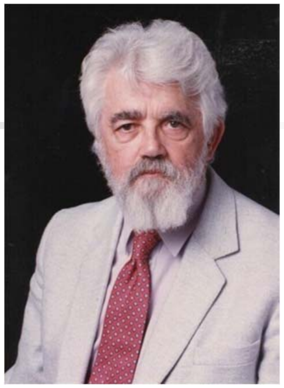
Problem Solving with Graphical Models

Rina Dechter
Donald Bren School of Computer Science
University of California, Irvine, USA

Dagshtul, 2011

What is Artificial Intelligence

(John McCarthy , Basic Questions)



- **What is artificial intelligence?**
- It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

- **Yes, but what is intelligence?**
- Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines.

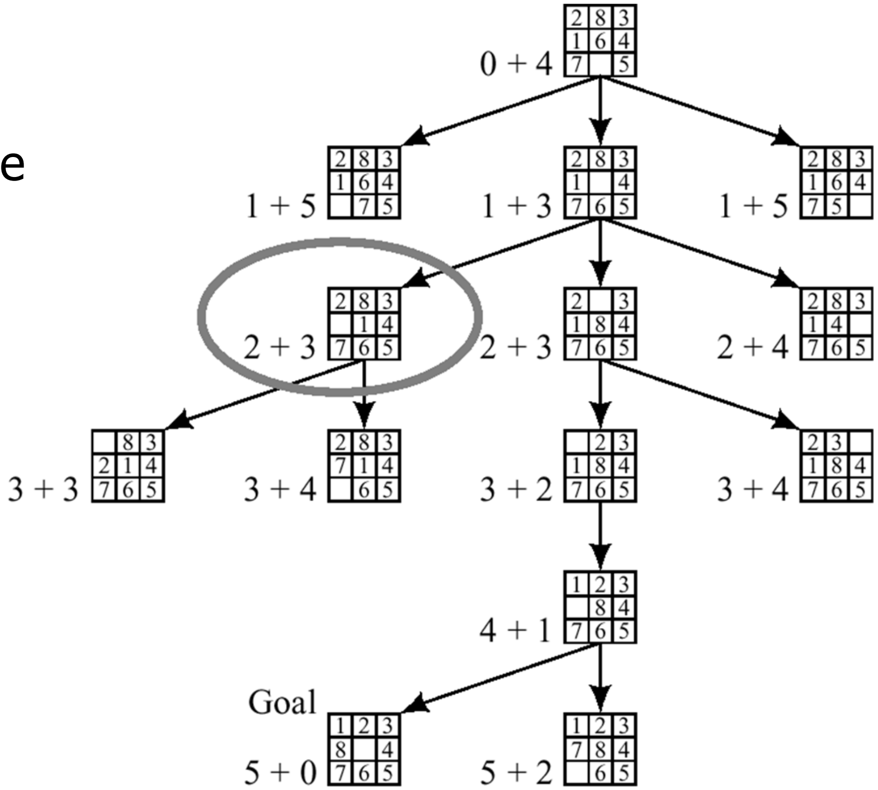
- **Isn't there a solid definition of intelligence that doesn't depend on relating it to human intelligence?**
- Not yet. The problem is that we cannot yet characterize in general what kinds of computational procedures we want to call intelligent. We understand some of the mechanisms of intelligence and not others.

- **More in: http://www-formal.stanford.edu/jmc/whatisai/node1.html**

Mechanical Heuristic generation

Observation: People generate heuristics by consulting simplified/relaxed models.
Context: Heuristic search (A*) of state-space graph (Nillson, 1980)
Context: Weak methods vs. strong methods
Domain knowledge: Heuristic function

h(n):Heuristic underestimate the best cost from n to the solution



The Simplified models Paradigm

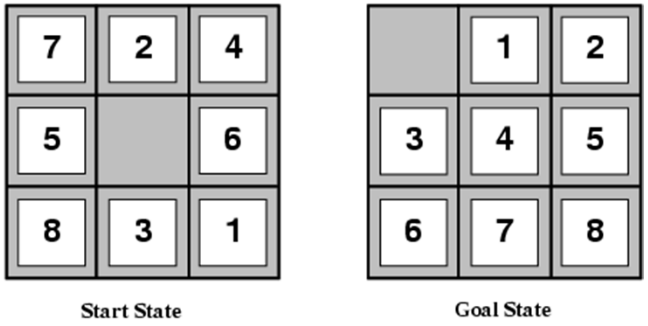
Pearl 1983 (*On the discovery and generation of certain Heuristics, 1983, AI Magazine, 22-23*) : "knowledge about easy problems could serve as a heuristic in the solution of difficult problems, i.e., that it should be possible to manipulate the representation of a difficult problem until it is approximated by an easy one, solve the easy problem, and then use the solution to guide the search process in the original problem."

The implementation of this scheme requires three major steps:
a) simplification,
b) solution, and
c) advice generation.

Simplified = relaxed is appealing because:
1. implies admissibility, monotonicity,
2. explains many human-generated heuristics (15-puzzle, traveling salesperson)

"We must have a simple **a-priori** criterion for deciding when a problem lends itself to easy solution."

Systematic relaxation of STRIPS



STRIPS (**St**anford **R**esearch **I**nstitute
Problem **S**olver, Nillson and Fikes 1971)
action representation:

Move(x,c1,c2)
Precond list: on(x1,c1), clear(c2), adj(c1,c2)
Add-list: on(x1,c2), clear(c1)
Delete-list: on(x1,c1), clear(c2)

Relaxation (Sacerdoti, 1974): Remove literals from the precondition-list:
1. clear(c2), adj(c2,c3) → #misplaced tiles
2. Remove clear(c2) → manhatten distance
3. Remove adj(c2,c3) → h3, a new procedure that transfer to the empty location a tile appearing there in the goal

But the main question remained:
“Can a program tell an easy problem from a hard one without actually solving?” (Pearl 1984, Heuristics)

Easy = Greedily solved?

Pearl, 84: Most easy problems we encounter are solved by **"greedy" hill-climbing methods without backtracking"** and that the features that make them amenable to such methods is their "decomposability"

The question now:
Can we recognize a greedily solved STRIPS problem?"

Freuder, JACM 1982 : “A sufficient condition for backtrack-free search”

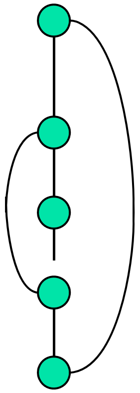
Whow! Backtrack-free is greedy!
I read Montanari (1974),I read mackworth, (1977)
Got absorbed…

Sufficient condition (Freuder 82):

1. Trees (width-1) and arc-consistency implies backtrack-free
2. Width=i and (i+1)-consistency implies backtrack-free search

If 3-consistent
no deadends

W=2



W=1

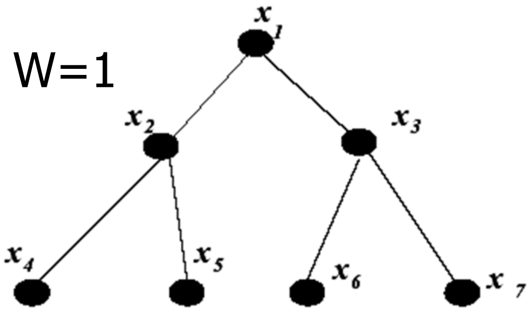


Figure 4.10: A tree network

Arc-consistent
No dead-ends

This moved me to constraint network and ultimately to graphical models.
But: Is it indeed the case that heuristics are generated by simplified Models?

Outline of the talk

- Introduction to graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - How can we predict problem instance hardness?

Outline

- What are graphical models
- Overview of Inference Search and their hybrids

- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking

What are Graphical Models

- A way to represent *global* knowledge, mostly declaratively, using small local pieces of functions/relations. Combined, they give a global view of a world about which we want to reason, namely to answer queries.

- Different types of graphs can capture variable interaction through the local functions.

- Because representation is modular, reasoning can be modular too.

Constraint Networks

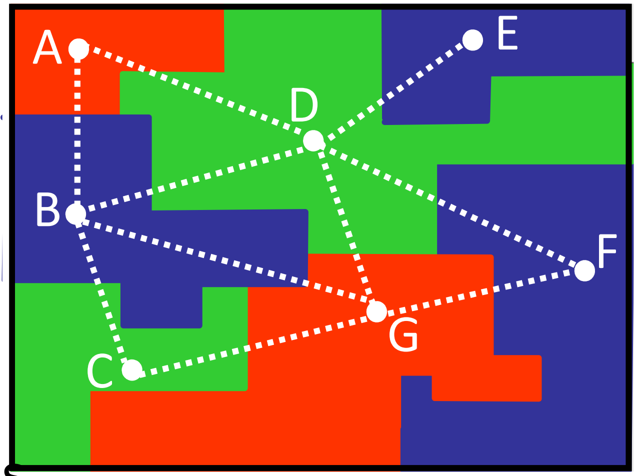
Map coloring

Variables: countries (A B C etc.)

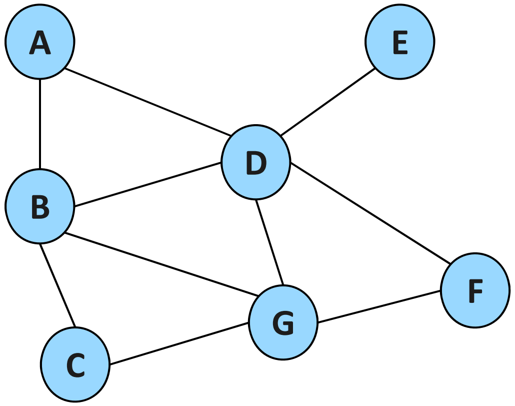
Values: colors (red green blue)

Constraints: $\mathbf{A \neq B}$, $\mathbf{A \neq D}$, $\mathbf{D \neq E}$, …

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

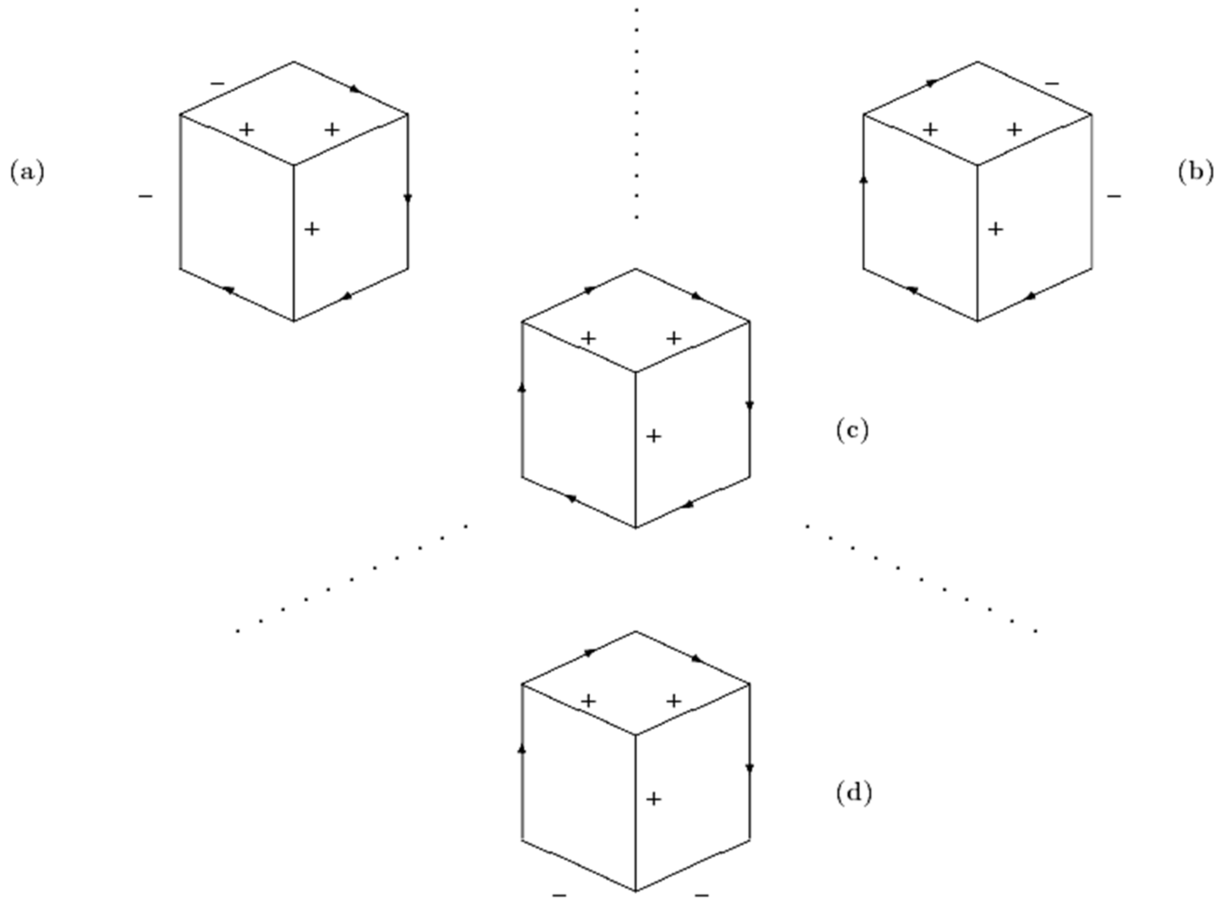


Constraint graph

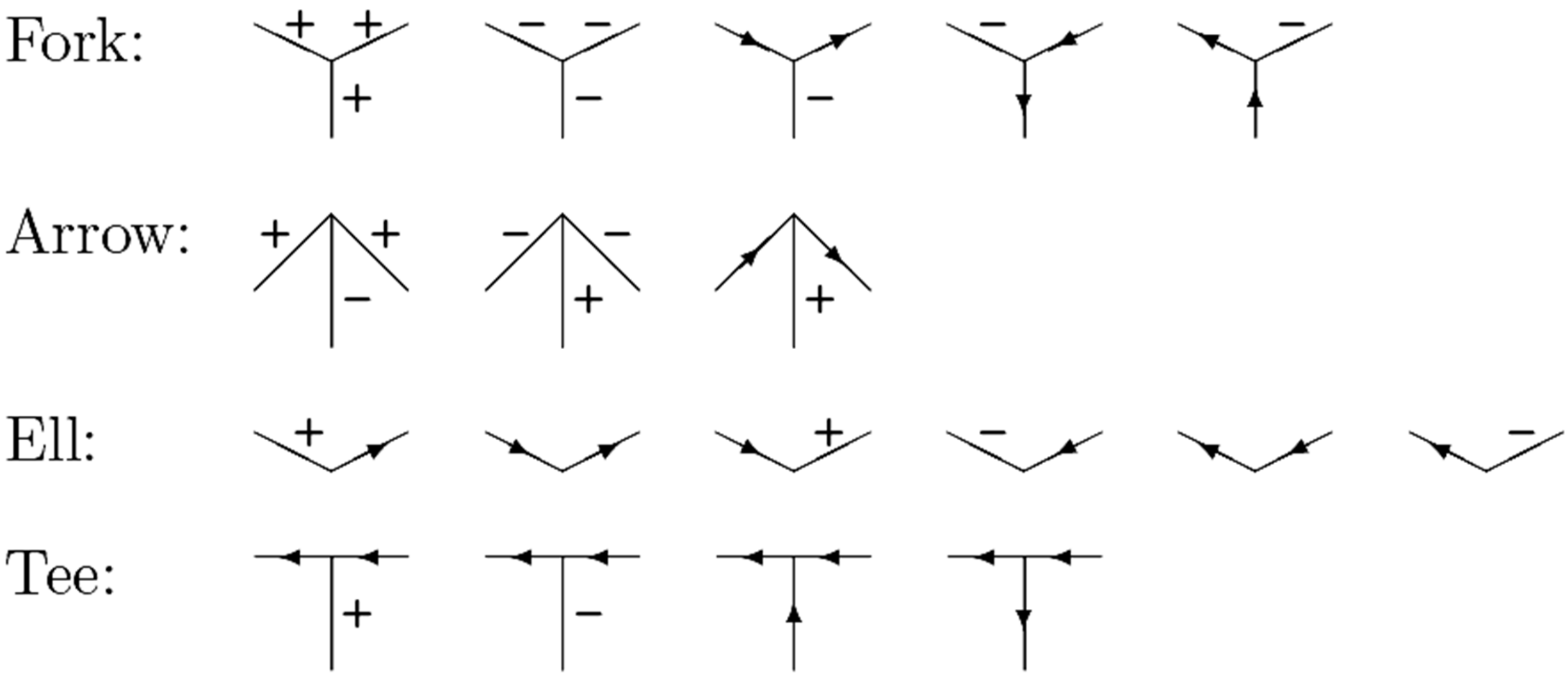


Global view: all solutions.
Tasks: Is there a solution?, find one, fine all, count all

Three dimensional interpretation of 2 dimentional drawing



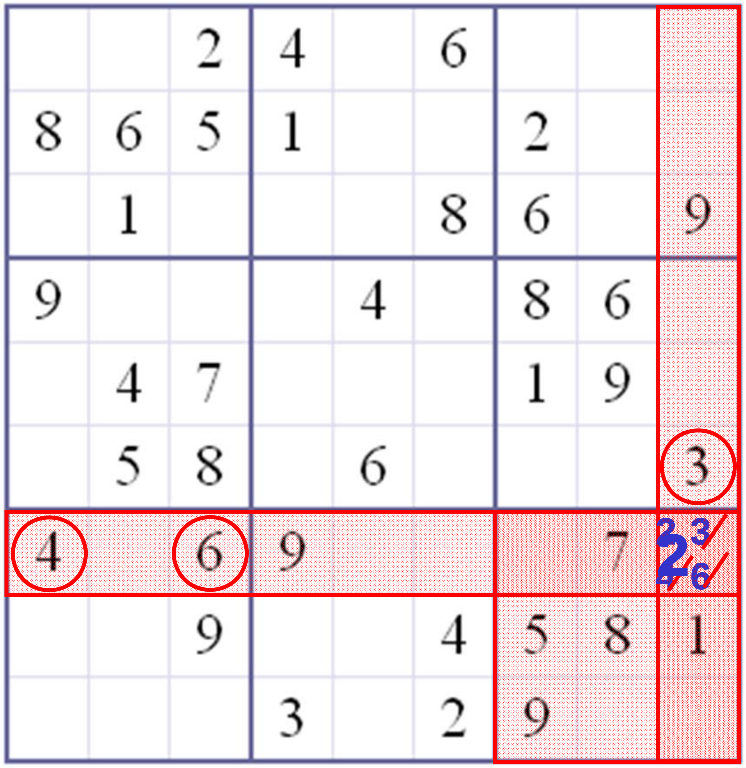
Huffman-Clowes junction labelings (1975)



Sudoku –Constraint Satisfaction

- **Constraint**
- **Propagation**

- **Inference**



		2	4		6			
8	6	5	1			2		
	1				8	6		9
9				4		8	6	
	4	7				1	9	
	5	8		6				3
4		6	9				7	2
		9			4	5	8	1
			3		2	9		

- **Variables**: empty slots

- **Domains** = {1,2,3,4,5,6,7,8,9}

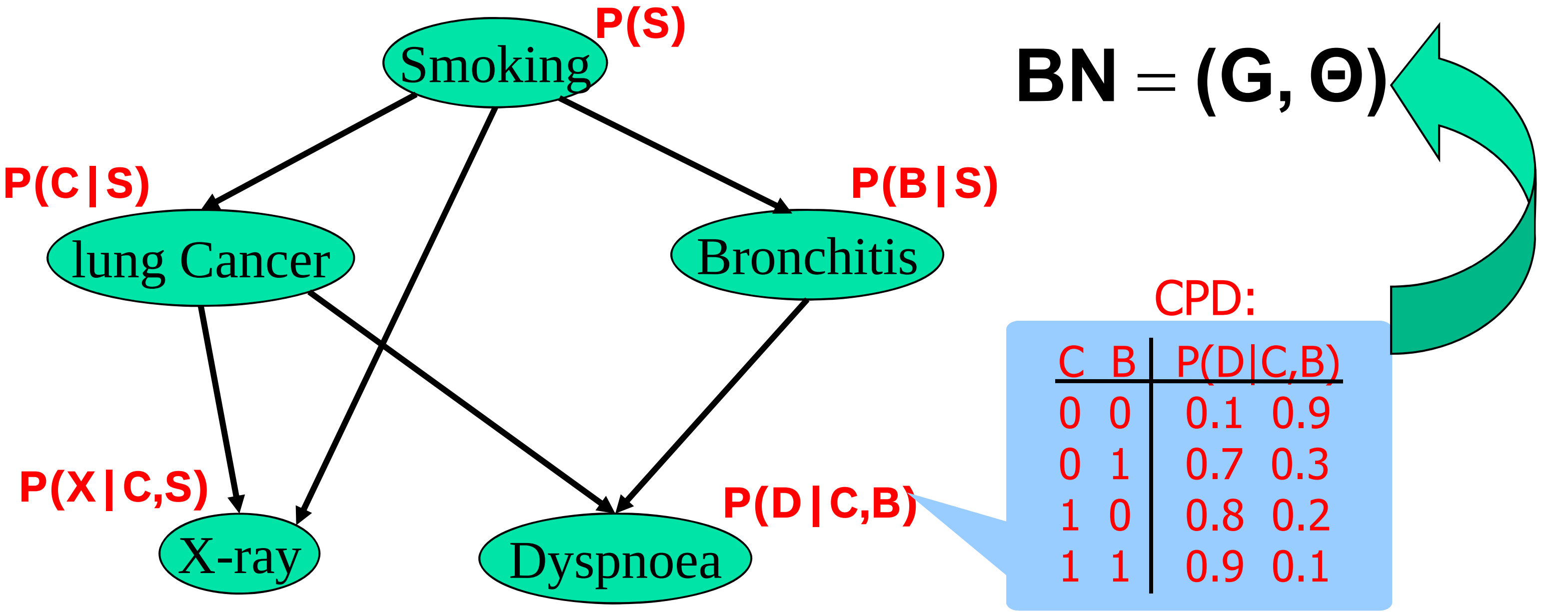
- **Constraints**:
  - 27 all-different

Each row, column and major block must be alldifferent

“Well posed” if it has unique solution: 27 constraints

Bayesian Networks

(Pearl, 1988)



$\mathbf{BN} = (\mathbf{G}, \mathbf{\Theta})$

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

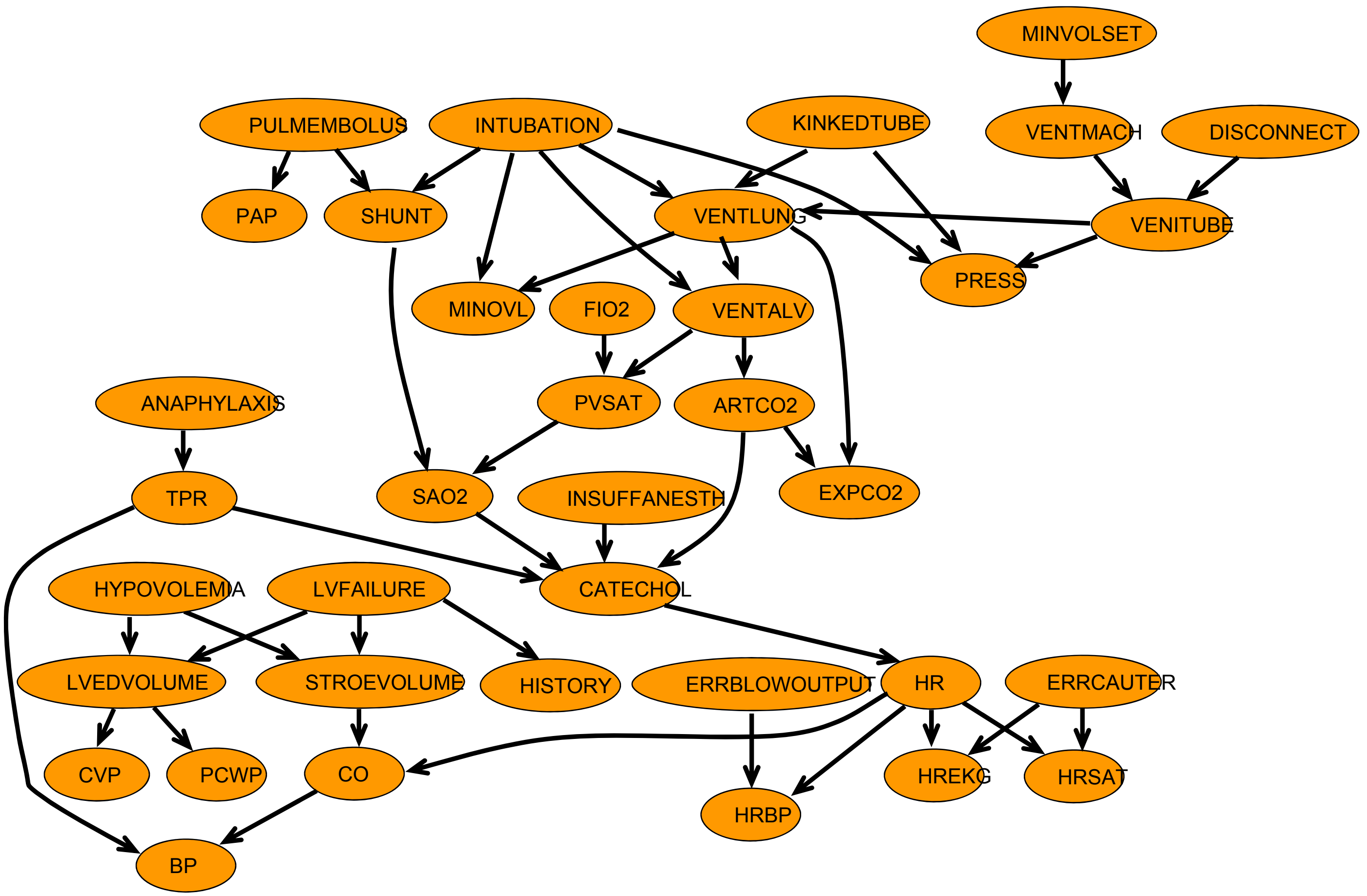
Gloabl view: $P(S, C, B, X, D) = P(S)\, P(C|S)\, P(B|S)\, P(X|C,S)\, P(D|C,B)$

Belief Updating, Most probable tuple (MPE)

= find argmax $P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B) =?$

Monitoring Intensive-Care Patients

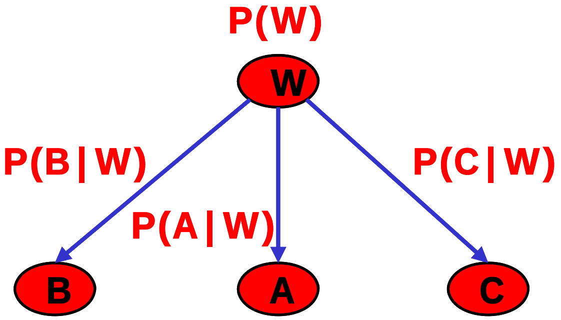
The "alarm" network - 37 variables, 509 parameters (instead of 2^{37})



Mixed Probabilistic and Deterministic networks

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

PN



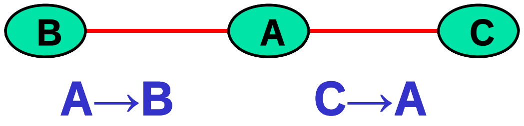
W	A	P(A|W)
good	0	.01
good	1	.99
bad	0	.1
bad	1	.9

Semantics?

Algorithms?

If Alex goes, then Becky goes:
If Chris goes, then Alex goes:

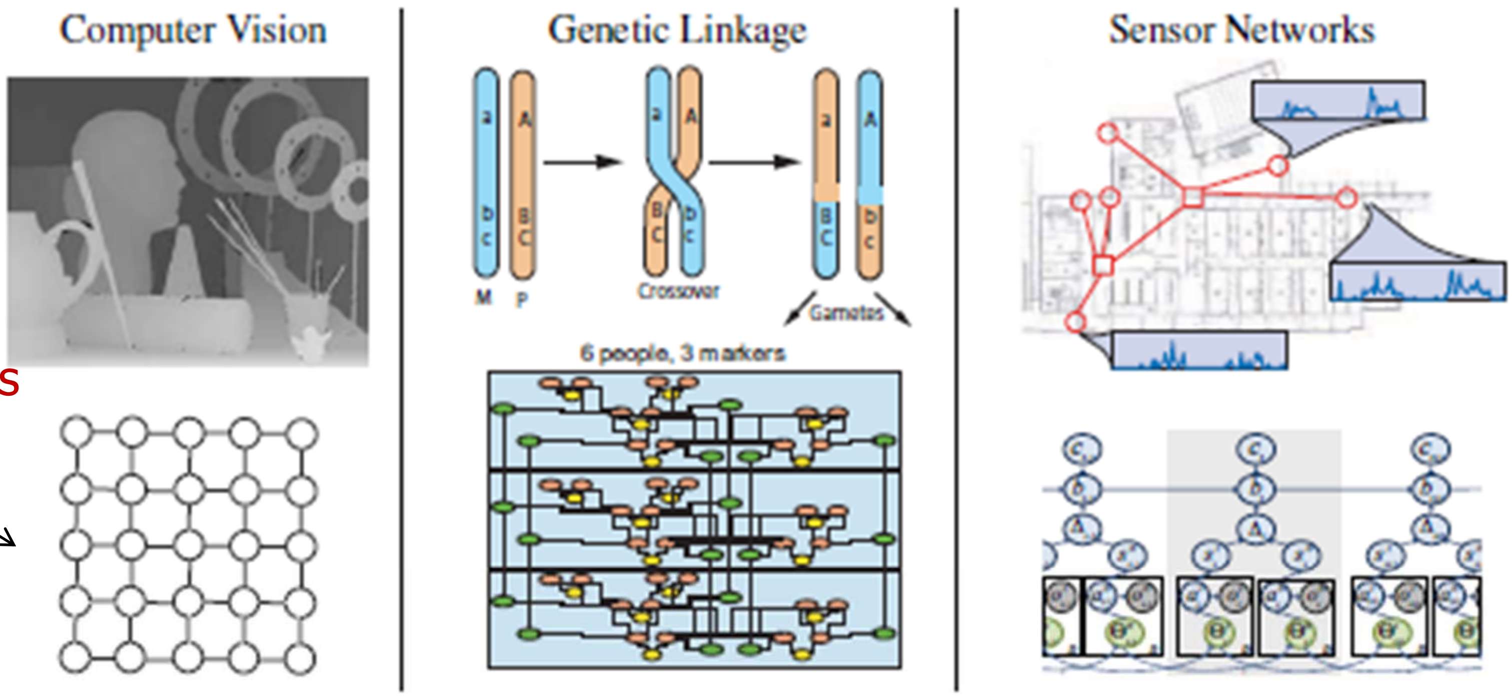
CN



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

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

Applications



Computer Vision

Genetic Linkage

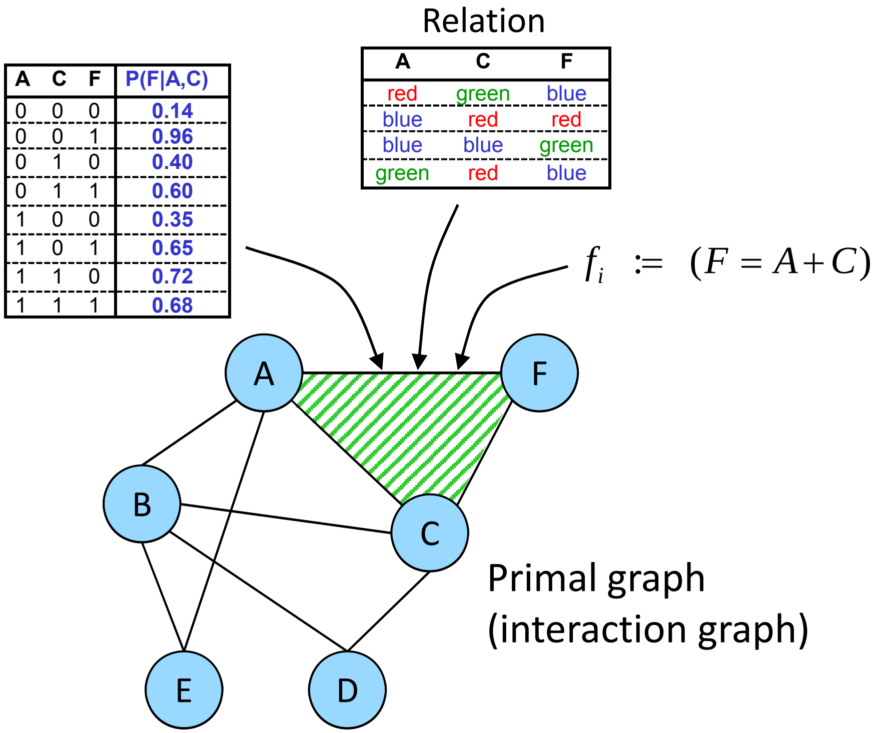
Sensor Networks

Markov networks

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.

Graphical Models

- A graphical model **(X,D,F)**:
  - **X** = $\{X_1,\ldots X_n\}$ variables
  - **D** = $\{D_1, \ldots D_n\}$ domains
  - **F** = $\{f_1,\ldots,f_m\}$ functions
- Operators:
  - combination
  - elimination (projection)



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

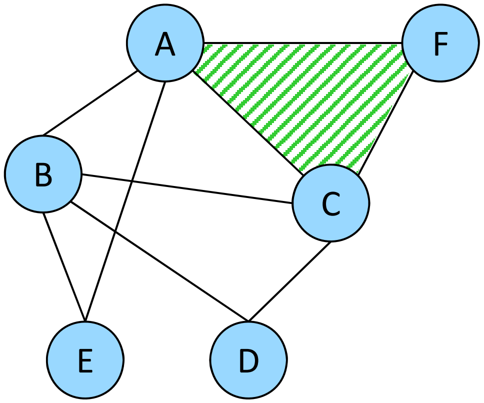
A	C	F
red	green	blue
blue	red	red
blue	blue	green
green	red	blue

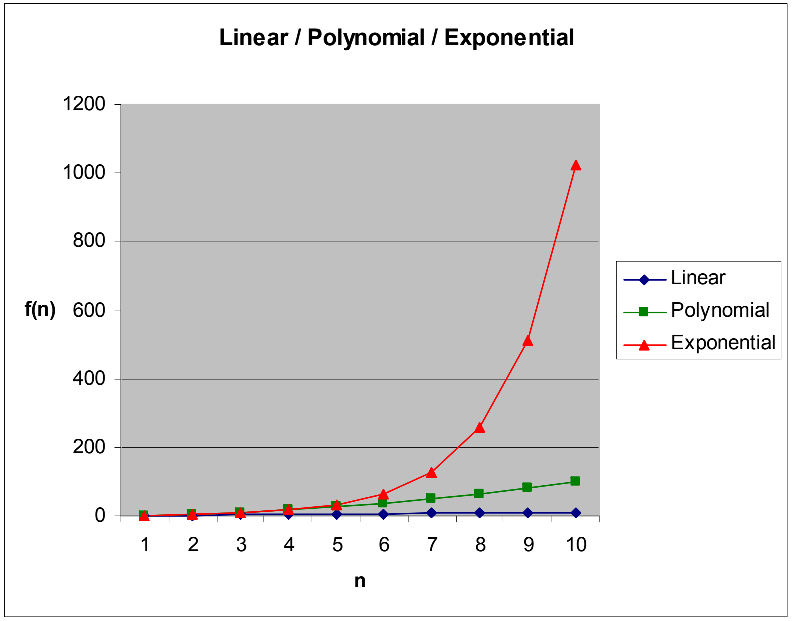
$$f_i \;:=\; (F = A + C)$$

Primal graph (interaction graph)

Complexity of Reasoning Tasks

- Constraint satisfaction
- Counting solutions
- Combinatorial optimization
- Belief updating
- Most probable explanation
- Decision-theoretic planning





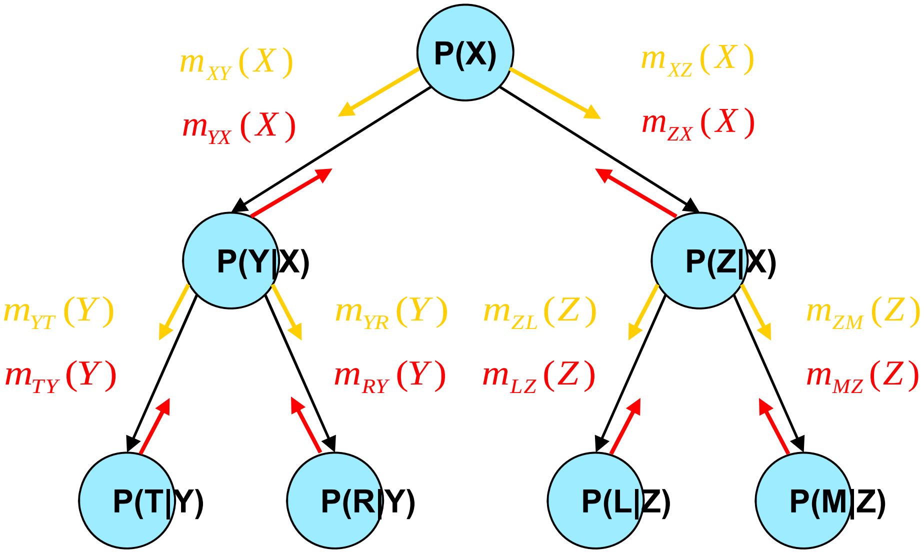
Reasoning is computationally hard

Complexity is
Time and space(memory) exponential

Tree-solving is easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



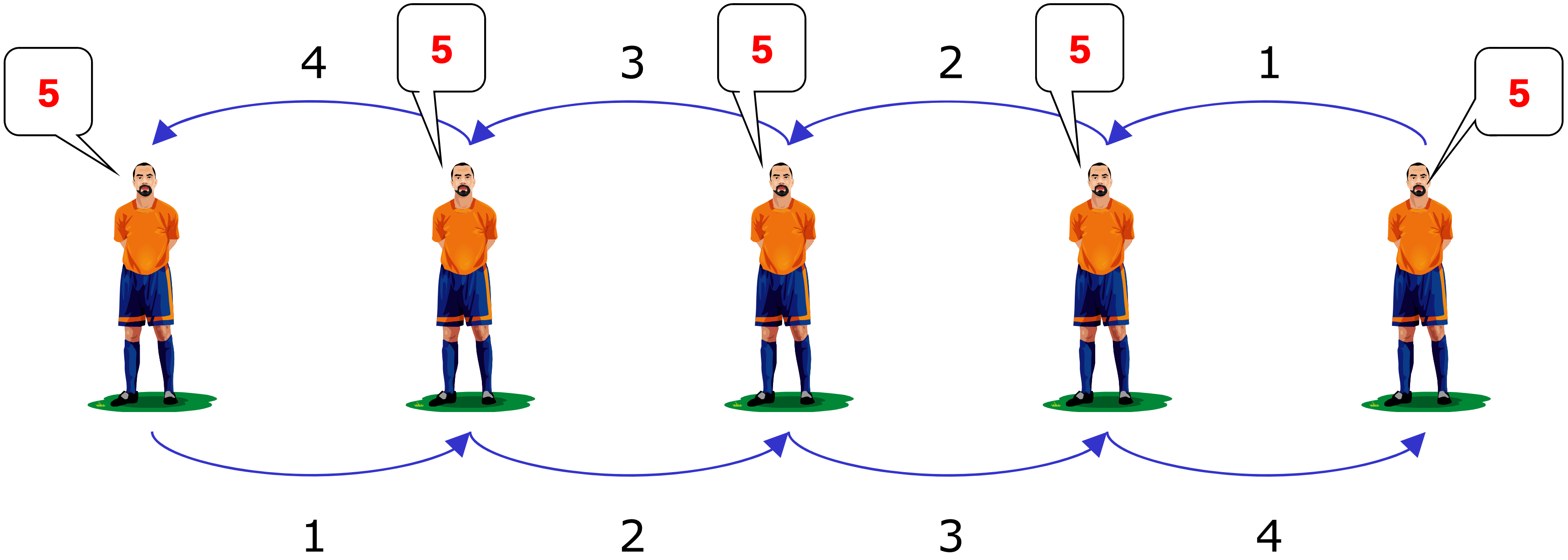
MPE (max-prod)

#CSP (sum-prod)

Trees are processed in linear time and memory

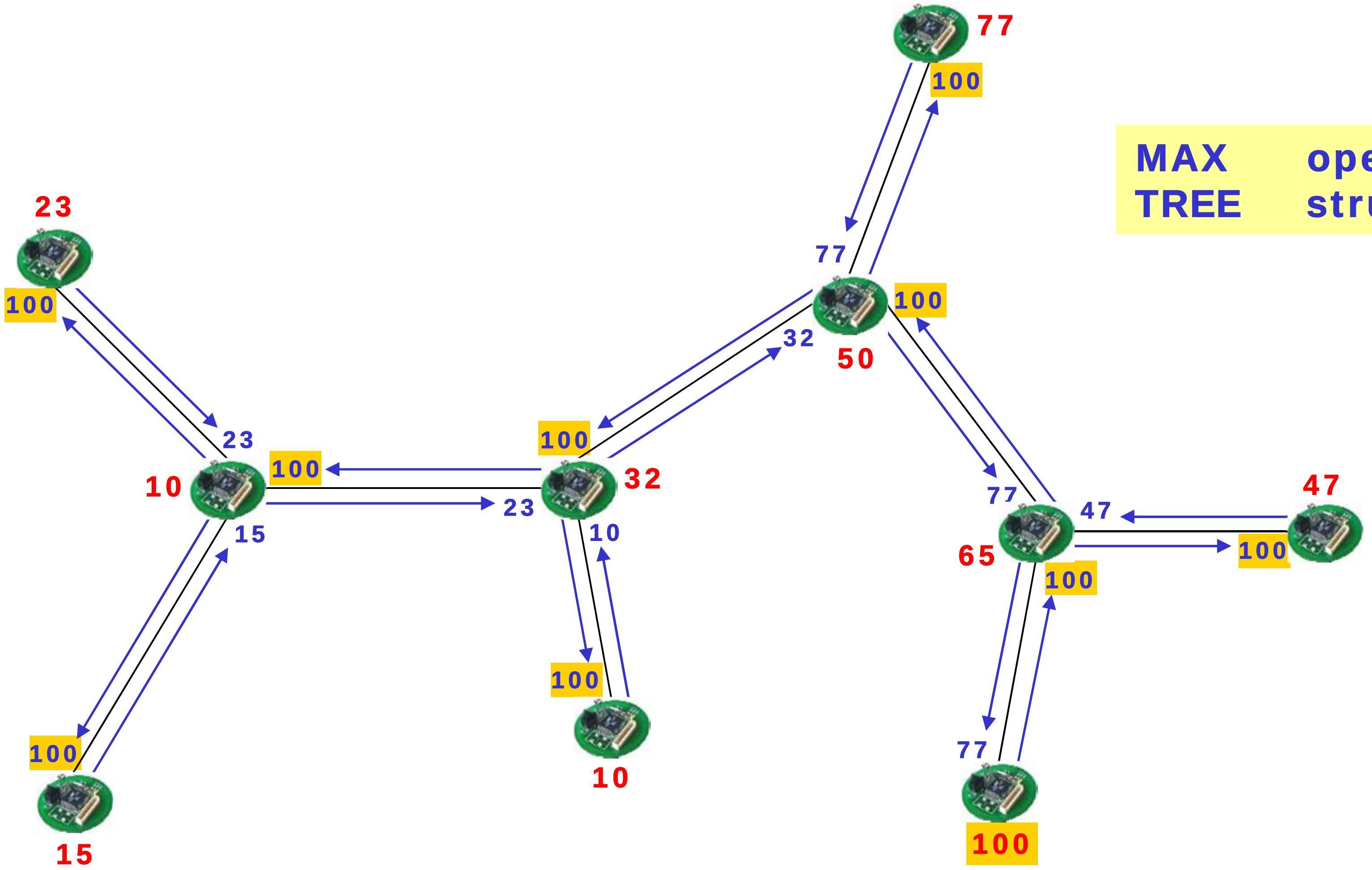
Counting

SUM operator
CHAIN structure



How many people?

Maximization



MAX operator
TREE structure

What is the maximum?

Min-Cost Assignment

12”	14”	15”
75	80	105

+

12”	14”	15”
30	40	50

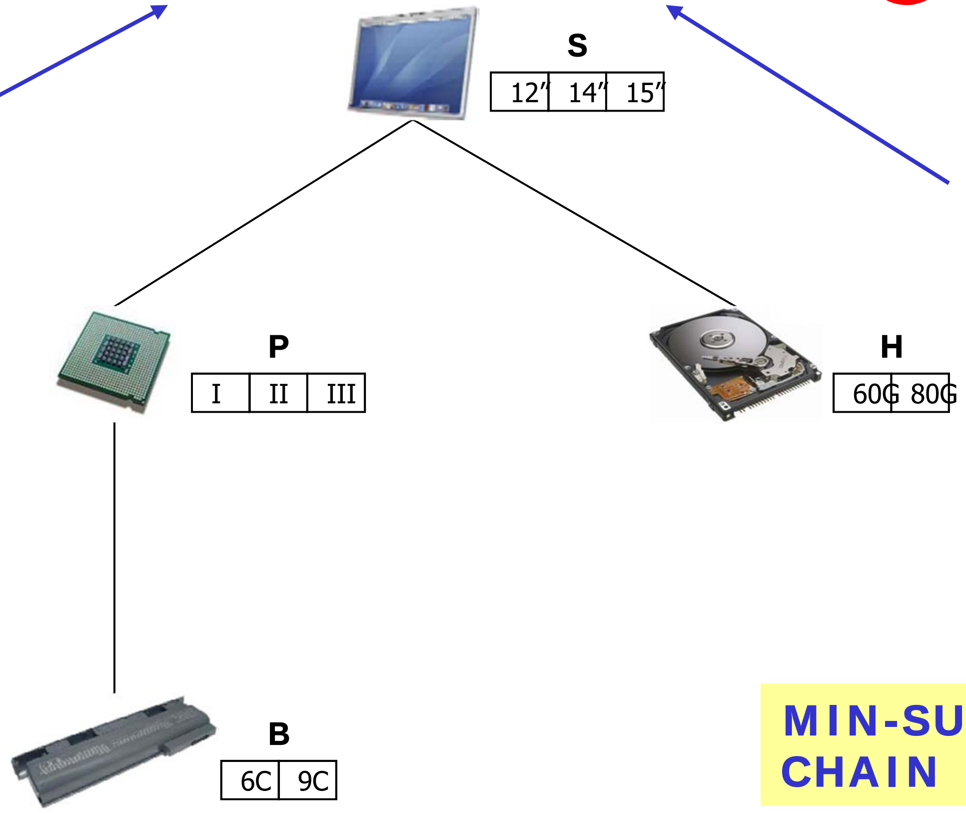
=

12”	14”	15”
105	120	155

	I	II	III
12”	**75**	∞	∞
14”	**80**	1600	1730
15”	∞	**105**	1840

+

I	II	III
30	40	60



	60G	80G
12”	**30**	50
14”	**40**	45
15”	**50**	∞

	6C	9C
I	**30**	50
II	**40**	55
III	∞	**60**

MIN-SUM operators
CHAIN structure

What is minimum cost configuration?

Belief Updating

SUM-PROD operators
POLY-TREE structure

H	P(H)
0	.9
1	.1

F	P(F)
0	.99
1	.01

*

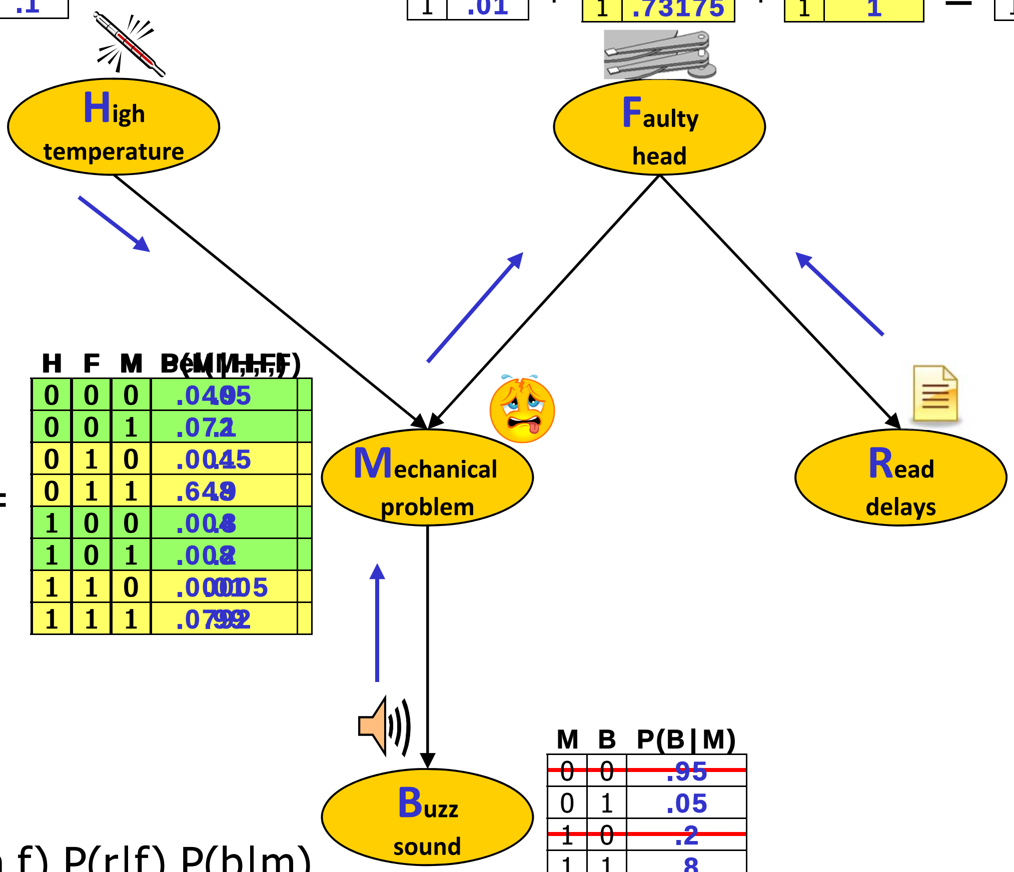
F	h_3(F)
0	.1245
1	.73175

*

F	h_4(F)
0	1
1	1

=

F	P(F,B=1)
0	.123255
1	.073175



H	F	M	P(M|H,F)
0	0	0	.9
0	0	1	.1
0	1	0	.1
0	1	1	.9
1	0	0	.8
1	0	1	.2
1	1	0	.01
1	1	1	.99

*

M	h_1(M)
0	.05
1	.8

*

H	h_2(H)
0	.9
1	.1

=

H	F	M	Bel(M|H,F)
0	0	0	.0405
0	0	1	.072
0	1	0	.0045
0	1	1	.648
1	0	0	.004
1	0	1	.008
1	1	0	.00005
1	1	1	.0792

F	R	P(R|F)
0	0	.8
0	1	.2
1	0	.3
0	1	.7

M	B	P(B|M)
~~0~~	~~0~~	~~.95~~
0	1	.05
~~1~~	~~0~~	~~.2~~
1	1	.8

P(h,f,r,m,b) = P(h) P(f) P(m|h,f) P(r|f) P(b|m)

P(F | B=1) = ?

P(B=1) = .19643

Probability of evidence

P(F=1|B=1) = .3725

Updated belief

Belief Propagation

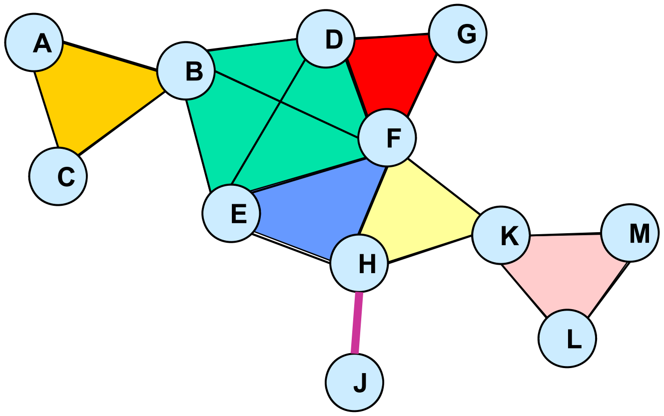
(Pearl, 1988)

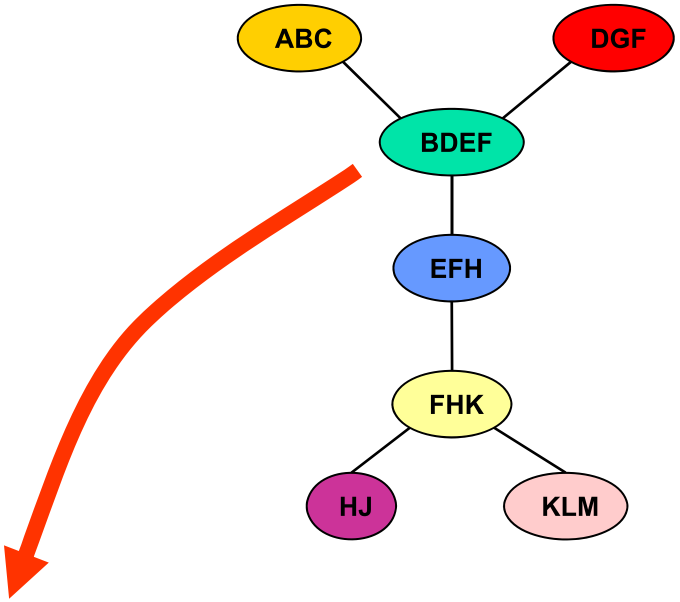
- Instances of tree message passing algorithm
- Exact for trees
- Linear in the input size
- Importance:
  - One of the first algorithms for inference in Bayesian networks
  - Gives a cognitive dimension to its computations
  - Basis for conditioning algorithms for arbitrary Bayesian network
  - Basis for Loopy Belief Propagation (approximate algorithms)

Transforming into a Tree

- **By Inference (thinking)**
  - Transform into a single, equivalent tree of sub-problems
- **By Conditioning (guessing)**
  - Transform into many tree-like sub-problems.

Inference and Treewidth





Inference algorithm:

Time: exp(tree-width)

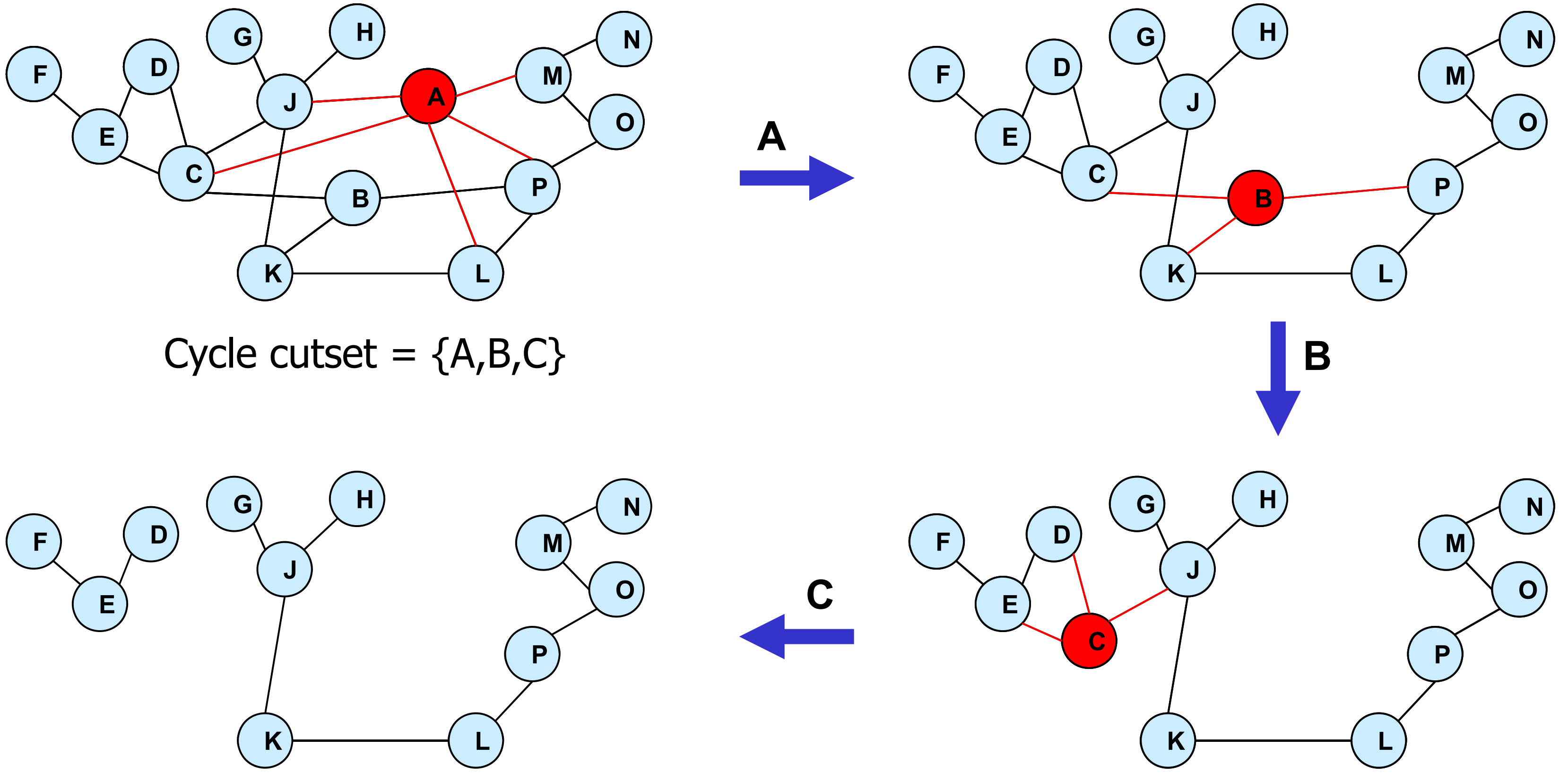
Space: exp(tree-width)

Key parameter: w*

treewidth = 4 - 1 = 3

treewidth = (maximum cluster size) - 1

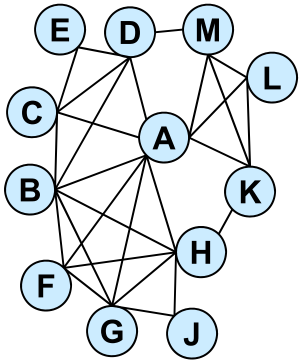
Conditioning and Cycle cutset



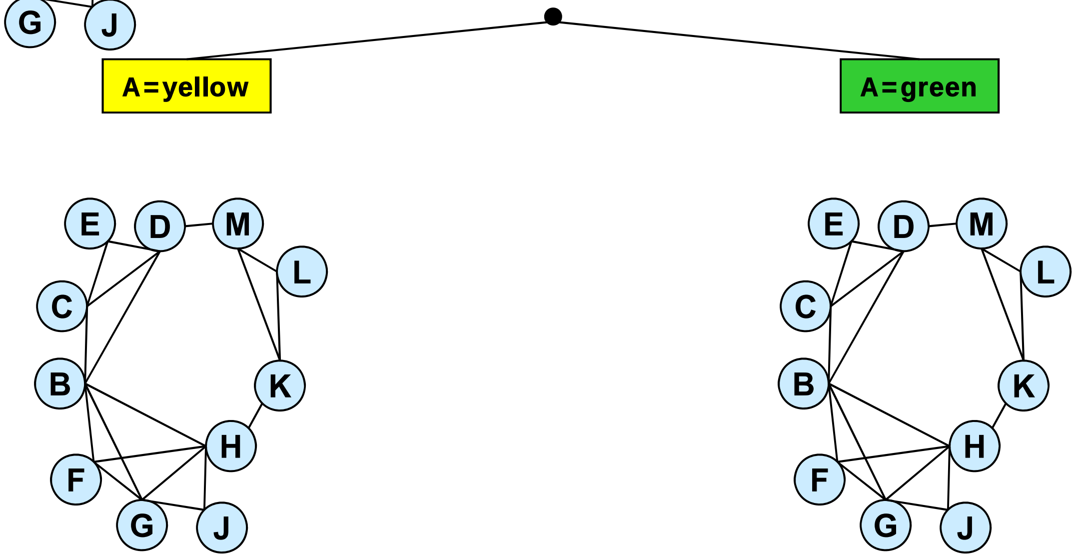
Cycle cutset = {A,B,C}

Search over the Cutset

Graph
Coloring
problem



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

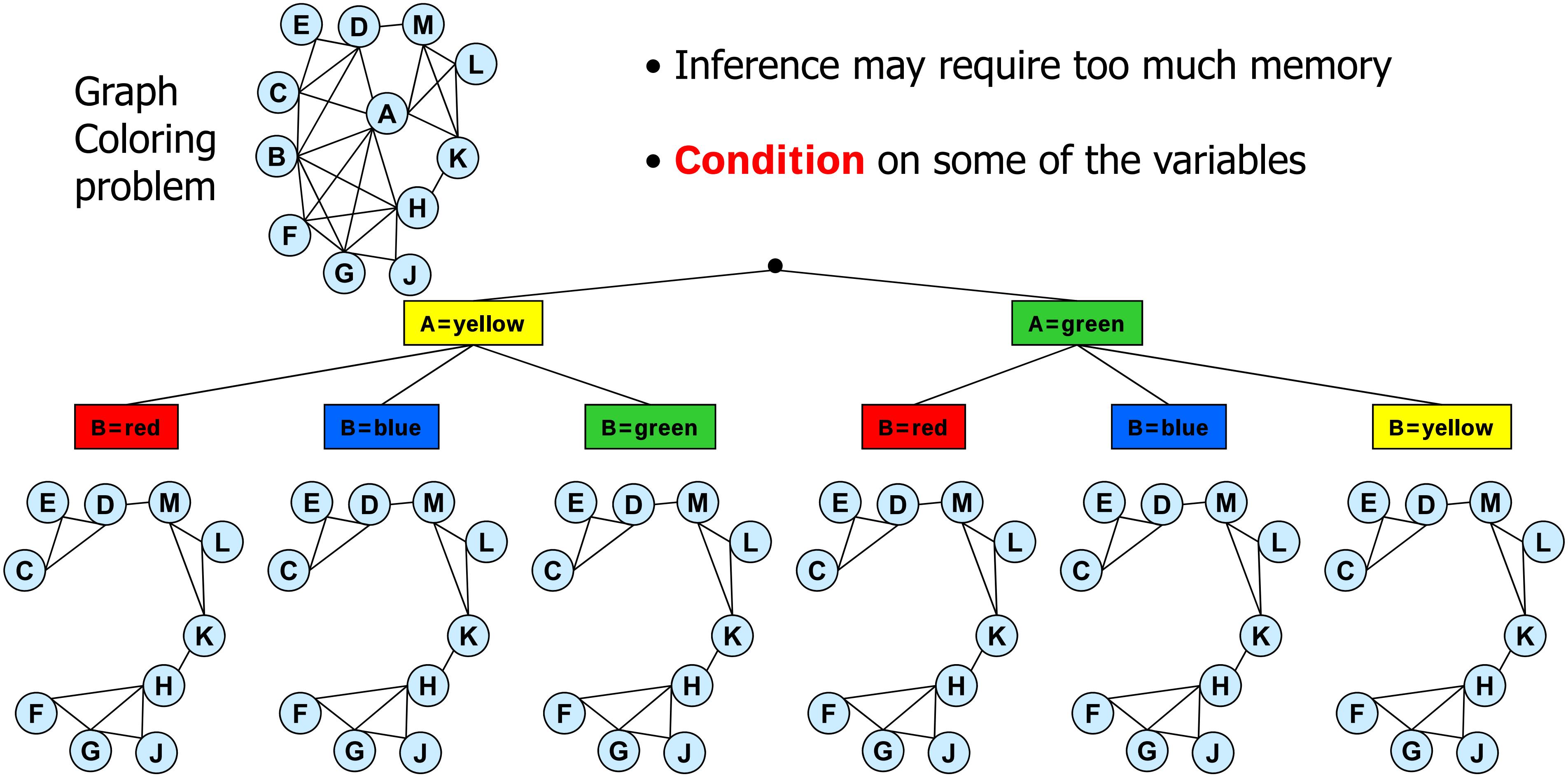


Search over the Cutset (cont)

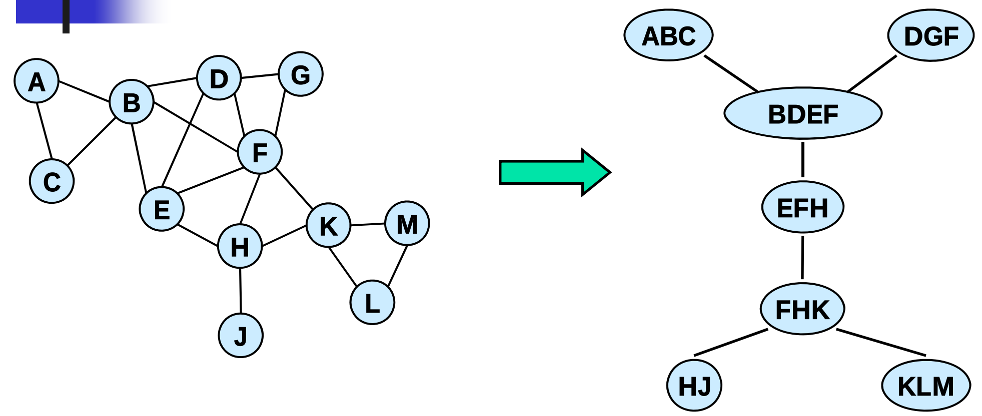
Key parameters:
the cycle-cutset, w-cutset
Complexity: exp(cutset-size)

Graph
Coloring
problem

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

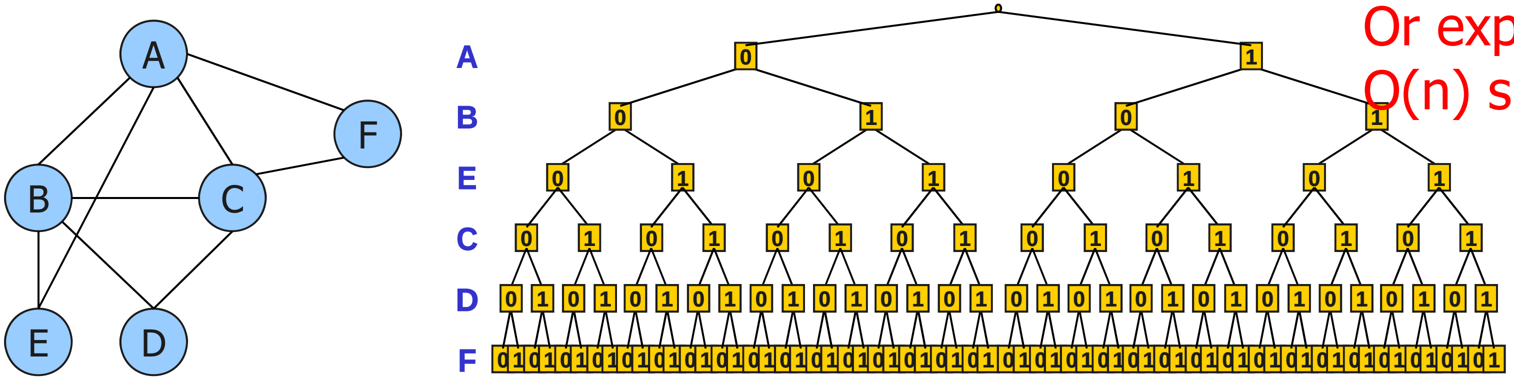


Inference vs Conditioning-Search



Inference

exp(treewidth) time/space

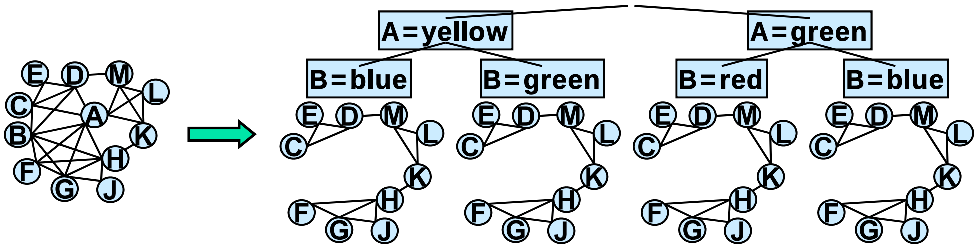


Search

exp(n) time

Or exp(pseudo-tree

O(n) space



Search+inference:

Space: exp(w of sub problem)

Time: exp(w+cutset(w))

Approximation Algorithms

- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:

- Bounding inference: Bounding the clusters by i-bound
    - mini-bucket(i) and bounded-i-consistency
    - Belief propagation and constraint propagation

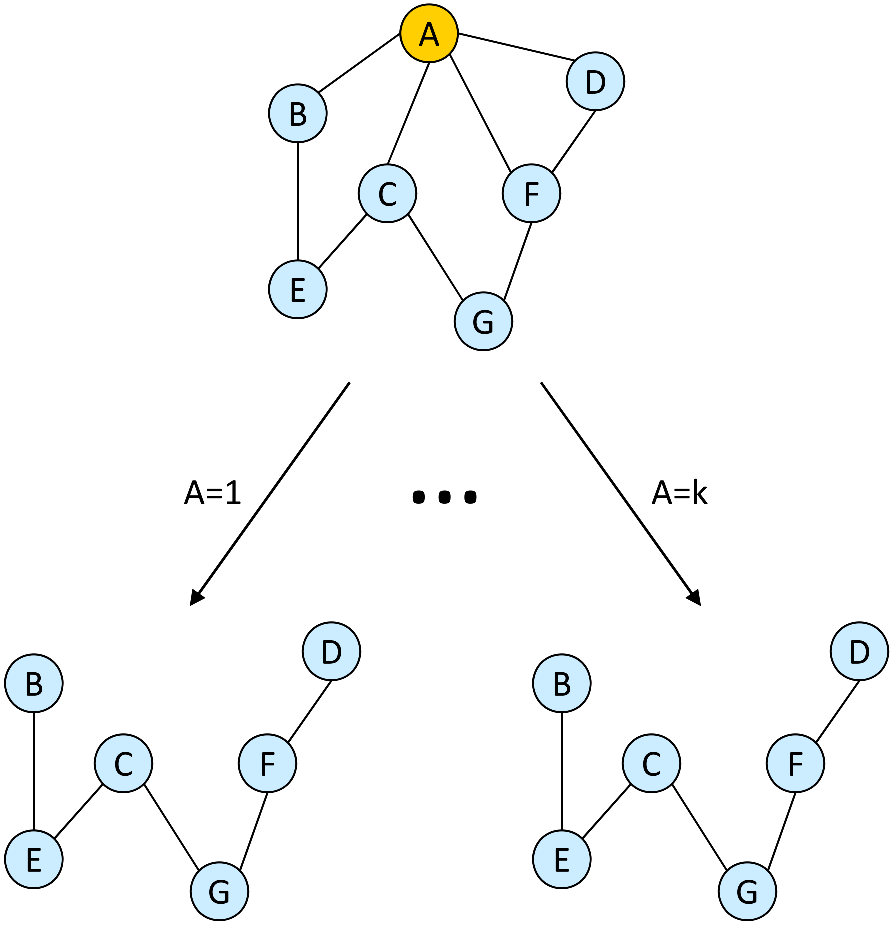
- Bounding search:
    - Sampling
    - Stochastic local search

- Hybrid of sampling and bounded inference

- Goal: an anytime scheme

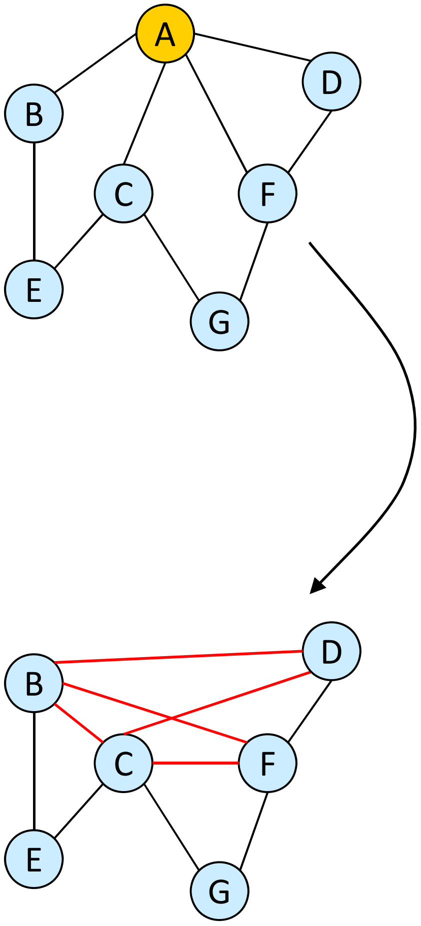
Search vs. Inference

Search (conditioning)



k “sparser” problems

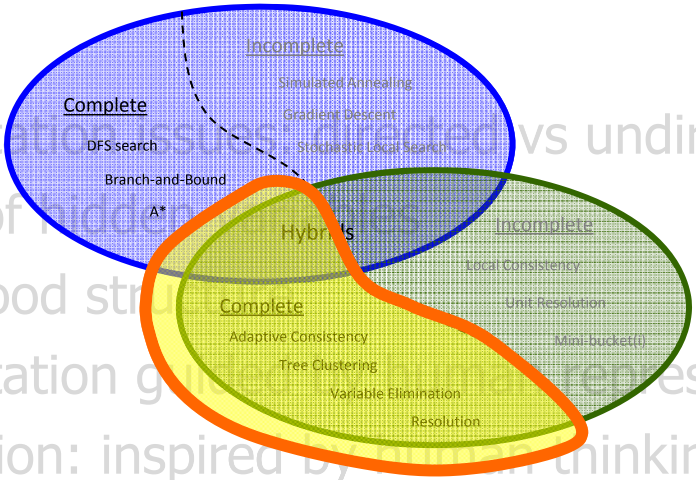
Inference (elimination)



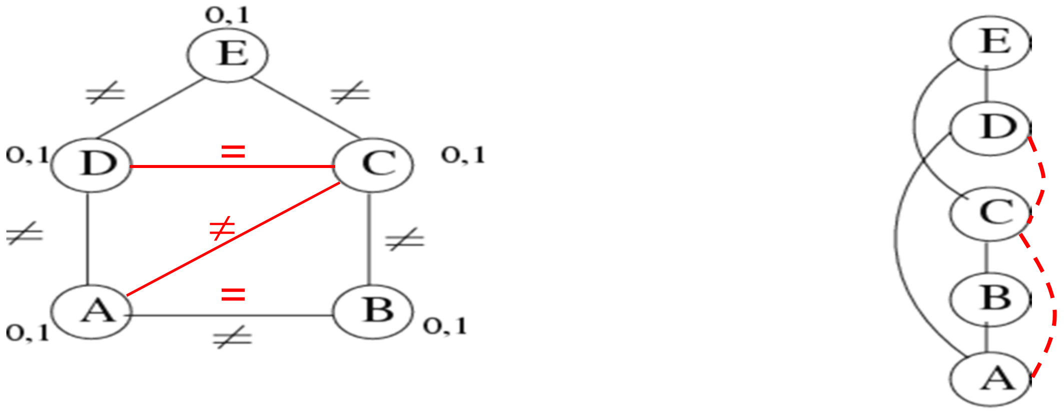
1 “denser” problem

Outline

- What are graphical models
- **Inference: Exact** and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking



Bucket Elimination, Variable elimination



Bucket E: $E \neq D$, $E \neq C$

Bucket D: $D \neq A$ **$D = C$**

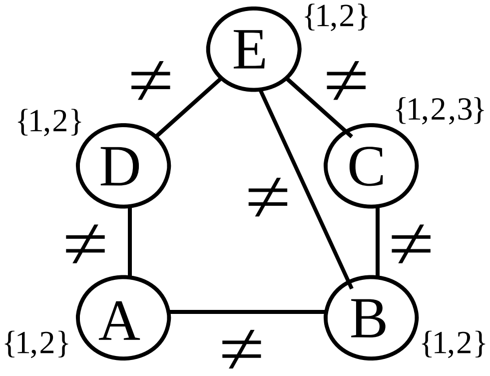
Bucket C: $C \neq B$ **$A \neq C$**

Bucket B: $B \neq A$ **$B = A$**

Bucket A: **contradiction**

Bucket Elimination

Adaptive Consistency (Dechter & Pearl, 1987)



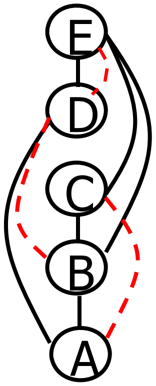
$Bucket(E):\ E \neq D,\ E \neq C,\ E \neq B$

$Bucket(D):\ D \neq A\ \|\ R_{DCB}$

$Bucket(C):\ C \neq B\ \|\ R_{ACB}$

$Bucket(B):\ B \neq A\ \|\ R_{AB}$

$Bucket(A):\ R_{A}$



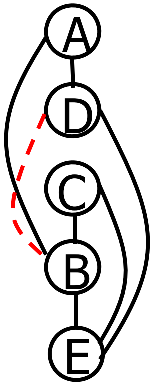
$Bucket(A):\ A \neq D,\ A \neq B$

$Bucket(D):\ D \neq E\ \|\ R_{DB}$

$Bucket(C):\ C \neq B,\ C \neq E$

$Bucket(B):\ B \neq E\ \|\ R^{D}_{BE},\ R^{C}_{BE}$

$Bucket(E):\ \|\ R_{E}$

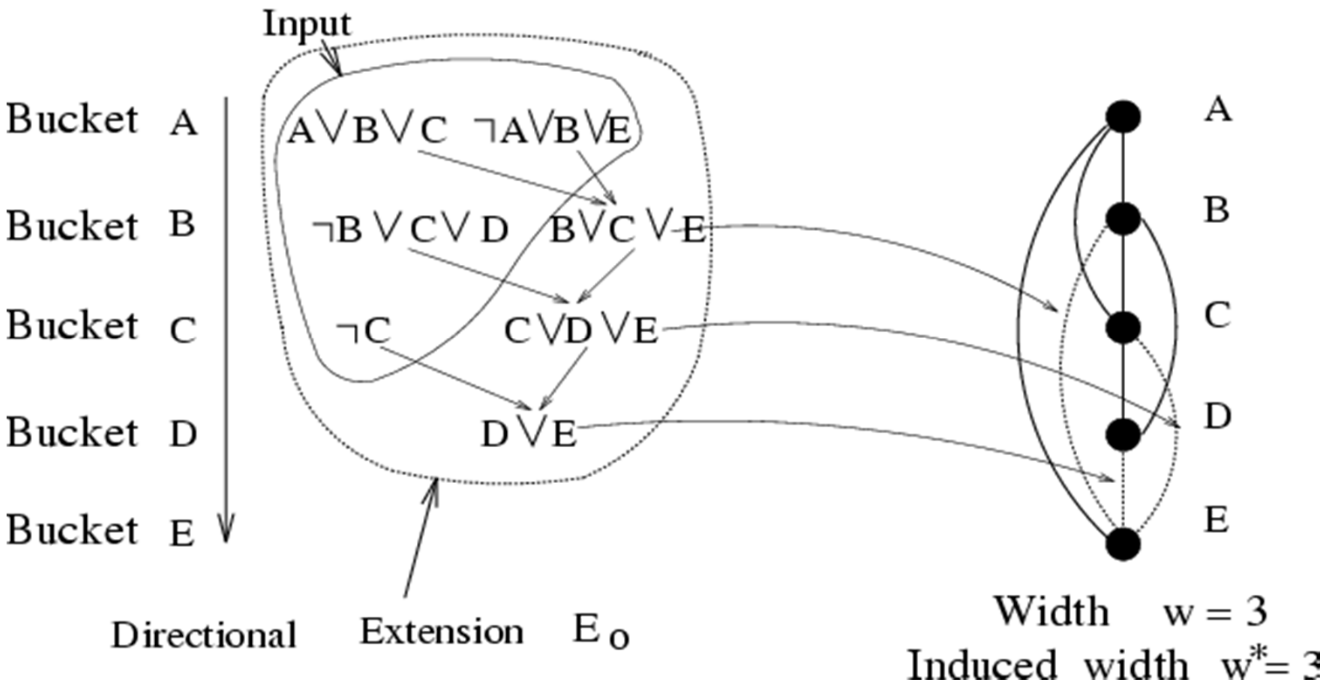


Complexity: $O(n\,exp(w^{*}(d)))$,

$w^{*}(\mathrm{d})$ - ***induced width along ordering d***

we get a greedily solved problem (backtrack- free)

Directional Resolution ⇔ Bucket-elimination

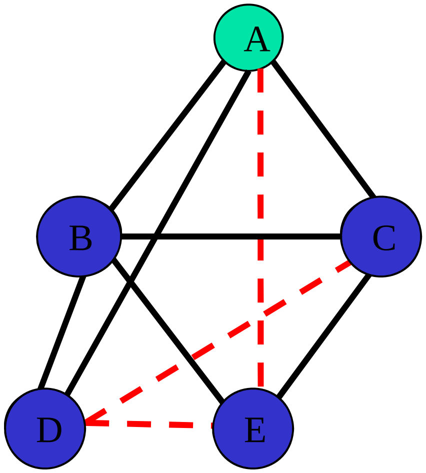


$$|\, bucket_i \,| = O(\exp(w^*))$$

$$\text{DR time and space} : O(n\exp(w^*))$$

(Original Davis-Putnam algorithm 1960)

Belief Updating/Probability of evidence Partition function



“Moral” graph

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

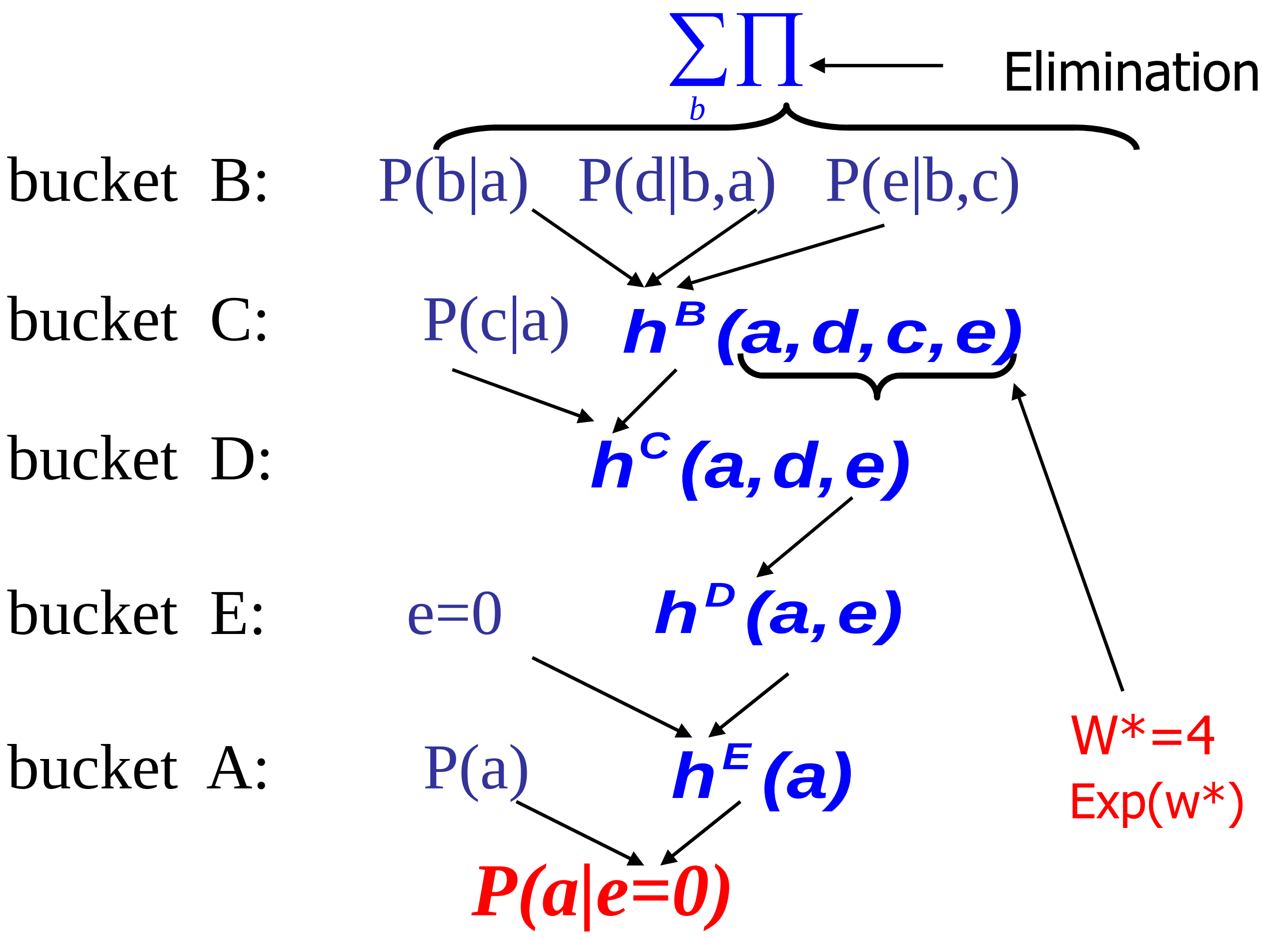
$$\sum_{e=0,d,c,b} P(a)\underbrace{P(b|a)}P(c|a)\underbrace{P(d|b,a)P(e|b,c)}=$$

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

Variable Elimination

Bucket Elimination

Algorithm *elim-bel* (Dechter 1996)



$\sum_{b} \prod$ Elimination operator

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

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

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

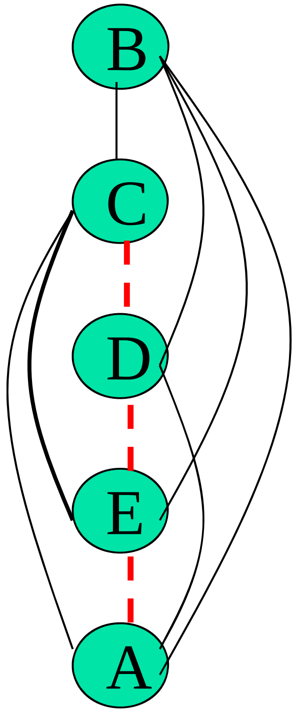
bucket E: $e=0$ $h^{D}(a,e)$

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

$P(a|e=0)$

$W^{*}=4$

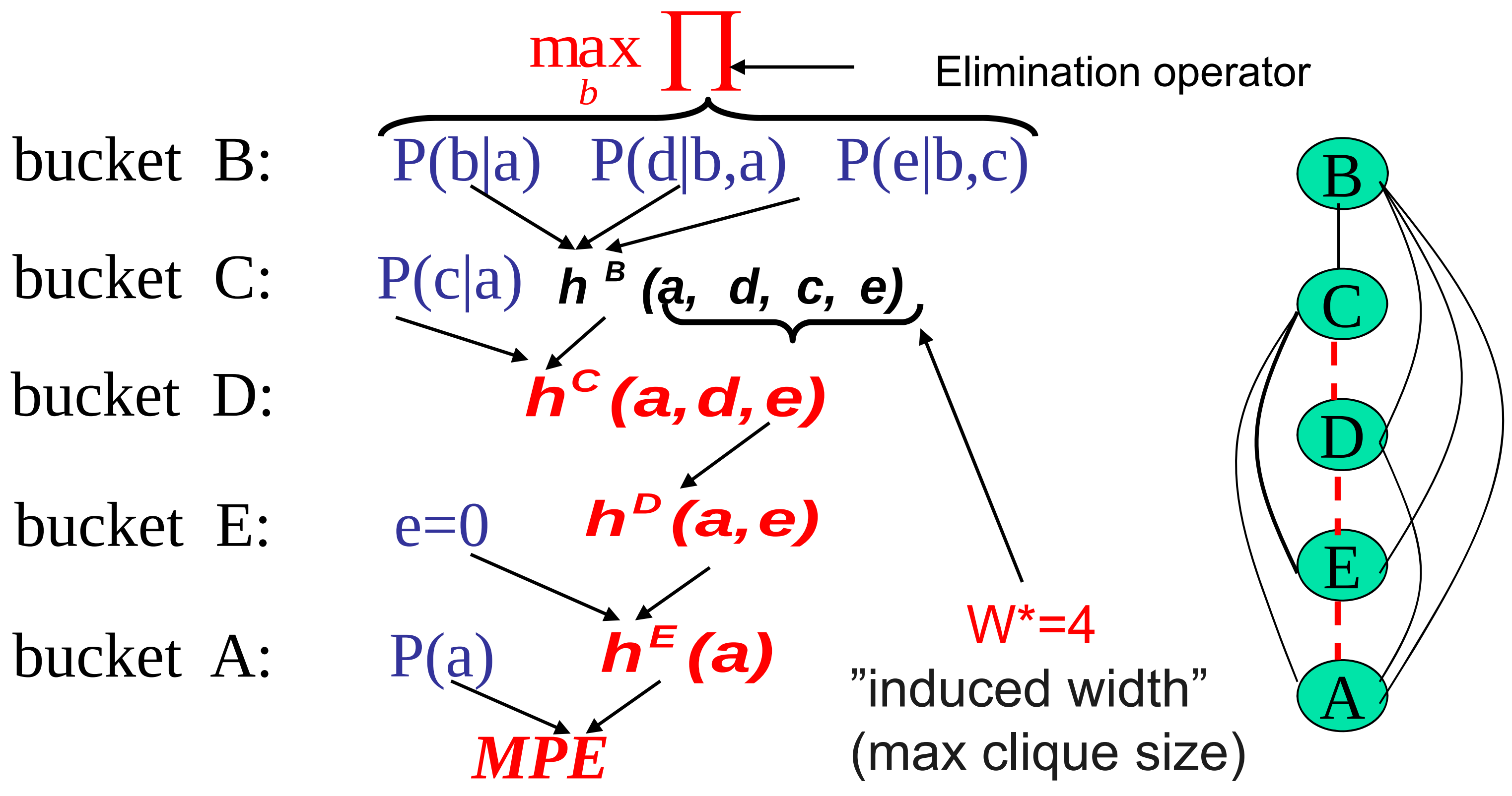
Exp(w*)



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

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

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



$\max_{b} \prod$ ← Elimination operator

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

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

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

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

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

MPE

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

Generating the MPE-tuple

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

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

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

2. $e' = 0$

1. $a' = \arg\max_{a} P(a) \cdot h^{E}(a)$

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

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

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

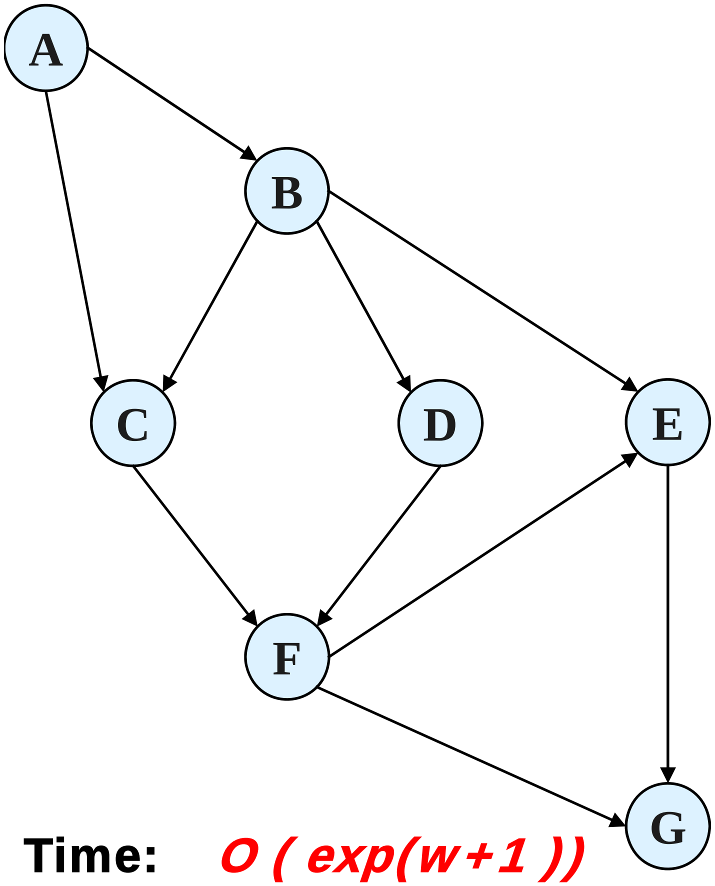
E: $e=0$ $h^{D}(a,e)$

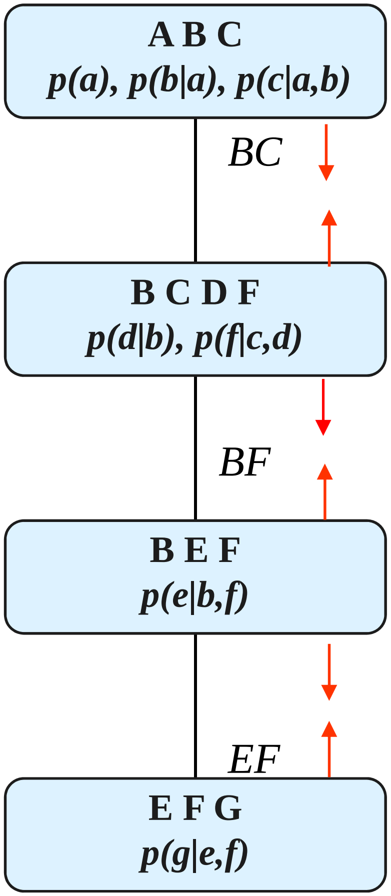
A: $P(a)$ $h^{E}(a)$

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

Cluster Tree Propagation

Join-tree clustering (Spigelhalter et. Al. 1988, Dechter, Pearl 1987)





A B C
p(a), p(b|a), p(c|a,b)

BC

B C D F
p(d|b), p(f|c,d)

BF

B E F
p(e|b,f)

EF

E F G
p(g|e,f)

$$h_{(1,2)}(b,c) = \sum_{a} p(a)\cdot p(b\,|\,a)\cdot p(c\,|\,a,b)$$

$$h_{(2,1)}(b,c) = \sum_{d,f} p(d\,|\,b)\cdot p(f\,|\,c,d)\cdot h_{(3,2)}(b,f)$$

$$h_{(2,3)}(b,f) = \sum_{c,d} p(d\,|\,b)\cdot p(f\,|\,c,d)\cdot h_{(1,2)}(b,c)$$

$$h_{(3,2)}(b,f) = \sum_{e} p(e\,|\,b,f)\cdot h_{(4,3)}(e,f)$$

$$h_{(3,4)}(e,f) = \sum_{b} p(e\,|\,b,f)\cdot h_{(2,3)}(b,f)$$

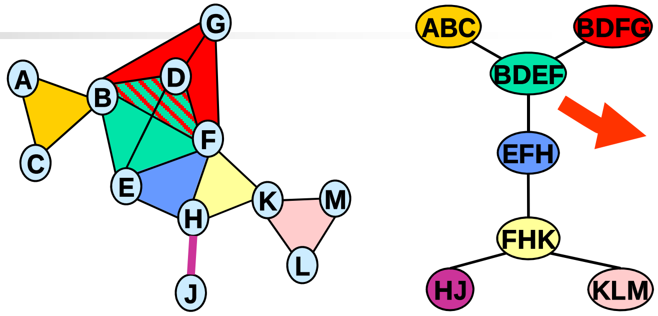
$$h_{(4,3)}(e,f) = p(G = g_e\,|\,e,f)$$

Time: ***O (exp(w+1))***
Space: ***O (exp(sep))***

For each cluster P(X|e) is computed

Complexity of Elimination

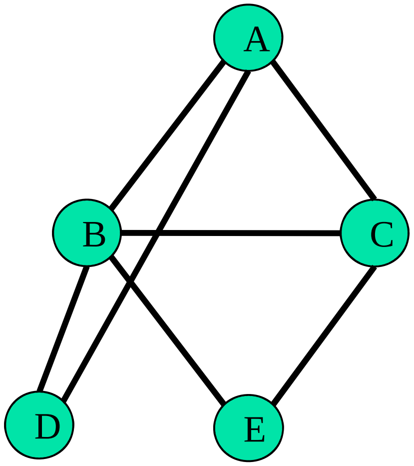
Trees are easy



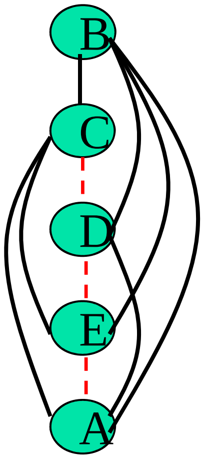
$$O(n \exp(w^*(d)))$$

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

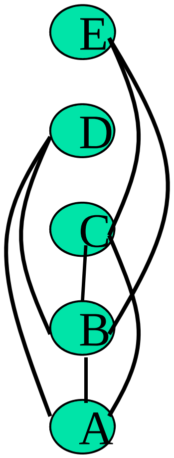
The effect of the ordering:



"Moral" graph



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



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

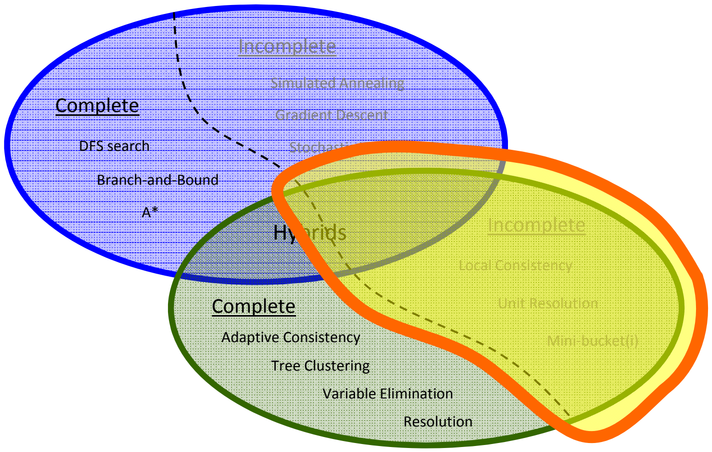
Spigelhalter et. Al. 1983, Junction tree algorithm (join-tree algorithm)

Outline

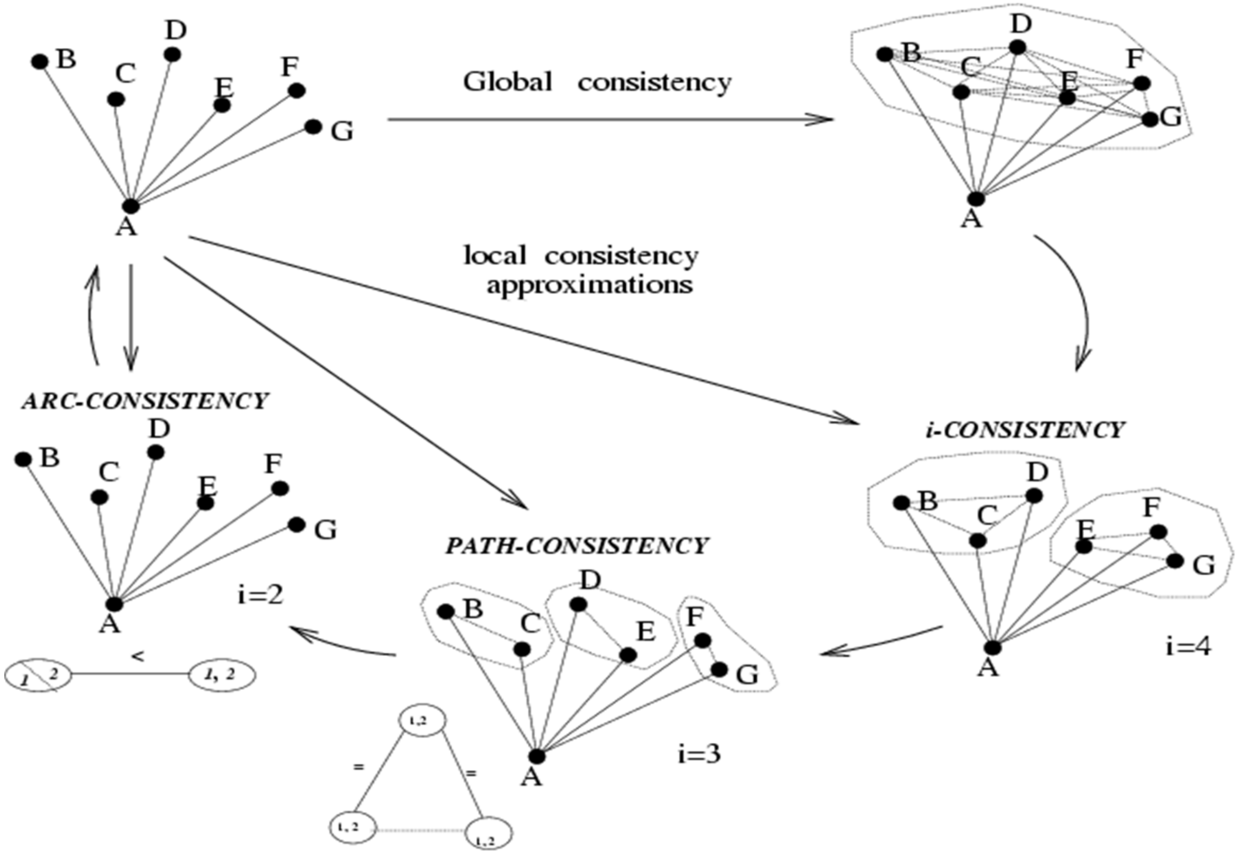
- What are graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking

Approximate Inference

- Mini-buckets, mini-clusters, i-consistency
- Belief propagation, constraint propagation, Generalized belief propagation

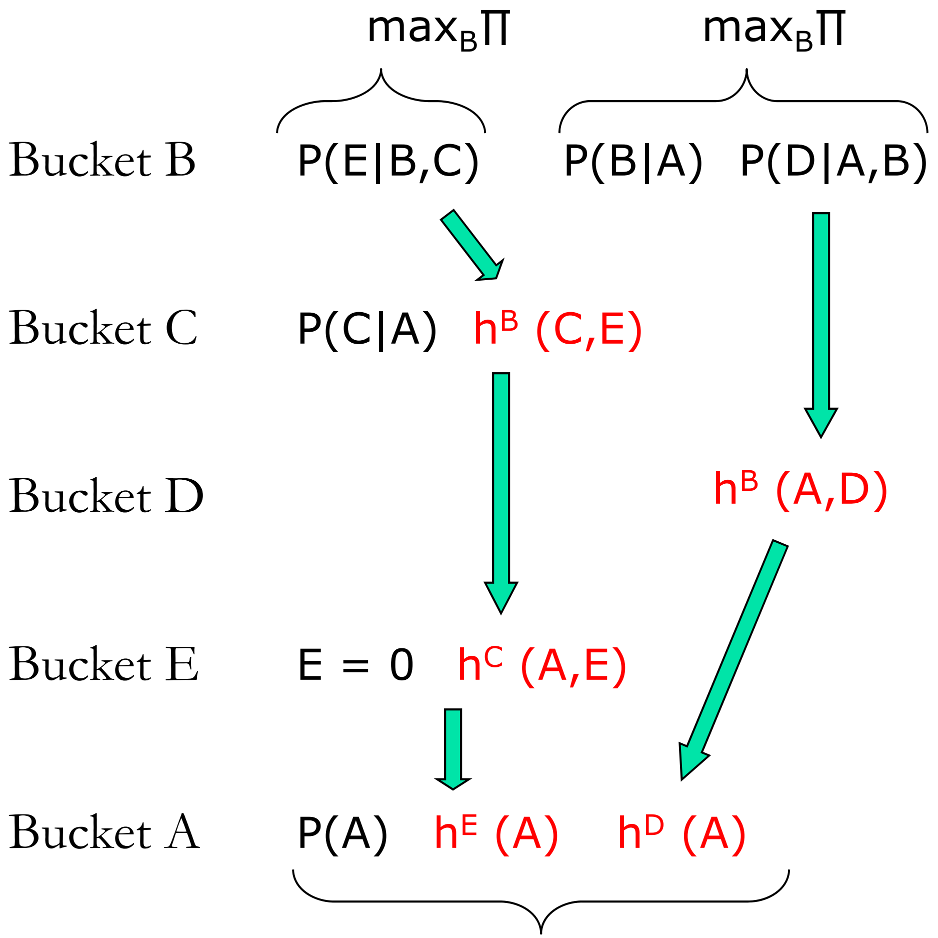


From Global to Local Consistency



Leads to one pass directional bounded inference, or Iterative propagation algorithms

Mini-Bucket Elimination



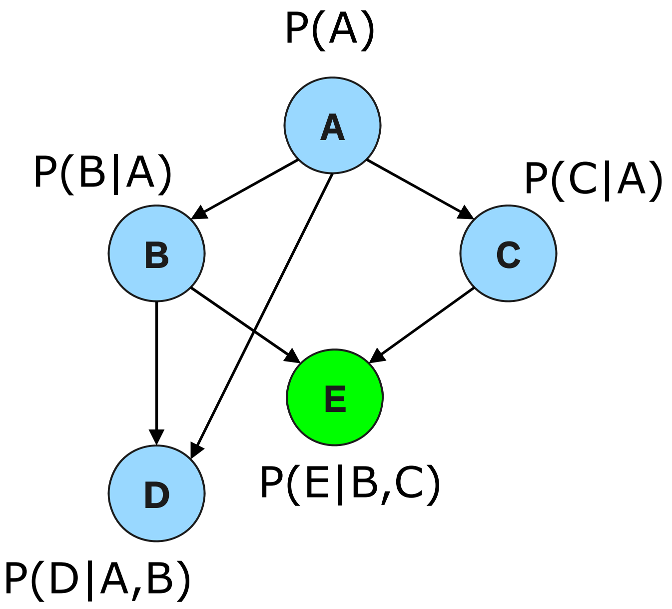
Bucket B $\overbrace{P(E|B,C)}^{\max_B \prod}$ $\overbrace{P(B|A)\ \ P(D|A,B)}^{\max_B \prod}$

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

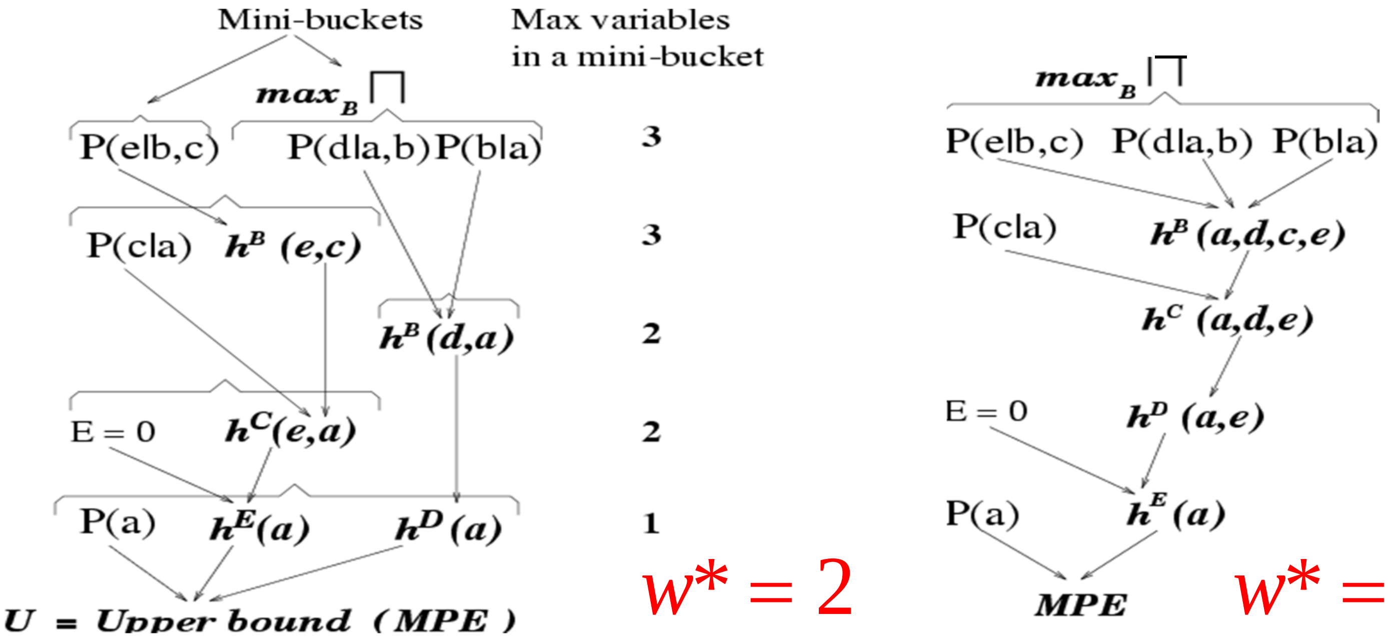


MPE* is an upper bound on MPE --U
Generating a solution yields a lower bound--L

MBE(i) (Dechter and Rish 1997)

- **Input: i – max number of variables allowed in a mini-bucket**
- **Output: [lower bound (P of a sub-optimal solution), upper bound]**

Example: approx-mpe(3) versus elim-mpe



$w^* = 2$

$w^* = 4$

Properties of MBE(i)/mc(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.

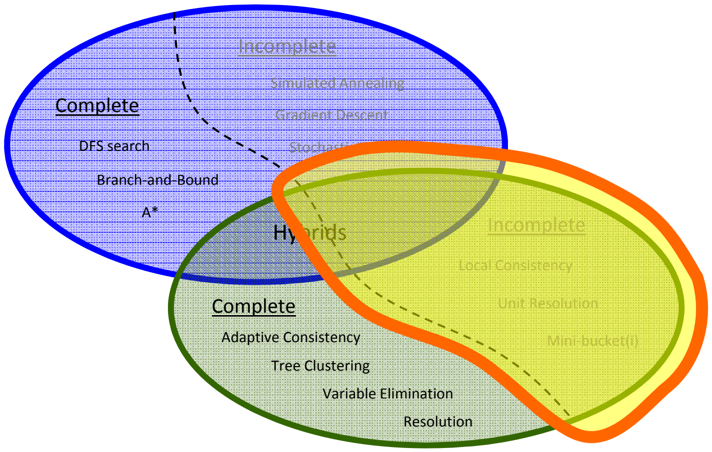
- As ***i*** increases, both accuracy and complexity increase.

- 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)

Approximate Inference

- Mini-buckets, mini-clusters
- Belief propagation, constraint propagation, Generalized belief propagation



Iterative, directional algorithms, vs Propagation algorithms

Arcs-consistency

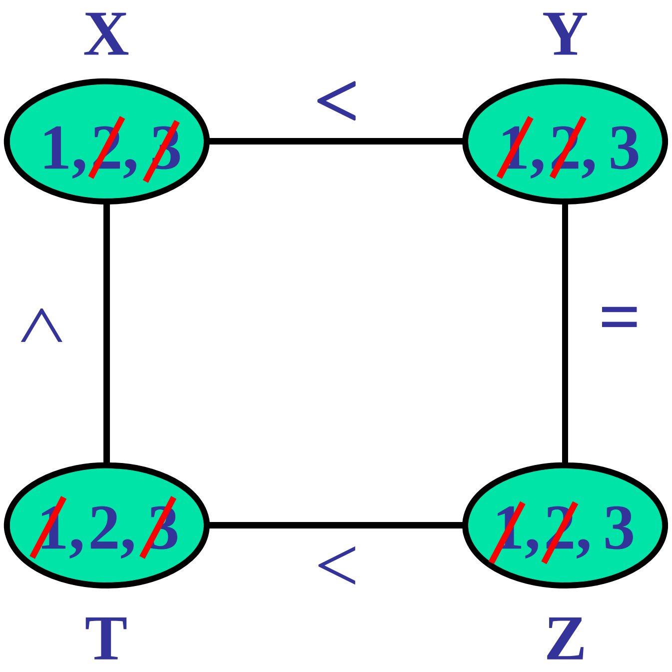
$1 \leq X, Y, Z, T \leq 3$

$X < Y$

$Y = Z$

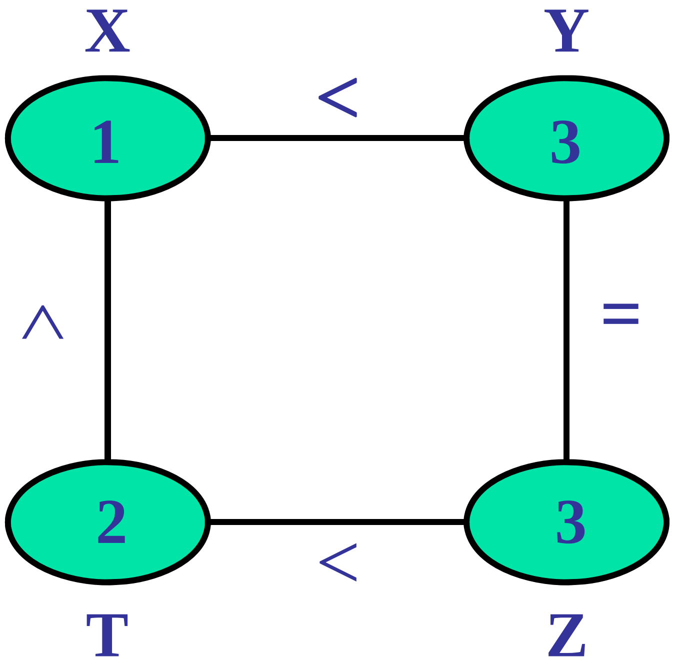
$T < Z$

$X \leq T$



Arc-consistency

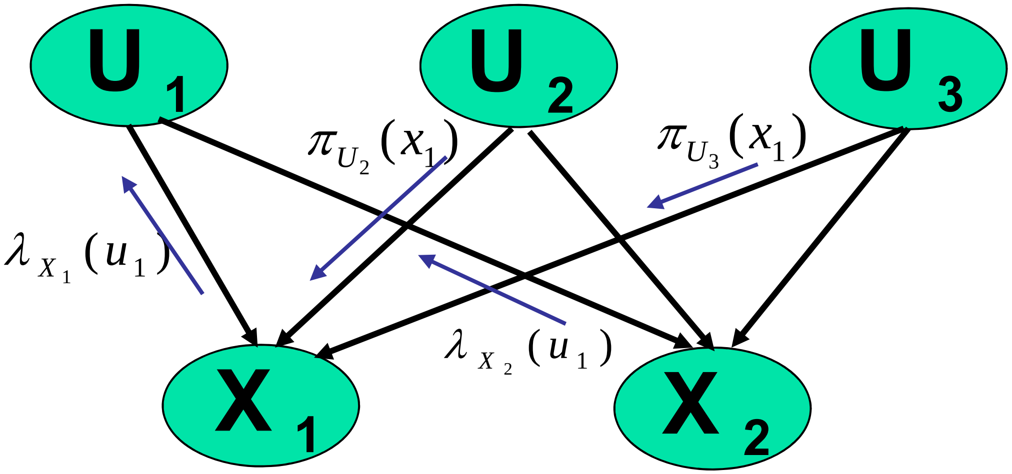
$1 \le X, Y, Z, T \le 3$
$X < Y$
$Y = Z$
$T < Z$
$X \le T$



$$R_X \leftarrow \prod_X R_{XY} \bowtie D_Y$$

Iterative (Loopy) Belief Proapagation

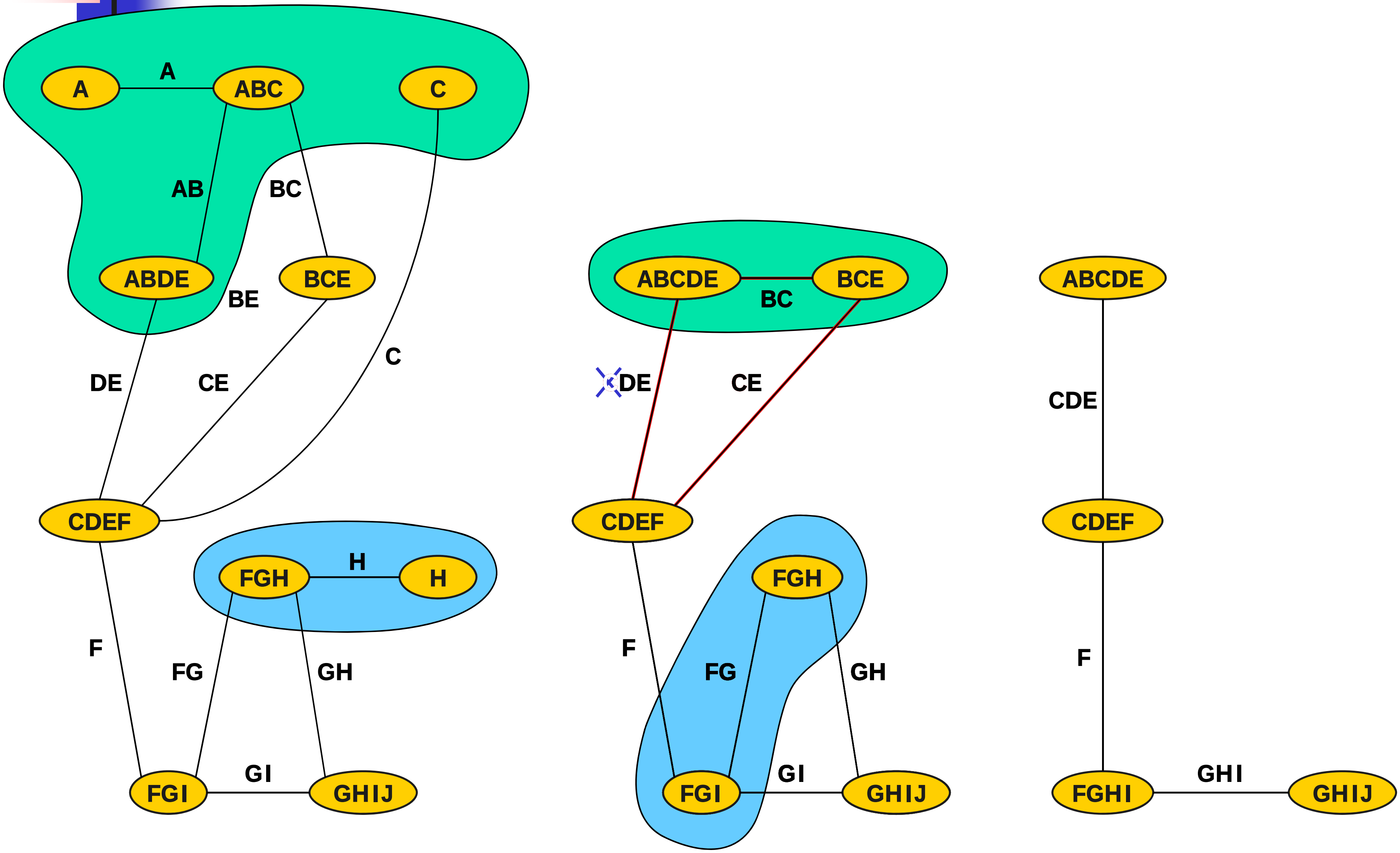
- Belief propagation is exact for poly-trees
- IBP - applying BP iteratively to cyclic networks



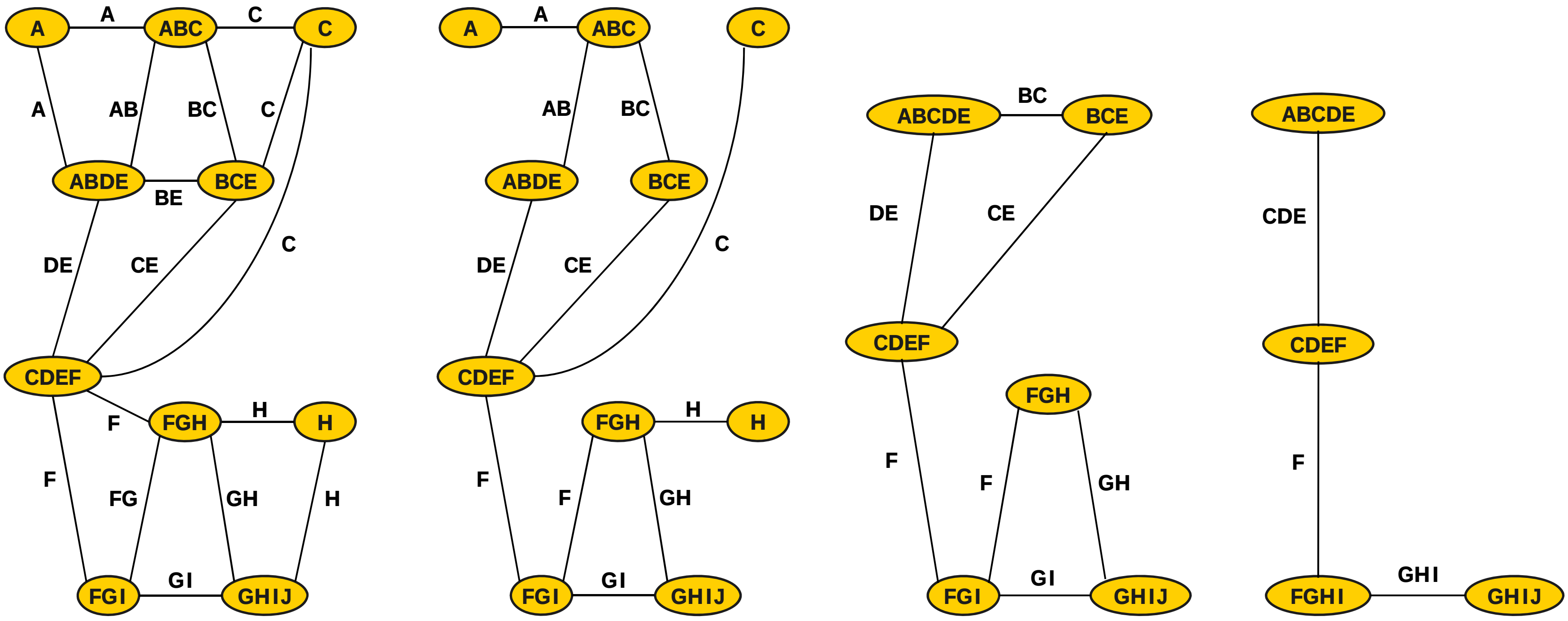
One step : update BEL(U_1)

- No guarantees for convergence
- Works well for many coding networks

Collapsing Clusters



Join-Graphs



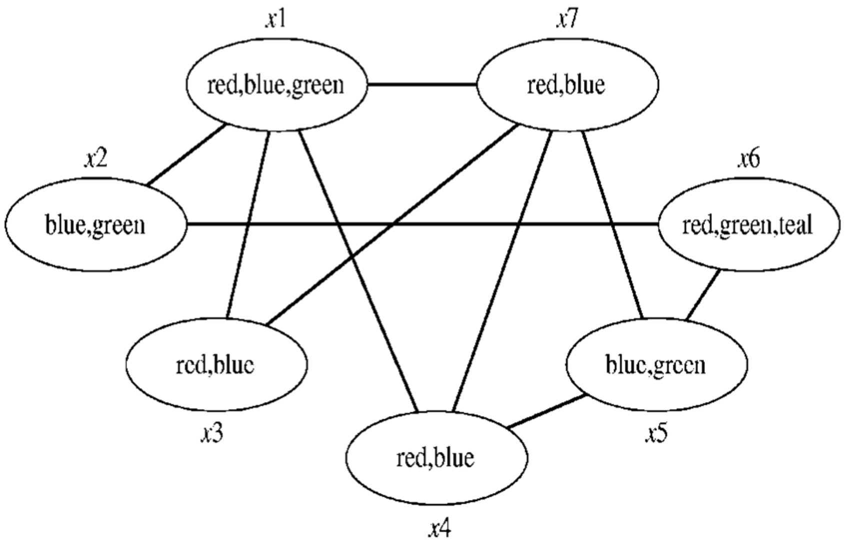
more accuracy

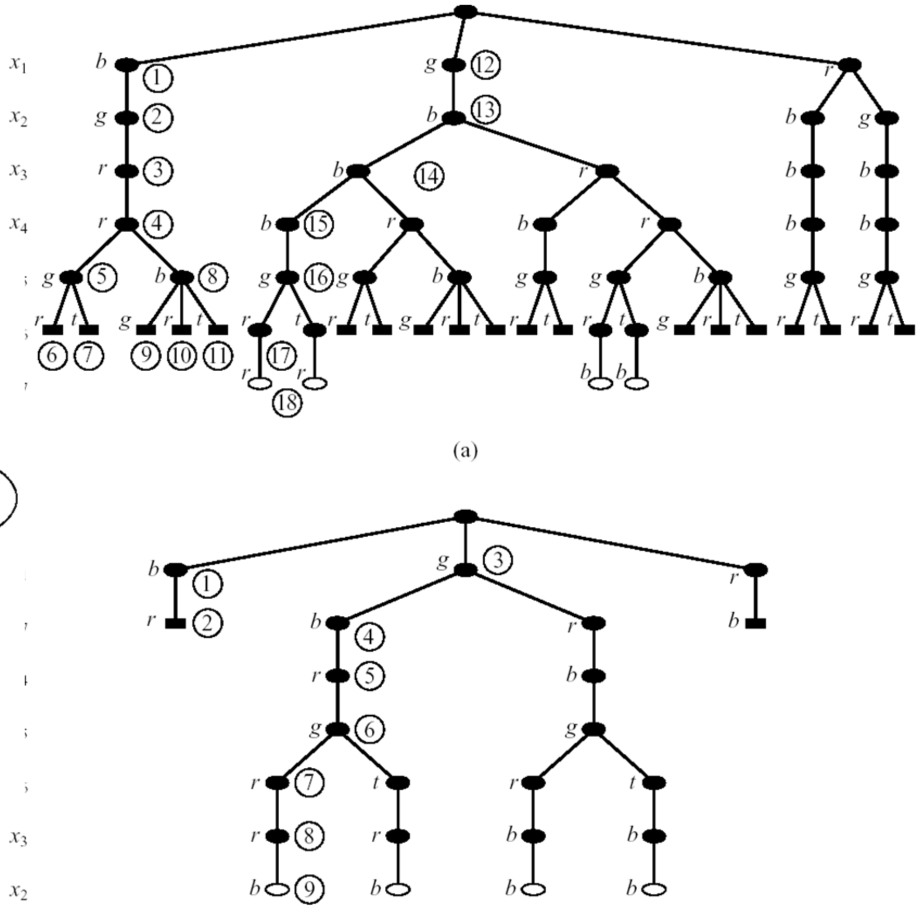
less complexity

Outline

- What are graphical models
- Inference: Exact and approximate
- **Conditioning Search: exact** and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking

Backtracking Search for a Solution



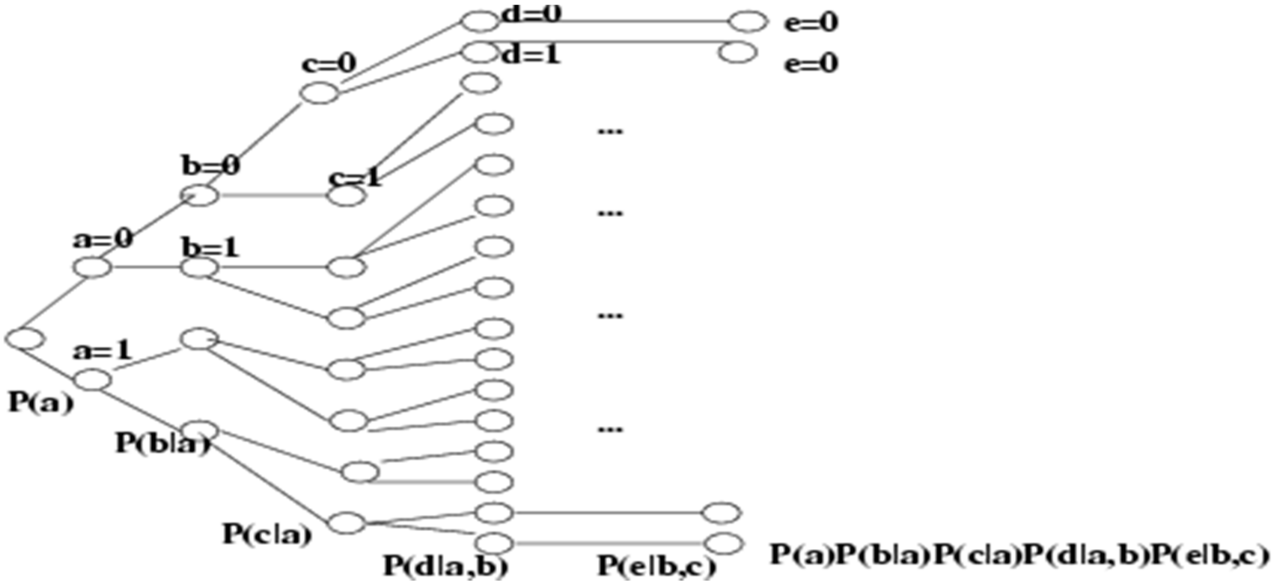


(a)

(b)

Belief Updating: Searching the Probability Tree

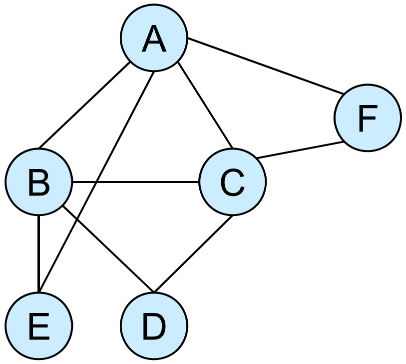
$$P(a,e=0)=P(a)\sum_{b}P(b\,|\,a)\sum_{c}P(c\,|\,a)\sum_{b}P(d\,|\,a,b)\sum_{e=0}P(e\,|\,b,c)$$



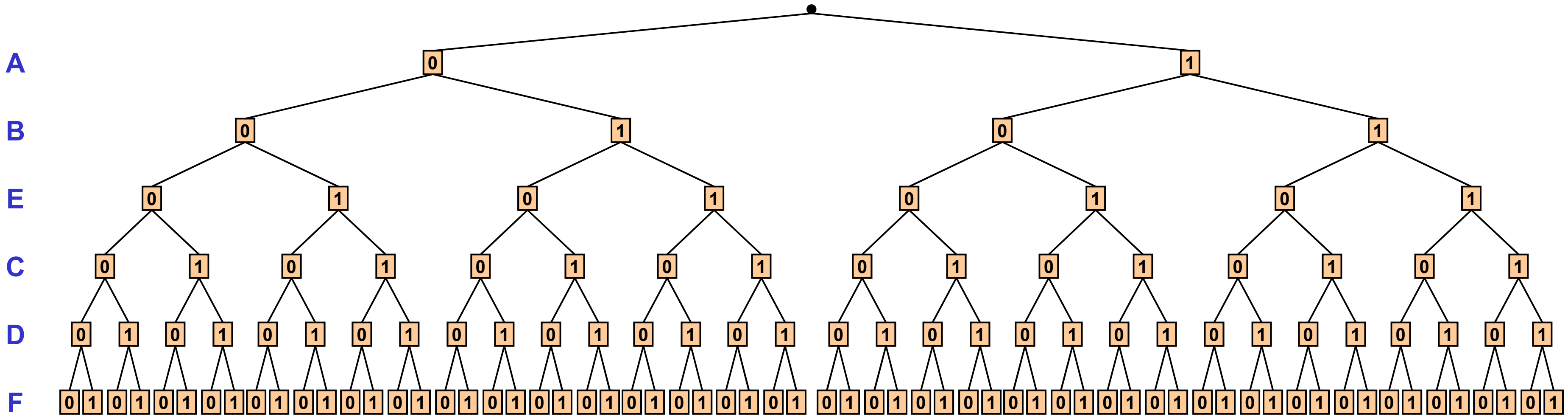
Brute-force Complexity: O(exp(n)), linear space
Same as counting solutions

OR search space

Ordering: A B E C D F

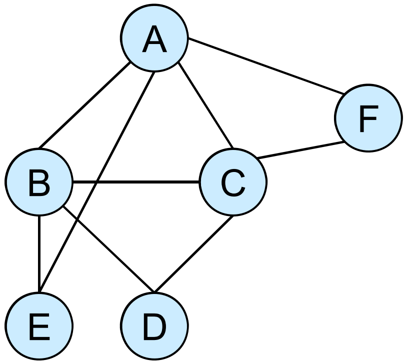


Constraint network

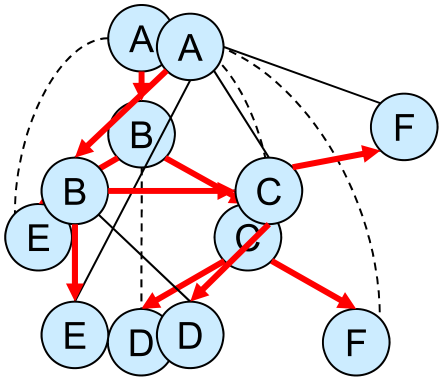


Size of search space: O(*exp* n)

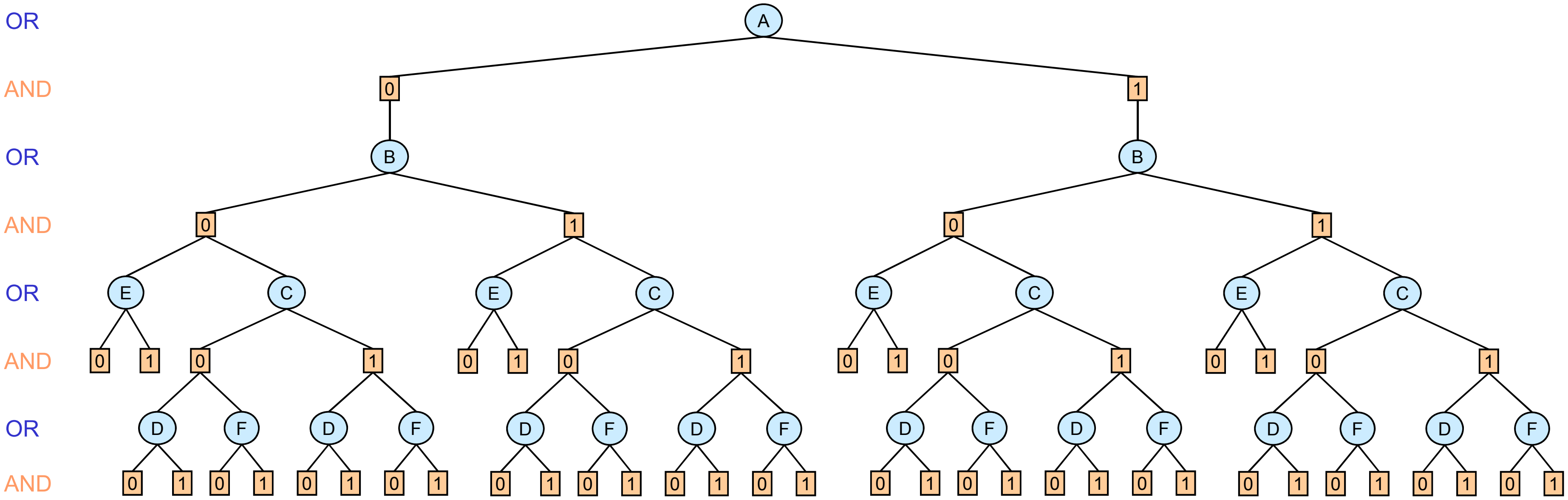
AND / OR Search Space



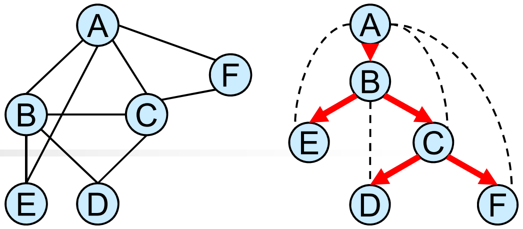
Primal graph

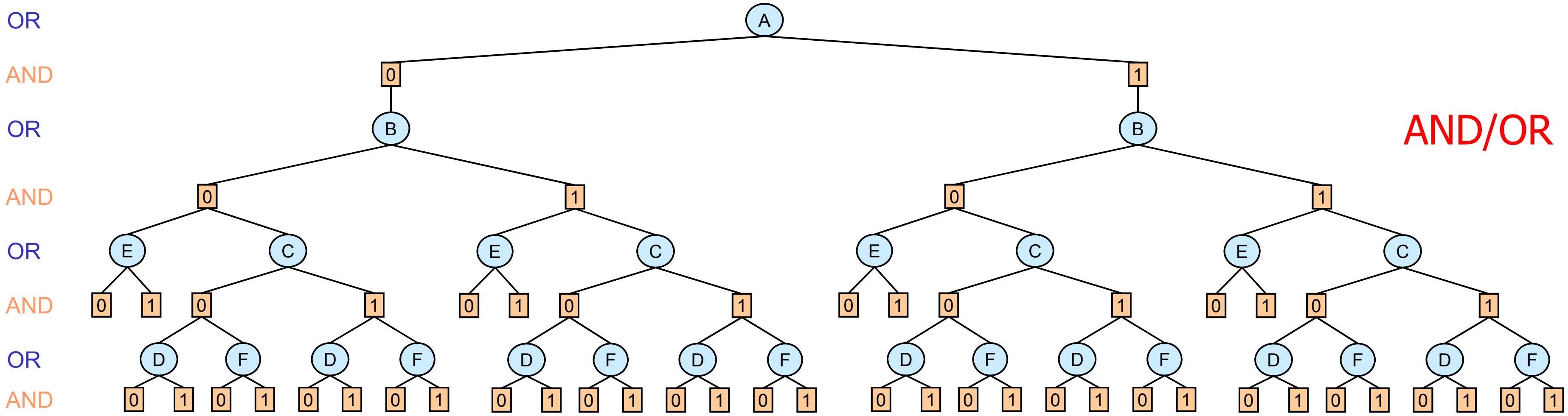


DFS tree

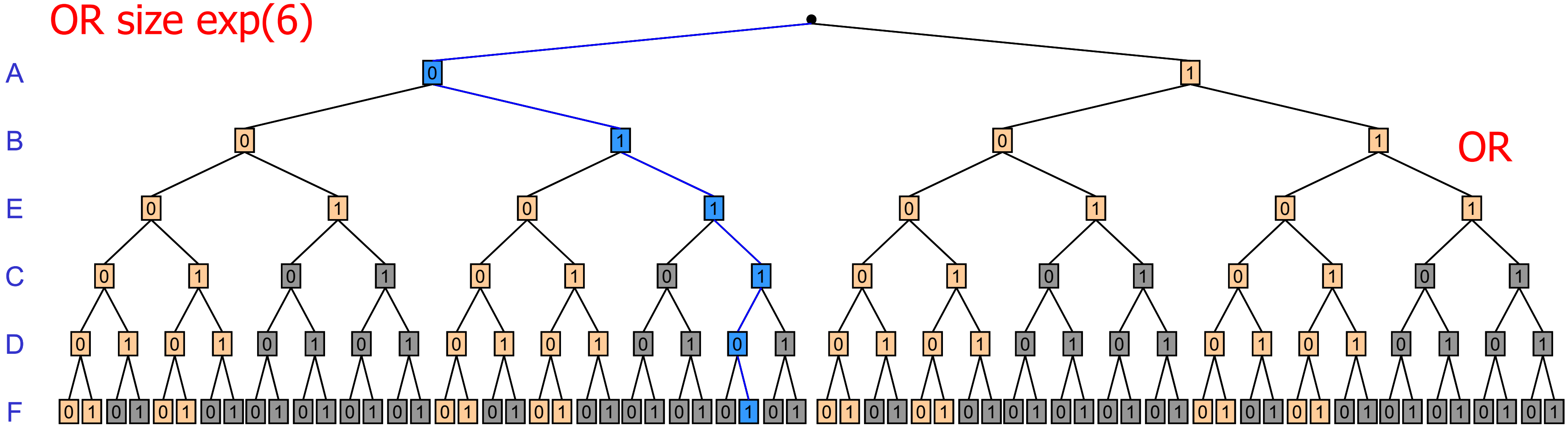


AND / OR vs. OR



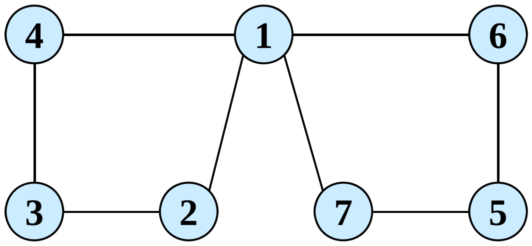


AND/OR size: exp(4),
OR size exp(6)



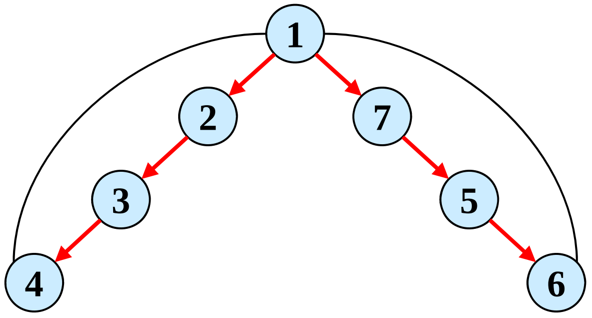
Pseudo-Trees

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

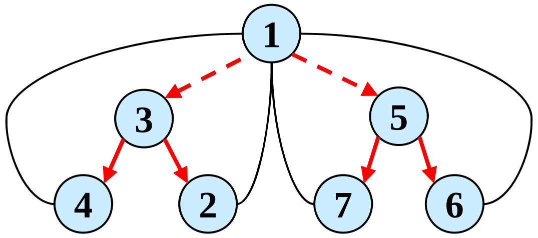


(a) Graph

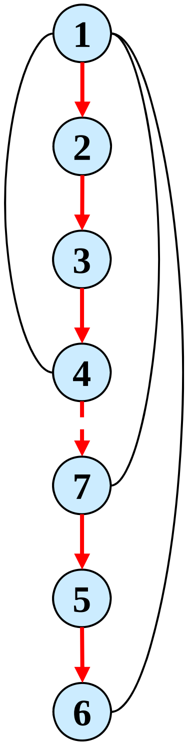
$h <= w^* \log n$



(b) DFS tree
depth=3



(c) pseudo- tree
depth=2



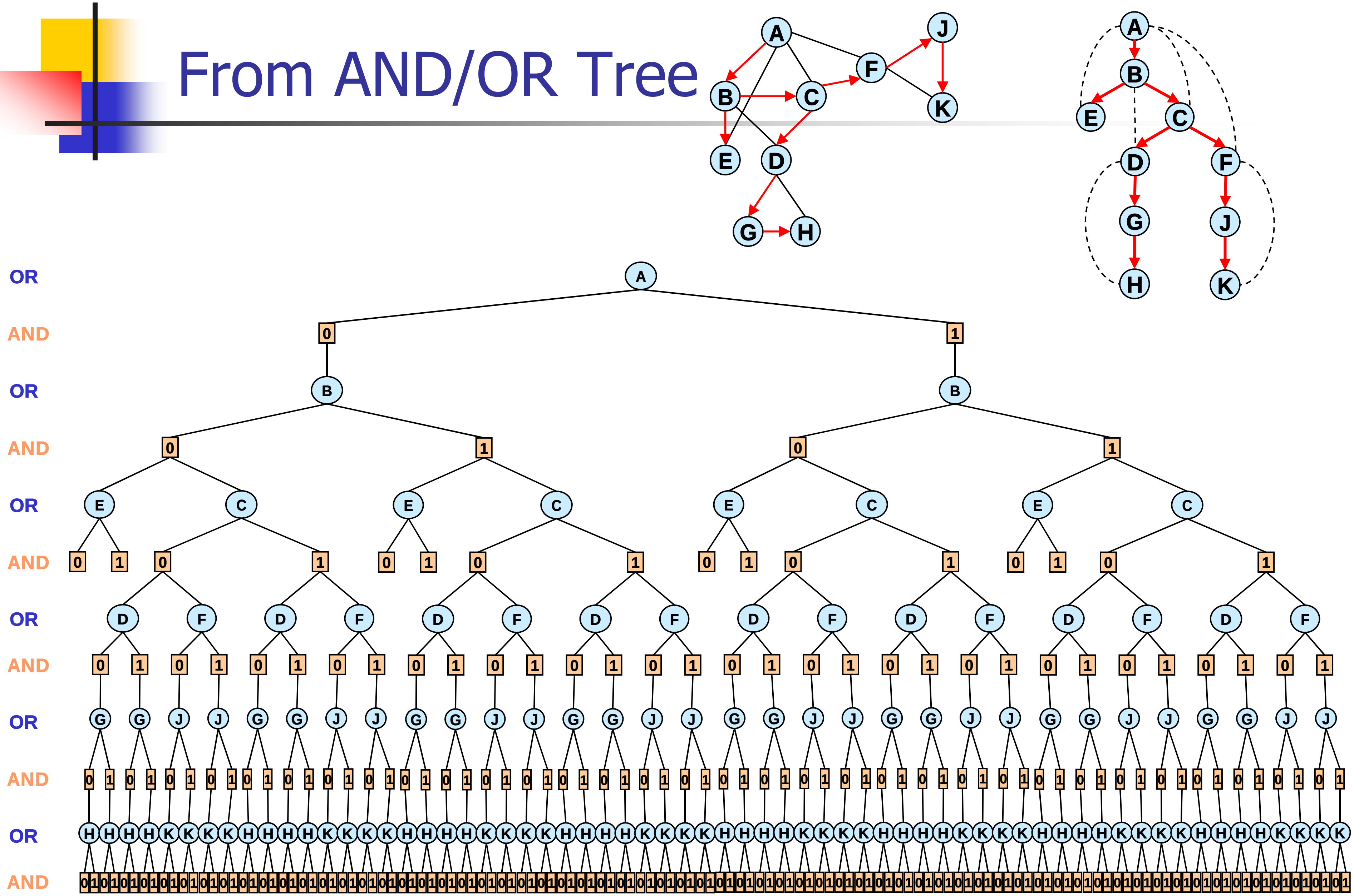
(d) Chain
depth=6

Complexity of AND/OR Tree Search

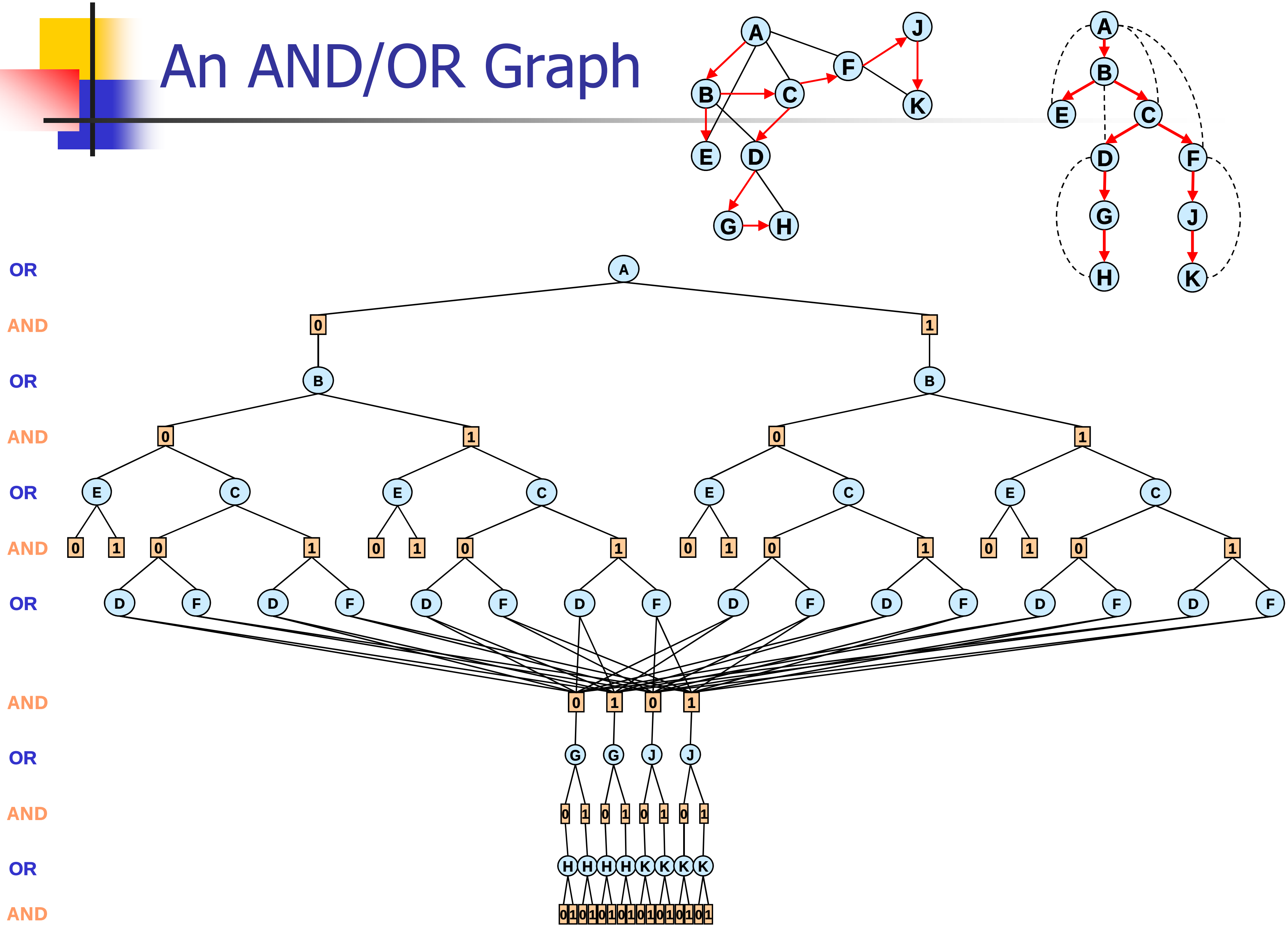
	AND/OR tree	**OR tree**
Space	$O(n)$	$O(n)$
Time	$O(n\ d^{h})$ $O(n\ d^{w^*\log n})$ (Freuder & Quinn85), (Collin, Dechter & Katz91), (Bayardo & Miranker95), (Darwiche01)	$O(d^{n})$

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

From AND/OR Tree



An AND/OR Graph



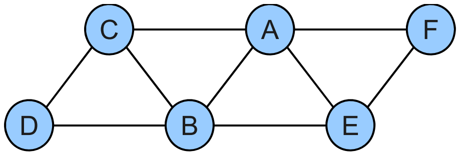
Complexity of AND/OR Graph Search

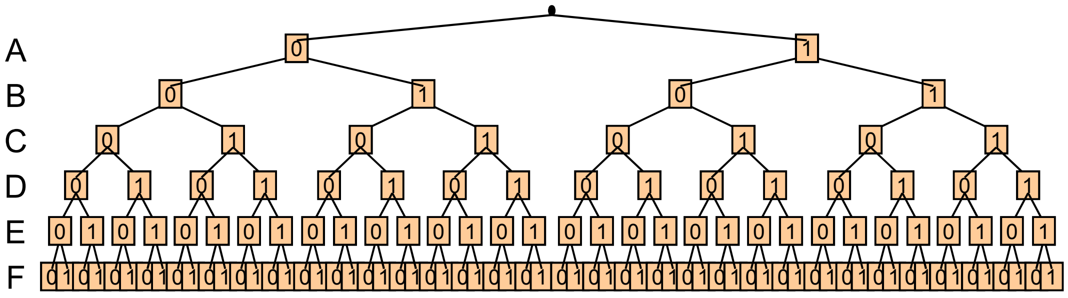
	AND/OR graph	OR graph
Space	$O(n\ d^{w^*})$	$O(n\ d^{pw^*})$
Time	$O(n\ d^{w^*})$	$O(n\ d^{pw^*})$

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

$w^* \leq pw^* \leq w^* \log n$

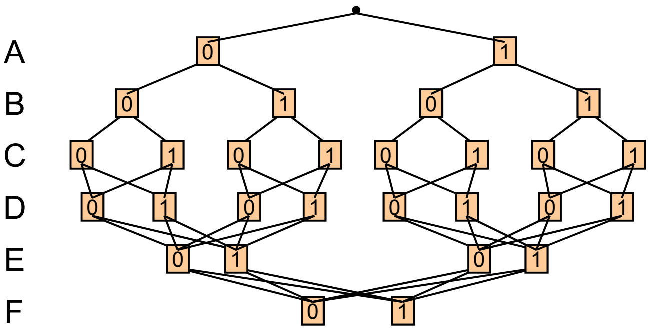
All Four Search Spaces





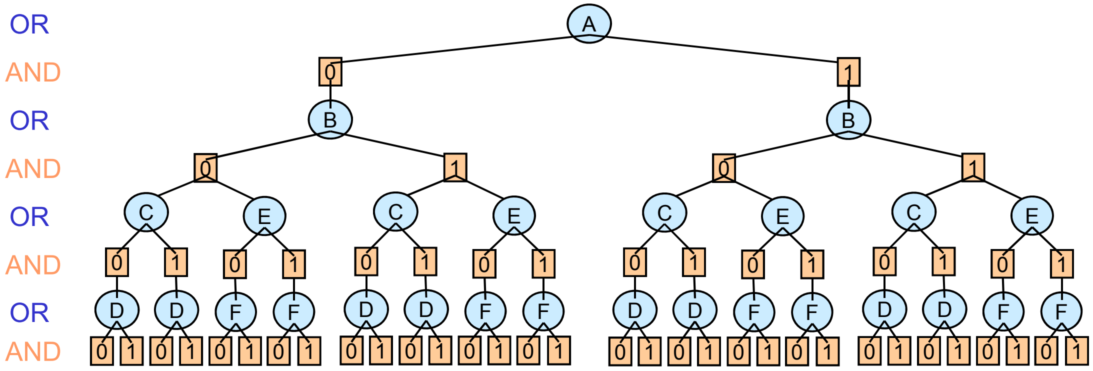
Full OR search tree

126 nodes



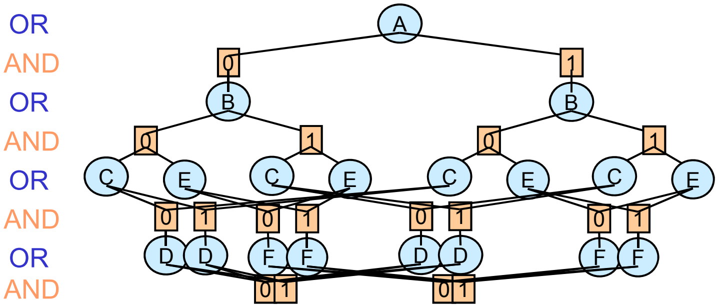
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes

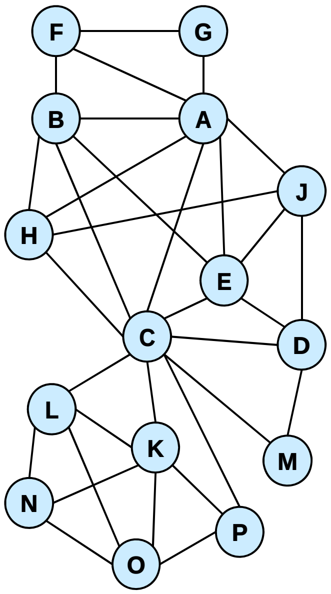


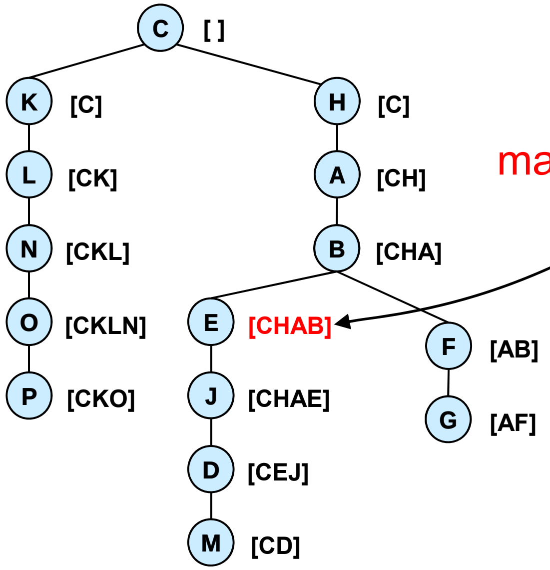
Context minimal AND/OR search graph

18 AND nodes

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.

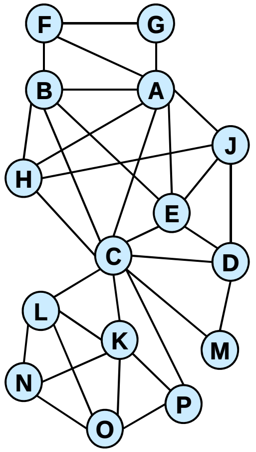


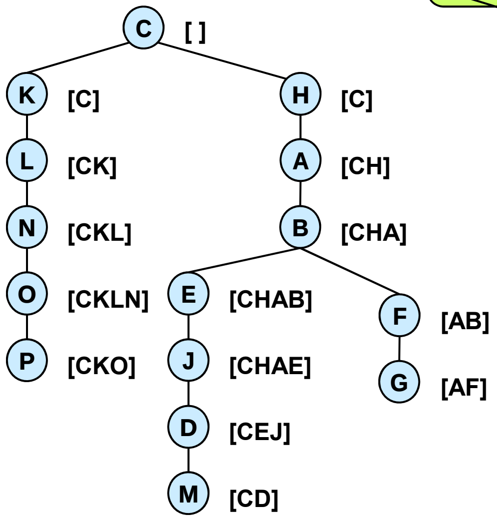


max context size = treewidth

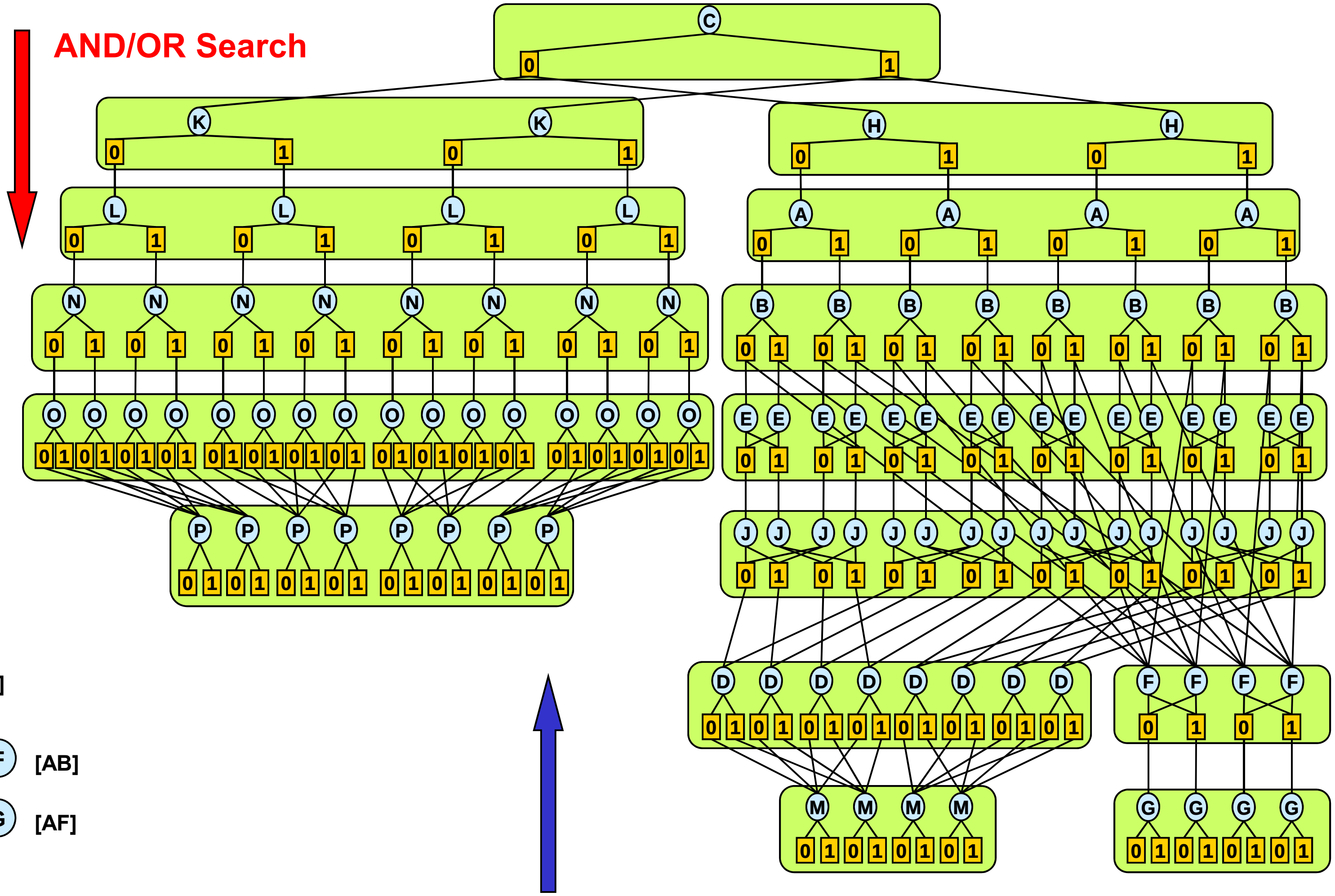
(C K H A B E J L N O D P M F G)

AND/OR Context Minimal Graph





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

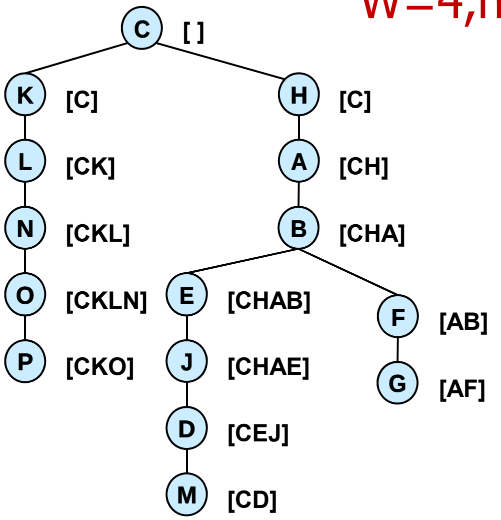


AND/OR Search

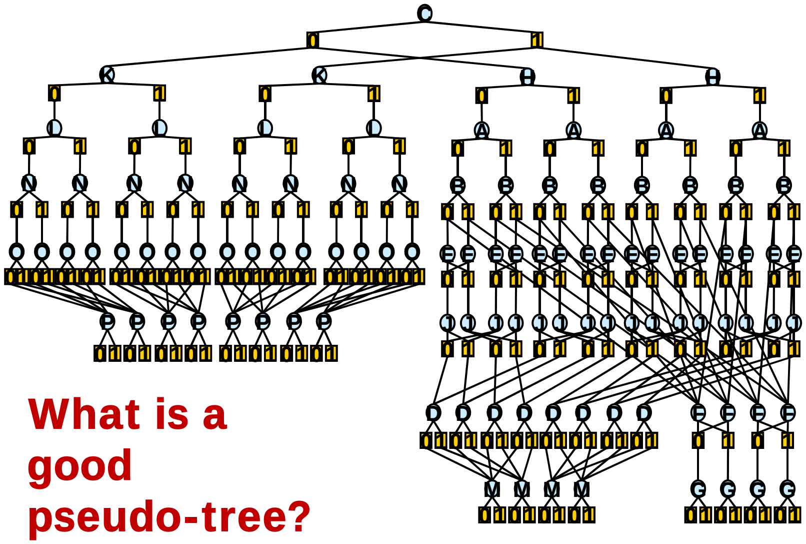
Variable Elimination

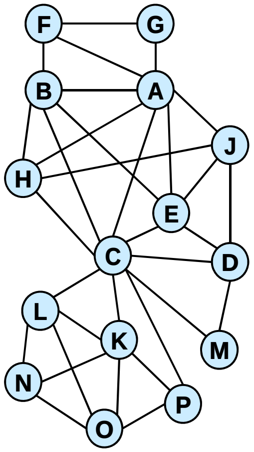
The impact of the pseudo-tree

W=4,h=8



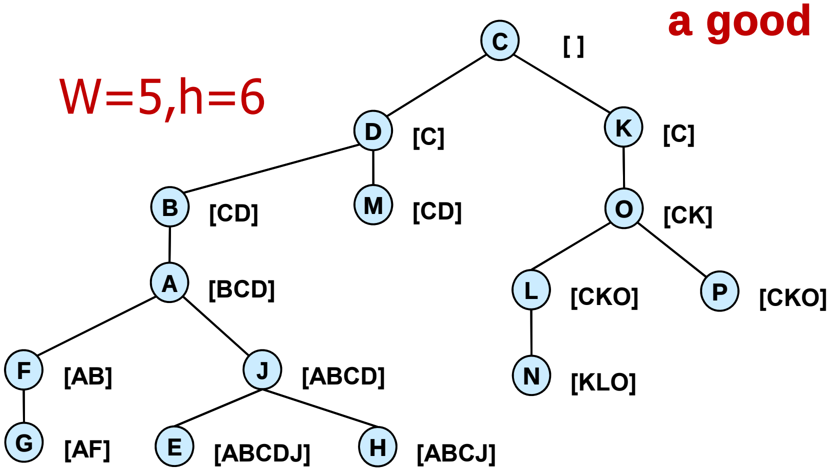
(C K H A B E J L N O D P M F G)



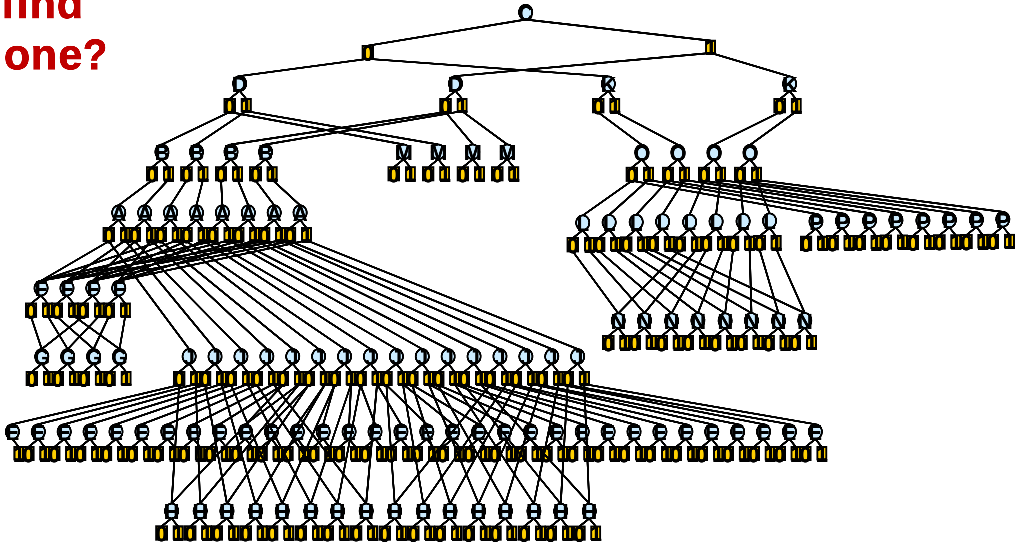


What is a good pseudo-tree? How to find a good one?

W=5,h=6



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

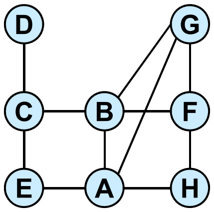


Outline

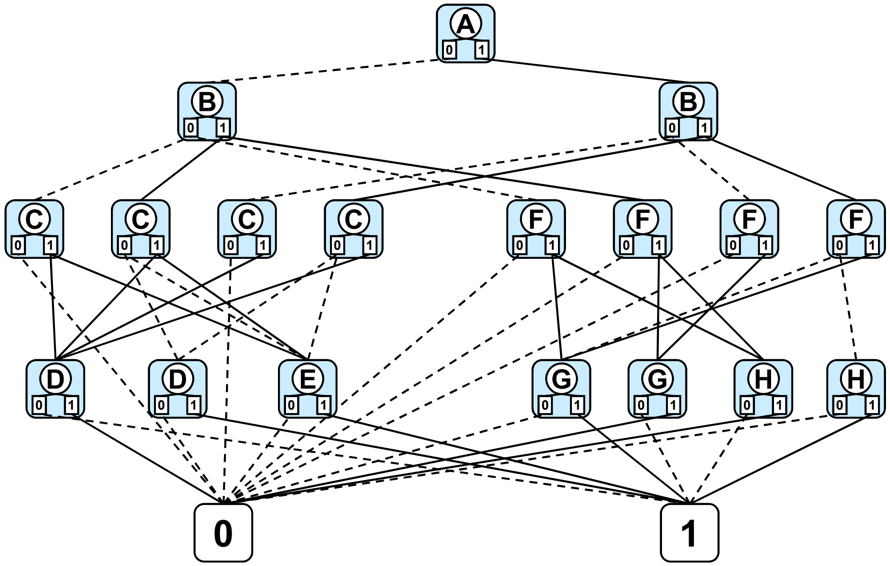
- What are graphical models
- Inference: Exact and approximate
- Conditioning Search: exact and approximate
- Hybrids of search and inference (exact)
- Compilation, (e.g., AND/OR Decision Diagrams)
- Questions:
  - Representation issues: directed vs undirected
  - The role of hidden variables
  - Finding good structure
  - Representation guided by human representation
  - Computation: inspired by human thinking

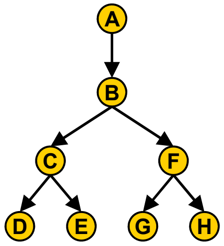
AOBDD vs. OBDD (Mateescu and Dechter 2006)

The context-minimal graph
Can be "minimized into an AOMDD
By merging and redundancy removal



primal graph

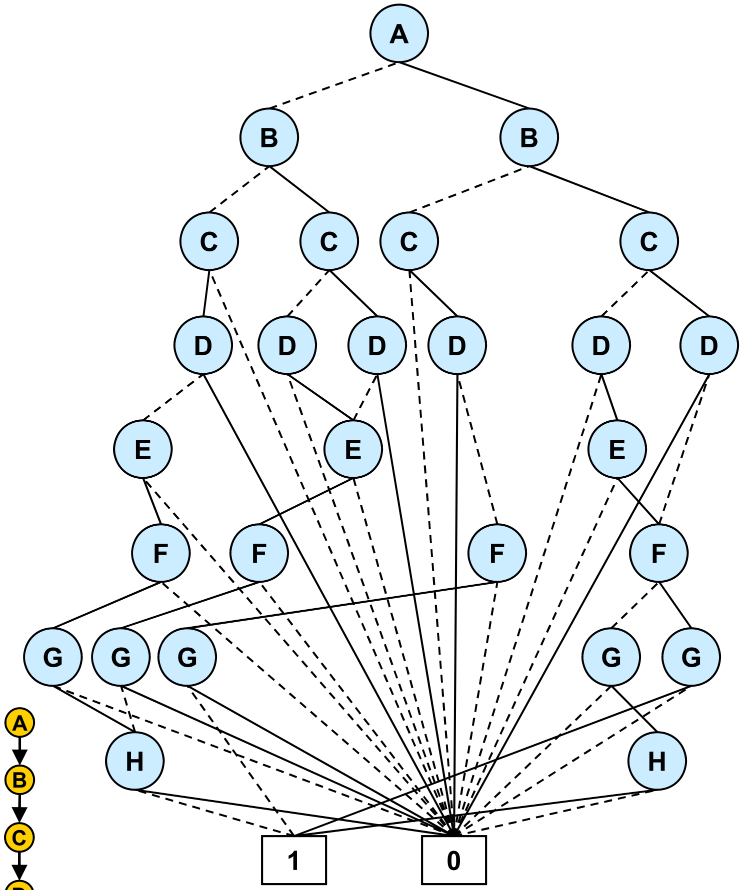


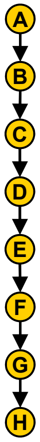


AOBDD

18 nonterminals

47 arcs





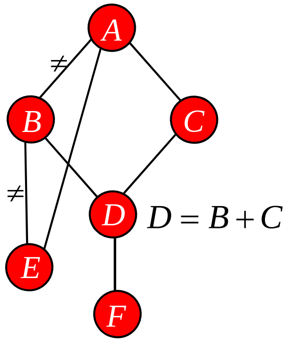
OBDD

27 nonterminals

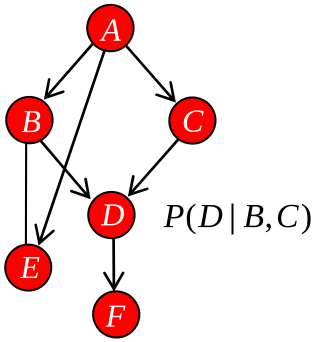
54 arcs

Constraint Network vs Bayesian Network

Constraint networks	**Probability networks**
Is it consistent?	Always consistent
Find solution	Find t s.t P(t)>0
- NP-complete	- Easy: backtrack-free
Count solutions	**Find P(X|e)?**
- #P-complete	- **#P-complete**
unminimal const	Explicit minimal tables
Solved by search	Solved by variable elimination
Hard to sample	Easy to sample



represents
$sol\ (A, B, C, D, E, F)$



represents
$P(A, B, C, D, E, F)$

The End

Thank You