Introduction

- Combining two frameworks
  - AND/OR Search Spaces
  - Multi-valued Decision Diagrams (MDDs)
- Both are more compact ways to represent problems.
- Their combination yields an even more compact representation.
- Decision Diagrams are known to allow online speed queries.

Introduction

$F(ACF)$

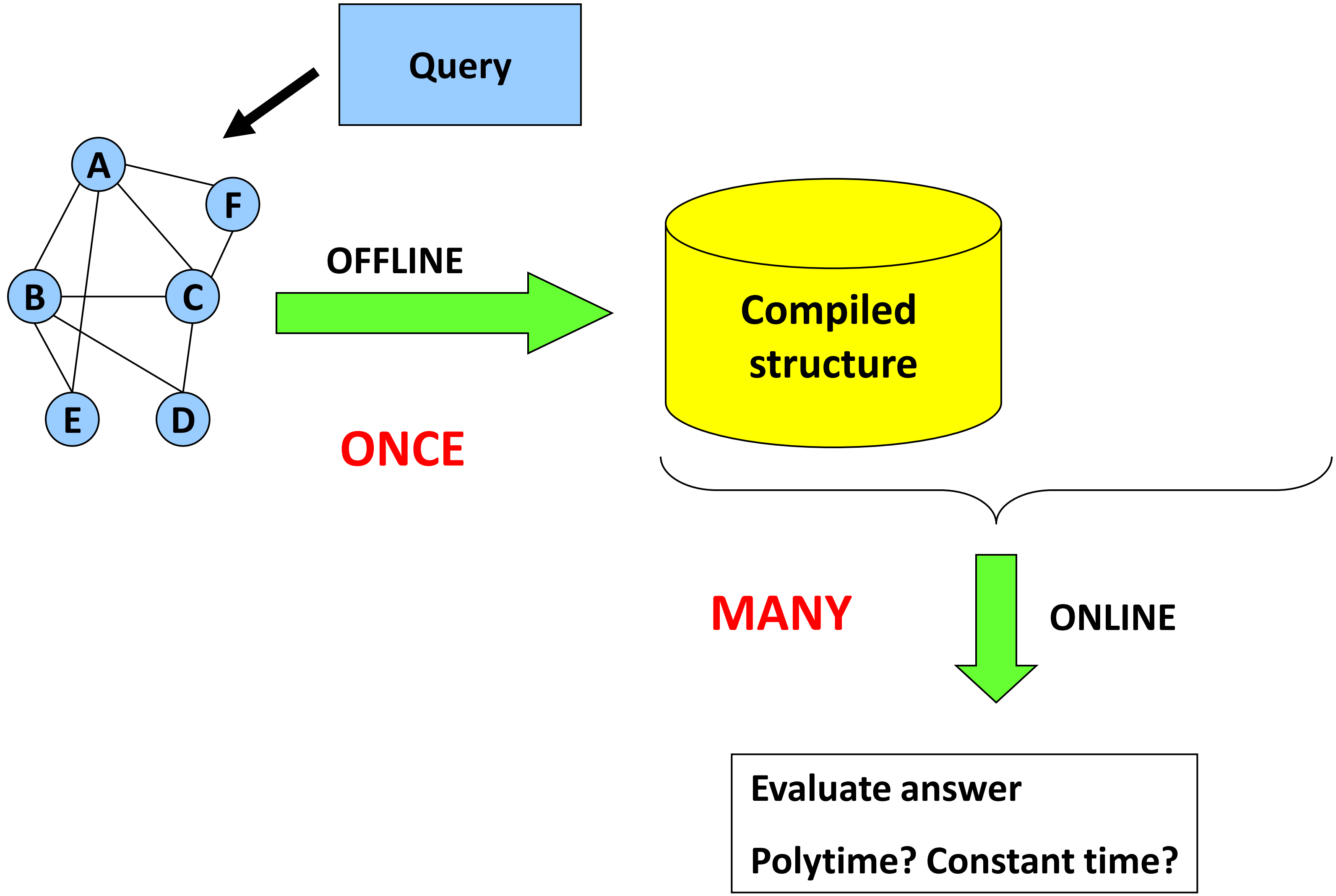
$F(ABE)$

$F(BC)$

$F(BD)$

$F(CD)$

$F(ABC)$

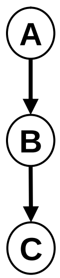


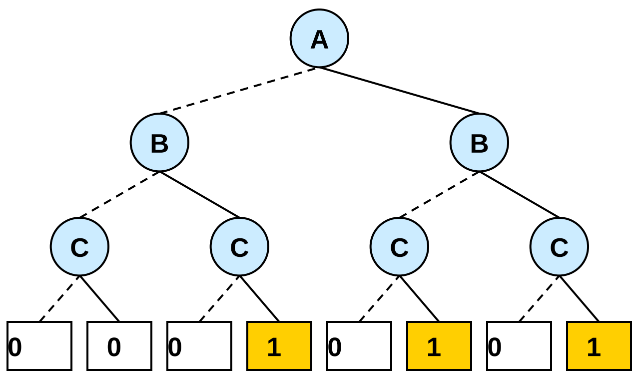
Ordered Binary Decision Diagram

B = {0,1}   f : $B^3 \rightarrow B$

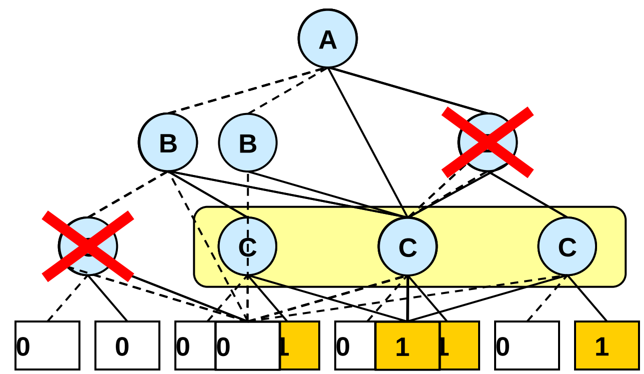
A	B	C	f(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table





Decision tree



1) Merge nodes with identical children

[Bryant86]

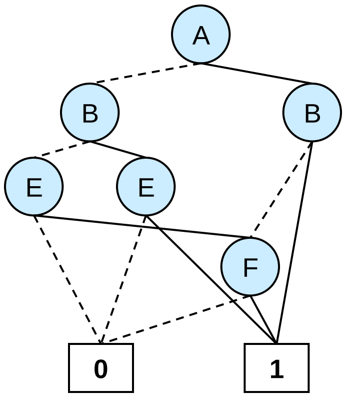
2) Remove redundant nodes

Ordering enables efficient operations

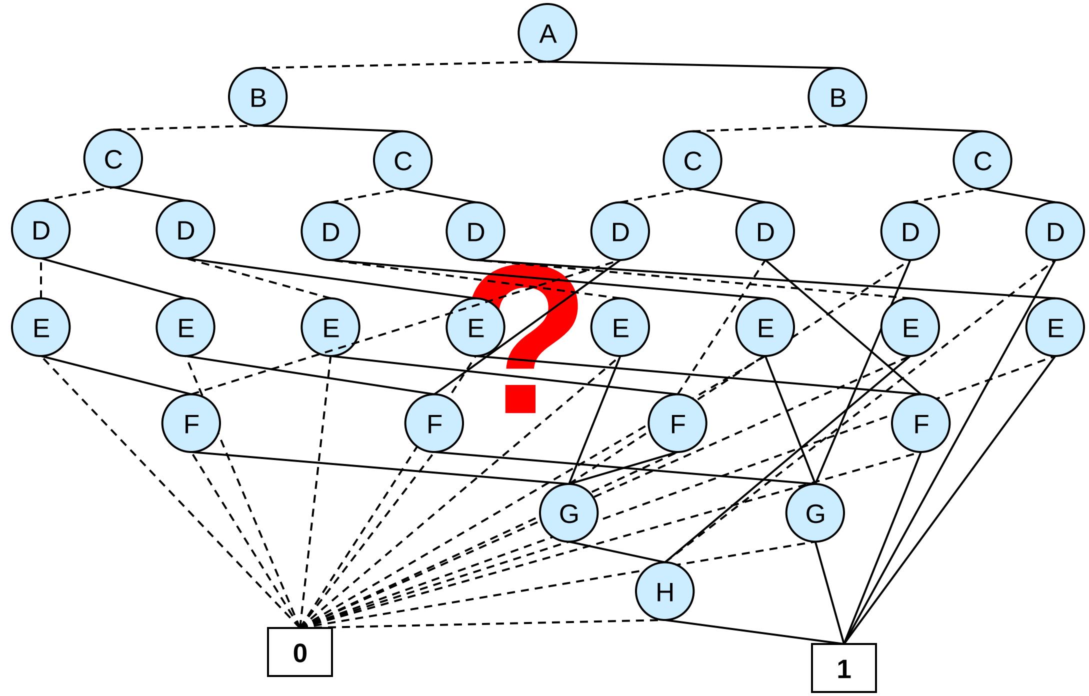
Decision Diagrams

f = (A ∨ E) ∧ (B ∨ F)

OBDD

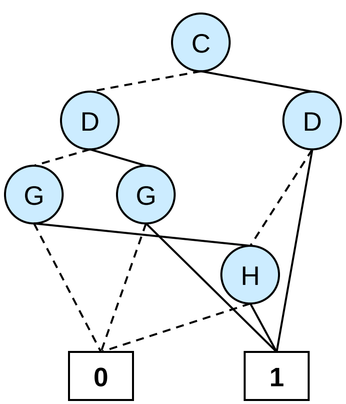


f ∧ g =

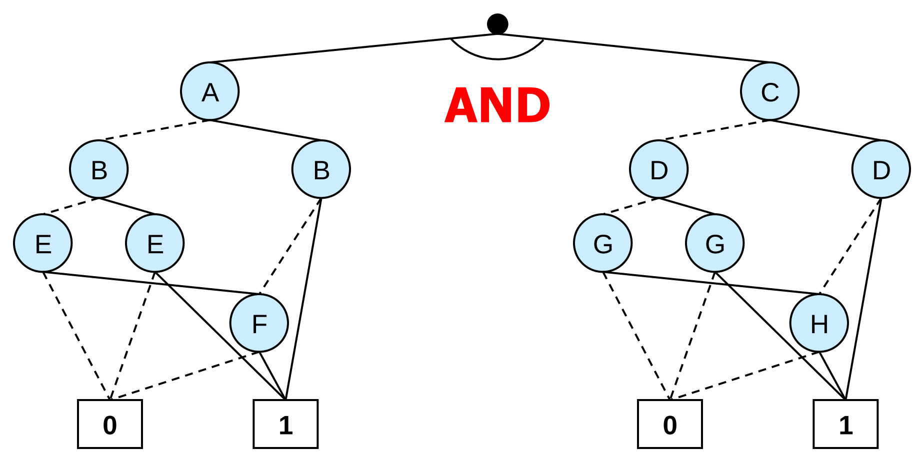


g = (C ∨ G) ∧ (D ∨ H)

OBDD

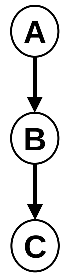


f ∧ g =



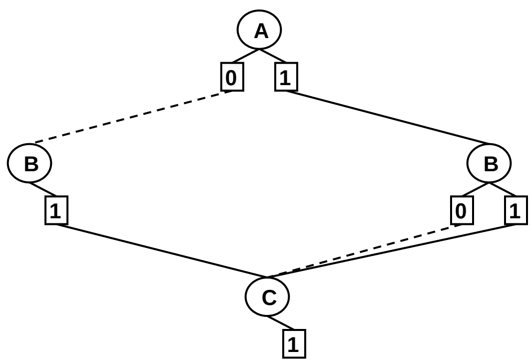
AOMDDs

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

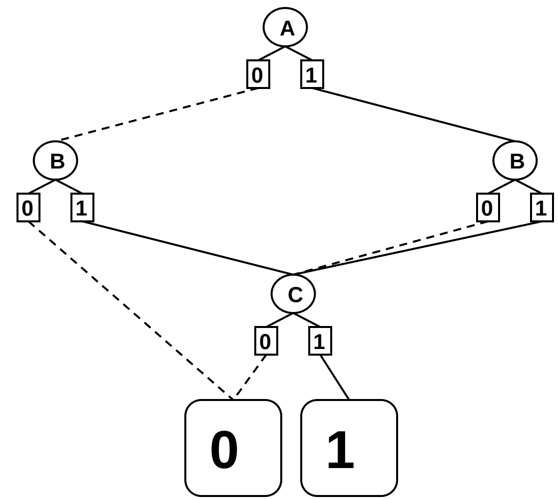


Point dead-ends to terminal node "0"

Point goods to terminal node "1"



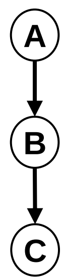
Minimal AND/OR graph



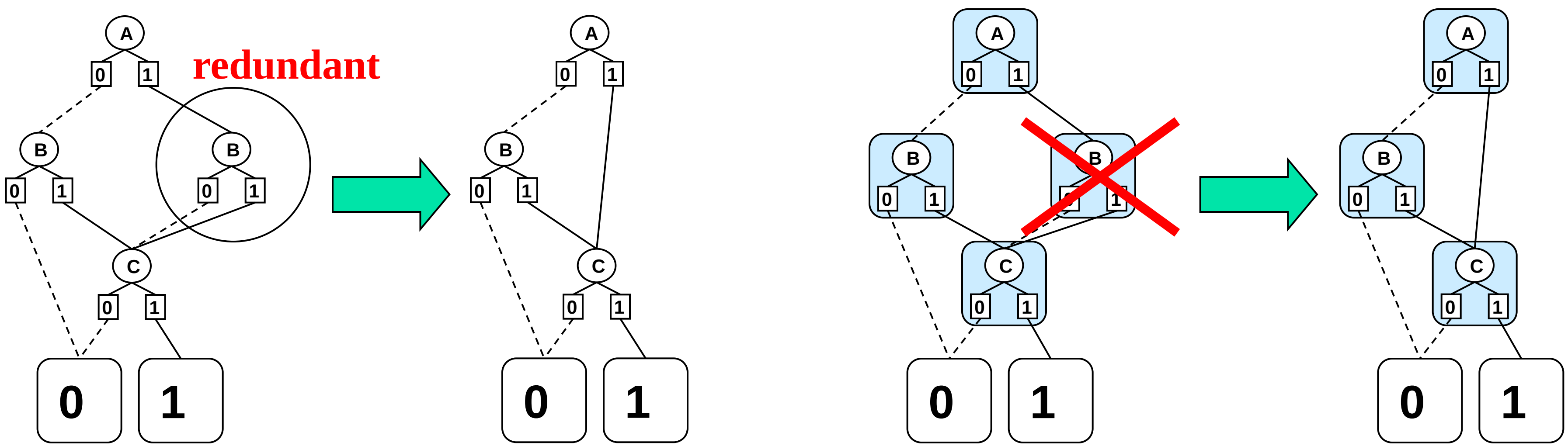
Decision Diagram

Removing Redundancy

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



Group OR node together with its AND children into a **meta-node**

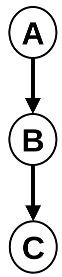


OBDD

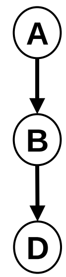
(pseudo tree is a chain)

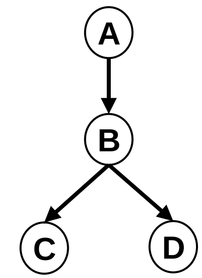
AOBDD

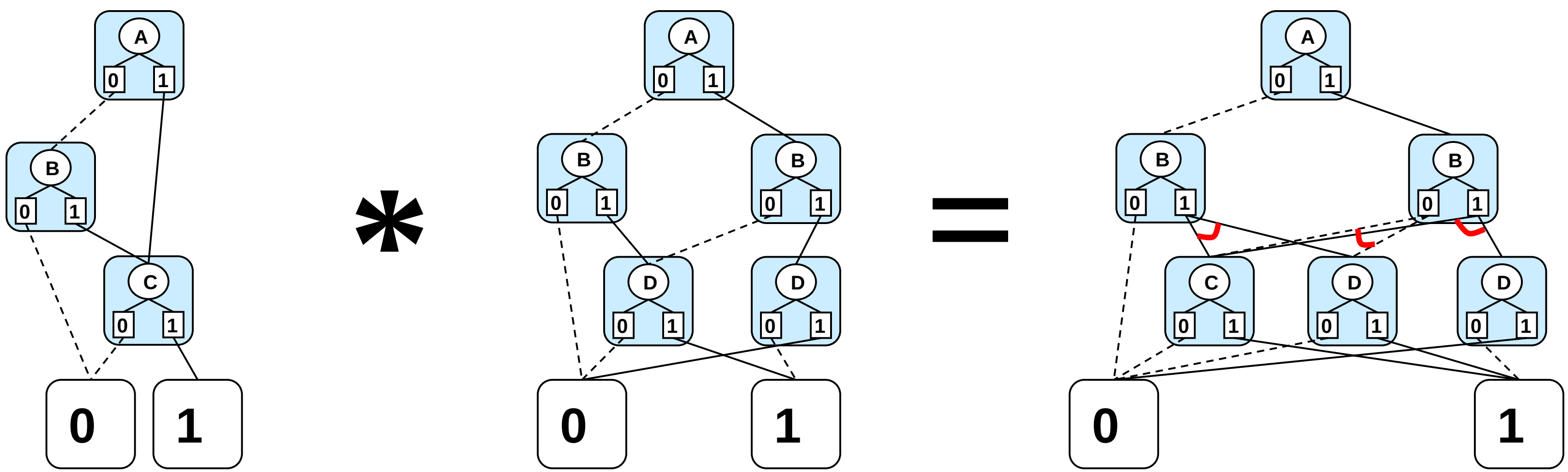
A	B	C	f(ABC)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



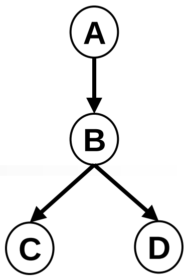
A	B	D	g(ABD)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

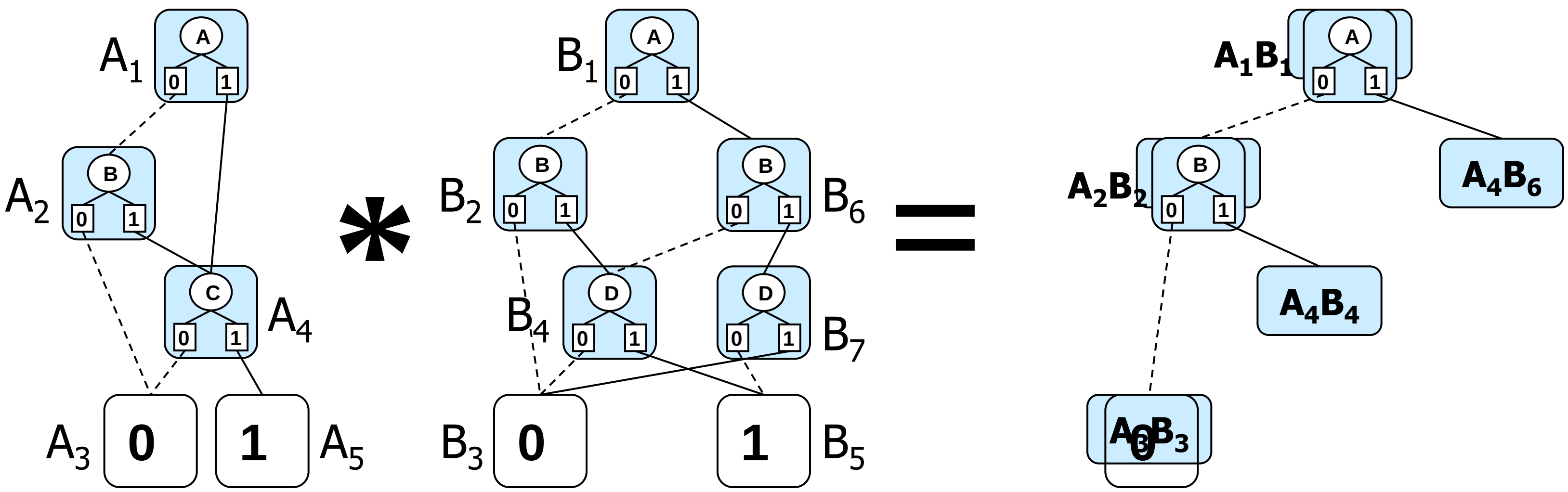




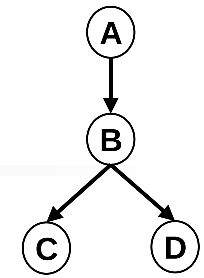


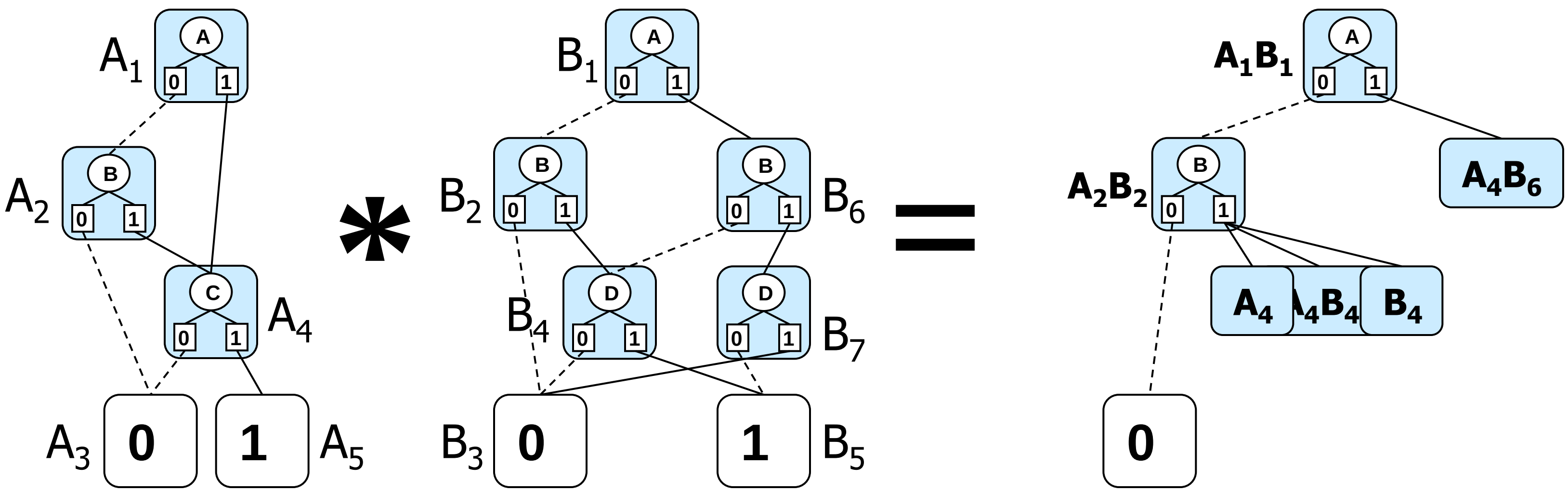
Apply Operator



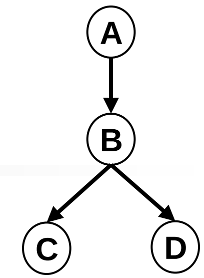


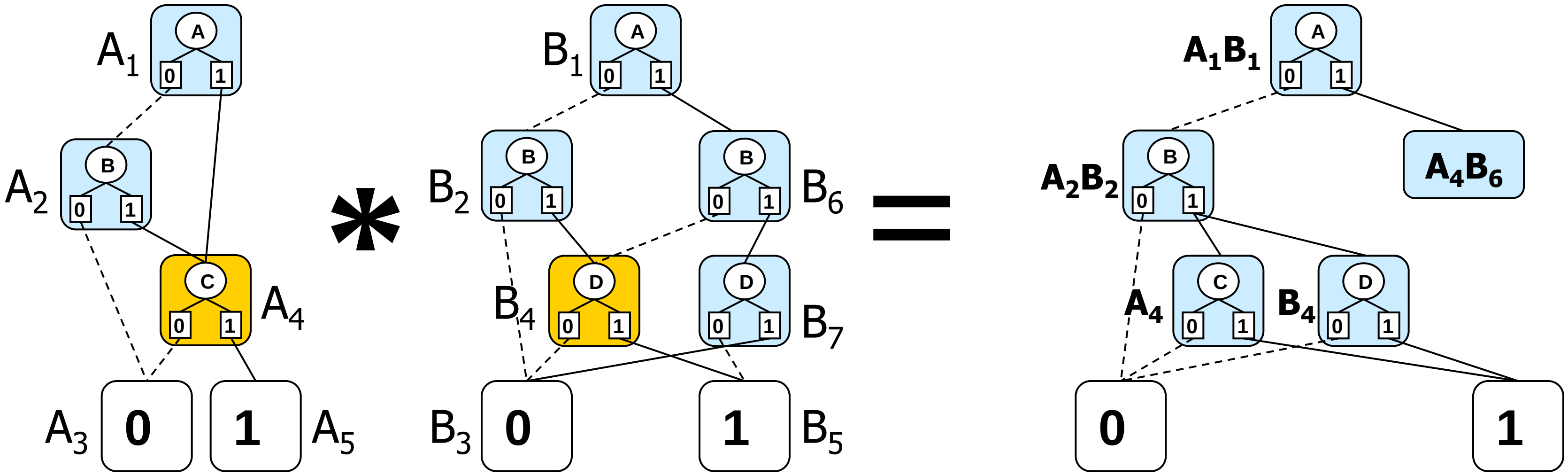
Apply Operator



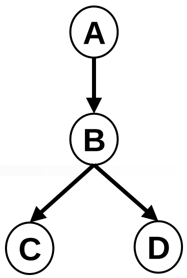


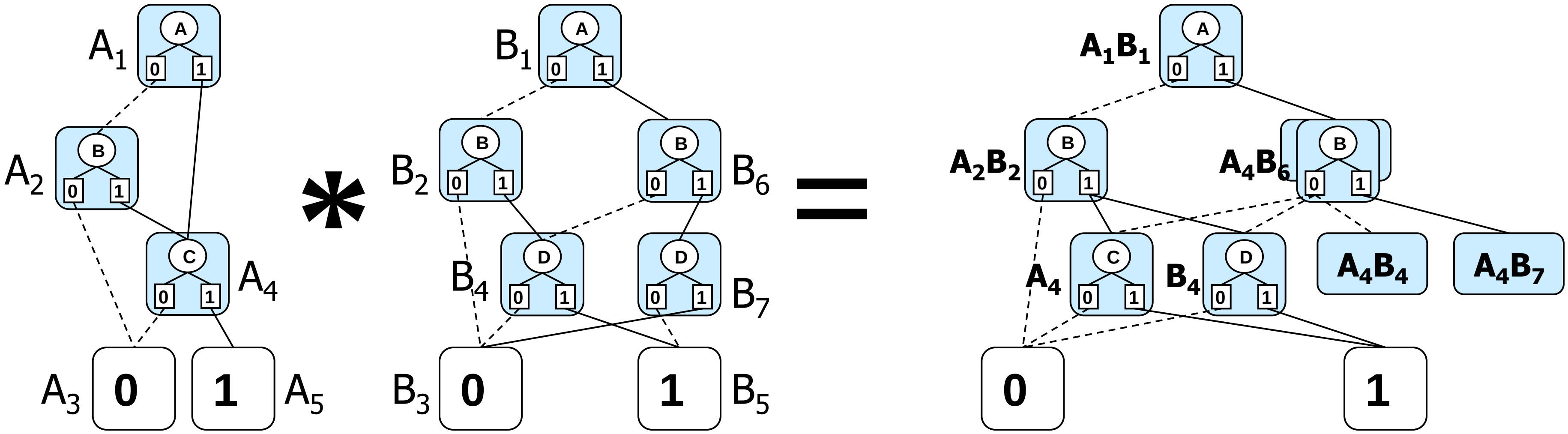
Apply Operator



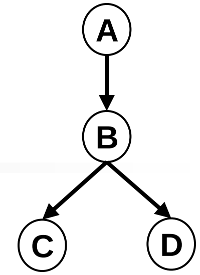


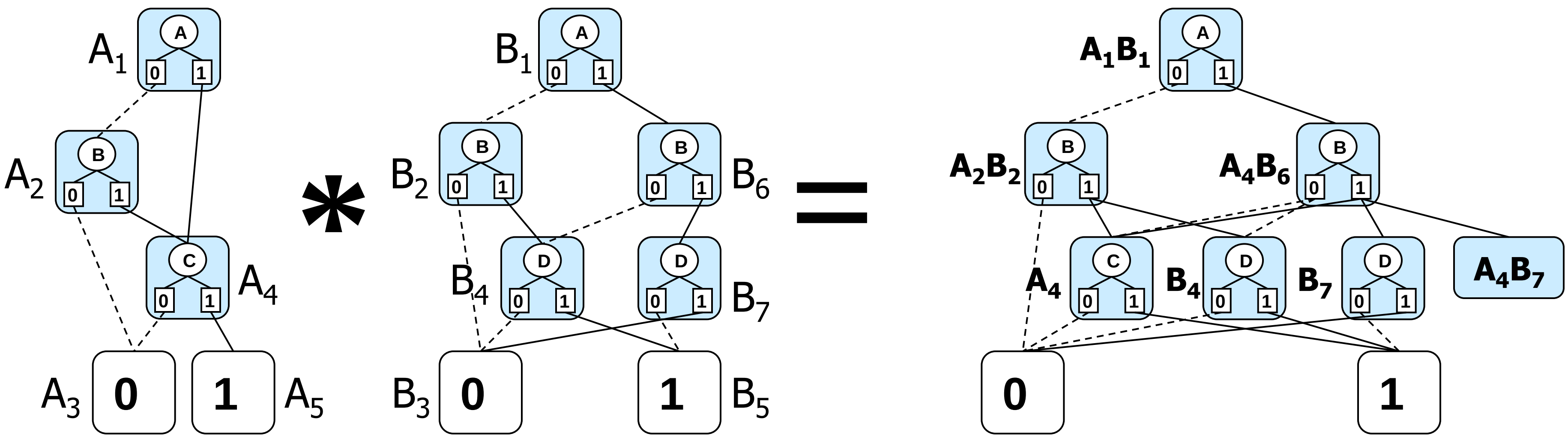
Apply Operator





Apply Operator

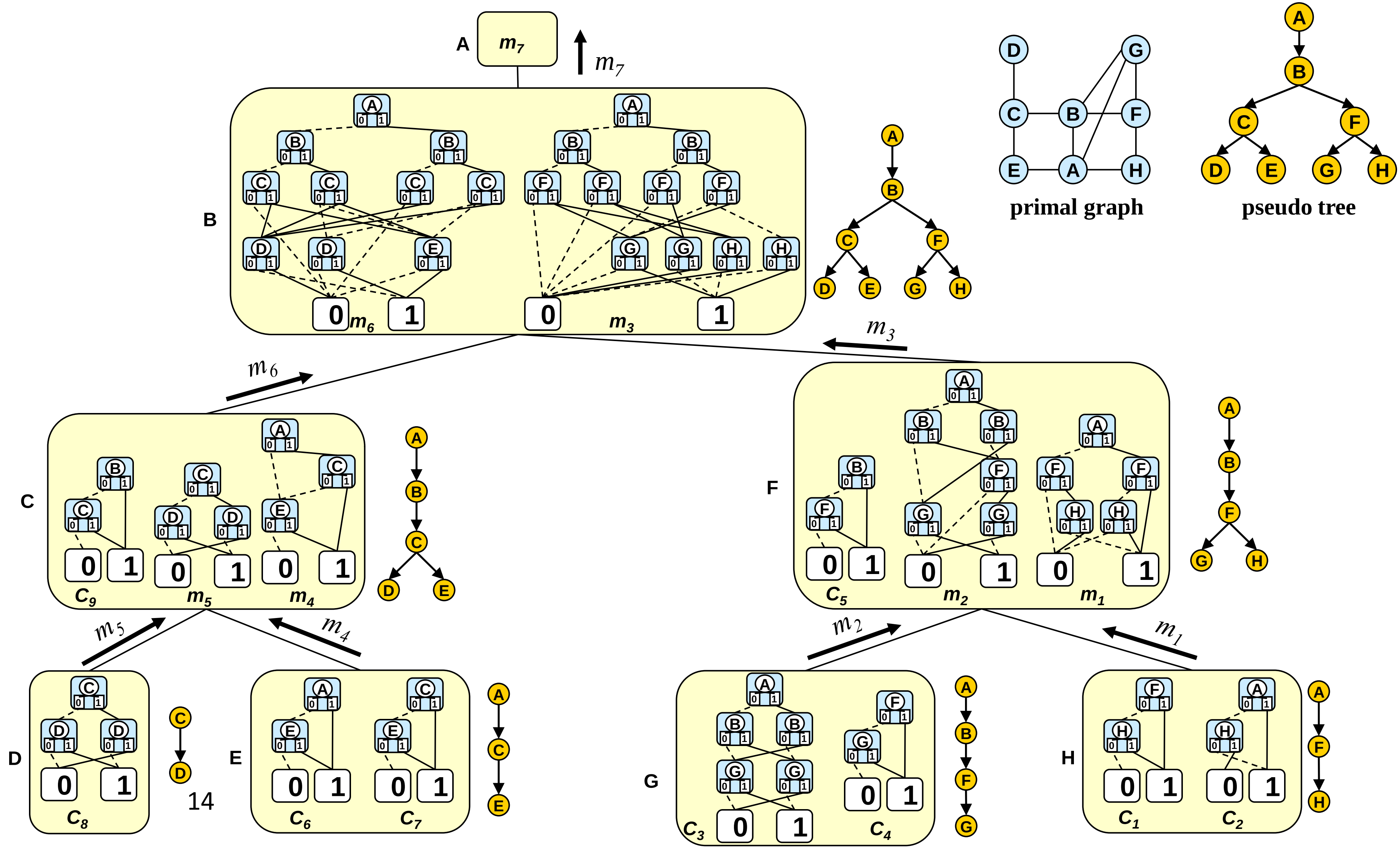




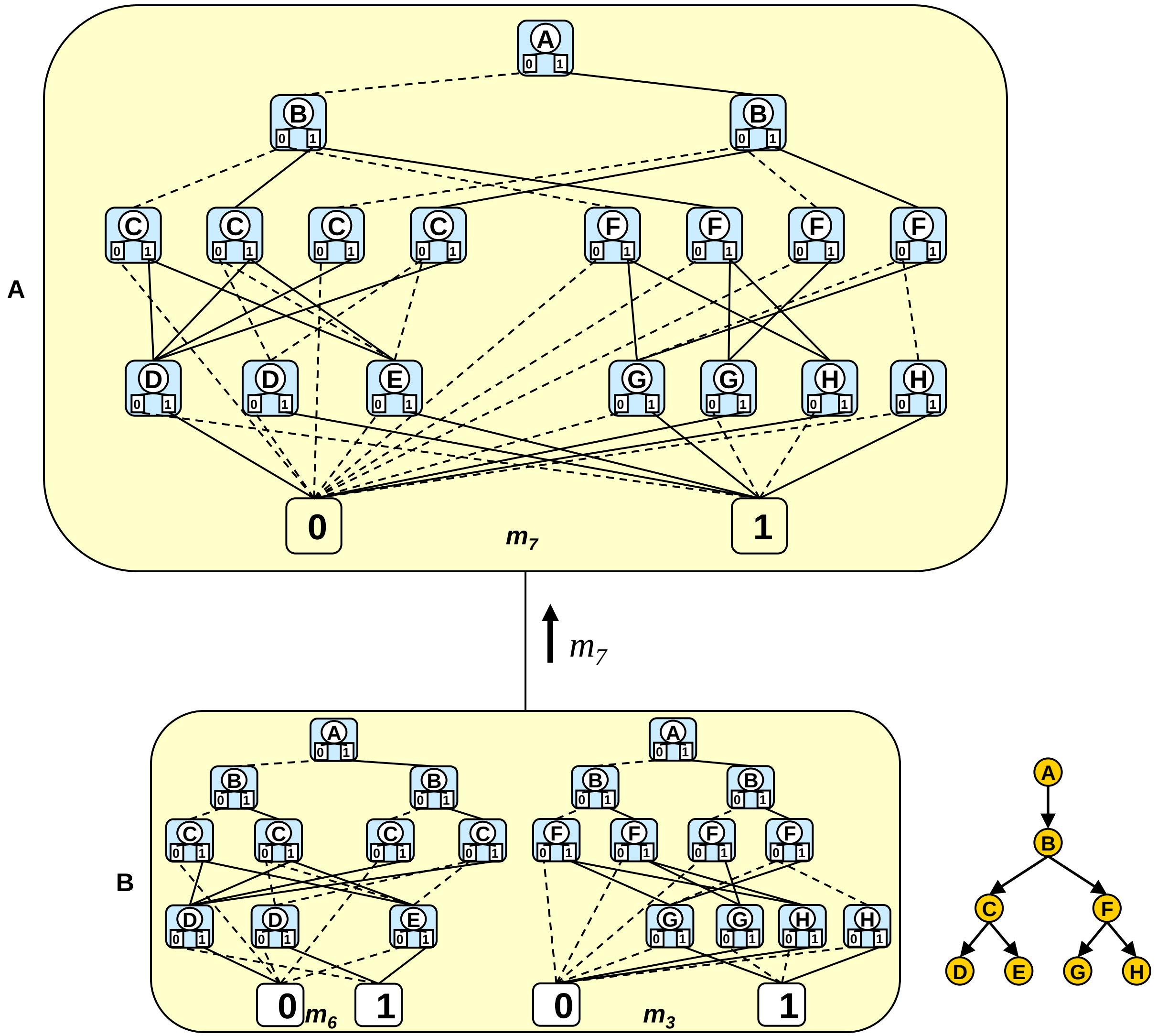
And/Or Multi-Valued Decision Diagrams

- AOMDDs are:
  - 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

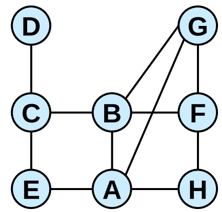
Example: $(f \lor h) \land (a \lor !h) \land (a\#b\#g) \land (f \lor g) \land (b \lor f) \land (a \lor e) \land (c \lor e) \land (c\#d) \land (b \lor c)$



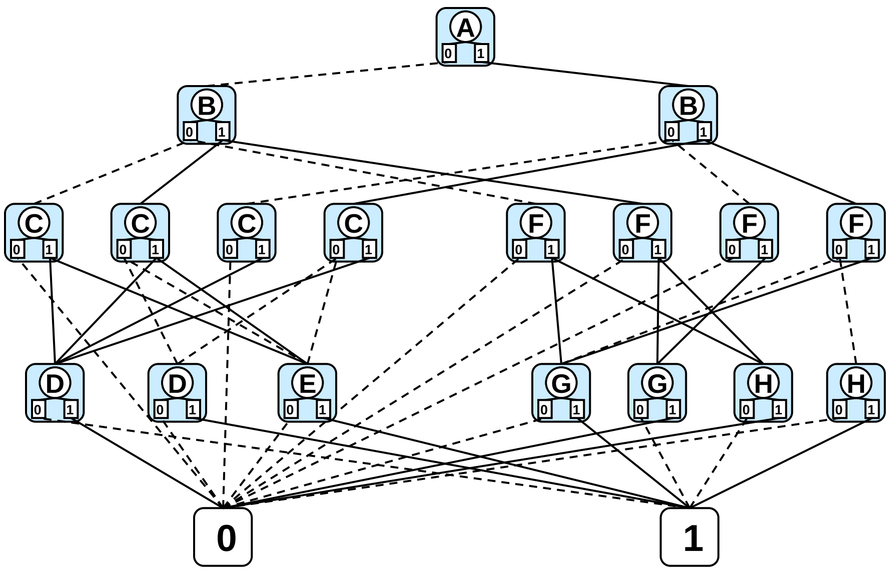
Example (continued)

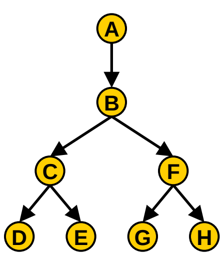


AOBDD vs. OBDD



primal graph

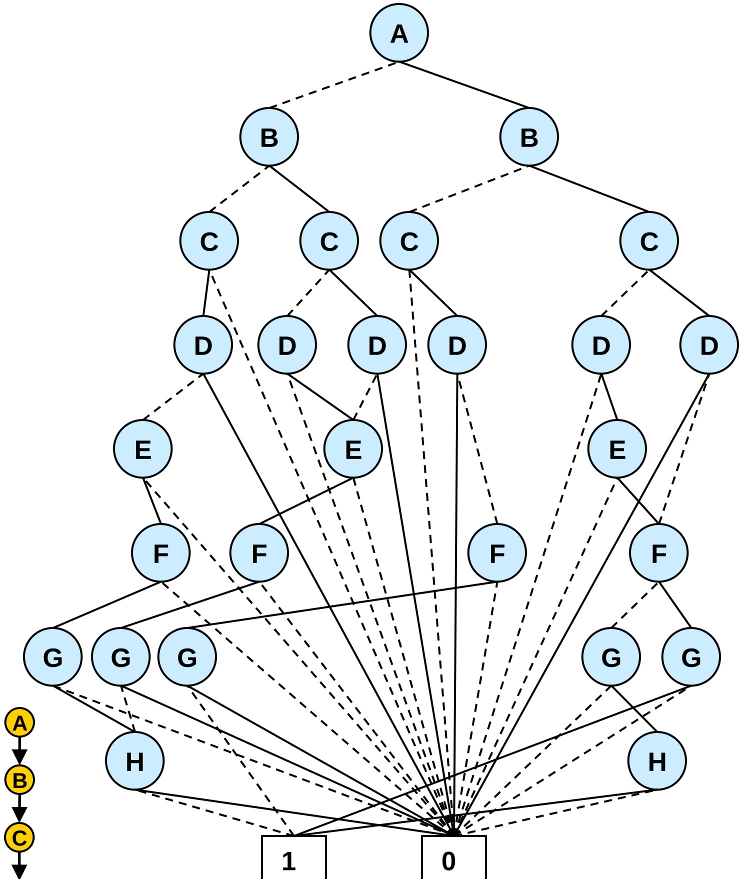


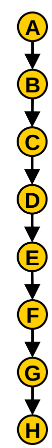


AOBDD

18 nonterminals

47 arcs





OBDD

27 nonterminals

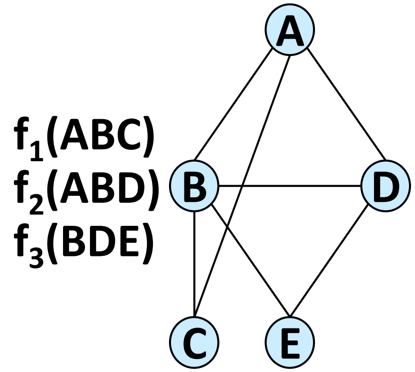
54 arcs

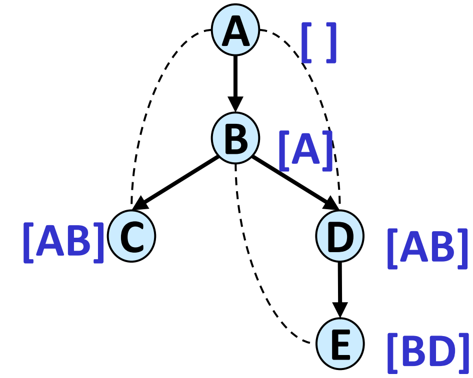
Constraint Optimization - AND/OR Tree

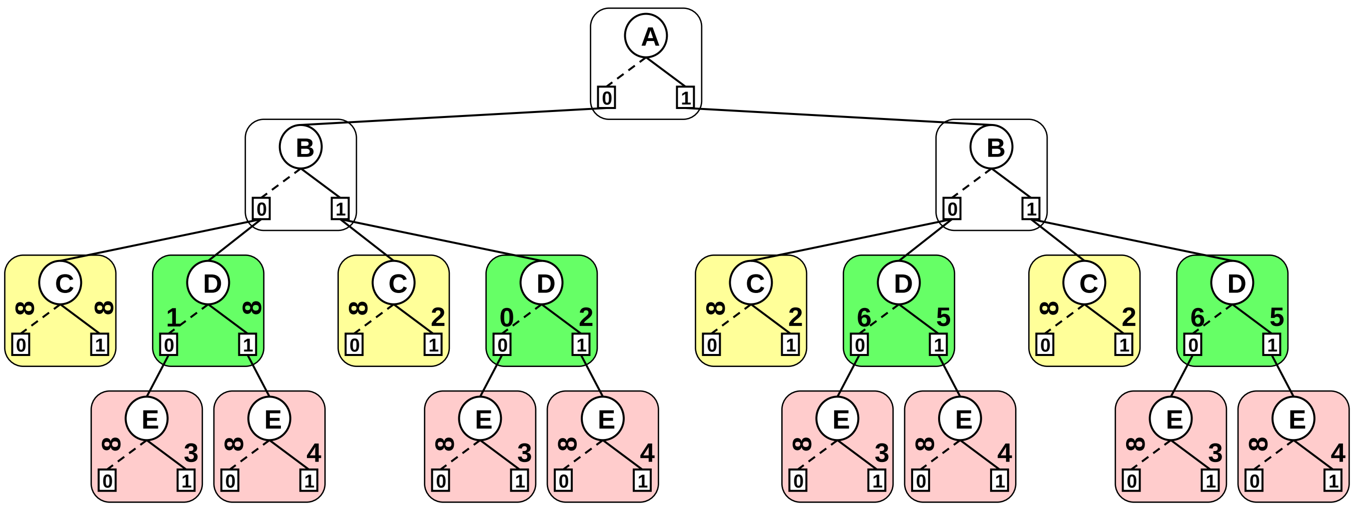
A	B	C	$f_1(ABC)$
0	0	0	∞
0	0	1	∞
0	1	0	∞
0	1	1	2
1	0	0	∞
1	0	1	2
1	1	0	∞
1	1	1	2

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







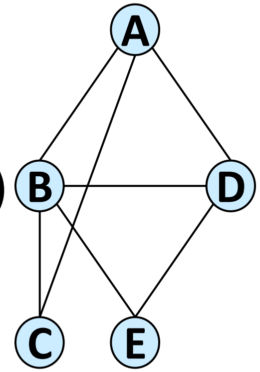
AND/OR Context Minimal Graph

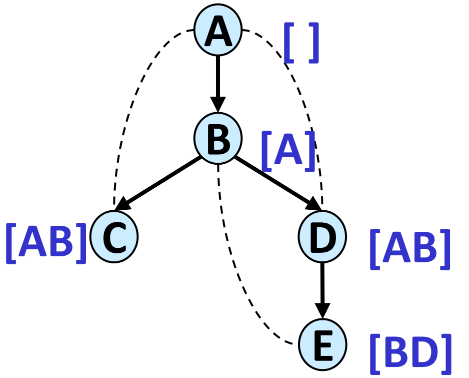
A	B	C	$f_1(ABC)$
0	0	0	∞
0	0	1	∞
0	1	0	∞
0	1	1	2
1	0	0	∞
1	0	1	2
1	1	0	∞
1	1	1	2

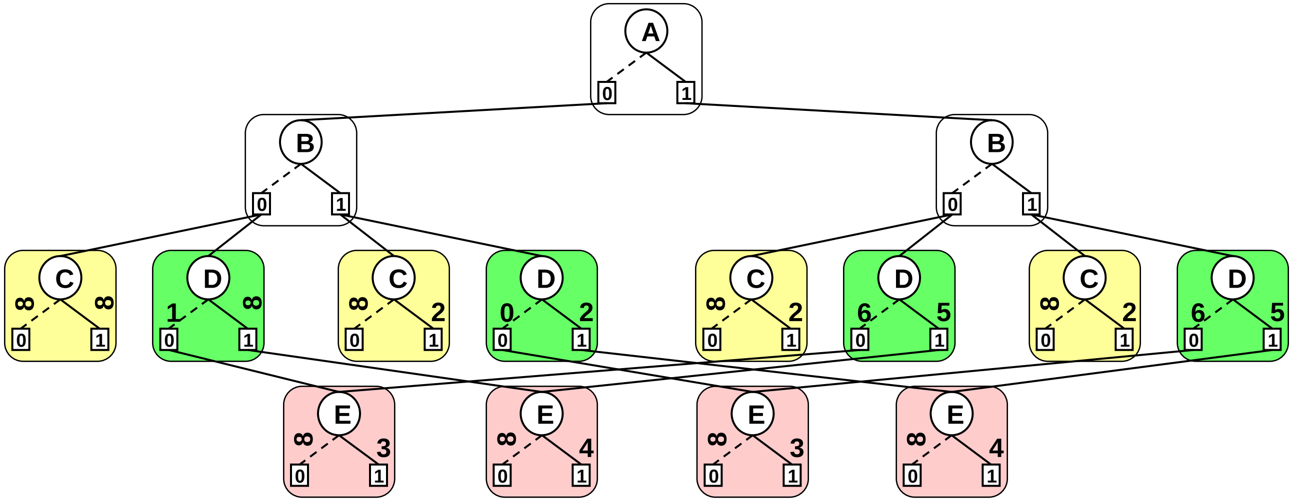
A	B	D	$f_2(ABD)$
0	0	0	1
0	0	1	∞
0	1	0	0
0	1	1	2
1	0	0	6
1	0	1	5
1	1	0	6
1	1	1	5

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

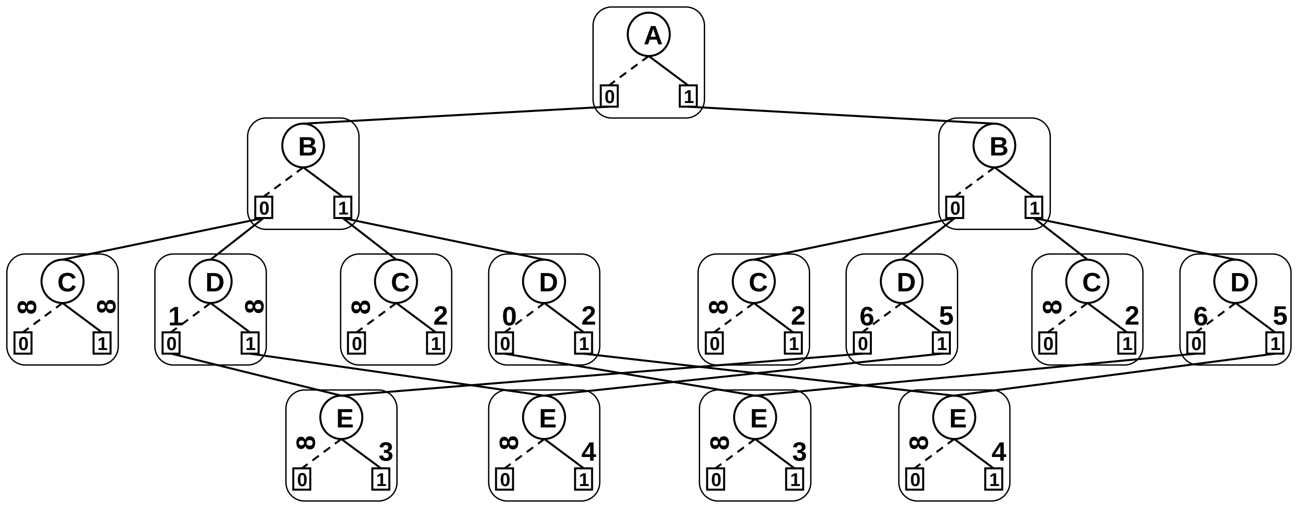
$f_1(ABC)$
$f_2(ABD)$
$f_3(BDE)$

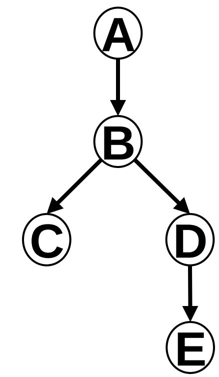


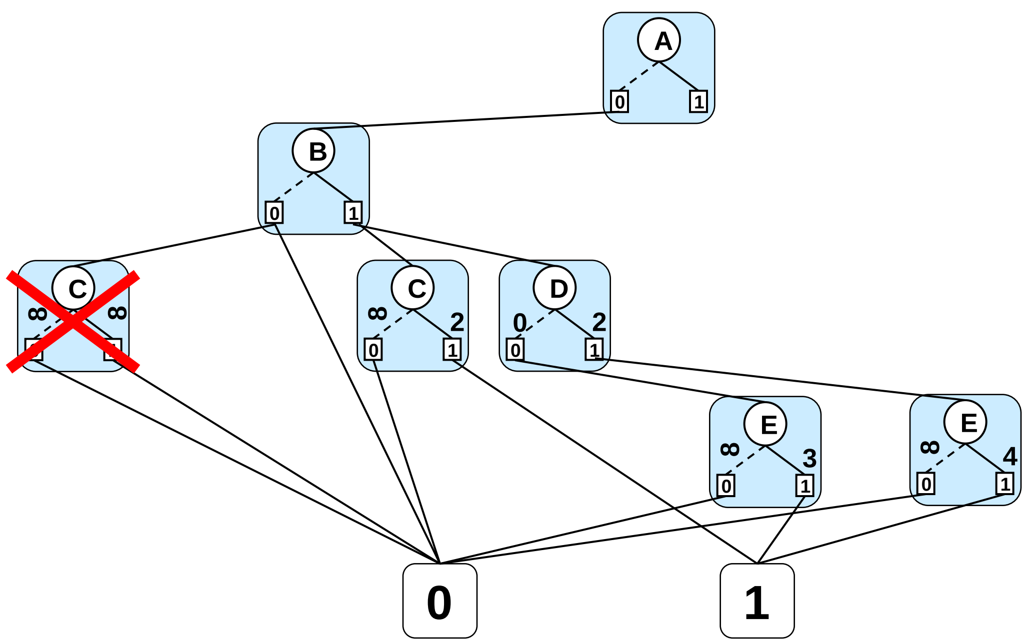




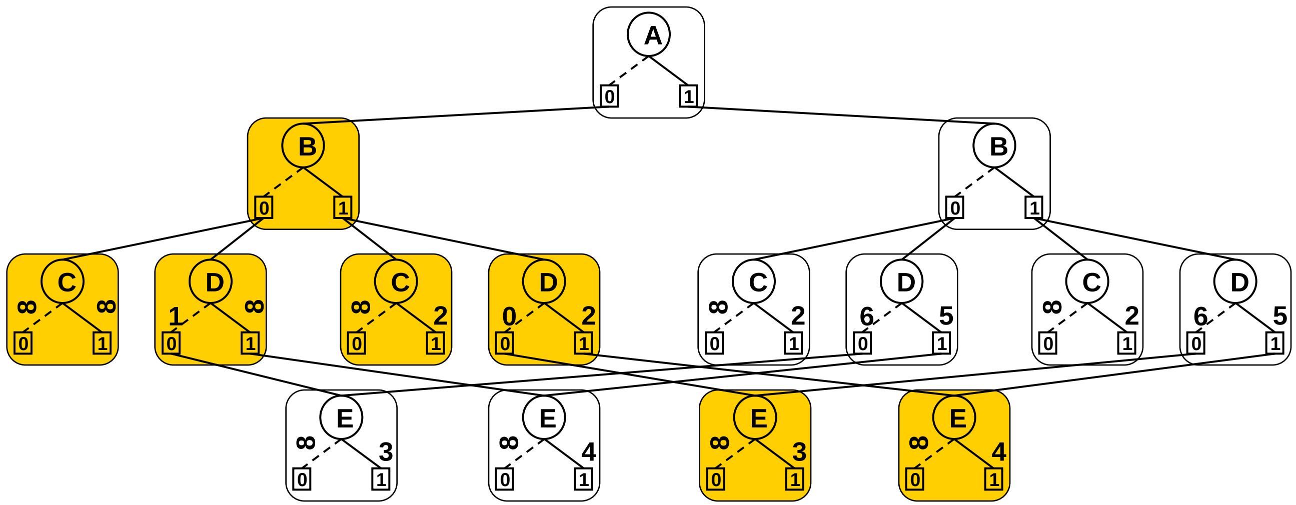
AOMDD – Compilation by Search

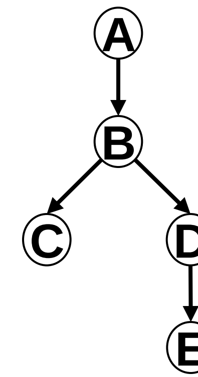




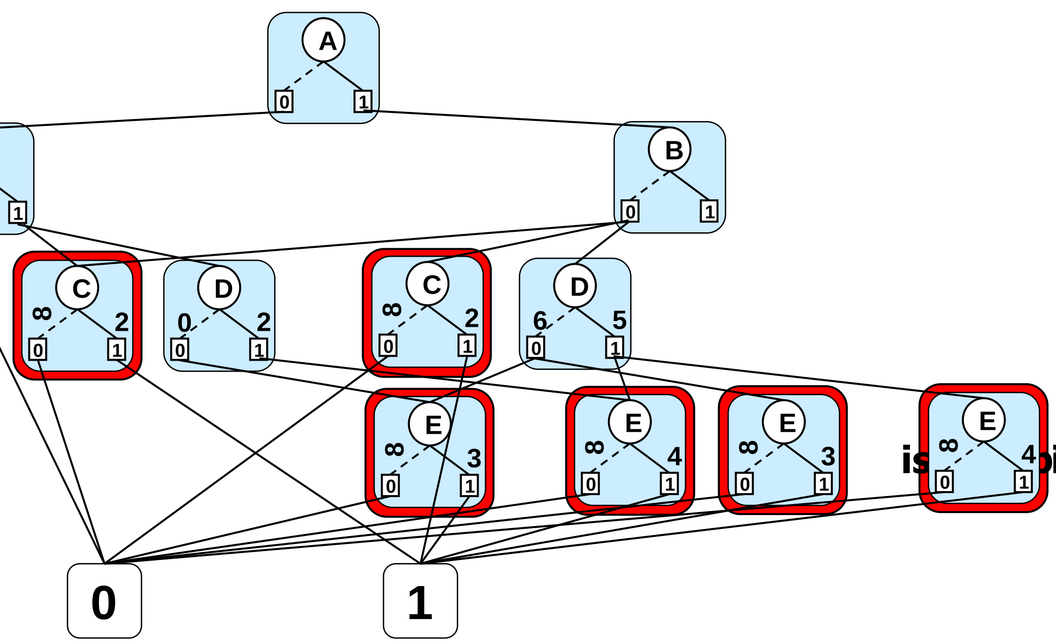


AOMDD – Compilation by Search



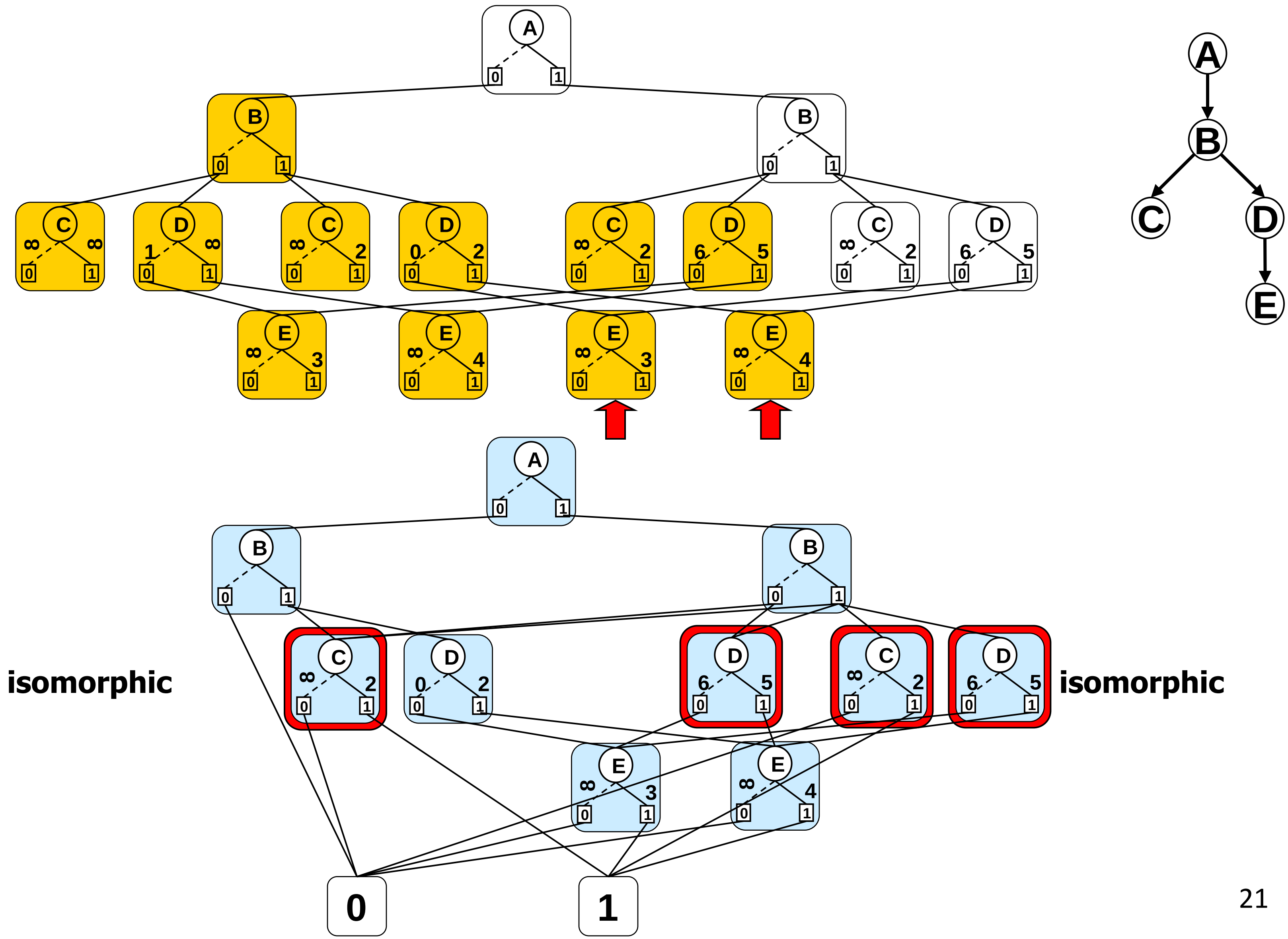


isomorphic

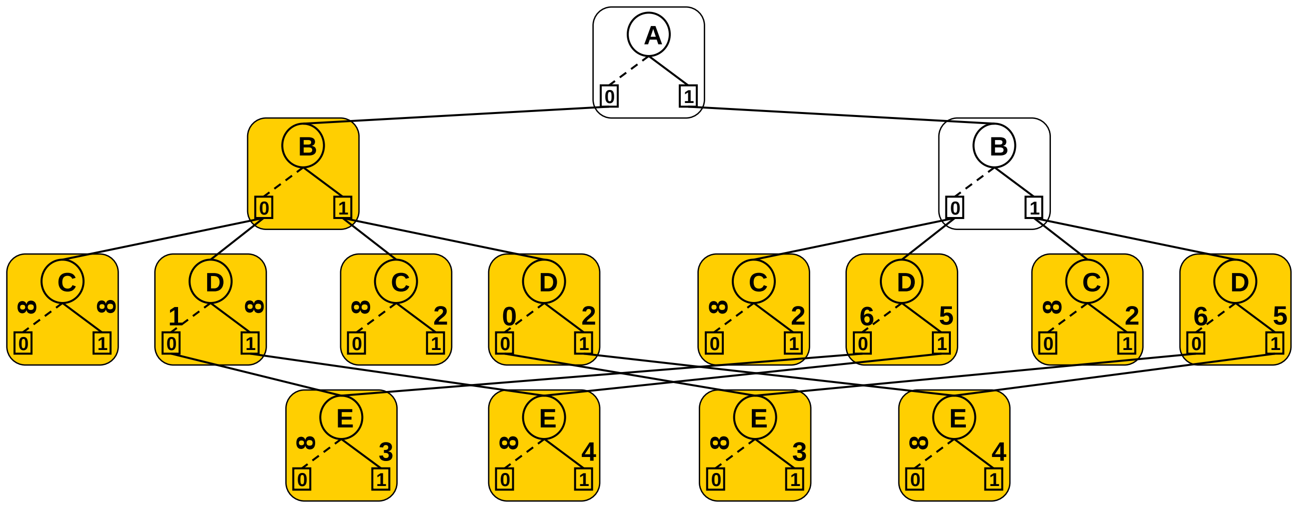


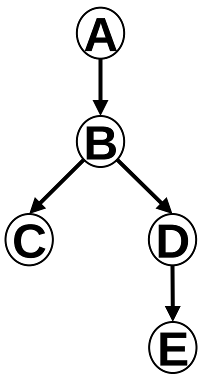
isomorphic

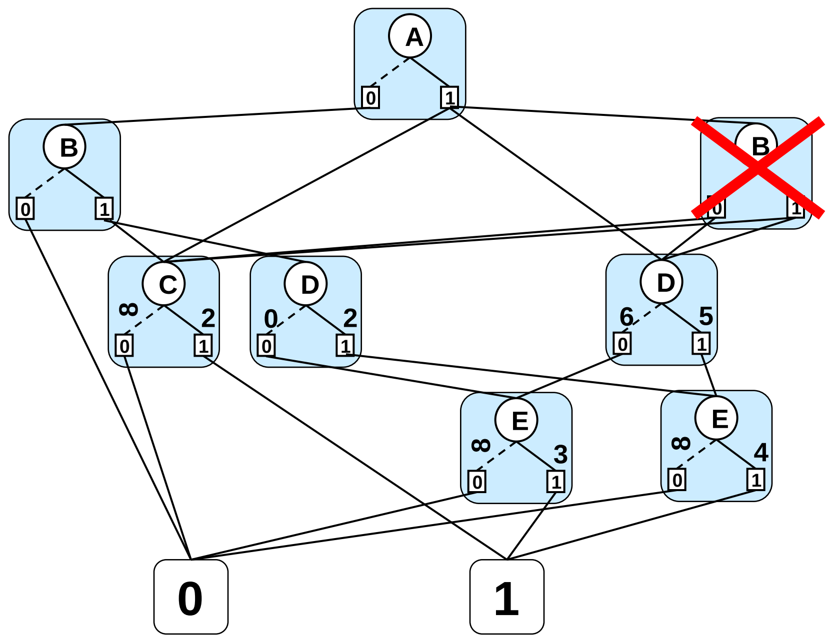
AOMDD – Compilation by Search



AOMDD – Compilation by Search

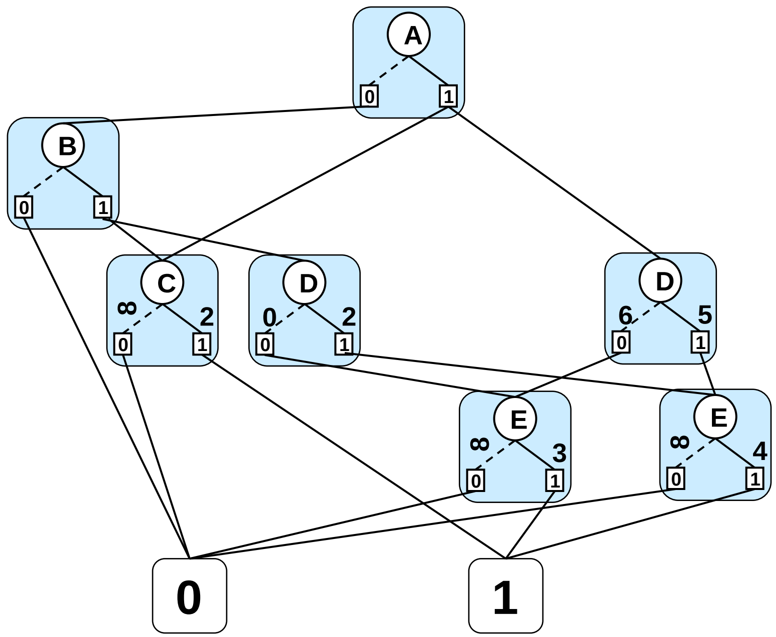




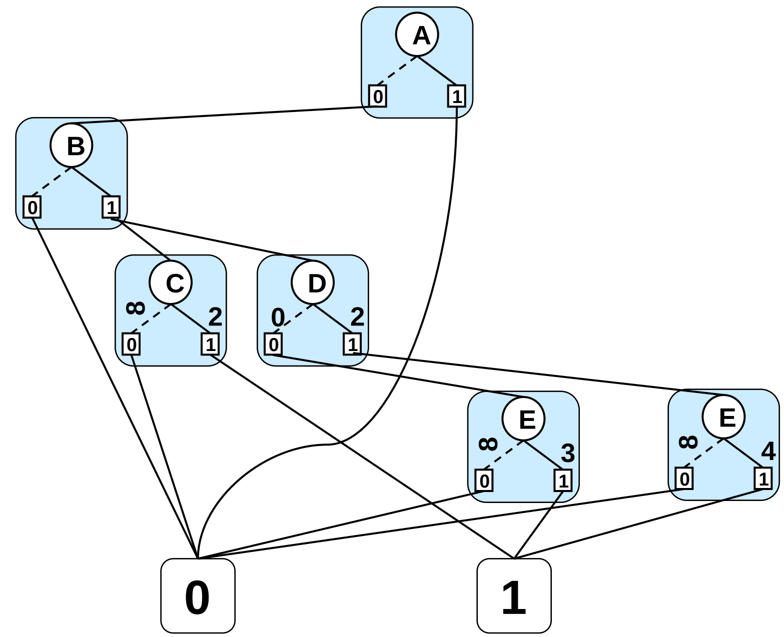


redundant

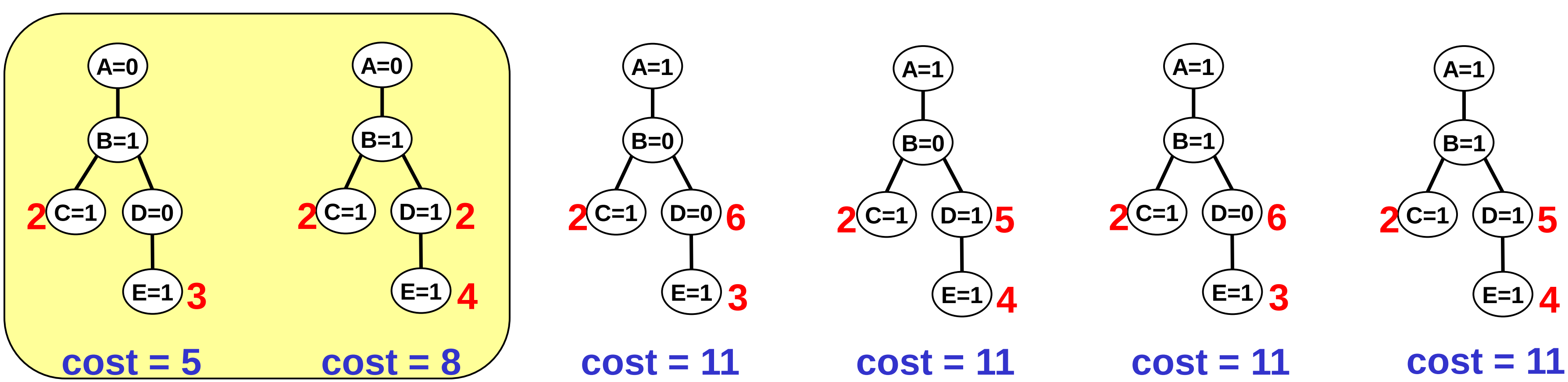
AOMDD for Constraint Optimization



AOMDD for all solutions



AOMDD for two best solutions



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

Semantic Treewidth

- Given a network, there may exist a sparser equivalent network.
- Challenges the idea of using induced width to measure the difficulty of the problem
- AOMDD sizes are much smaller than the bound

Semantic Treewidth

- With respect to a pseudo tree, this is the smallest treewidth over all equivalent networks that can have that pseudo tree
- With respect to the network, this is the smallest semantic treewidth over all pseudo trees that can express the set of solutions
- Instead of the induced width bounding AOMDD size, we can use semantic treewidth.

Semantic Treewidth

eq(n,10)	graph	c	w*	h	time	#aomdd
n=10	chain	9	1	5	0.0240	**91**
	complete	45	9	9	0.0660	**91**
n=50	chain	49	1	25	0.1420	**491**
	complete	1225	49	49	1.1130	**491**
n=100	chain	99	1	50	0.3120	**991**
	complete	4950	99	99	5.5900	**991**

Equality constraint network
results

Constraint Propagation

- We can also prune the search space during compilation without removing possible solutions.
- In Bayesian networks, prune a subtree if the weight of the assignment is 0.

Experiments

- What about the pseudo-tree height parameter?
- Problems: WCSP instances
  - ISCAS 89 Circuits
  - SPOT5 Satellites
  - Mastermind
  - CELAR6 Radio Frequencies
- Time bound for compilation: 3 hours

Experiments

- Compilation was for finding the optimal solution
- Used AOBB with static mini-bucket heuristics (i-bound = 10)
- Tried different implementations of MinFill
  - Existing implementation in the compiler
  - daoopt (gets lower h because it considers it too)
  - CVO

Experiments

- BnB pruning makes the size unpredictable as a function of the parameters
- Need to modify the routine for solution counting so the entire AOMDD is actually compiled

Experiments (BN)

- Reproduce and extend BN results in JAIR 2008 paper
- UAI 2006 Bayesian network benchmarks
  - Domain sizes of 2
  - Evidence on 30 random variables (to simplify the networks slightly)
  - Many elements with "0" support
- Compile with constraint propagation

To do

- Perform experiments to compare optimization vs. full compilation
  - Need to extend code for WCSPs
- Another way to deal with unpredictability of w/h vs. size in optimization?
  - Compute many orderings with equal w/h and average the search space/AOMDD sizes.

Future Work?

- Try regressing curves that depend on w, h, or both
- Evaluation of bottom-up version of compilation (Robert’s algorithm in CP 2006)