AND/OR Search for Probabilistic and Deterministic Graphical Models

Rina Dechter
Bren school of ICS
University of California, Irvine

Students:

Radu Marinescu,
Robert Mateescu
Vibhav Gogate

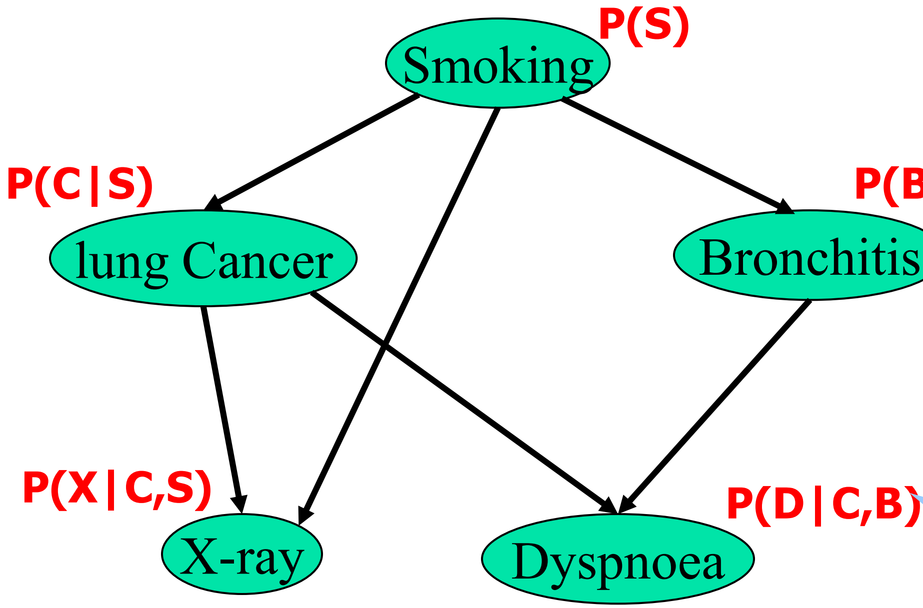
Overview

- 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

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

Bayesian Networks
(Pearl, 1988)



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

CPD:

C	B	P(D|C,B)	
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

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

Belief Updating, Most probable tuple (MPE)

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

Constraint Networks

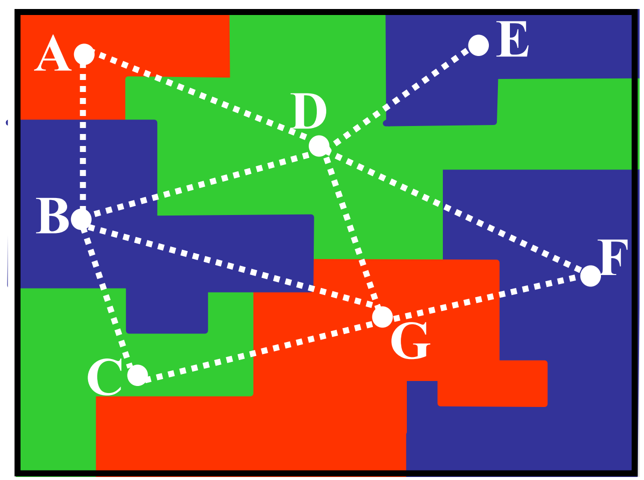
Example: map coloring

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

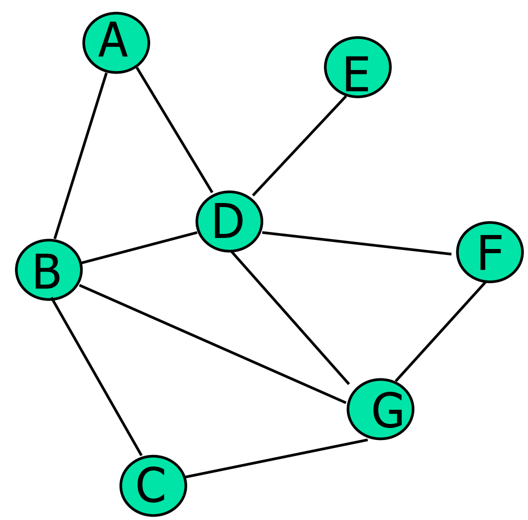
Values - colors (red, green, blue)

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

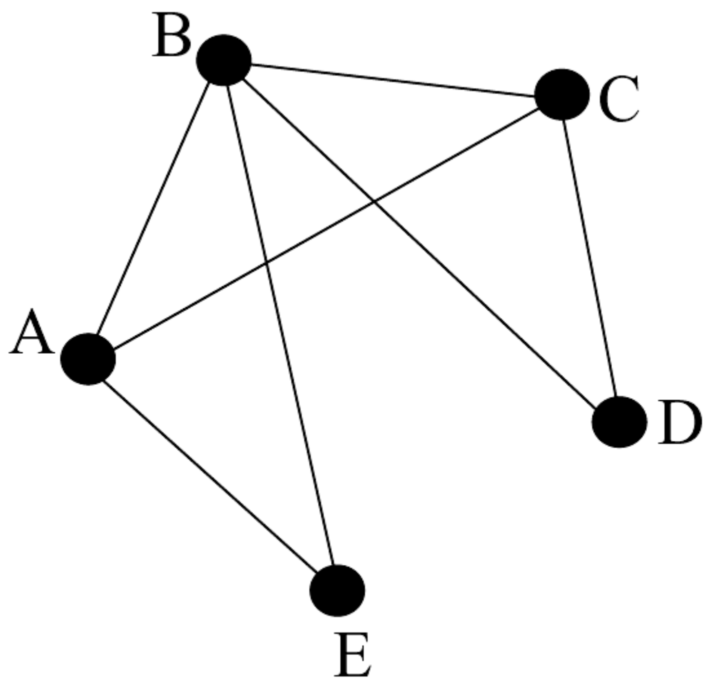


Semantics: set of all solutions

Primary task: find a solution

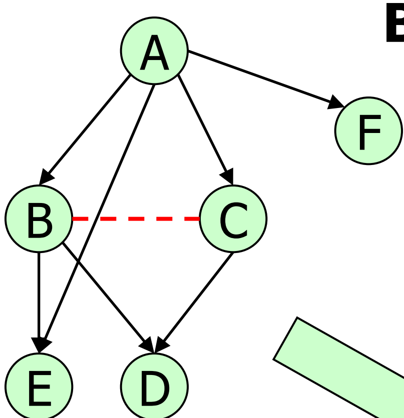
Propositional Satisfiability

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



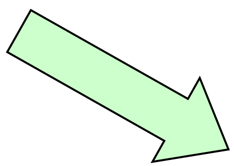
Mixed Networks
(Mateescu and Dechter, 2004)

Belief Network

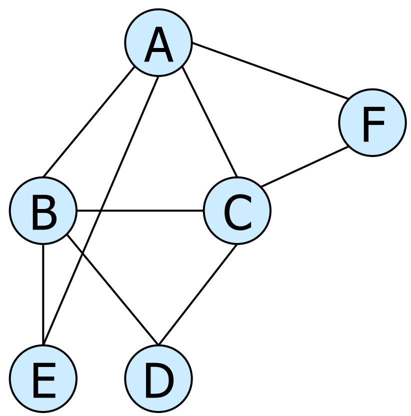


$P(D \mid B, C)$

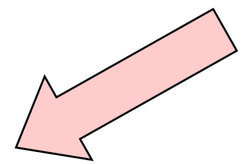
B	C	D=0	D=1
0	**0**	.2	.8
0	**1**	.1	.9
1	**0**	.3	.7
1	**1**	.5	.5

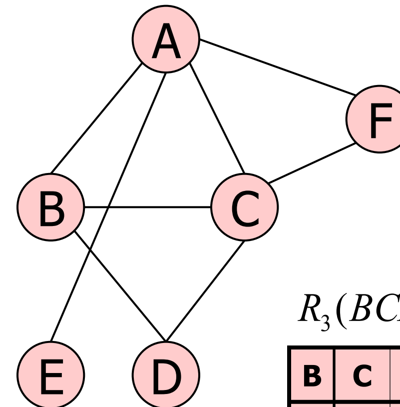


Moral mixed graph



Constraint Network





$R_3(BCD)$

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

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

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

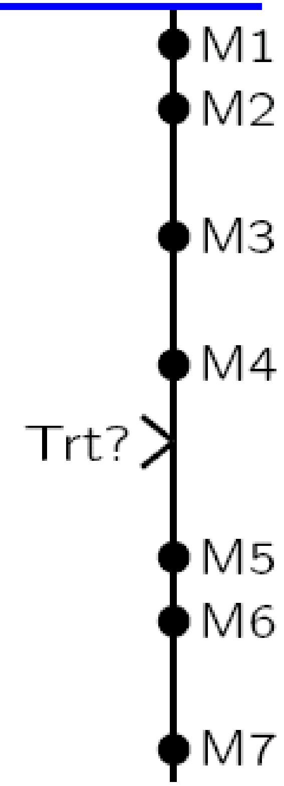
Linkage analysis with pedigree data

GIVEN:

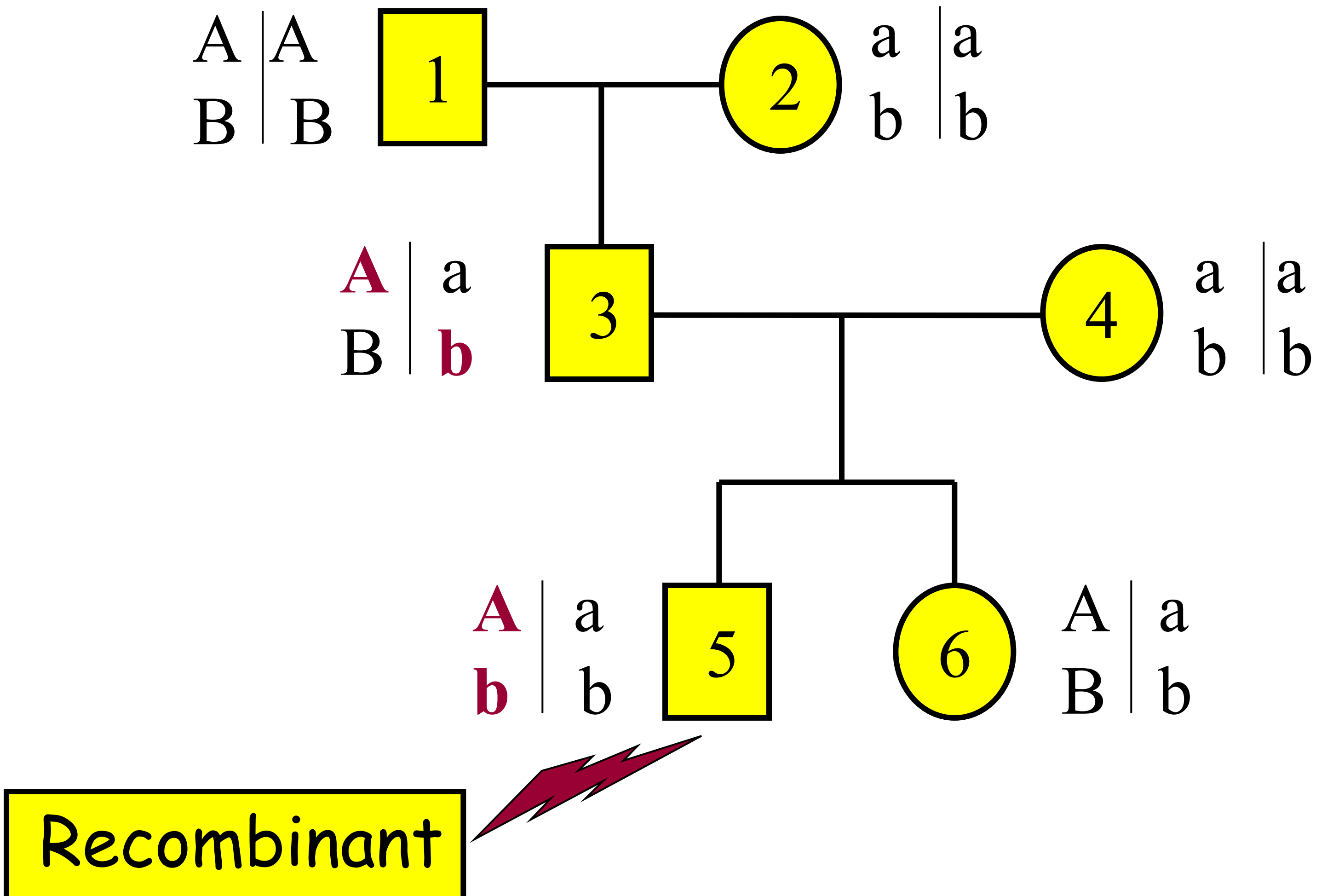
- A set of pedigrees, and some trait of interest.
- A set of DNA markers, with known genetic model (genetic map, and allele frequencies).
- Data on trait(s) and at markers, for some subset of the individuals.

QUESTION: Testing and estimation.

- Does any DNA on the chromosome of the markers affect the trait? H_0 : No.
- If so, what is the likely location of this DNA, relative to markers.



Two Loci Inheritance



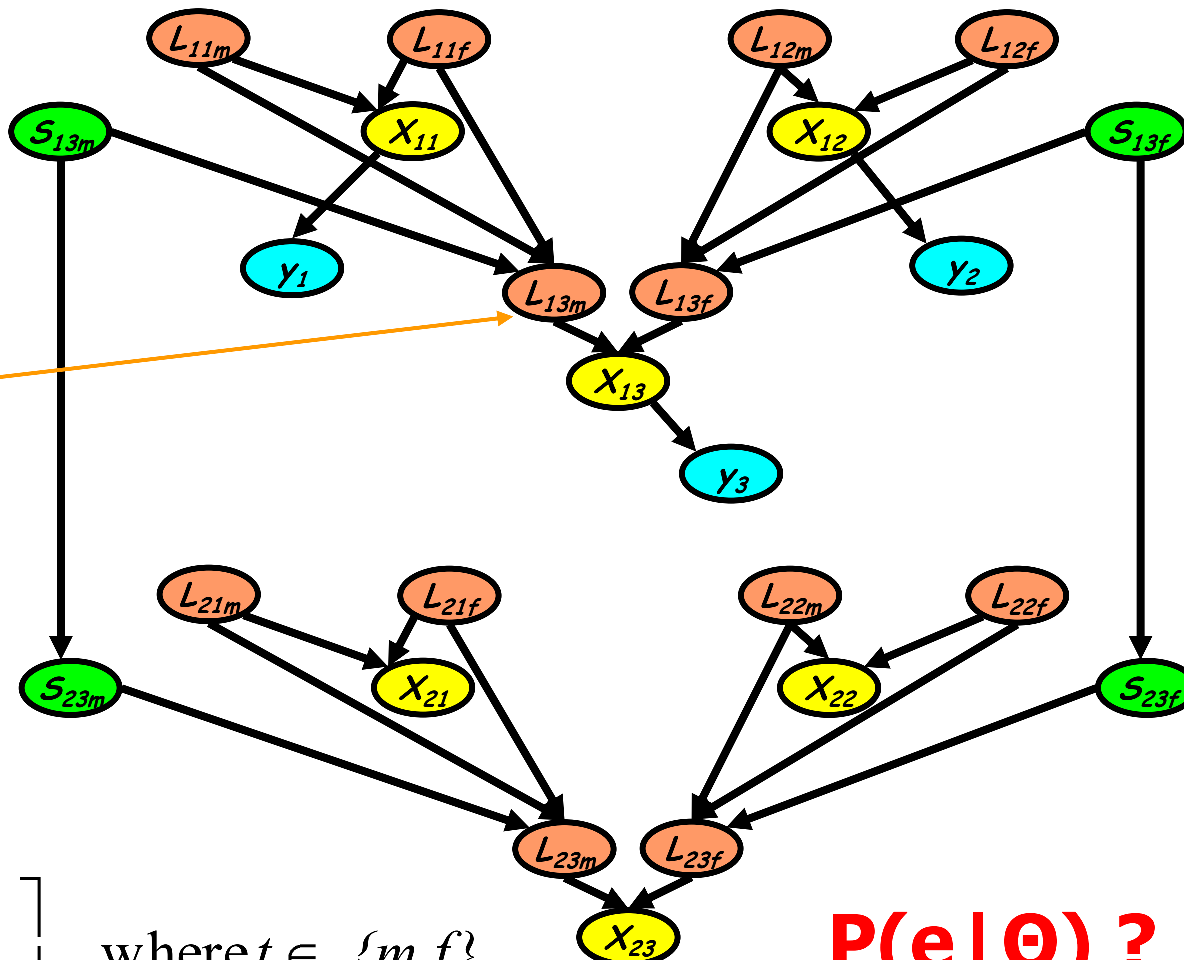
Bayesian Network for Recombination

Locus 1

Deterministic relationships

Probabilistic relationships

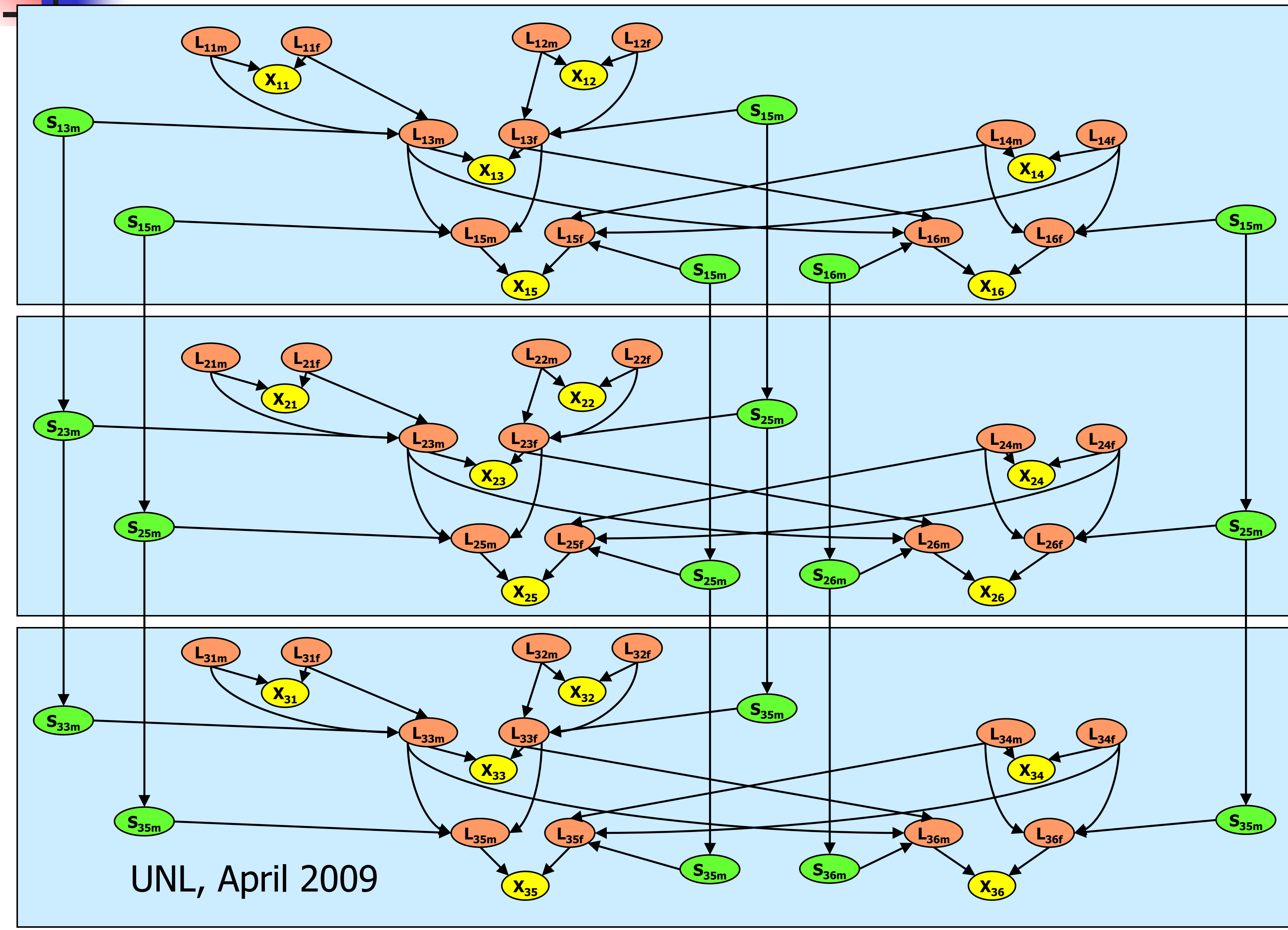
Locus 2



$$P(s_{23t} \mid s_{13t}, \theta) = \begin{bmatrix} 1-\theta & \theta \\ \theta & 1-\theta \end{bmatrix} \quad \text{where } t \in \{m,f\}$$

P(e|Θ) ?

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 …)

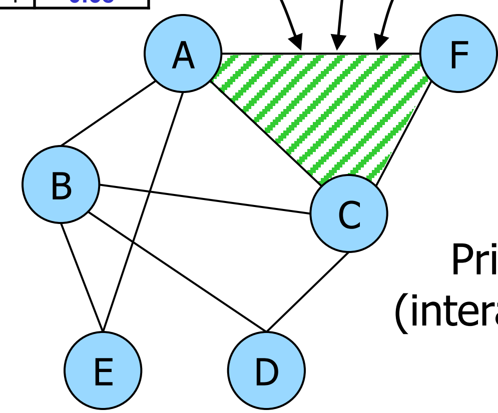
Conditional Probability Table (CPT)

A	C	F	P(F|A,C)
0	0	0	0.14
0	0	1	0.96
0	1	0	0.40
0	1	1	0.60
1	0	0	0.35
1	0	1	0.65
1	1	0	0.72
1	1	1	0.68

Relation

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

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



Primal graph (interaction graph)

- Operators:
  - combination
  - elimination (projection)

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

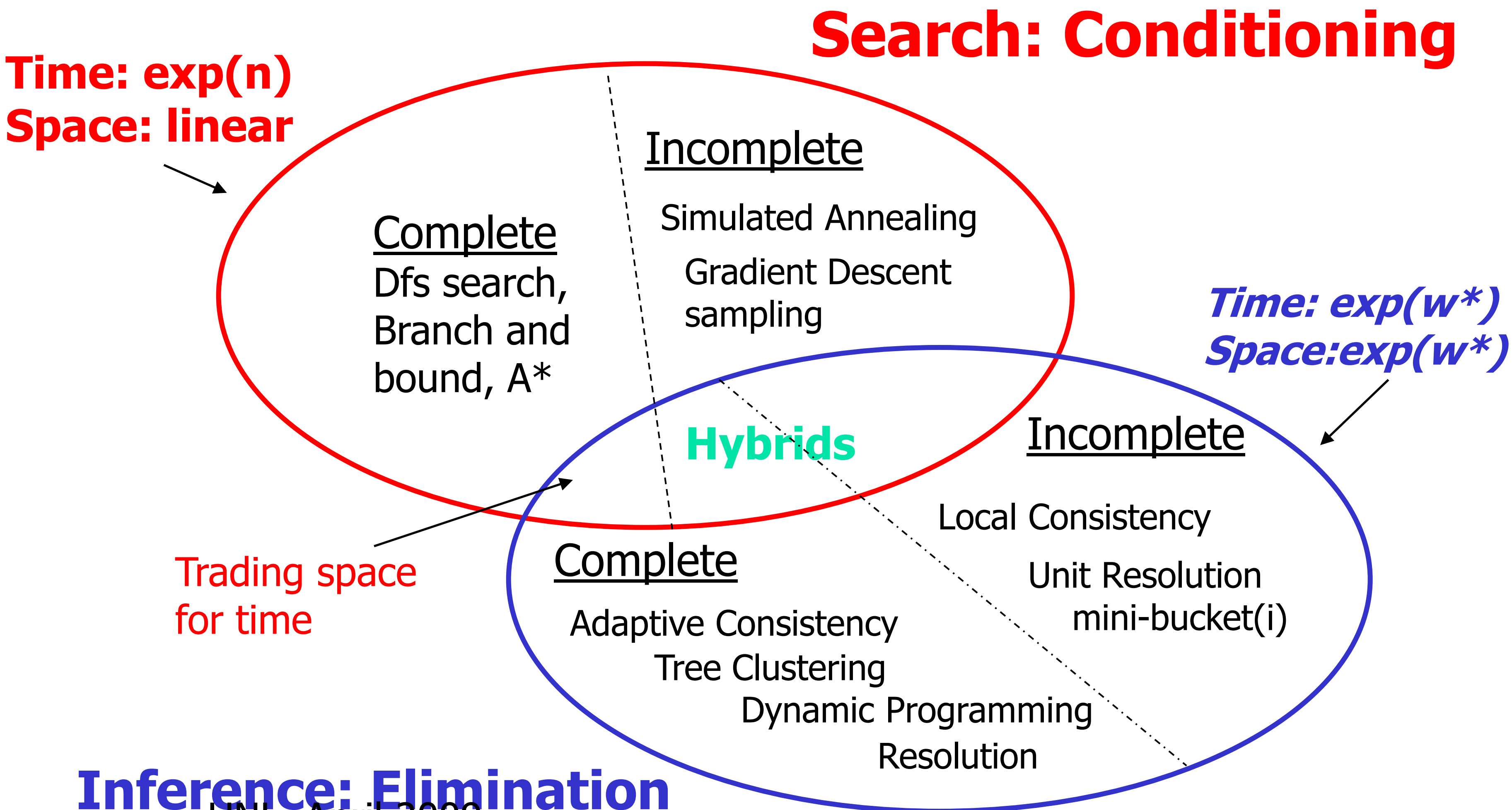
- All these tasks are NP-hard
  - exploit problem structure
  - identify special cases
  - approximate

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.

Solution Techniques

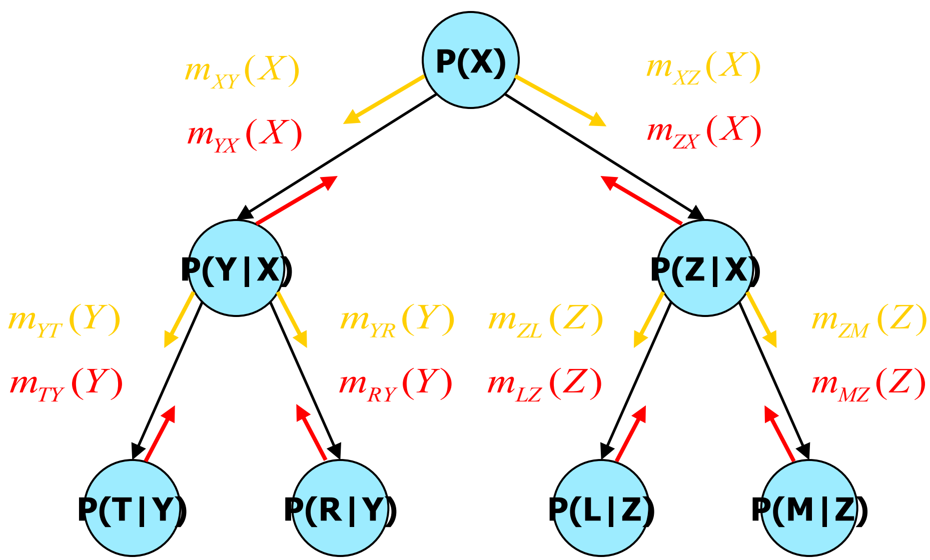
All queries are NP-hard so: exploit structure, identify tractable classes, approximate



Tree-solving is Easy

Belief updating (sum-prod)

CSP – consistency (projection-join)



Dynamic Programming

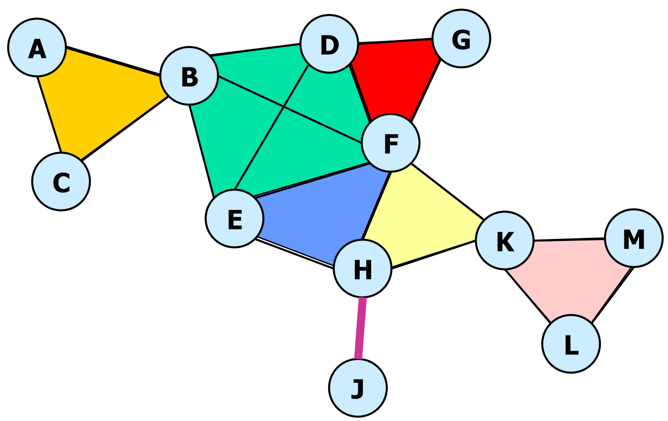
MPE (max-prod)

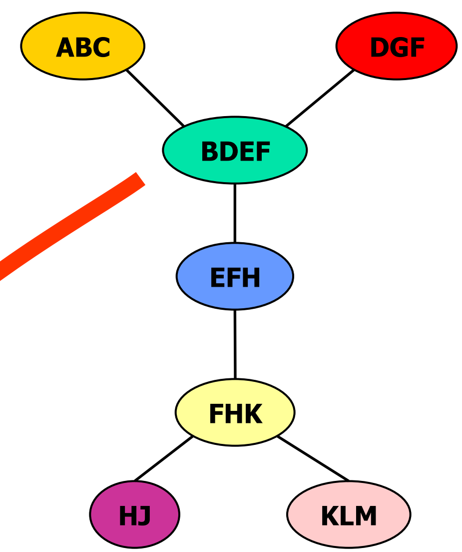
#CSP (sum-prod)

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)**

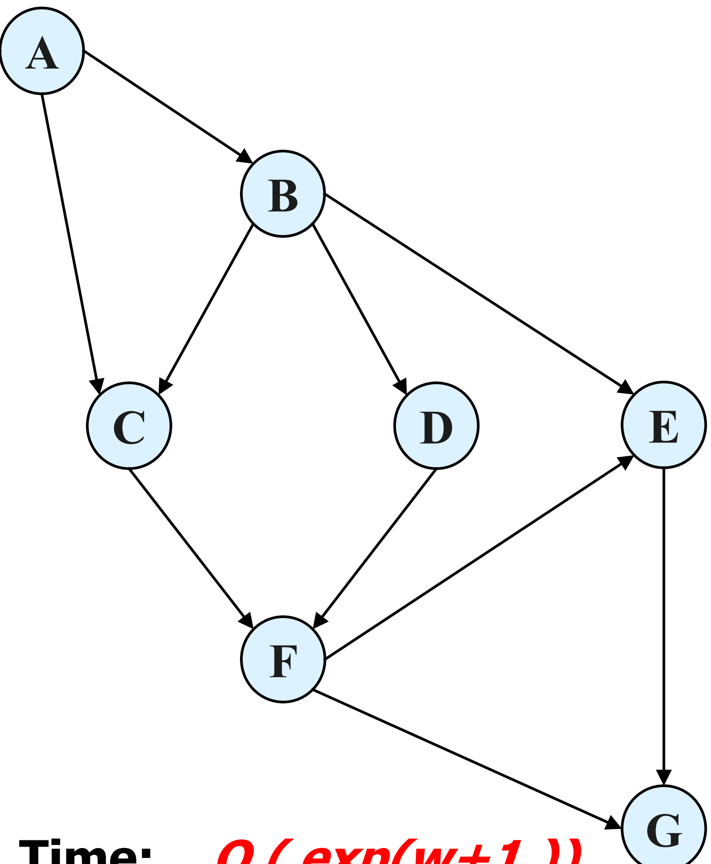
***treewidth* = 4 - 1 = 3**

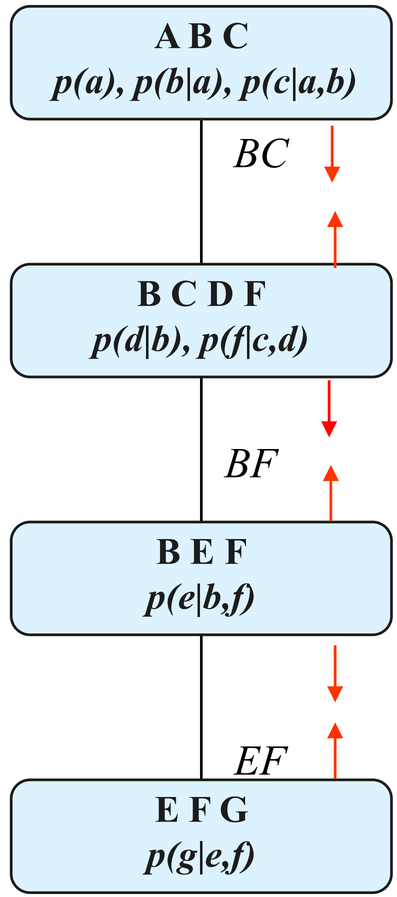
treewidth = (maximum cluster size) – 1

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

Bucket Elimination

Algorithm *elim-bel* (Dechter 1996, Zhang and Poole, 1995)

$\underbrace{\sum_b \prod}$ ← Elimination operator

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

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

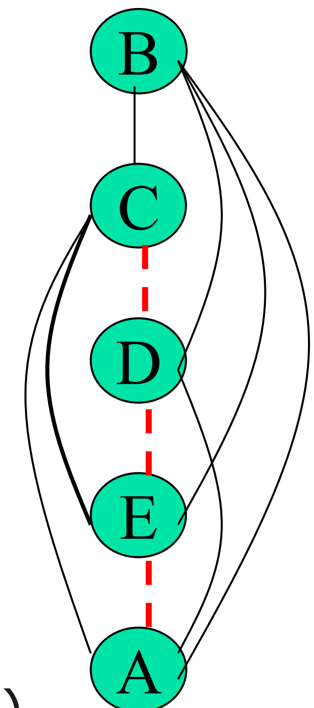
bucket D: $h^C(a,d,e)$

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

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

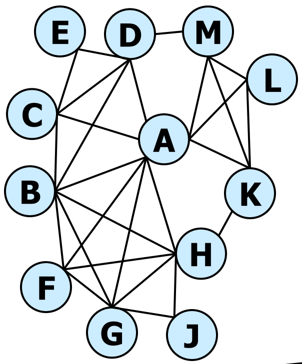
P(a|e=0)

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

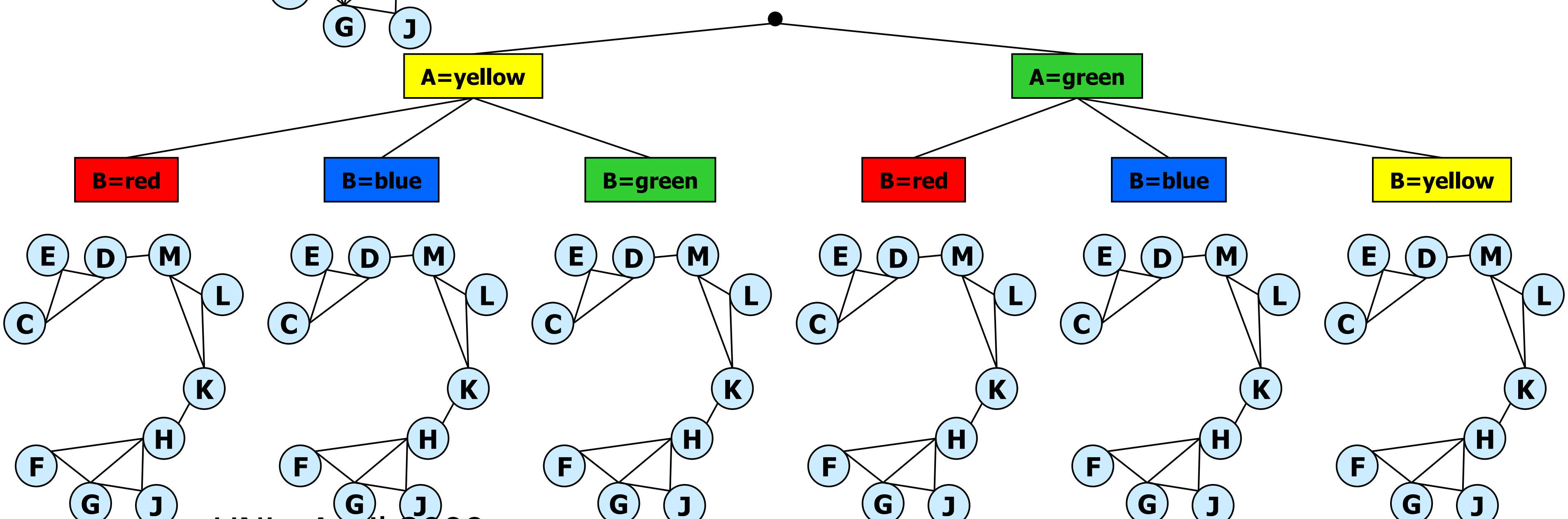


Search over the Cutset (cont)

Graph Coloring problem



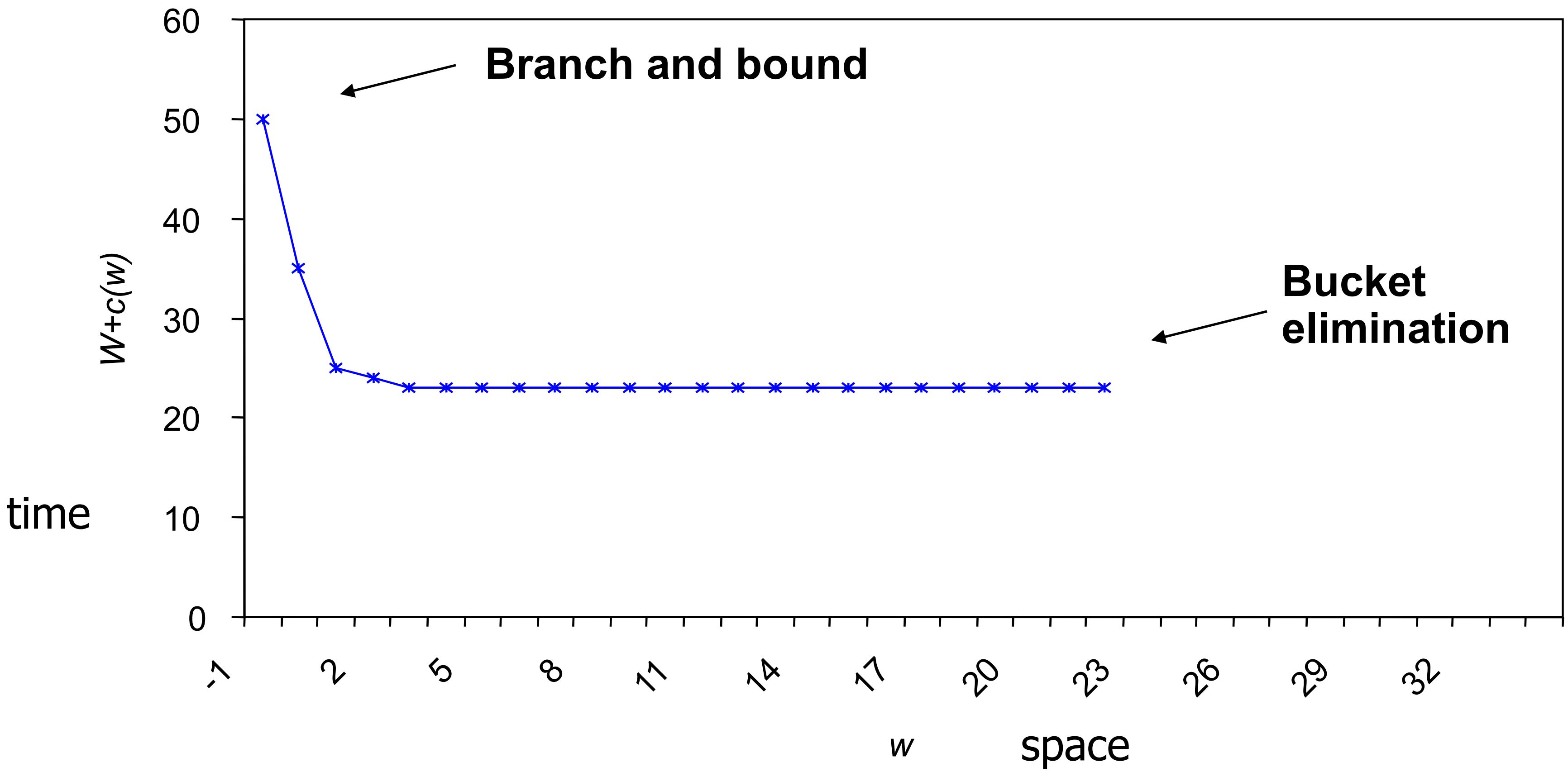
- Inference may require too much memory
- **Condition** on some of the variables



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**≈23)**



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

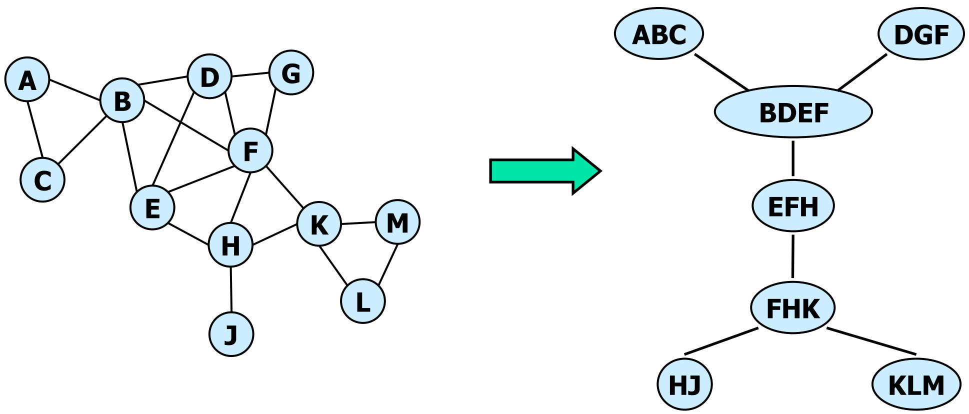
- Bounding search:
  - Sampling

- Goal: an anytime scheme

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

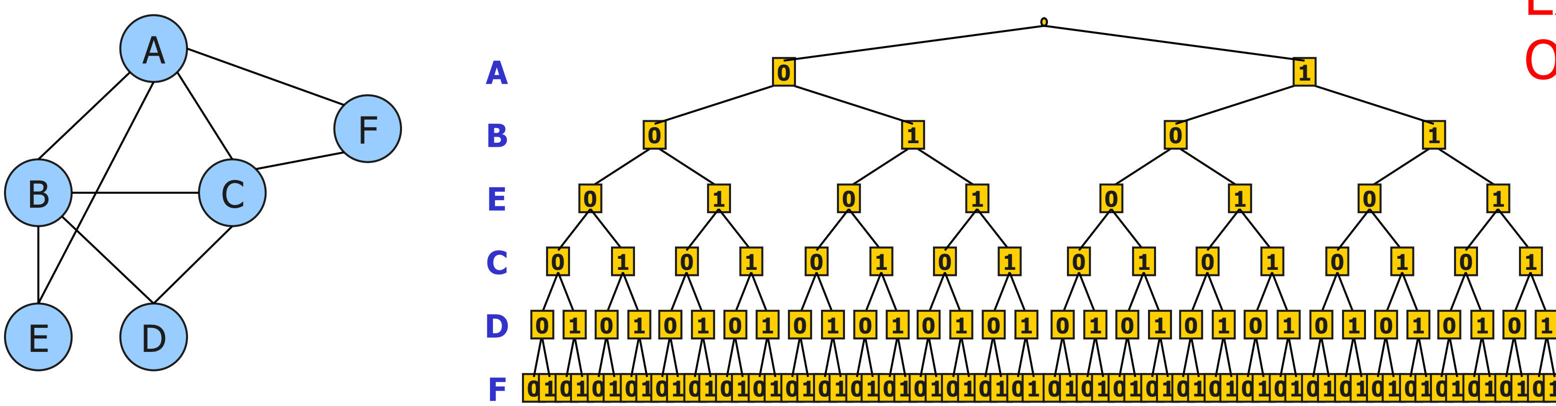
Inference vs Search



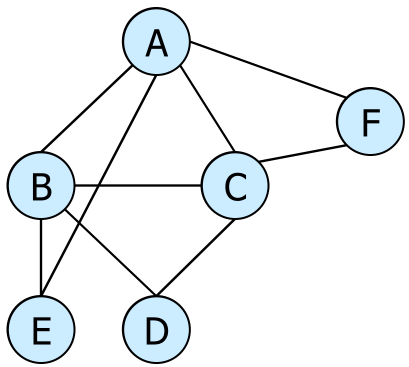
Inference
- •decomposition
- •Redundancy

Exp(w*) time/space

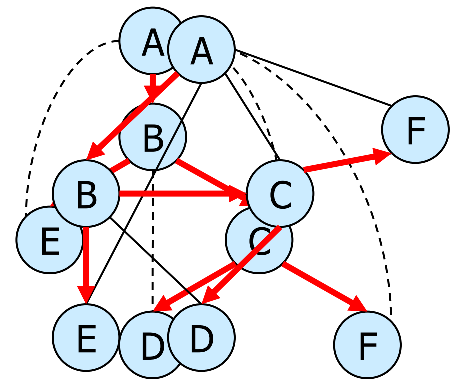
Search
- •Pruning → Exp(n) time O(n) space



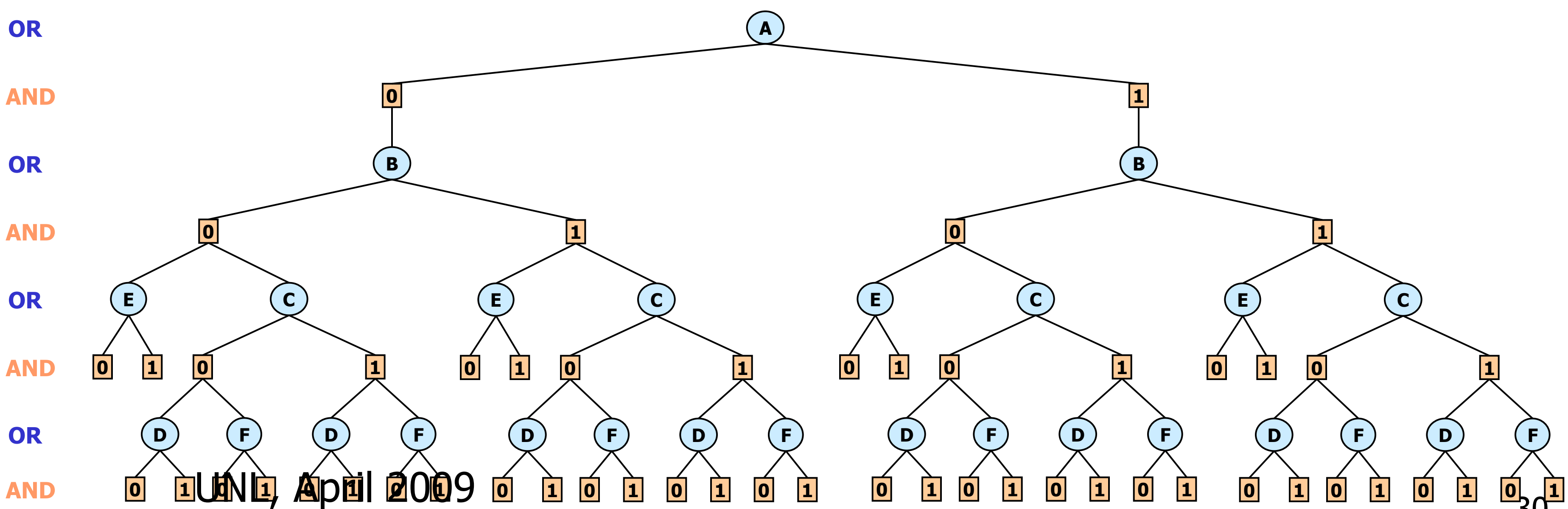
AND/OR Search Space



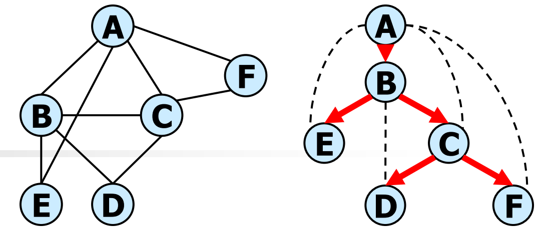
Primal graph

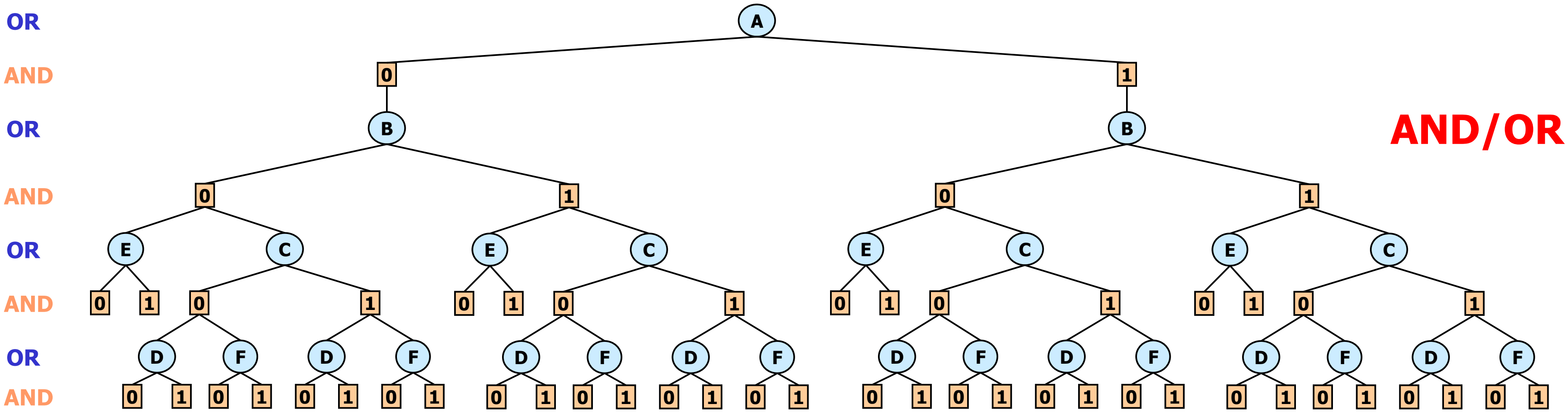


DFS tree

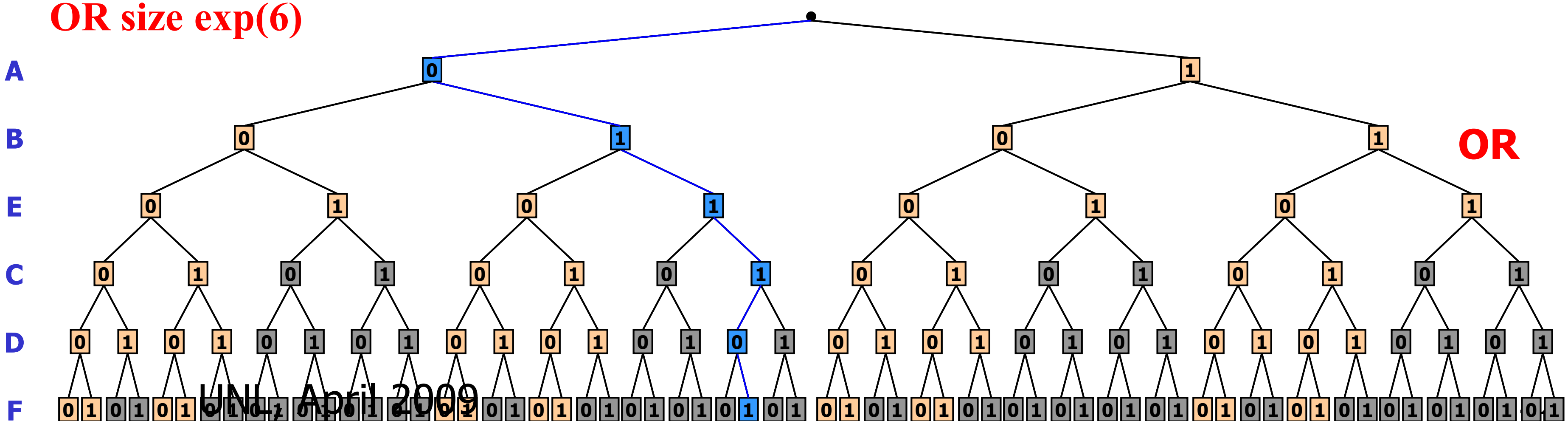


AND/OR vs. OR



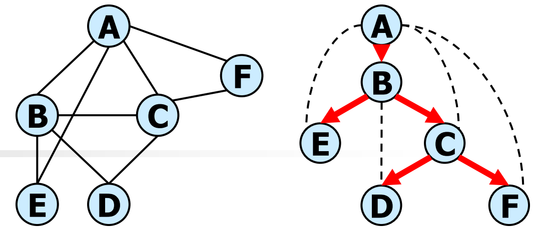


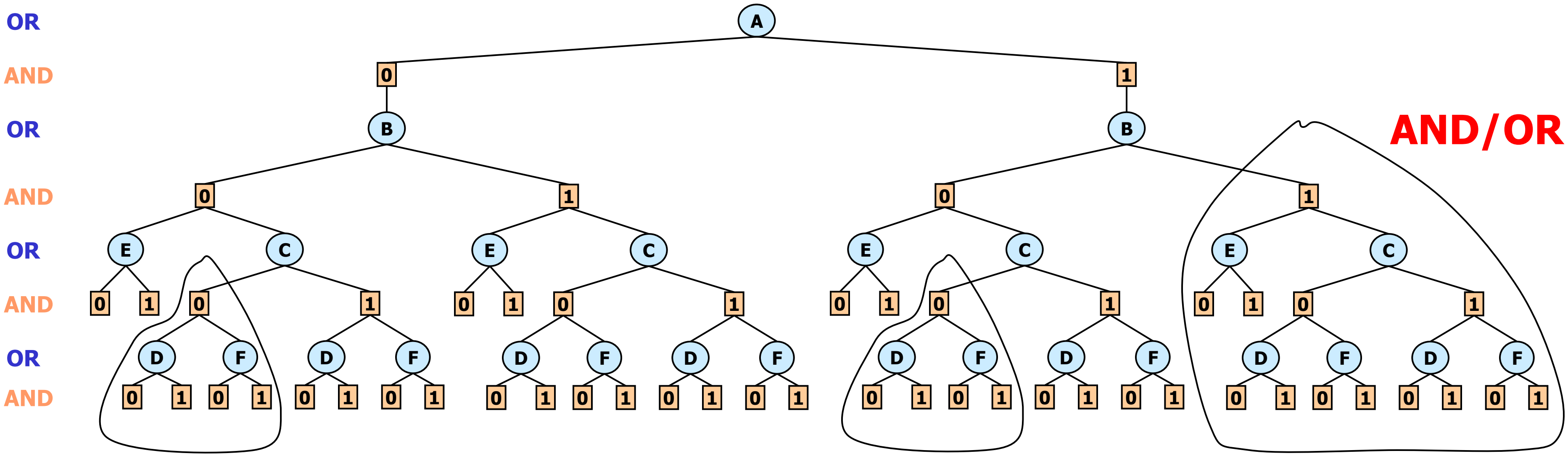
AND/OR size: exp(4),
OR size exp(6)

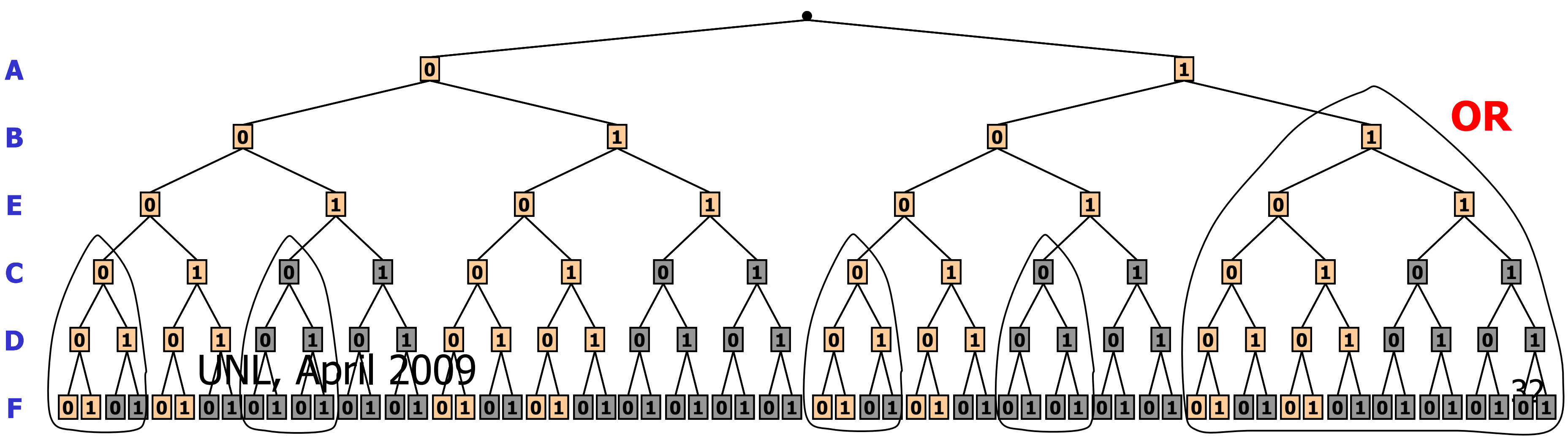


AND/OR vs. OR
with Constraints

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

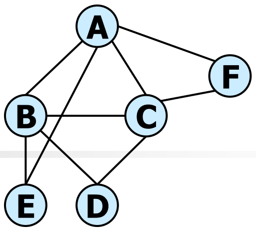


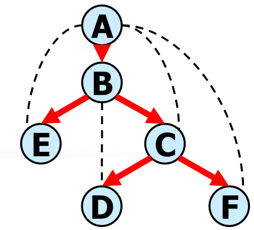


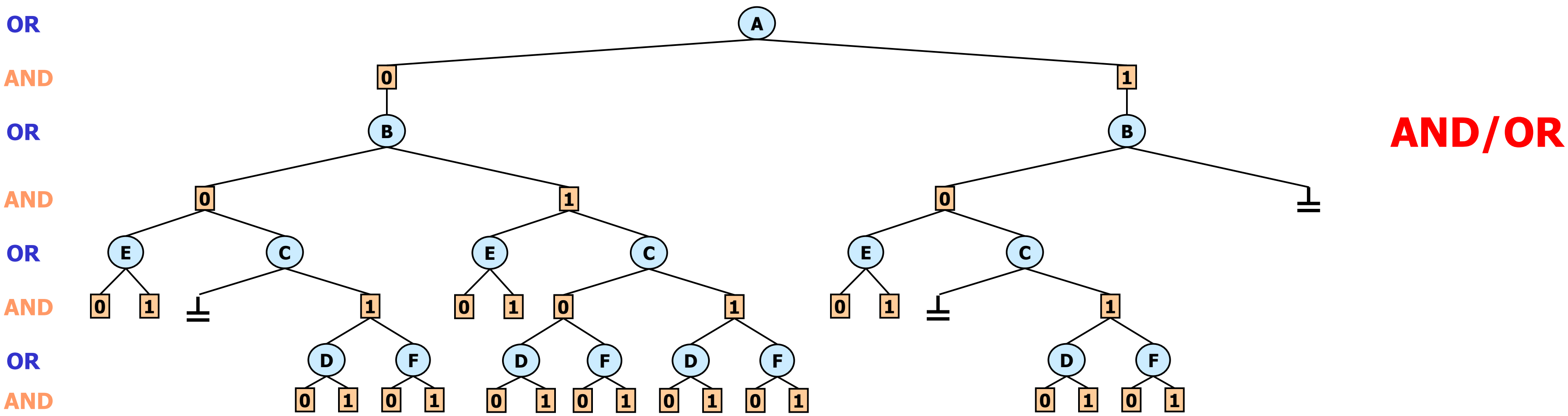


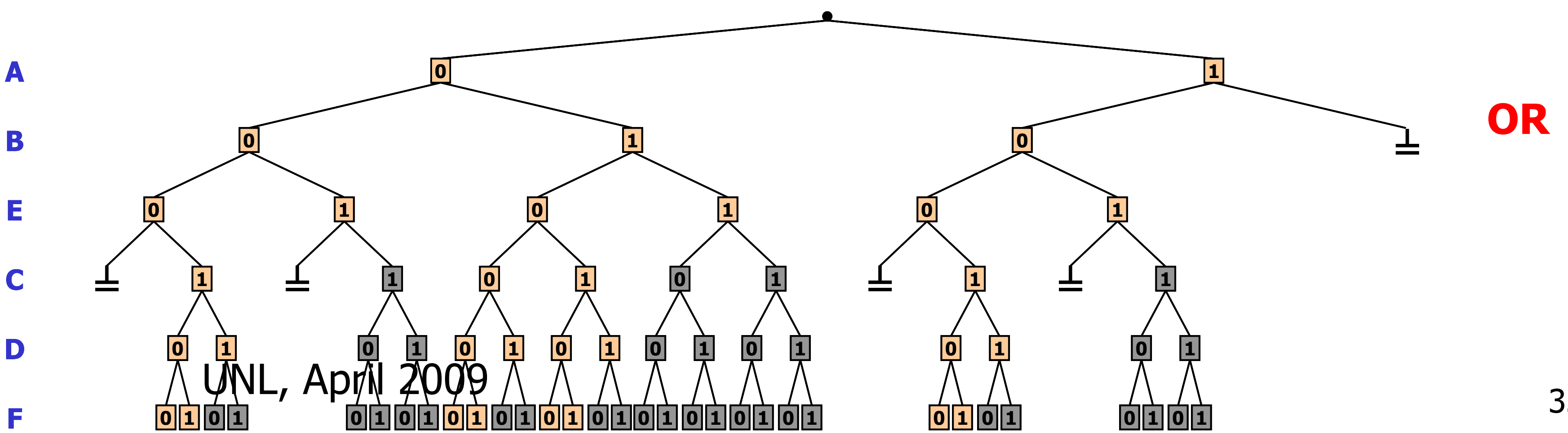
AND/OR vs. OR
with Constraints

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



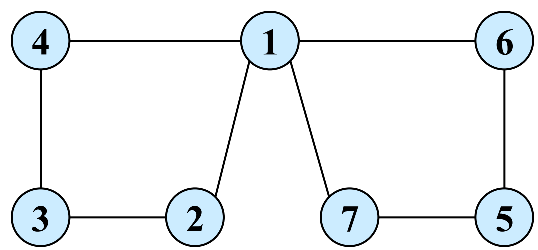






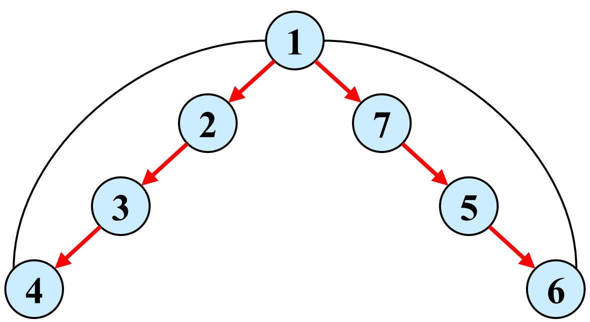
Pseudo-Trees

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

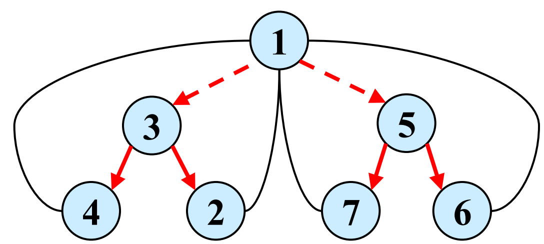


(a) Graph

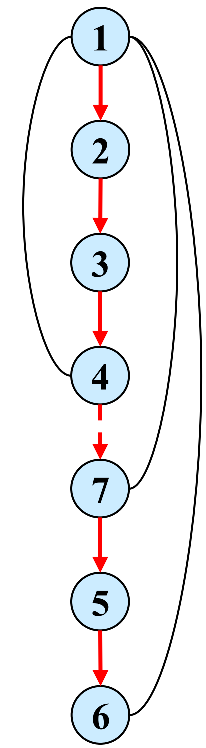
m <= w* log n



(b) DFS tree depth=3

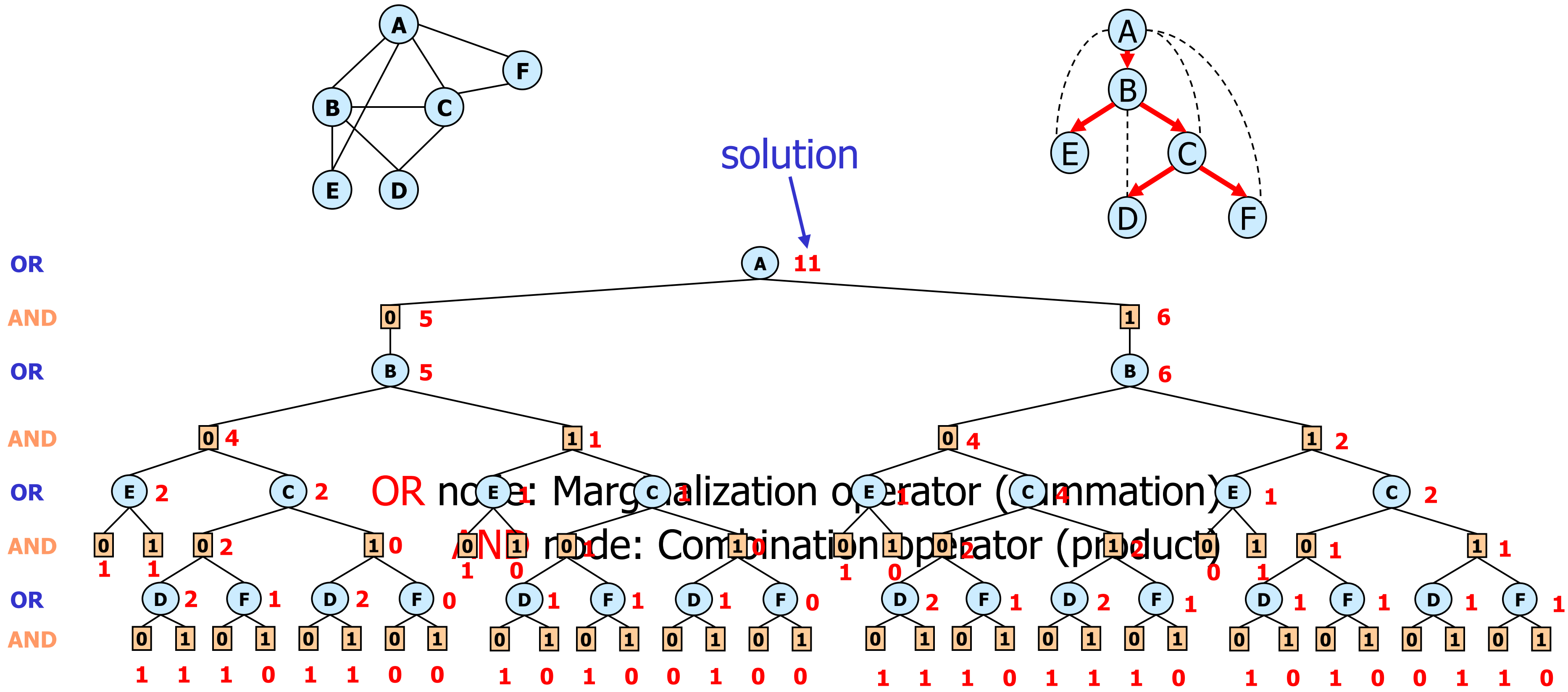


(c) pseudo- tree depth=2



(d) Chain depth=6

DFS algorithm (#CSP example)



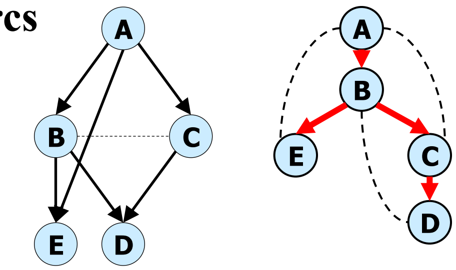
OR node: Marginalization operator (summation)

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

$P(C \mid A)$

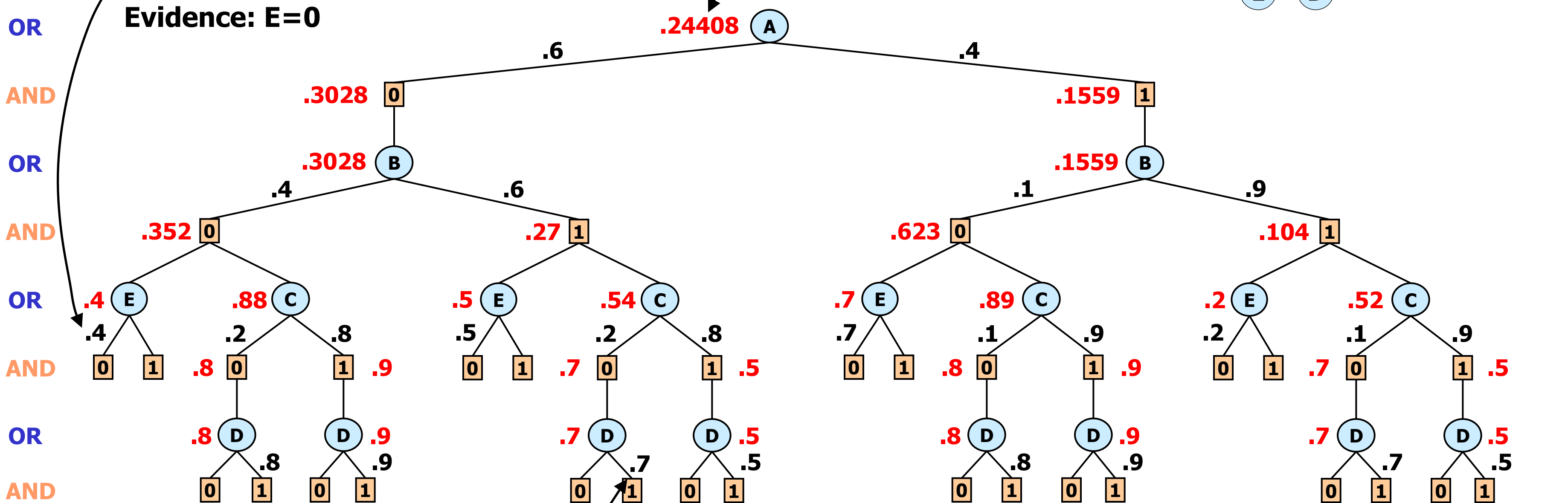
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4

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

Evidence: 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

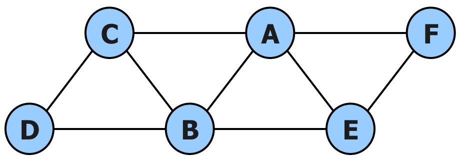
Evidence: D=1

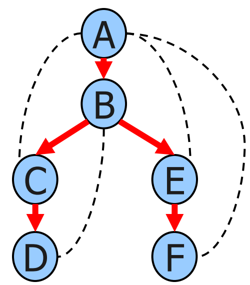
OR node: Marginalization operator (summation)

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

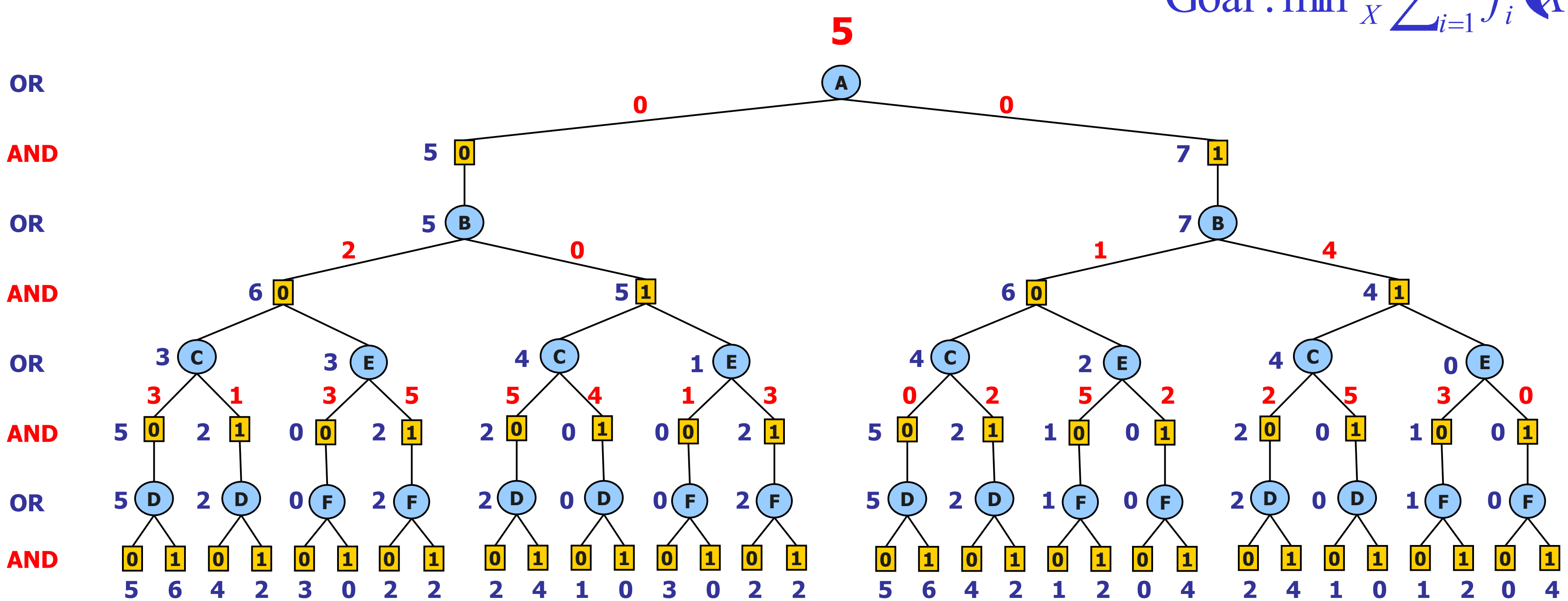
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

	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

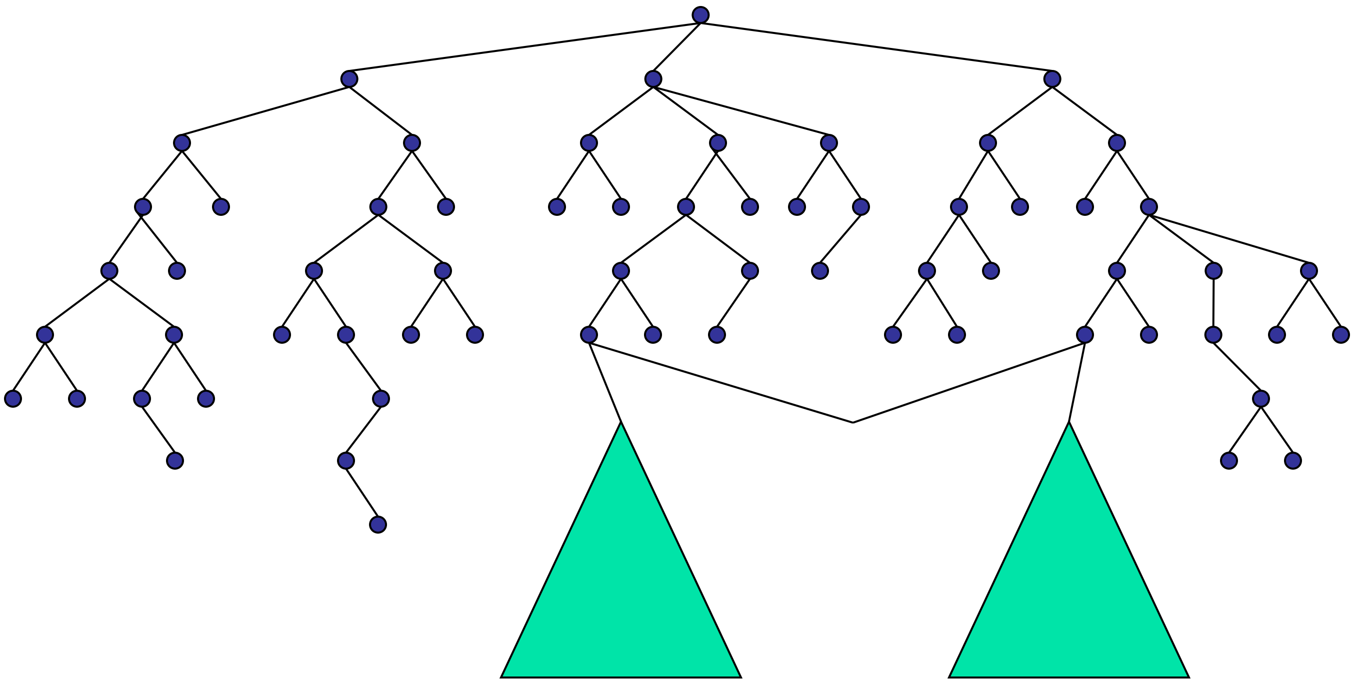
Tasks: Consistency, Counting, Optimization, Belief updating Max-expected utility, partition function

Overview

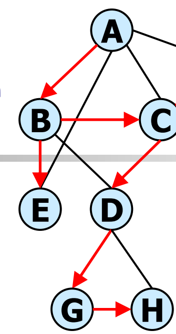
- Introduction to graphical models algorithms: Inference, search and hybrids.
- AND/OR search spaces
  - Decomposition in AND/OR trees
  - Equivalence AND/OR Graphs
- AND/OR search for combinatorial optimization
- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

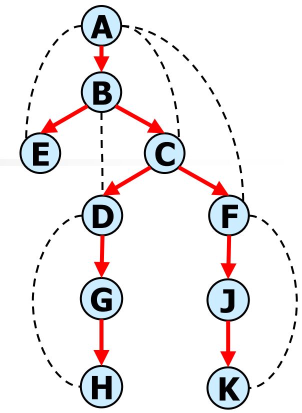
From Search Trees to Search Graphs

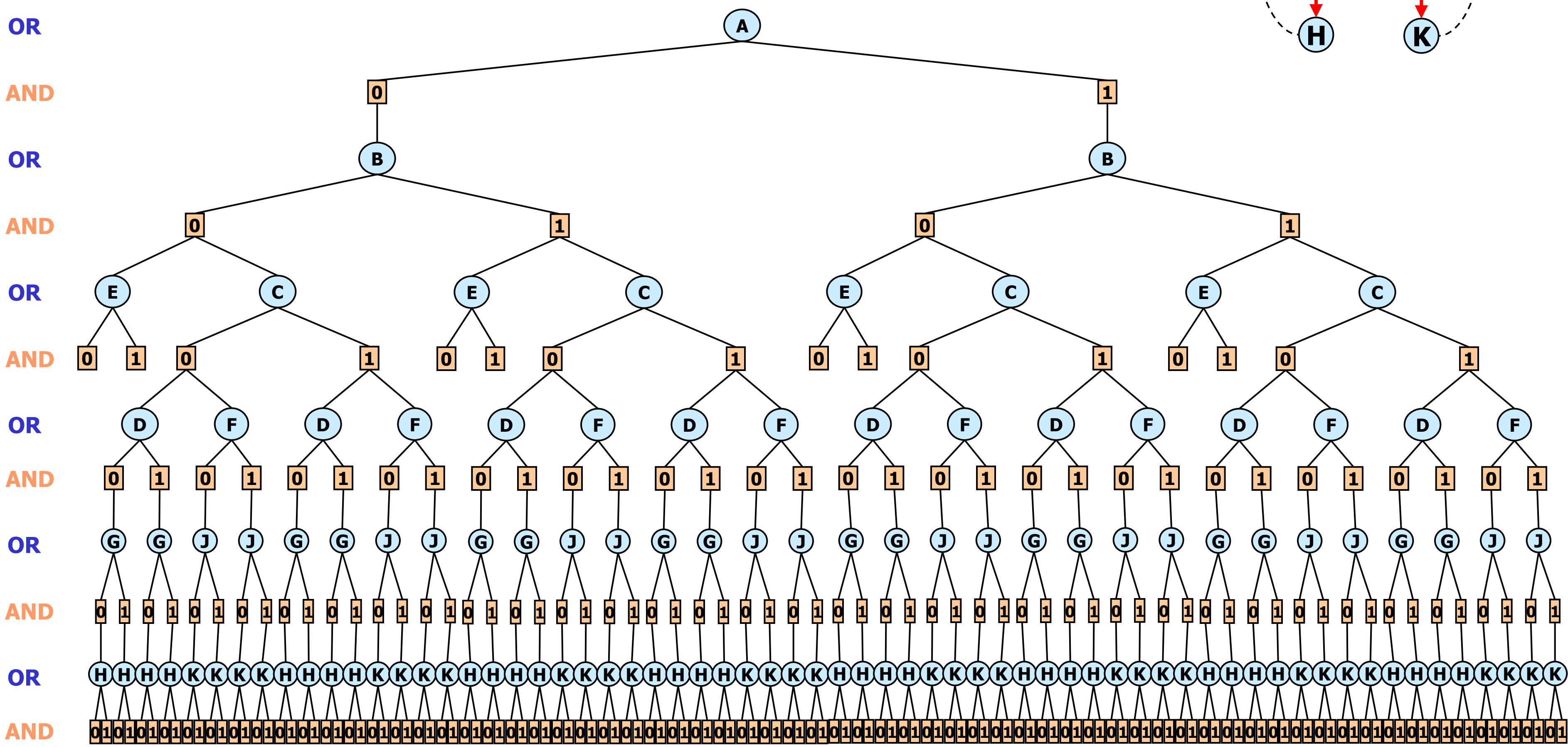
- Any two nodes that root identical subtrees (subgraphs) can be merged



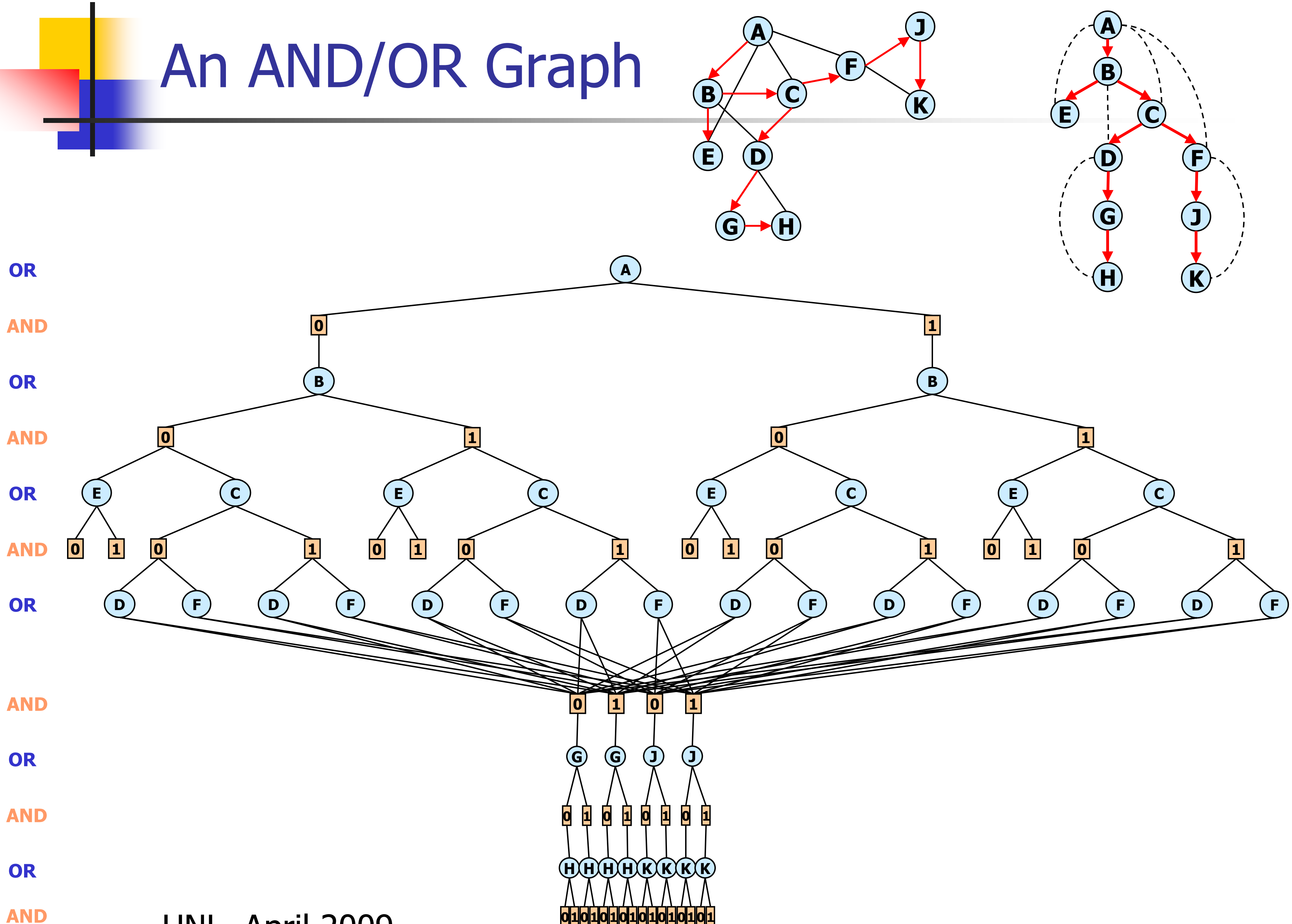
From AND/OR Tree







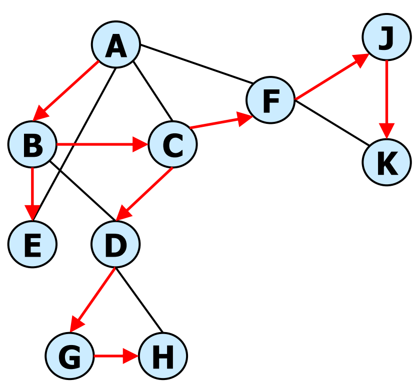
An AND/OR Graph

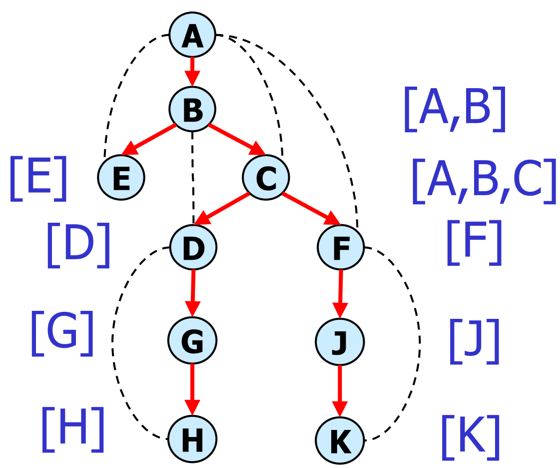


Context-based Caching

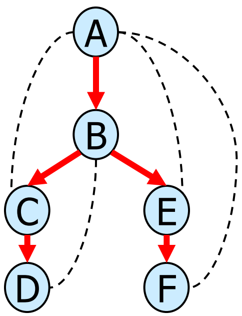
- context = current variable + ancestors connected to subtree below

- Caching is possible when context is the same

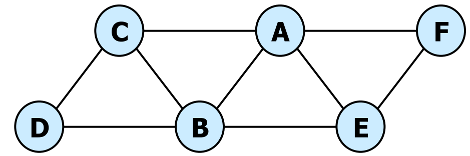




Context-based Caching



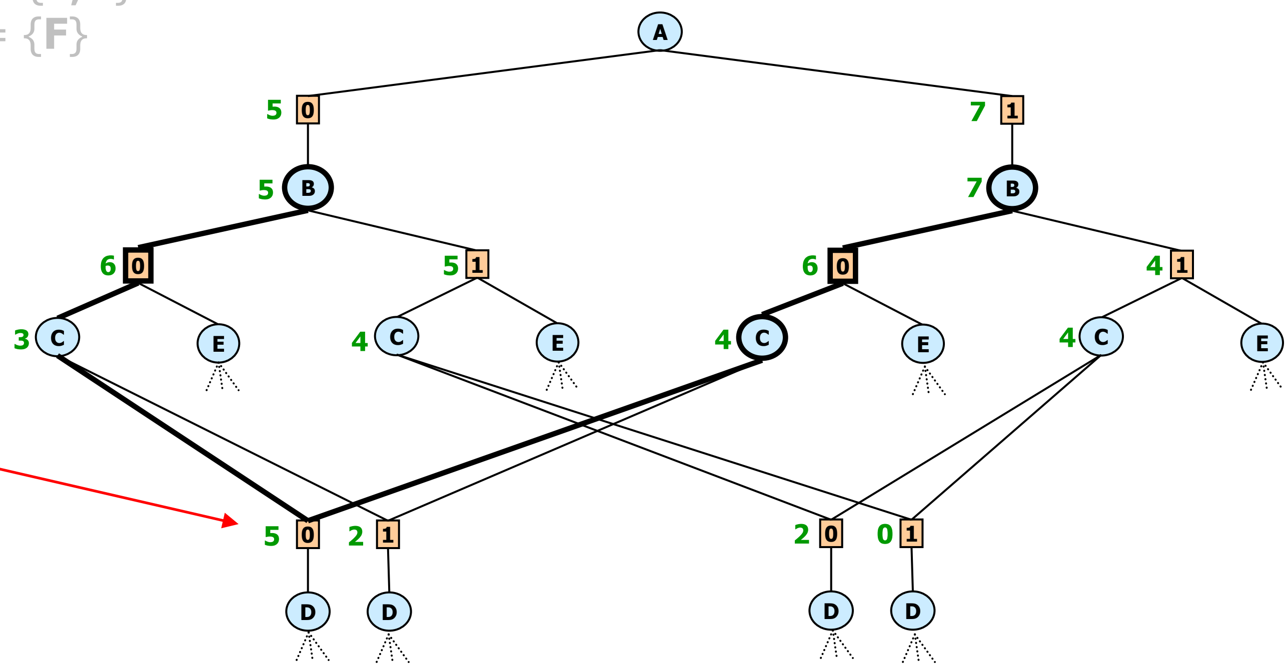
context(A) = {**A**}
context(B) = {**B,A**}
context(C) = {C,B}
context(D) = {**D**}
context(E) = {**E,A**}
context(F) = {**F**}



Primal graph

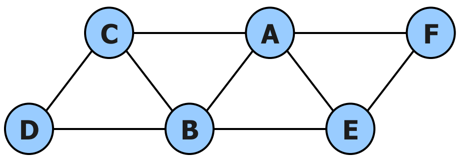
Cache Table (C)

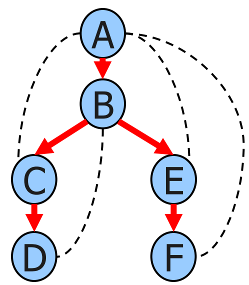
B	C	Value
0	0	5
0	1	2
1	0	2
1	1	0



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

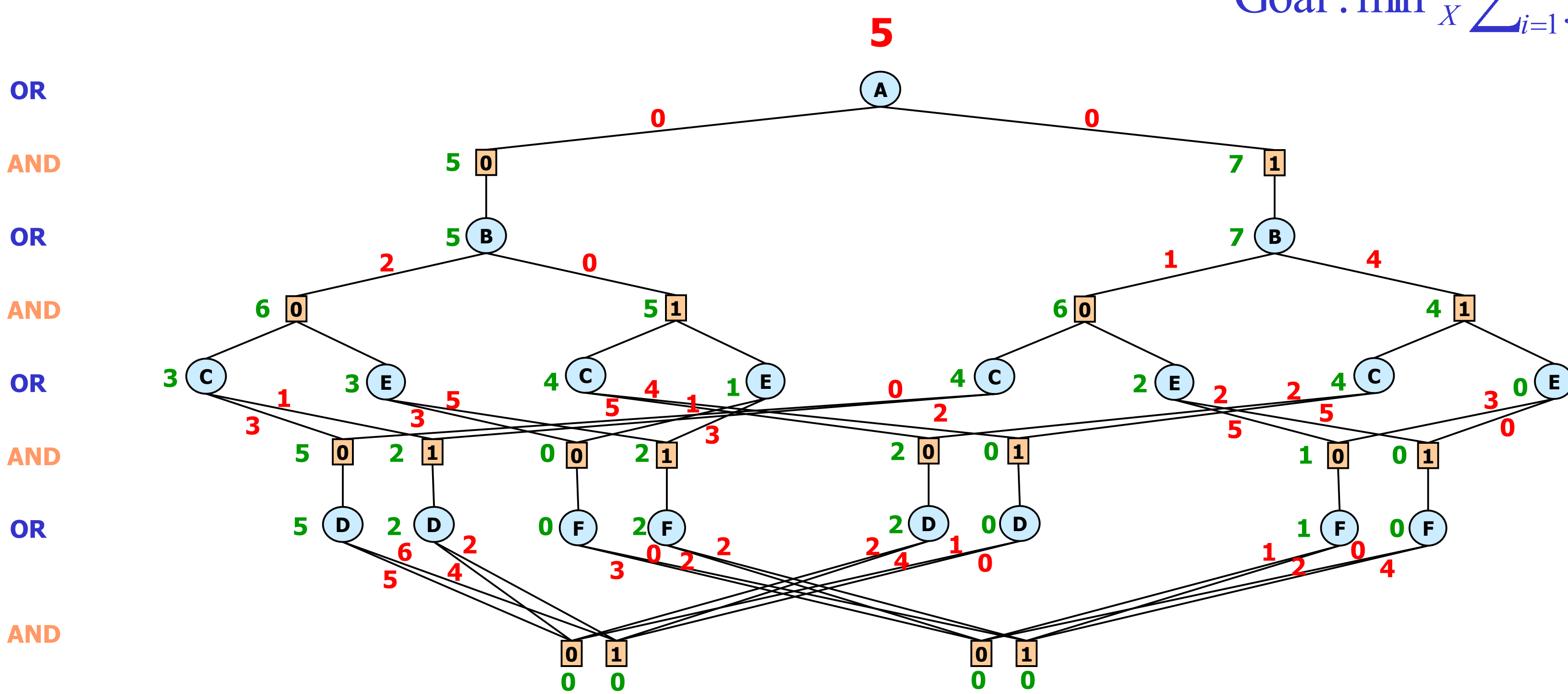
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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)

$P(E \mid A,B)$

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

$P(B \mid A)$

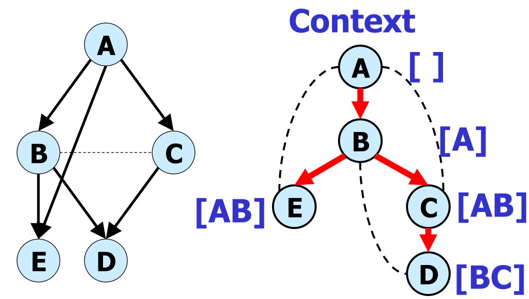
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

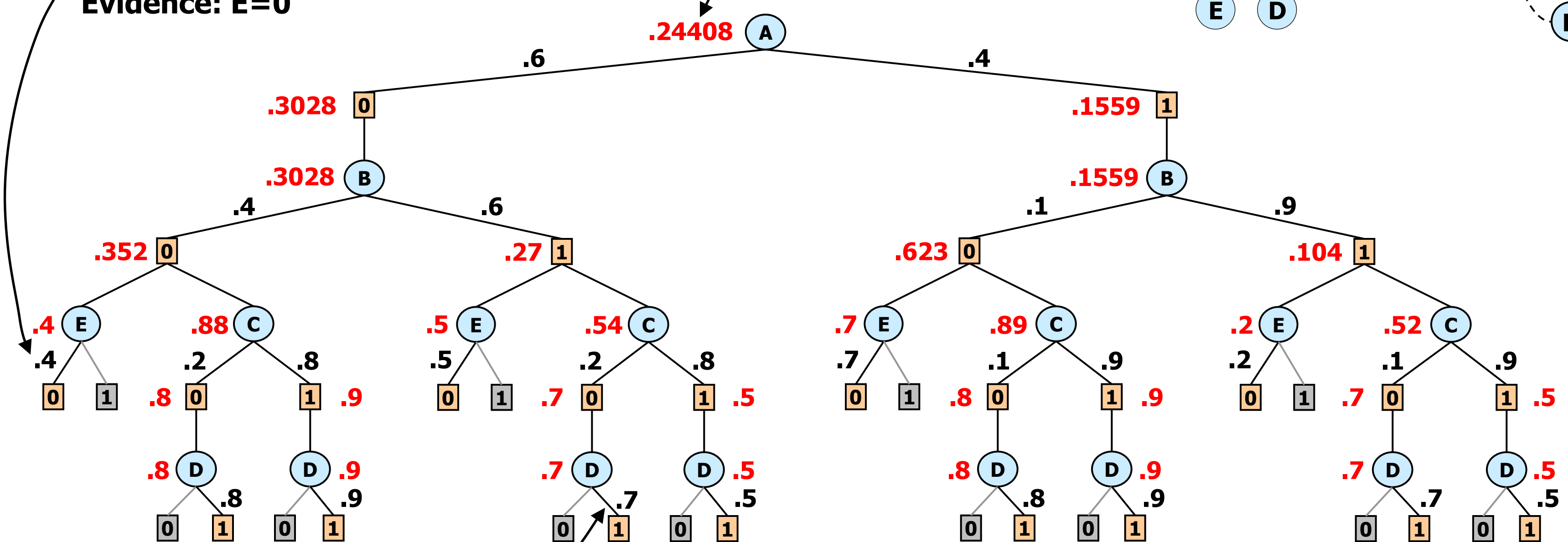
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



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



$P(D \mid B,C)$

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

Evidence: D=1

OR node: Marginalization operator (summation)
AND node: Combination operator (product)
Value of node = updated belief for sub-problem below

AND/OR Graph DFS Algorithm
(Belief Updating)

$P(E \mid A,B)$

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

Evidence: E=0

$P(B \mid A)$

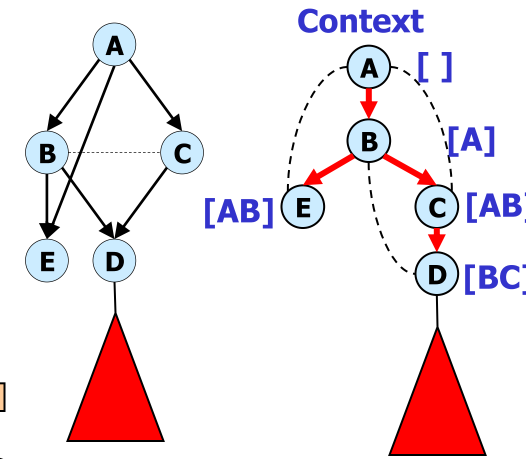
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

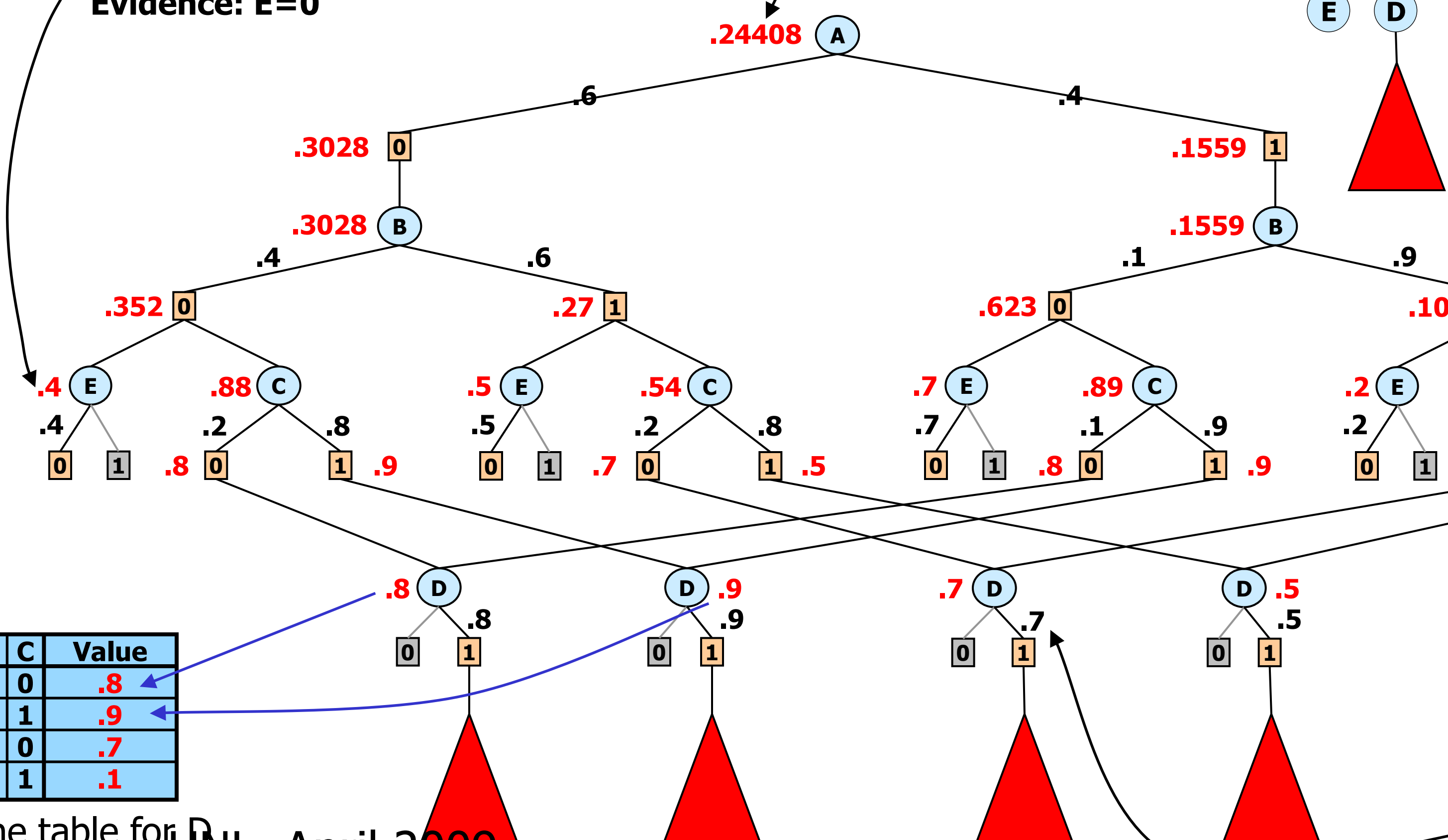
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

A	P(A)
0	.6
1	.4



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



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

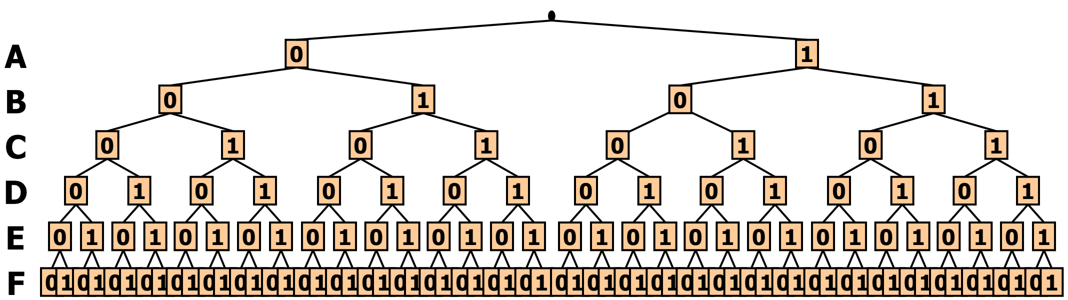
Cache table for D

$P(D \mid B,C)$

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

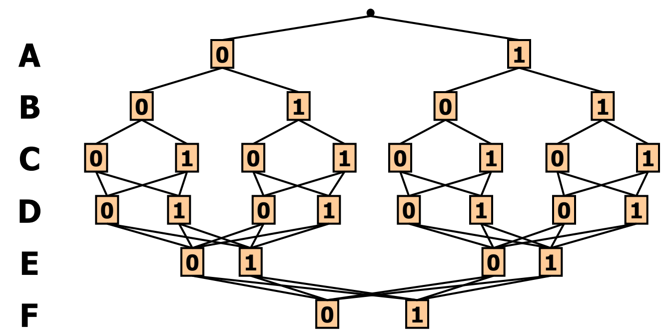
Evidence: D=1

All Four Search Spaces



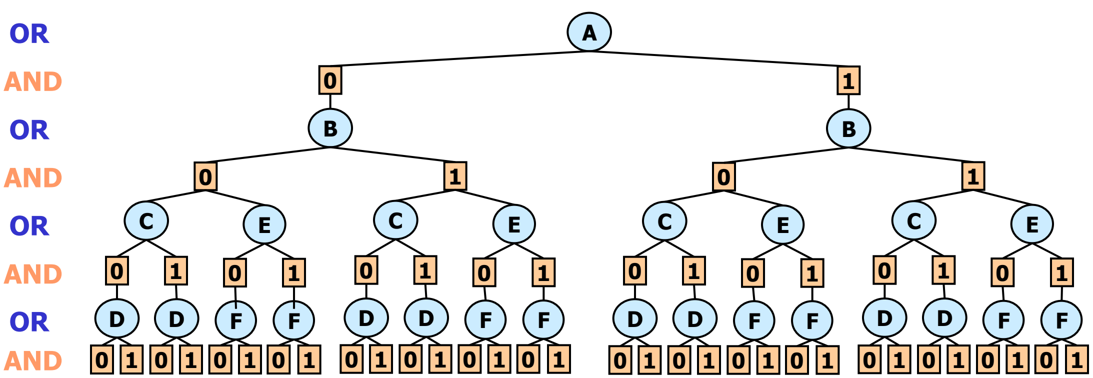
Full OR search tree

126 nodes



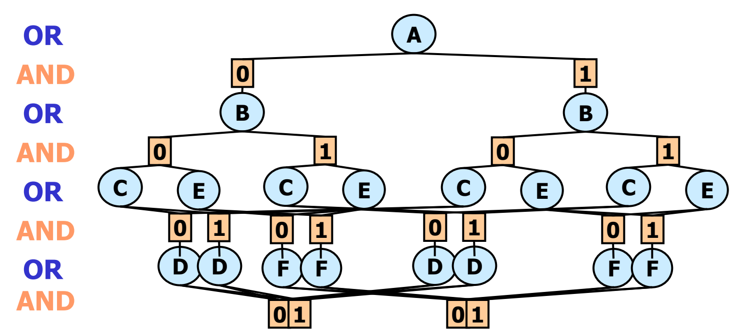
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes

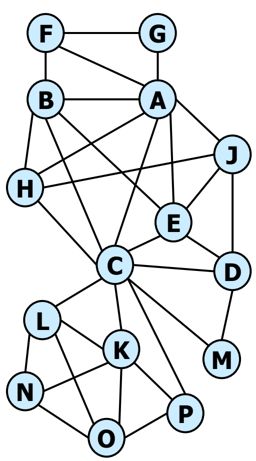


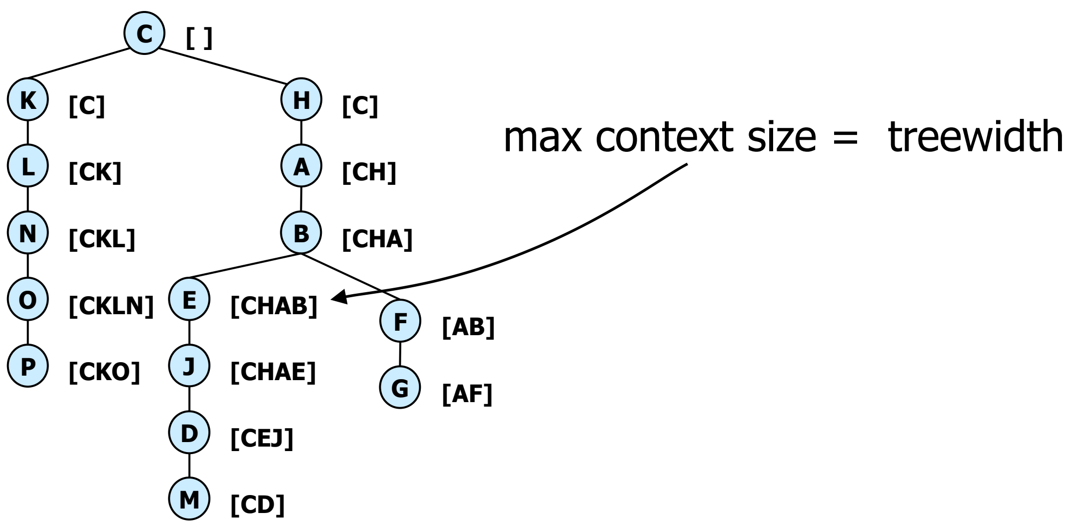
Context minimal AND/OR search graph

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

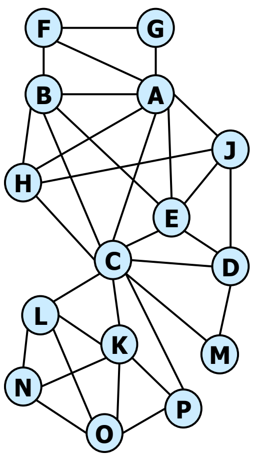
	AND/OR graph	OR graph
Space	$O(n\ k^{w^*})$	$O(n\ k^{pw^*})$
Time	$O(n\ k^{w^*})$	$O(n\ k^{pw^*})$

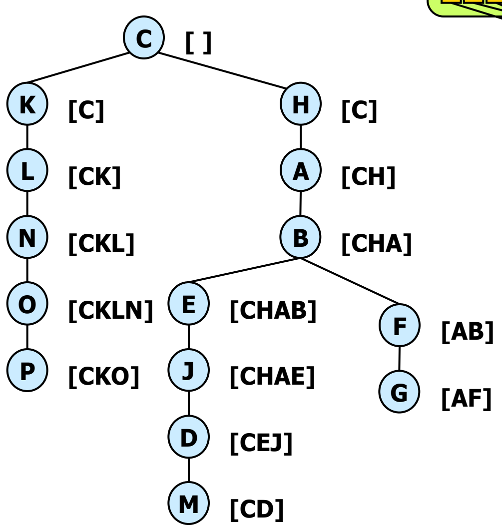
$w^* \le pw^* \le w^* \log n$

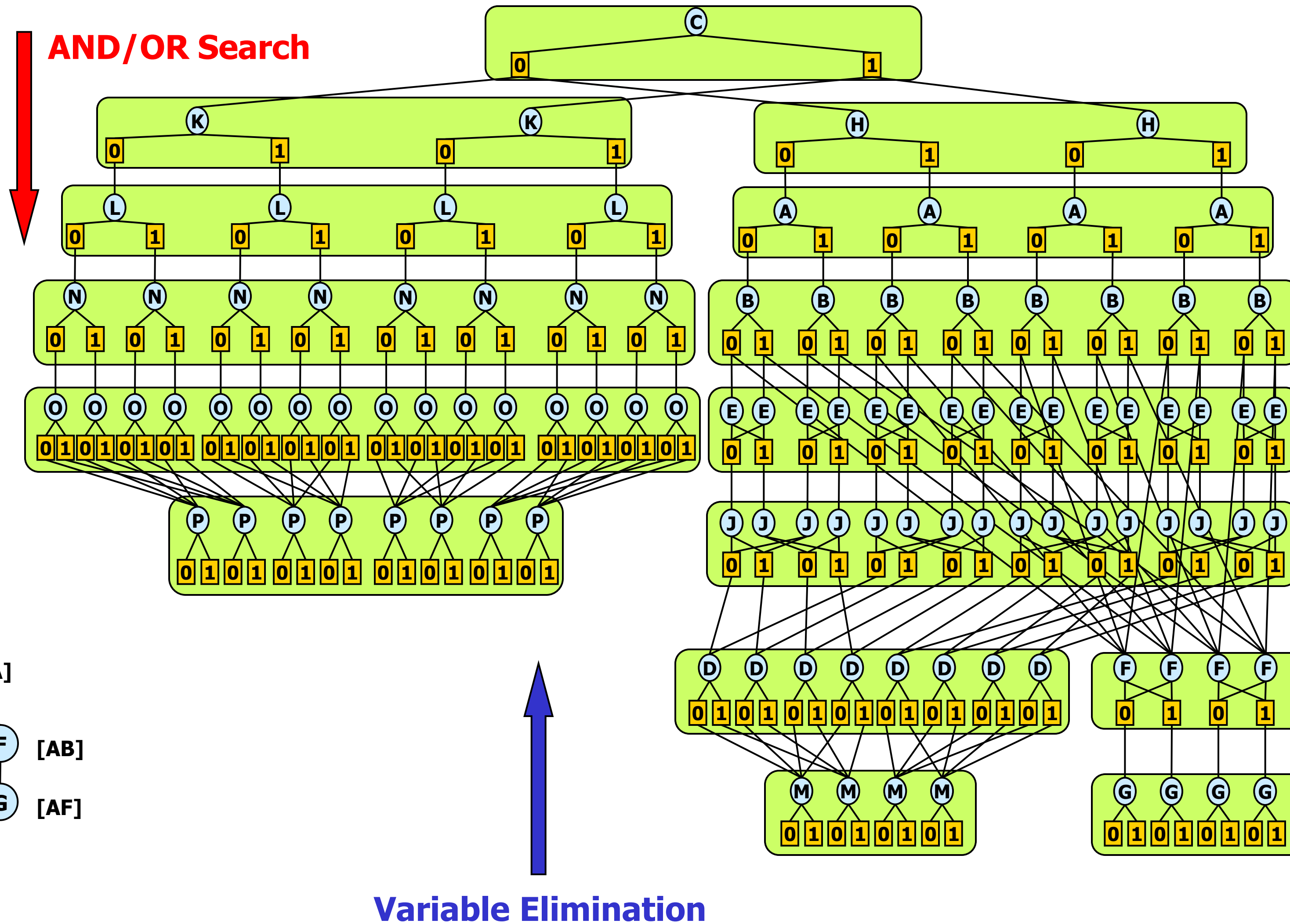
k = domain size
n = number of variables
w* = treewidth
pw* = pathwidth

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

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))

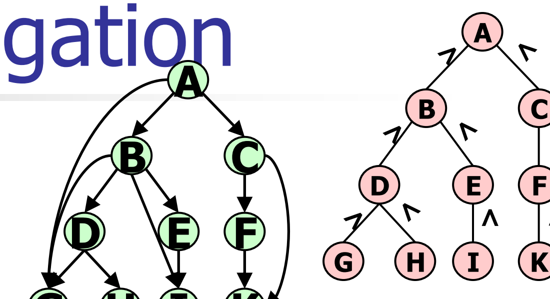
Space: O(exp(j))

Time: O(exp(m_j+j)

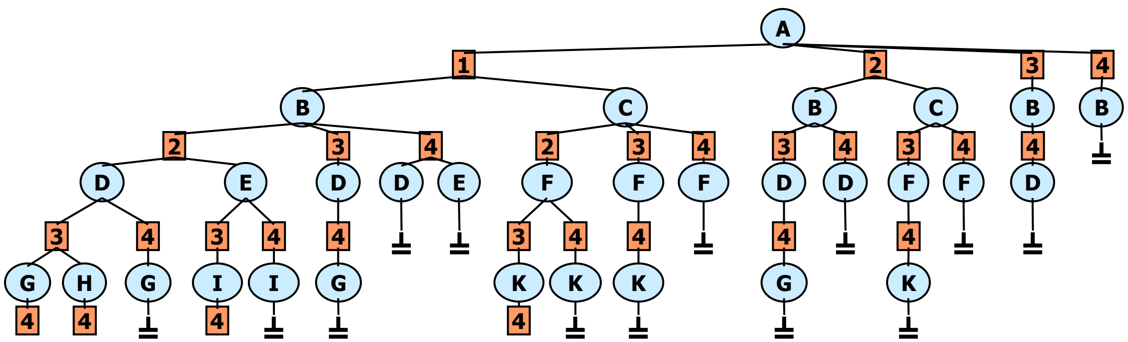
Space: O(exp w*)

Time: O(exp w*)

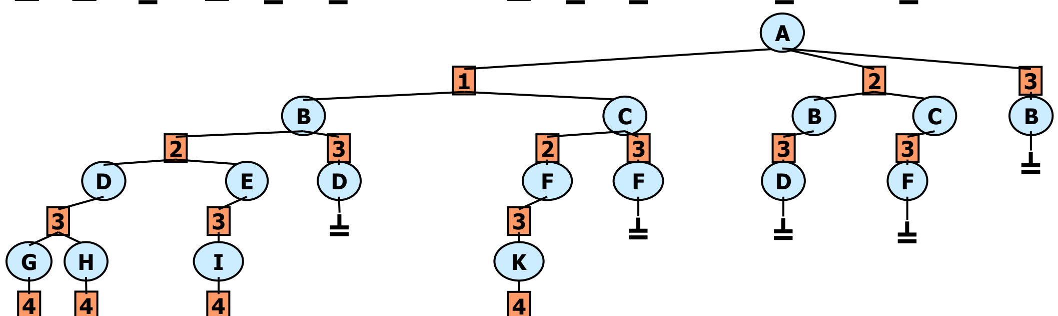
The Effect of Constraint Propagation



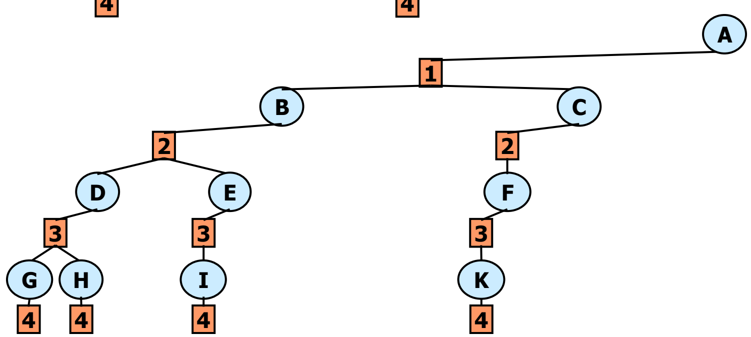
Domains are {1,2,3,4}



CONSTRAINTS ONLY



FORWARD CHECKING



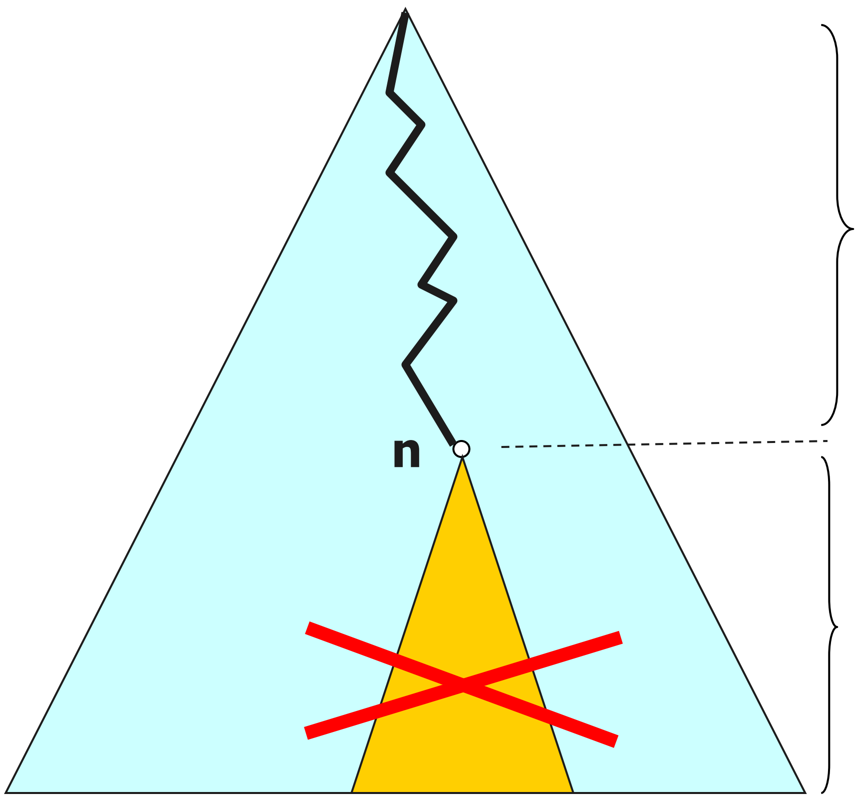
MAINTAINING ARC CONSISTENCY

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 Approximation
(Dechter & Rish, 1997)

Split a bucket into mini-buckets => bound complexity

bucket (X) =

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

$$h^X = \min_X \sum_{i=1}^{n} h_i$$

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

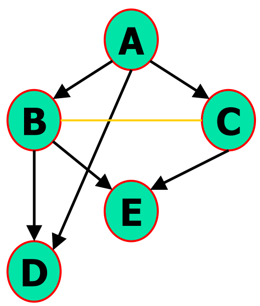
$$g^X = \left(\min_X \sum_{i=1}^{r} h_i\right) + \left(\min_X \sum_{i=r+1}^{n} h_i\right)$$

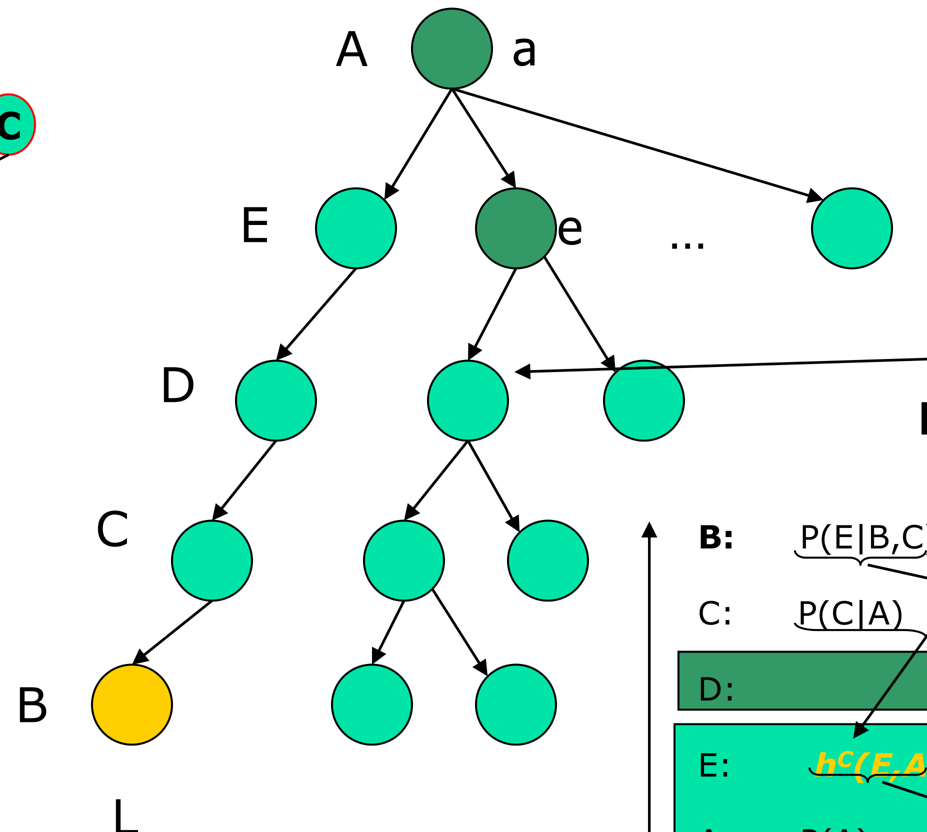
$$g^X \leq i^X$$

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

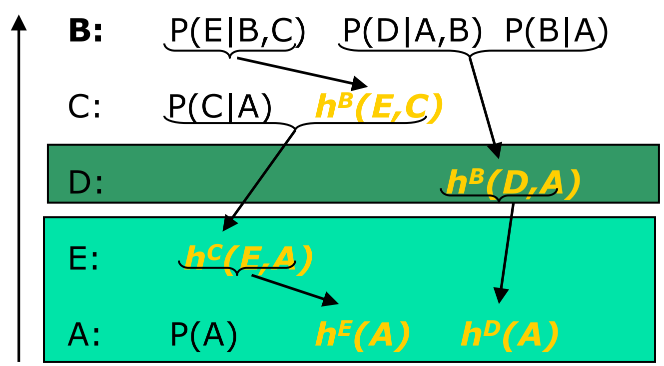
Mini-bucket Heuristics for BB search

(Kask and dechterAIJ, 2001, Kask, Dechter and Marinescu UAI 2003)



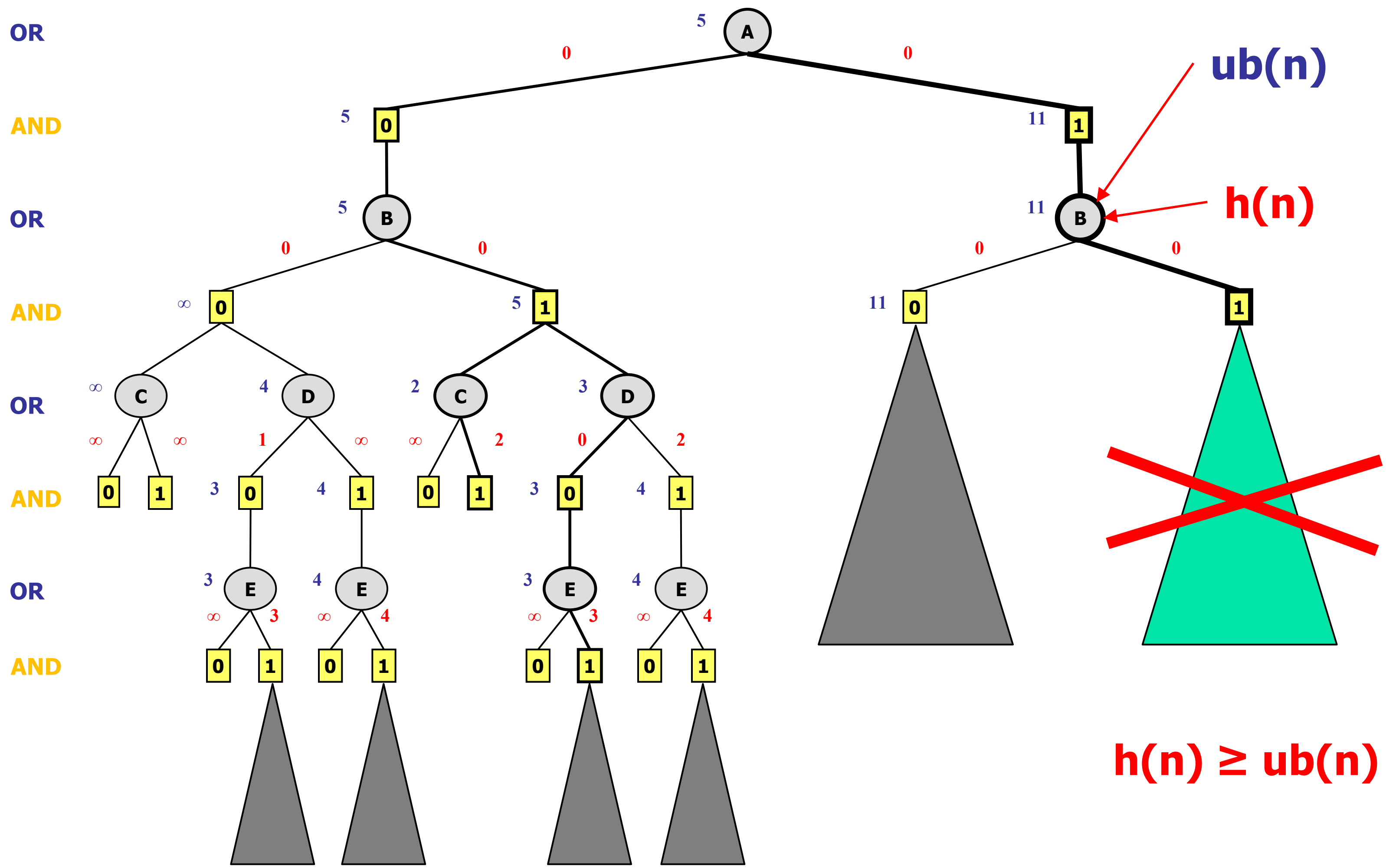


h(x) computed by MB(i) before or during search



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

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

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

Experiments

- Benchmarks
  - Belief Networks (BN)
  - Weighted CSPs (WCSP)
- Algorithms
  - AND/OR Branch-and-Bound
  - Best-first AND/OR Search
  - SamIam (BN)
  - Superlink (linkage)
  - Toolbar, Toolbar-BTD (WCSP)
- Heuristics
  - Mini-Bucket heuristics

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

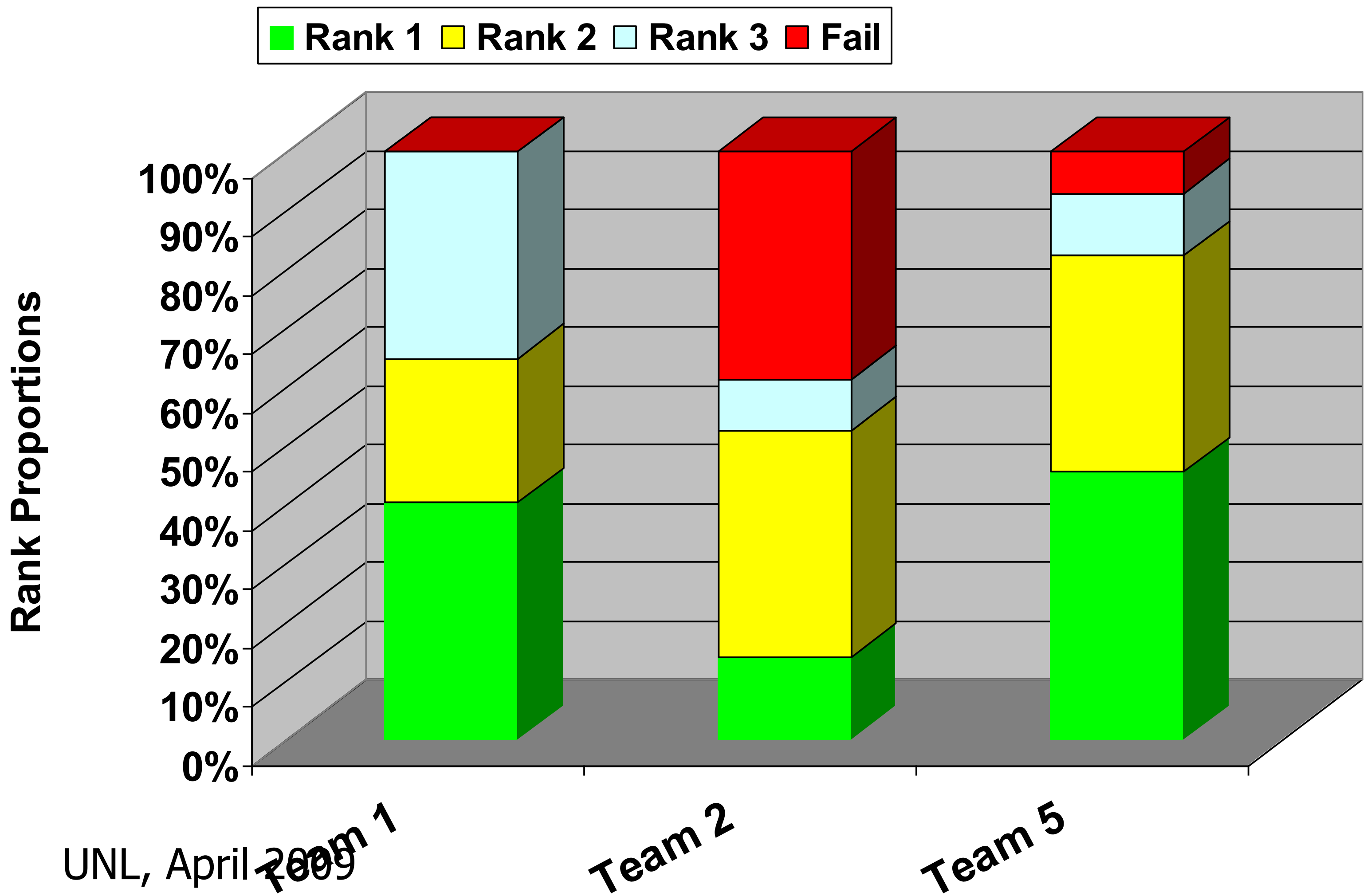
MAX-SAT Instances
(Marinescu & Dechter, CPAIOR'07)

pret (w*, h)	**CPLEX** time	nodes	**AOBB-C** time	nodes	**AOBF-C** time	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

UAI’06 Results

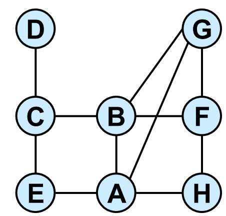
Rank Proportions (how often was each team a particular rank, rank 1 is best)



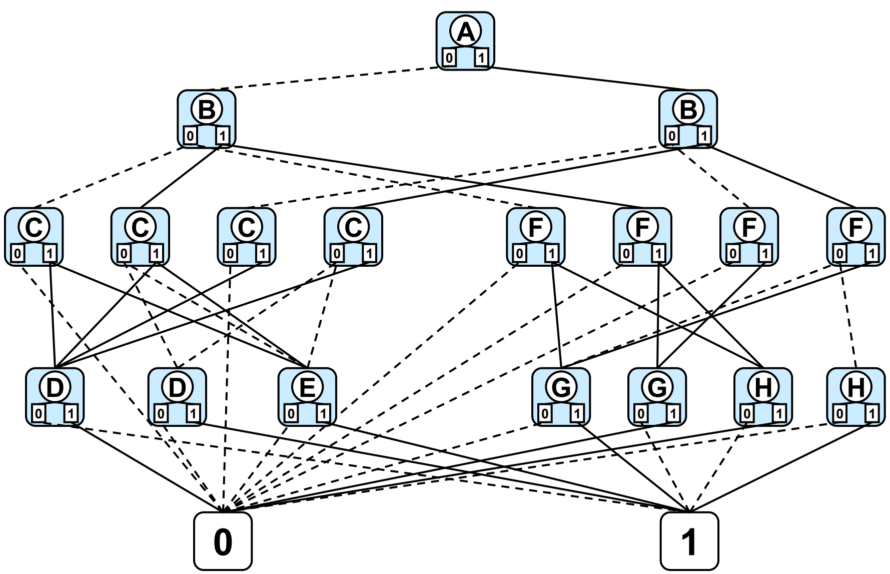
Overview

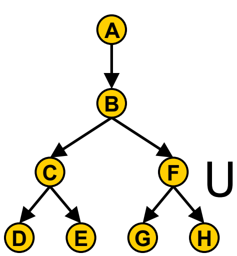
- Introduction to graphical models algorithms: Inference, search and hybrids.
- AND/OR search spaces
  - AND/OR trees
  - AND/OR Graphs
- AND/OR search for combinatorial optimization
  - The mini-bucket heuristic
  - AO depth-first and best-first Branch and Bound
  - Empirical evaluation
- Current focus:
  - AND/OR Compilation
  - Approximation by Sampling and belief propagation

AOBDD vs. OBDD



primal graph

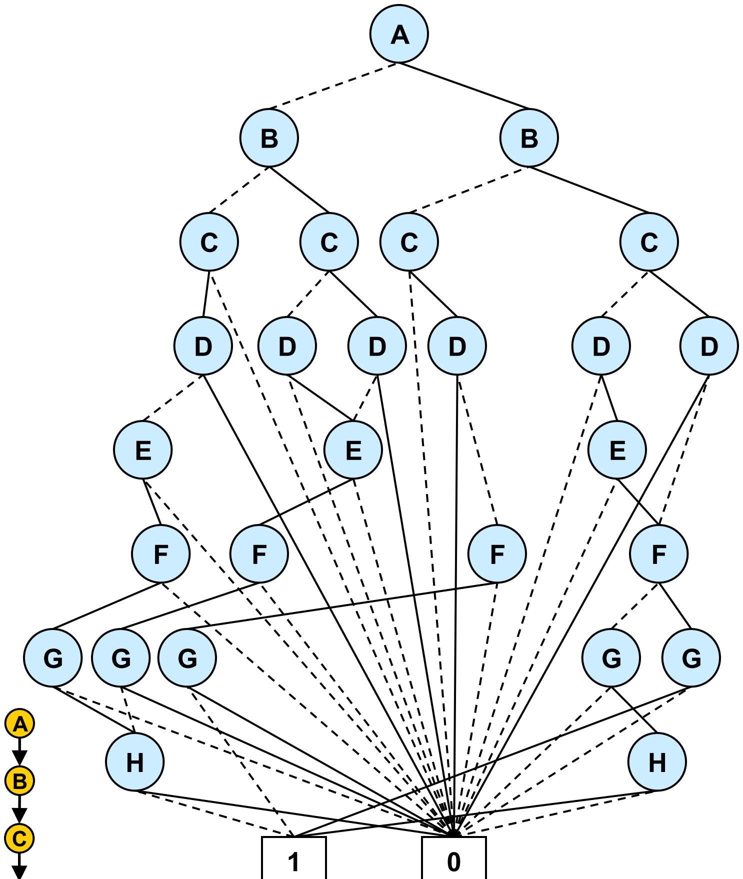


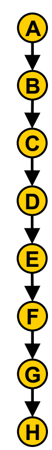


AOBDD

18 nonterminals

47 arcs





OBDD

27 nonterminals

54 arcs

Improving Importance Sampling

Sampling: a scheme that generates a portion of the search space.

SampleSearch: generates a solution sample by exploiting backtrack techniques.

AND/OR sampling: samples the AND/OR search tree or graph.

W-cutset sampling: samples the hybrid search and inference space

Genetic Linkage Analysis (BN)

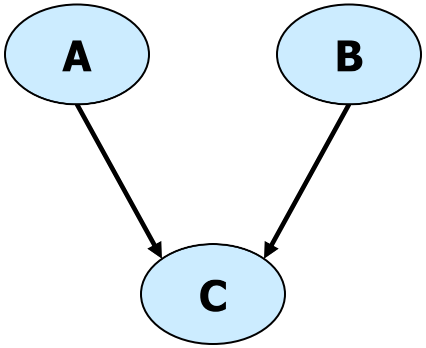
(Fishelson&Geiger02)

pedigree (w*, h) (n, d)	SamIam Superlink	MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=12		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=14		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=16		MBE(i) BB-C+SMB(i) AOBB+SMB(i) AOBB-C+SMB(i) AOBF-C+SMB(i) i=18	
		time	nodes	time	nodes	time	nodes	time	nodes
ped30 (23, 118) (1016, 5)	out 13095.83	0.42 - - 10212.70 out	- - 93,233,570	0.83 - - 8858.22 out	- - 82,552,957	1.78 - - - out	- - -	5.75 - 214.10 34.19 **30.39**	- 1,379,131 193,436 72,798
ped33 (37, 165) (581, 5)	out -	0.58 - 2804.61 1426.99 out	- 34,229,495 11,349,475	2.31 - 737.96 307.39 140.61	- 9,114,411 2,504,020 407,387	7.84 - 3896.98 1823.43 out	- 50,072,988 14,925,943	33.44 - 159.50 86.17 **74.86**	- 1,647,488 453,987 134,068
ped42 (25, 76) (448, 5)	out 561.31	4.20 - - - out	- - -	31.33 - - - out	- - -	96.28 - - 2364.67 **133.19**	- - 22,595,247 93,831	out	

Additional advances

- Extend well known principles to AND/OR search
  - Constraint propagation (handle determinism)
  - Good initial upper bounds (via local search)
  - Improving the quality of the guiding pseudo tree

Example – CNF encoding



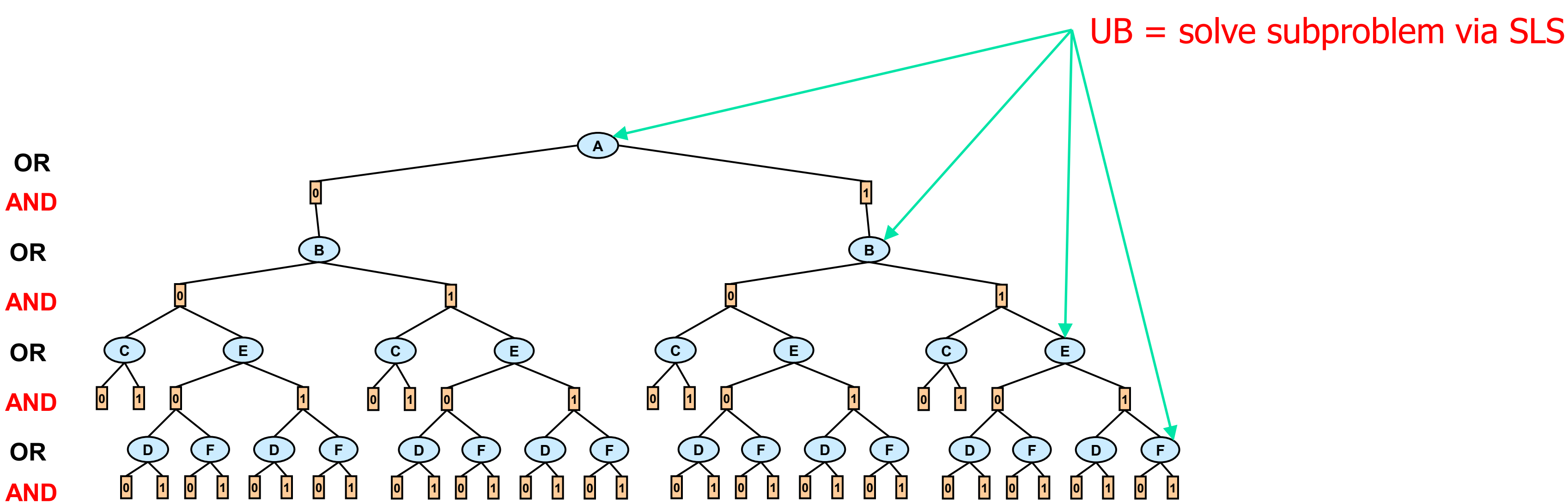
P(C|A, B)

A	B	C	$P(C\|A,B)$	Clauses
1	1	1	1	
1	1	2	0	$(\neg L_{A,1} \vee \neg L_{B,1} \vee \neg L_{C,2})$
1	1	3	0	$(\neg L_{A,1} \vee \neg L_{B,1} \vee \neg L_{C,3})$
1	2	1	0	$(\neg L_{A,1} \vee \neg L_{B,2} \vee \neg L_{C,1})$
1	2	2	1	
1	2	3	0	$(\neg L_{A,1} \vee \neg L_{B,2} \vee \neg L_{C,3})$
2	1	1	.2	
2	1	2	.8	
2	1	3	0	$(\neg L_{A,2} \vee \neg L_{B,1} \vee \neg L_{C,3})$
2	2	1	.7	
2	2	2	.3	
2	2	3	0	$(\neg L_{A,2} \vee \neg L_{B,2} \vee \neg L_{C,3})$

Clauses representing 0 probability entries, based on direct encoding (Walsh00)

Initial Upper Bounds

- AND/OR Branch-and-Bound search assumes a trivial initial upper bound at OR nodes
  - Guarantees optimality but provides limited pruning

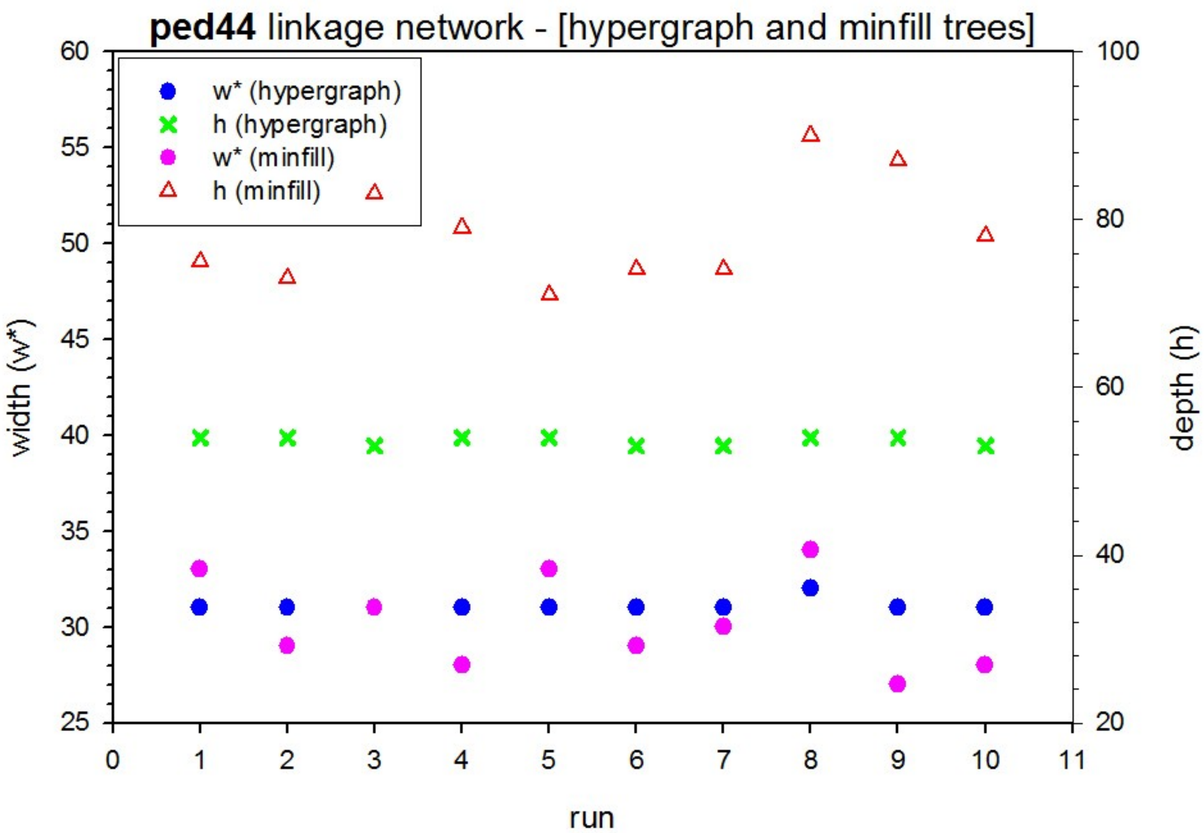


Genetic Linkage Analysis

pedigree (n, d)	SamIam Superlink CPLEX	**hypergraph pseudo tree** (w*, h)	AOBB-C+SMB(i) AOBB-C+SAT+SMB(i) AOBB-C+GLS+SMB(i) AOBF-C+SMB(i) i=20 time	 nodes	AOBB-C+SMB(i) AOBB-C+SAT+SMB(i) AOBB-C+GLS+SMB(i) AOBF-C+SMB(i) i=22 time	 nodes	**min-fill pseudo tree** (w*, h)	AOBB-C+SMB(i) AOBB-C+SAT+SMB(i) AOBB-C+GLS+SMB(i) AOBF-C+SMB(i) i=20 time	 nodes	AOBB-C+SMB(i) AOBB-C+SAT+SMB(i) AOBB-C+GLS+SMB(i) AOBF-C+SMB(i) i=22 time	 nodes
ped7 (868, 4)	out - out	(36, 60)	30504.84 31701.54 30349.92 out	285,084,124 285,084,124 284,635,328	3005.66 3116.07 **2955.06** out	27,761,219 27,761,219 **27,371,526**	(32, 133)	- - - out	- - -	out	
ped9 (936, 7)	out - out	(35, 58)	8922.81 10075.90 8866.40 out	117,328,162 117,328,162 117,011,941	3292.30 3657.91 3336.86 out	40,251,723 40,251,723 40,251,661	(27, 130)	1434.74 1515.50 **1163.09** out	15,825,340 15,825,340 12,444,961	out	
ped19 (693, 5)	out - out	(35, 53)	45075.31 47986.66 44585.84 out	466,748,365 466,748,365 459,741,495	8321.42 8774.51 **8070.95** out	90,665,870 90,665,870 **87,060,723**	(24, 122)	out		out	
ped34 (923, 4)	out - out	(34, 60)	67647.42 74020.63 64136.36 out	1,293,350,829 1,293,350,829 1,230,870,576	11719.28 12847.33 **11005.18** out	220,199,927 220,199,927 **218,890,668**	(32, 127)	out		out	
ped41 (886, 5)	out - out	(36, 61)	3891.86 4055.15 3869.31 out	31,731,270 31,731,270 31,729,654	380.01 390.93 **374.95** out	2,318,544 2,318,544 **2,317,321**	(33, 128)	out		out	

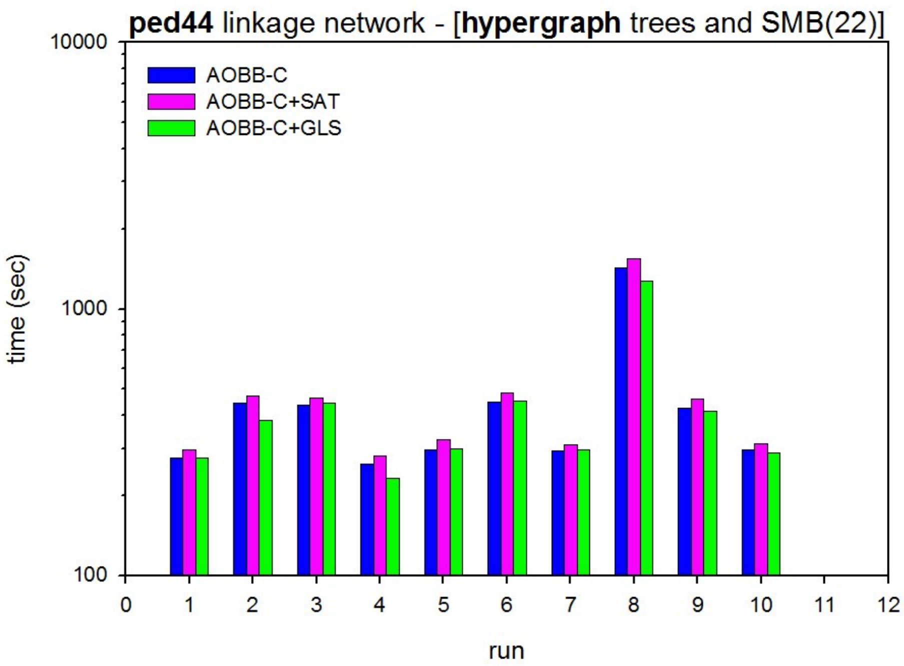
Impact of the pseudo tree quality. Time limit 24 hours.

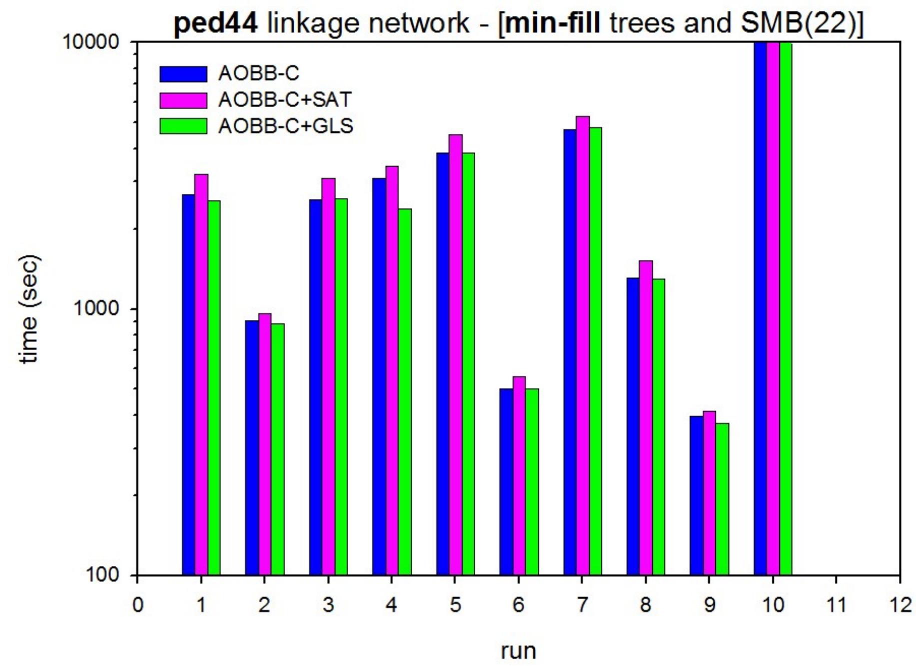
Ped44: hypergraph vs minfill trees



Pseudo tree depth and induced width obtained with randomized hypergraph and min-fill heuristics. The tree depth is plotted on a different scale to the right.

Ped44: random runs





Detailed execution of **AOBB-C+SMB(22)**, **AOBB-C+SAT+SMB(22)** and **AOBBC+GLS+SMB(22)** on the **ped44** linkage network over 10 runs using randomized min-fill and hypergraph based pseudo trees.

Recent work

- **Radu Marinescu (PhD 2008):** 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 (AOMDDs) (CP2006,UAI2007, CP2007)
  - AND/OR compilation for weighted models and optimization (JAIR-2008)

- **Vibhav Gogate:** Sampling schemes for mixed networks
  - (UAI2005, IJCAI05, CP2006)
  - SampleSearch scheme, for inference and lower-bounding (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 UAI-2008 results

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

are available at:

  - http://graphmod.ics.uci.edu/group
  - http://graphmod.ics.uci.edu/uai08/Evaluation/Report

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**

Thank you