From Constraint Programming to Graphical models; the role of AND/OR search

Rina Dechter

Bren school of ICS

University of California, Irvine

Collaborators:
Kalev Kask
Bozhena Bidyuk
Radu Marinescu,
Robert Mateescu
Vibhav Gogate

DOD SAT workshop, March 2008

From Constraint Programming to Graphical Models

Languages/modeling
- Eclipse, ILOG solver
- CPLex

Algorithms over:
- Constraint networks
- queries: constraint satisfaction,
- Satisfiability/counting

Principles:
- Decomposition
- Equivalence
- pruning

Graphical models:
- Probabilistic networks
- Cost networks
- Influence diagrams
- MDPs

Queries:
- Liklihood computation
- Constraint Optimization

SAT/CSP:
- Using the simplest model
- Focus on algorithms/data-structures

Code perfection/code sharing

Current focus: Mixed networks
- Use sat as subroutine
- Apply the same principle.

Principles for SAT/CSPs

Constraint Satisfaction/counting:

- Problem decomposition: backjumping
- Subproblem equivalence:
  - Learn nogoods (clause learning)
  - Learn goods
- Pruning: constraint propagation, unit propagation

Combinatorial optimization/ Likelihood queries:

- Decomposition: (AND/OR)
- Equivalence: caching optimal conditioned solutions
- Pruning: by mini-bucket, soft arc-consistency, belief propagation, lower-bound heuristic

Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.
- Exact Algorithms: AND/OR search spaces
- AND/OR search for combinatorial optimization
- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

Constraint Networks (Montanari 1974)

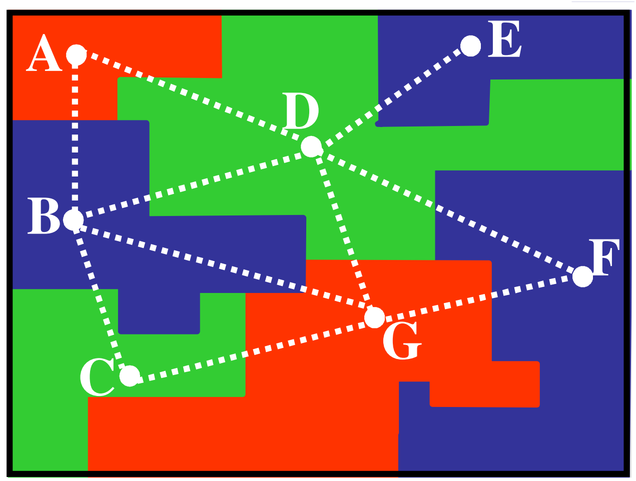
Example: map coloring

Variables - countries (A,B,C,etc.)

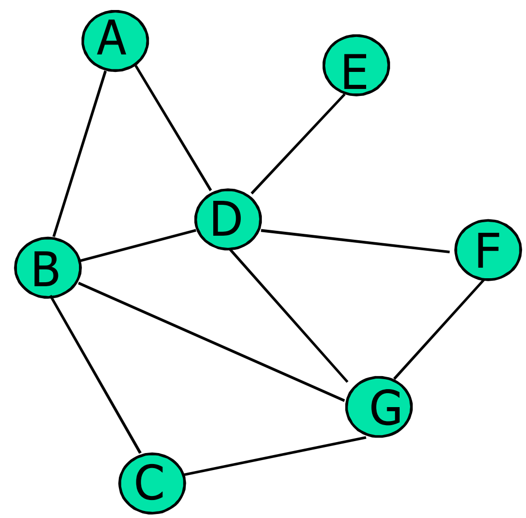
Values - colors (red, green, blue)

Constraints: $A \neq B$, $A \neq D$, $D \neq E$, *etc.*

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



Constraint graph

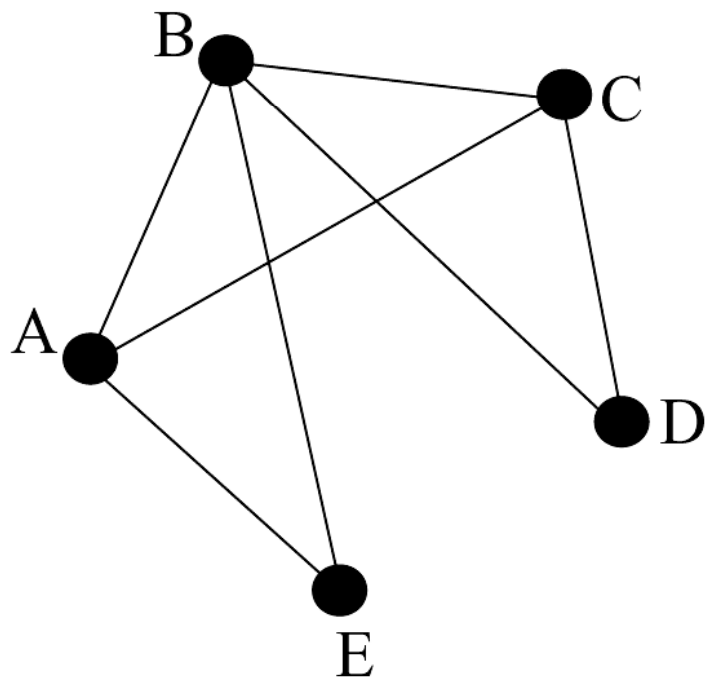


Semantics: set of all solutions

Primary task: find a solution

Propositional Satisfiability

$\varphi = \{(\neg C), (A \vee B \vee C), (\neg A \vee B \vee E), (\neg B \vee C \vee D)\}.$

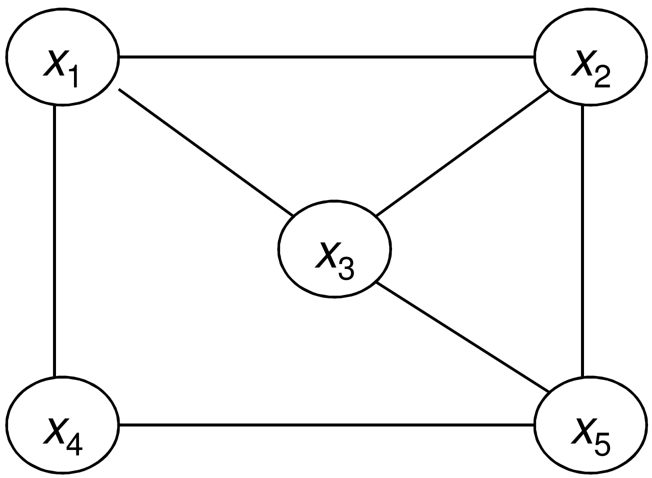


Constraint Optimization

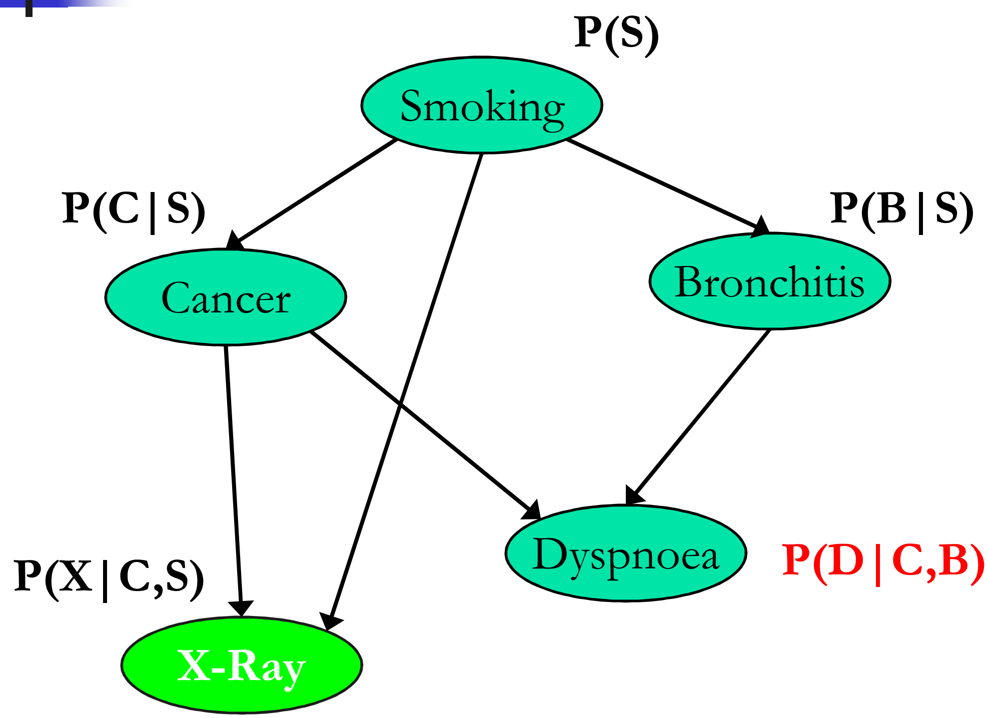
- Variables $\Rightarrow$ Nodes
- Constraints $\Rightarrow$ Edges
- e.g.:

$\boldsymbol{f_1(x_1,x_2,x_3)}$
$\boldsymbol{f_2(x_2,x_3,x_5)}$
$\boldsymbol{f_3(x_1,x_4)}$
$\boldsymbol{f_4(x_4,x_5)}$

$$\min_{t \in Sol}\{\sum_{i=1}^{m'} f_i(t)\}$$



Probabilistic Networks
(Pearl 1988)



P(D|C,B)

C	B	D=0	D=1
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

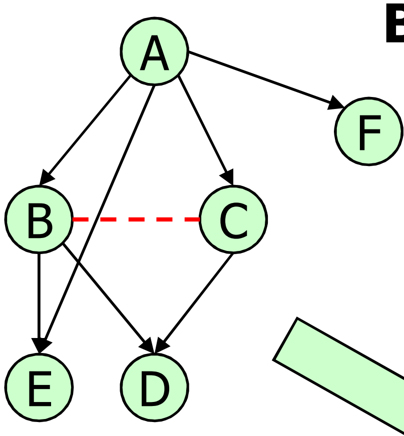
$$P(S,C,B,X,D) = P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B)$$

Belief Updating, Most probable tuple (MPE)

- P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?
- **MPE= find argmax** $P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B)$ =?

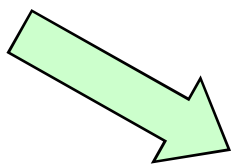
Mixed Networks
(Mateescu and Dechter, 2004)

Belief Network

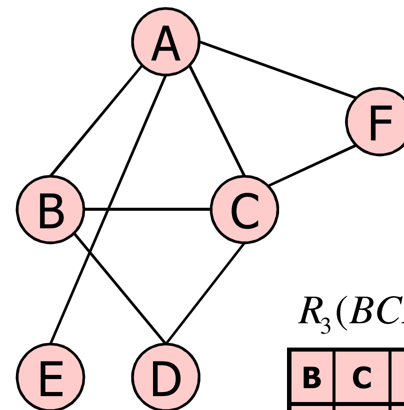


$P(D \mid B, C)$

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

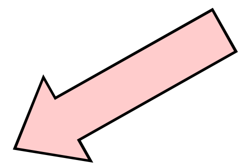


Constraint Network

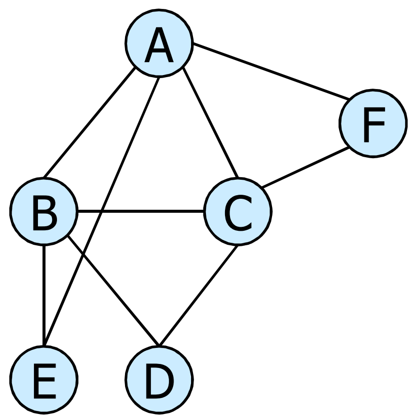


$R_3(BCD)$

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



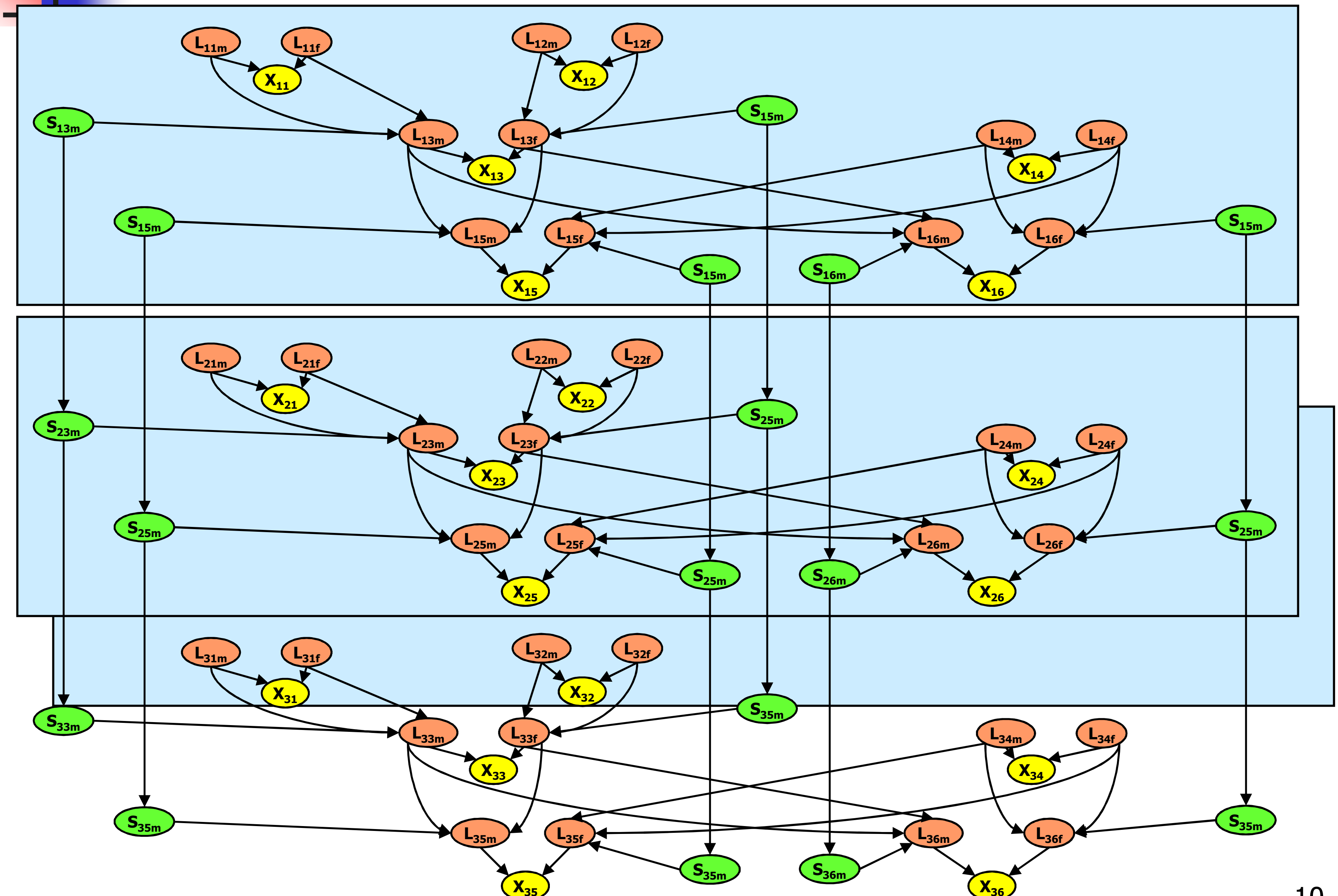
Moral mixed graph



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

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

Linkage analysis: 6 people, 3 markers



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
  - elimination (projection)

- Primary tasks:
  - **Belief updating**: $\Sigma_{X-y} \prod_j P_i$
  - **Combinarorial optimization**: $\max_X \prod_j P_j$
  - **Constraint satisfaction**: $\prod_X \times_j C_j$
  - **Max expected utility**

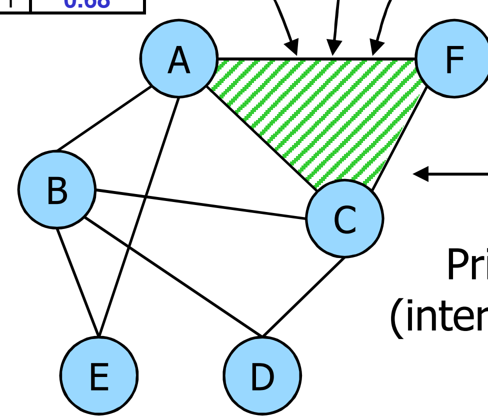
Conditional Probability Table (CPT)

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

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



$(\neg A \vee C \vee F)$

Primal graph (interaction graph)

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

Application Areas

- Constraints:
  - Scheduling, design, diagnosis, planning
- Belief networks, Markov fields:
  - Prediction, diagnosis, situation assessment, monitoring, learning
- Influence diagrams, Factored MDPS:
  - Planning and decision making under ucertainty.
- Decision making agents require
  - Constraints and probabilities to model the world.
  - Decision variable, and cost functions to model agents goals and actions.

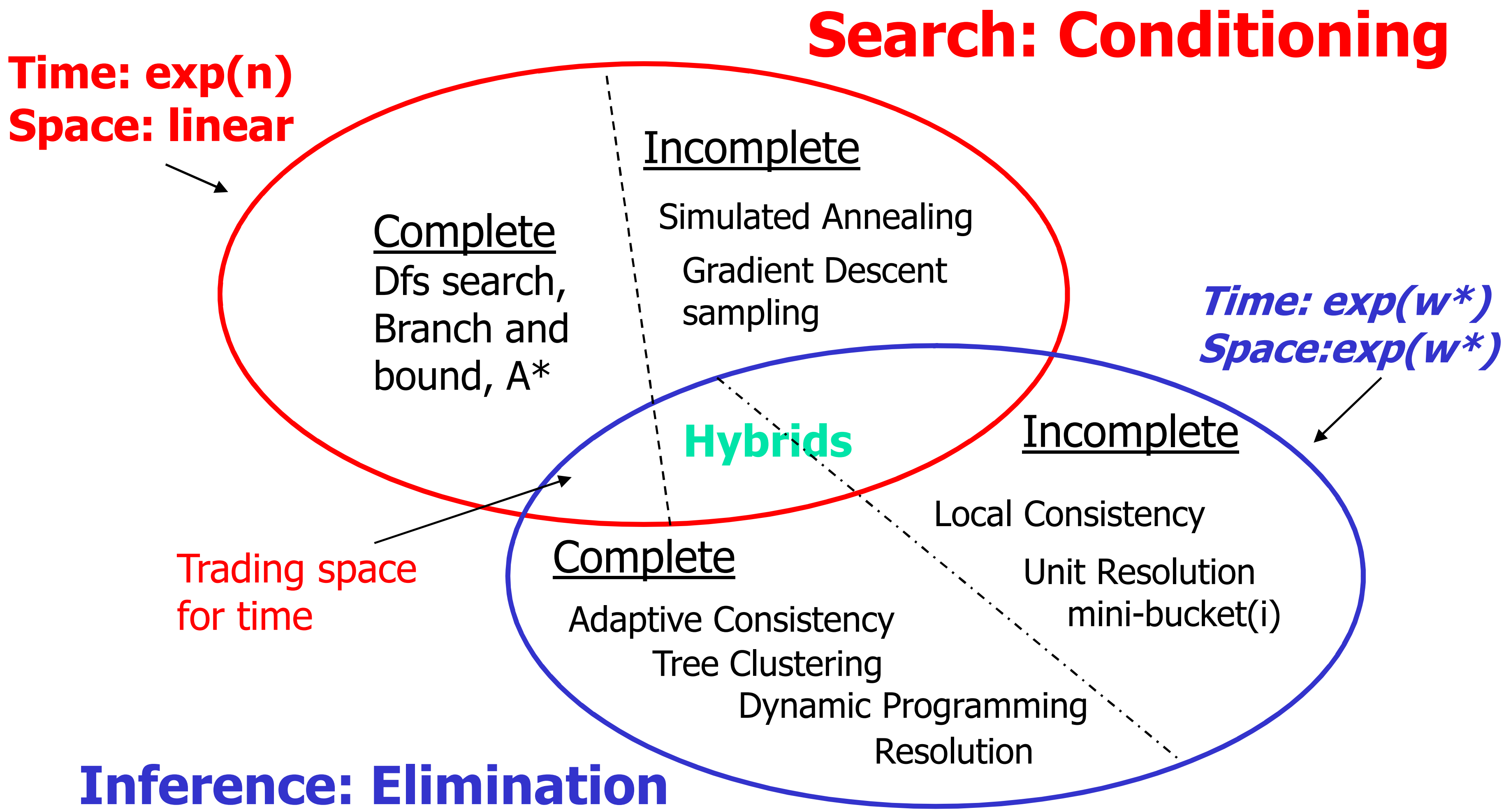
Sample Domains for Graphical Moldels

- Web Pages and Link Analysis
- Linkage analysis
- Communication Networks (Cell phone Fraud Detection)
- Natural Language Processing (e.g. Information Extraction and
- Semantic Parsing
- Object Recognition and Scene Analysis
- Battle-space Awareness
- Epidemiological Studies
- Citation Networks
- Geographical Information Systems
- Intelligence Analysis (Terrorist Networks)
- Financial Transactions (Money Laundering)
- Computational Biology

...

Solution Techniques

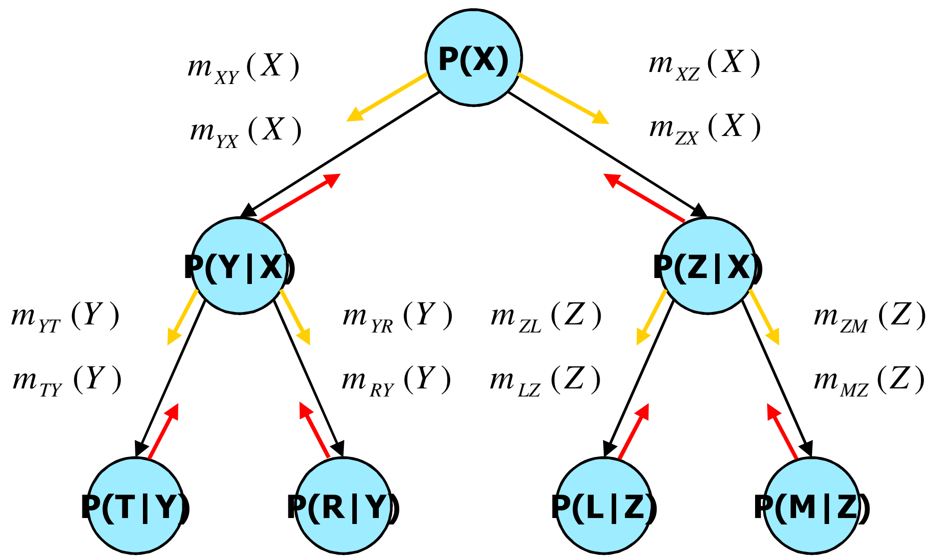
All queries are NP-hard so: exploit structure, identify tractable classes, approximate



Tree-solving is Easy

Belief updating (sum-prod)

CSP – consistency (projection-join)

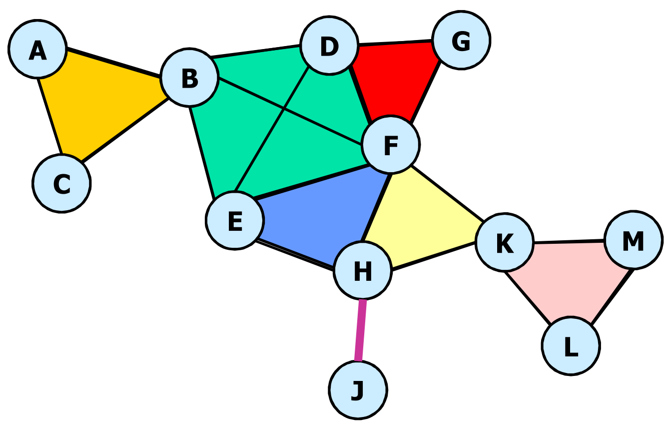


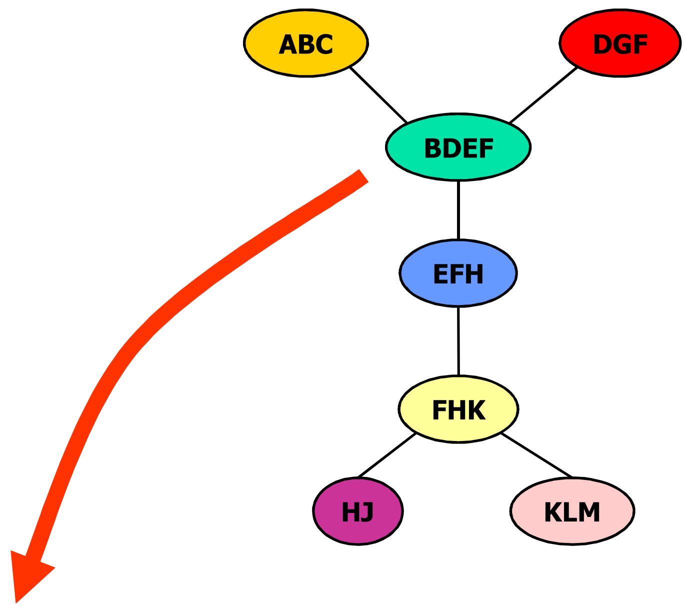
MPE (max-prod)

#CSP (sum-prod)

Trees are processed in linear time and memory
Also Acyclic graphical models

Inference and Treewidth





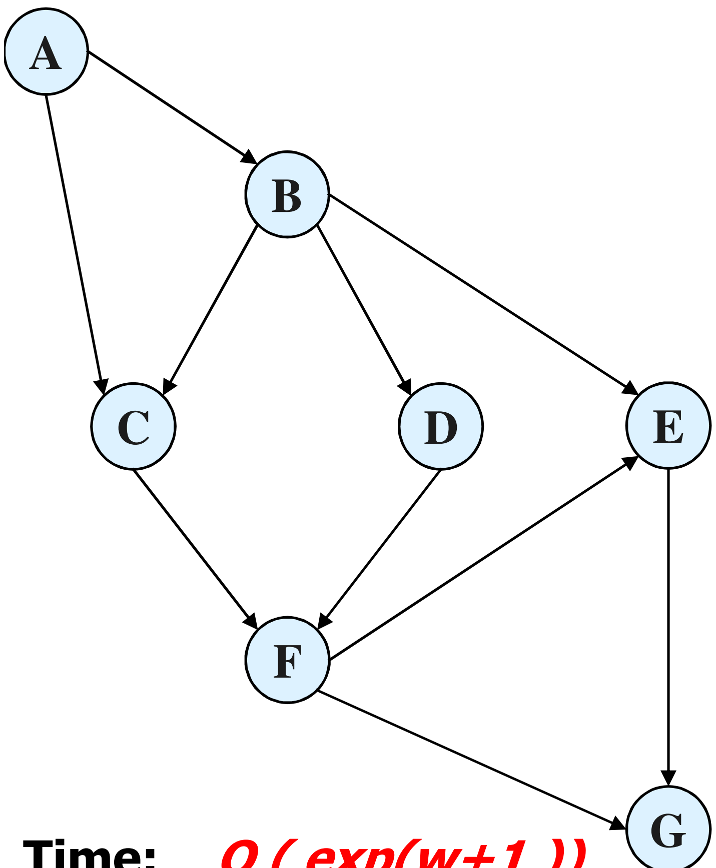
Inference algorithm:
Time: exp(tree-width+1)

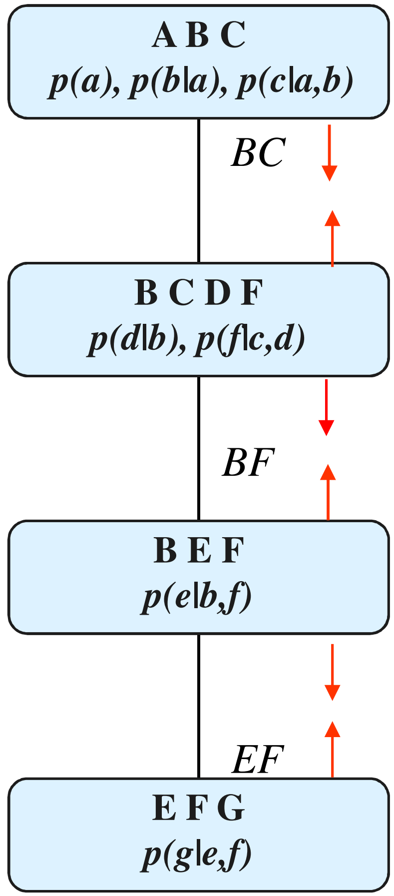
Space: exp(separator-width)

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

Cluster Tree Propagation

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





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

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

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

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

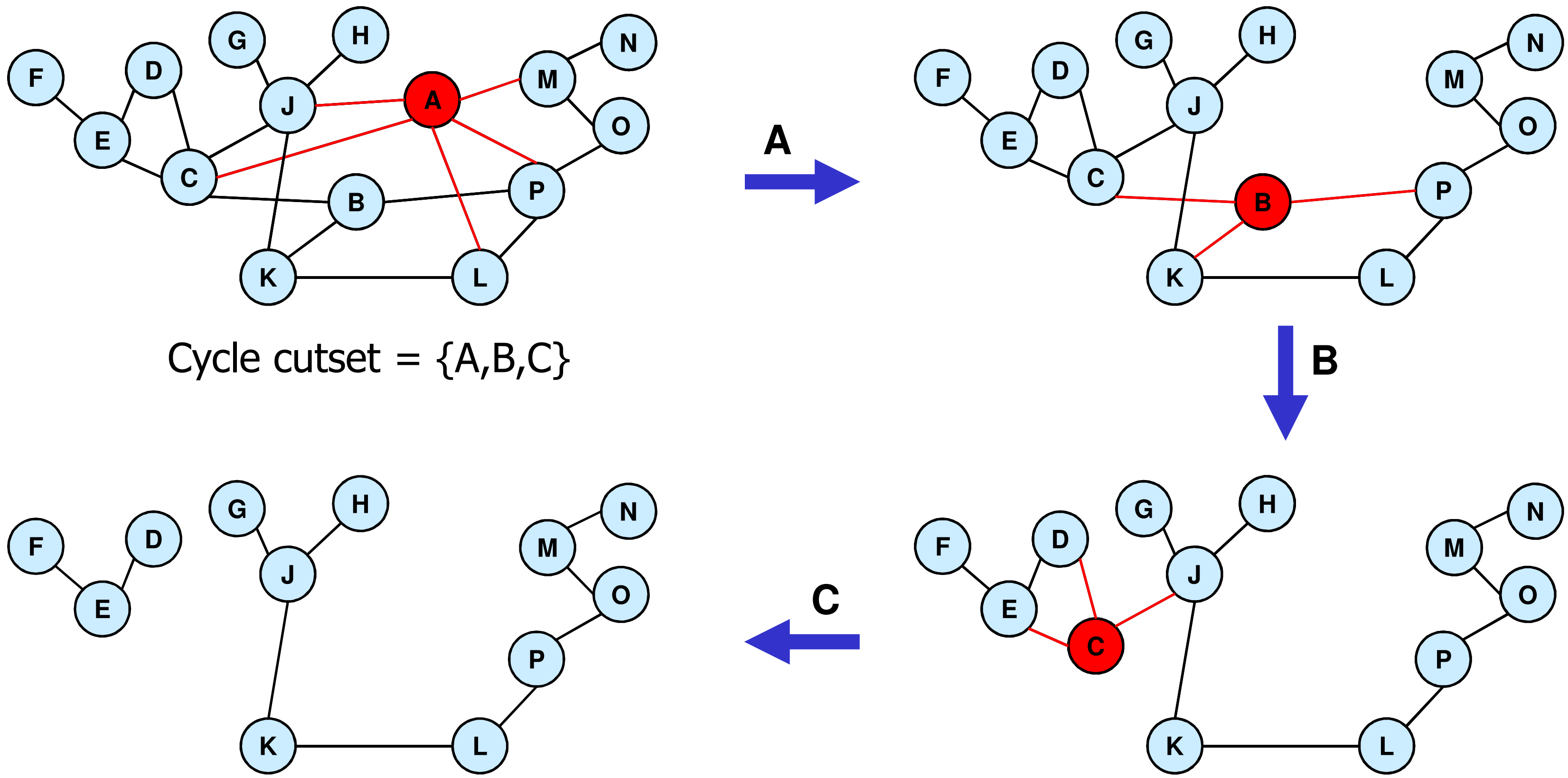
$$h_{(3,4)}(e,f) = \sum_b p(e \mid b,f) \cdot h_{(2,3)}(b,f)$$

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

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

For each cluster P(X|e) is computed

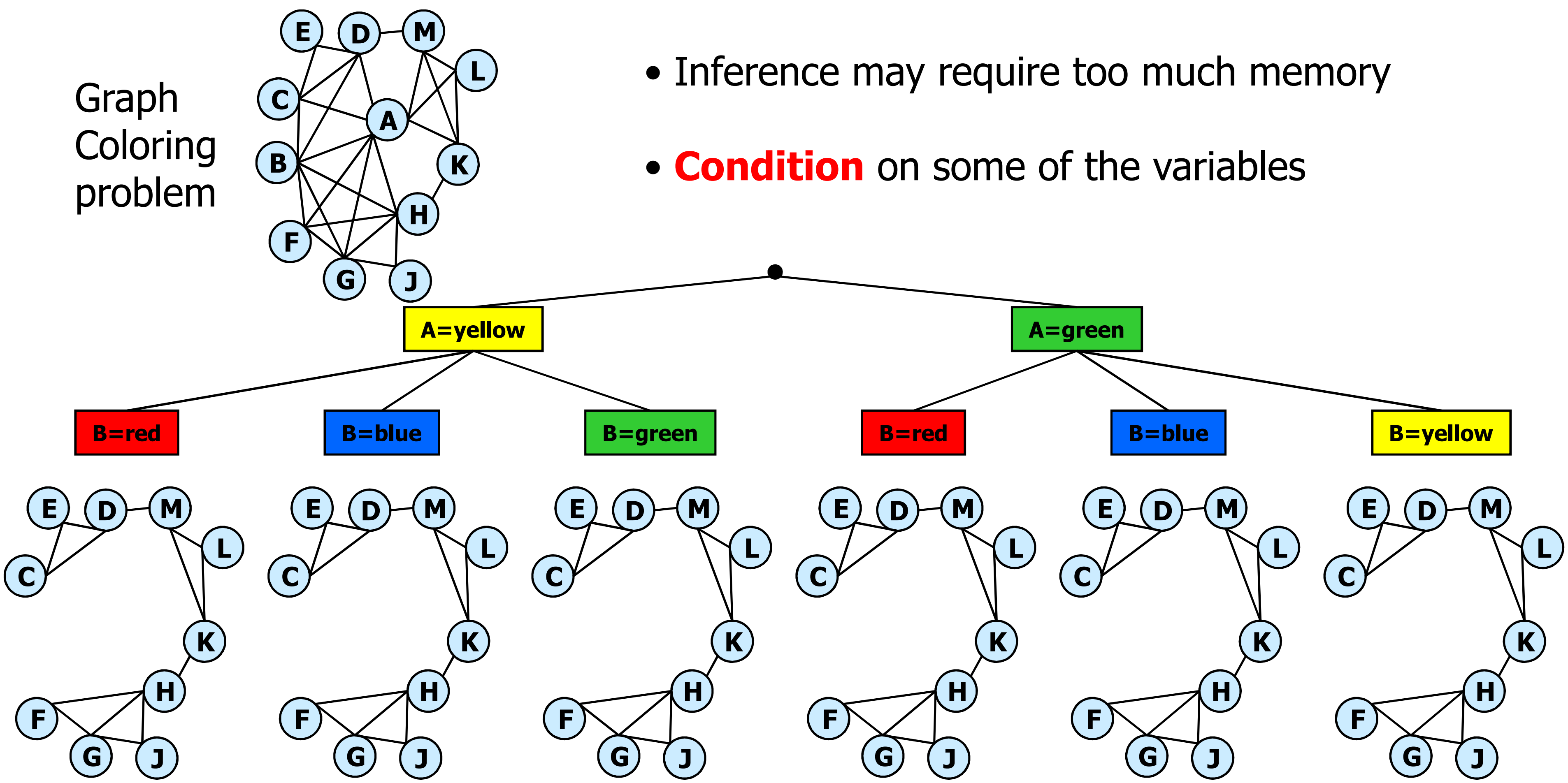
Conditioning and Cycle cutset



Cycle cutset = {A,B,C}

Search over the Cutset (cont)

Graph Coloring problem

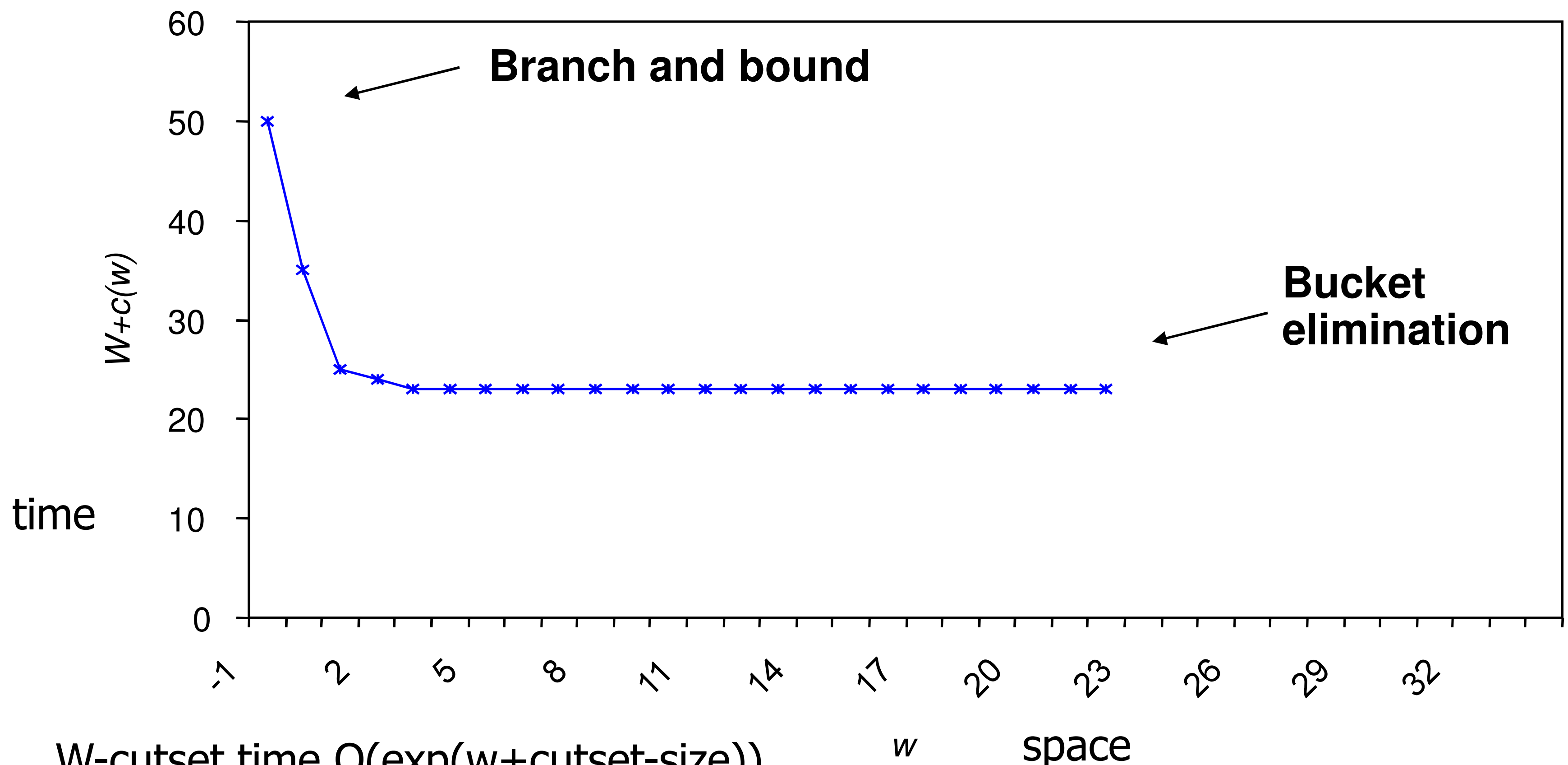


- Inference may require too much memory
- **Condition** on some of the variables

Time vs Space for w-cutset

(Dechter and El-Fatah, 2000)
(Larrosa and Dechter, 2001)
(Rish and Dechter 2000)

- **Random Graphs (50 nodes, 200 edges, average degree 8, $w^* \approx 23$)**



W-cutset time O(exp(w+cutset-size))
Space O(exp(w))

Approximation

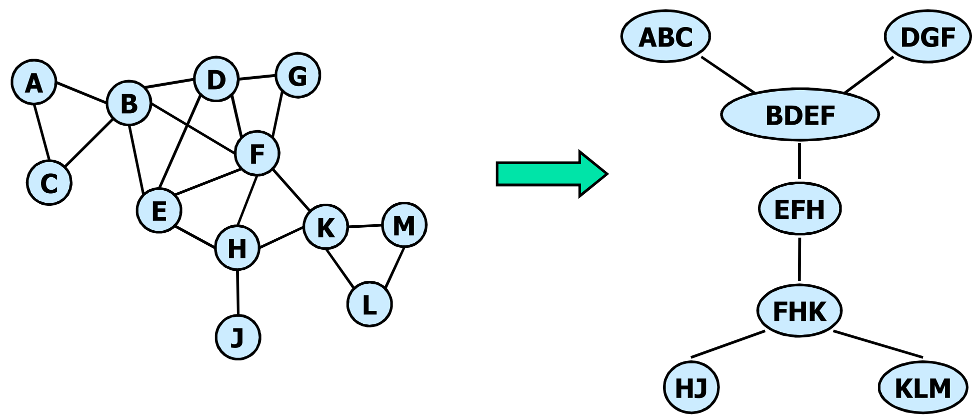
- Since inference, search and hybrids are too expensive when graph is dense; (high treewidth) then:

- Bounding inference:
    - mini-bucket and mini-clustering
    - Belief propagation

- Bounding search:
    - Sampling

- Goal: an anytime scheme

Inference vs Search



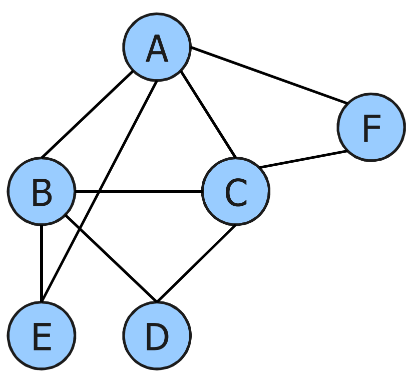
Inference

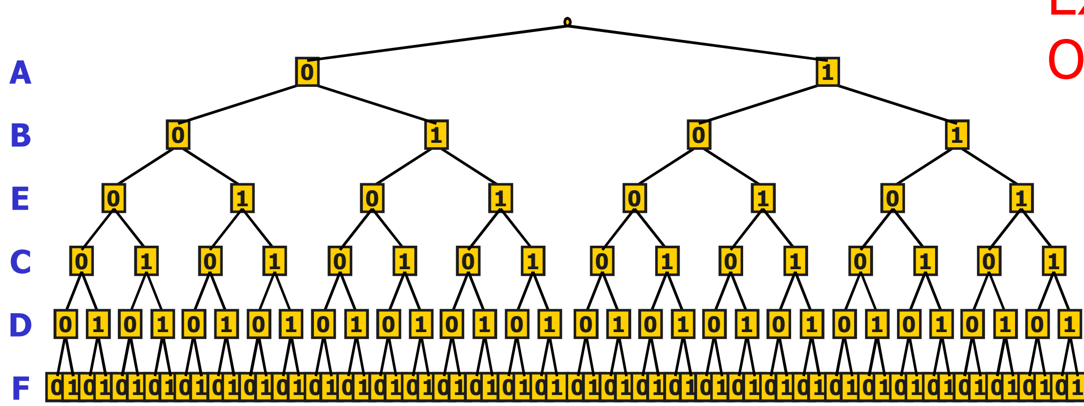
- decomposition
- equivalence

Exp(w*) time/space

Search

- Pruning → Exp(n) time O(n) space

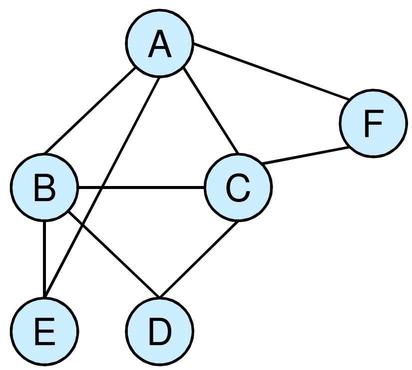




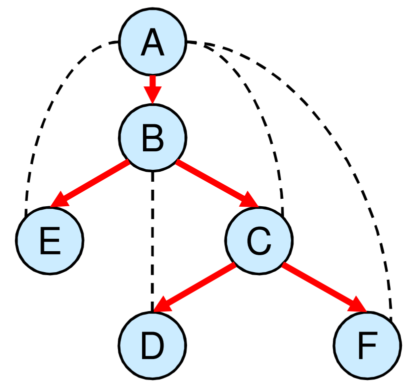
Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.
- AND/OR search spaces
  - Decomposition in AND/OR trees
  - Equivalence in AND/OR Graphs
- AND/OR search for combinatorial optimization
- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

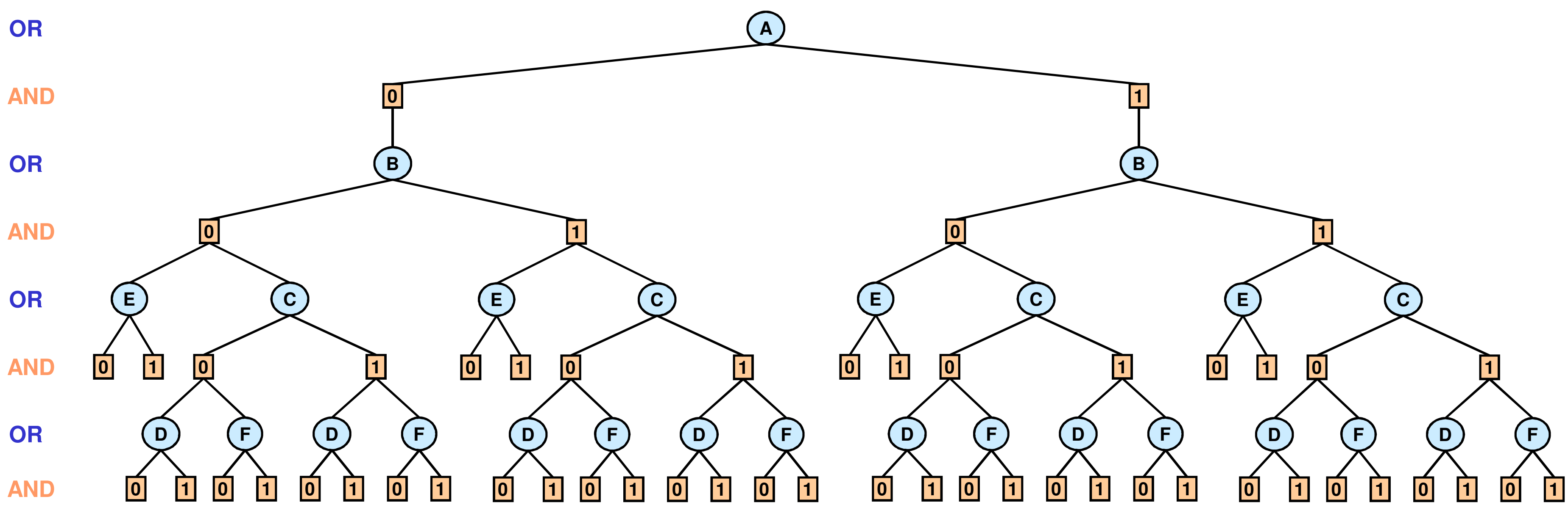
AND/OR Search Space



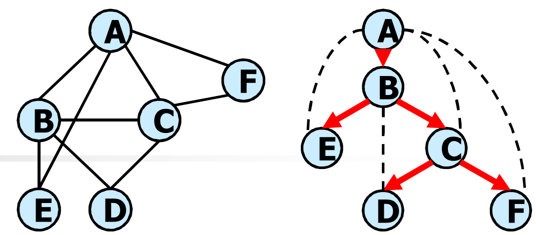
Constraint network

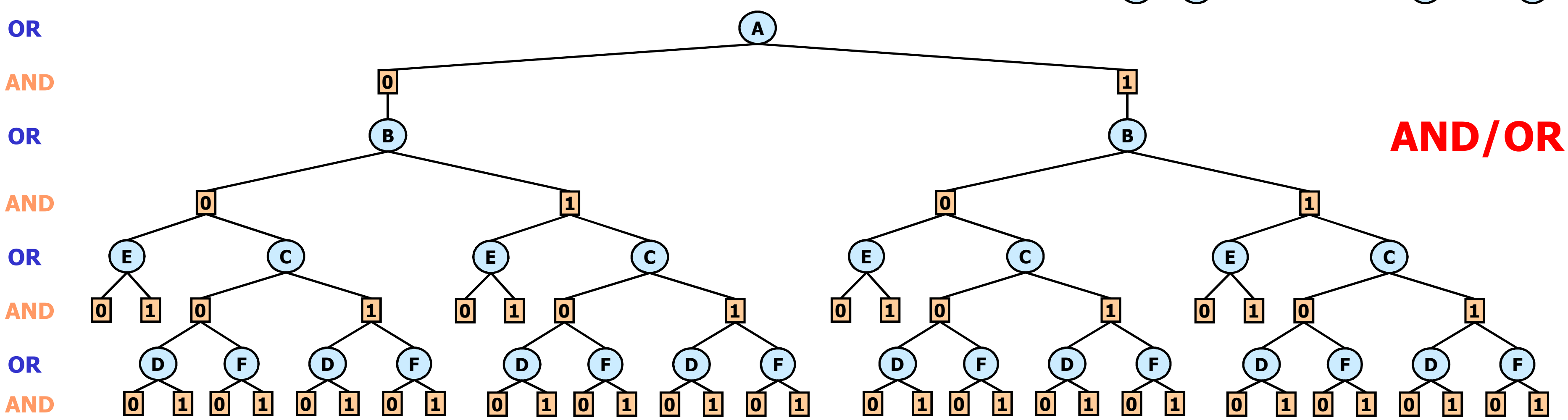


DFS tree

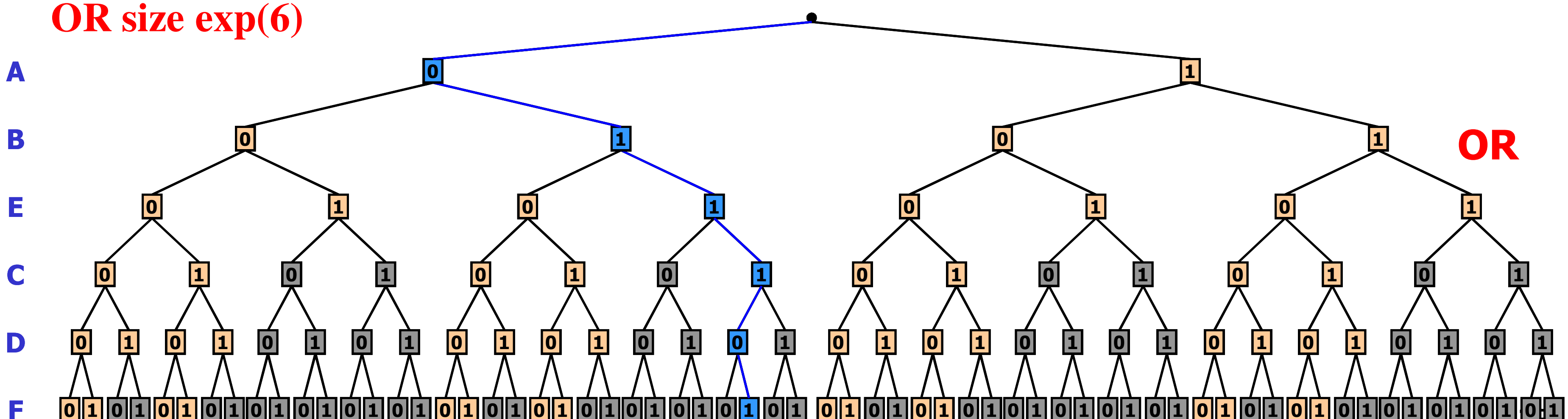


AND/OR vs. OR





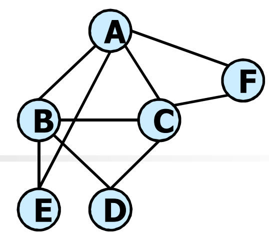
AND/OR size: exp(4),
OR size exp(6)

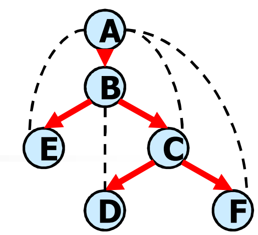


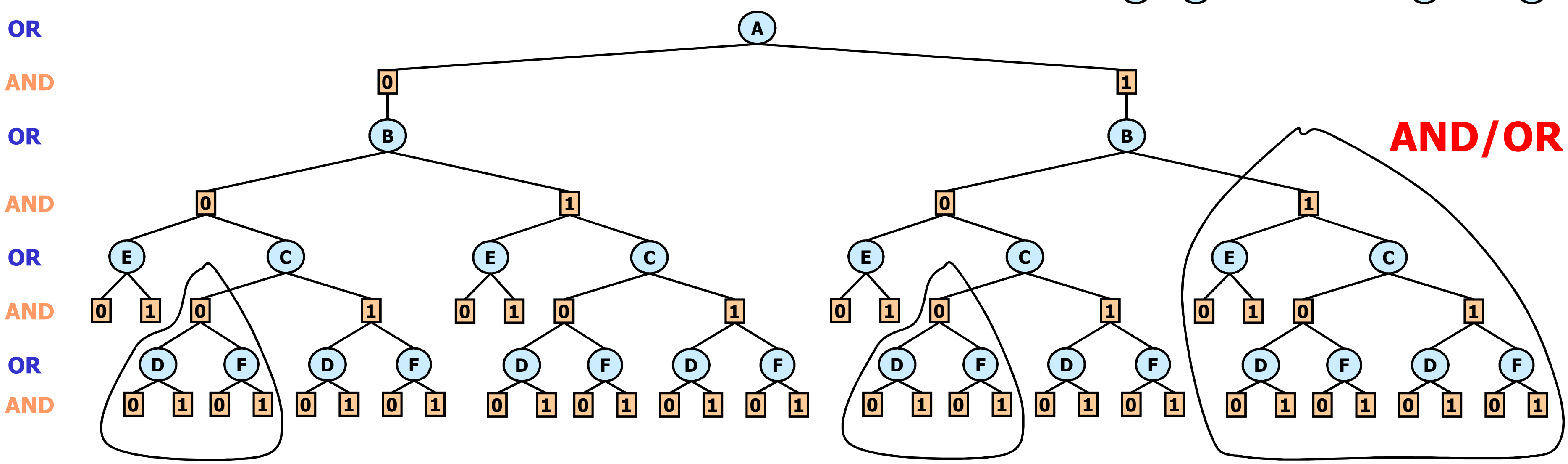
AND/OR vs. OR

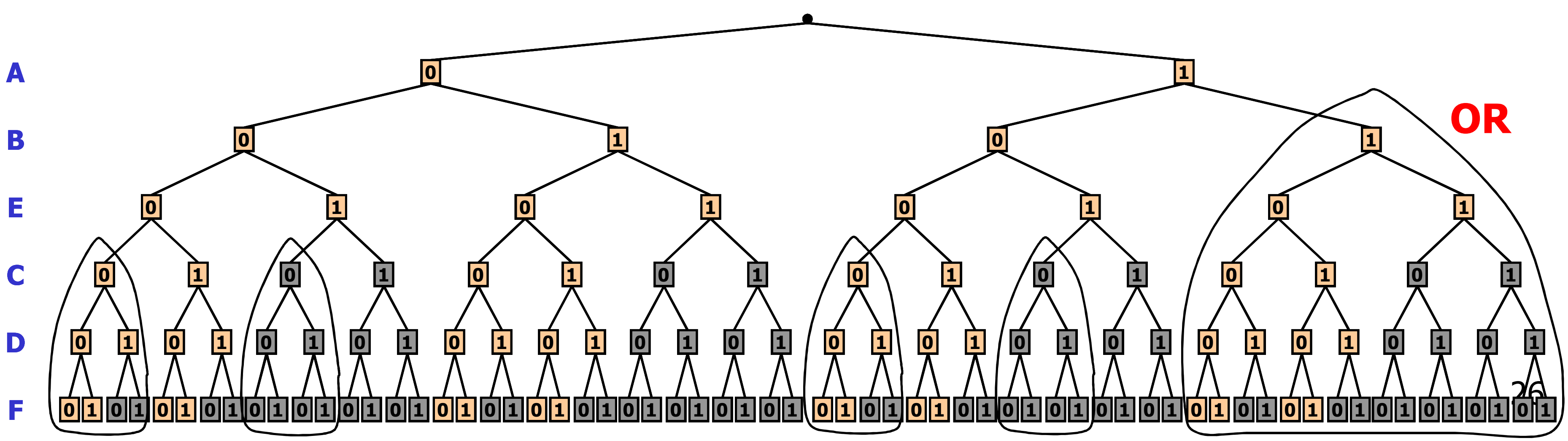
with Constraints

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





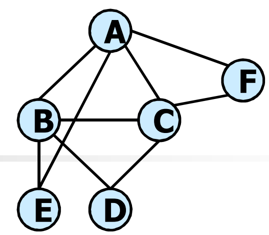


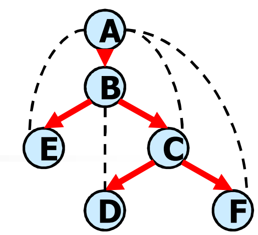


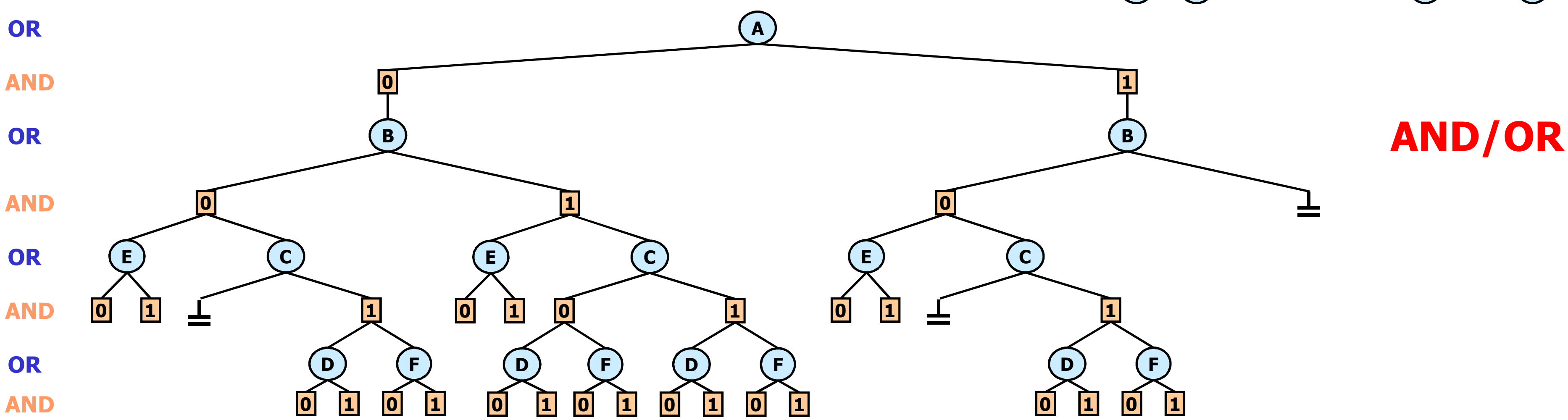
AND/OR vs. OR

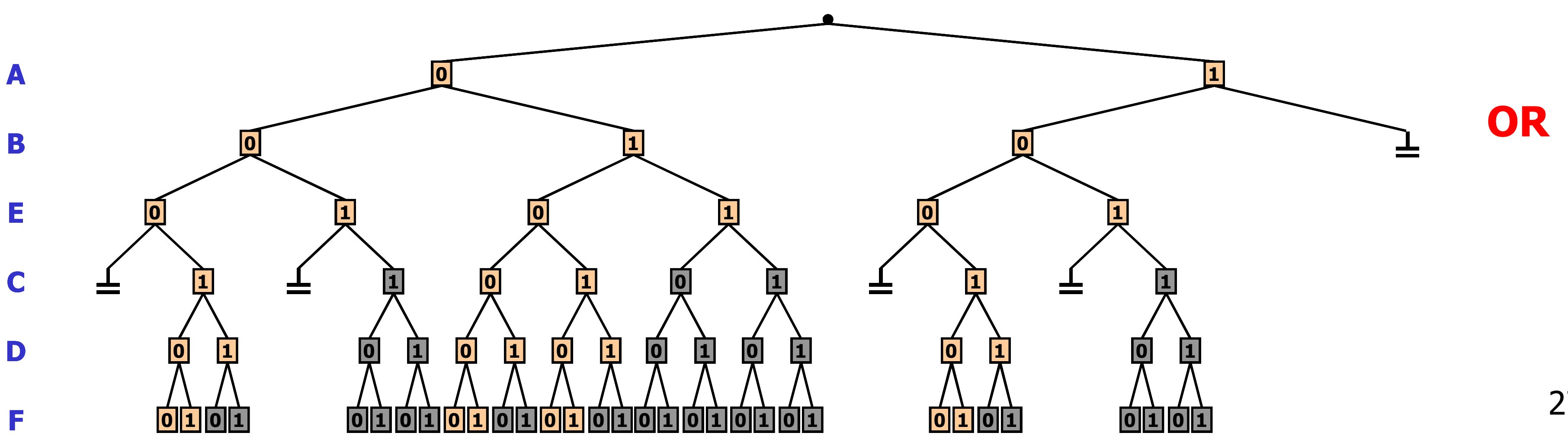
with Constraints

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



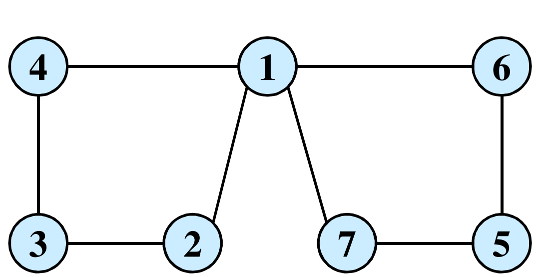






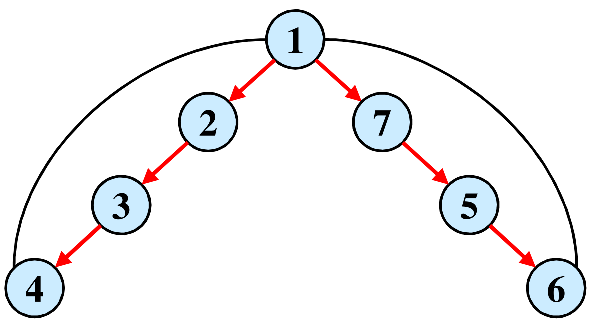
Pseudo-Trees

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

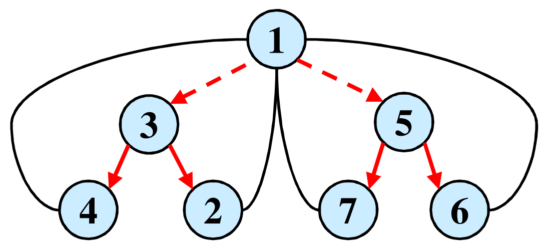


(a) Graph

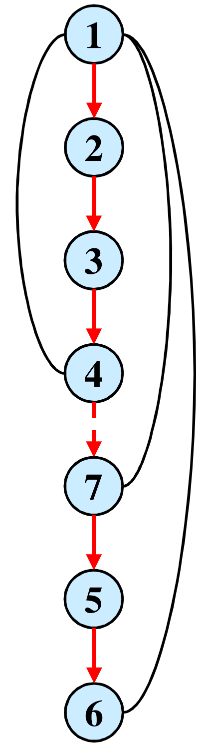
m <= w* log n



(b) DFS tree
depth=3

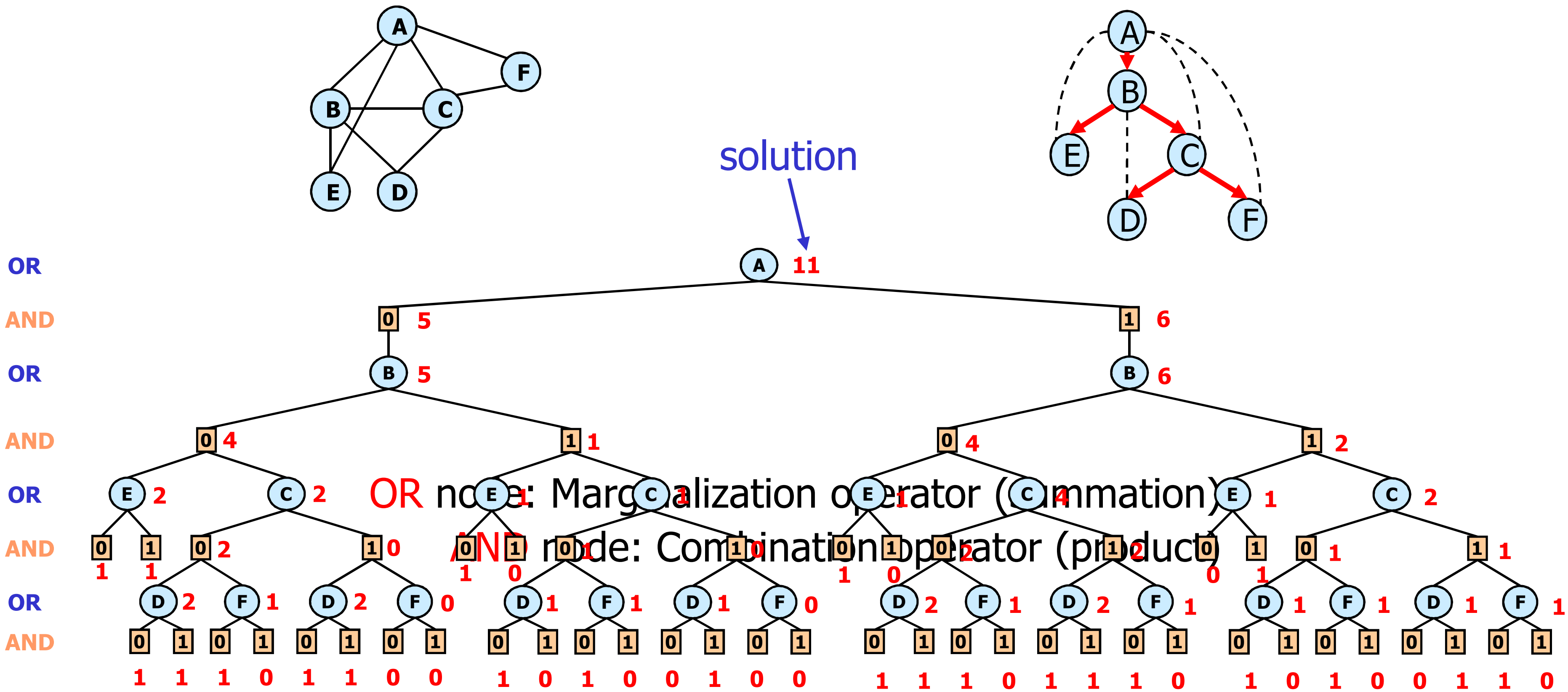


(c) pseudo- tree
depth=2



(d) Chain
depth=6

DFS algorithm (#CSP example)



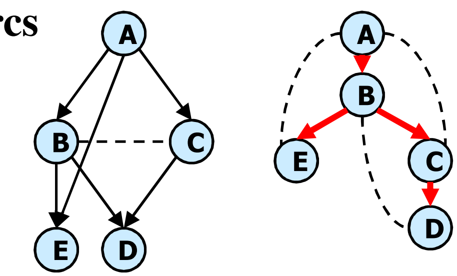
OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = number of solutions below it

AND/OR tree search (belief updating)

Weighted AND/OR
Has weights on arcs



$P(E \mid A,B)$

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

$P(B \mid A)$

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

$P(C \mid A)$

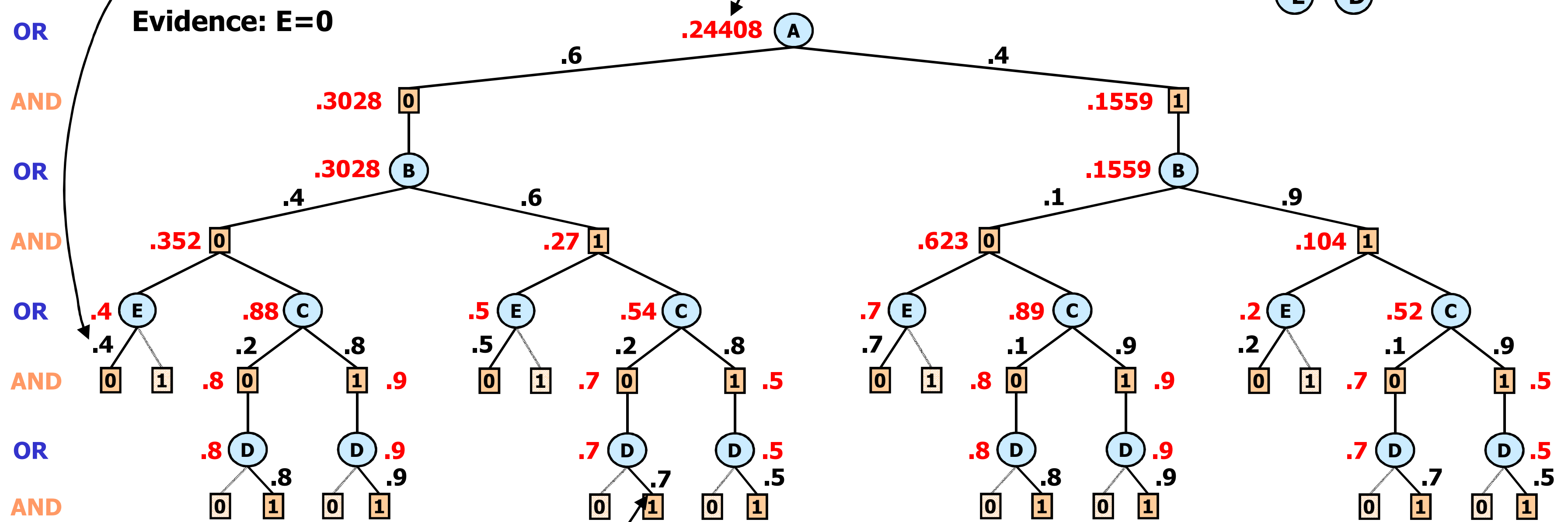
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4

Evidence: E=0

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



$P(D \mid B,C)$

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

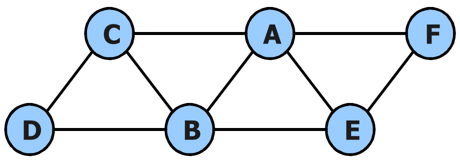
Evidence: D=1

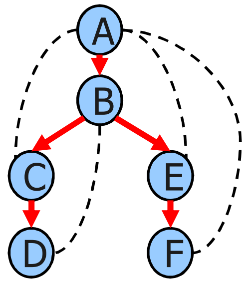
OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = updated belief for subproblem 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

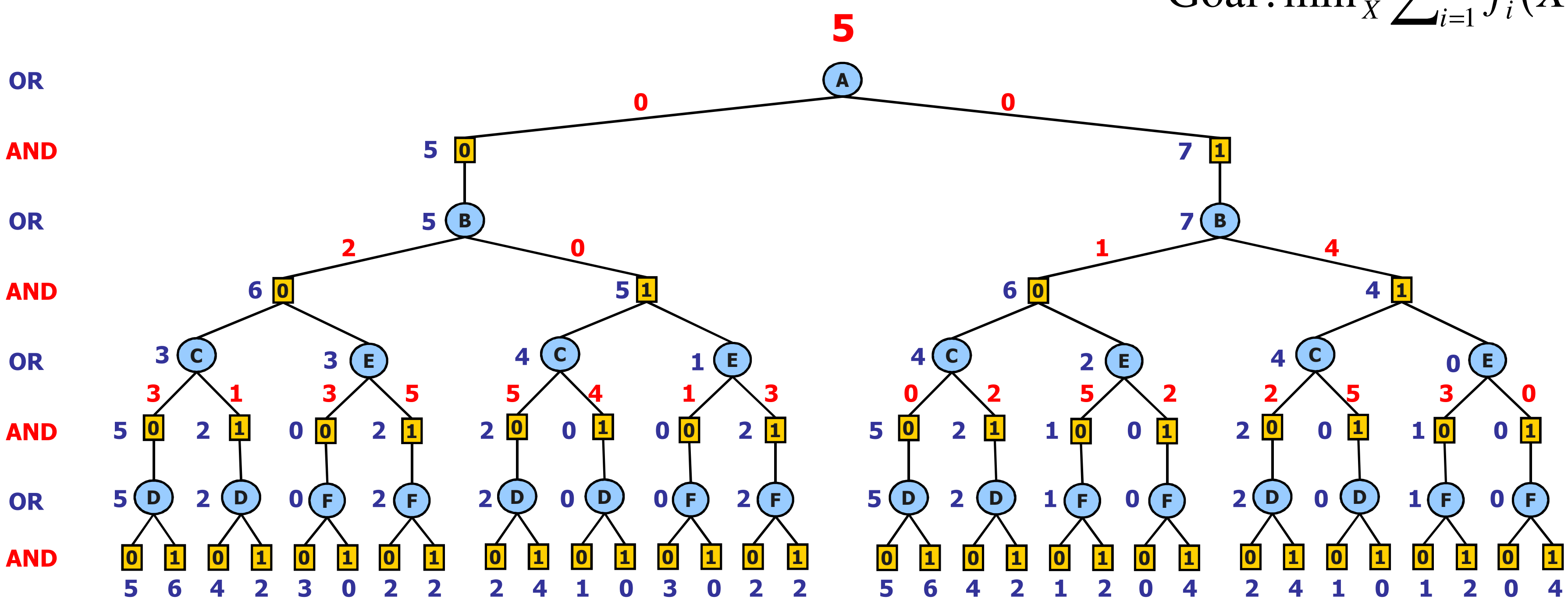
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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)

Complexity of AND/OR Tree Search

SAT: Backjumping will do
Counting: special care needed

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

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

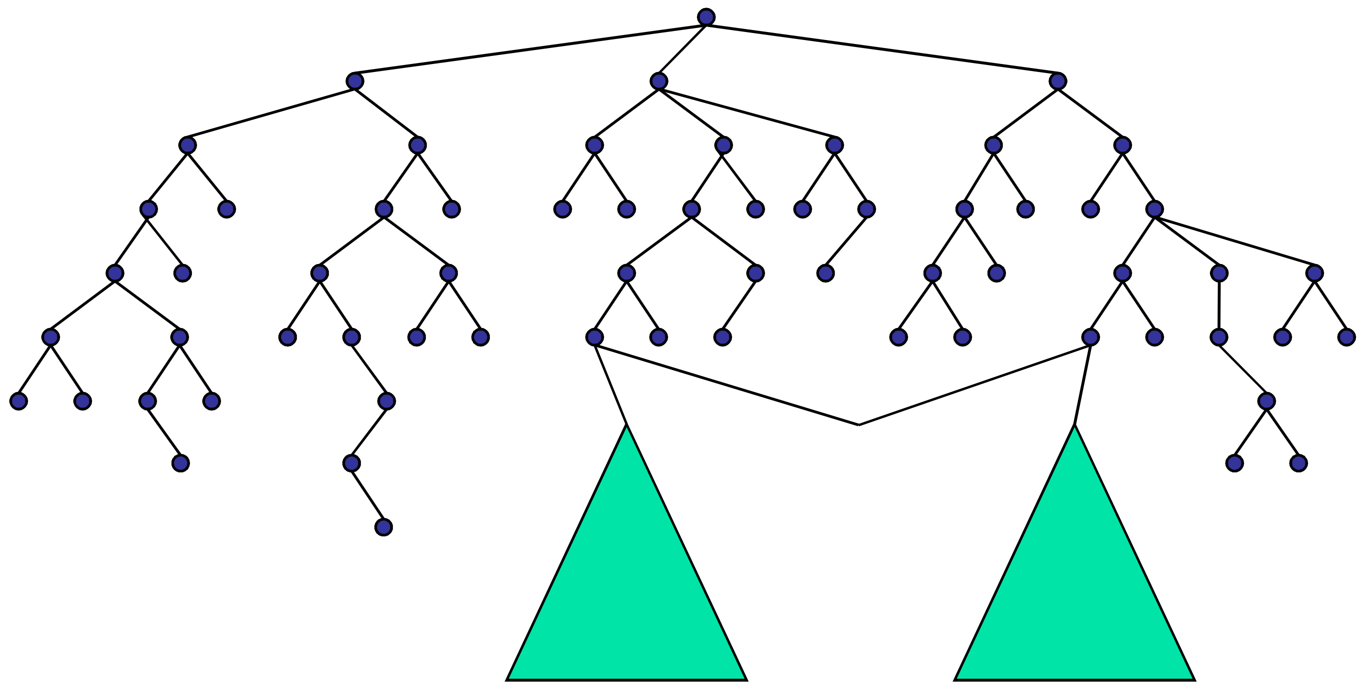
Tasks: Consistency, Counting, Optimization, Belief updating Max-expected utility, partition function

Overview

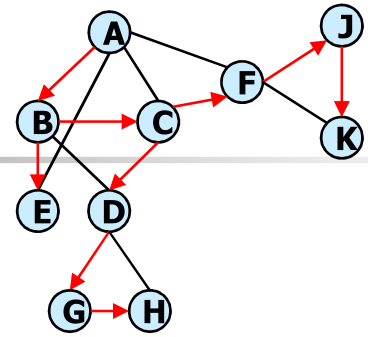
- Introduction to graphical models algorithms: Inference, search and hybrids.
- AND/OR search spaces
  - Decomposition in AND/OR trees
  - Equivalence AND/OR Graphs
- AND/OR search for combinatorial optimization
- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

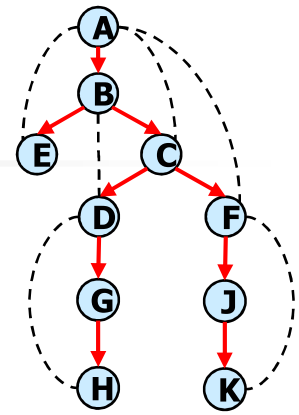
From Search Trees to Search Graphs

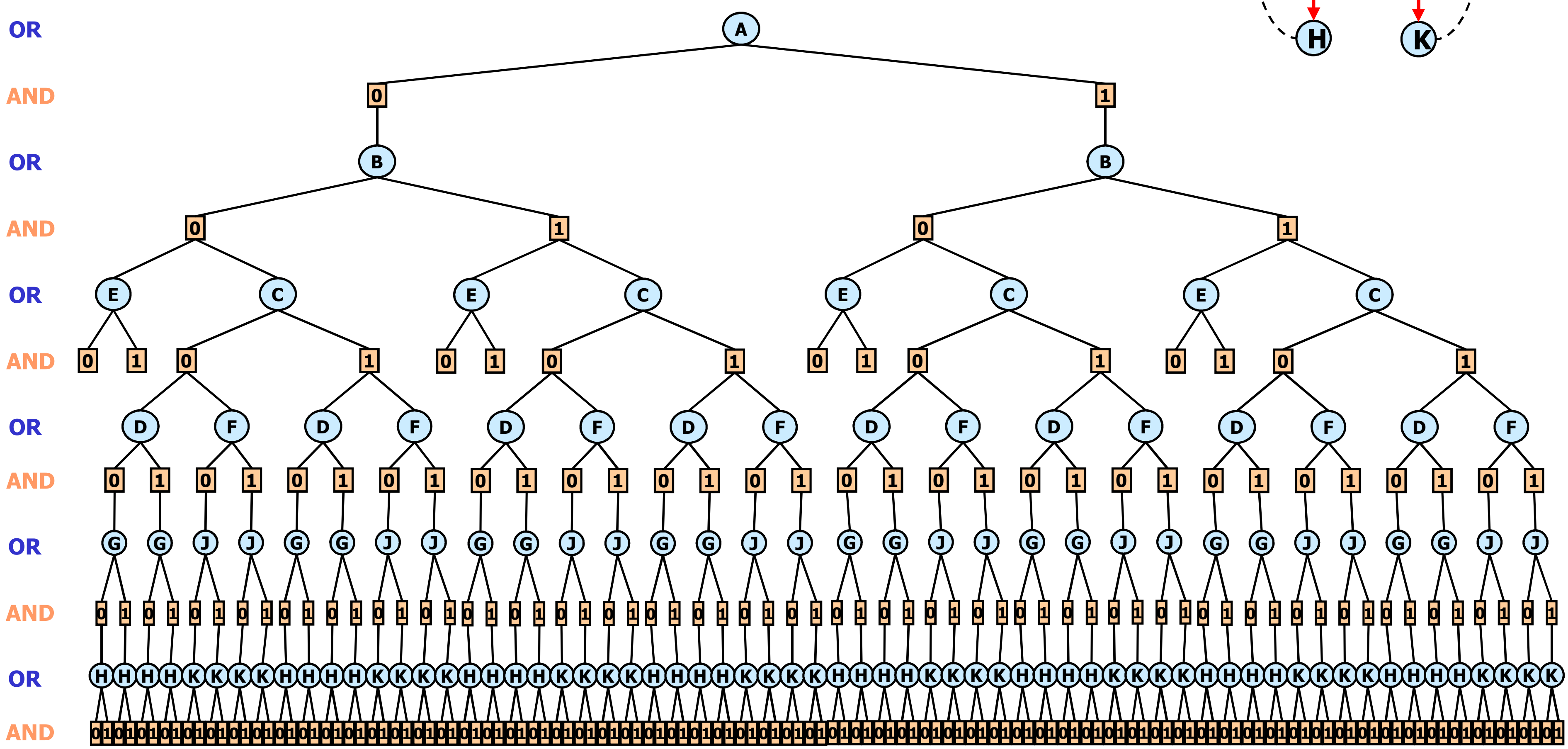
- Any two nodes that root identical subtrees (subgraphs) can be merged



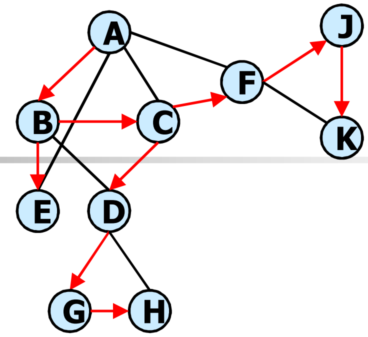
From AND/OR Tree

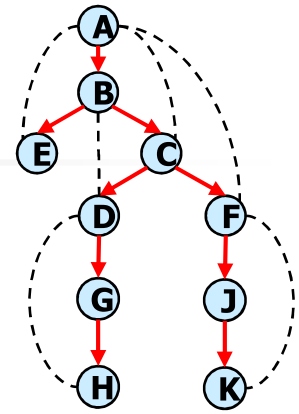


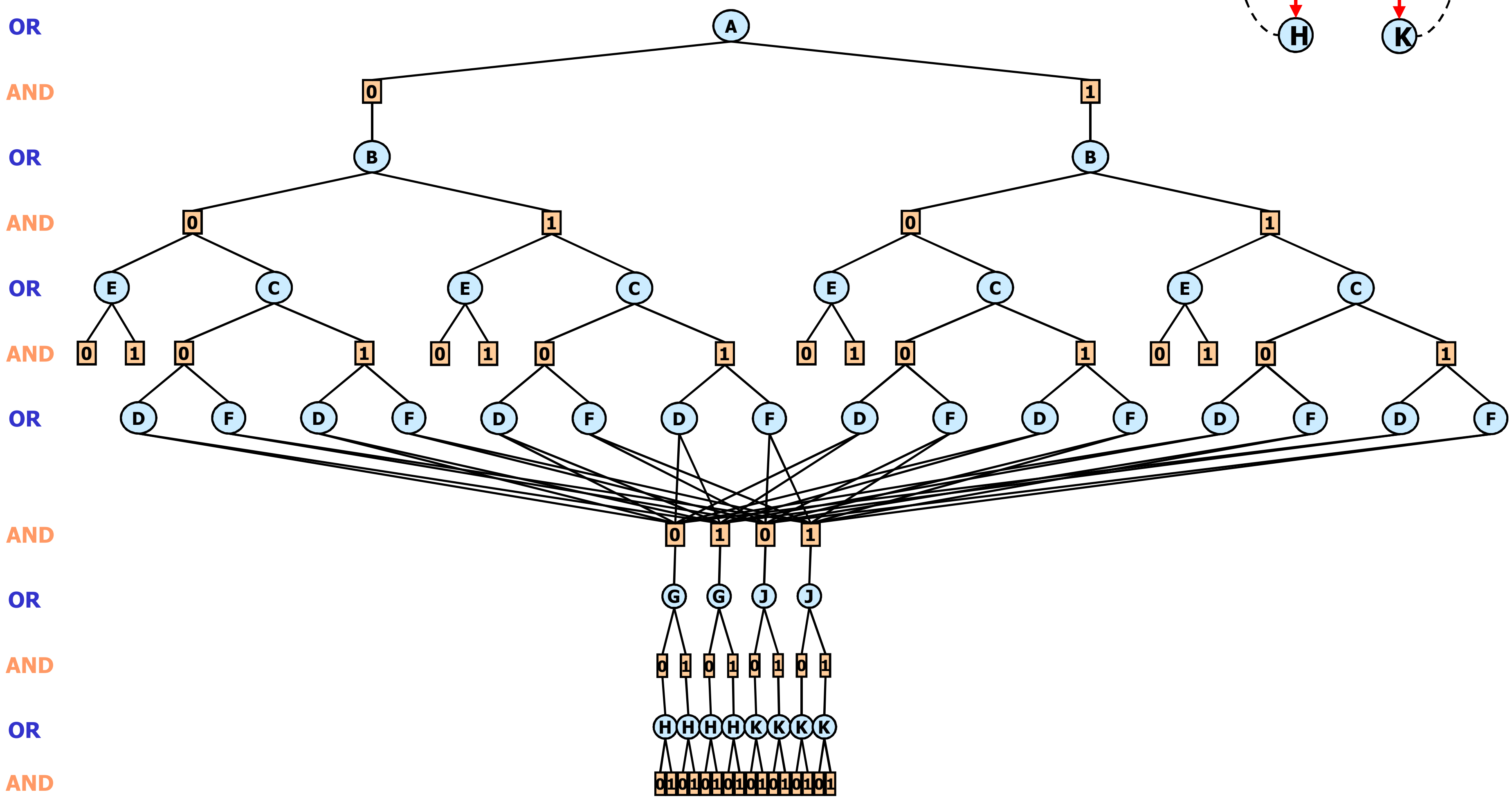




An AND/OR Graph



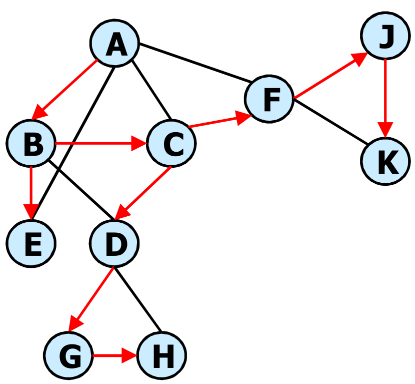


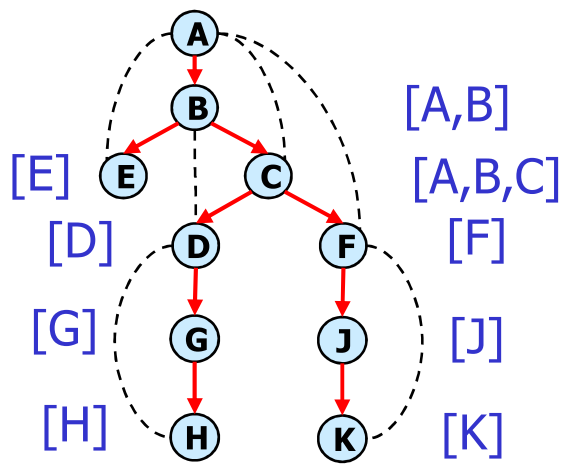


Context-based Caching

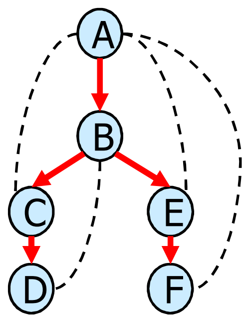
- context = current variable + ancestors connected to subtree below

- Caching is possible when context is the same

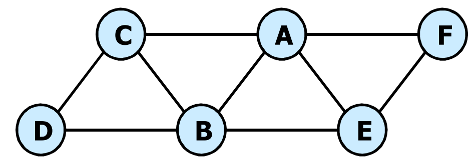




Context-based Caching

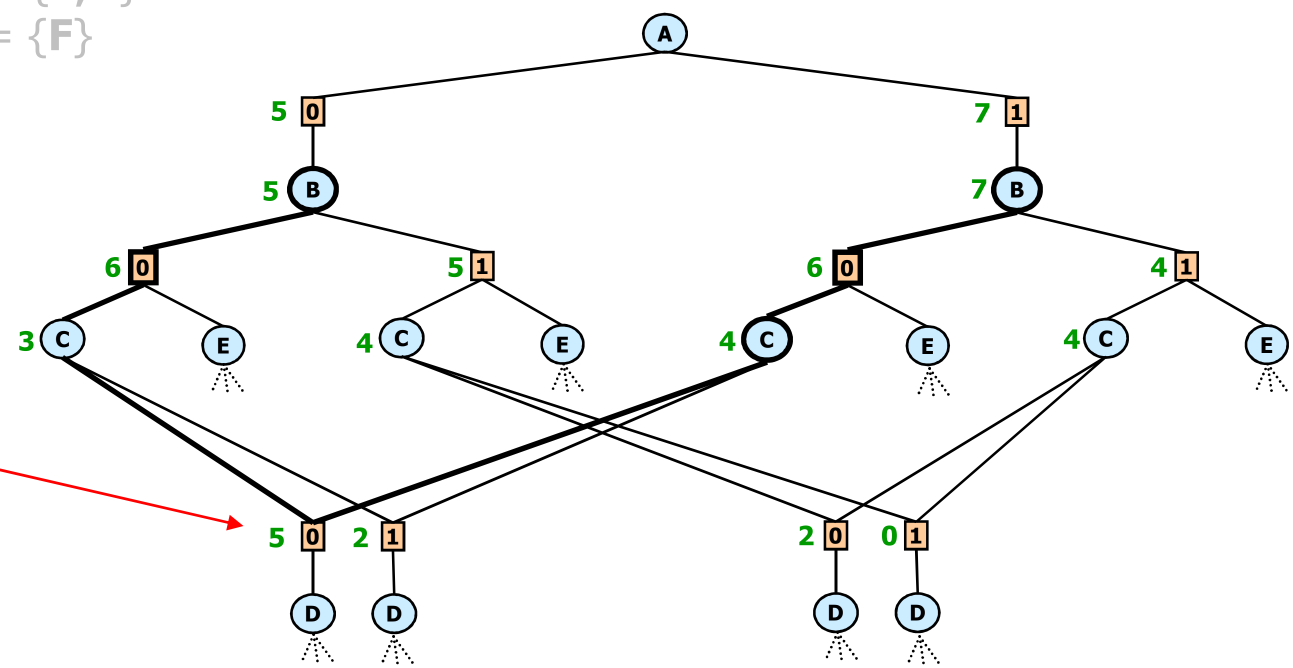


context(A) = {**A**}
context(B) = {**B**,**A**}
context(C) = {C,B}
context(D) = {**D**}
context(E) = {**E**,**A**}
context(F) = {**F**}



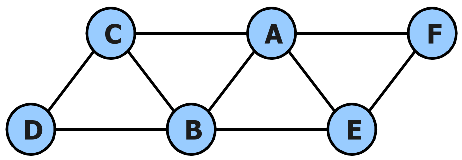
Cache Table (C)

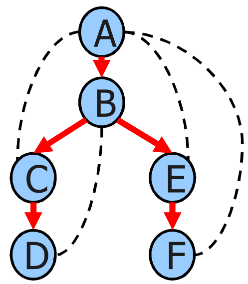
B	C	Value
0	0	5
0	1	2
1	0	2
1	1	0



Space: **O(exp(2))**

Example (graph search)





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

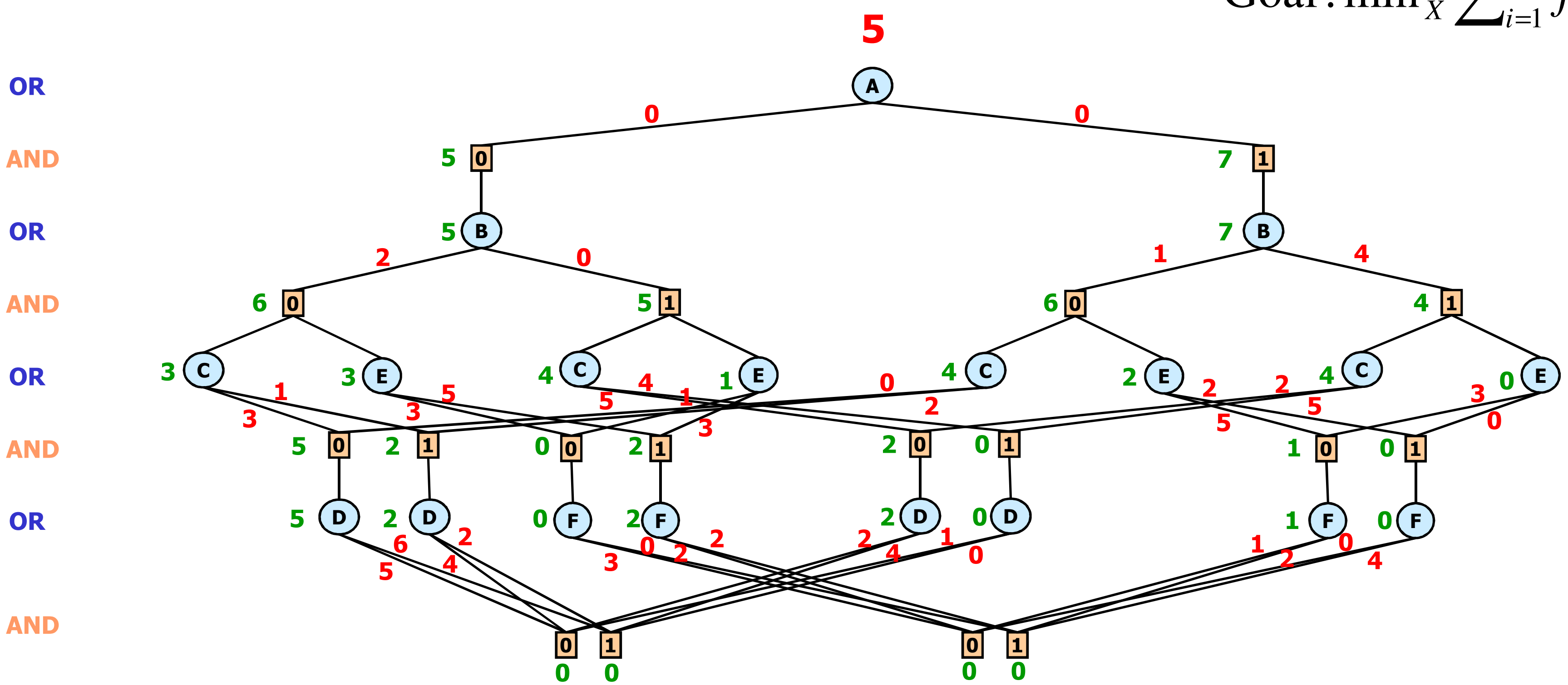
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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/OR Tree DFS Algorithm
(Belief Updating)

$P(E|A,B)$

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

Evidence: E=0

$P(B|A)$

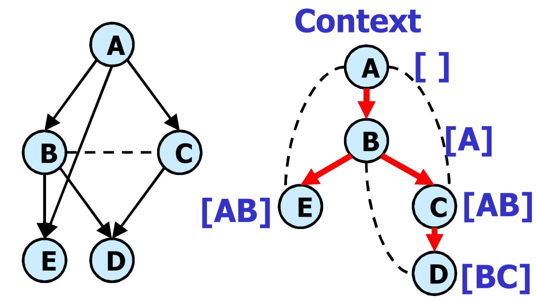
A	B=0	B=1
0	.4	.6
1	.1	.9

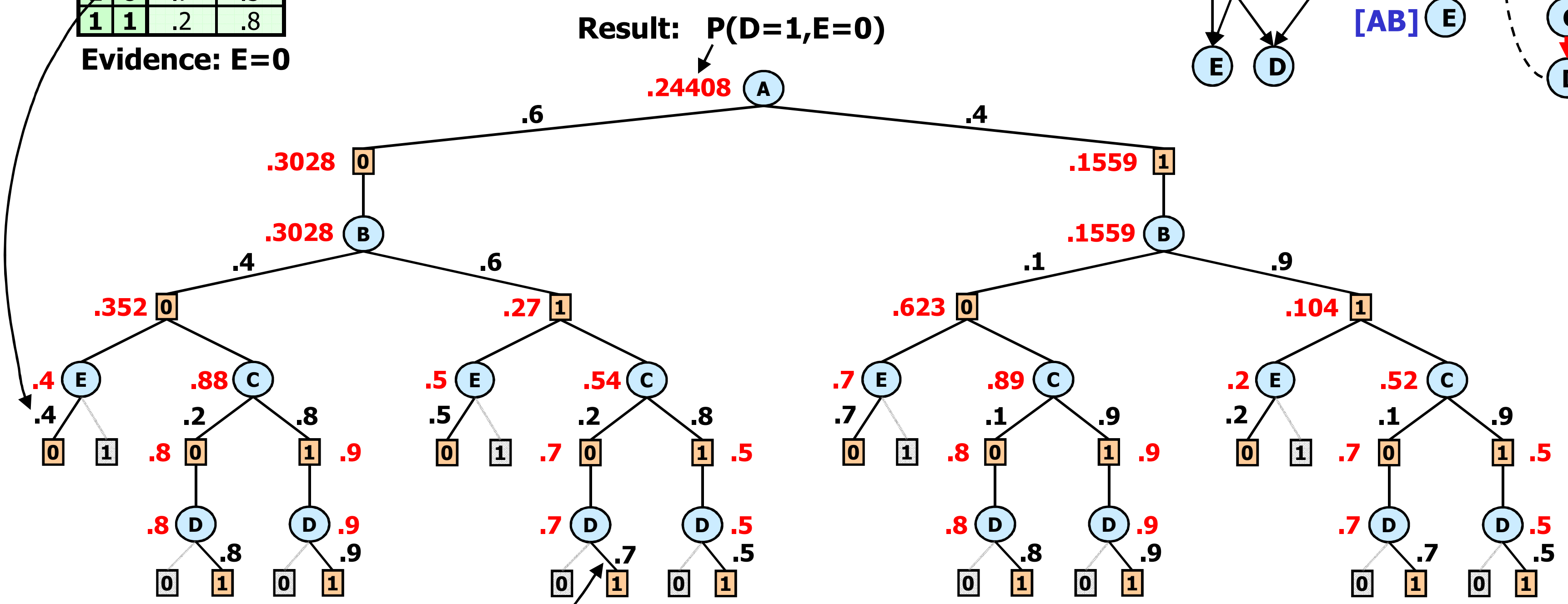
$P(C|A)$

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

$P(A)$

A	P(A)
0	.6
1	.4





$P(D|B,C)$

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

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

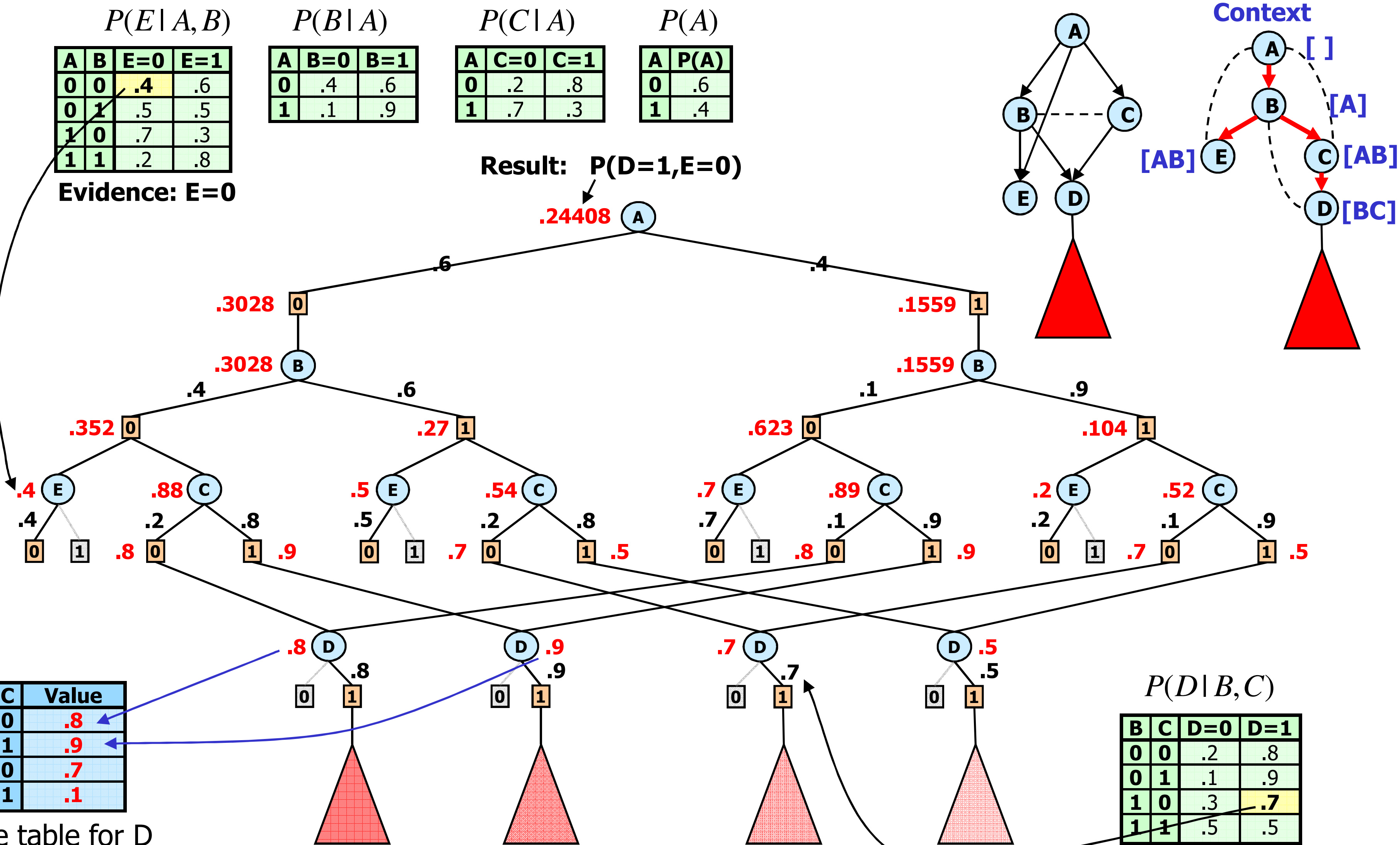
$P(C \mid A)$

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

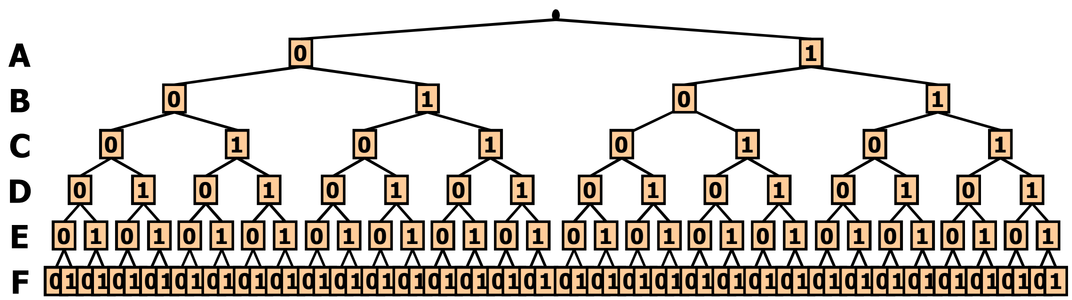
Cache table for D

$P(D \mid B,C)$

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

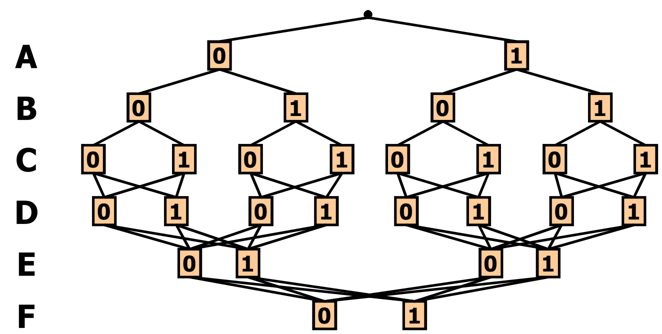
Evidence: D=1

All Four Search Spaces



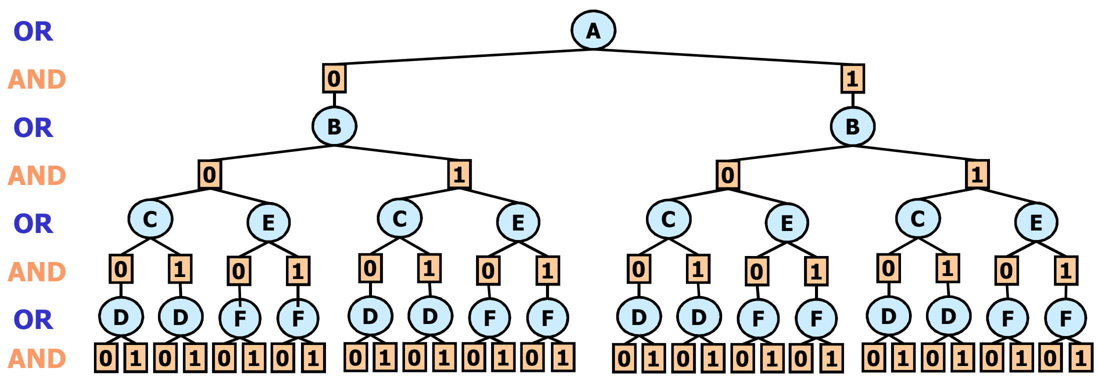
Full OR search tree

126 nodes



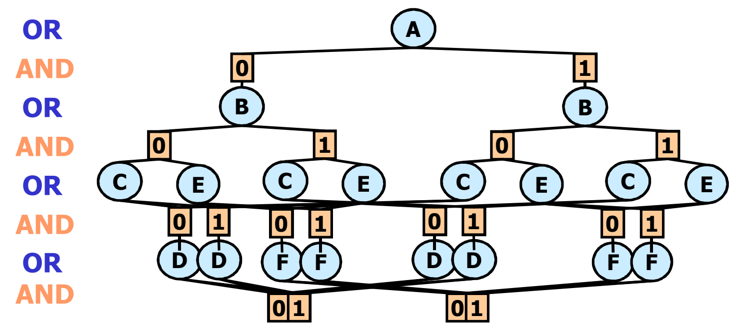
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes

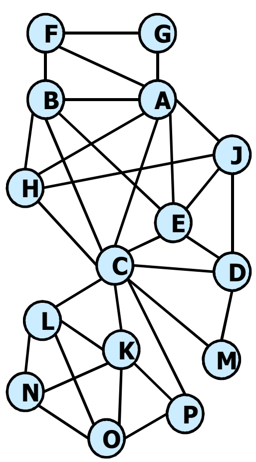


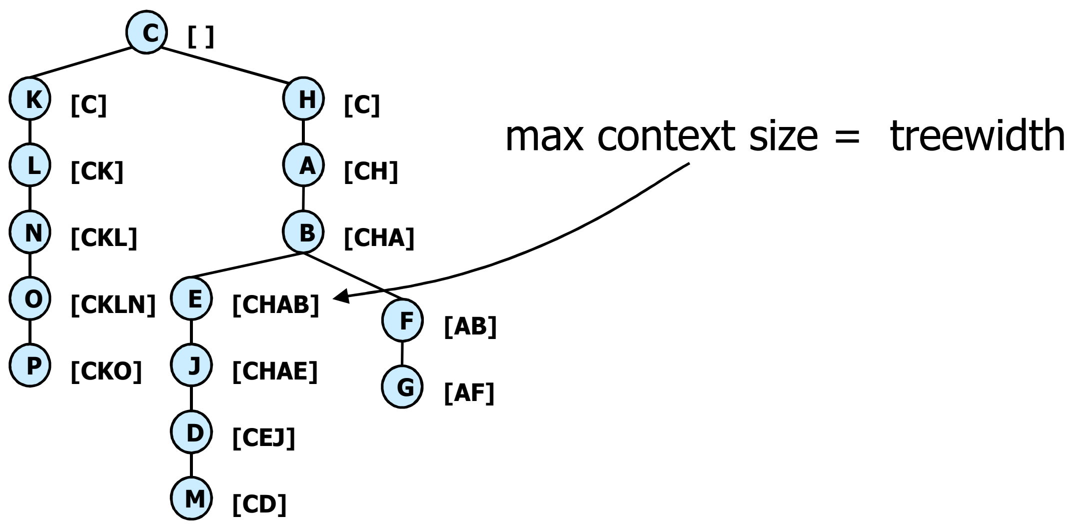
Context minimal AND/OR search graph

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.*





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

Complexity of AND/OR Graph Search

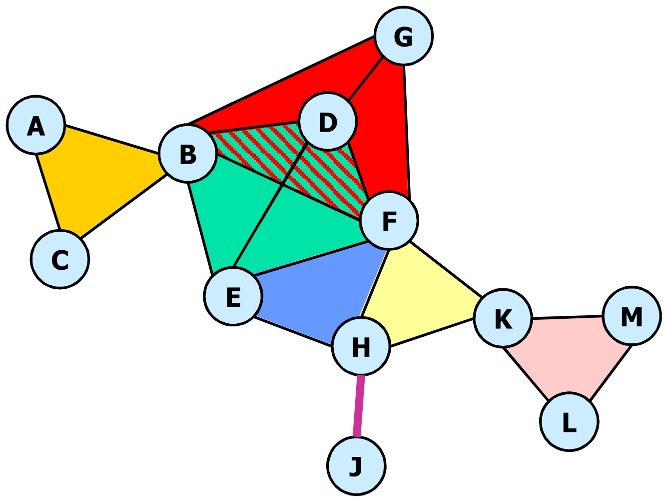
	AND/OR graph	OR graph
Space	$O(n\ k^{w^*})$	$O(n\ k^{pw^*})$
Time	$O(n\ k^{w^*})$	$O(n\ k^{pw^*})$

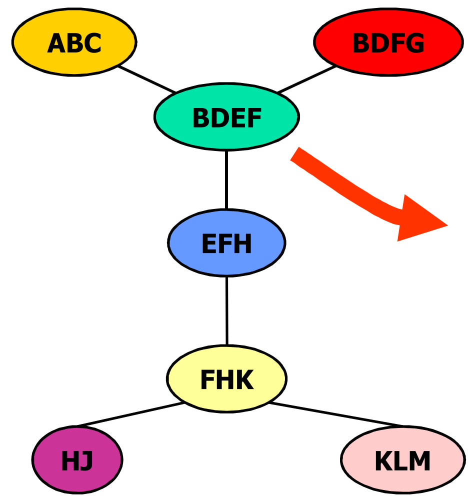
$w^* \leq pw^* \leq w^* \log n$

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

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

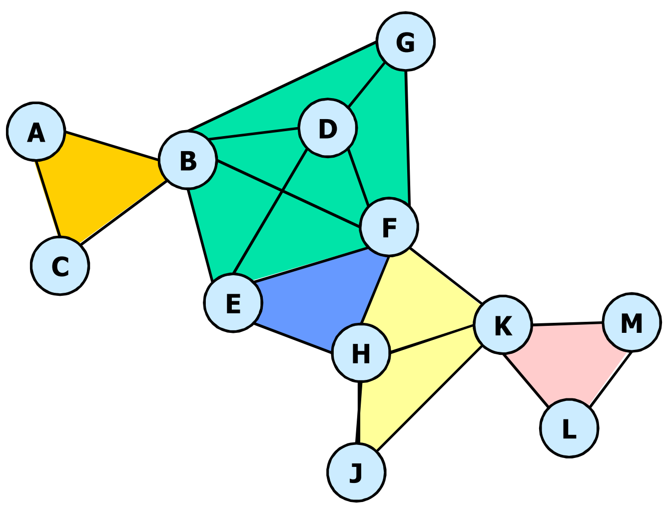
Treewidth vs. Pathwidth

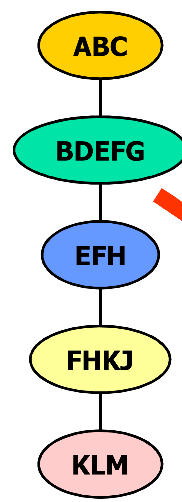




TREE

***treewidth* = 3**
= (max cluster size) - 1

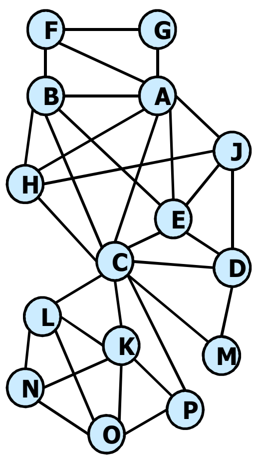


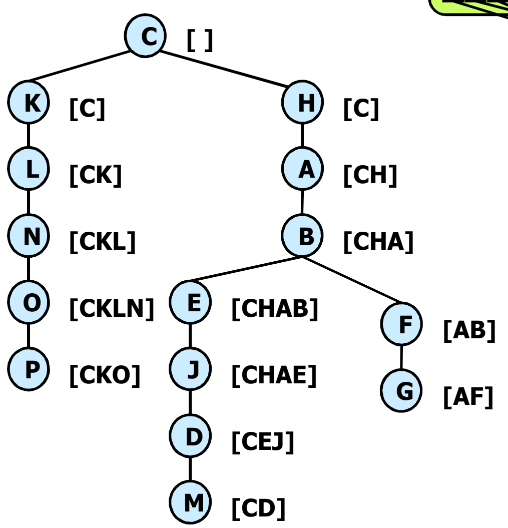


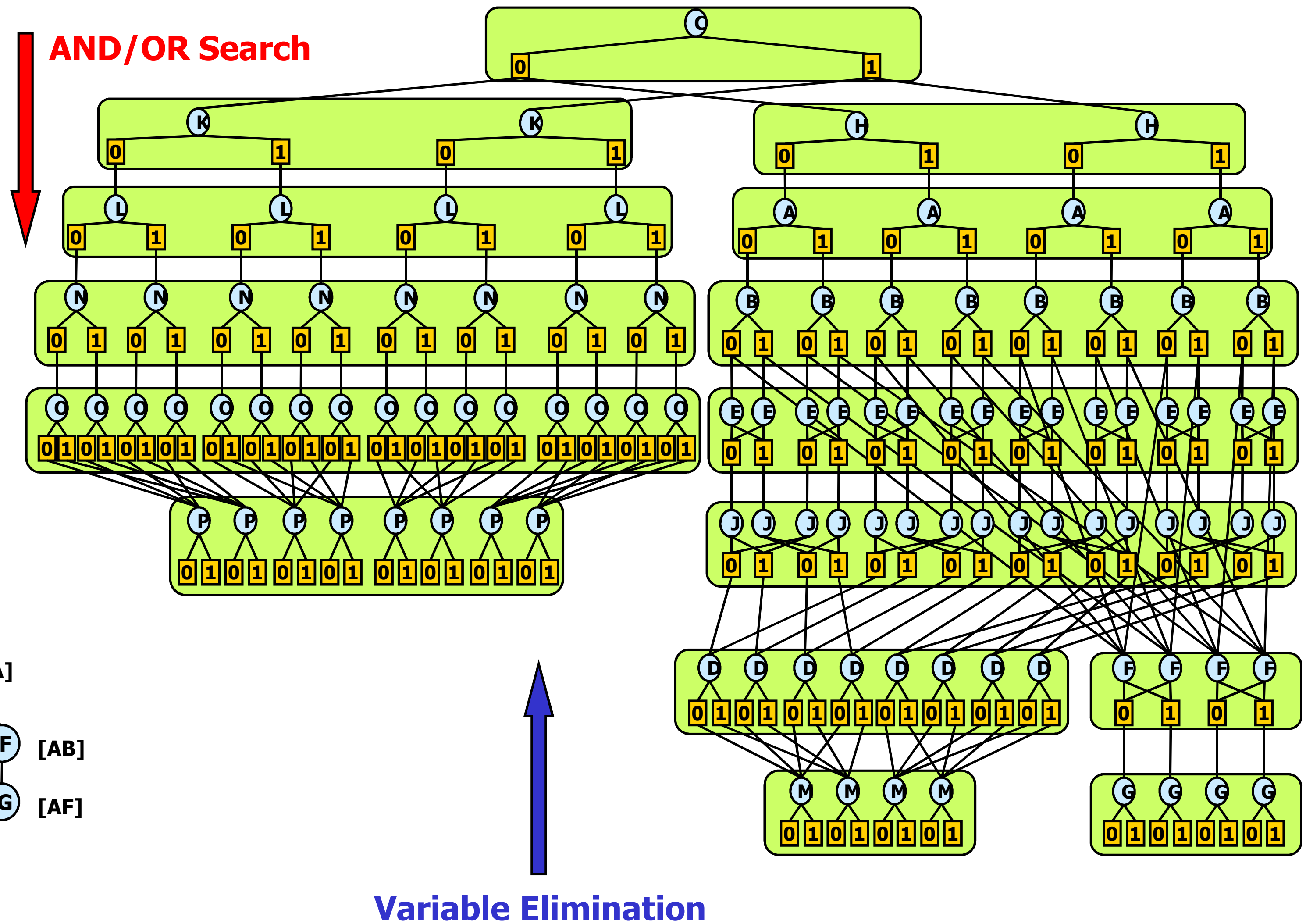
CHAIN

***pathwidth* = 4**
= (max cluster size) - 1

AND/OR Context Minimal Graph







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

Searching AND/OR Graphs

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

i=0 **j** i=w*

Space: O(n)

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

Space: O(exp(j))

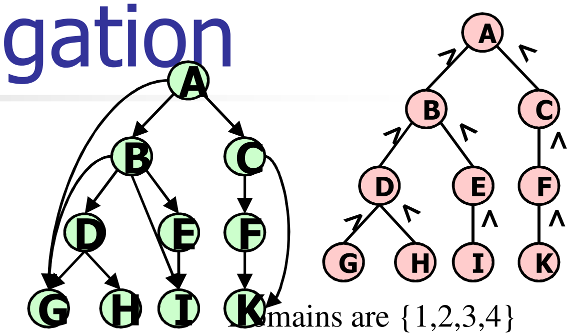
Time: O(exp(m_j+j)

Space: O(exp w*)

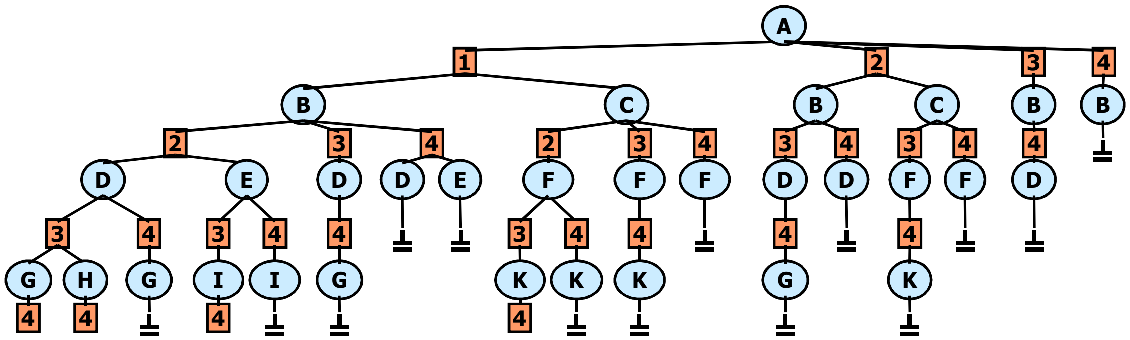
Time: O(exp w*)

For SAT: formula caching?
Clause learning

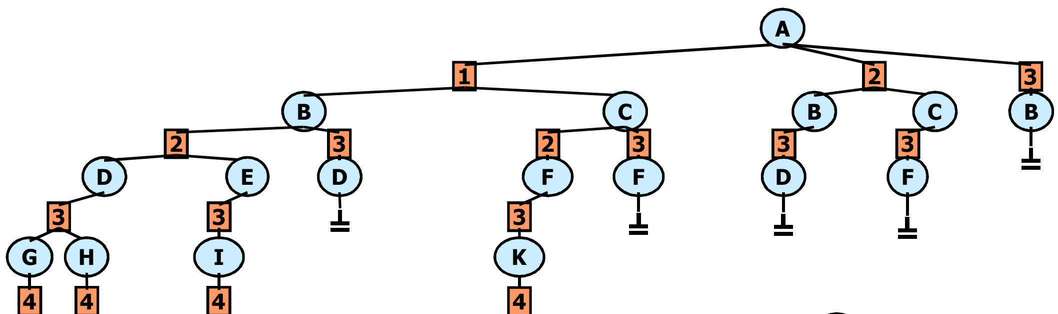
The Effect of Constraint Propagation



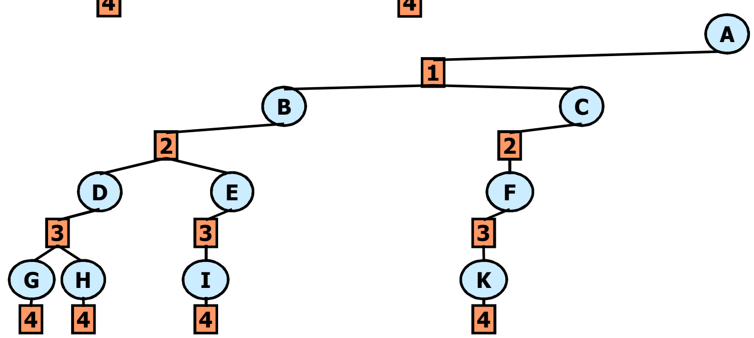
Domains are {1,2,3,4}



CONSTRAINTS ONLY



FORWARD CHECKING



MAINTAINING ARC CONSISTENCY

Overview

- Introduction to graphical models algorithms: Inference, search and hybrids.

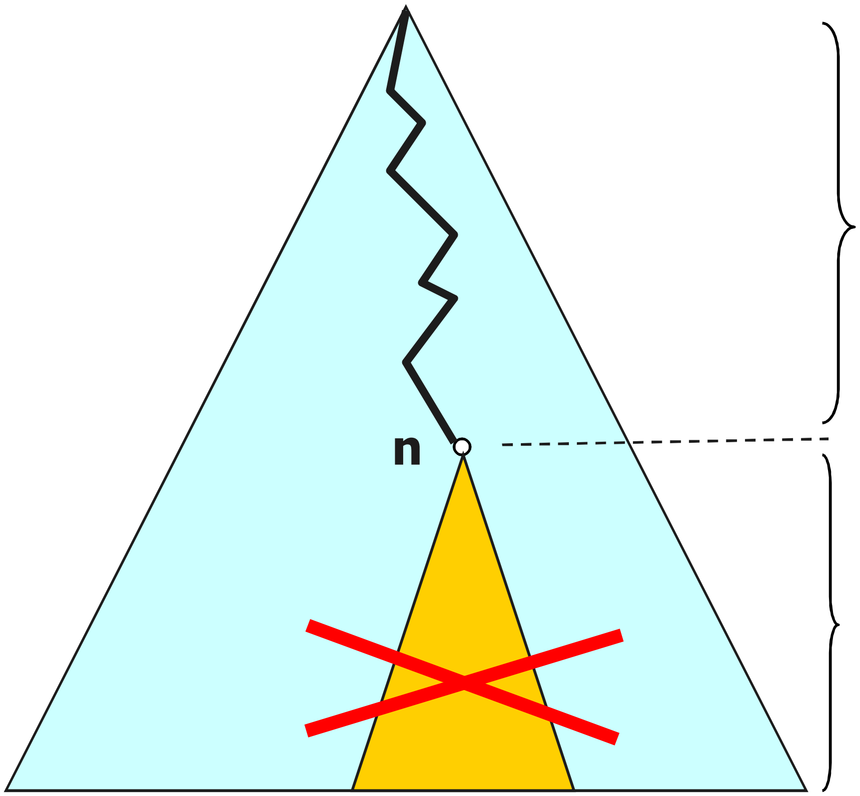
- AND/OR search spaces
  - AND/OR trees
  - AND/OR Graphs

- AND/OR search for combinatorial optimization
  - The mini-bucket heuristic
  - AO depth-first and best-first Branch and Bound
  - Empirical evaluation

- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

AND/OR Branch-and-Bound (AOBB)

(Marinescu & Dechter, IJCAI'05)



Maintain

ub = best solution found so far

g(n)

lb(n) = g(n) + h(n)

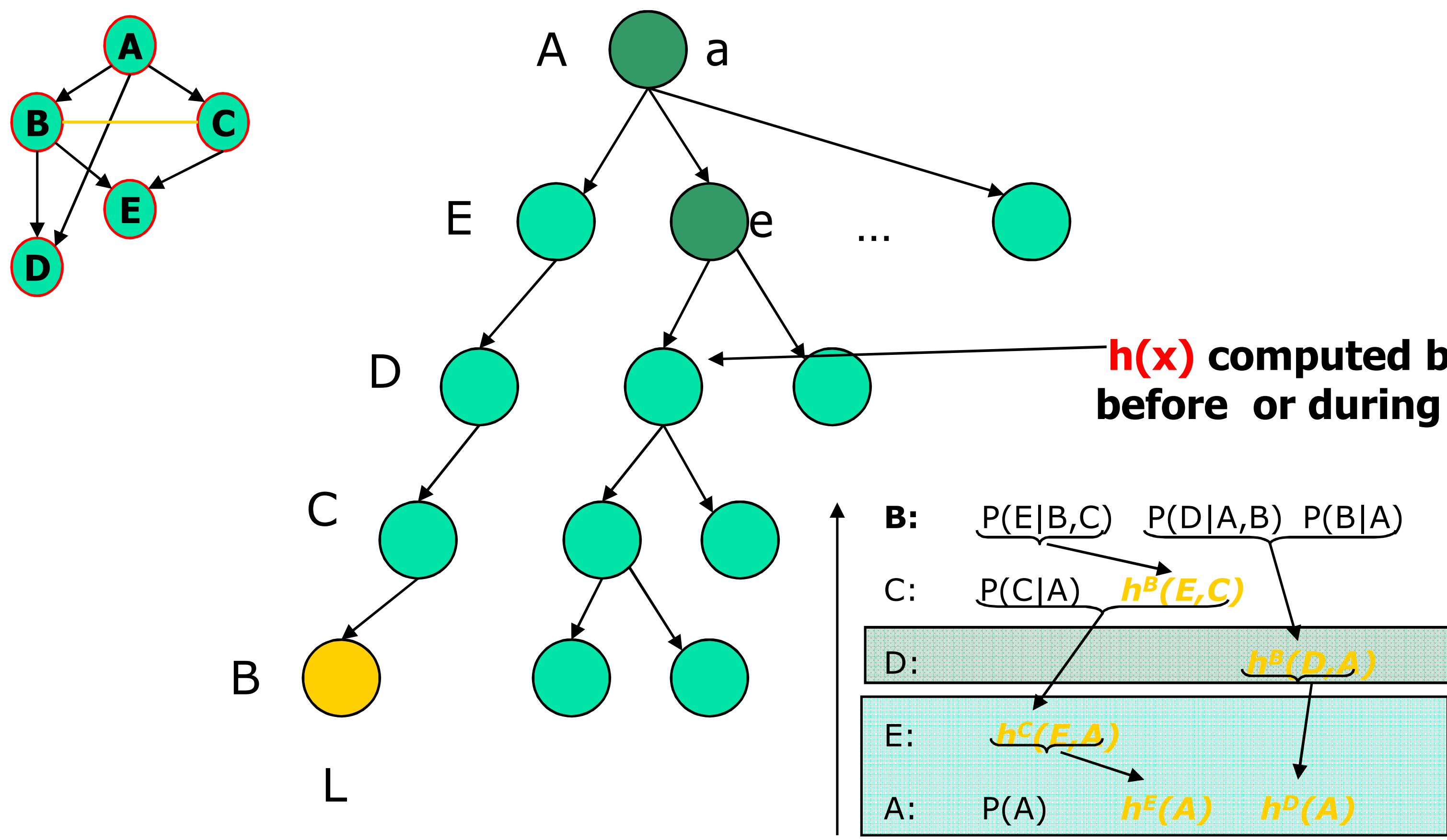
h(n)

estimates the optimal cost below n

OR Branch-and-Bound

Prune subtree below n if lb(n) ≥ ub

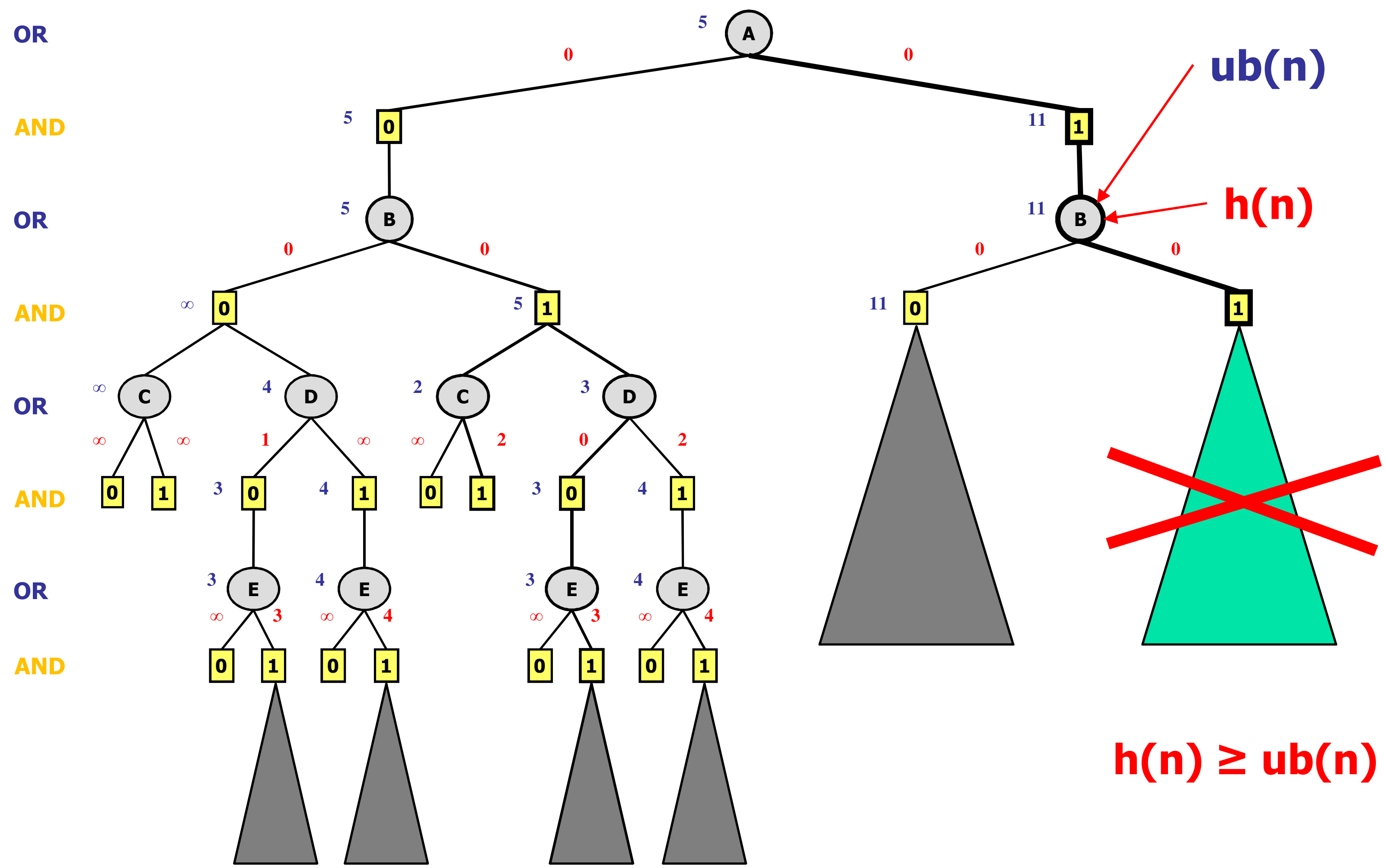
Mini-bucket Heuristics for BB search
(Kask and dechterAIJ, 2001, Kask, Dechter and Marinescu UAI 2003)



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

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

AND/OR Branch-and-Bound (contd.)



AND/OR Branch and Bound for Constraint Optimization
(Marinescu and Dechter, IJCAI 2005, UAI 2005, AAAI 2006, ECAI 2006)

- Search AND/OR Context-minimal graph
  - exploit decomposition and equivalence
- Prune irrelevance via mini-bucket heuristics, and constraint propagation
- Depth-first (AOBB) and best-first (AOBF)
- Dynamic variable orderings
- Applied to MPE and weighted CSPs
- Applied to Integer Programming

Genetic Linkage Analysis

(Marinescu & Dechter, AAAI'07; Marinescu & Dechter, UAI'07)

ped (w*, h)	Samlam	Superlink	BB-C+SMB(i) time	BB-C+SMB(i) nodes	AOBB-C+SMB(i) time	AOBB-C+SMB(i) nodes	AOBF-C+SMB(i) time	AOBF-C+SMB(i) nodes
			i = 10					
ped1 (15, 61)	5.44	54.73	1.14	7,997	0.39	4,576	**0.26**	1,177
ped38 (17, 59)	out	**28.36**	-	-	2046.95	11,868,672	216.94	583,401
ped50 (18, 58)	out	-	-	-	66.66	403,234	**12.75**	25,507
			i=18					
ped18 (21, 119)	157.05	139.06	-	-	23.83	118,869	**19.85**	53,961
ped25 (29, 53)	out	-	-	-	**2041.64**	6,117,320	out	
ped39 (23, 94)	out	322.14	-	-	61.20	313,496	**41.69**	79,356

0-1 Integer Linear Programs
(Marinescu & Dechter, CPAIOR'07)

uwlp50-400	CPLEX		AOBB+PVO		AOBF+PVO	
(w*, h)	time	nodes	time	nodes	time	nodes
uwlp-1 (50, 123)	**10.76**	12	106.63	29	81.63	**8**
uwlp-4 (50, 123)	**6.52**	6	55.10	10	51.85	**3**
uwlp-5 (50, 123)	**30.55**	58	247.03	50	131.58	**8**
uwlp-6 (50, 123)	**3.59**	0	32.31	1	32.65	**1**
uwlp-8 (50, 123)	**3.40**	0	96.66	21	60.27	**3**
uwlp-9 (50, 123)	**9.02**	6	97.00	9	78.05	**2**

Uncapacitated Warehouse Location Problems with 50 stores and 400 locations

MAX-SAT Instances

(Marinescu & Dechter, CPAIOR'07)

pret (w*, h)	CPLEX time	CPLEX nodes	AOBB-C time	AOBB-C nodes	AOBF-C time	AOBF-C nodes
pret60-40 (6, 13)	676.94	3,926,422	7.38	1,216	**3.58**	**568**
pret60-60 (6, 13)	535.05	2,963,435	7.30	1,140	**3.56**	**538**
pret60-75 (6, 13)	402.53	2,005,738	6.34	1,067	**3.08**	**506**
pret150-40 (6, 15)	out		75.19	5,625	**19.70**	**1,379**
pret150-60 (6, 15)	out		78.25	5,813	**19.75**	**1,393**
pret150-75 (6, 15)	out		84.97	6,144	**20.95**	**1,430**

pret MAX-SAT instances solved as 0-1 ILPs

Genetic Linkage Analysis

(Marinescu & Dechter, AAAI'07; Marinescu & Dechter, UAI'07)

pedigree (w*, h) (n, d)	Samlam v. 2.3.2	Superlink v. 1.6	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=6 time	nodes	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=8 time	nodes	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=10 time	nodes	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=12 time	nodes	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=14 time	nodes
			0.05		0.05		0.11		0.31		0.97	
ped1			-	-	-	-	1.14	7,997	0.73	3,911	1.31	2,704
(15, 61)	5.44	54.73	24.30	416,326	13.17	206,439	1.58	24,361	1.84	25,674	1.89	15,156
(299, 5)			4.19	69,751	2.17	33,908	0.39	4,576	0.65	6,306	1.36	4,494
			1.30	7,314	2.17	13,784	**0.26**	1,177	0.87	4,016	1.54	3,119
			0.12		0.45		5.38		60.97		out	
ped38			-	-	-	-	-	-	-	-		
(17, 59)	out	**28.36**	-	-	8120.58	85,367,022	-	-	3040.60	35,394,461		
(582, 5)			5946.44	34,828,046	1554.65	8,986,648	2046.95	11,868,672	272.69	1,412,976		
			out		134.41	348,723	216.94	583,401	103.17	242,429		
			0.11		0.74		5.38		37.19		out	
ped50			-	-	-	-	-	-	-	-		
(18, 58)	out	-	-	-	-	-	476.77	5,566,578	104.00	748,792		
(479, 5)			4140.29	28,201,843	2493.75	15,729,294	66.66	403,234	52.11	110,302		
			78.53	204,886	36.03	104,289	**12.75**	25,507	38.52	5,766		

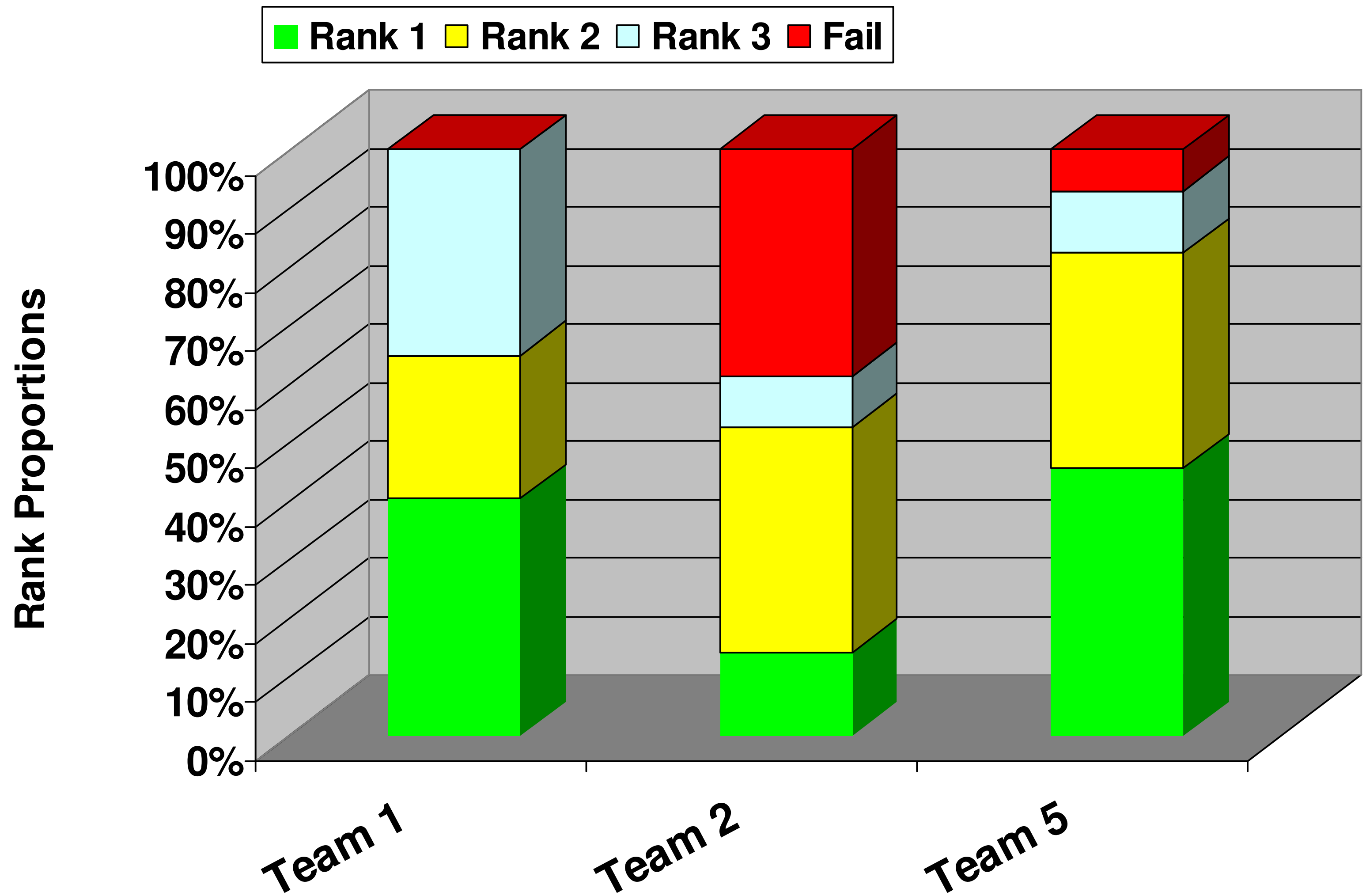
Genetic Linkage Analysis

(Marinescu & Dechter, AAAI'07; Marinescu & Dechter, UAI'07)

pedigree (w*, h) (n, d)	**Samlam** v. 2.3.2	**Superlink** v. 1.6	**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=12 time	 nodes	**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=14 time	 nodes	**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=16 time	 nodes	**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=18 time	 nodes	**MBE(i)** **BB-C+SMB(i)** **AOBB+SMB(i)** **AOBB-C+SMB(i)** **AOBF-C+SMB(i)** i=20 time	 nodes
			0.31		0.81		2.66		8.41		24.88	
ped30			-	-	-	-	-	-	-	-	-	-
(25, 51)	out	13095.83	5563.22	63,068,960	1397.14	15,336,772	1811.34	20,275,620	550.57	5,535,261	82.25	588,558
(1016, 5)			1440.26	11,694,534	597.88	5,580,555	1023.90	10,458,174	151.96	1,179,236	43.83	146,896
			186.77	692,870	58.38	253,465	85.53	350,497	49.38	179,790	**33.03**	37,705
			0.41		1.08		5.28		14.45		51.24	
ped33			-	-	-	-	-	-	-	-	-	-
(26, 48)	out	-	2335.28	32,444,818	806.12	11,403,812	62.91	807,071	67.92	701,030	76.47	320,279
(581, 5)			886.05	8,426,659	370.41	4,032,864	26.31	229,856	33.11	219,047	54.89	83,360
			out		194.78	975,617	**24.16**	102,888	32.55	101,862	58.52	57,593
			0.52		2.32		8.41		33.15		81.27	
ped39			-	-	-	-	-	-	-	-	-	-
(23, 94)	out	322.14	-	-	-	-	4041.56	52,804,044	386.13	2,171,470	141.23	407,280
(1272, 5)			-	-	-	-	968.03	7,880,928	61.20	313,496	93.19	83,714
			out		out		68.52	218,925	**41.69**	79,356	87.63	14,479
			4.20		31.33		96.28		out		out	
ped42			-	-	-	-	-	-				
(25, 76)	out	561.31	-	-	-	-	-	-				
(448, 5)			-	-	-	-	2364.67	22,595,247				
			out		out		**133.19**	93,831				

UAI'06 Results

Rank Proportions (how often was each team a particular rank, rank 1 is best)



Overview

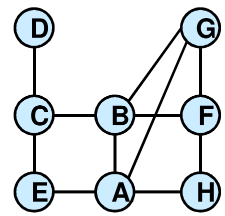
- Introduction to graphical models algorithms: Inference, search and hybrids.

- AND/OR search spaces
  - AND/OR trees
  - AND/OR Graphs

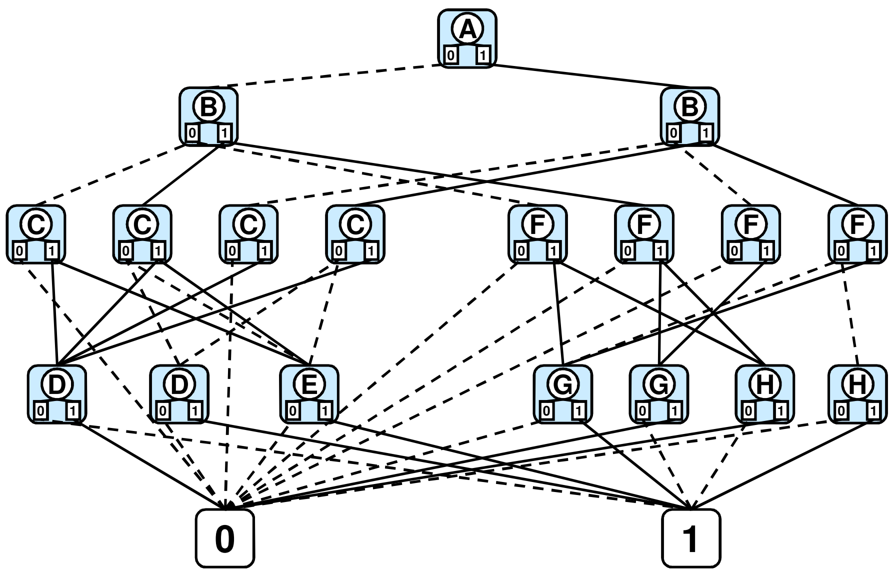
- AND/OR search for combinatorial optimization
  - The mini-bucket heuristic
  - AO depth-first and best-first Branch and Bound
  - Empirical evaluation

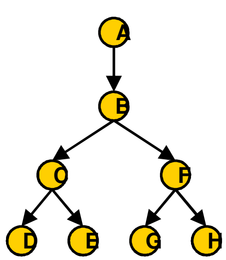
- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

AOBDD vs. OBDD



primal graph

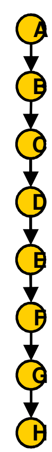


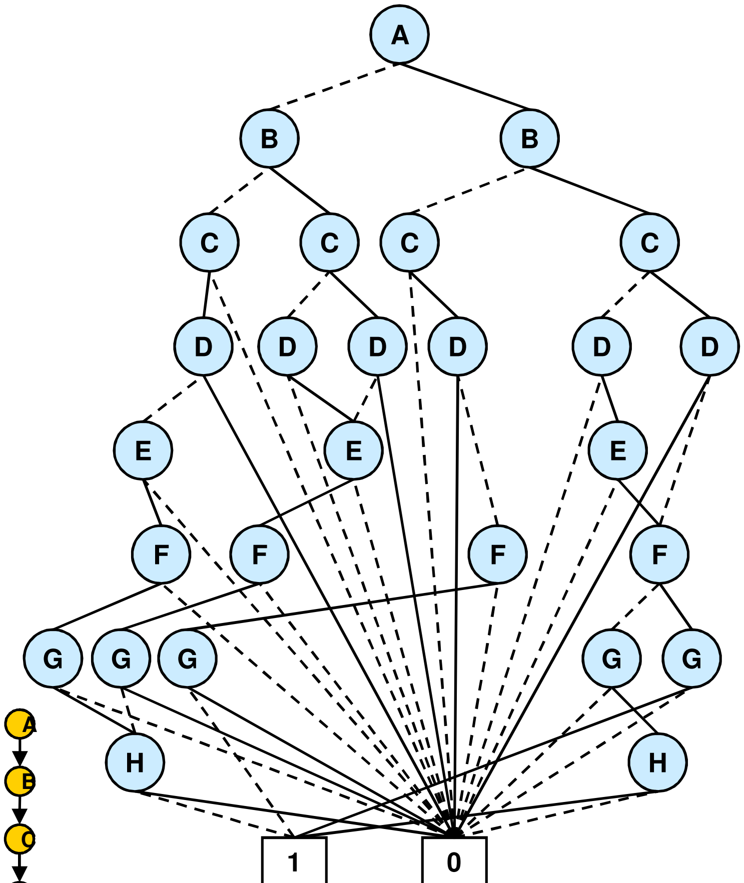


AOBDD

18 nonterminals

47 arcs





OBDD

27 nonterminals

54 arcs

Recent work

- **Radu Marinescu:** Constraint optimization
  - AND/OR Branch and Bound with mini-bucket heuristics (IJCAI 2005, UAI 2005, AAAI 2006, ECAI 2006)
  - AND/OR branch and bound for integer programming (CPAIOR 2006)
  - AO* for constraint optimization
  - AO Best first (UAI 2007, AAAI 2007, CPAIOR 2007)

- **Robert Mateescu (Phd 2007):** Time-Space tradeoff schemes
  - AND/OR for mixed networks (UAI 2004)
  - AND/OR for counting (CP 2004)
  - AND/OR cutset decomposition (IJCAI 2005)
  - Bucket-elimination vs AND/OR search (UAI 2005, IJCAI 2007)
  - AND/OR compilations schemes (AOBDDs) (CP2006)
  - AND/OR compilation for weighted models and optimization (UAI, 2007, CP 2007)

- **Vibhav Gogate:** Sampling schemes for mixed networks
  - (UAI2005, IJCAI05, CP2006)
  - SampleSearch scheme, for inference and lowerbounding (AISTAT 2007, UAI 2007, AAAI 2007)

- **Boznea Bidyuk (Phd, 2006):** w-cutset sampling, w-cutset bounding
  - (UAI 2003, UAI 2004, AAAI 2006, UAI 2006, ECAI 2006)

Software

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

are available at:

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

Conclusion

- **AND/OR search spaces are a unifying framework for search or compilation applicable to any graphical models.**
- **With caching AND/OR is similar to inference (context-minimal graphs)**
- **AND/OR time and space bounds are equal to state of the art algorithms**
- **Empirical results**
  - **AND/OR search spaces are always more effective than traditional OR spaces**
  - **AND/OR allows a flexible tradeoff between space and time**
- **Graphical models should always use AND/OR search with embeded inference.**
- **Current work: Hybrid of inference and search: Heuristic generation and Branch and Bound, AO cycle-cutset**