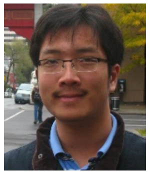
Weighted AND/OR Graphs/Diagrams for Probabilistic and constraints Databases.

Rina Dechter

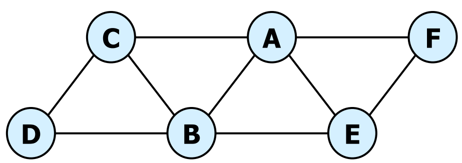
Bren School of Information and Computer Sciences, UC-Irvine,

Joint work with Robert Mateescu, Radu Marinescu and William Lam





A Constraint Network and its Search Graphs



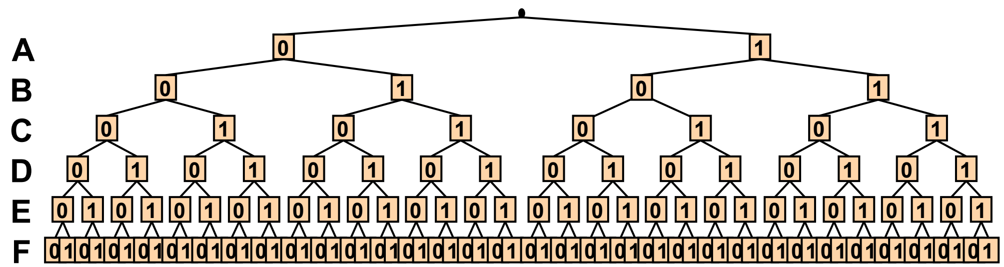
A	B	C	R_{ABC}
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

B	C	D	R_{BCD}
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

A	B	E	R_{ABE}
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

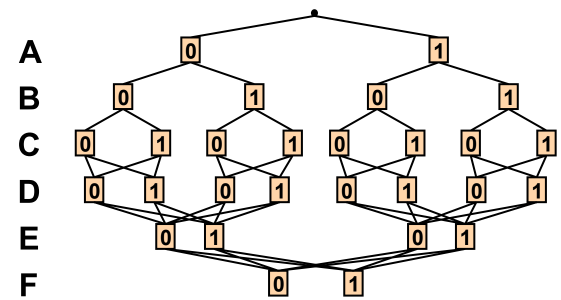
A	E	F	R_{AEF}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Context-Minimal AND/OR Graph



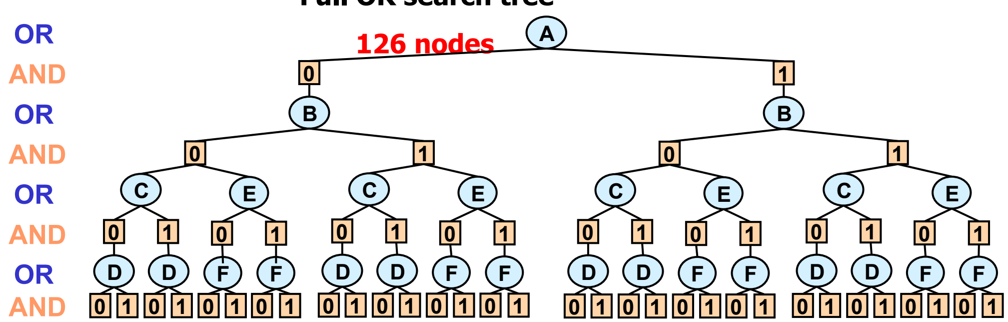
Full OR search tree

126 nodes



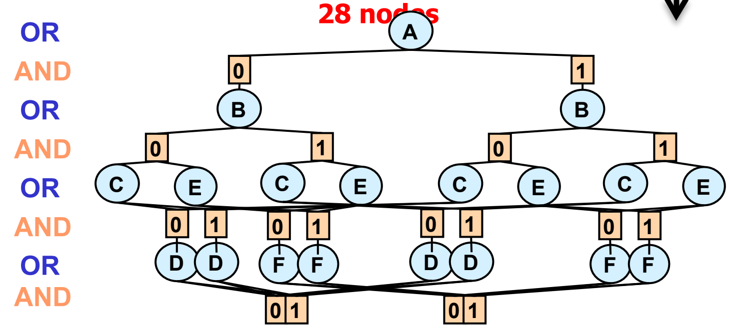
Context minimal OR search graph

28 nodes



Full AND/OR search tree

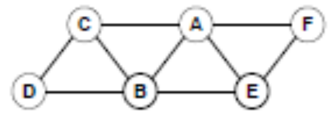
54 AND nodes



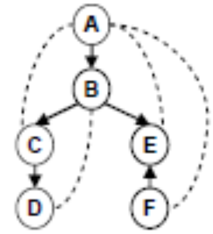
Context minimal AND/OR search graph

18 AND nodes

AND/OR Search Tree for Constraint Networks



(a) Constraint graph



(b) Pseudo tree

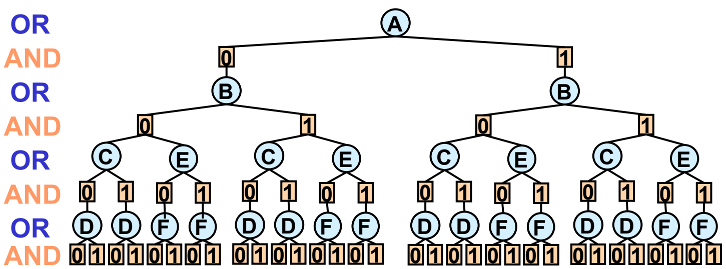
A	B	C	R_{ABC}
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

B	C	D	R_{BCD}
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

A	B	E	R_{ABE}
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

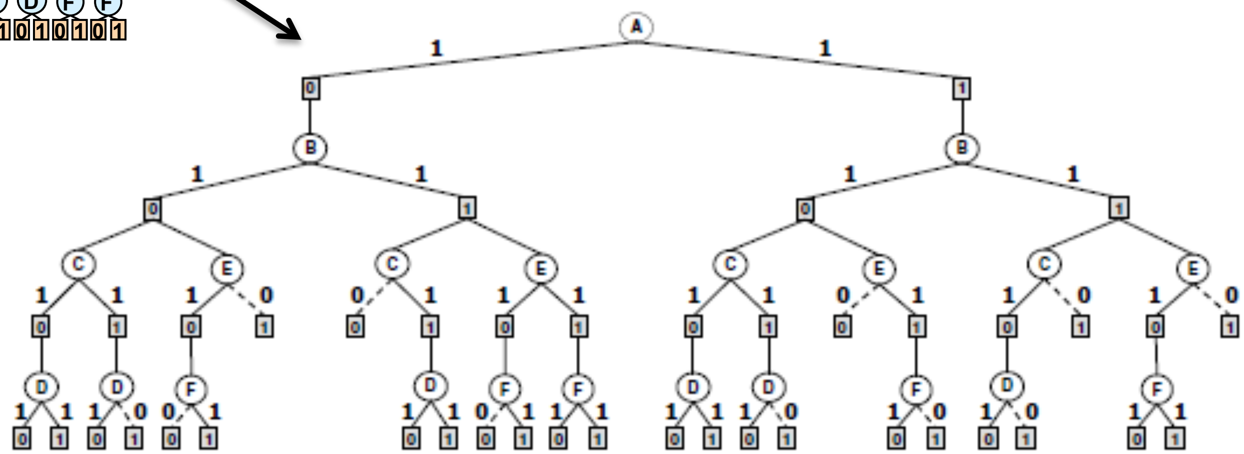
A	E	F	R_{AEF}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(c) Relations



Full AND/OR search tree

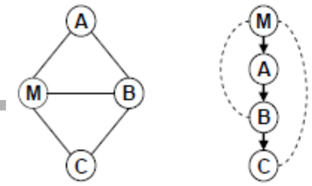
54 AND nodes



(d) AND/OR tree

Taking the constraints into account

Weighted AND/OR Search Tree and Context Minimal Graph for Cost Networks



M	A	B	f(M,A,B)
0	0	0	12
0	0	1	5
0	1	0	18
0	1	1	2
1	0	0	4
1	0	1	10
1	1	0	6
1	1	1	4

M	B	C	g(M,B,C)
0	0	0	3
0	0	1	5
0	1	0	14
0	1	1	12
1	0	0	9
1	0	1	15
1	1	0	7
1	1	1	6

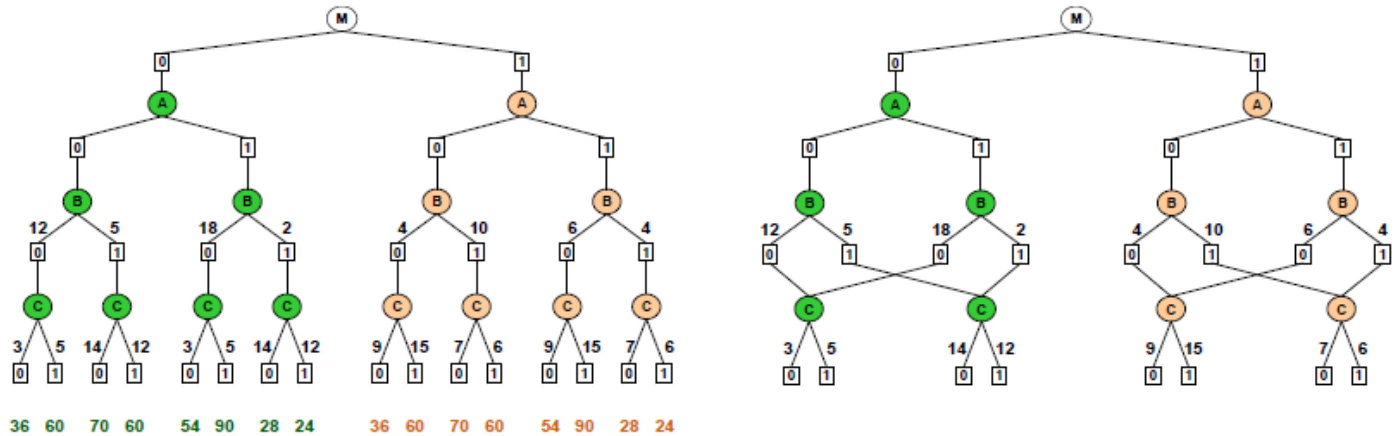


Figure 20: AND/OR search tree and context minimal graph

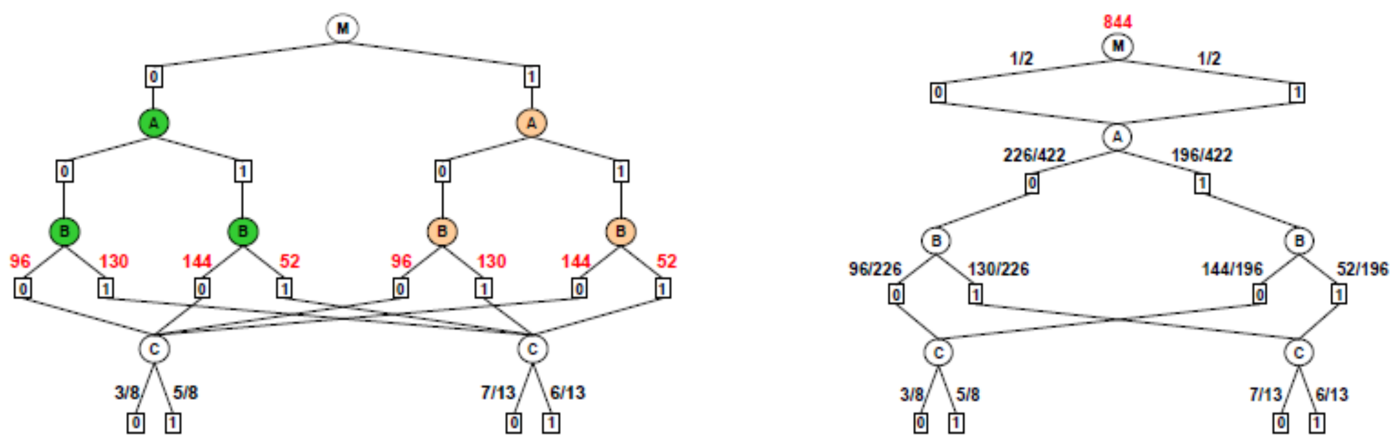


Figure 22: AOMDD for the weighted graph

Three Treewidths

- Treewidth: tw
- Semantic Treewidth: stw
- Semantic AOMDD width: AO-w

Outline

- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
- Semantic Width
- Empirical demonstration
- Summary

Outline

- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
- Semantic Width
- Empirical demonstration
- Learning AOMDDs

Constraint Networks

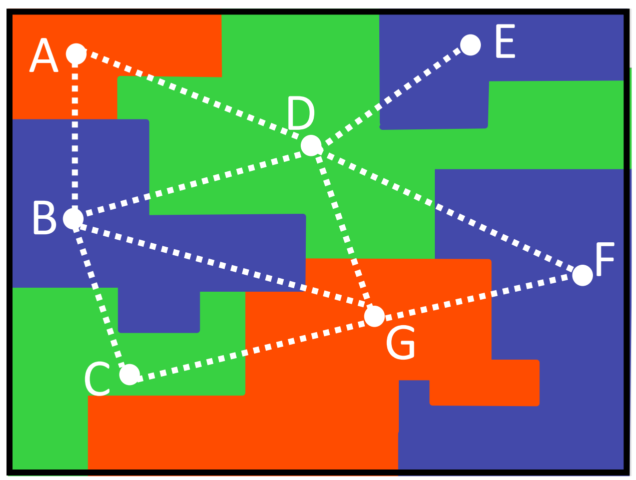
Map coloring

Variables: countries (A B C etc.)

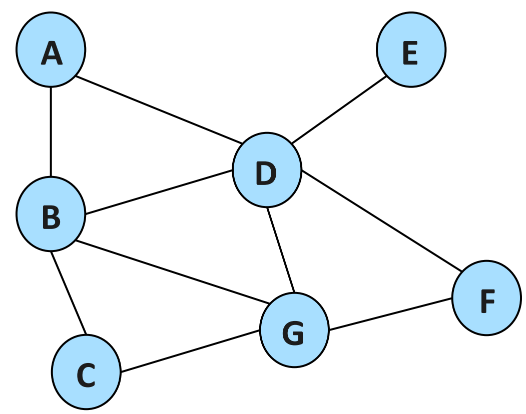
Values: colors (red green blue)

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

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

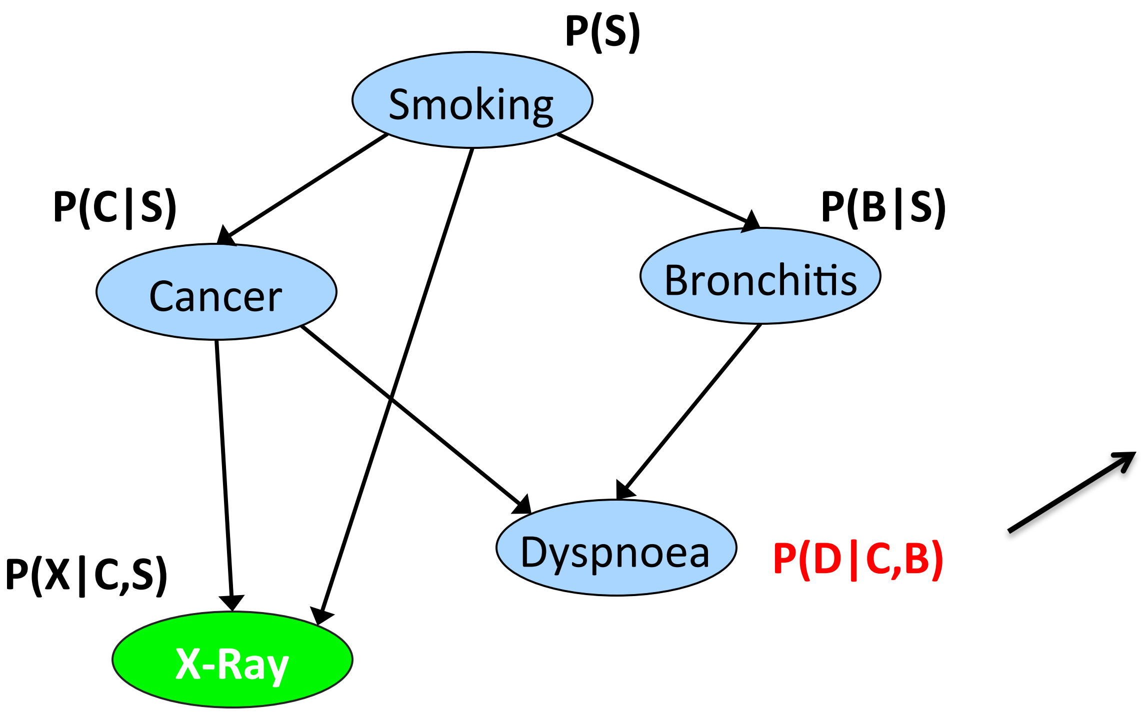


Constraint graph



Bayesian Networks

BN = (X,D,G,P)



P(D|C,B)

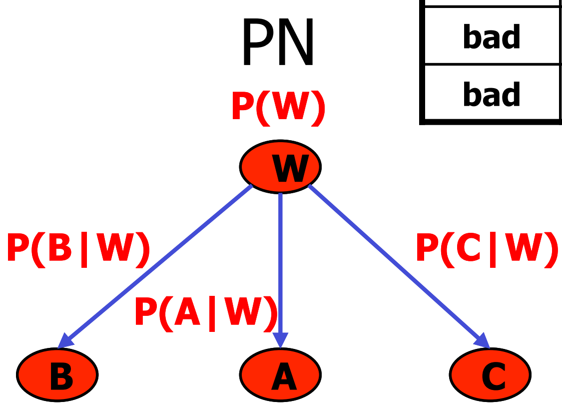
C	B	D=0	D=1
0	0	0.1	0.9
0	1	0.7	0.3
1	0	0.8	0.2
1	1	0.9	0.1

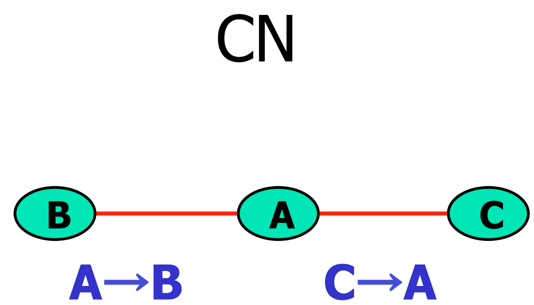
$$P(S,C,B,X,D) = P(S)\cdot P(C|S)\cdot P(B|S)\cdot P(X|C,S)\cdot P(D|C,B)$$

A counting query: probability of evidence
Optimization query: MPE

Mixed Probabilistic and Deterministic networks

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





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

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

Graphical Models

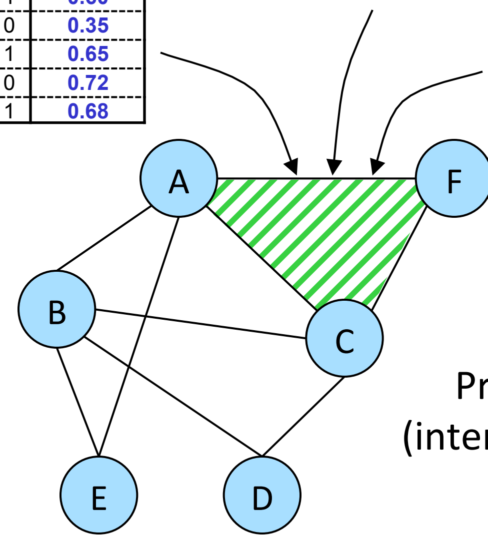
- A graphical model **(X,D,F)**:
  - **X** = $\{X_1,\ldots X_n\}$ variables
  - **D** = $\{D_1, \ldots D_n\}$ domains
  - **F** = $\{f_1,\ldots,f_m\}$ functions

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)

- Tasks:
  - **Belief updating**: $\Sigma_{X-y} \prod_j P_i$
  - **MPE**: $\max_X \prod_j P_j$
  - **CSP**: $\prod_X \times_j C_j$
  - **Max-CSP**: $\min_X \Sigma_j f_j$

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

Sample Applications for Graphical Models

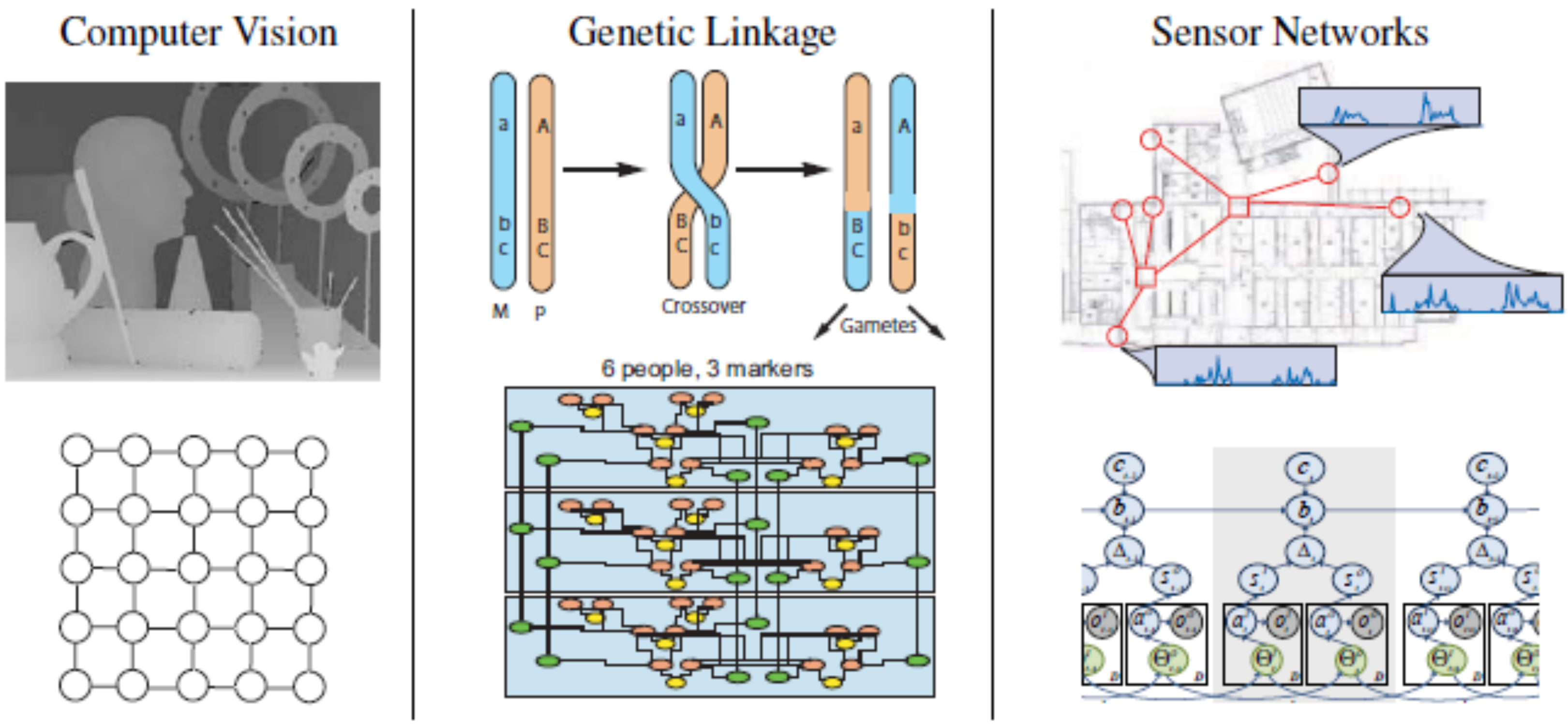
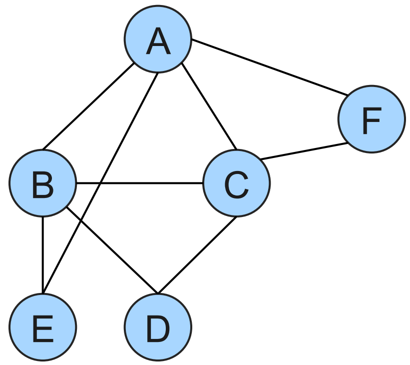


Figure 1: Application areas and graphical models used to represent their respective systems: (a) Finding correspondences between images, including depth estimation from stereo; (b) Genetic linkage analysis and pedigree data; (c) Understanding patterns of behavior in sensor measurements using spatio-temporal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

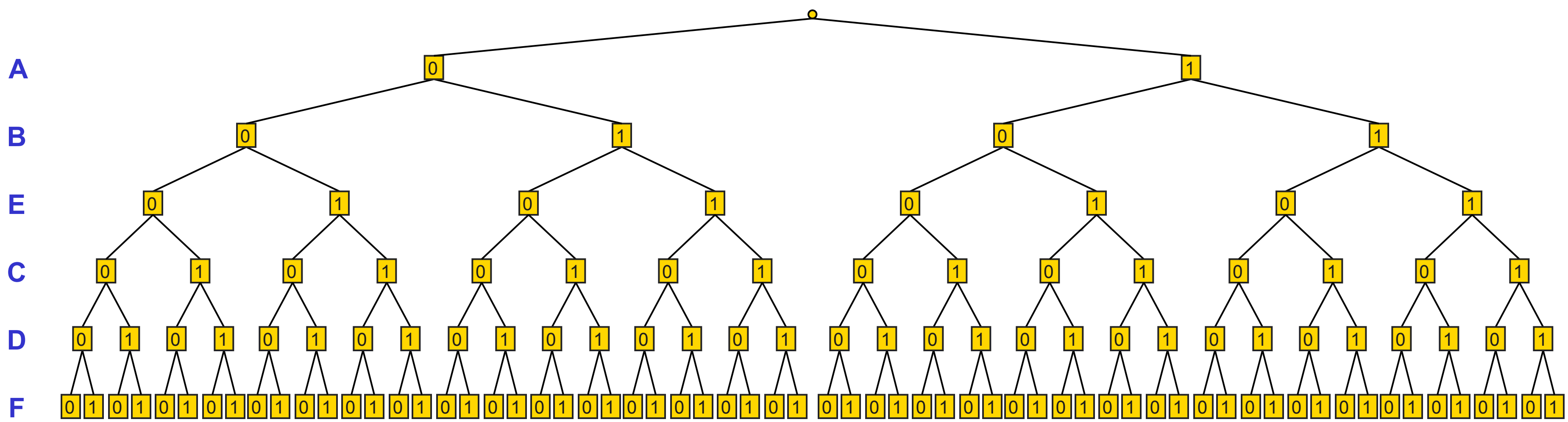
Outline

- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
- Semantic Width
- Empirical demonstration
- Learning AOMDDs

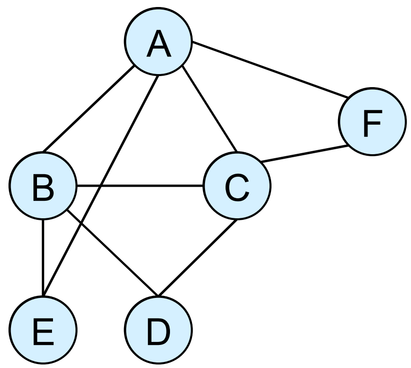
Classic OR Search Space



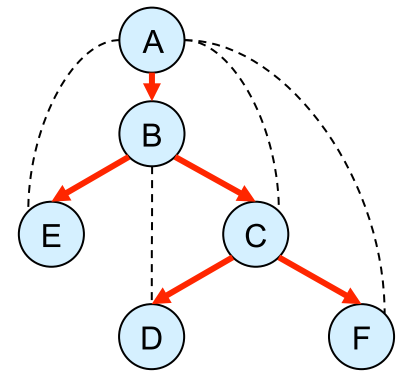
Ordering: A B E C D F



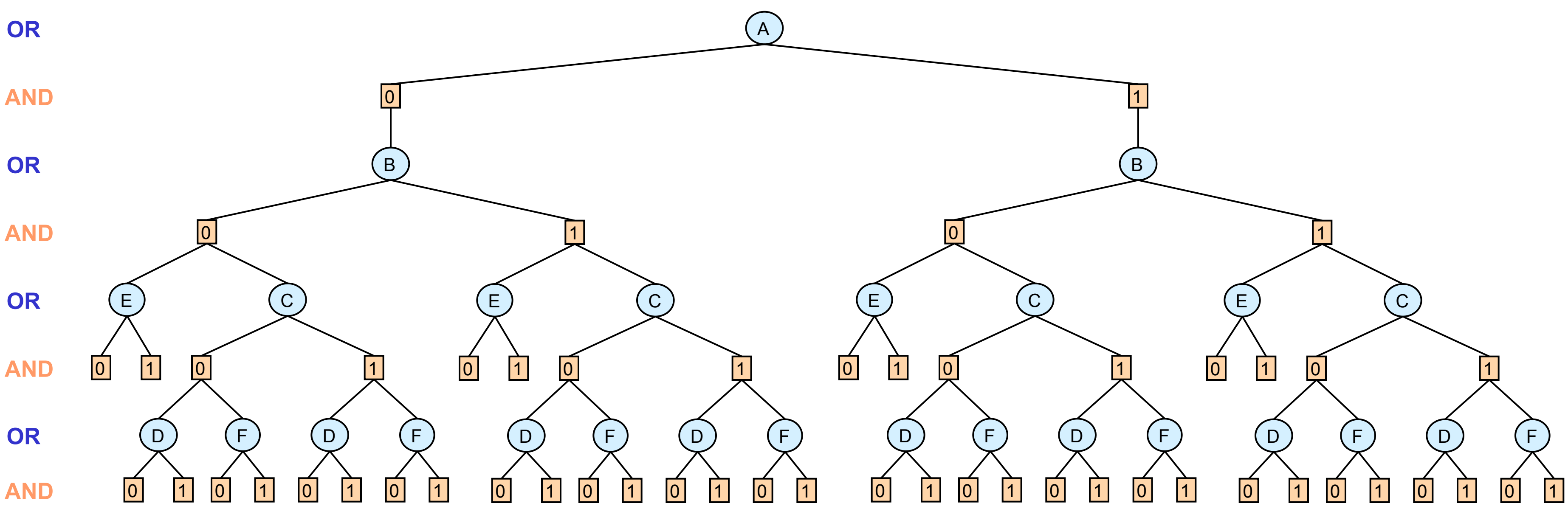
AND/OR Search Space



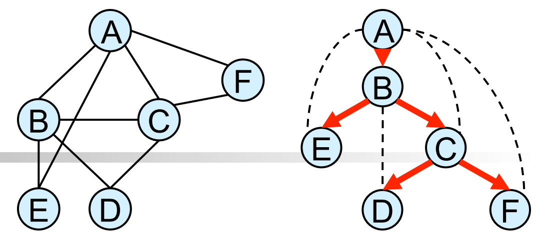
Primal graph

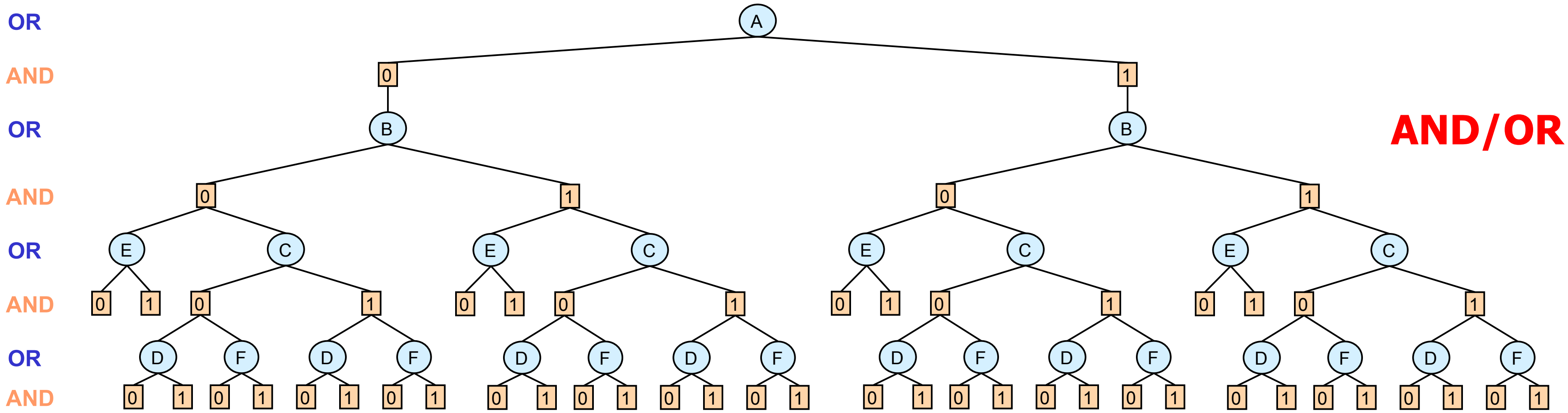


DFS tree

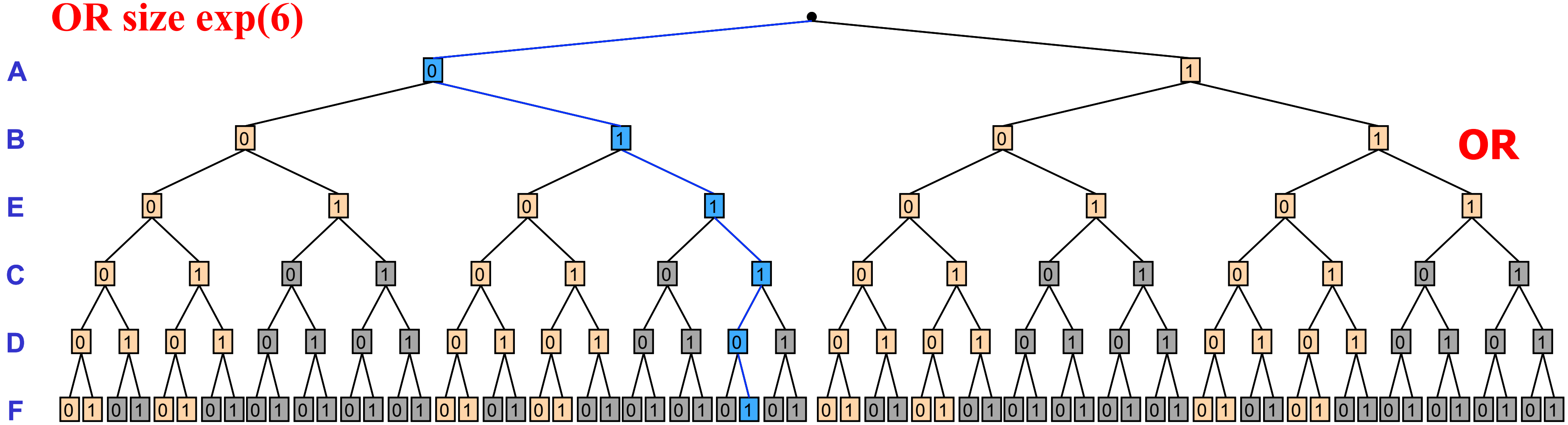


AND/OR vs. OR



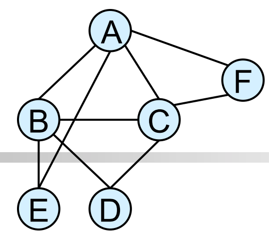


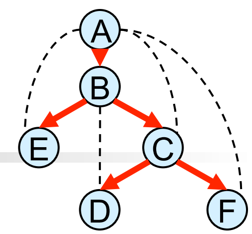
AND/OR size: exp(4),
OR size exp(6)

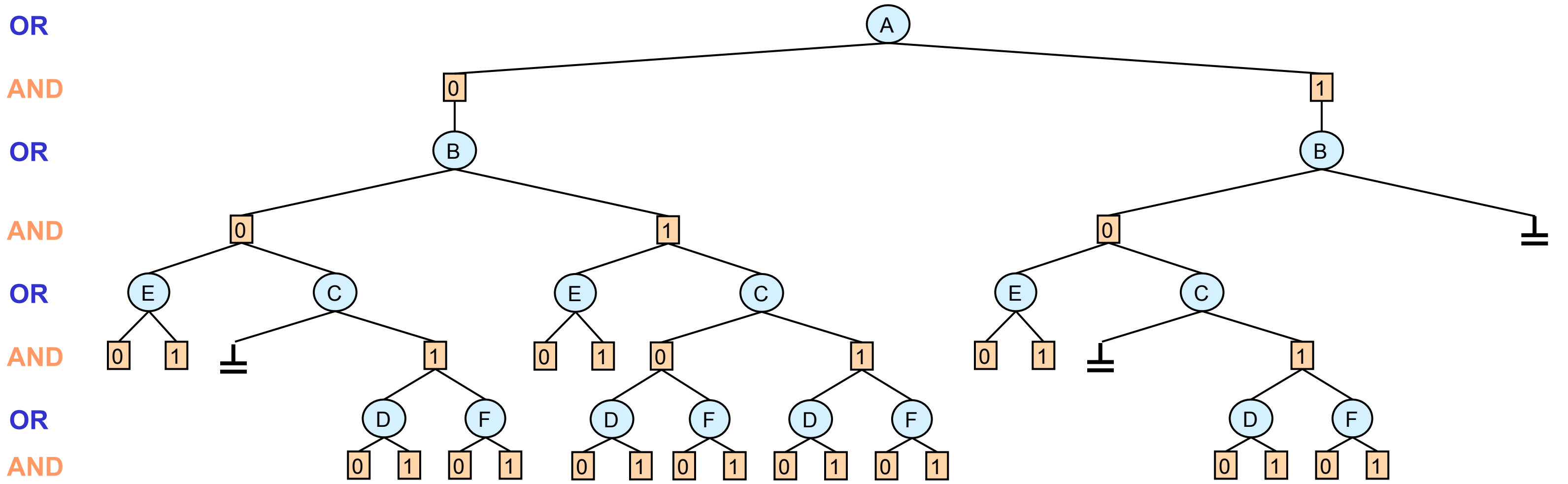


AND/OR vs. OR with Constraints

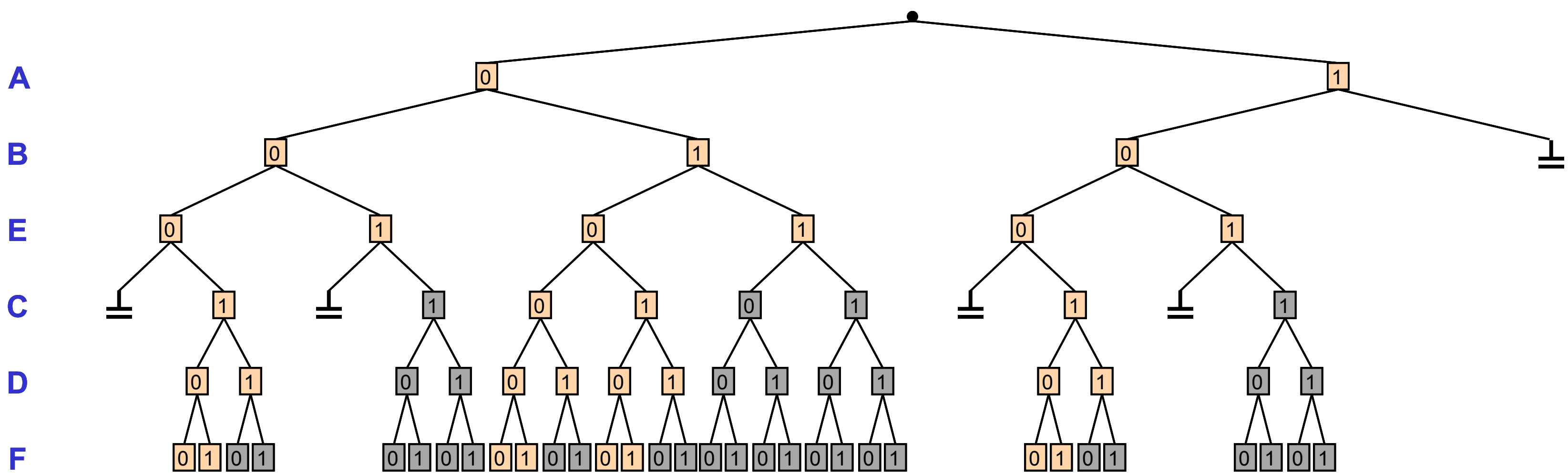
No-goods
(A=1,B=1)
(B=0,C=0)





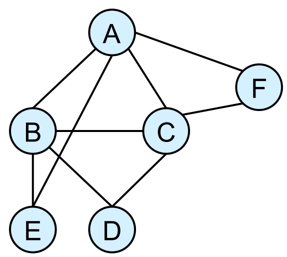


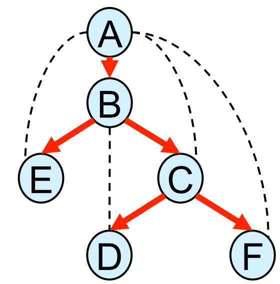
AND/OR

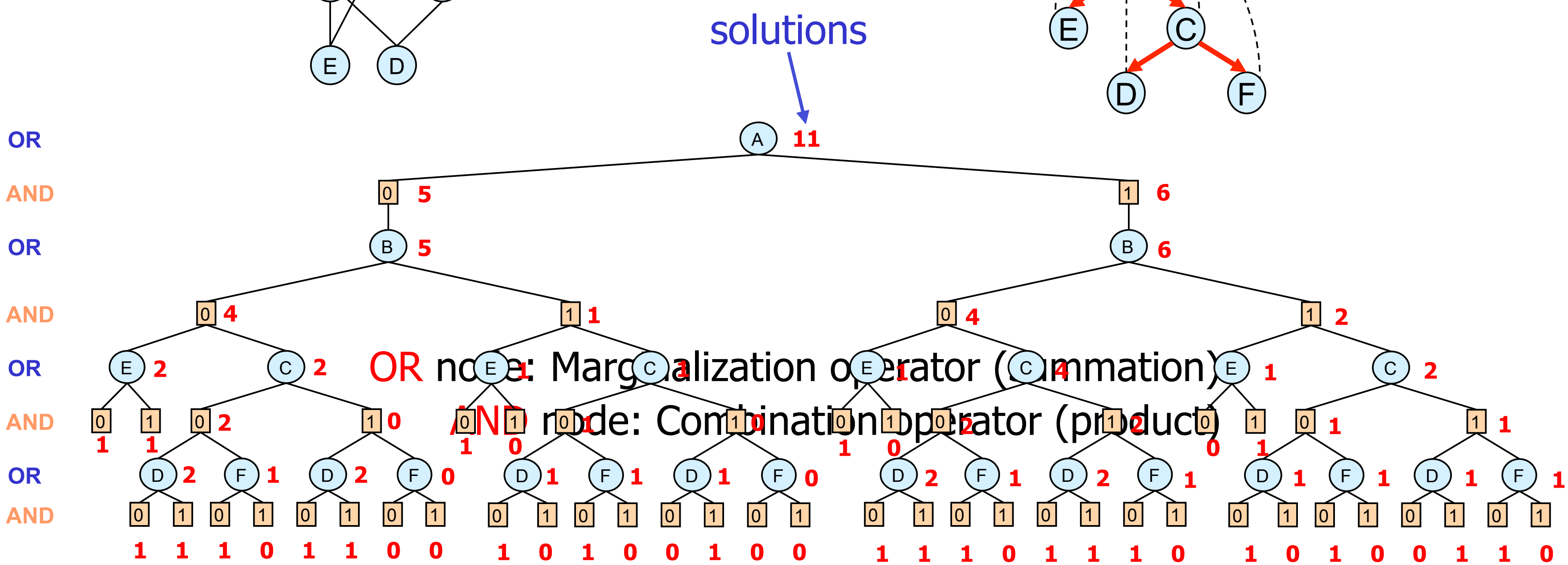


OR

Counting Solutions by DFS traversal (Sum-Product Networks)





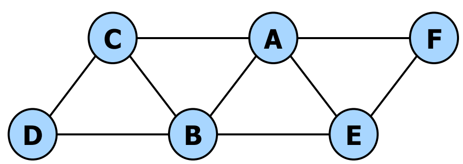


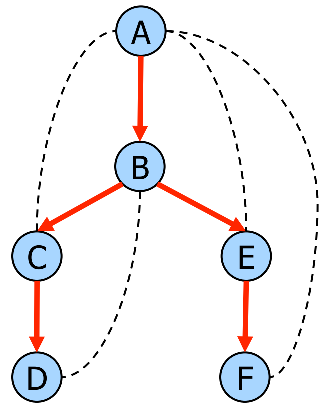
OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = number of solutions below it

Weighted AND/OR Search Tree for a Cost Network





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

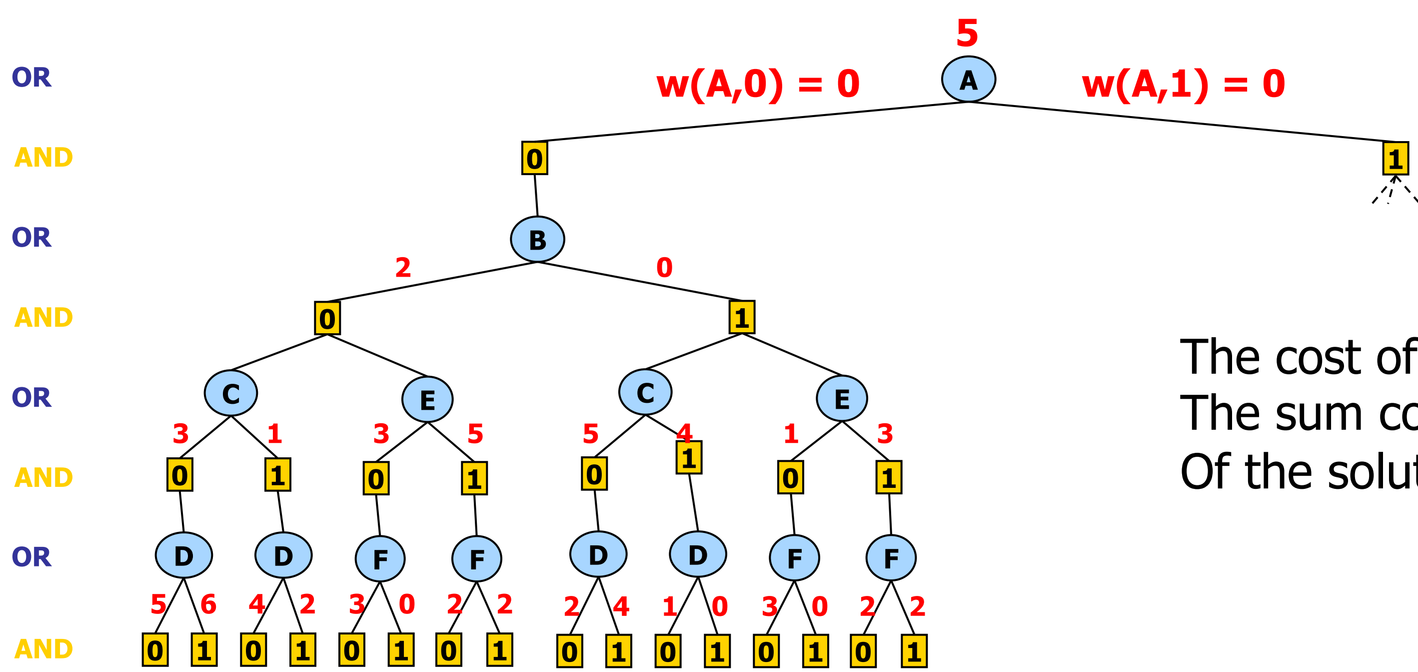
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

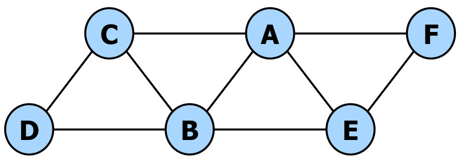
E	F	f_9
0	0	1
0	1	0
1	0	0
1	1	2

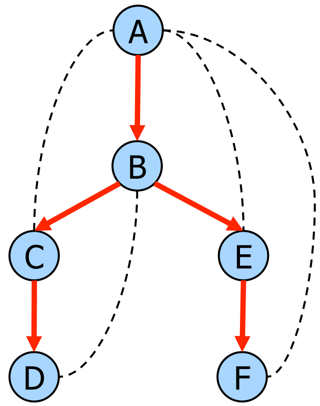
$$f(X) = \sum_{i=1}^{9} f_i(X)$$



The cost of a solution is
The sum cost of weights
Of the solution tree

Optimizing over Weighted AND/OR Tree for a Cost Network





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

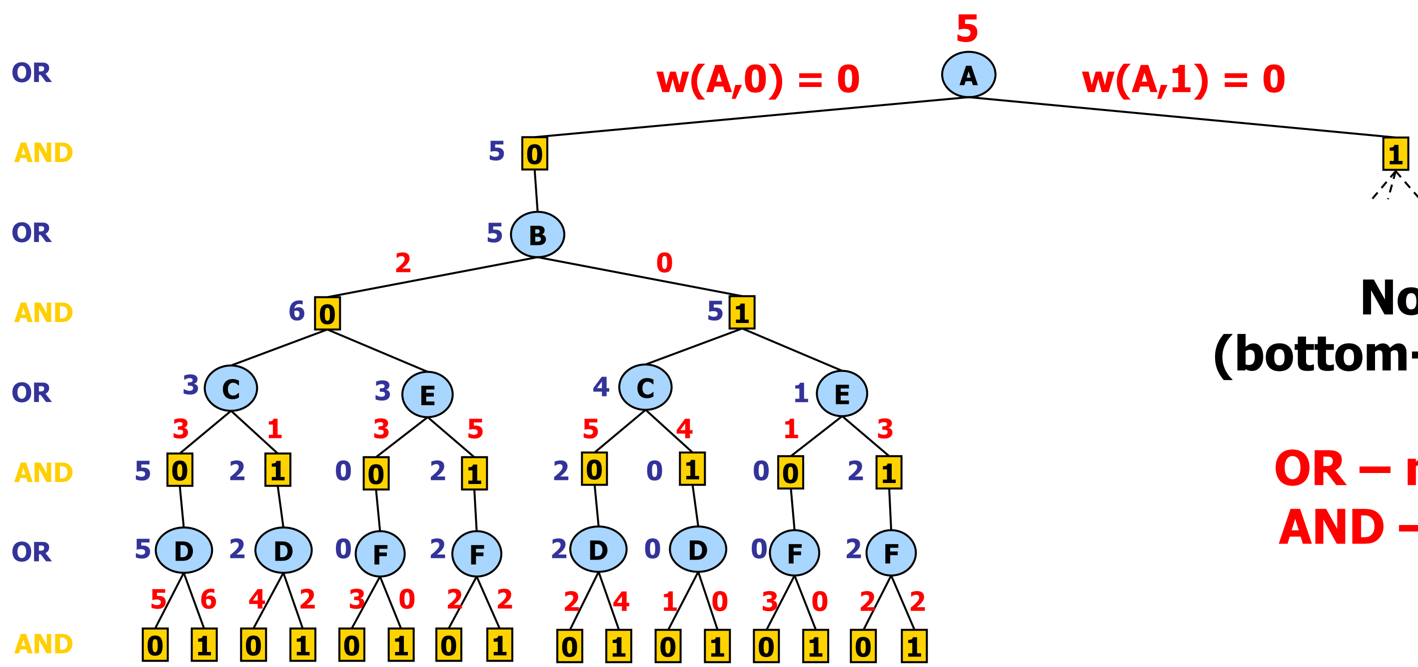
B	D	f_6
0	0	4
0	1	2
1	0	1
1	1	0

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

$$f(X) = \sum_{i=1}^{9} f_i(X)$$



Node Value (bottom-up evaluation)

OR – minimization
AND – summation

Weighted AND/OR Tree for Bayesian Network

$P(E \mid A,B)$

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

Evidence: E=0

$P(B \mid A)$

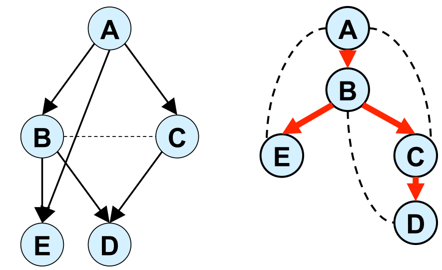
A	B=0	B=1
0	.4	.6
1	.1	.9

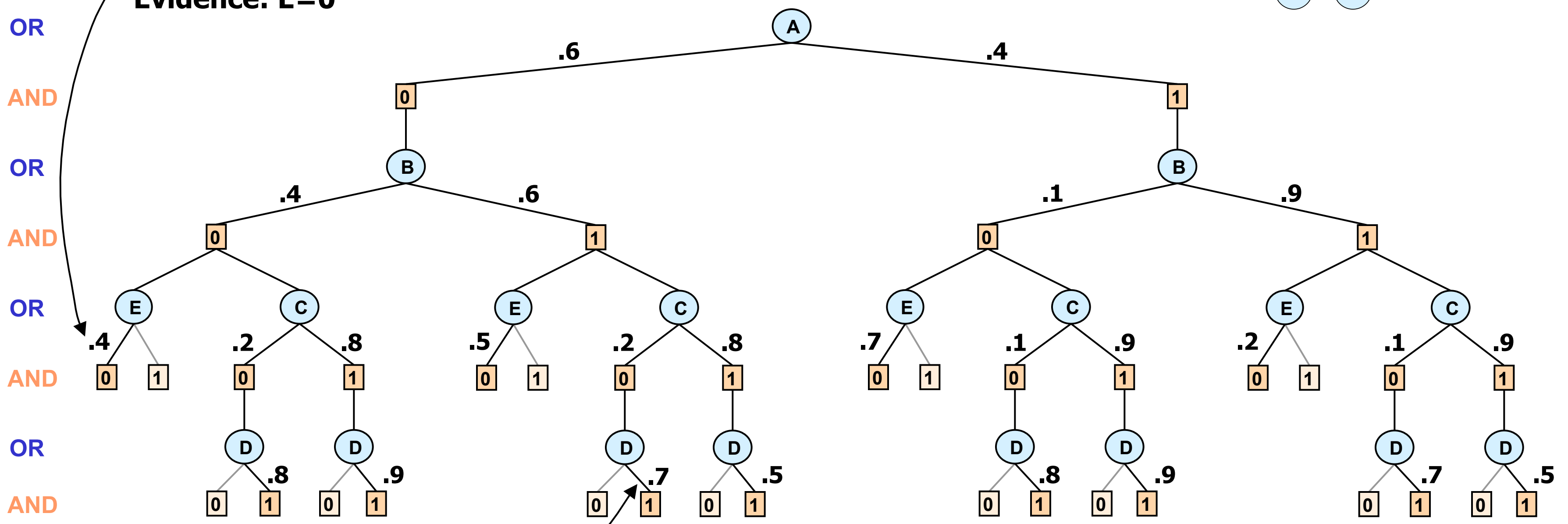
$P(C \mid A)$

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

$P(A)$

A	P(A)
0	.6
1	.4





$P(D \mid B,C)$

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

Evidence: D=1

Weighted AND/OR Tree for Bayesian Network (Sum-Product Networks)

$P(E \mid A,B)$

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

$P(B \mid A)$

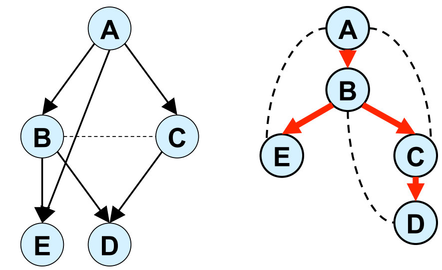
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

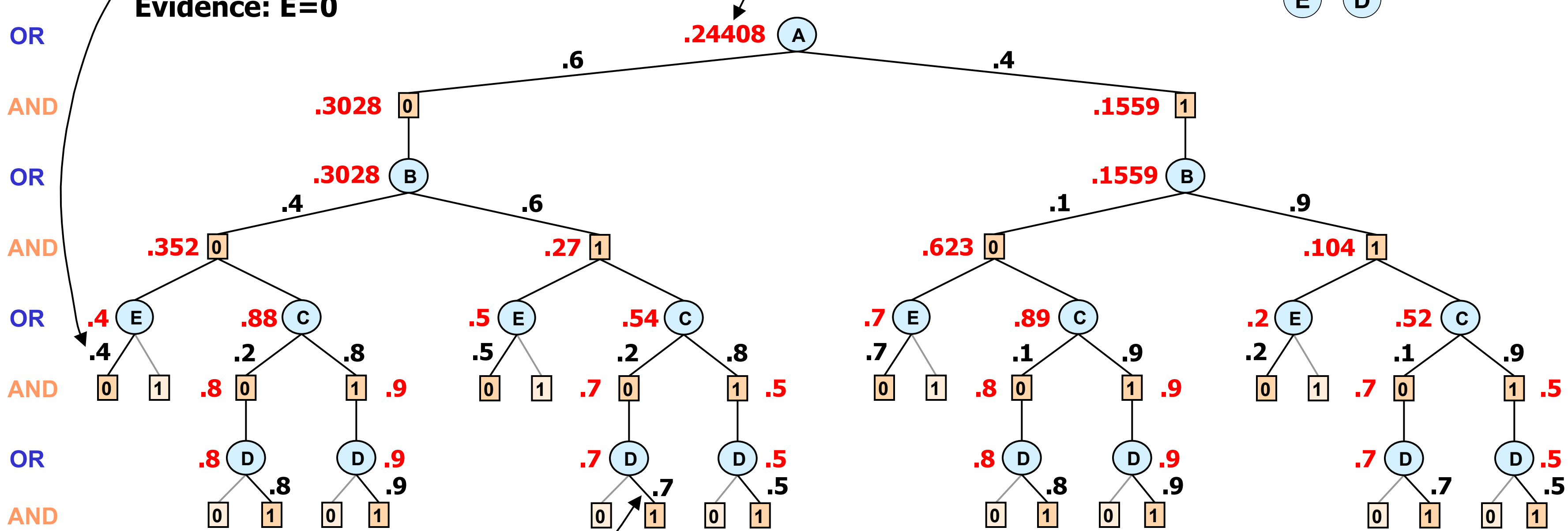
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

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

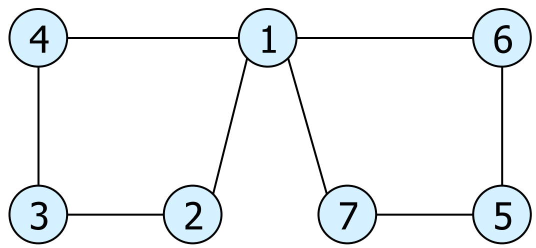
OR node: Marginalization operator (summation)

AND node: Combination operator (product)

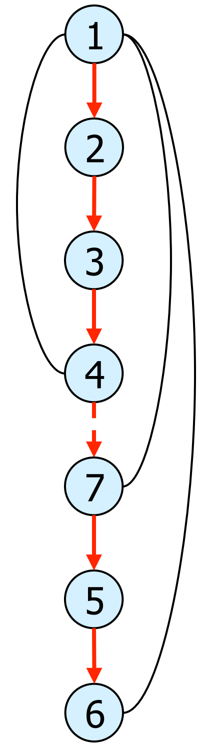
Value of node = updated belief for sub-problem below

Pseudo-Trees

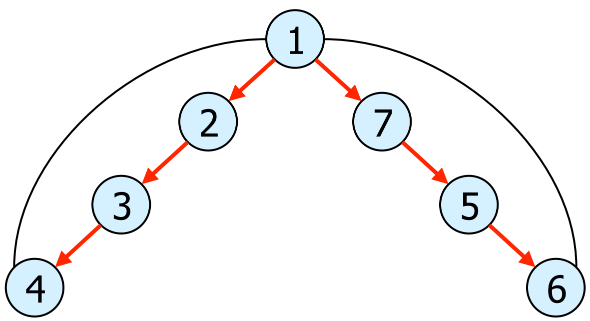
(Freuder 85, Bayardo 95, Bodlaender and Gilbert, 91)

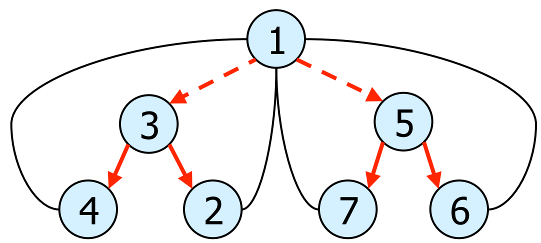


(a) Graph



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

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

Tasks and value of nodes

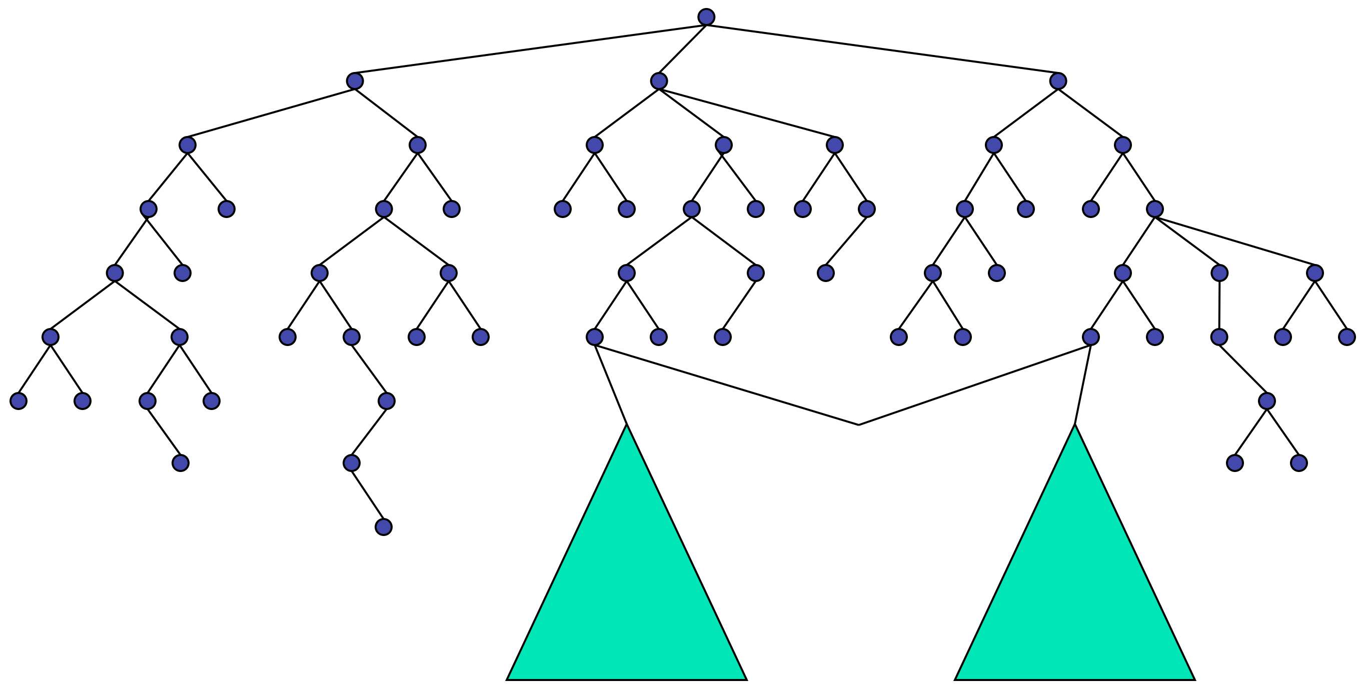
- **V(n) is the value of the tree T(n) for the task:**
  - Counting: v(n) is number of solutions in T(n)
  - Consistency: v(n) is 0 if T(n) inconsistent, 1 othewise.
  - **Optimization: v(n) is the optimal solution in T(n)**
  - **Belief updating: v(n), probability of evidence in T(n).**
  - **Partition function: v(n) is the total probability in T(n).**

- **Goal: compute the value of the root node recursively using dfs search of the AND/OR tree.**

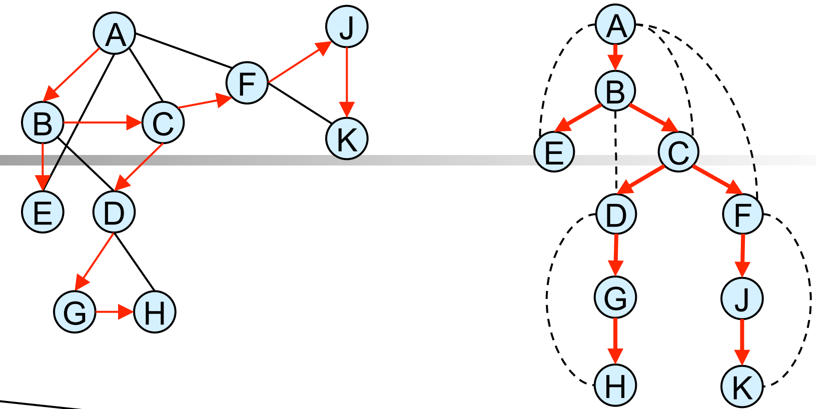
- **Theorem: Complexity of AO dfs search is**
  - **Space: O(n)**
  - **Time: $O(n\ k^m)$**
  - **Time: O(exp(w* log n))**

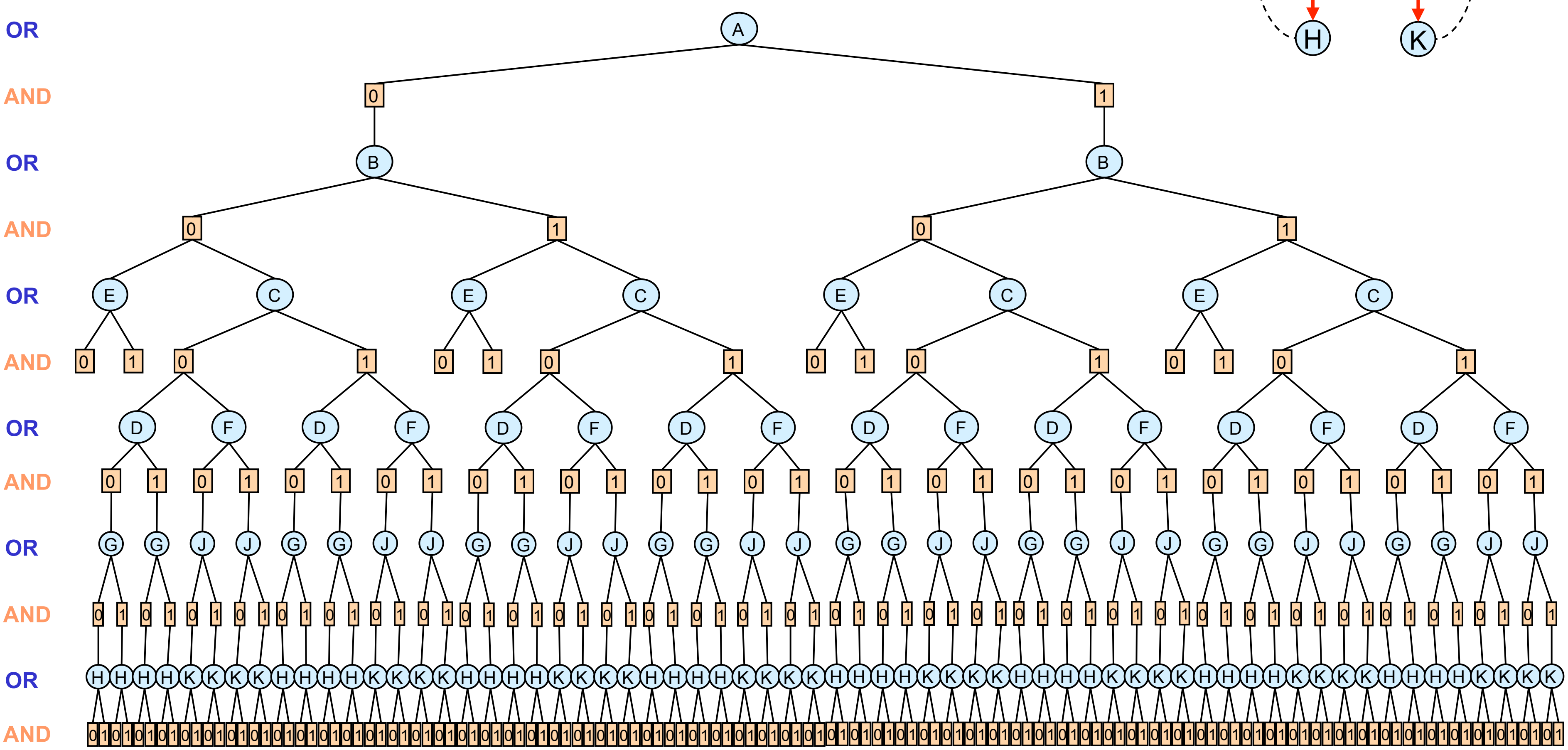
From Search Trees to Search Graphs

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

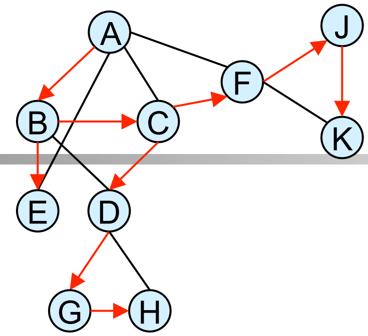


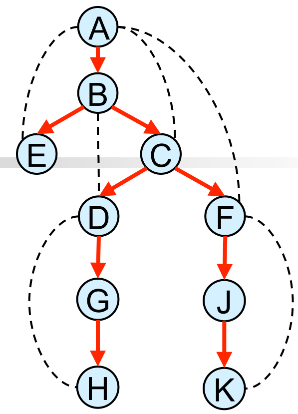
From AND/OR Tree

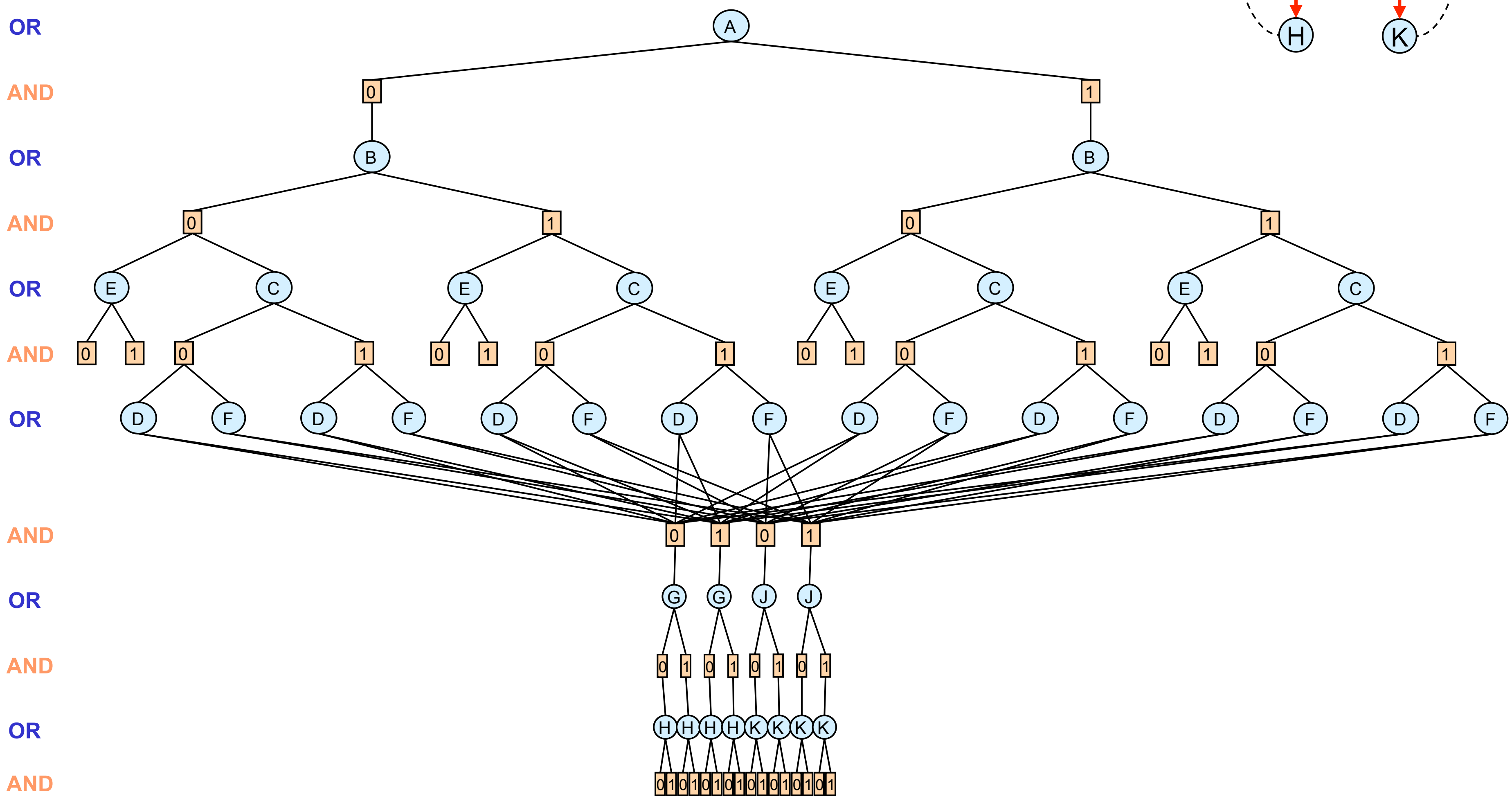




To an AND/OR Graph







A Bayesian Network AND/OR Search Tree

$P(E \mid A,B)$

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

Evidence: E=0

$P(B \mid A)$

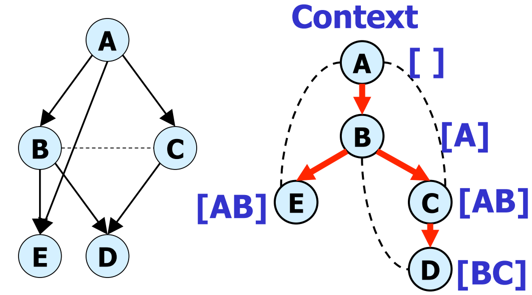
A	B=0	B=1
0	.4	.6
1	.1	.9

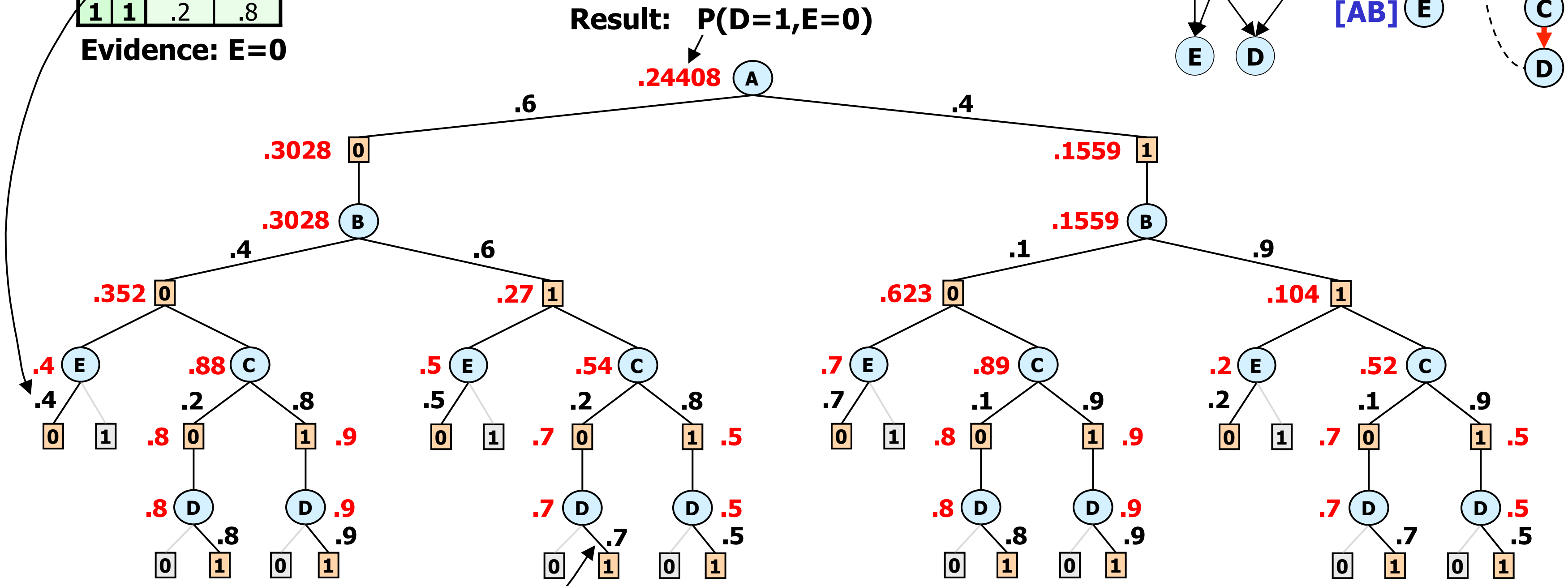
$P(C \mid A)$

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

$P(A)$

A	P(A)
0	.6
1	.4





$P(D \mid B,C)$

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

Evidence: D=1

OR node: Marginalization operator (summation)

AND node: Combination operator (product)

Value of node = updated belief for sub-problem below

AND/OR Graph DFS Algorithm

(Belief Updating)

$P(E \mid A,B)$

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

Evidence: E=0

$P(B \mid A)$

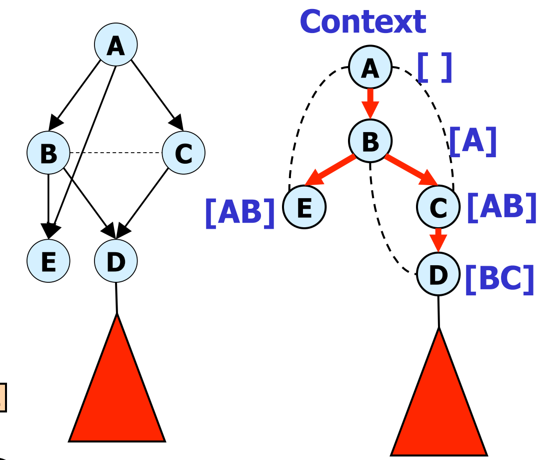
A	B=0	B=1
0	.4	.6
1	.1	.9

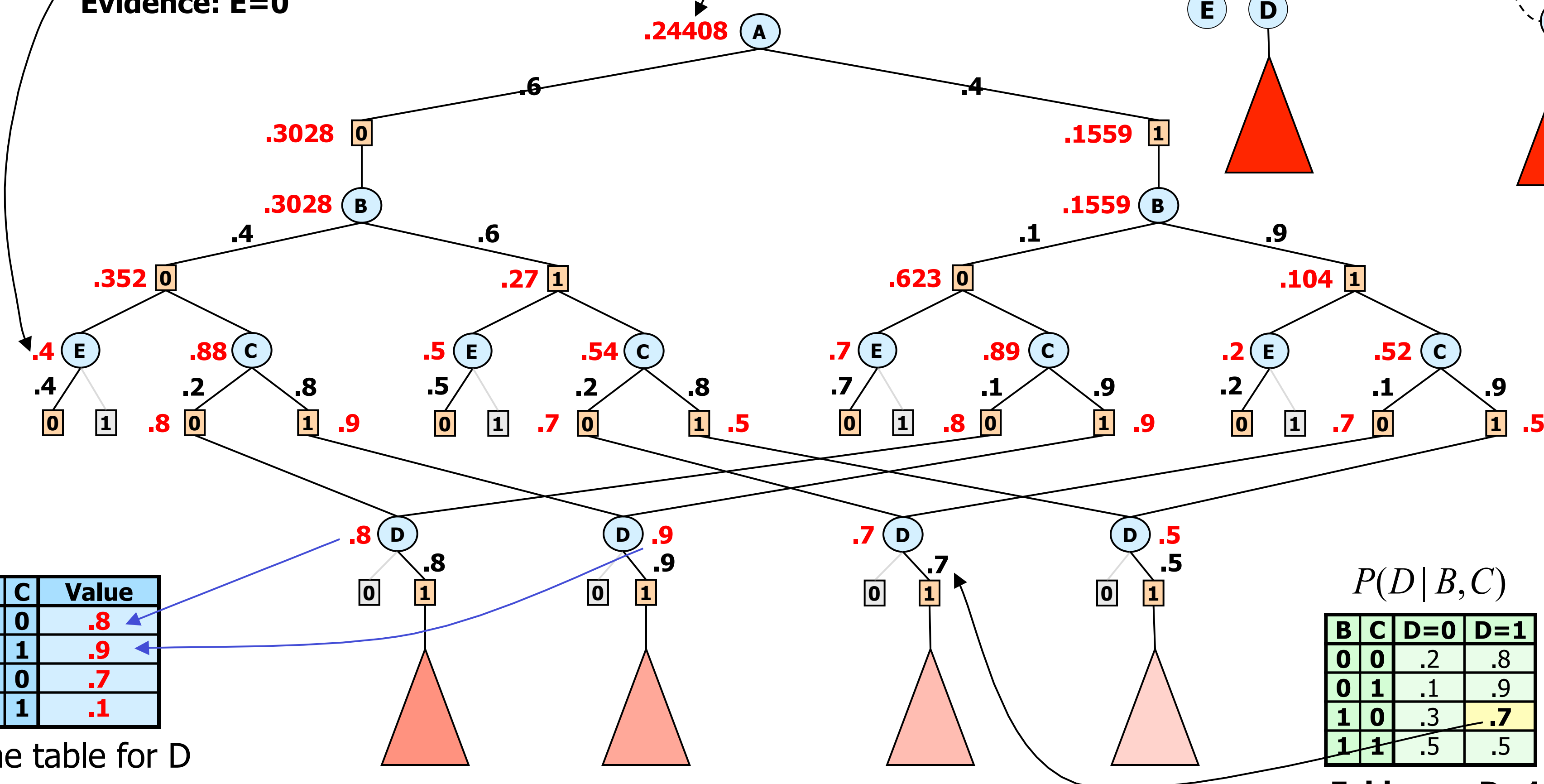
$P(C \mid A)$

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

$P(A)$

A	P(A)
0	.6
1	.4





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

Cache table for D

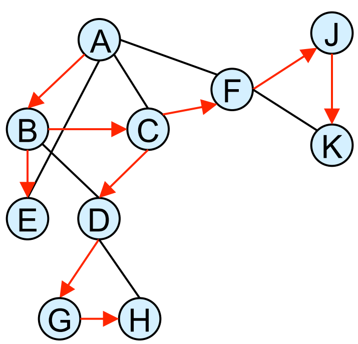
$P(D \mid B,C)$

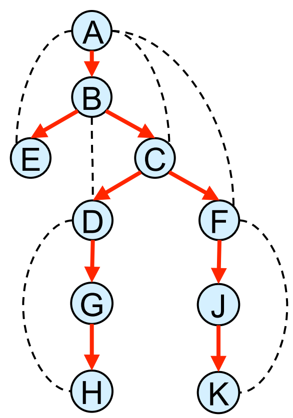
B	C	D=0	D=1
0	0	.2	.8
0	1	.1	.9
1	0	.3	.7
1	1	.5	.5

Evidence: D=1

AND/OR Context Minimal Graph

- Caching is possible when context is the same
- context = parent-separator set in induced pseudo-graph
  = current variable +
  ancestors connected to subtree below





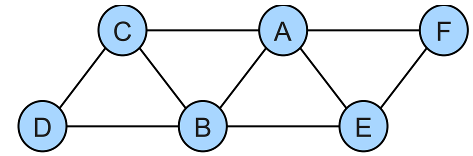
context(B) = {A, B}

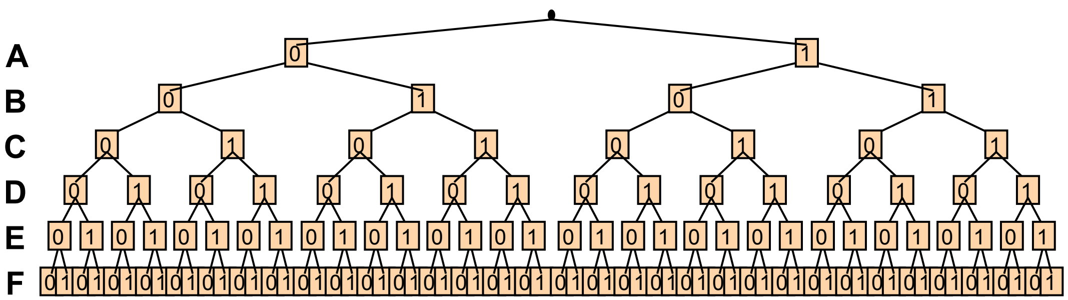
context(C) = {A,B,C}

context(D) = {D}

context(F) = {F}

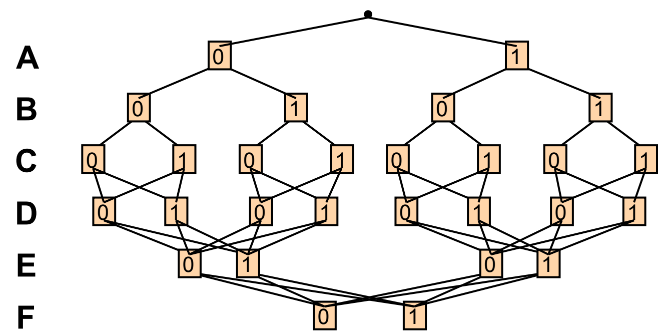
All Four Search Spaces





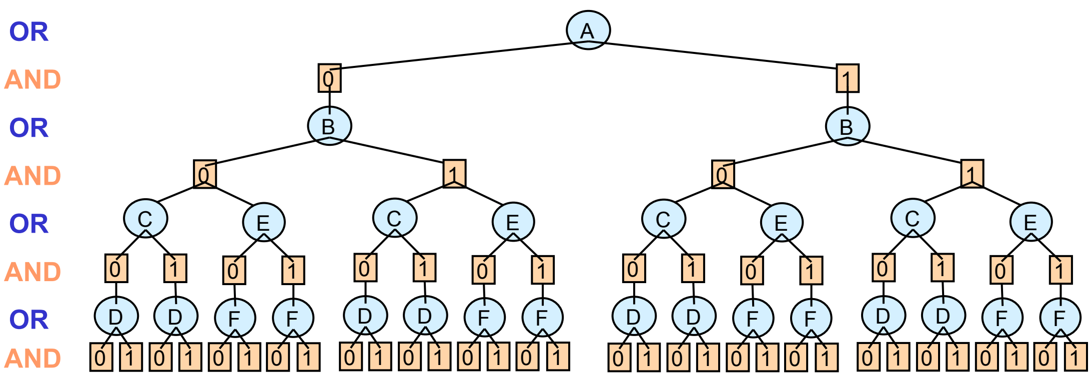
Full OR search tree

126 nodes



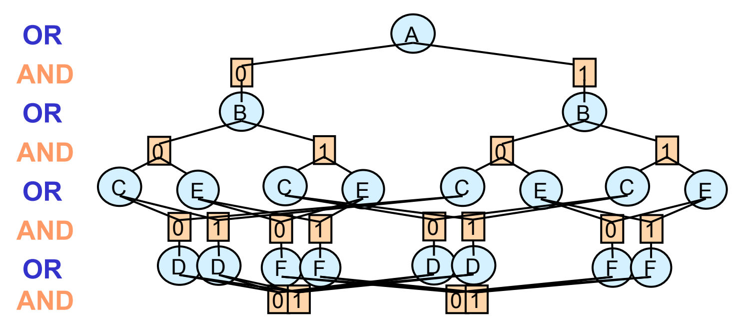
Context minimal OR search graph

28 nodes



Full AND/OR search tree

54 AND nodes

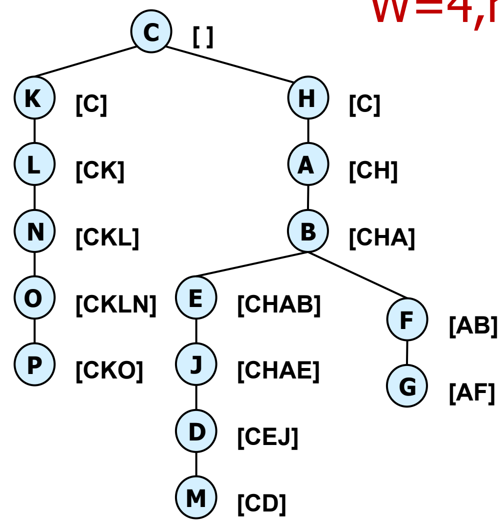


Context minimal AND/OR search graph

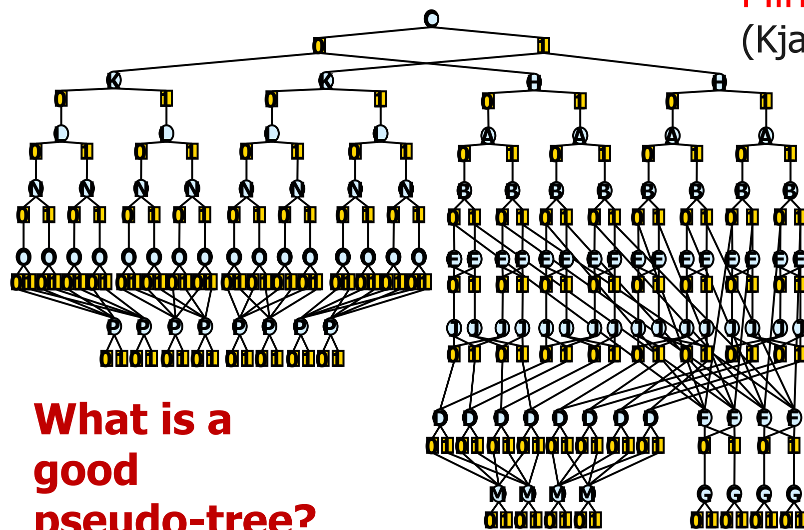
18 AND nodes

Two AND/OR Context-Minimal Graphs

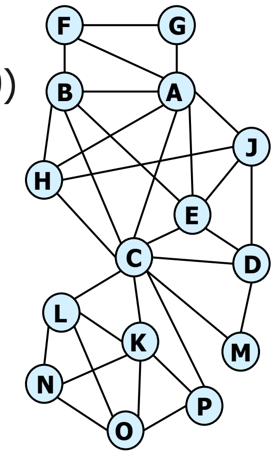
W=4,h=8



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



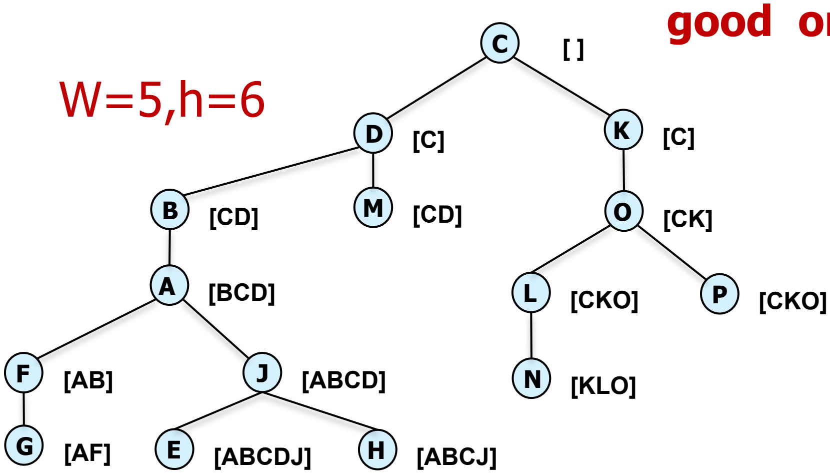
Min-Fill
(Kjaerulff90)

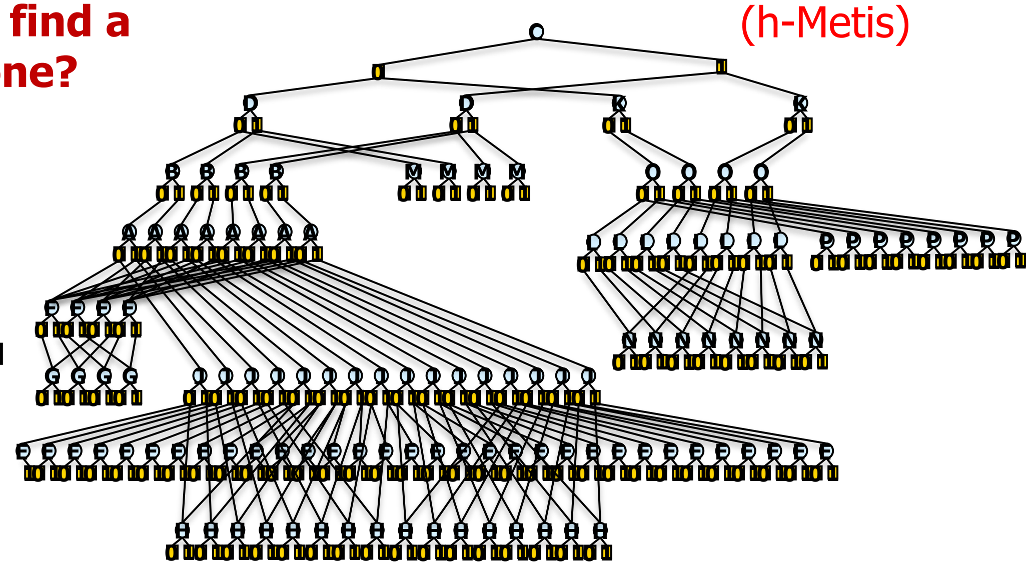


What is a good pseudo-tree? How to find a good one?

Hypergraph Partitioning
(h-Metis)

W=5,h=6





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

Complexity of AND/OR Graph Search

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

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

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

Queries; Satisfiability, optimization, counting

AOBB+MBE(i): won PASCAL competition 2011

- AND/OR Branch and Bound searching the context-minimal search space using the mini-bucket heuristics, improved by soft-consistency…
- **Placed 1st in all three MPE tracks.**

- Baseline: AND/OR Branch-and-Bound with mini-bucket heuristic .
  - 3rd place for MPE at UAI 2010 Evaluation.

- Source code available under GPL:
  - `http://github.com/lotten/daoopt`

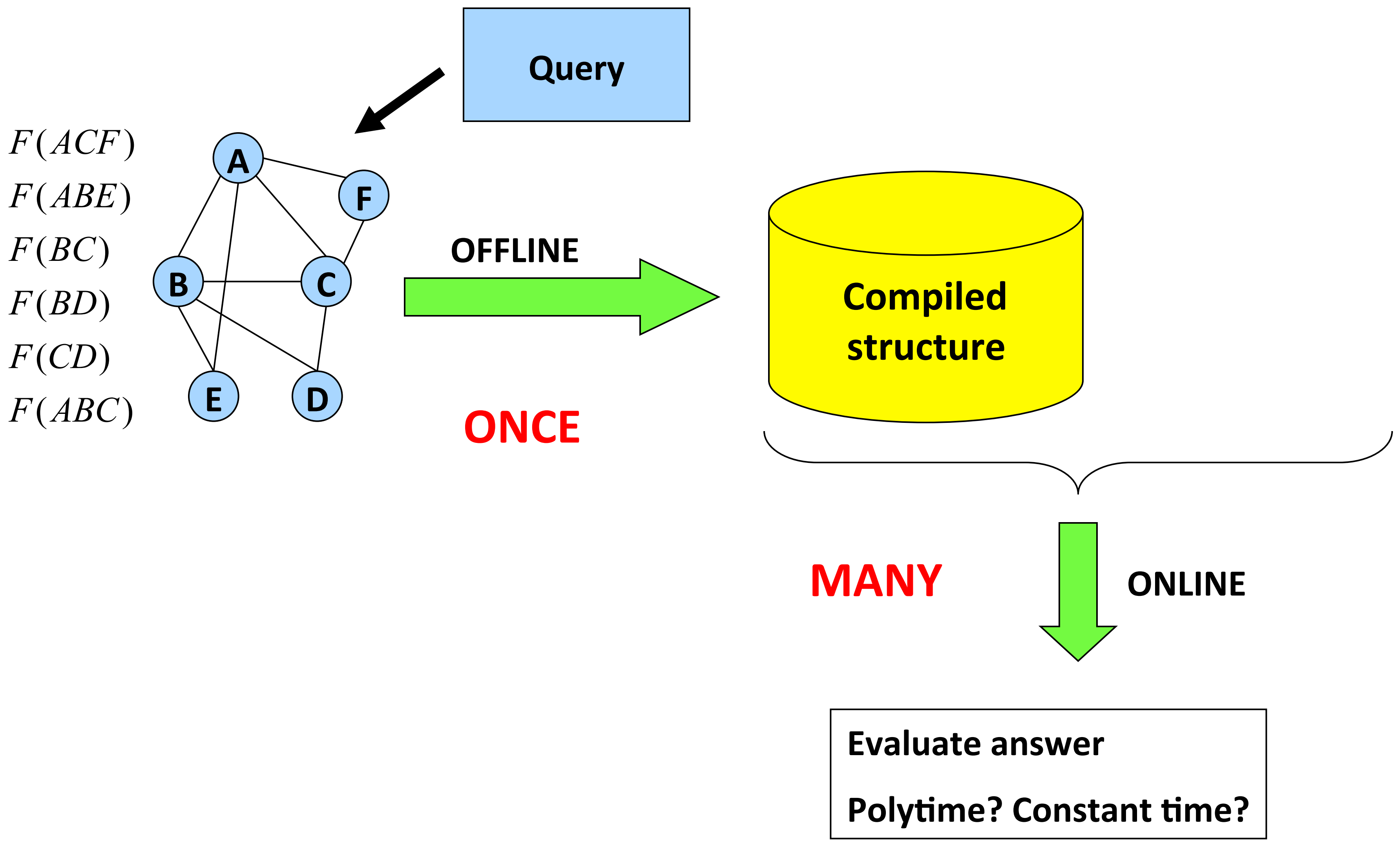
From Context-Minimal to Minimal AND/ORs

- Any two nodes that root identical subtrees/subgraphs (are unifiable) can be merged
- Minimal AND/OR search graph: closure under merge of its AND/OR search tree, where inconsistent subtrees are pruned.
- Canonicity: The minimal AND/OR search graph AOMDD is unique (canonical) for all equivalent formulas (Boolean or Constraints, or weighted GM), consistent with its pseudo tree.
- AOMDD: AND/OR Multi-valued Decision Diagrams are minimal AND/OR search graph representation
- Complexity: Minimal AND/OR GM for T is exponential in the treewidth along T.

Outline

- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
- Semantic Width
- Empirical demonstration
- Learning AOMDDs

Compilation of Graphical Models

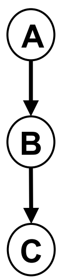


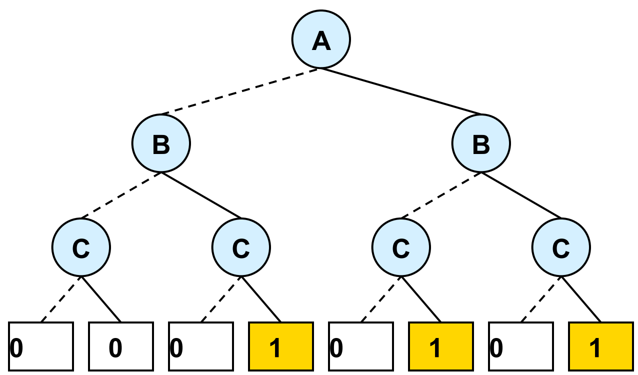
Ordered Binary Decision Diagram

B = {0,1}   f : B^3 → B

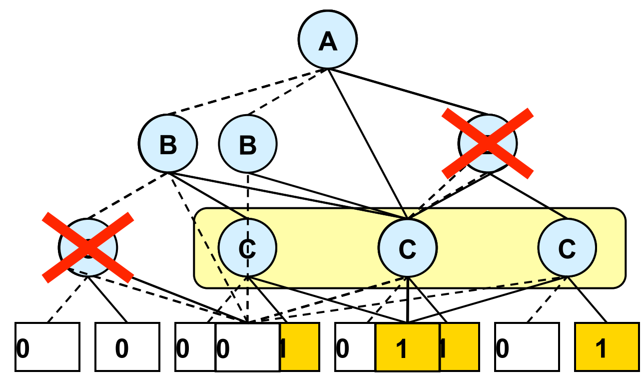
A	B	C	f(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table





Decision tree



1) Merge nodes with identical children

[Bryant86]

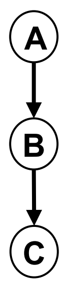
2) Remove redundant nodes

Ordering enables efficient operations

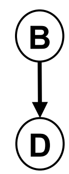
AND/OR CM Graph vs. AOMDD

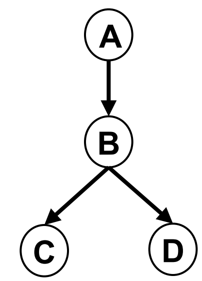
For a constraint network

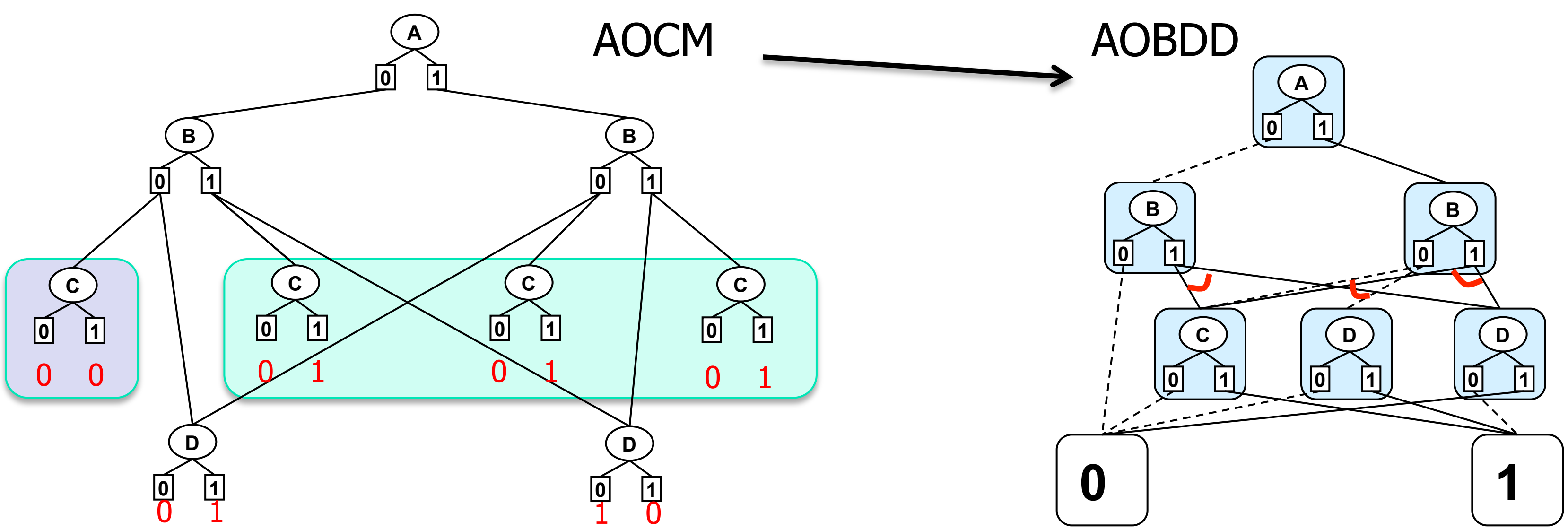
A	B	C	f(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



B	D	G(BD)
0	0	0
0	1	1
1	0	1
1	1	0



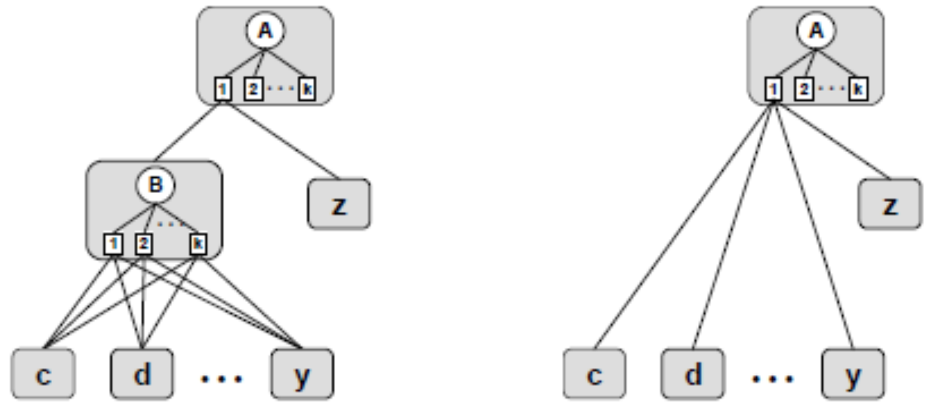




AND/OR Multi-Valued Decision Diagrams

- AOMDDs are:
  - Weighted AND/OR search graphs
  - Canonical representations, given a pseudo tree
  - Defined by two rules:
    - All isomorphic subgraphs are merged
    - There are no redundant (meta) nodes

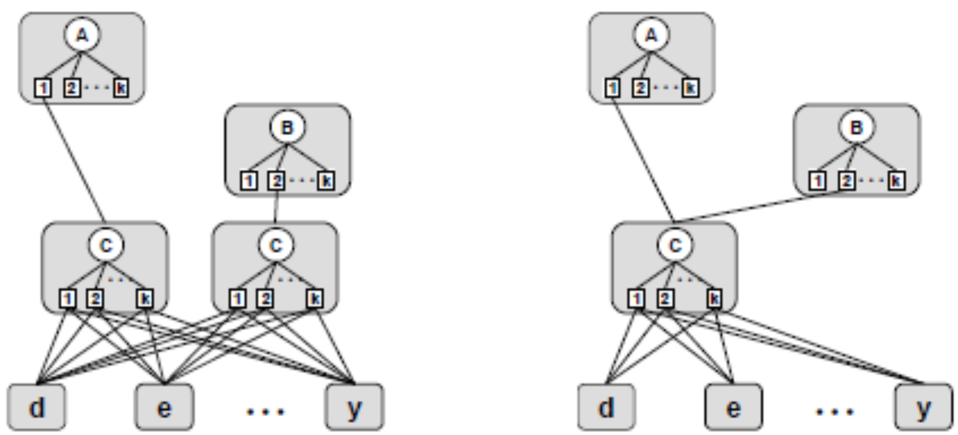
Redundancy and Isomorphism Rules



(a) Fragment of an AOMDD

(b) After eliminating the B meta-node

Figure 13: Redundancy reduction



(a) Fragment of an AOMDD

(b) After merging the isomorphic C meta-nodes

Outline

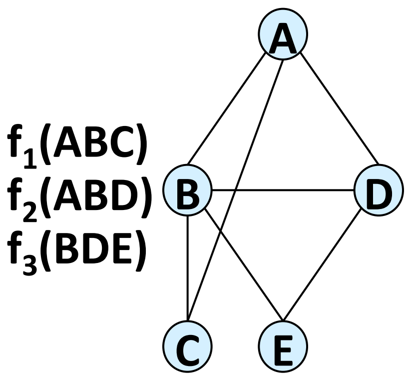
- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
  - Top down
  - Bottom up
- AOMDDs and earlier BDDs

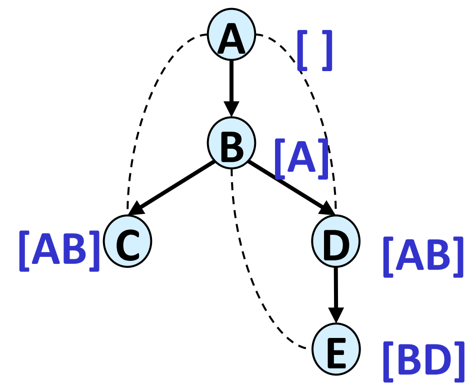
Cost Networks- Weighted AND/OR Tree

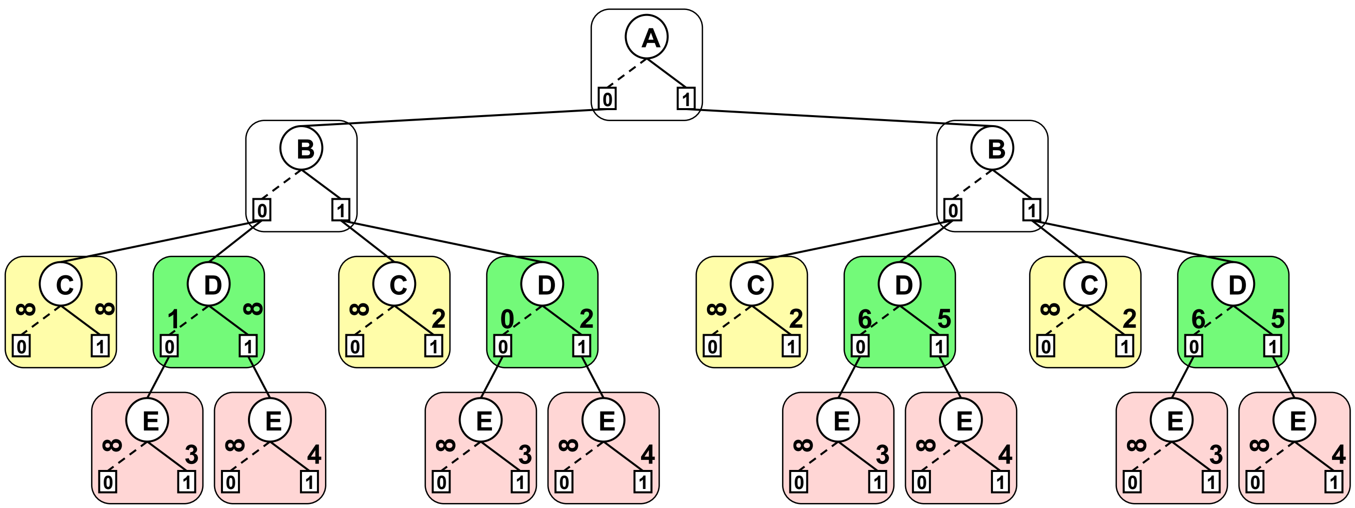
A	B	C	f_1(ABC)
0	0	0	∞
0	0	1	∞
0	1	0	∞
0	1	1	2
1	0	0	∞
1	0	1	2
1	1	0	∞
1	1	1	2

A	B	D	f_2(ABD)
0	0	0	1
0	0	1	∞
0	1	0	0
0	1	1	2
1	0	0	6
1	0	1	5
1	1	0	6
1	1	1	5

B	D	E	f_3(BDE)
0	0	0	∞
0	0	1	3
0	1	0	∞
0	1	1	4
1	0	0	∞
1	0	1	3
1	1	0	∞
1	1	1	4





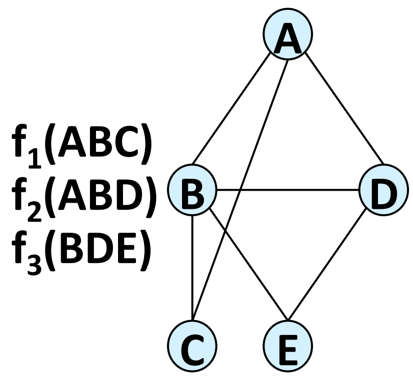


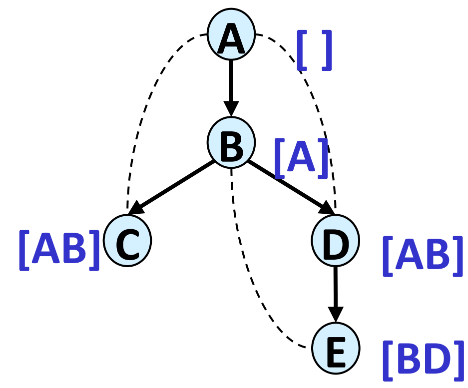
Weighted AND/OR Context Minimal Graph

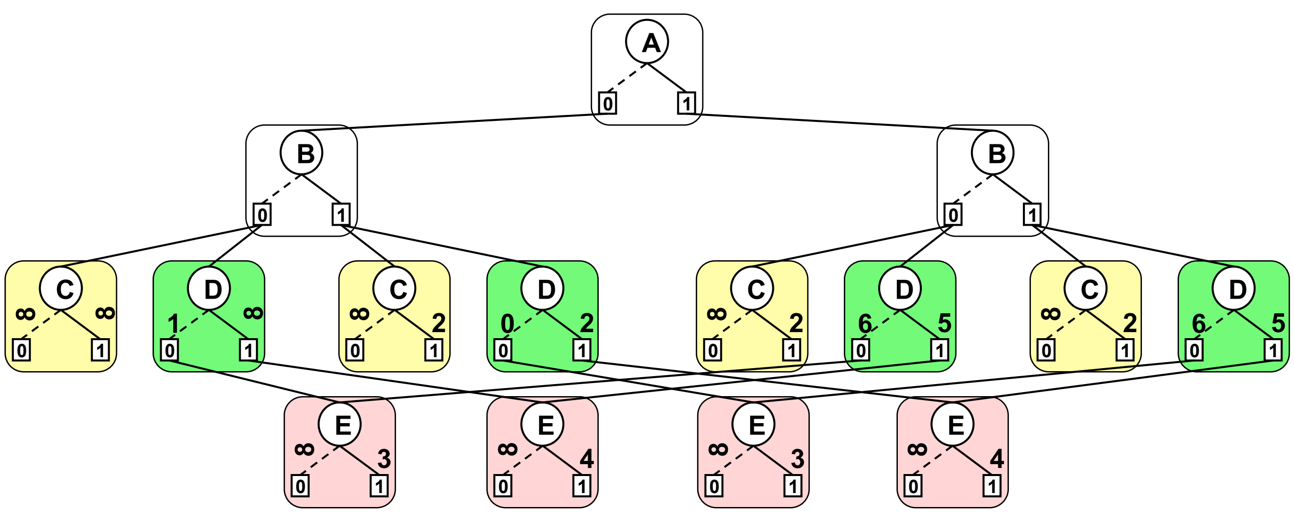
A	B	C	$f_1(ABC)$
0	0	0	∞
0	0	1	∞
0	1	0	∞
0	1	1	2
1	0	0	∞
1	0	1	2
1	1	0	∞
1	1	1	2

A	B	D	$f_2(ABD)$
0	0	0	1
0	0	1	∞
0	1	0	0
0	1	1	2
1	0	0	6
1	0	1	5
1	1	0	6
1	1	1	5

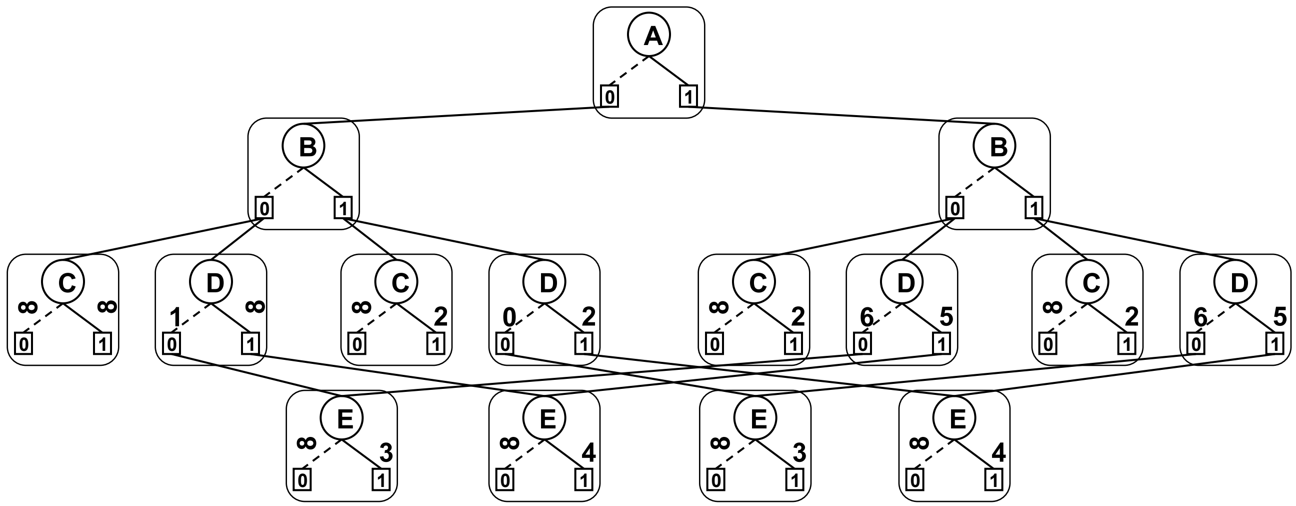
B	D	E	$f_3(BDE)$
0	0	0	∞
0	0	1	3
0	1	0	∞
0	1	1	4
1	0	0	∞
1	0	1	3
1	1	0	∞
1	1	1	4

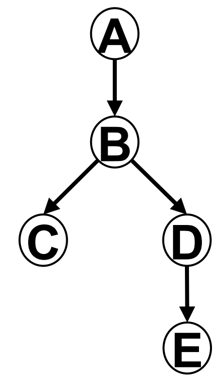




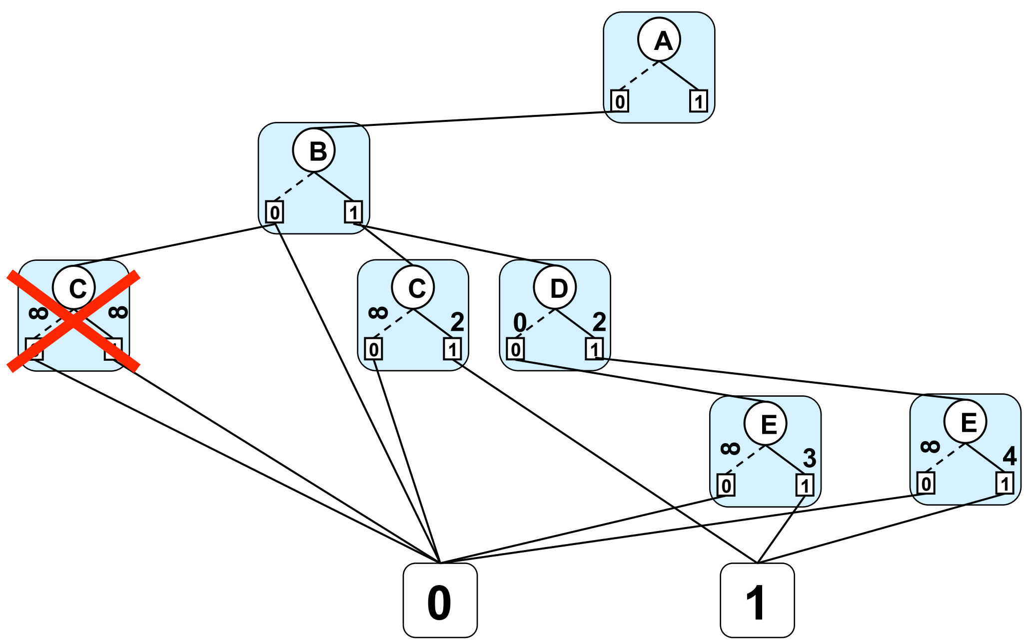


AOMDD – Compilation by Search

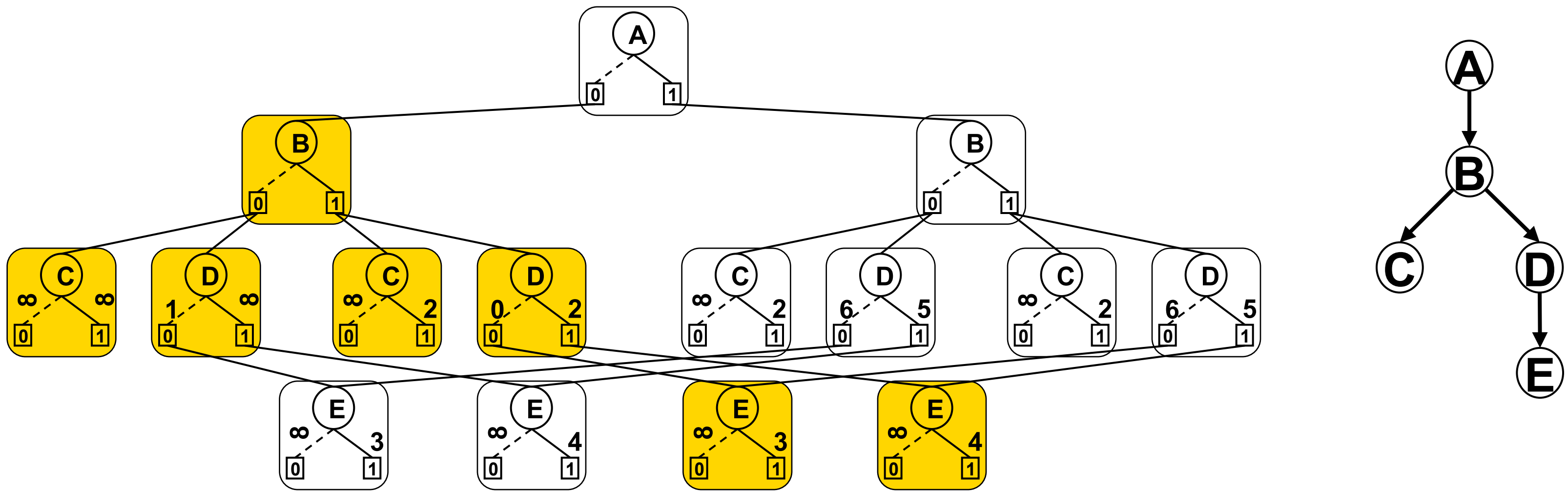


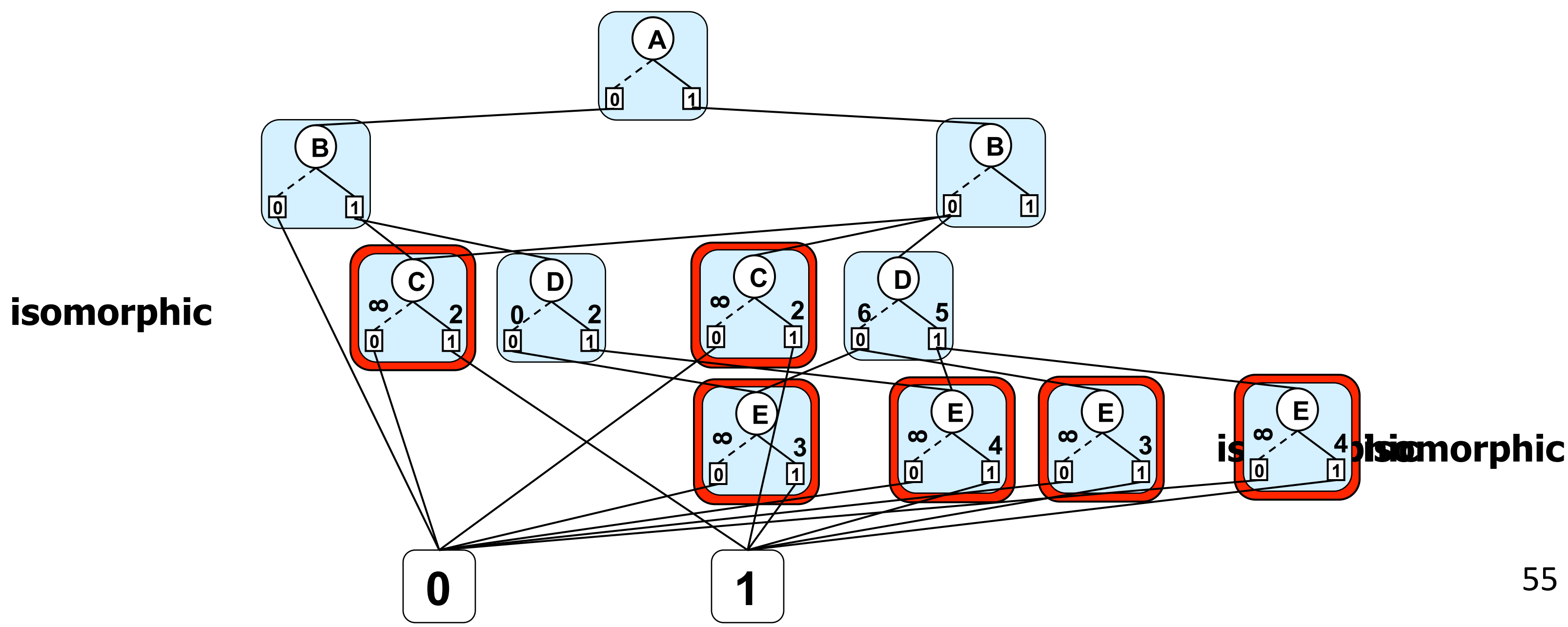


redundant

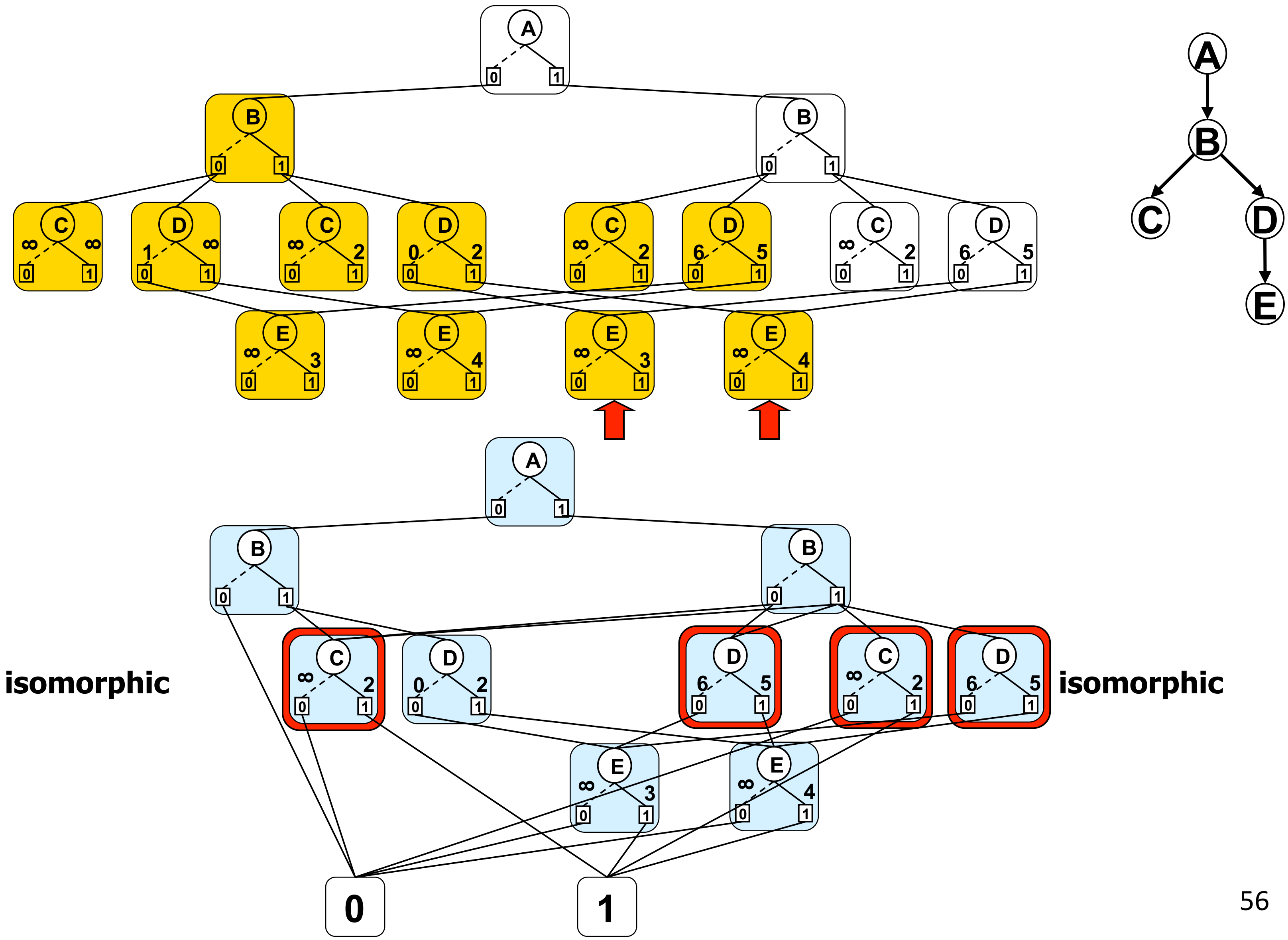


AOMDD – Compilation by Search

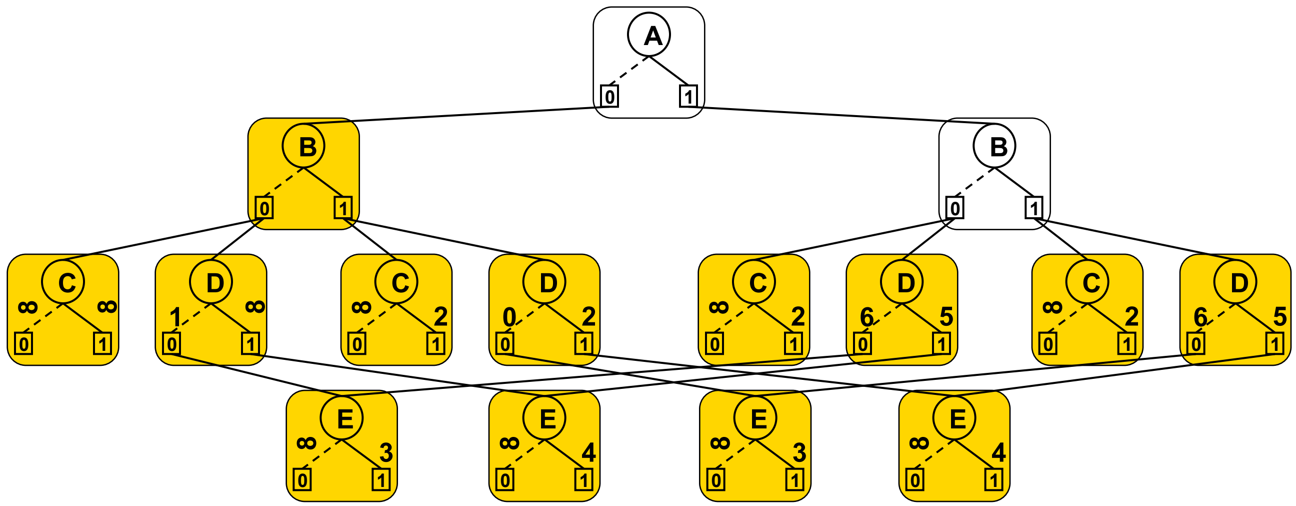


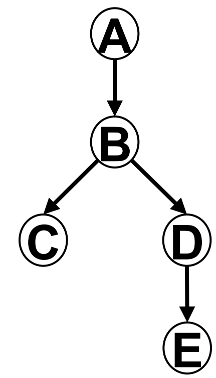


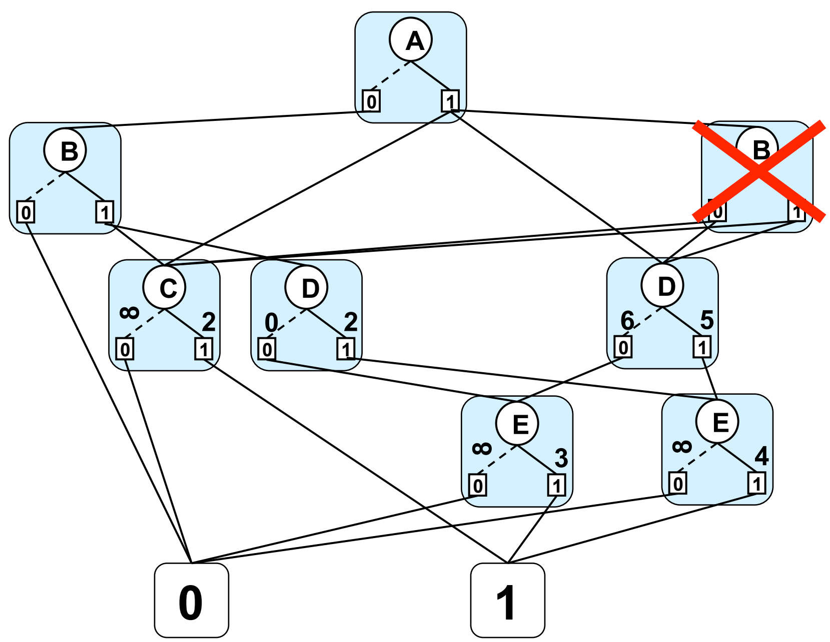
AOMDD – Compilation by Search



AOMDD – Compilation by Search





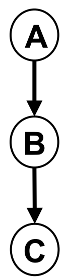


Outline

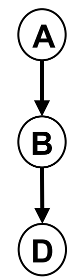
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
  - Top down
  - Bottom up
- AOMDDs and earlier BDDs

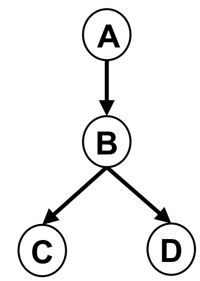
The Apply Operator

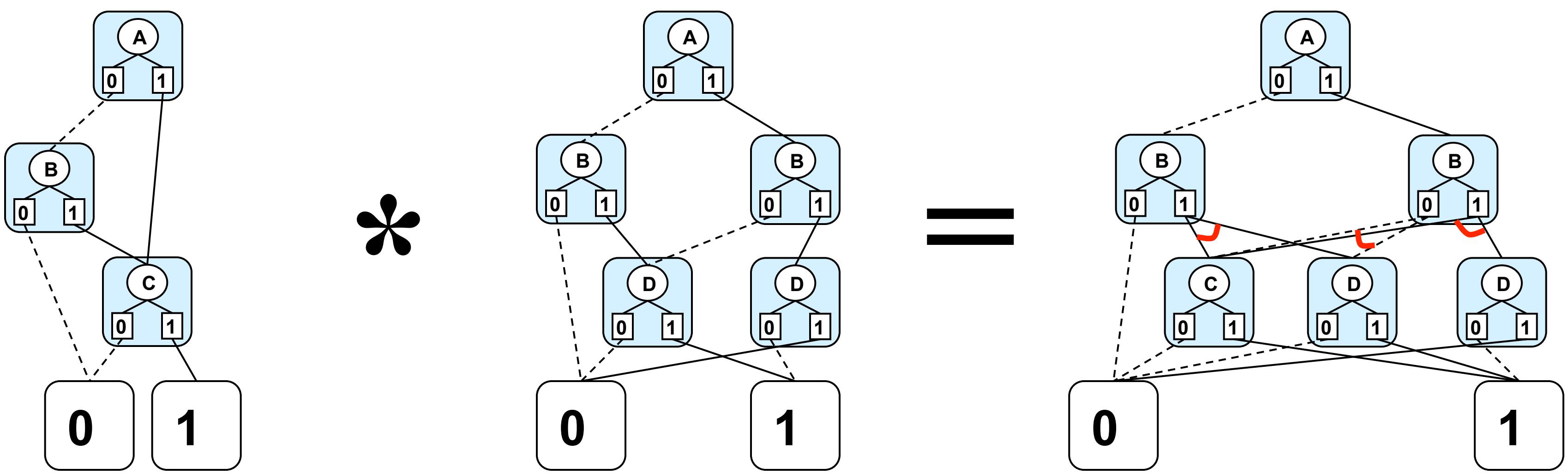
A	B	C	f(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



A	B	D	g(ABD)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

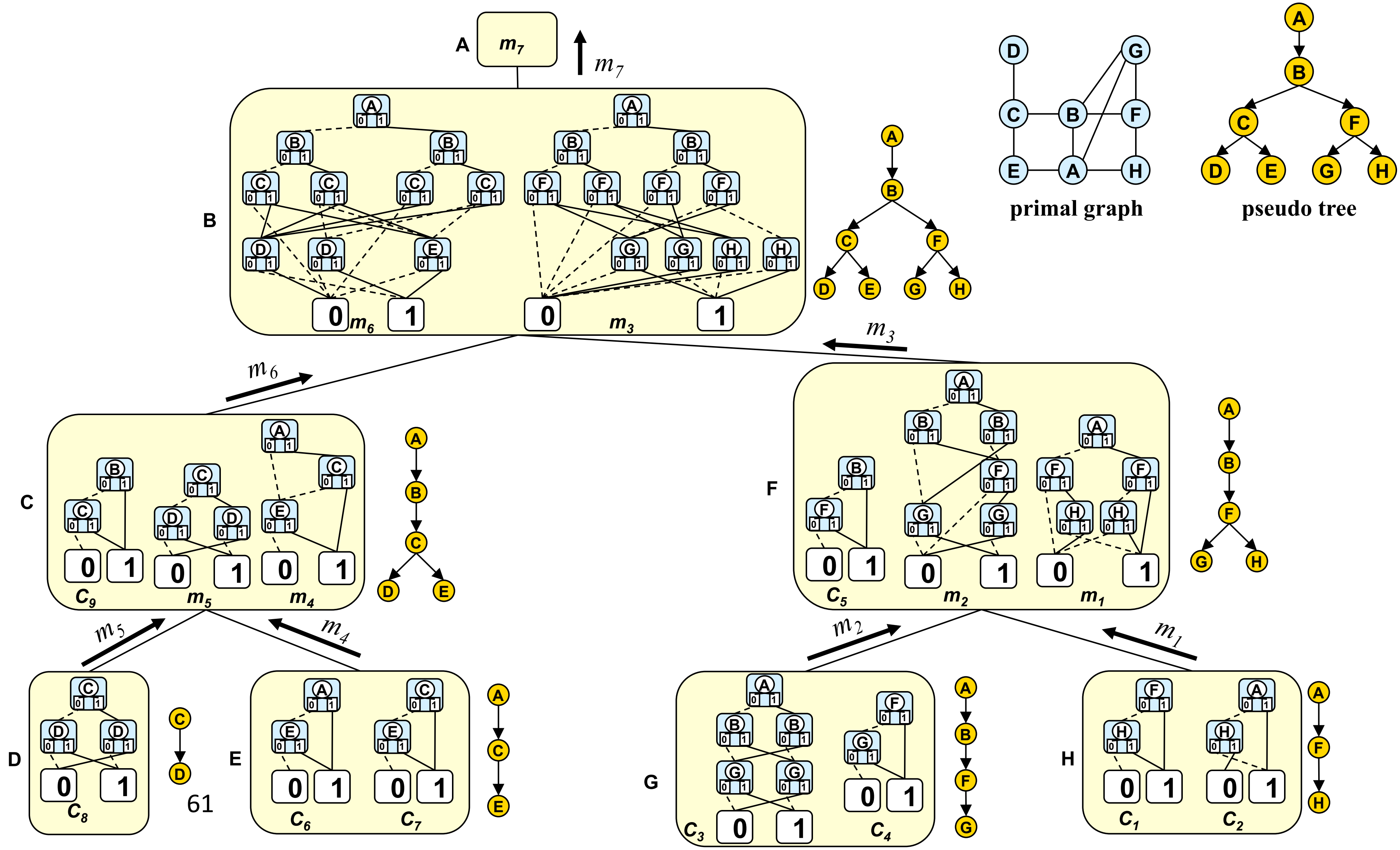






Example:

$$(F \vee H) \wedge (A \vee \neg H) \wedge (A \neq B \neq G) \wedge (F \vee G) \wedge (B \vee F) \wedge (A \vee E) \wedge (C \vee E) \wedge (C \neq D) \wedge (B \vee C)$$



Complexity of Compilation

- The size of the AOMDD is $O(n\ k^{w^*})$
- The compilation time is also bounded by $O(n\ k^{w^*})$

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

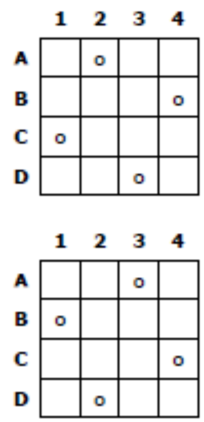
Outline

- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
- Semantic Width
- Empirical demonstration
- Learning AOMDDs

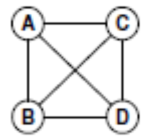
Semantic Treewidth

- Given a graphical model, there may exist a simpler equivalent graphical model
- Semantic treewidth of a pseudo tree: The smallest treewidth over equivalent graphical models that can have that pseudo tree
- Semantic treewidth of a graphical model: The smallest treewidth over all equivalent graphical models with any legal pseudo tree
- Theorem: The size of the AOMDD along T is $O(k^{sw(T)})$
- Theorem: Computing the semantic width of T is NP-hard

Example: Semantic Width



(a) The two solutions



A	B
1	3
1	4
2	4
3	1
4	1
4	2

A	C
1	2
1	4
2	1
2	3
3	2
3	4
4	1
4	3

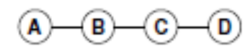
A	D
1	2
1	3
2	1
2	3
2	4
3	1
3	2
3	4
4	2
4	3

B	C
1	3
1	4
2	4
3	1
4	1
4	2

B	D
1	2
1	4
2	1
2	3
3	2
3	4
4	1
4	3

C	D
1	3
1	4
2	4
3	1
4	1
4	2

(b) First model



A	B
2	4
3	1

B	C
1	4
4	1

C	D
1	3
4	2

(c) Second model

Figure 23: The 4-queen problem

The SW of 4 queen is 1.

The SW of an inconsistent network is 0,
If we have k solutions, the sw ≤ k

Width of AOMDD

- AOMDD width:
  - S(T) = number of AND nodes in an AOMDD of a T , then

$$wao(T) = \log_k S(T) - \log_k n - 1.$$

(Because $S = n\, k^{wao^* + 1}$)

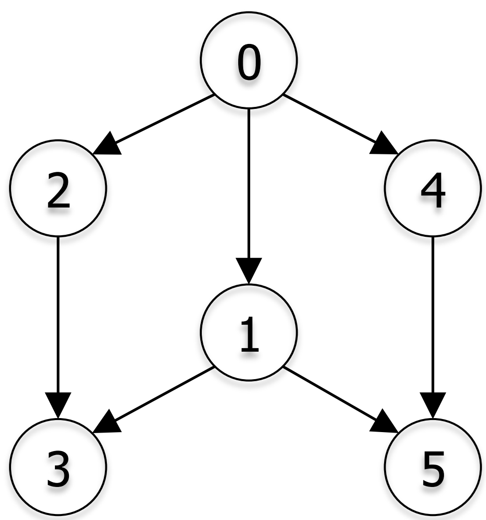
- Effective AOMDD:
  - Let S(v) be the number of AND nodes for v:

$$eao(T) = \max_{v \in T} \log S(v)$$

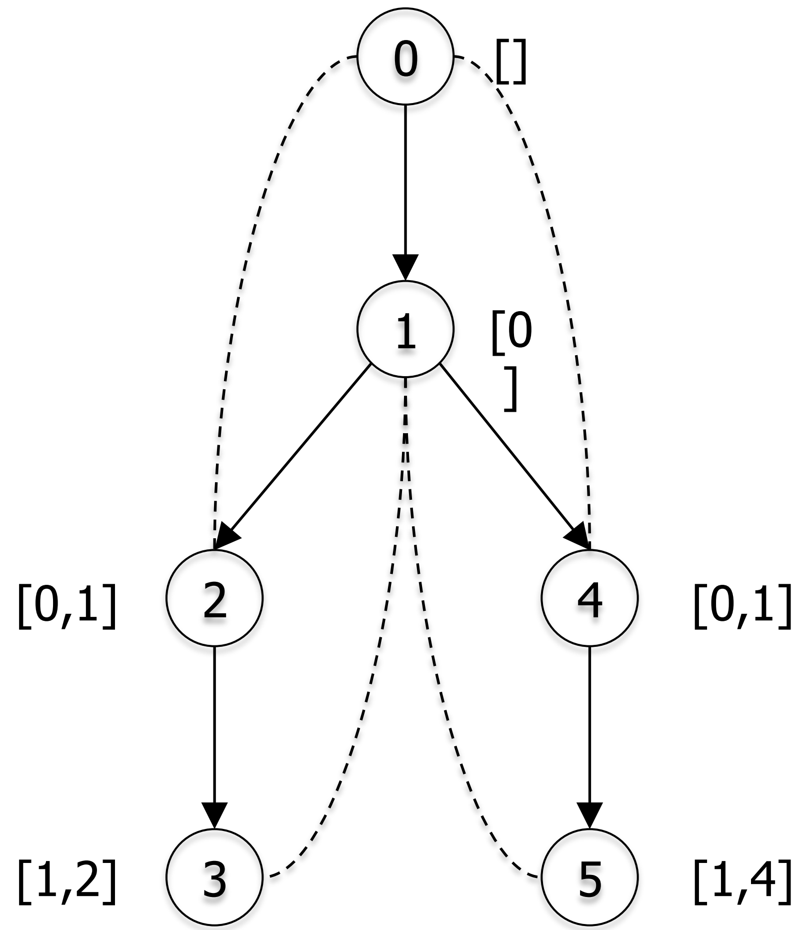
Clearly: $wao(T) \leq sw(T)$

$$eao(T) \leq sw(T)$$

AOMDD Example



Bayesian network structure



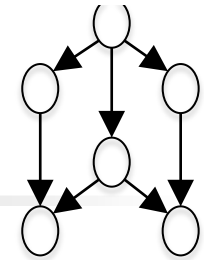
Pseudotree

Demonstrating the Impact of AOMDD

- Parameters generated randomly
  - With no additional properties (M_1)
  - With identical CPTs on a variable (M_2)
  - With determinism (M_3)

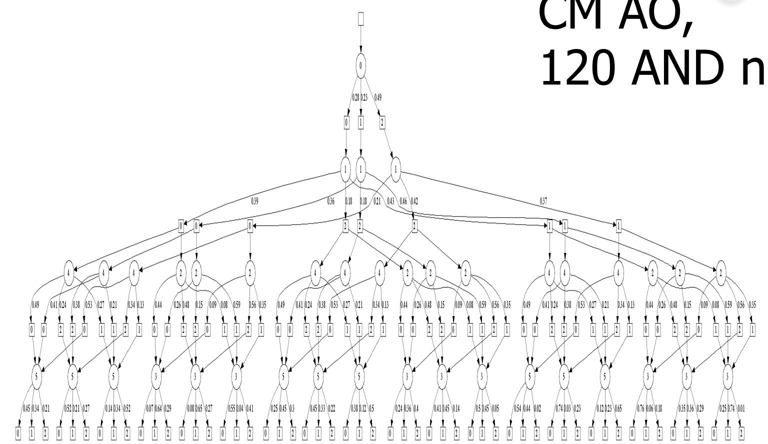
- 3 representations:
  - CM AO graph with pruning for determinism
  - MDD
  - AOMDD

Model: random parameters (M_1)

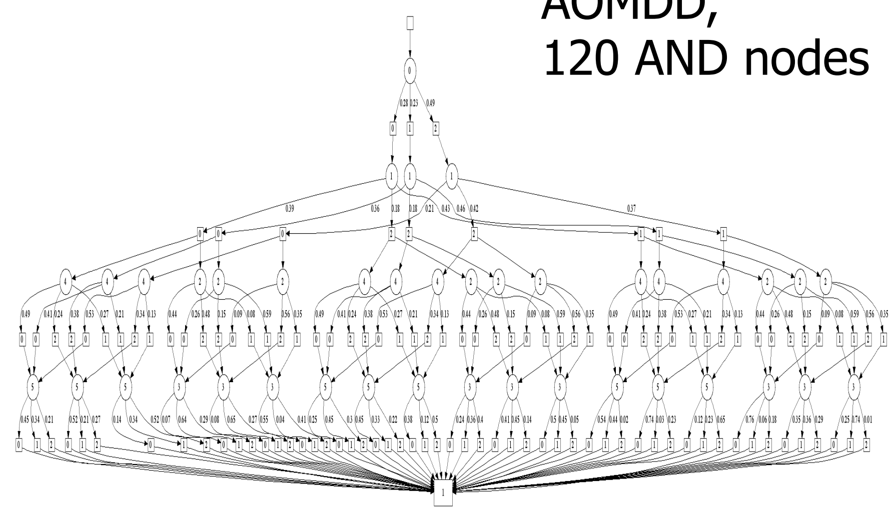


Model	OR Tree	AO Tree	CM AO	MDD	AOMDD
M_1	1092	258	120	228	120
M_2	1092	258	120	99	66
M_3	366	30	30	24	21

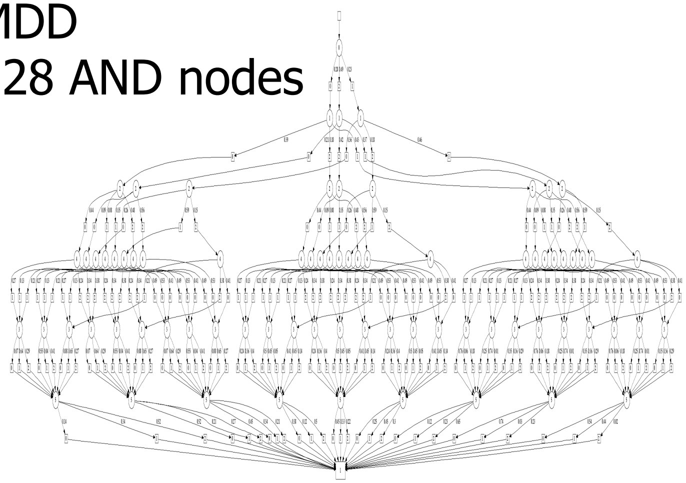
CM AO,
120 AND nodes



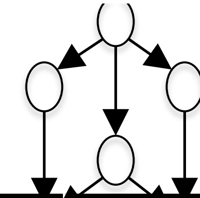
AOMDD,
120 AND nodes



MDD
228 AND nodes

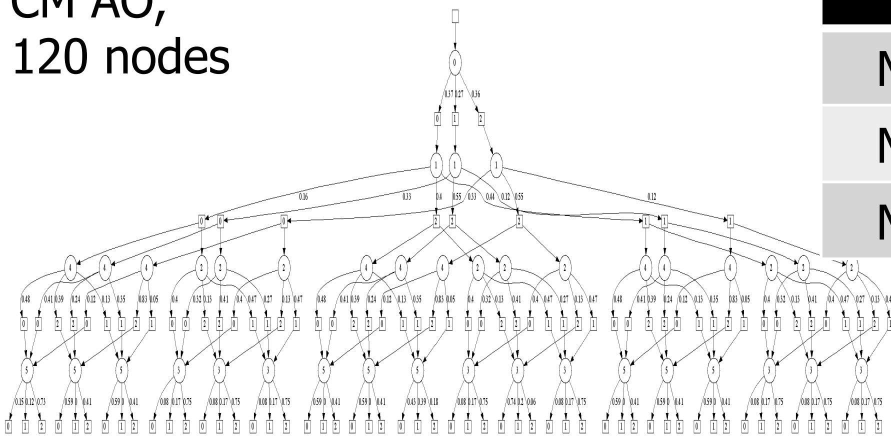


Model : (some) identical parameters (M_2)

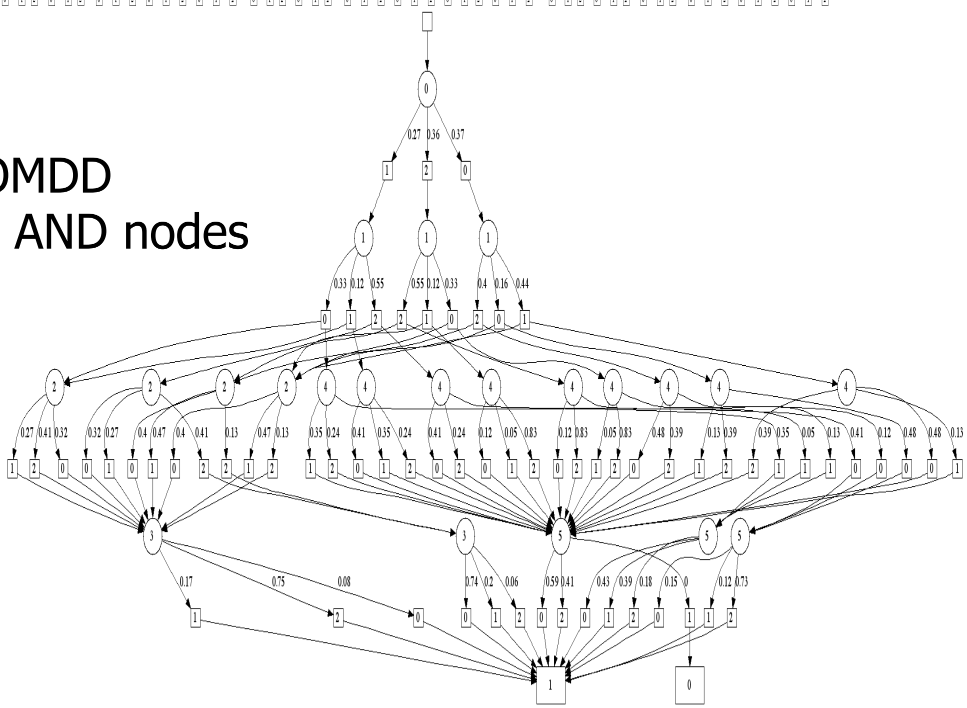


Model	OR Tree	AO Tree	CM AO	MDD	AOMDD
M_1	1092	258	120	228	120
M_2	1092	258	120	99	66
M_3	366	30	30	24	21

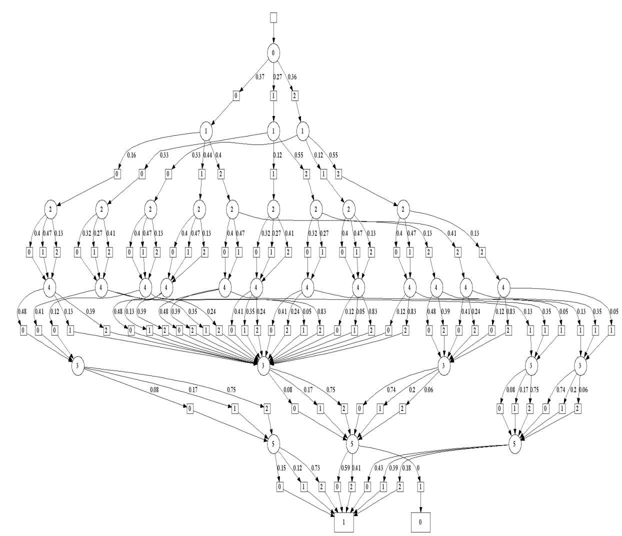
CM AO,
120 nodes



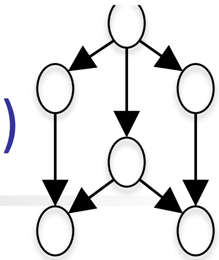
AOMDD
66 AND nodes



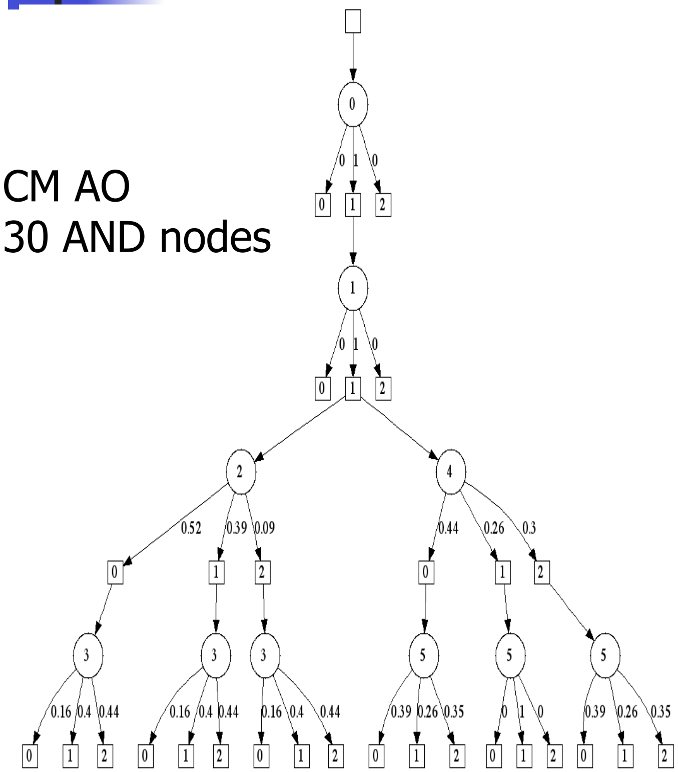
MDD,
99 AND nodes



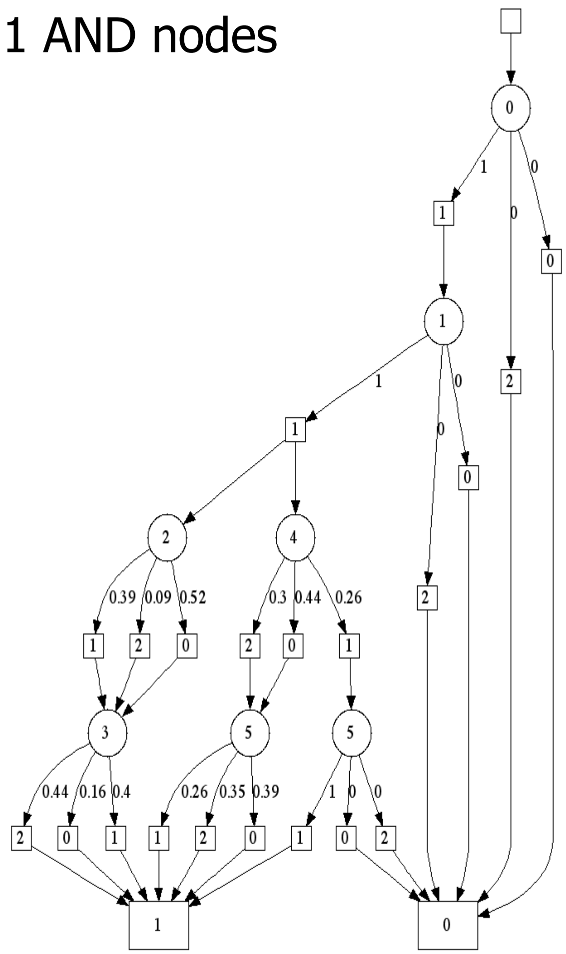
Model: (some) identical parameters with determinism (M_3)



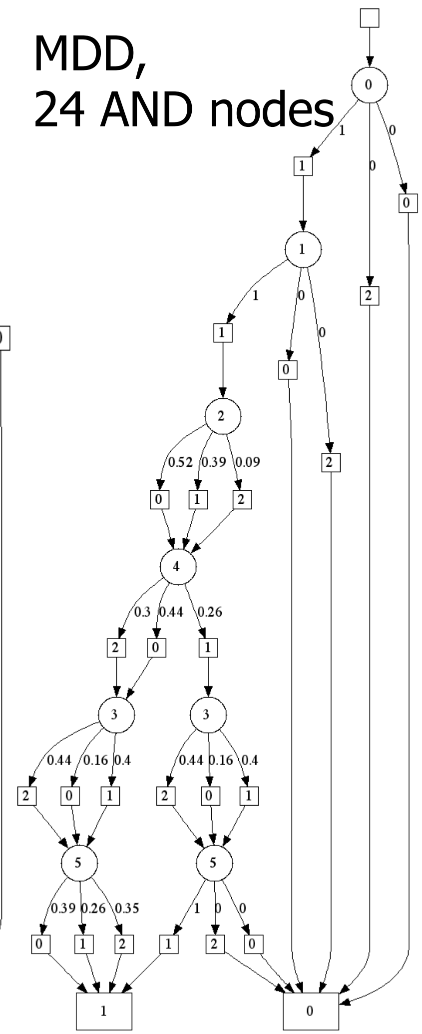
CM AO
30 AND nodes



AOMDD
21 AND nodes



MDD,
24 AND nodes



Model	OR Tree	AO Tree	CM AO	MDD	AOMDD
M_1	1092	258	120	228	120
M_2	1092	258	120	99	66
M_3	366	30	30	24	21

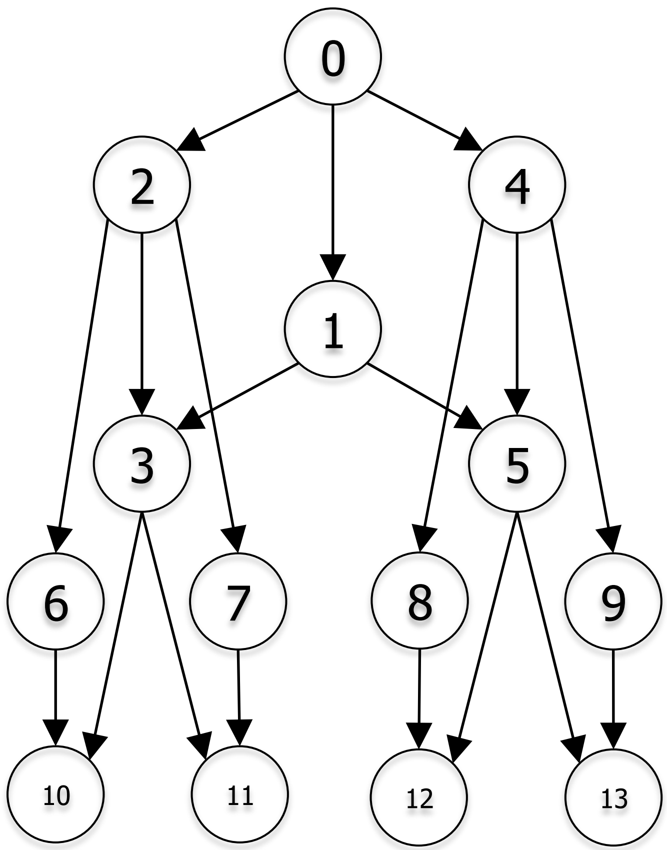
AOMDD-width

- AOMDD-width
  - $\log_k(\#AOMDD\text{-}AND) - \log_k(n) - 1$

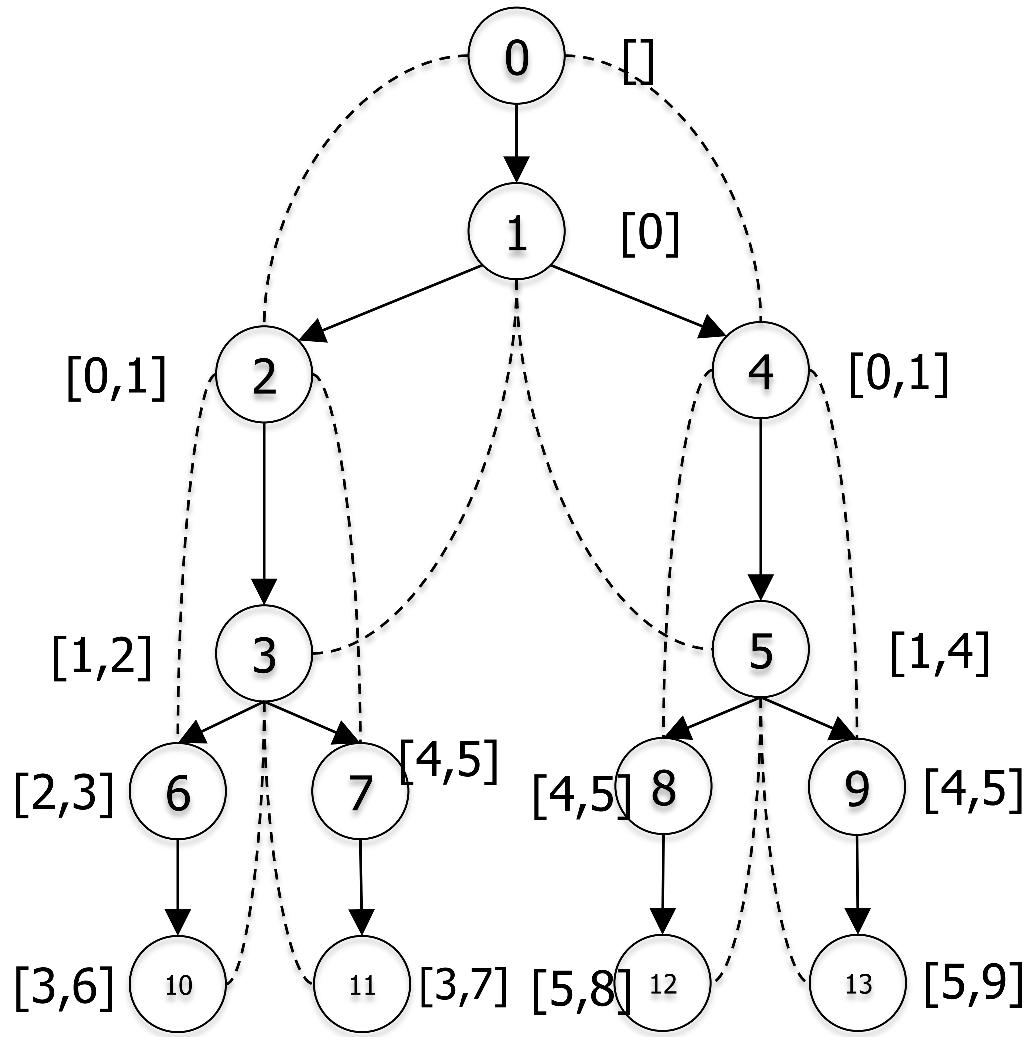
Model	#AOMDD-AND	AOMDD-width
M_1	120	~1.73
M_2	66	~1.18
M_3	21	~0.14

Model	SW
M_1	2
M_2	2
M_3	1

A Larger Example



Bayesian network structure

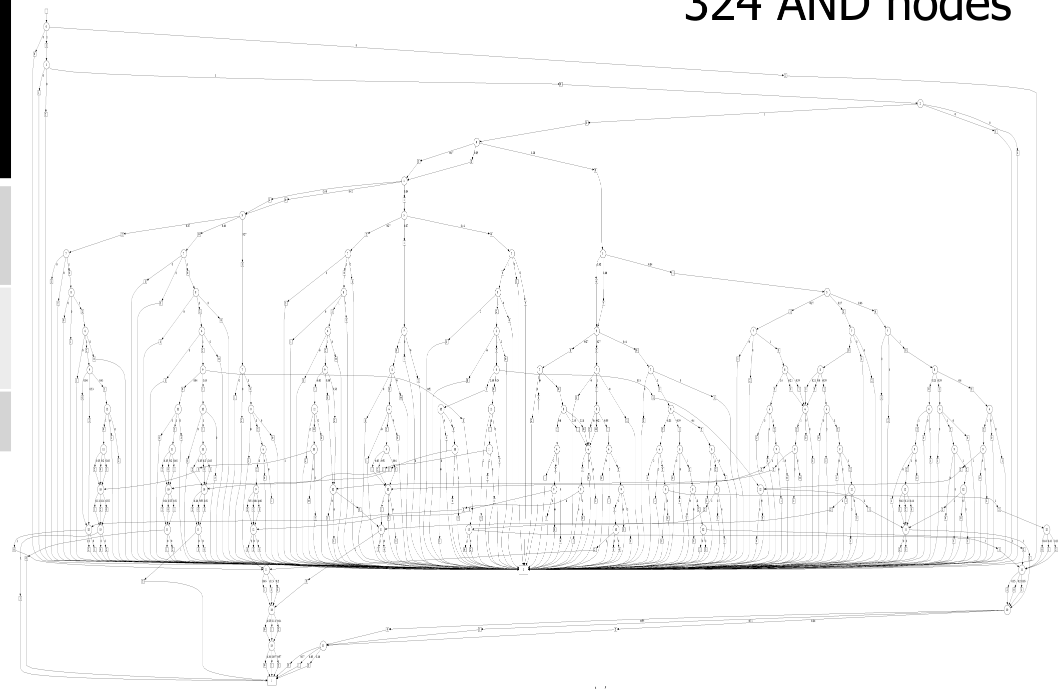


Pseudotree

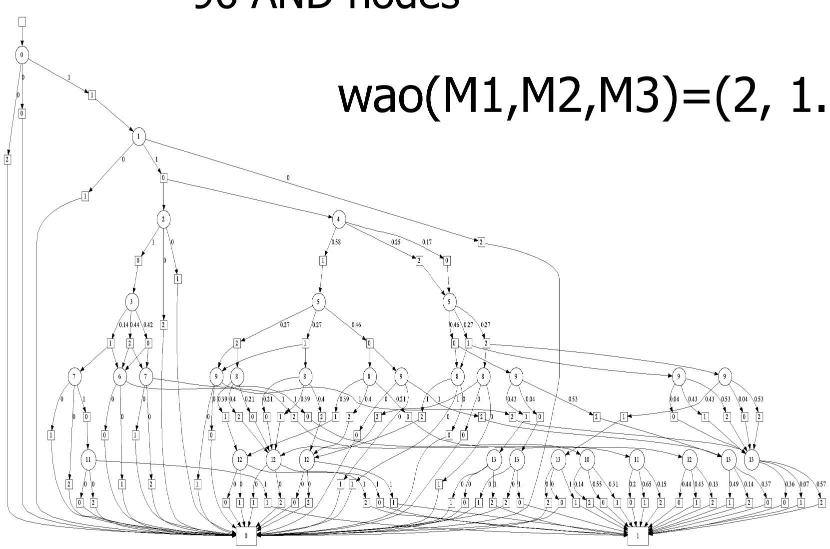
CM AO: (some) identical parameters with determinism (M_3)

Model	OR Tree	AO Tree	CM AO	MDD	AOMD D
M_1	7174452	4080	336	8976	336
M_2	7174452	4080	336	666	144
M_3	3420	222	168	324	96

MDD
324 AND nodes

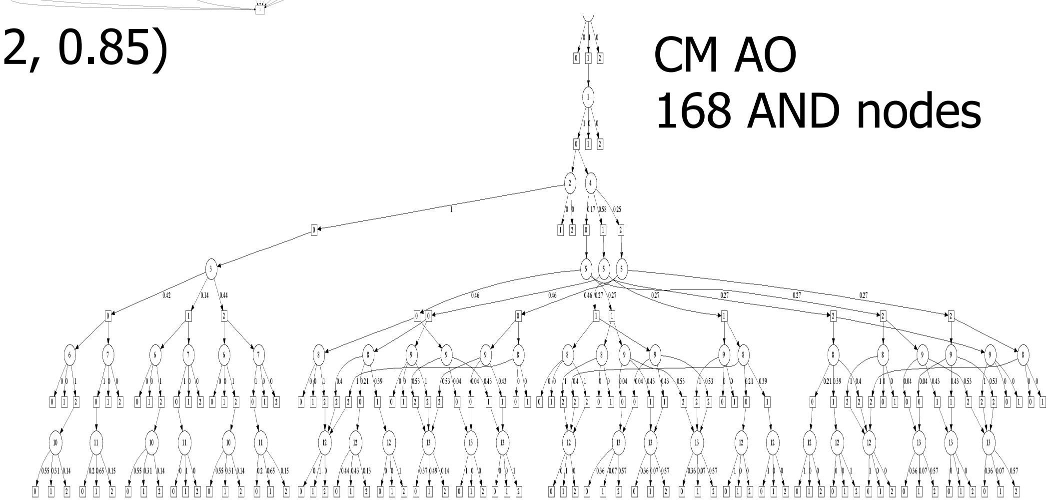


AOMDD,
96 AND nodes



wao(M1,M2,M3)=(2, 1.2, 0.85)

CM AO
168 AND nodes



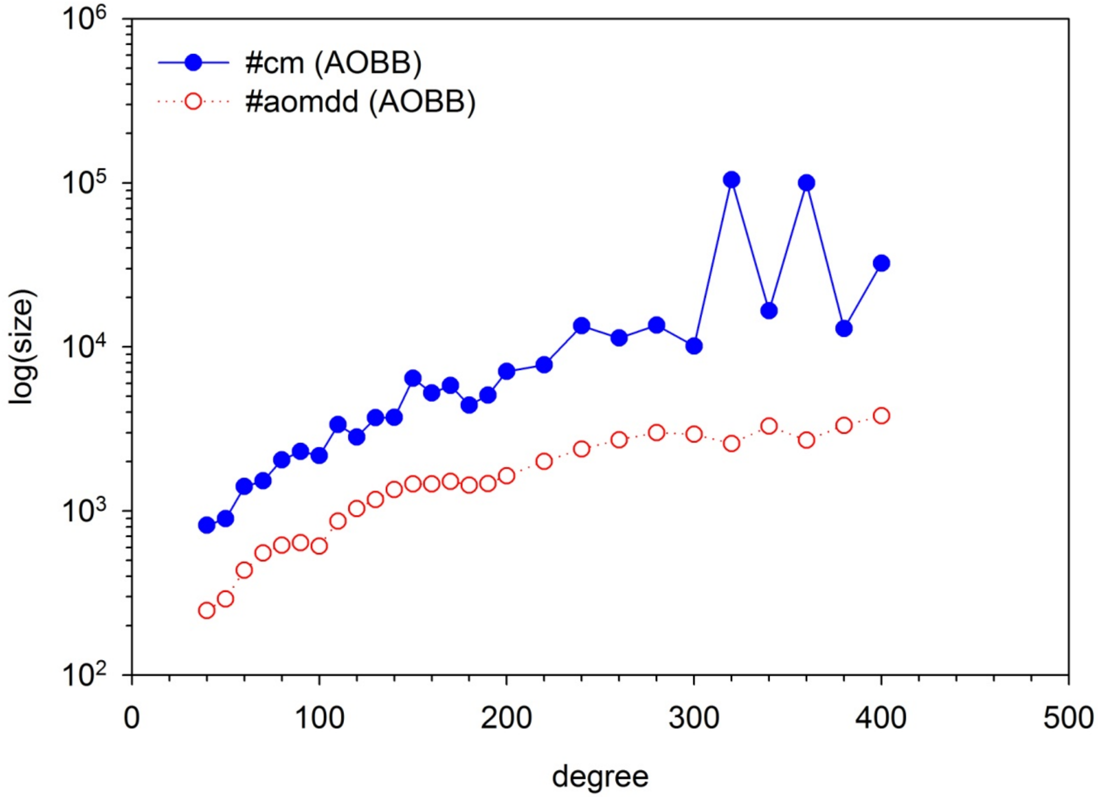
Outline

- Motivation
- Background in Graphical models
- AND/OR search trees and Graphs
- Minimal AND/OR graphs
- From AND/OR search graphs to AOMDDs
- Compilation of AOMDDs
- Semantic Width
- Empirical demonstration
- Learning AOMDDs

Empirical Evaluation

- Bayesian Networks (UAI 2006 evaluation)
- Weighted CSPs
- Randomly generated Bayesian Networks
- Pedigree networks

MAX-SAT Instances (ILP)



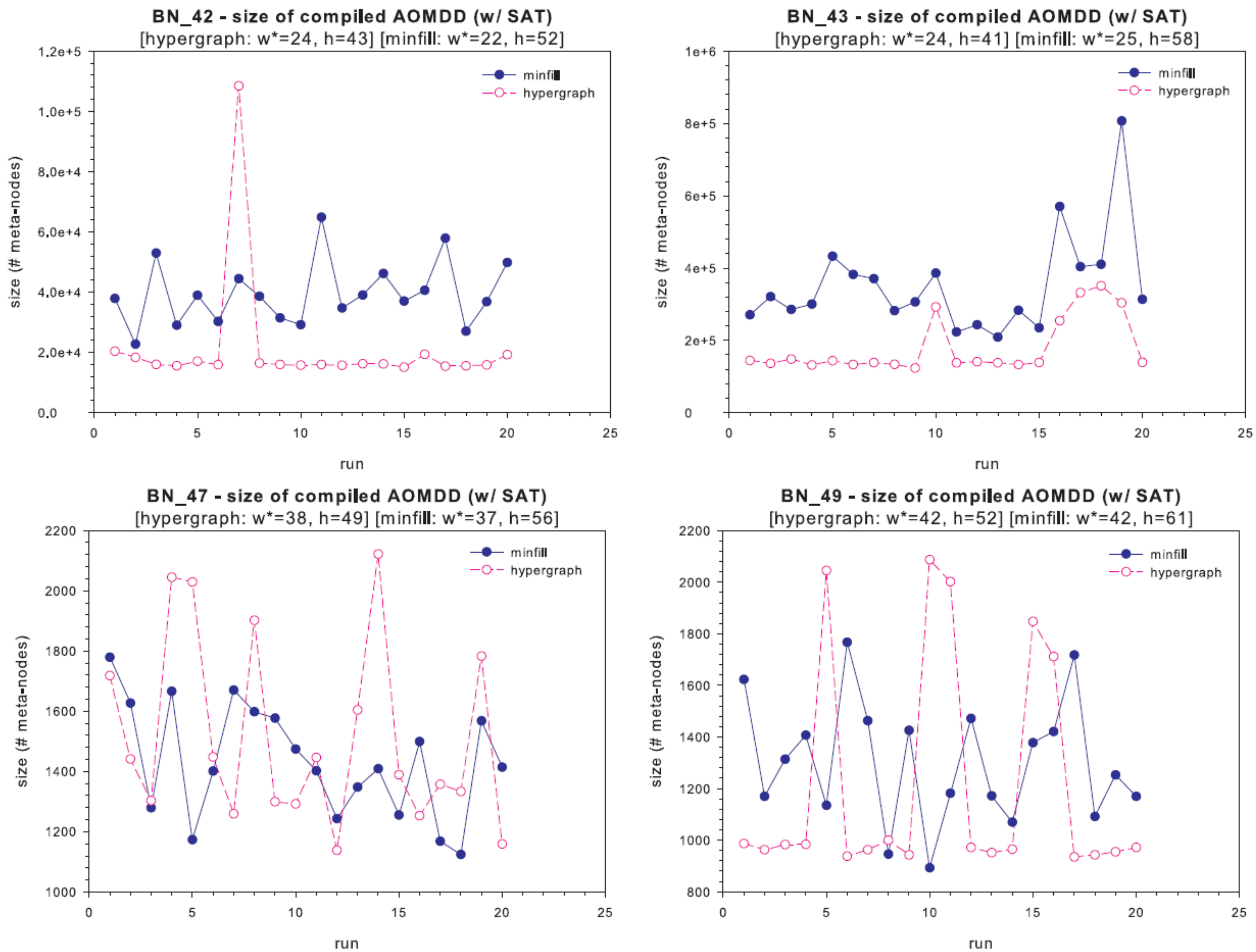
Results for dubois MAX-SAT instances

Bayesian Networks Repository

Network	(w*, h)	(n, k)	ACE		MDD w/ BCP			AOMDD w/ BCP			AOMDD w/ SAT		
			#nodes	time	#meta	#cm(OR)	time	#meta	#cm(OR)	time	#meta	#cm(OR)	time
Bayesian Network Repository													
alarm	(4, 13)	(37, 4)	1,511	0.01	208,837	682,195	73.35	**320**	459	0.05	**320**	459	0.22
cpcs54	(14, 23)	(54, 2)	196,933	0.06	-	-	-	**65,158**	66,405	6.97	**65,158**	66,405	6.97
cpcs179	(8, 14)	(179, 4)	67,919	0.05	-	-	-	**9,990**	32,185	46.56	**9,990**	32,185	46.56
cpcs360b	(20, 27)	(360, 2)	**5,258,826**	1.72	-	-	-	-	-	-	-	-	-
diabetes	(4, 77)	(413, 21)	**7,615,989**	1.81	-	-	-	-	-	-	-	-	-
hailfinder	(4, 16)	(56, 11)	8,815	0.01	-	-	-	2,068	2,202	0.34	**1,893**	2,202	1.48
mildew	(4, 13)	(35, 100)	823,913	0.39	-	-	-	73,666	110,284	1367.81	**62,903**	65,599	3776.82
mm	(20, 57)	(1220, 2)	47,171	1.49	-	-	-	38,414	58,144	4.54	**30,274**	52,523	99.55
munin2	(9, 32)	(1003, 21)	2,128,147	1.91	-	-	-	-	-	-	-	-	-
munin3	(9, 32)	(1041, 21)	1,226,635	1.27	-	-	-	-	-	-	-	-	-
munin4	(9, 32)	(1044, 21)	2,423,009	4.44	-	-	-	-	-	-	-	-	-
pathfinder	(6, 11)	(109, 63)	18,250	0.05	610,854	1,303,682	352.18	6,984	16,267	30.71	**2,265**	15,963	50.36
pigs	(11, 26)	(441, 3)	636,684	0.19	-	-	-	261,920	294,101	174.29	**198,284**	294,101	1277.72
water	(10, 15)	(32, 4)	59,642	0.52	707,283	1,138,096	95.14	18,744	20,926	2.02	**18,503**	19,225	7.45

Size (number of nodes), time (seconds)

Effect of Variable Ordering



AOMDD Compilation Results

name	n	w	h	k	# functions	time (s)	CM OR	AOMDD Meta	CM AND	AOMDD AND	Effective semantic width	Max UniqueTable Memory (MB)	Max Operation Cache Memory (MB)	Compiled AOMDD memory (MB)
BN_42	850	20	50	2	879	93	5623680	25901	11237360	51802	10.35	203.5	189.65	5.41
BN_43	850	21	50	2	881	484	22731586	148255	45463172	296510	13.76	1181.3	1024	30.88
BN_44	850	21	53	2	880	394	11681649	80878	23363298	161756	13.58	962.73	822.8	16.81
BN_45	850	21	56	2	875	140	15778481	122816	31556962	245632	13.58	292.29	305.16	25.1
BN_46	850	19	47	2	499	268	4277086	4352	8554172	8704	8	618.04	492.24	0.93

(Lam and Dechter CP 2012)

AOMDD Compilation Results

name	n	w	h	k	# functions	time (s)	CM OR	AOMDD Meta	CM AND	AOMDD AND	Effective semantic width	Max UniqueTable Memory (MB)	Max Operation Cache Memory (MB)	Compiled AOMDD memory (MB)
BN_42	850	20	50	2	879	93	5623680	25901	11237360	51802	10.35	203.5	189.65	5.41
BN_43	850	21	50	2	881	484	22731586	148255	45463172	296510	13.76	1181.3	1024	30.88
BN_44	850	21	53	2	880	394	11681649	80878	23363298	161756	13.58	962.73	822.8	16.81
BN_45	850	21	56	2	875	140	15778481	122816	31556962	245632	13.58	292.29	305.16	25.1
BN_46	850	19	47	2	499	268	4277086	4352	8554172	8704	8	618.04	492.24	0.93

Recent Experiments (Lam and Dechter cp 2012)

name	n	w	h	k	# functions	time (s) [BE-AOMDD+R] [AOMDD-BCP]	CM OR	Metanodes [BE-AOMDD+R] [AOMDD-BCP]	Memory Usage (MB)	Compiled AOMDD mem (MB)
BN_42	850	20	50	2	879	10 36	5623680	**25841** 95963	405.21	8.12
BN_43	850	21	50	2	881	73 647	22731586	**148184** 629027	2132.53	46.37
BN_45	850	21	56	2	875	17 142	15778481	**122763** 260917	646.25	34.44

Table 1. Compilation results on UAI 2006 benchmarks (ISCAS circuits). Note that many instances are not shown here, which BE-AOMDD+R fails to compile due to memory limitations.

name	n	w	h	k	# functions	time (s)	CM OR	Metanodes [BE-AOMDD+R]	Max Memory Usage (MB)	Compiled AOMDD memory (MB)
pdb1fna	75	6	18	81	218	136	1983522	56377	467.61	44.44
pdb1j8e	39	6	12	81	119	294	2714323	258198	950.33	238.32
pdb1pef	17	6	11	81	55	430	4123288	342367	4499.79	772.83
pdb1rb9	42	7	14	81	128	1127	13370233	1163424	3789.48	1751.98
pdb2igd	50	6	19	81	146	1295	33711674	451081	3396.36	1132.93

Table 2. Compilation results on protein networks using BE-AOMDD+R.

The power of hidden variables

(a) OBDD representation

(b) Primal graph with hidden variables M, N and P.

Figure 24: The parity function

The AOMDD width can be far far smaller than the SW:
aow = 2, sw=n for parity

Summary

- Weighted AOMDD: a more compact representation than MDD and ADD for all graphical models. Comparable to ACE (Darwiche)
- Explicate structure hidden in the functions
- Canonical for a model along a pseudo-tree
- Provides a lower bound on the “best structure size”
- Semantic width: shedding some light on instance difficulty.
- AOMDD width(s):   lower bound on SW
- AOMDD may be learned directly

Publications

- **Rina Dechter and Robert Mateescu.** "AND/OR Search Spaces for Graphical Models". *Artificial Intelligence 171 (2-3), pp. 73-106, 2007.*
- **Robert Mateescu, Rina Dechter and Radu Marinescu.** "AND/OR Multi-Valued Decision Diagrams (AOMDDs) for Graphical Models *(JAIR), 2008.*
- **Robert Mateescu, Radu Marinescu and Rina Dechter.** "AND/OR Multi-Valued Decision Diagrams (AOMDDs) for Constraint Optimization". *In CP 2007*
- **Robert Mateescu and Rina Dechter.** "AND/OR Multi-Valued Decision Diagrams (AOMDDs) for Weighted Graphical Models". *In UAI'07.*
- **William Lam and Rina Dechter.** "Empirical Evaluation of AND/OR Multivalued Decision Diagrams for Inference" *in Doctoral Programme of CP 2012*.

Thank You !!