Exact Reasoning: AND/OR Search and Hybrids

COMPSCI 276, Fall 2009

Set 8, Rina dechter

Approximation Techniques bounded inference

COMPSCI 276, Fall 2009

Set 6: Rina Dechter

ing: Primary: Class Notes (7)

Probabilistic Inference Tasks

- Belief updating:

$$\mathbf{BEL(X_i) = P(X_i = x_i \,|\, evidence)}$$

- Finding most probable explanation (MPE)

$$\mathbf{\bar{x}^* = \arg\max_{\bar{x}} P(\bar{x}, e)}$$

- Finding maximum a-posteriory hypothesis

$$\mathbf{(a_1^*, \ldots, a_k^*) = \arg\max_{\bar{a}} \sum_{X/A} P(\bar{x}, e)}$$

$A \subseteq X$: **hypothesis variables**

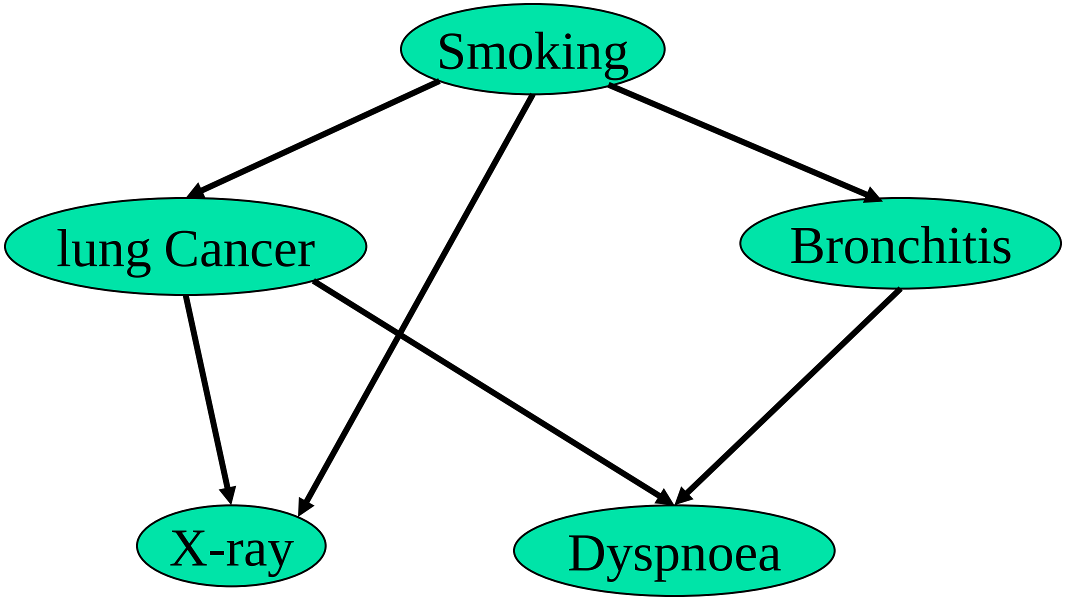
- Finding maximum-expected-utility (MEU) decision

$$\mathbf{(d_1^*, \ldots, d_k^*) = \arg\max_{\bar{d}} \sum_{X/D} P(\bar{x}, e) U(\bar{x})}$$

$D \subseteq X$: **decision variables**

$U(\bar{\mathbf{x}})$: **utility function**

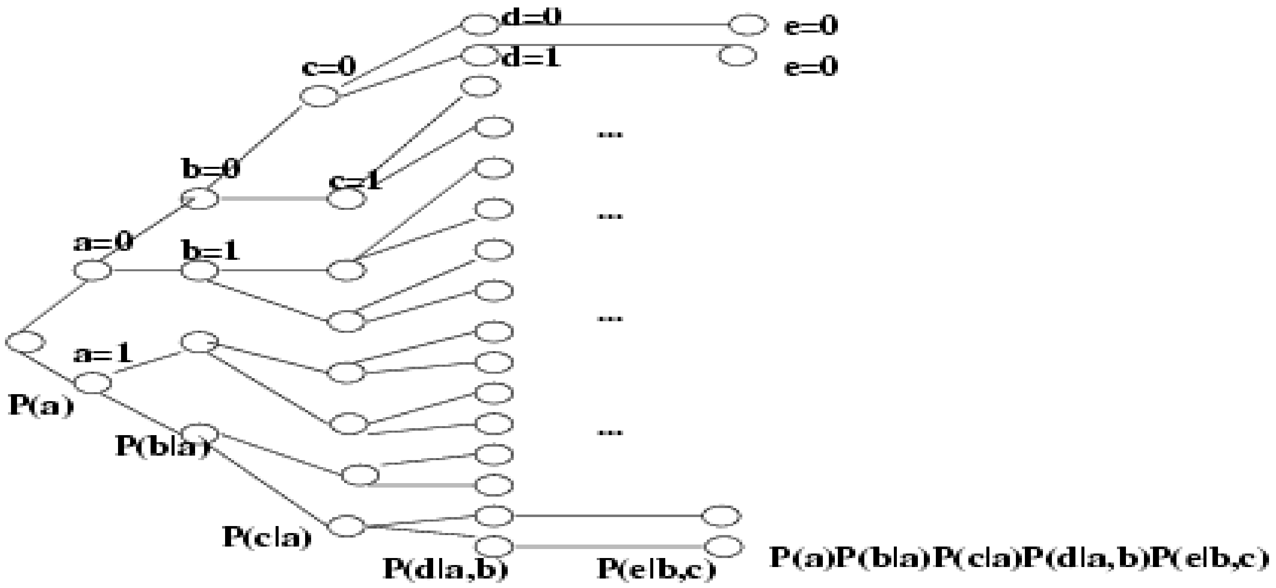
Belief Updating



P (lung cancer=yes | smoking=no, dyspnoea=yes) = ?

Conditioning generates the probability tree

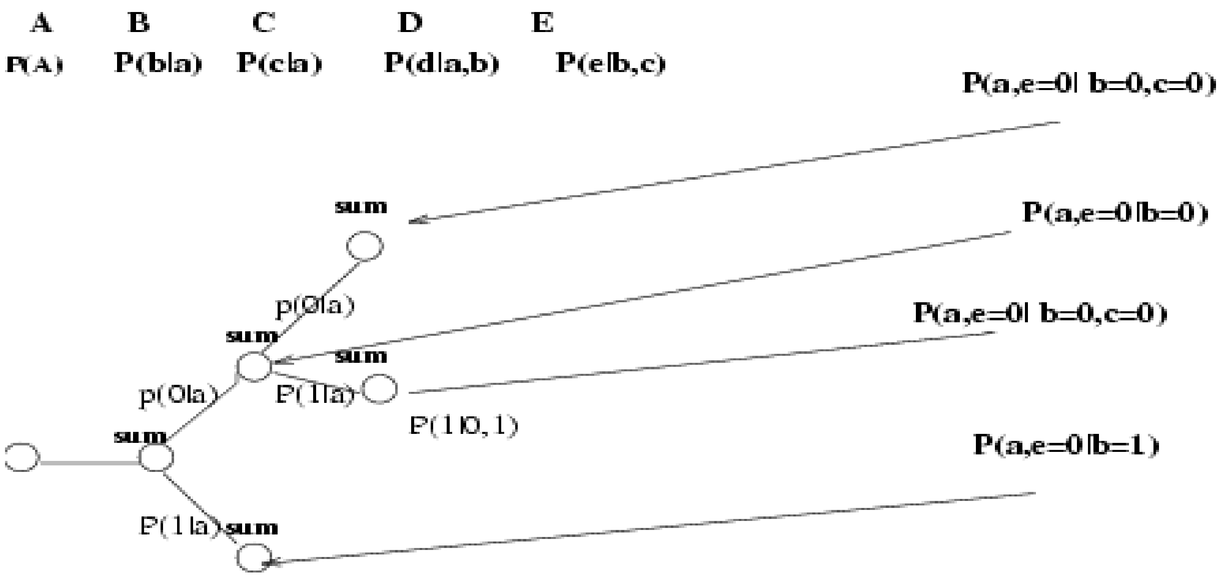
$$P(a,e=0)=P(a)\sum_{b}P(b\,|\,a)\sum_{c}P(c\,|\,a)\sum_{b}P(d\,|\,a,b)\sum_{e=0}P(e\,|\,b,c)$$



Complexity of conditioning: exponential time, linear space

Conditioning+Elimination

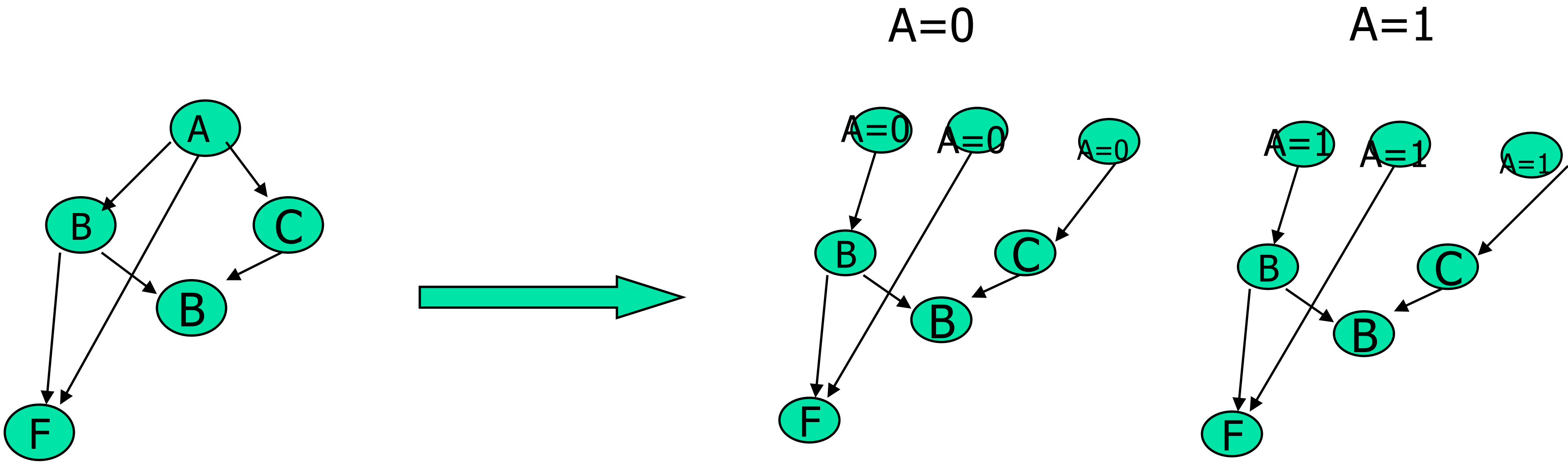
$$P(a,e=0)=P(a)\sum_{b}P(b\mid a)\sum_{c}P(c\mid a)\sum_{d}P(d\mid a,b)\sum_{e=0}P(e\mid b,c)$$



Idea: conditioning until w^* of a (sub)problem gets small

Loop-cutset decomposition

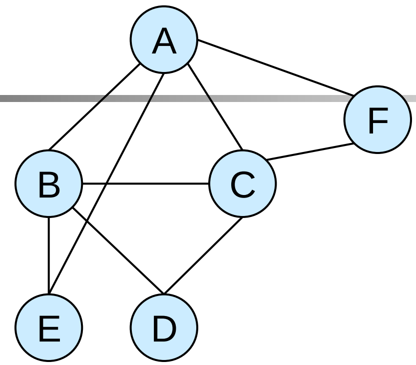
- You condition until you get a polytree



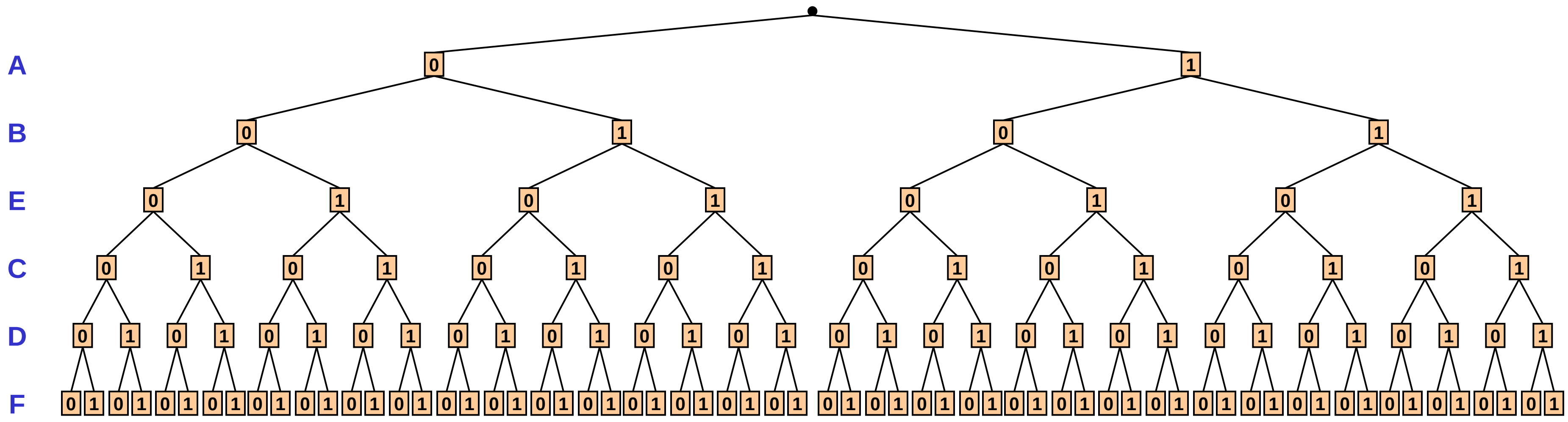
P(B|F=0) = P(B, A=0|F=0)+P(B,A=1|F=0)

Loop-cutset method is time exp in loop-cutset size
And linear space

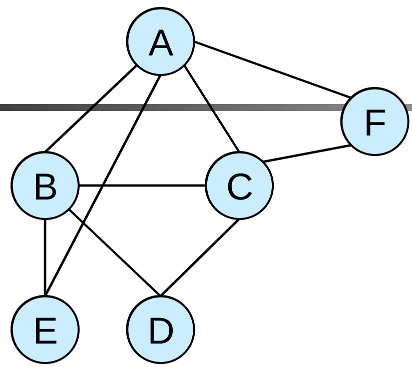
OR search space



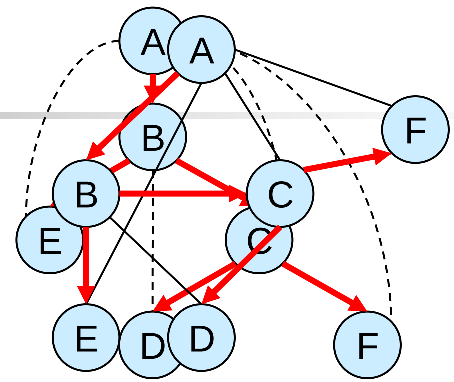
Ordering: A B E C D F



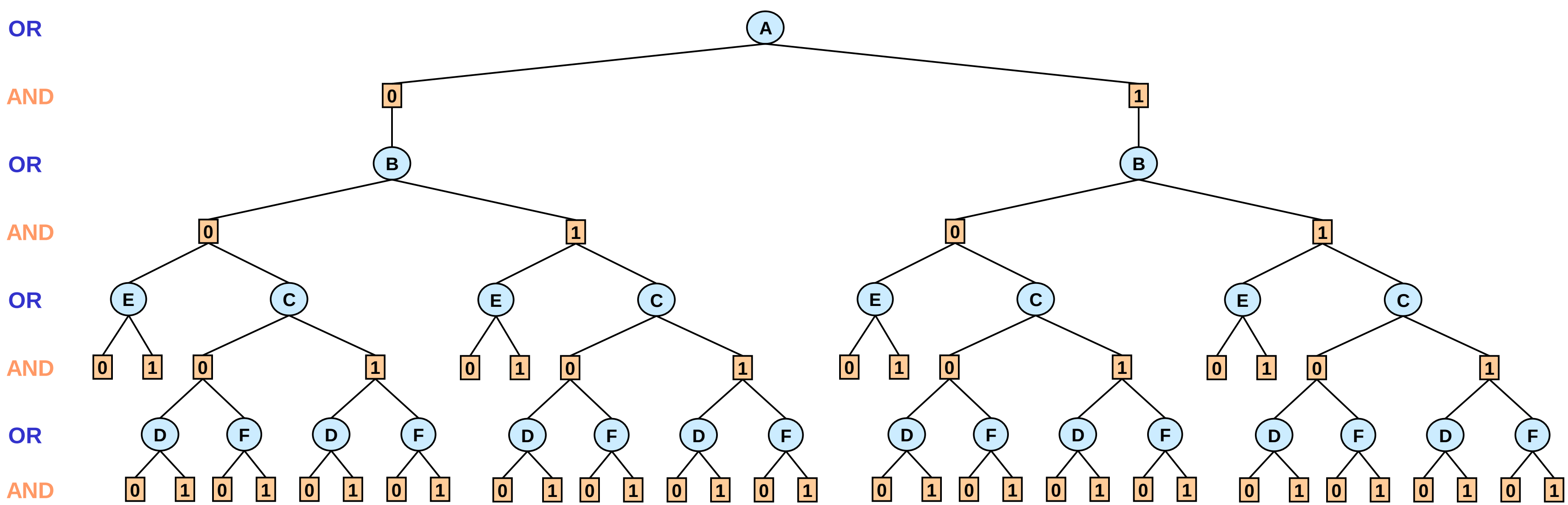
AND/OR search space



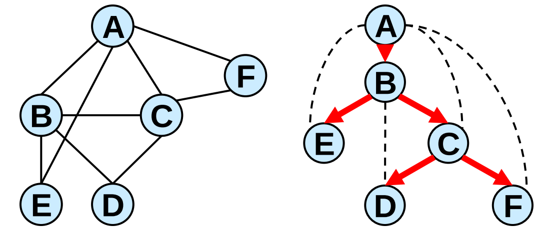
Primal graph

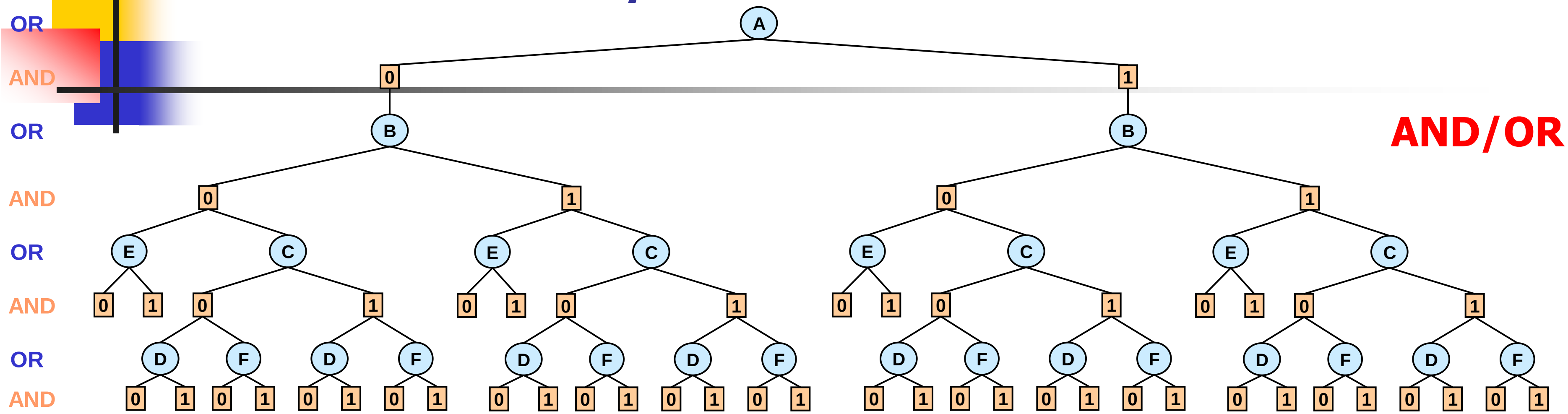


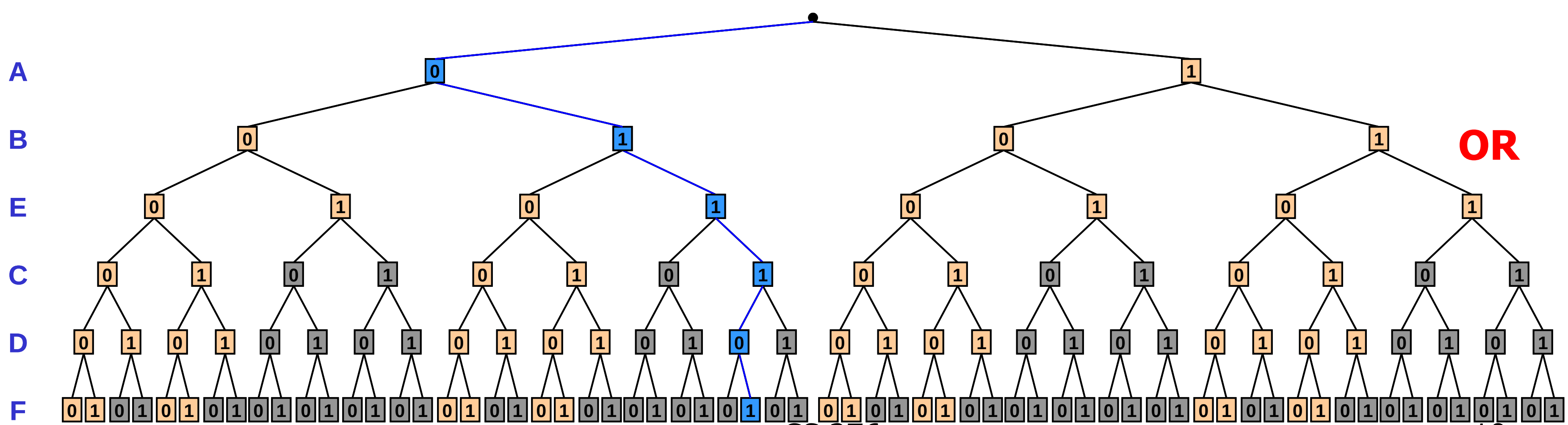
DFS tree



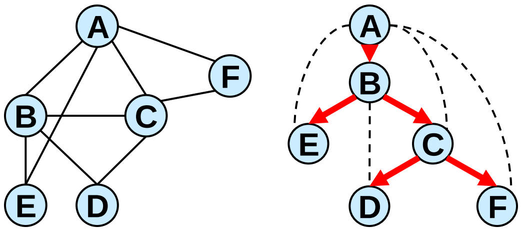
OR vs AND/OR

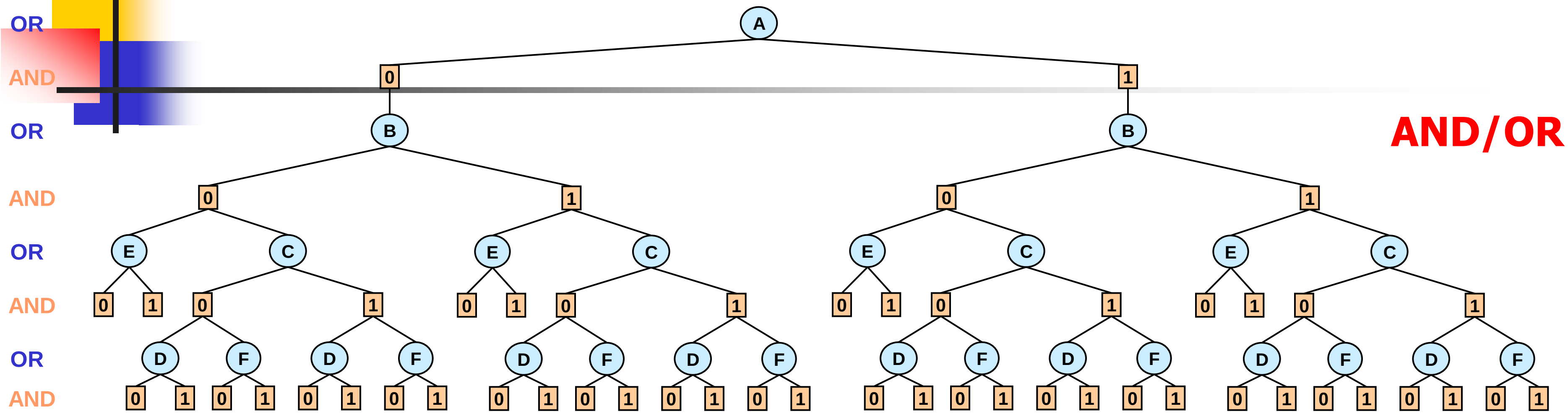




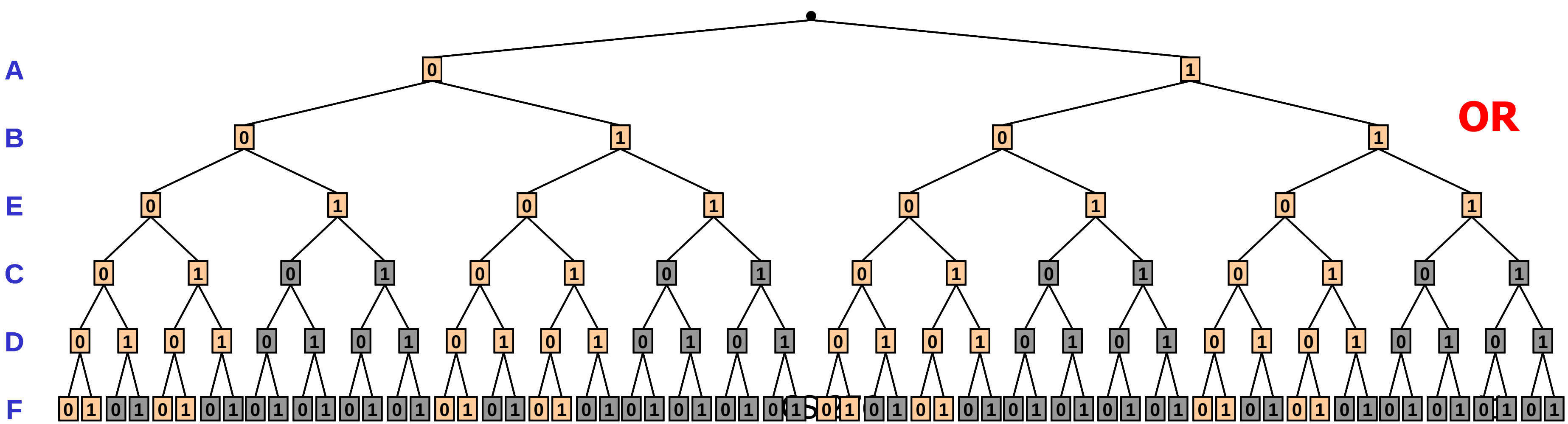


AND/OR vs. OR



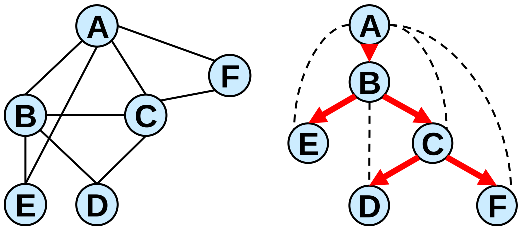


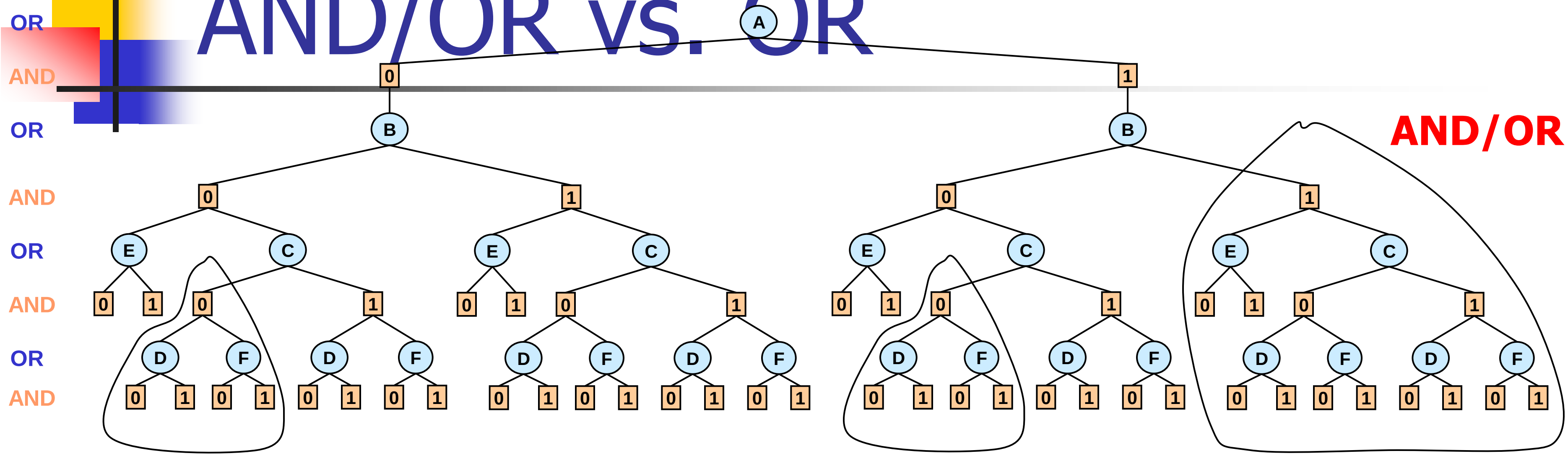
AND/OR size: exp(4), OR size exp(6)

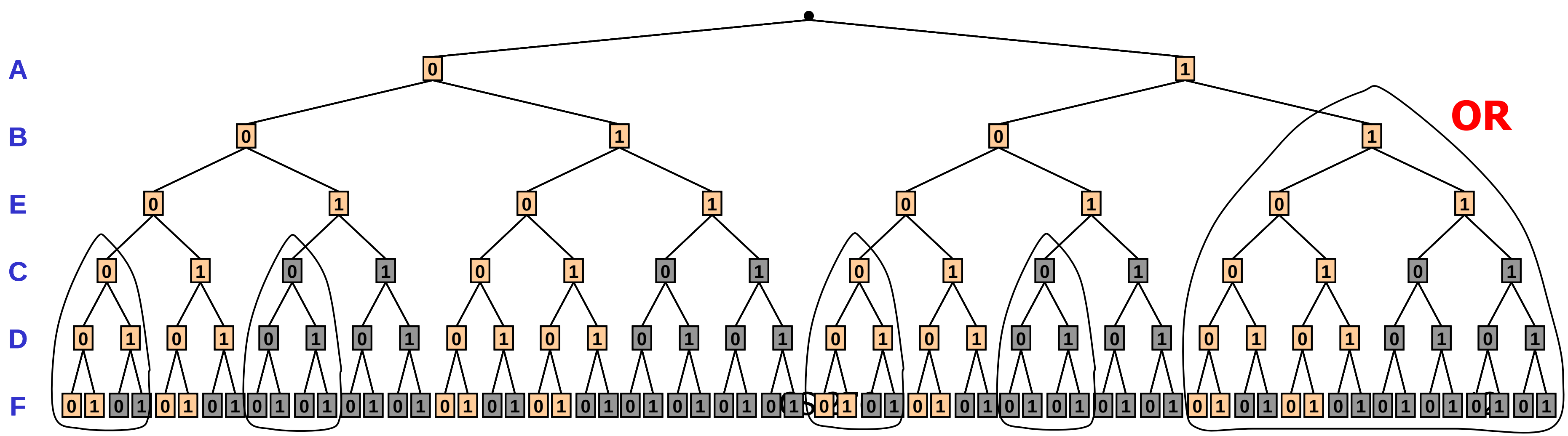


AND/OR vs. OR

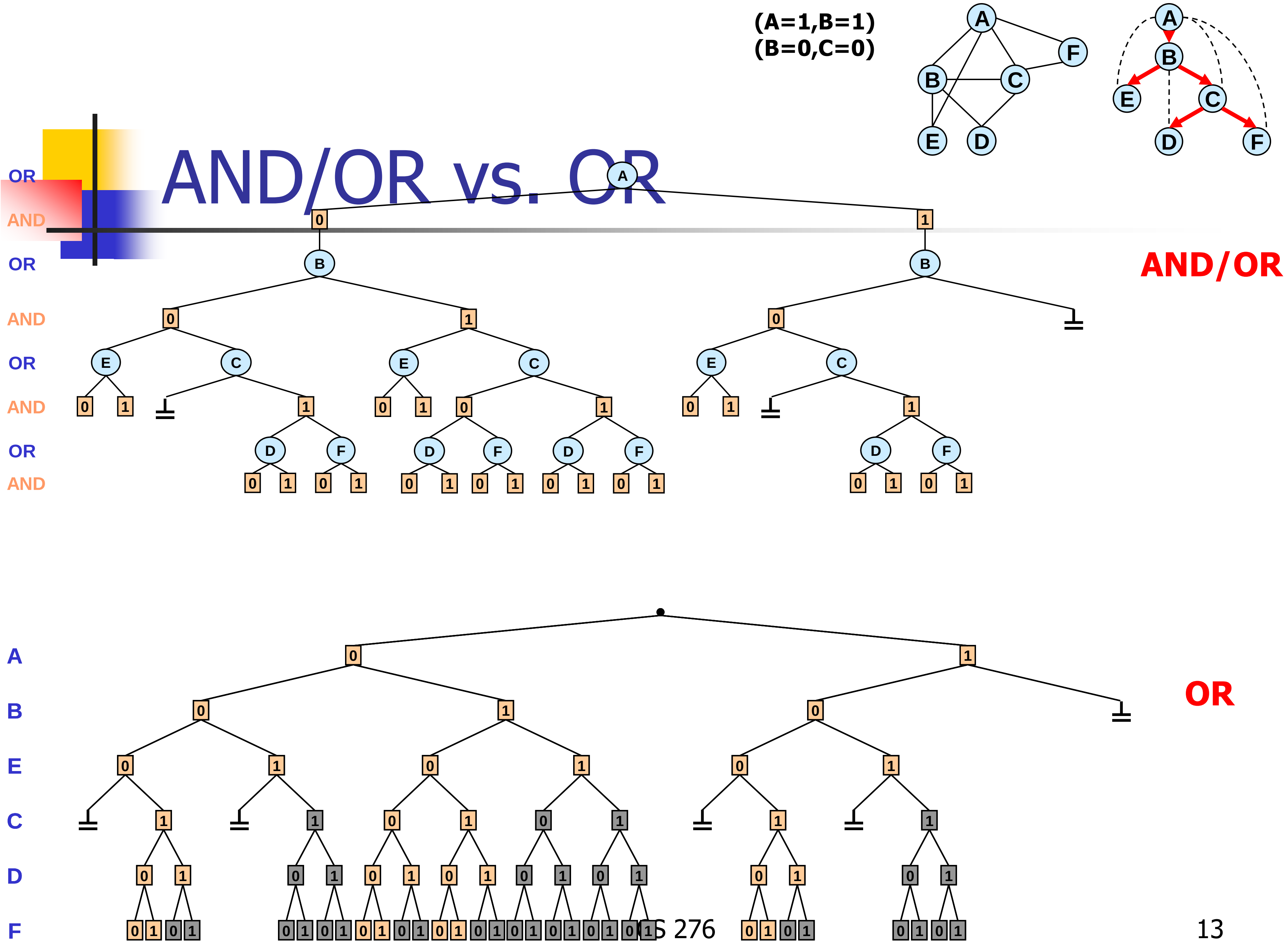
No-goods
(A=1,B=1)
(B=0,C=0)







AND/OR vs. OR

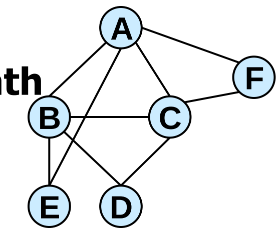


OR space vs. AND/OR space

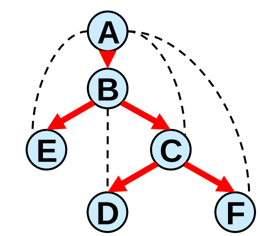
width	height	OR space			AND/OR space		
		time(sec.)	nodes	backtracks	time(sec.)	AND nodes	OR nodes
5	10	3.154	2,097,150	1,048,575	0.03	10,494	5,247
4	9	3.135	2,097,150	1,048,575	0.01	5,102	2,551
5	10	3.124	2,097,150	1,048,575	0.03	8,926	4,463
4	10	3.125	2,097,150	1,048,575	0.02	7,806	3,903
5	13	3.104	2,097,150	1,048,575	0.1	36,510	18,255
5	10	3.125	2,097,150	1,048,575	0.02	8,254	4,127
6	9	3.124	2,097,150	1,048,575	0.02	6,318	3,159
5	10	3.125	2,097,150	1,048,575	0.02	7,134	3,567
5	13	3.114	2,097,150	1,048,575	0.121	37,374	18,687
5	10	3.114	2,097,150	1,048,575	0.02	7,326	3,663

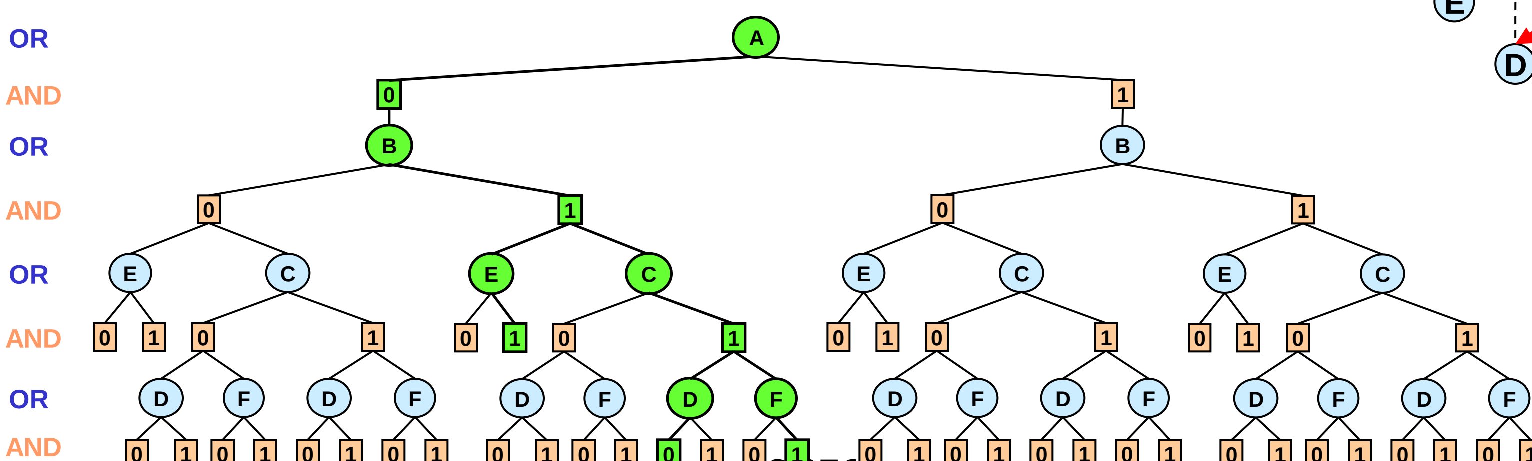
AND/OR search tree for graphical models

- **The AND/OR search tree of R relative to a spanning-tree, T, has:**
  - **Alternating levels of: OR nodes (variables) and AND nodes (values)**
- **Successor function:**
  - **The successors of OR nodes X are all its consistent values along its path**
  - **The successors of AND <X,v> are all X child variables in T**

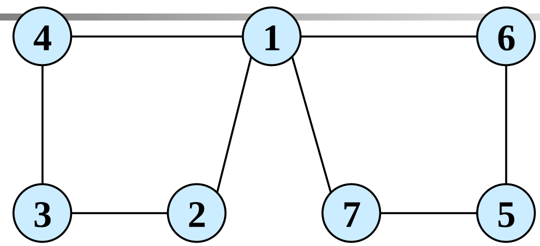


- **A solution is a consistent subtree**
- **Task: compute the value of the root node**

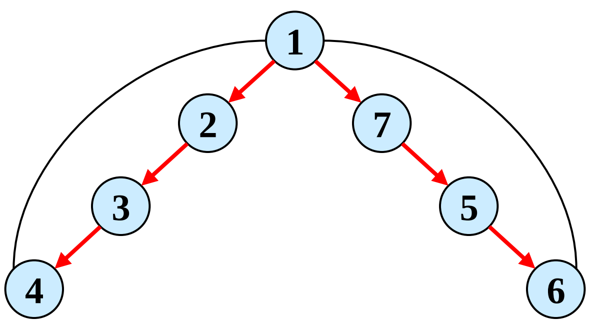




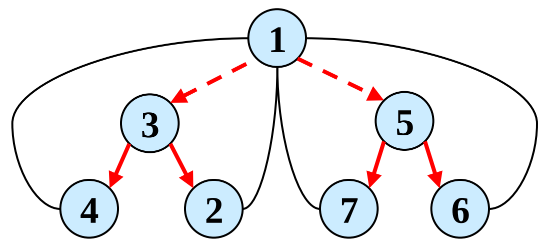
From DFS trees to pseudo-trees (Freuder 85, Bayardo 95)



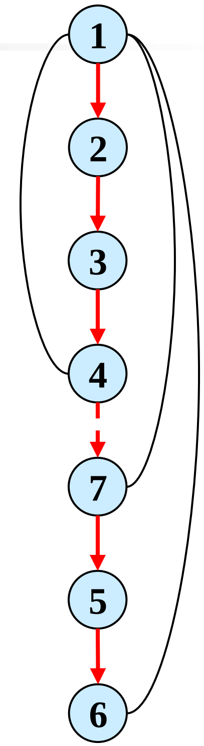
(a) Graph



(b) DFS tree
depth=3

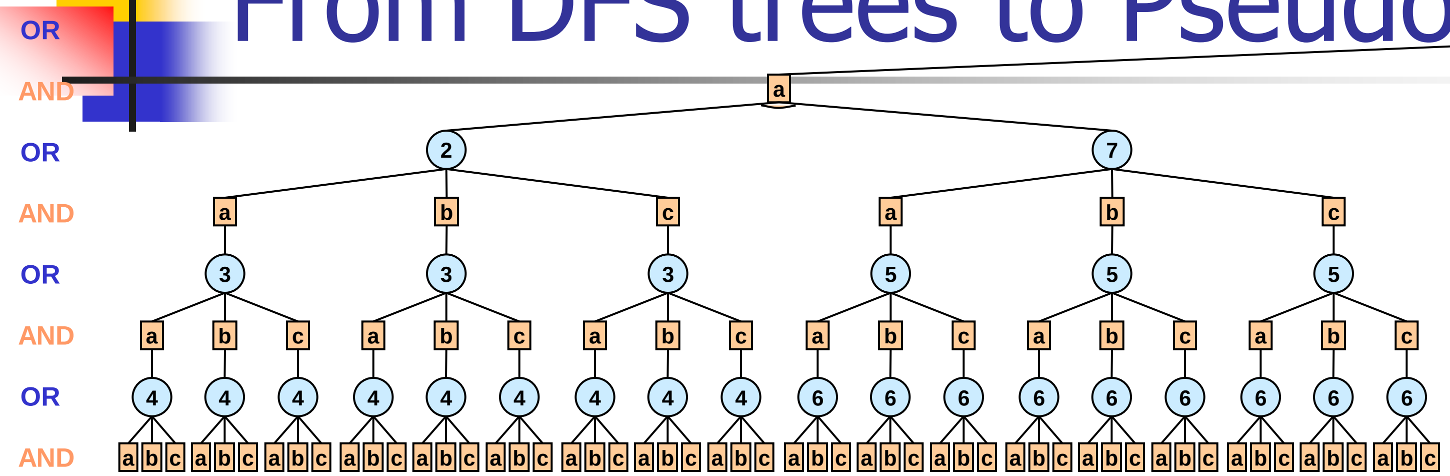


(c) pseudo- tree
depth=2



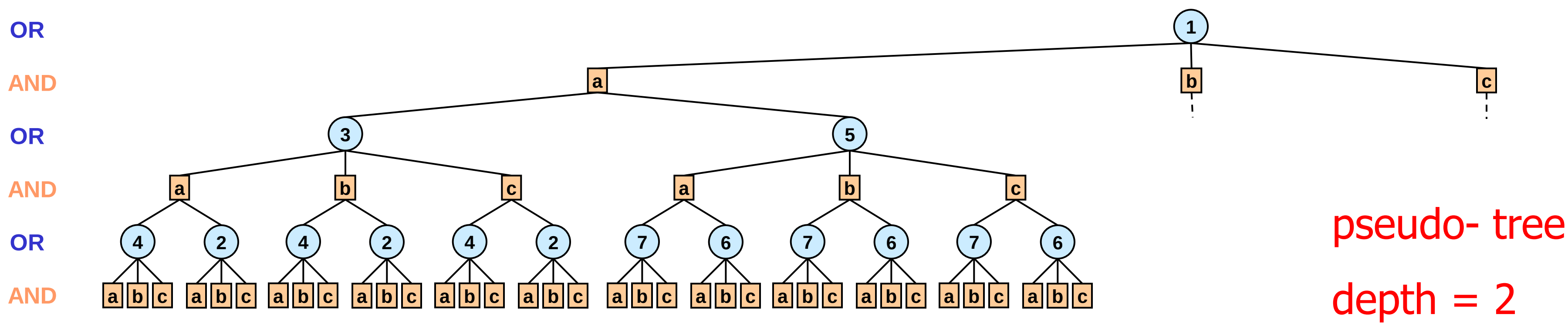
(d) Chain
depth=6

From DFS trees to Pseudo-trees



DFS tree

depth = 3



pseudo- tree

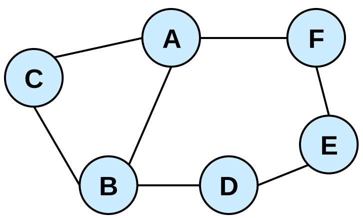
depth = 2

Finding min-depth backbone trees

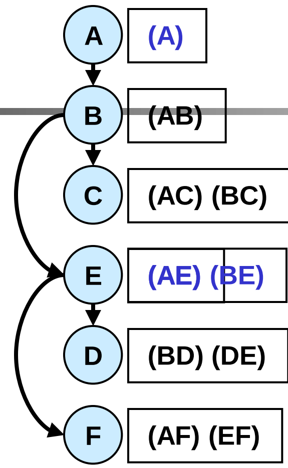
- Finding min depth DFS, or pseudo tree is NP-complete, but:
- Given a tree-decomposition whose tree-width is w*, there exists a pseudo tree T of G whose depth, satisfies (Bayardo and Mirankar, 1996, bodlaender and Gilbert, 91):

$$m <= w^* \log n,$$

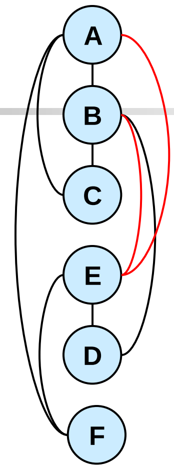
Generating pseudo-trees from Bucket trees



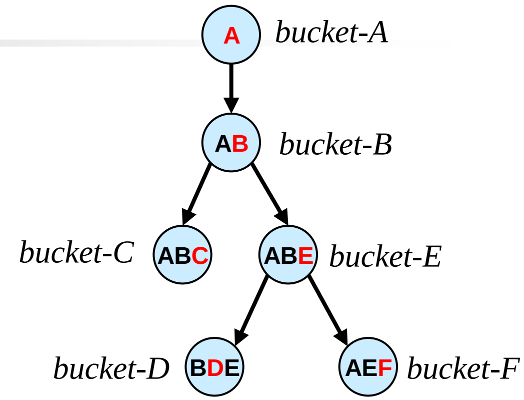
d: A B C E D F



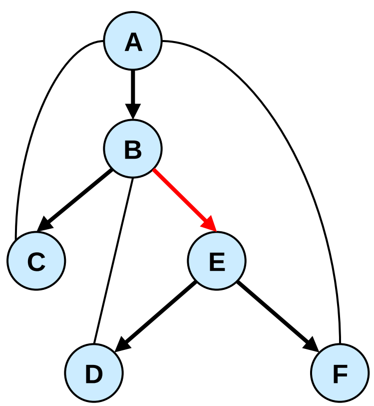
Bucket-tree based on *d*



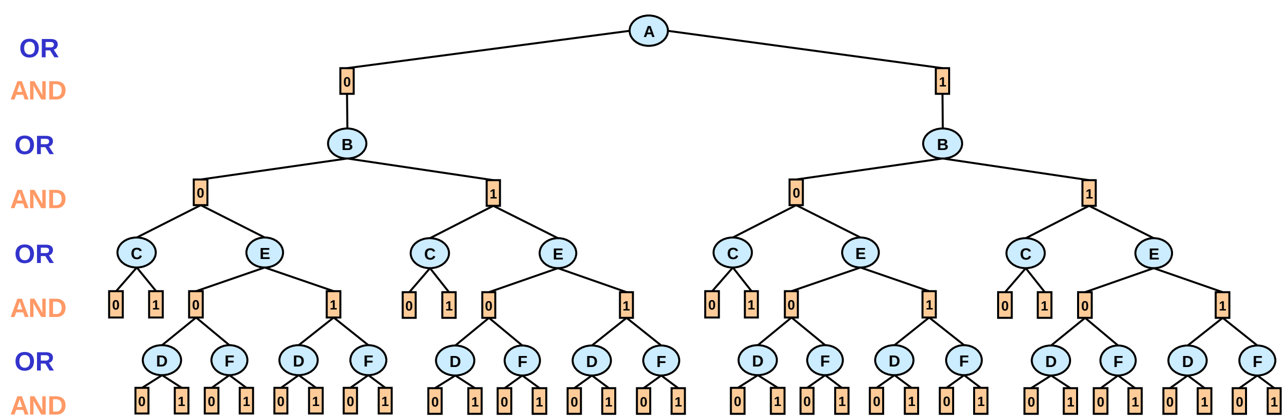
Induced graph



Bucket-tree



Bucket-tree used as pseudo-tree



AND/OR search tree

AND/OR Search-tree properties

(k **= domain size,** ***m*** **= pseudo-tree depth.** ***n*** **= number of variables)**

- **Theorem:** Any AND/OR search tree based on a pseudo-tree is sound and complete (expresses all and only solutions)

- **Theorem:** Size of AND/OR search tree is $O(n\ k^m)$

  Size of OR search tree is $O(k^n)$

- **Theorem:** Size of AND/OR search tree can be bounded by $O(\exp(w^* \log n))$

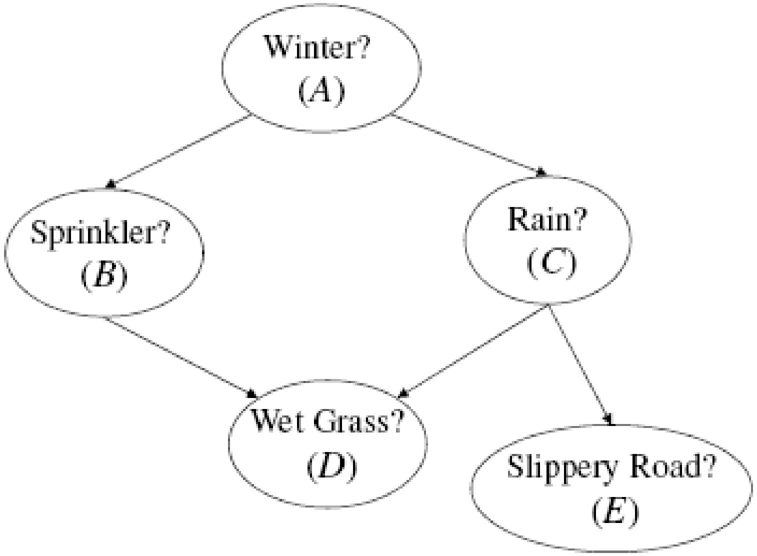
- Related to: (Freuder 85; Dechter 90, Bayardo et. al. 96, Darwiche 1999, Bacchus 2003)

- When the pseudo-tree is a chain we get an OR space

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: $O(\exp(w^* \log n))$**

A Bayesian Network



A	Θ_A
true	.6
false	.4

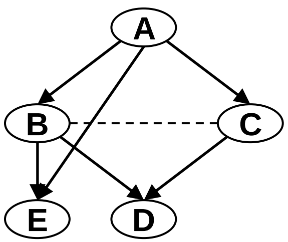
A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

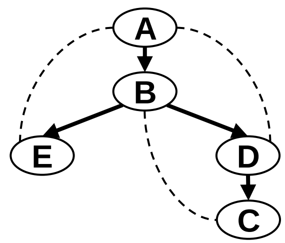
A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

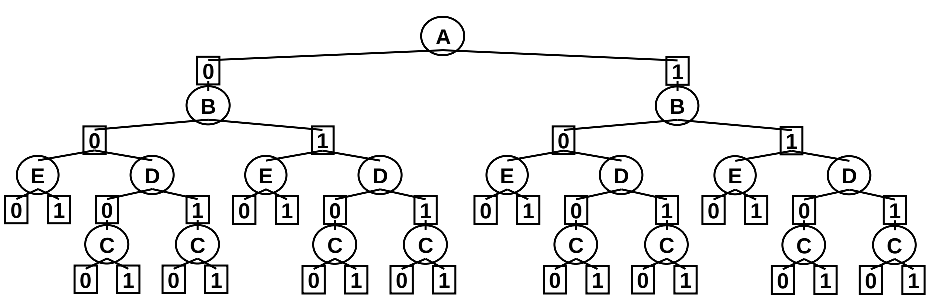
B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1

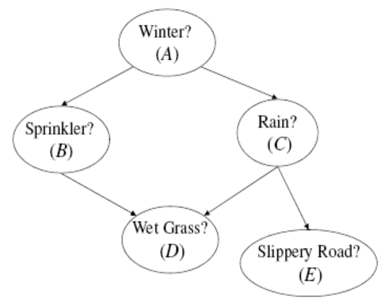
Belief-updating on example







A Bayesian Network



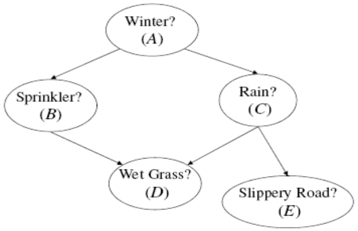
A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1



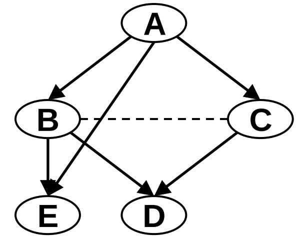
A	Θ_A
true	.6
false	.4

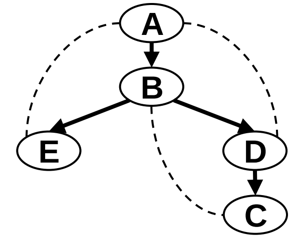
A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

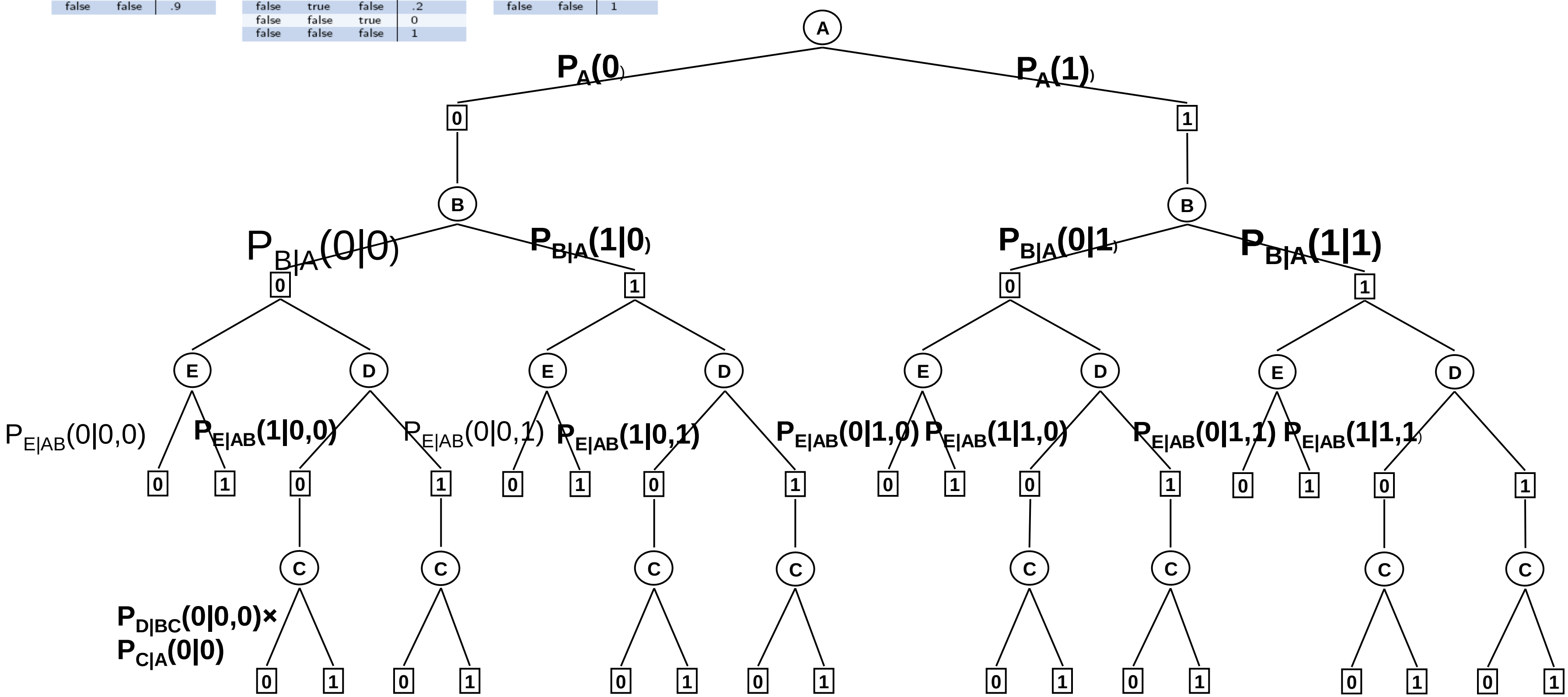
A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

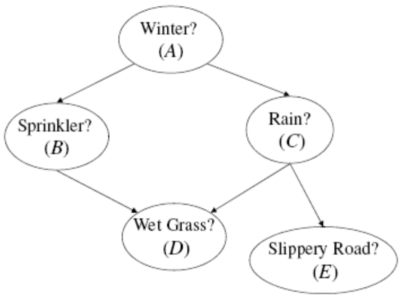
C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1







A Bayesian Network



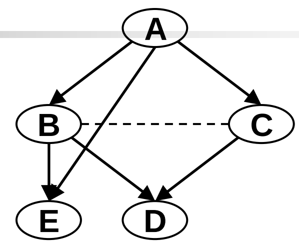
A	Θ_A
true	.6
false	.4

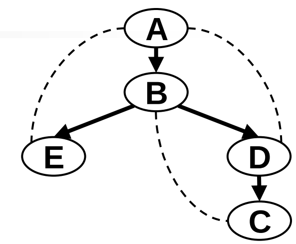
A	B	$\Theta_{B\|A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

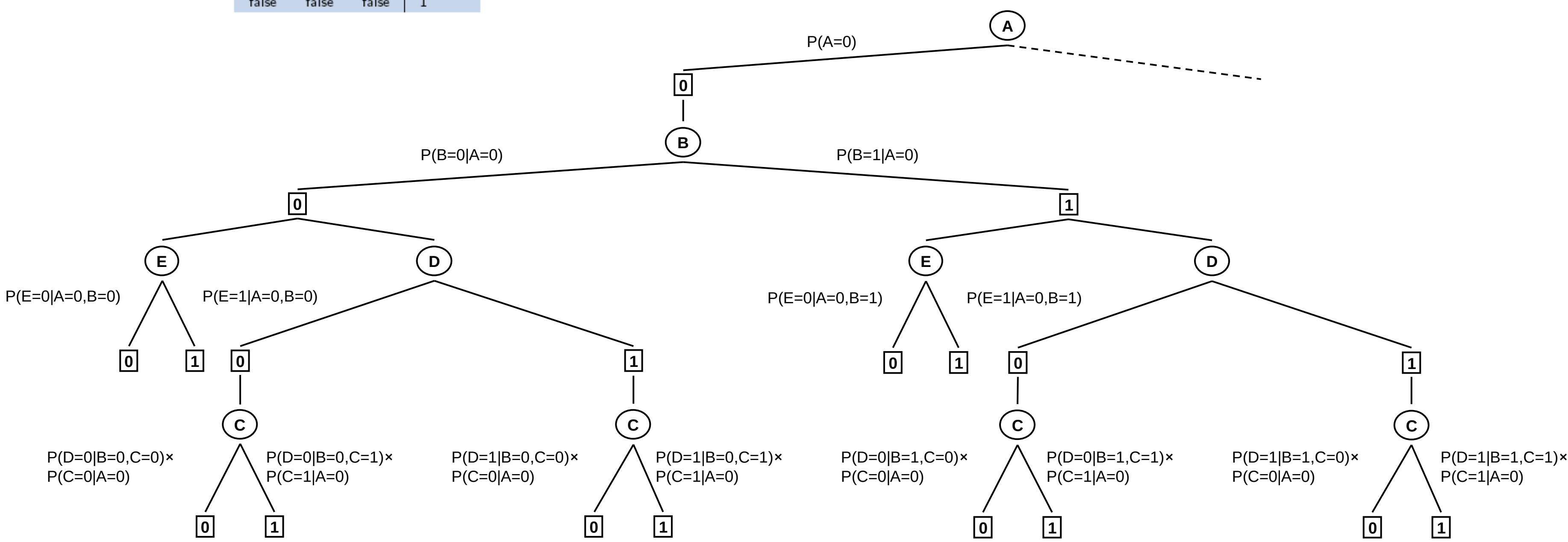
A	C	$\Theta_{C\|A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\|BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

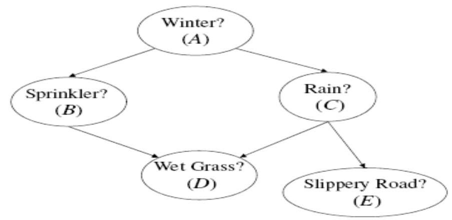
C	E	$\Theta_{E\|C}$
true	true	.7
true	false	.3
false	true	0
false	false	1







A Bayesian Network



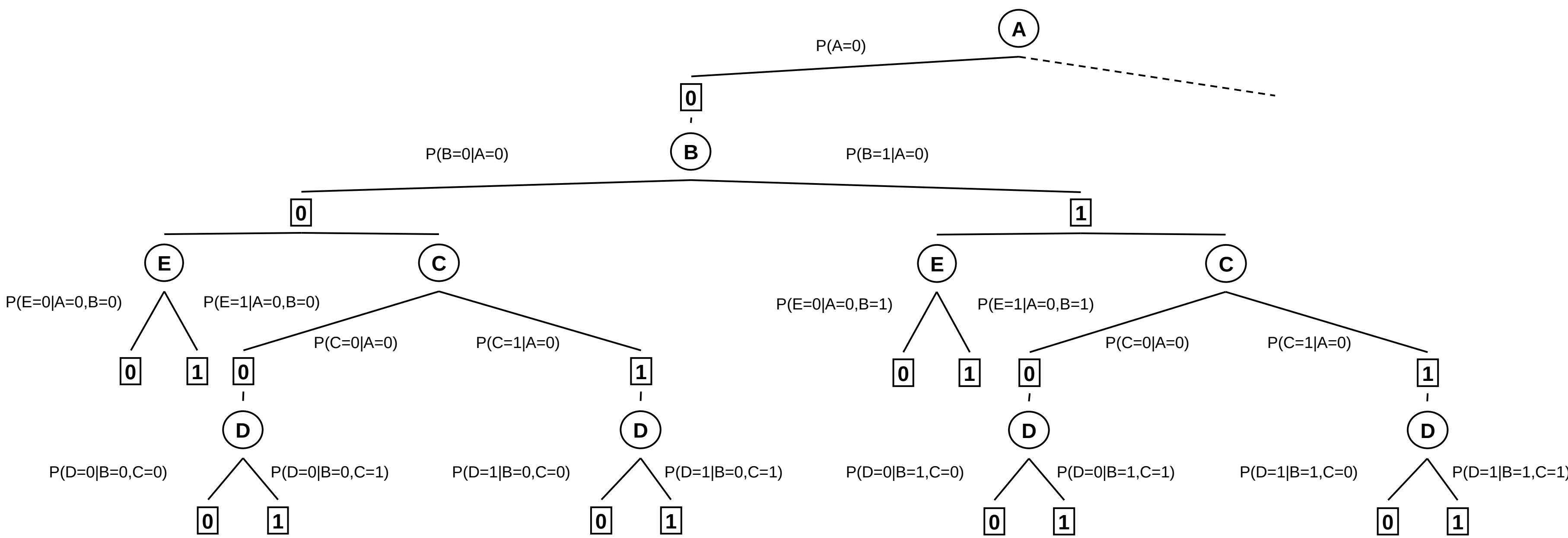
A	Θ_A
true	.6
false	.4

A	B	$\Theta_{B\mid A}$
true	true	.2
true	false	.8
false	true	.75
false	false	.25

A	C	$\Theta_{C\mid A}$
true	true	.8
true	false	.2
false	true	.1
false	false	.9

B	C	D	$\Theta_{D\mid BC}$
true	true	true	.95
true	true	false	.05
true	false	true	.9
true	false	false	.1
false	true	true	.8
false	true	false	.2
false	false	true	0
false	false	false	1

C	E	$\Theta_{E\mid C}$
true	true	.7
true	false	.3
false	true	0
false	false	1



(d)

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

$P(B \mid A)$

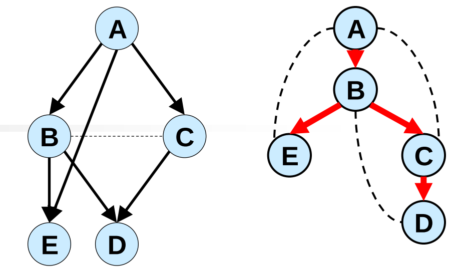
A	B=0	B=1
0	.4	.6
1	.1	.9

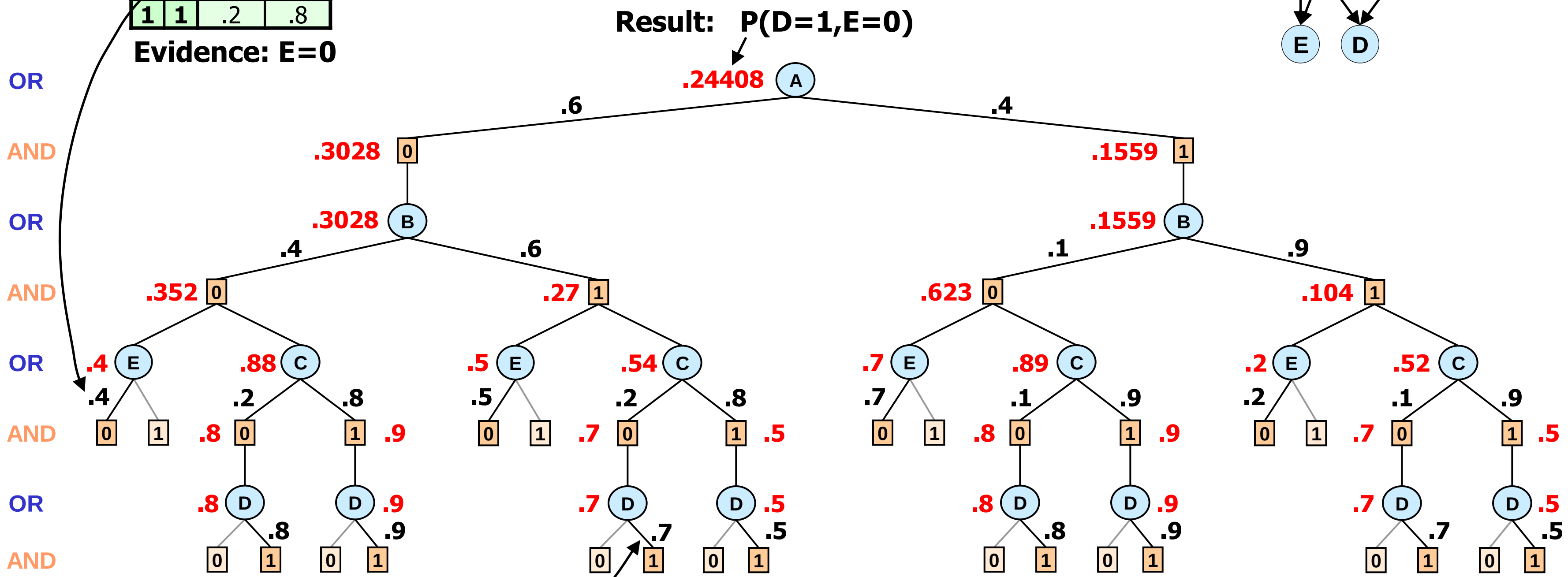
$P(C \mid A)$

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

$P(A)$

A	P(A)
0	.6
1	.4





$P(D \mid B, C)$

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

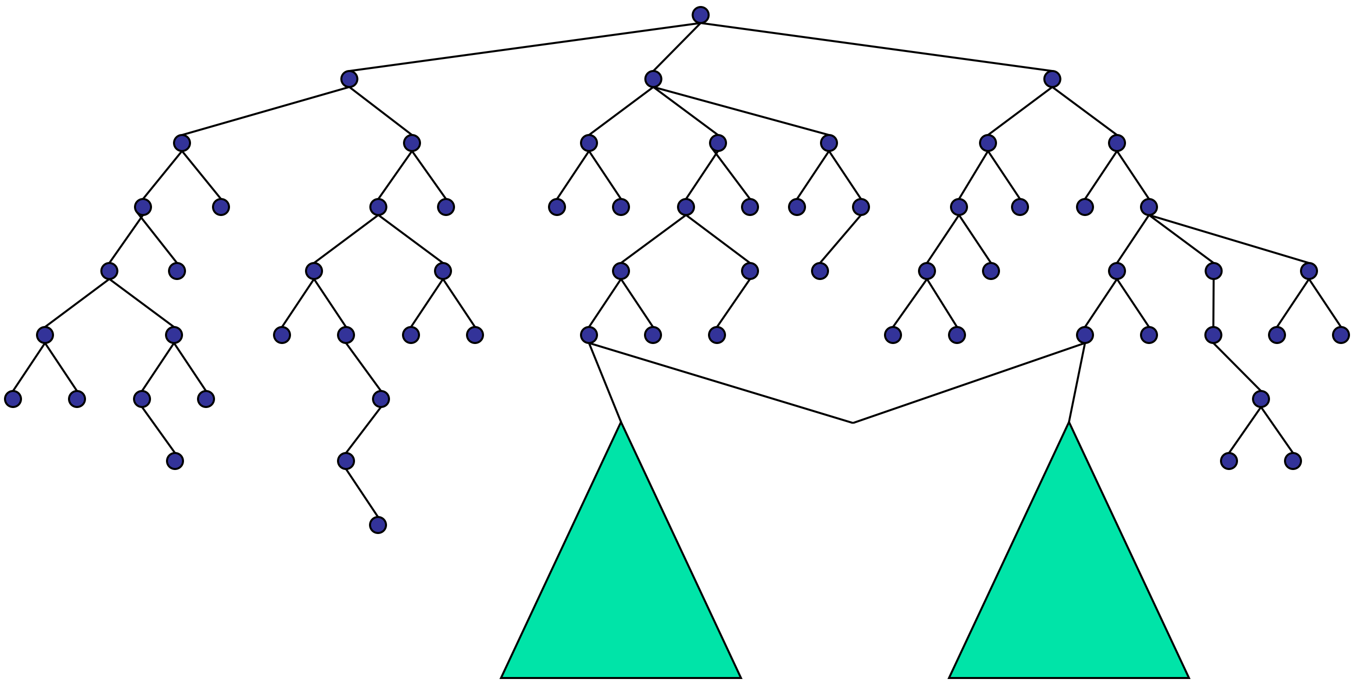
Complexity of AND/OR Tree Search

	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

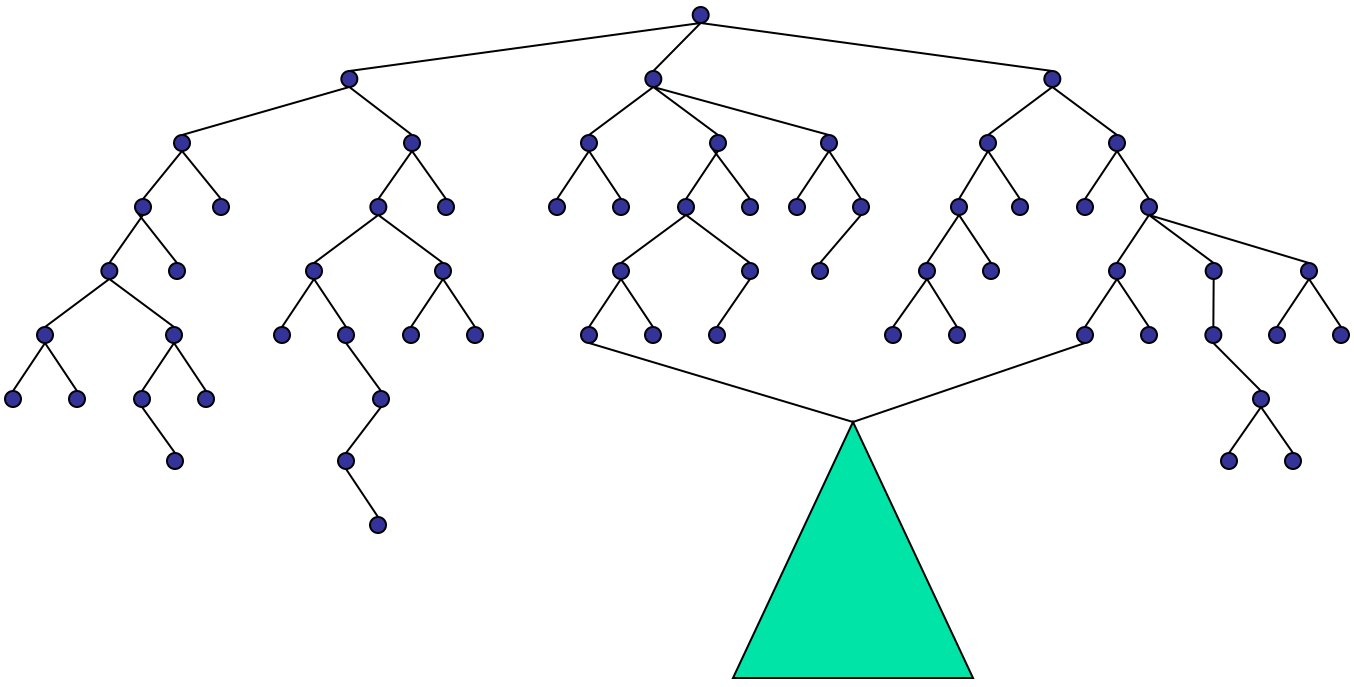
From Search Trees to Search Graphs

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

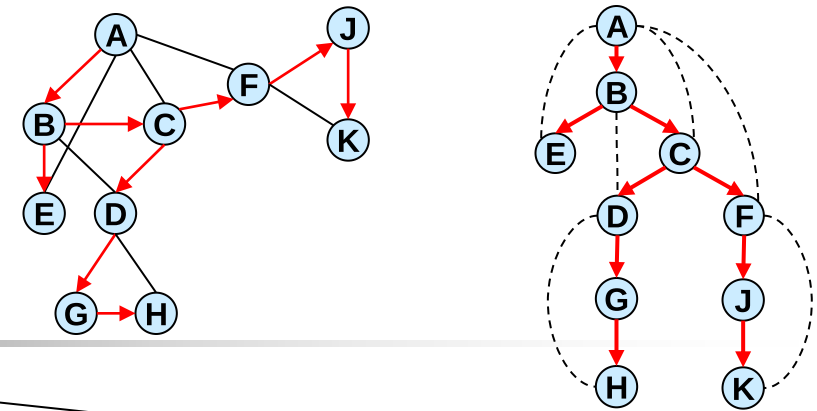


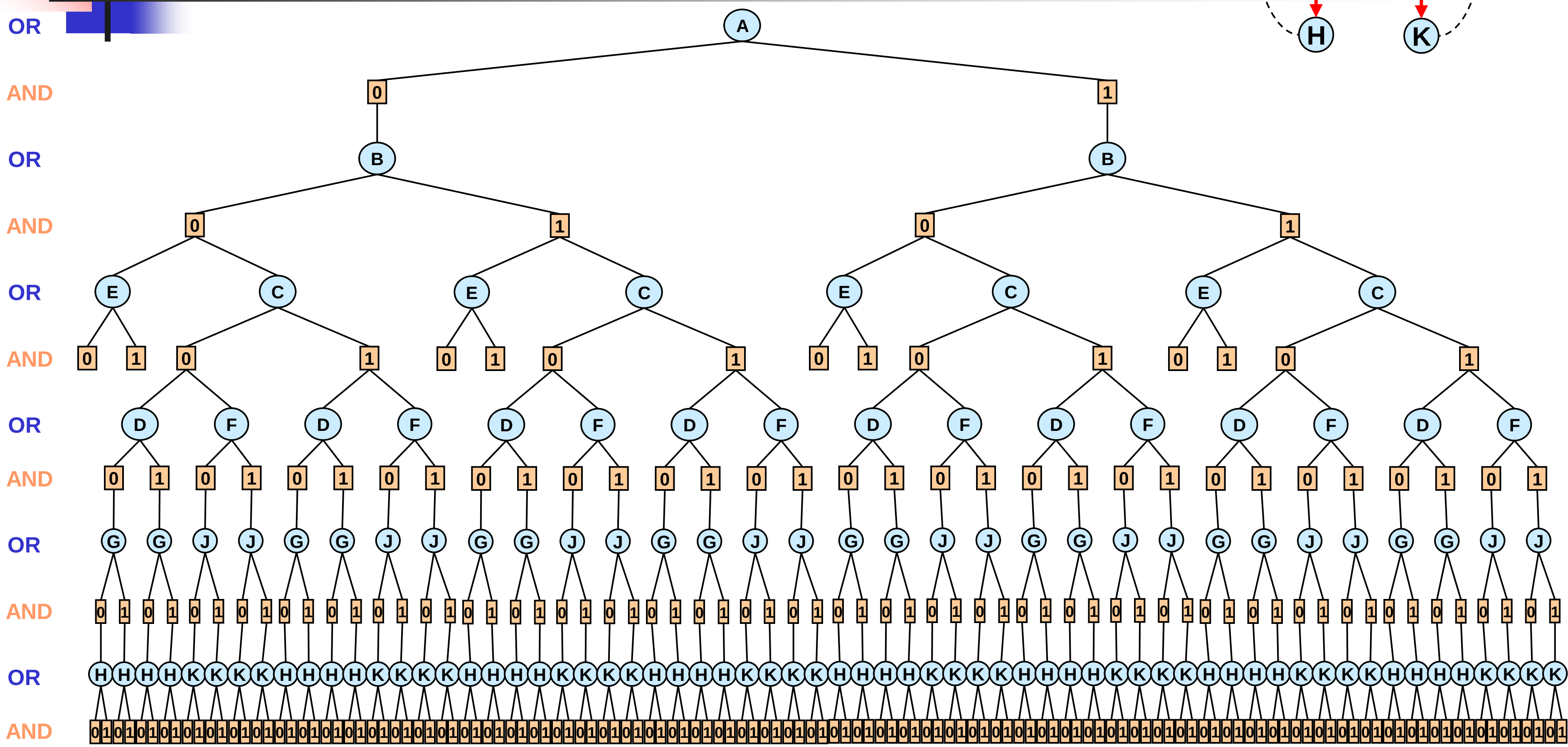
From Search Trees to Search Graphs

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

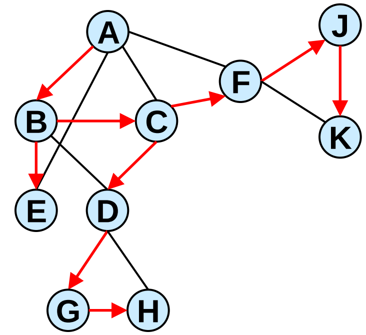


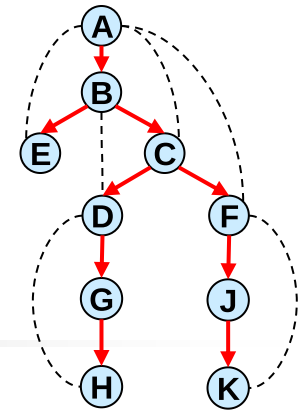
AND/OR Tree

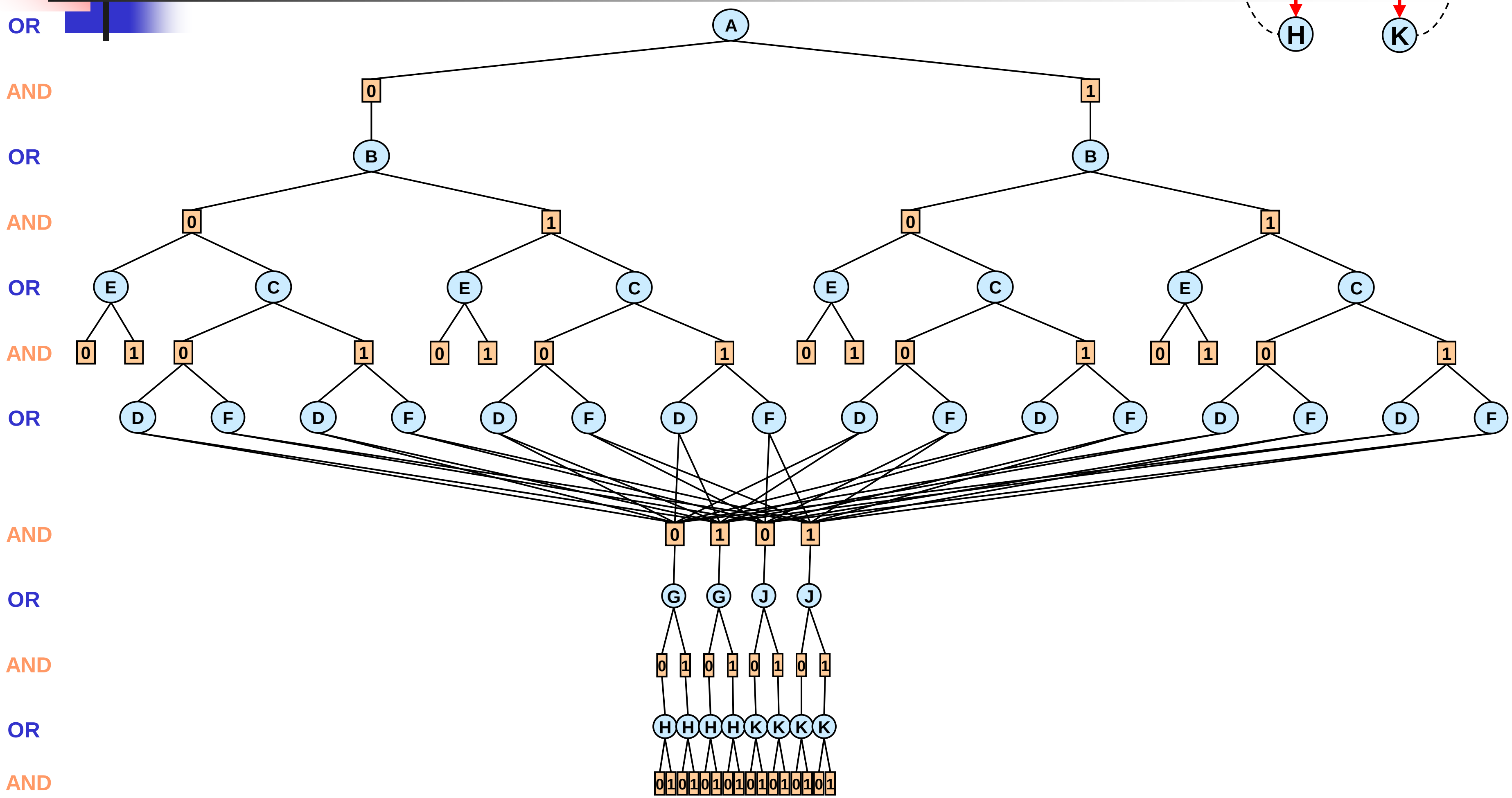




An AND/OR graph



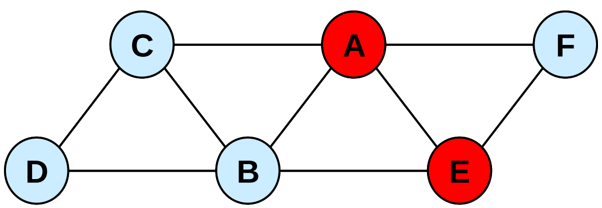


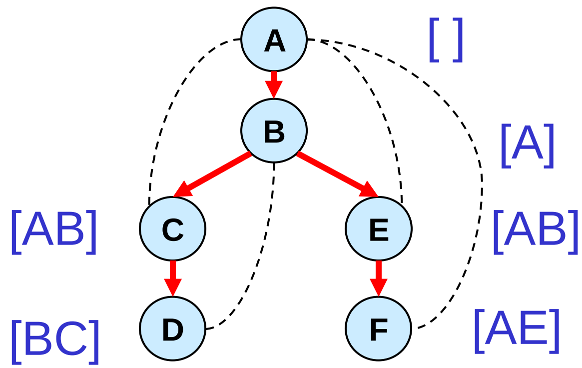


Merging Based on Context

One way of recognizing nodes that can be merged:

context (X) = ancestors of X in pseudo tree that are connected to X, or to descendants of X

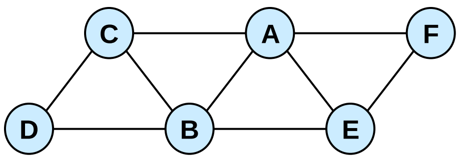




pseudo tree

AND/OR Search Graph

Constraint Satisfaction – Counting Solutions

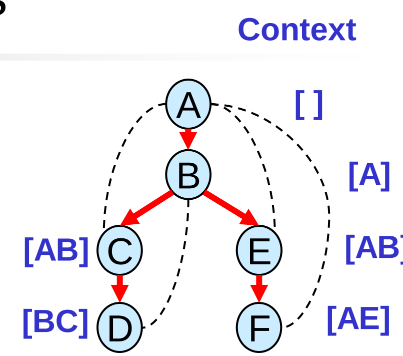


A	B	C	R_{ABC}
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

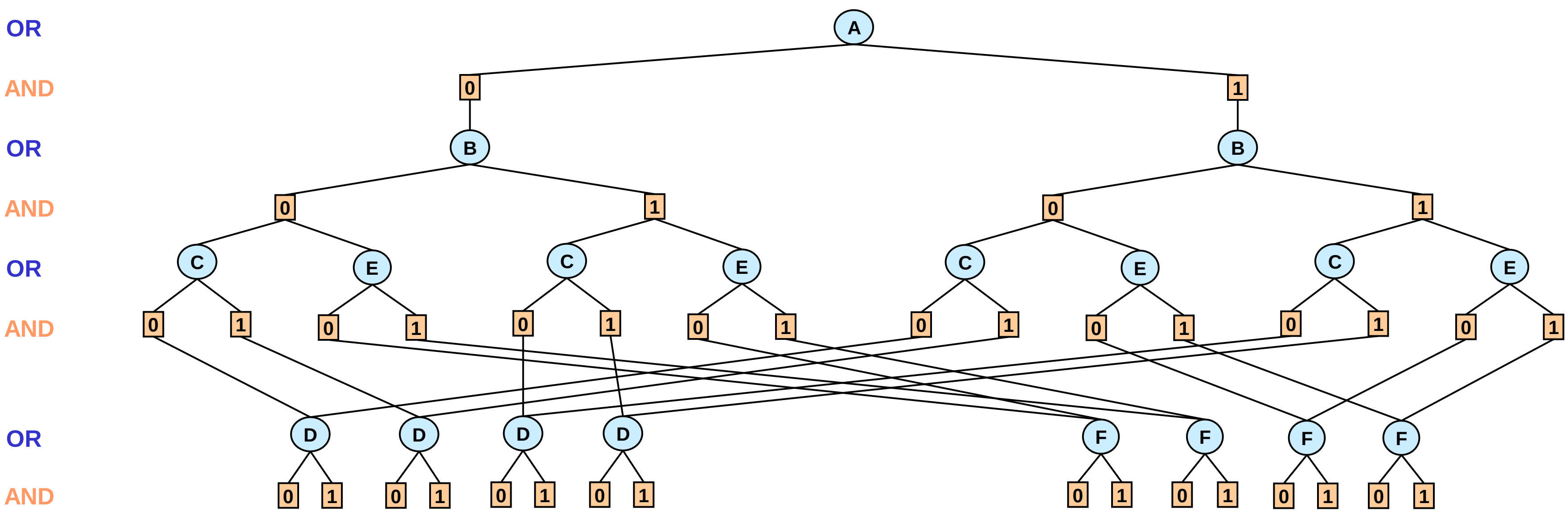
B	C	D	R_{BCD}
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

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



pseudo tree



context minimal graph

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

$P(B \mid A)$

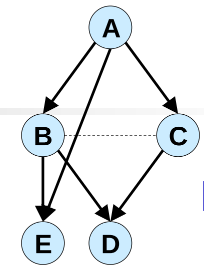
A	B=0	B=1
0	.4	.6
1	.1	.9

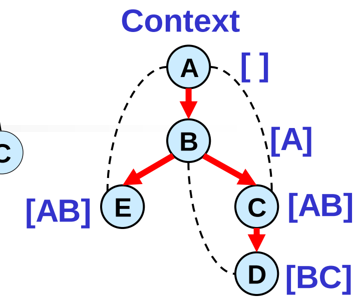
$P(C \mid A)$

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

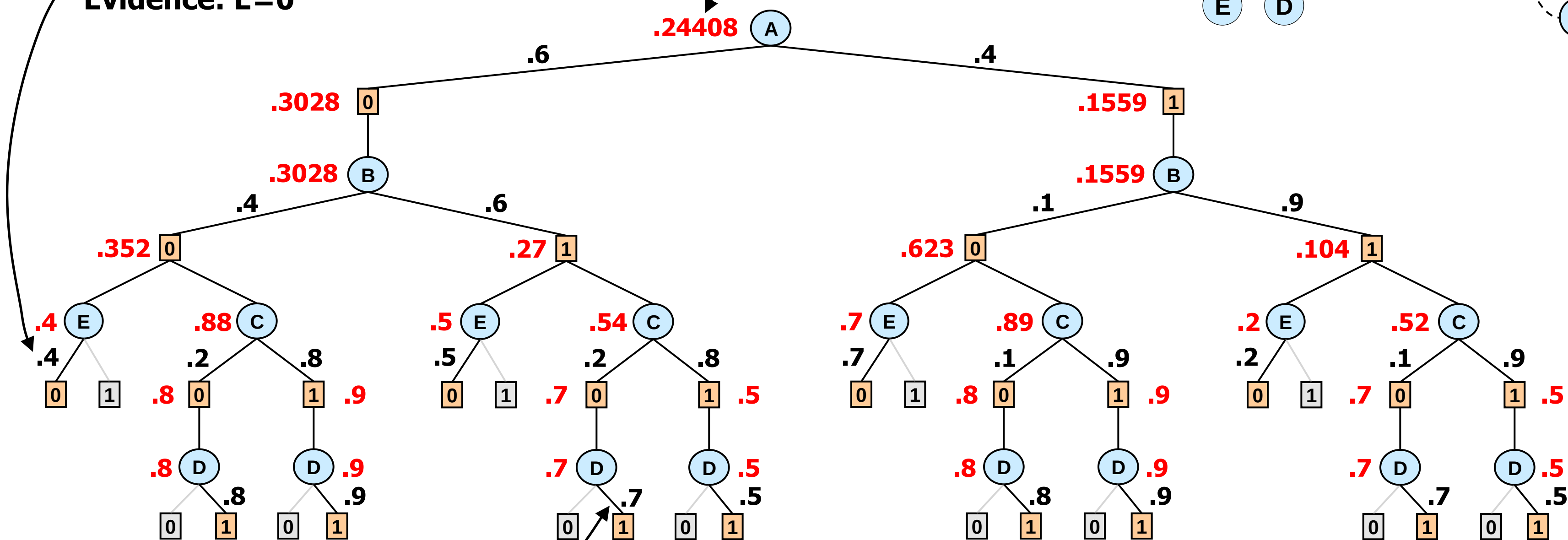
$P(A)$

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

$P(B \mid A)$

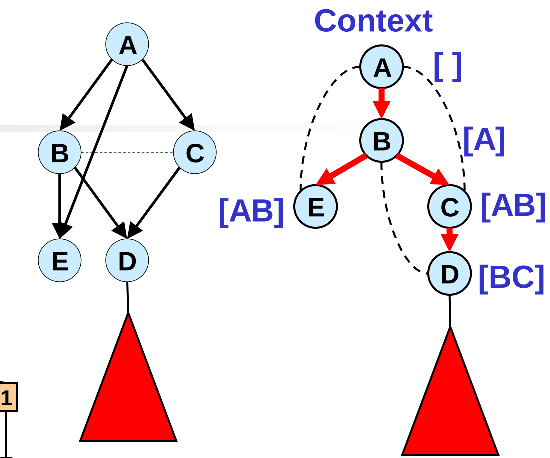
A	B=0	B=1
0	.4	.6
1	.1	.9

$P(C \mid A)$

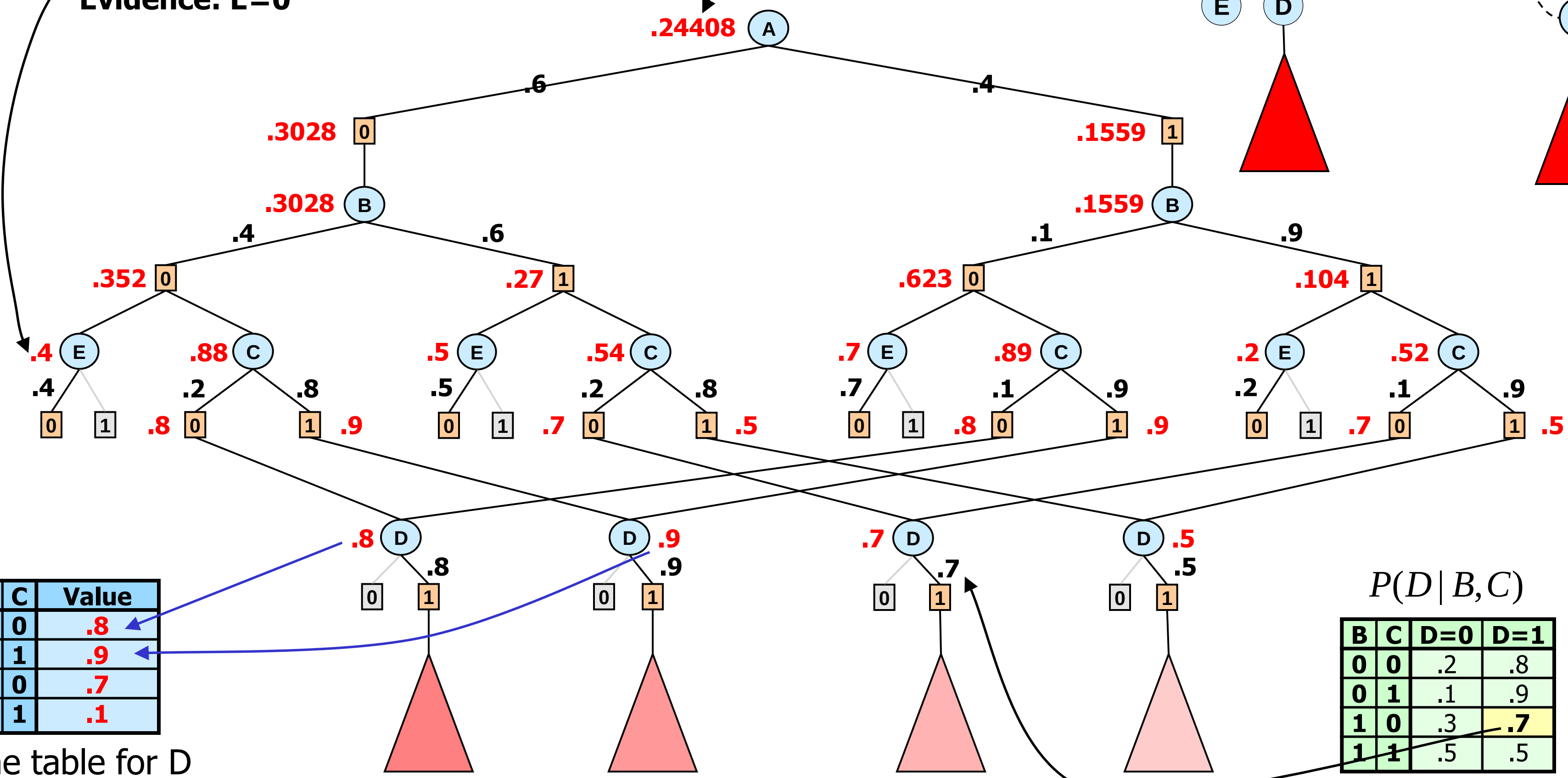
A	C=0	C=1
0	.2	.8
1	.7	.3

$P(A)$

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

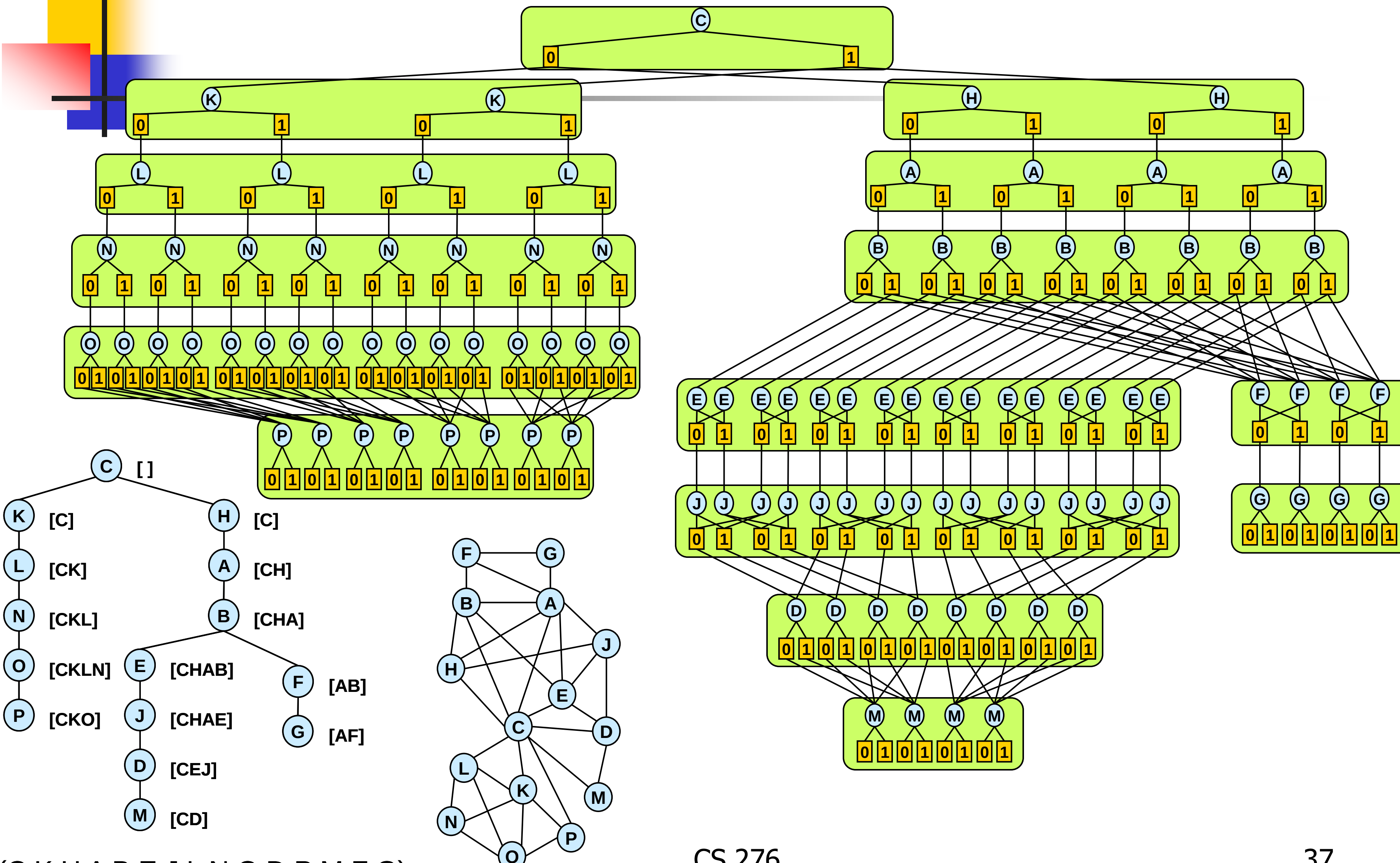
Cache table for D

$P(D \mid B,C)$

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

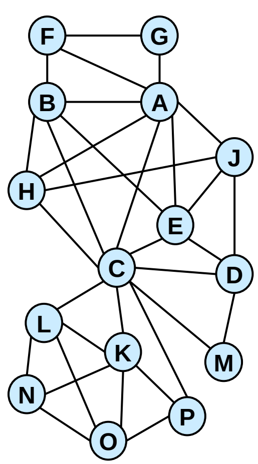
AND/OR context minimal graph

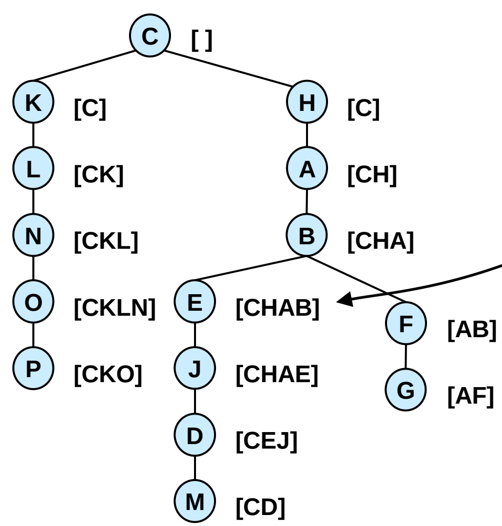


(C K H A B E J L N O D P M F G)

How Big Is the Context?

Theorem: *The maximum context size for a pseudo tree is equal to the treewidth of the graph along the pseudo tree.*

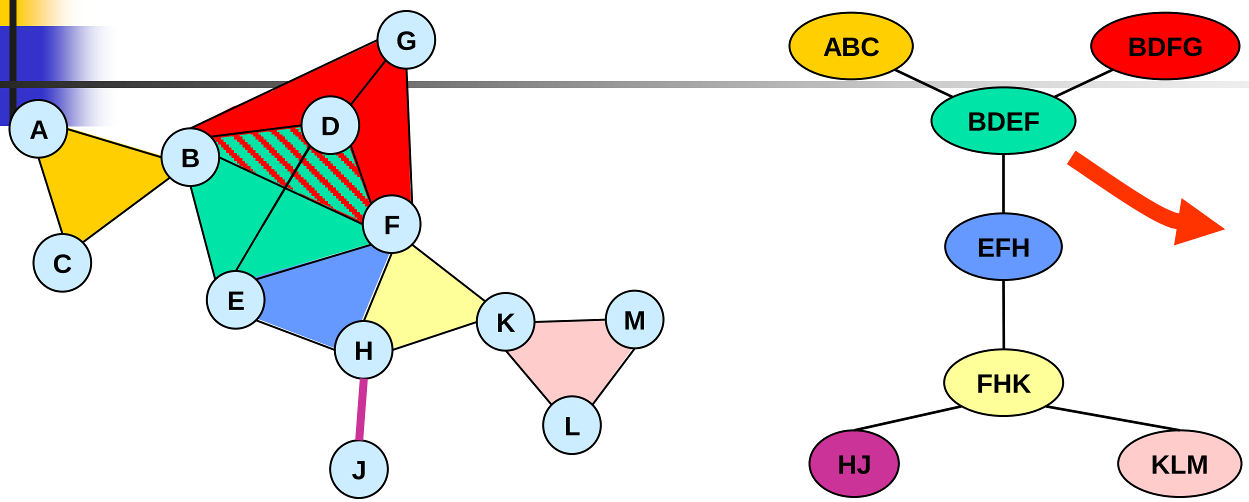




max context size = treewidth

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

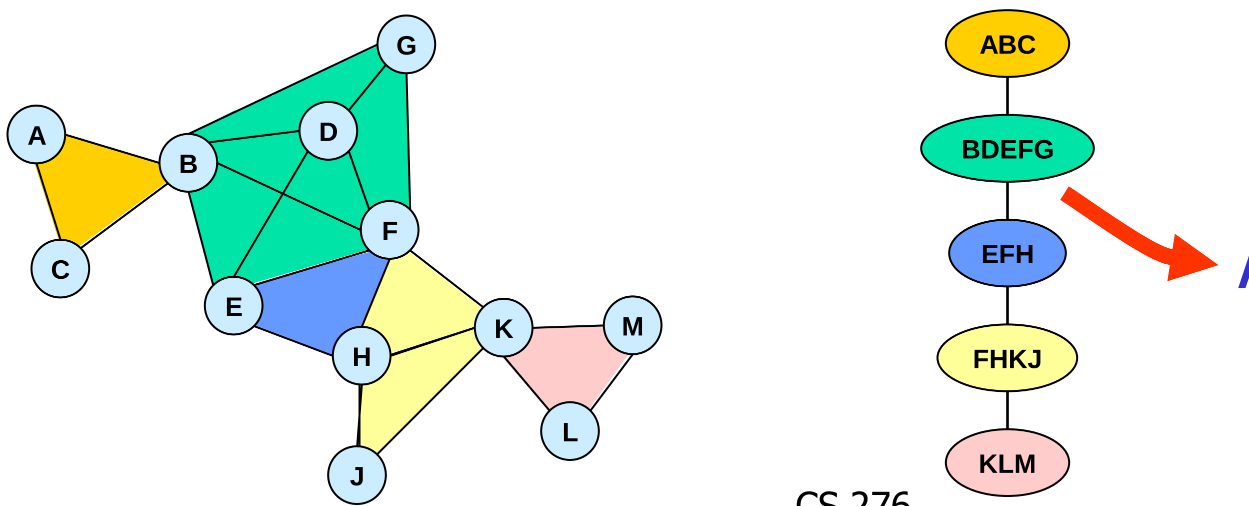
Treewidth vs. Pathwidth



TREE

***treewidth* = 3**

= (max cluster size) - 1



CHAIN

***pathwidth* = 4**

= (max cluster size) - 1

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

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

AND/OR search algorithms

- AO(i)
  - i = the max size of a cache table (i.e. number of variables in a context)

$i = 0$ → $i = w^*$ treewidth

	$i = 0$	$i = w^*$
Space:	$O(n)$	$O(n\ \mathit{exp}\ w^*)$
Time:	$O(\mathit{exp}\ (w^*\ \mathit{log}\ n))$	$O(n\ \mathit{exp}\ w^*)$

Searching AND/OR Graphs

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

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

Space: O(n)

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

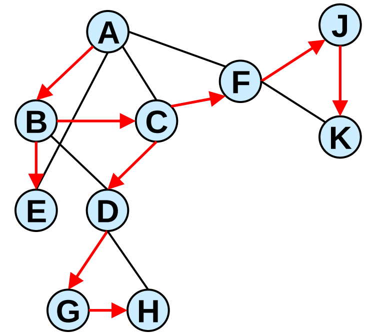
Space: O(exp(i))

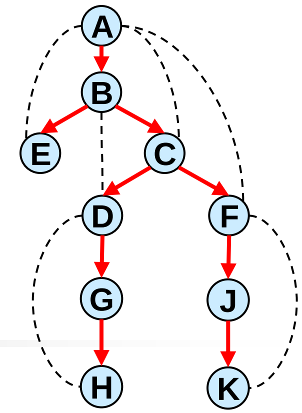
Time: O(exp(m_i+i)

Space: O(exp w*)

Time: O(exp w*)

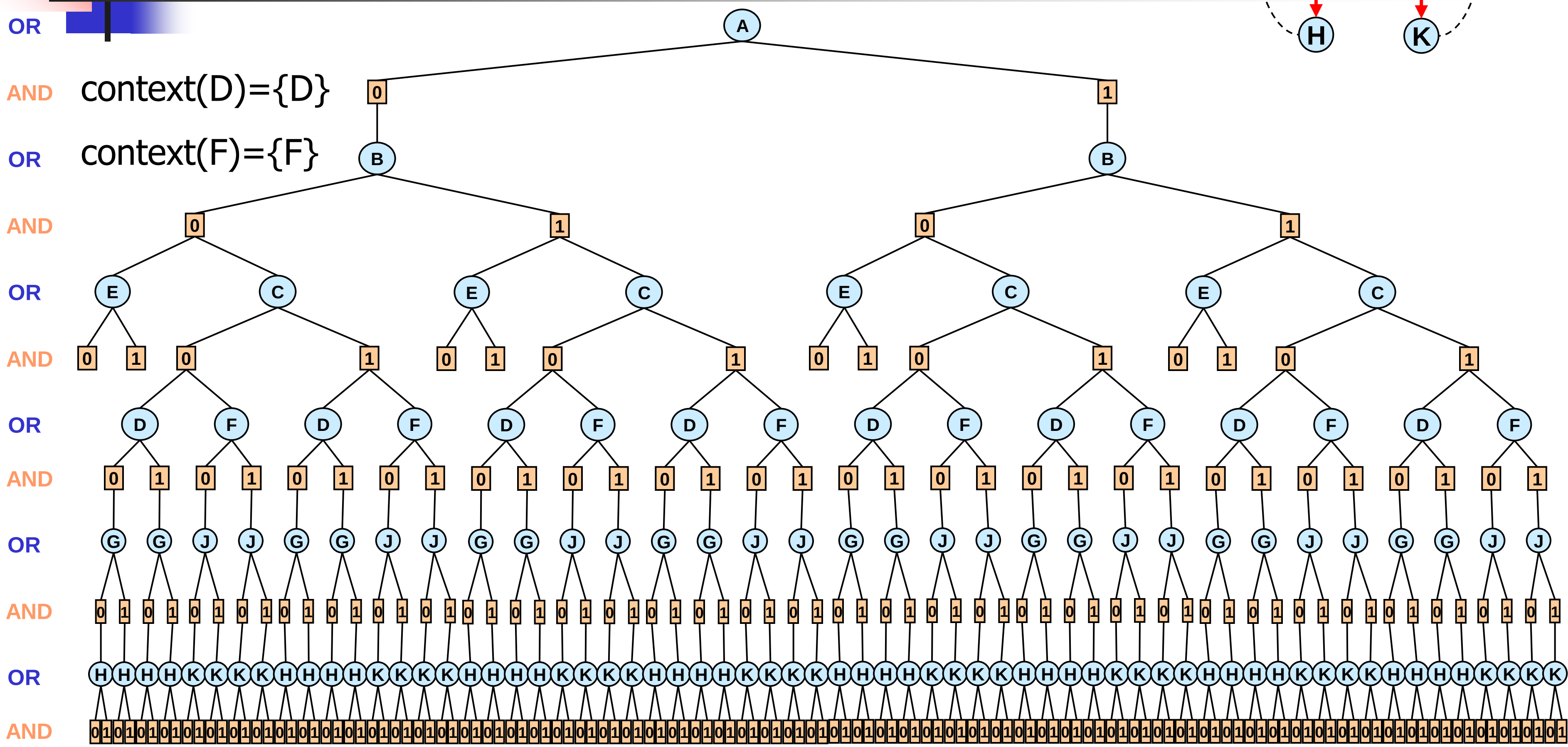
Caching

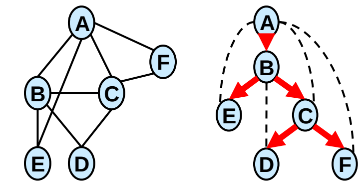




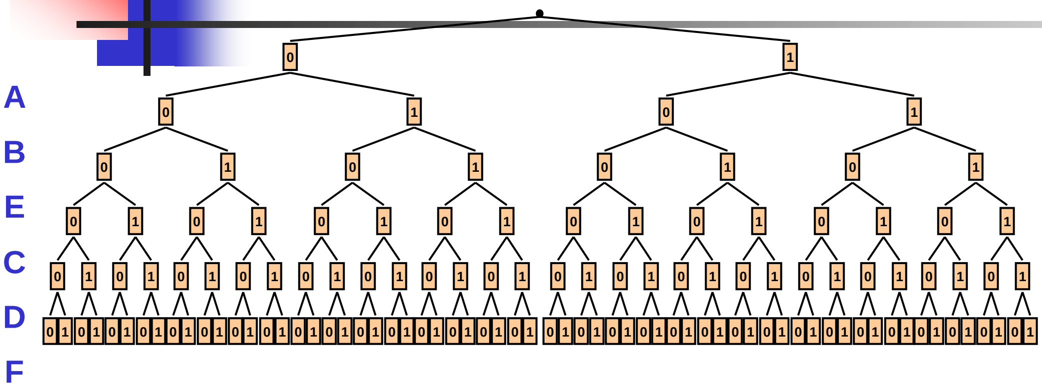
context(D)={D}

context(F)={F}

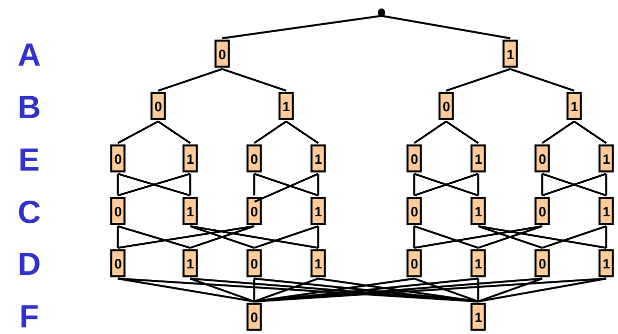




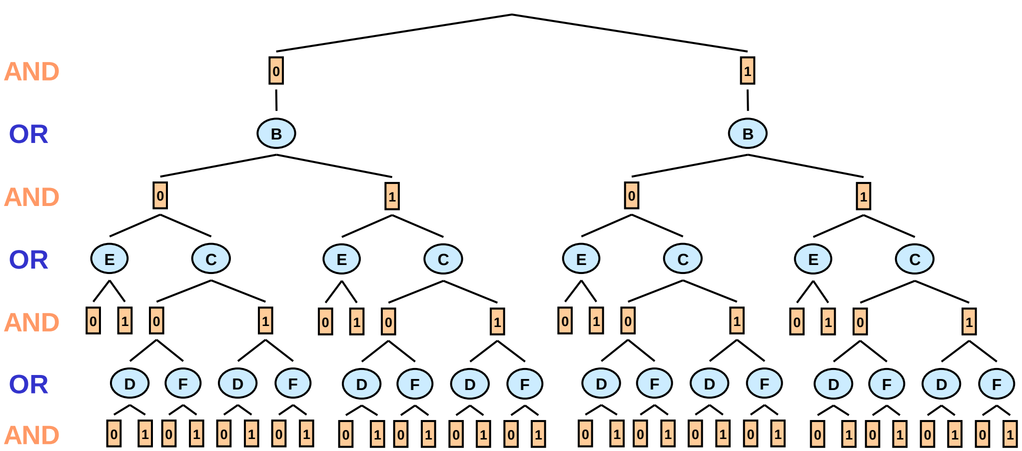
All four search spaces



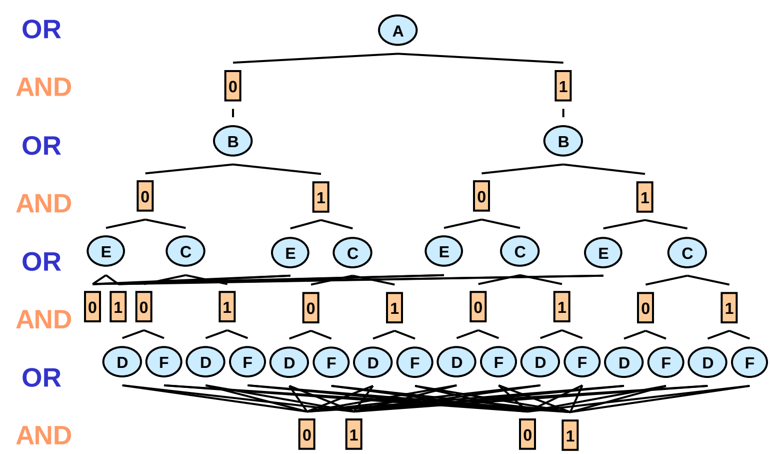
Full OR search tree



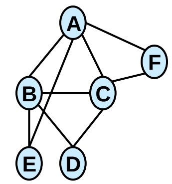
Context minimal OR search graph

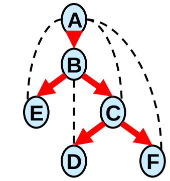


Full AND/OR search tree

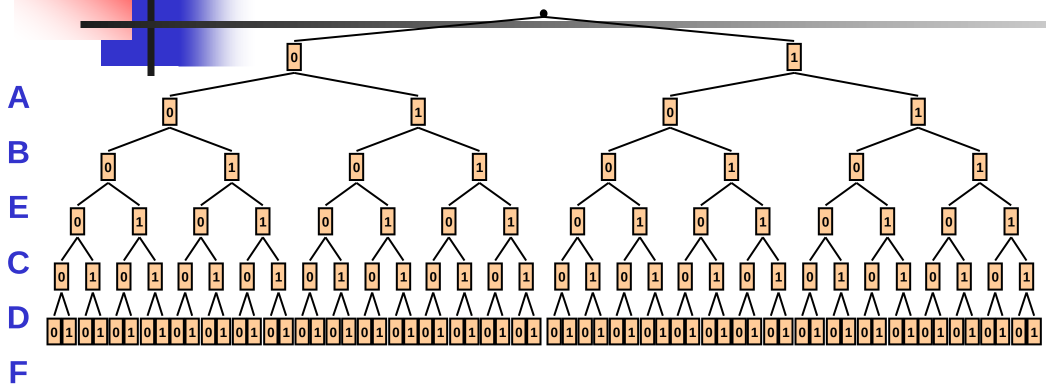


Context minimal AND/OR search graph

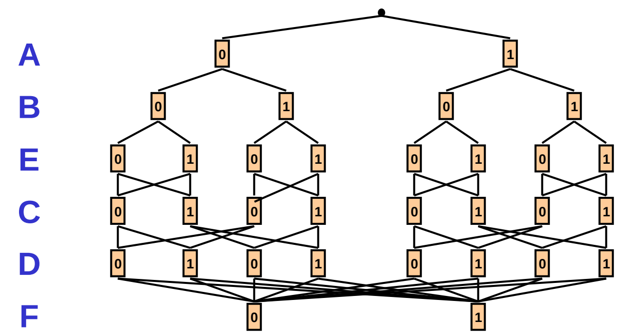




All four search spaces

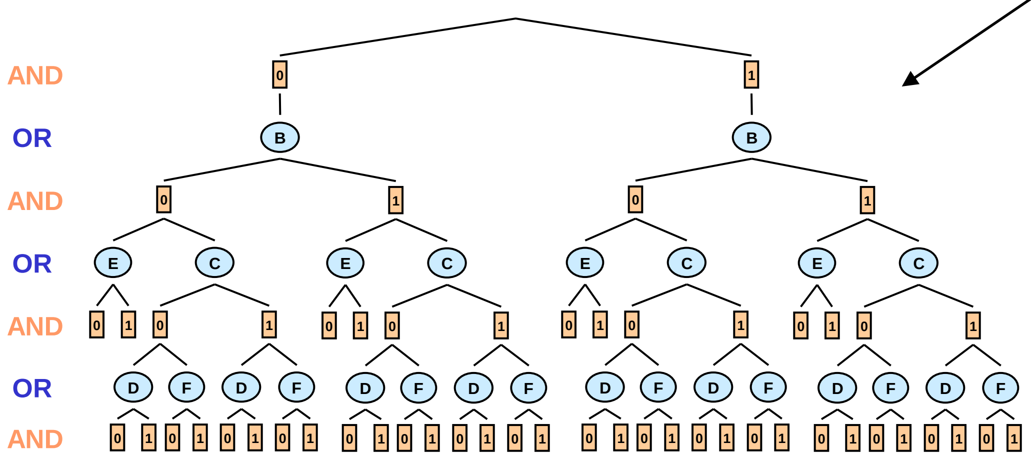


Full OR search tree

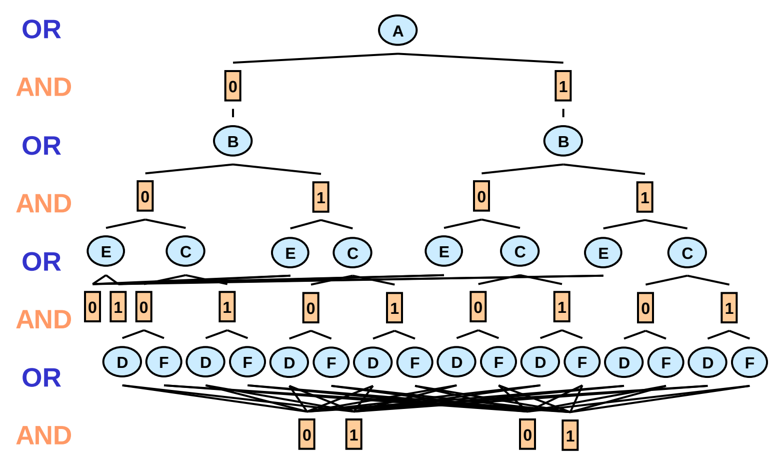


Context minimal OR search graph

Time-
space



Full AND/OR search tree



Context minimal AND/OR search graph

Available code

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