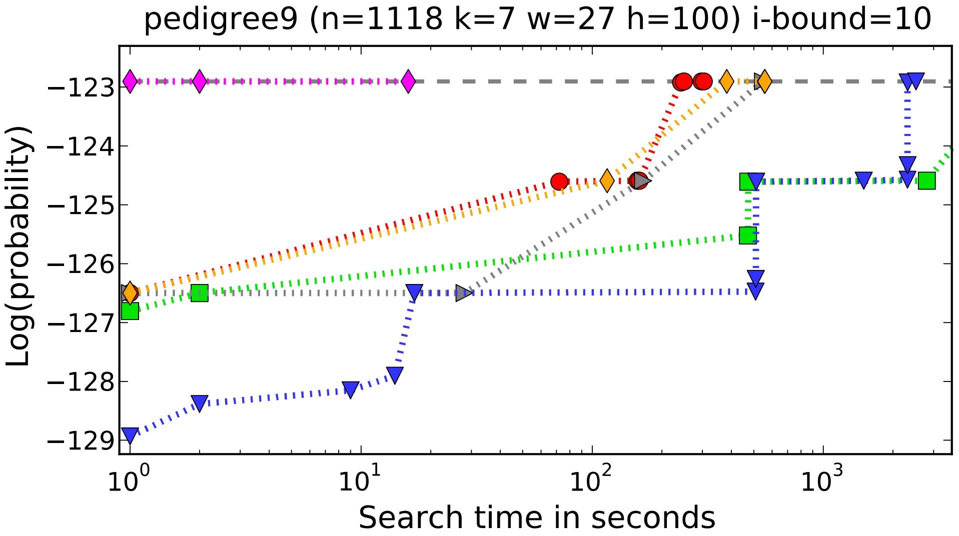


Time bound – 24 h

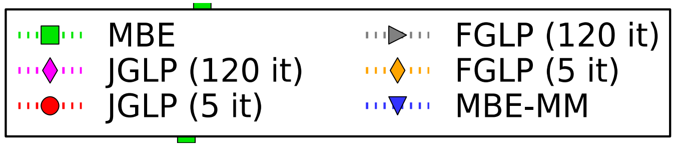


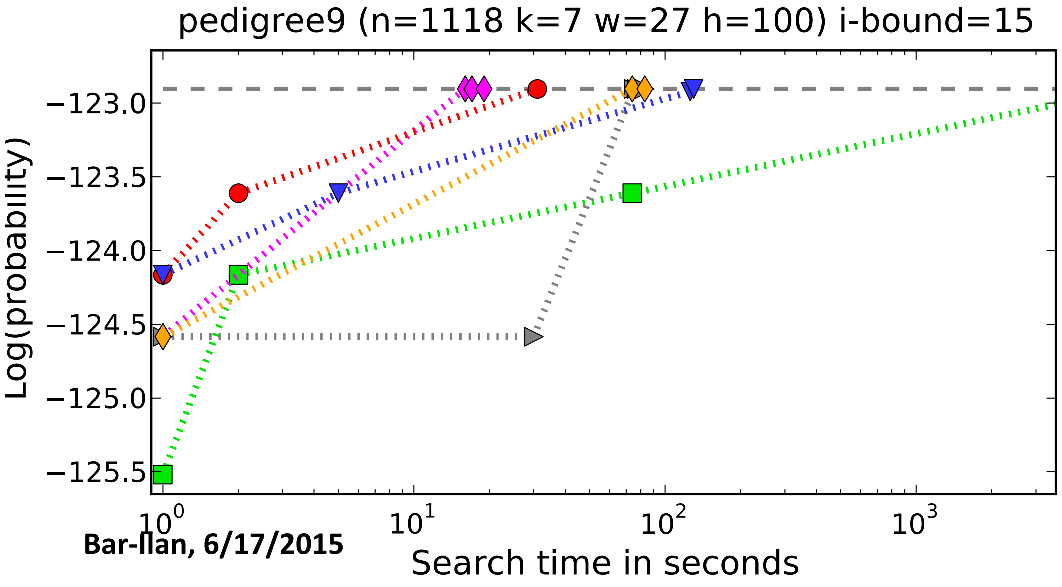


Empirical Evaluation: Haplotype problems



Time bound – 24 h





Marginal Map results

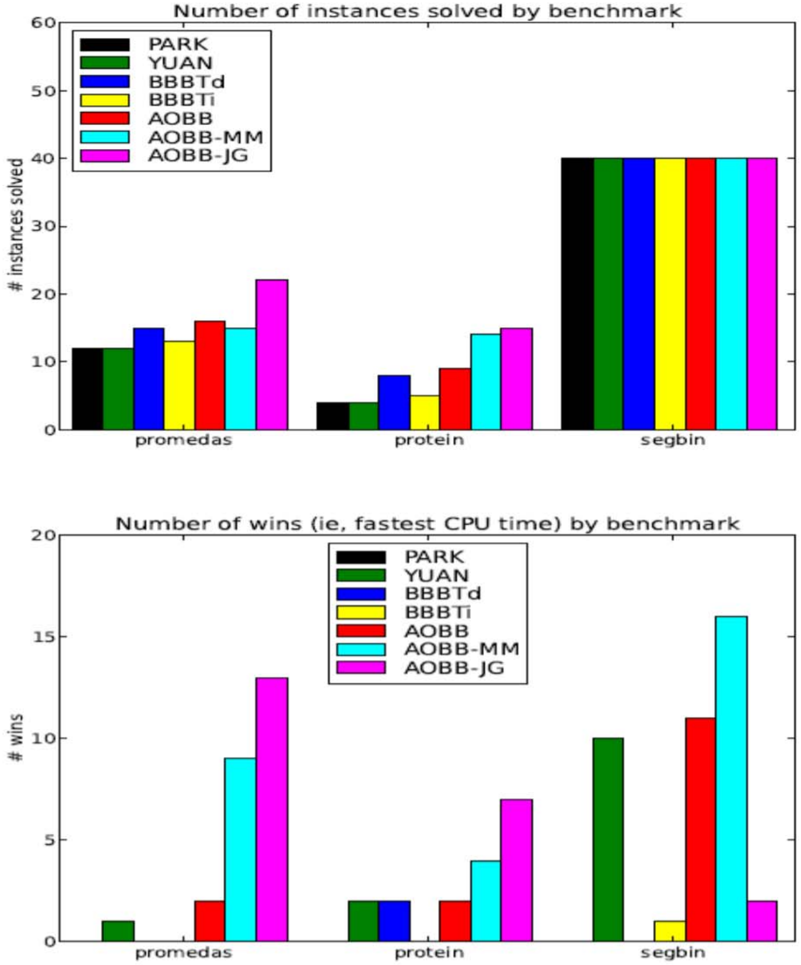


Figure 5: Number of instances solved (top) and number of wins (bottom) by benchmark.

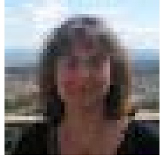
Summary and Future work

- I have shown the primal graph of a graphical model can:
  - □ suggest effective relaxation and heuristics
  - □ Suggest a compact search graph
  - □ Can accommodate vibrational schemes
- Yields one of the best solvers
- *ILP and Boolean methods can be included.*
- *Current work: anytime scheme minimizing the upper-lower gap at termination anytime*
- *Anytime summation solvers*

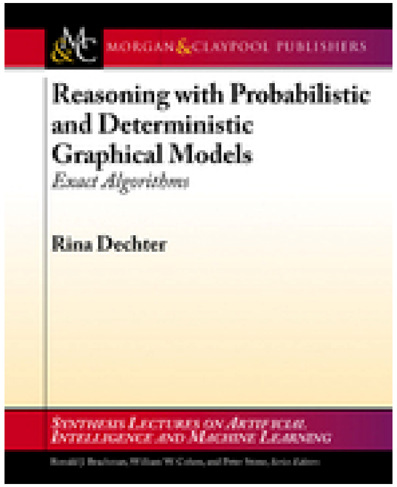
Thanks You!

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



MORGAN & CLAYPOOL PUBLISHERS

Reasoning with Probabilistic and Deterministic Graphical Models

Exact Algorithms

Rina Dechter

Synthesis Lectures on Artificial Intelligence and Machine Learning

Ronald J. Brachman, William W. Cohen, and Peter Stone, Series Editors